В. Дюк, А. Самойленко

Data Mining

учебный курс

CD-ROM прилагается

Новые технологии нового века

ПИТЕР

В. Дюк
А. Самойленко

Data Mining

Если вы хотите овладеть новыми технологиями, серия

учебный курс

это то, что вам нужно для быстрого обучения и продуктивной работы!

Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний (закономерностей), необходимых для принятия решений в различных сферах человеческой деятельности. Практически все крупнейшие корпорации активно принимают участие в разработке Data Mining.

В книге приводится объективный аналитический обзор методов и программных продуктов Data Mining. Подробно рассматриваются статистические пакеты, нейросети, эволюционные методы и алгоритмы поиска логических закономерностей. Описываются наиболее популярные инструментальные средства, разбираются практические примеры применения Data Mining.

Книга может быть использована в качестве учебника для студентов вузов.

На прилагаемом компакт-диске вы найдете учебные версии основных упоминаемых в книге программных продуктов и исходные данные для рассматриваемых примеров.

ISBN 5-318-00227-7

ПИТЕР®
WWW.PITER.COM

СЕРИЯ

учебный курс

ПИТЕР®

В. Дюк
А. Самойленко

Data Mining

учебный курс

Санкт-Петербург
Москва • Харьков • Минск
2001

ПИТЕР®

В. Дюк, А. Самойленко

Data Mining: учебный курс

Главный редактор	*Е. Строганова*
Заведующий редакцией	*И. Корнеев*
Руководитель проекта	*А. Пасечник*
Научный редактор	*А. Пасечник*
Литературный редактор	*М. Жданова*
Художник	*Н. Биржаков*
Корректоры	*С. Беляева, Н. Рощина*
Верстка	*Л. Чернышова*

ББК 32.973.233я7
УДК 681.3.01(075)

Дюк В., Самойленко. А.
Д95 Data mining: учебный курс (+CD). — СПб: Питер, 2001. — 368 с.: ил.

ISBN 5-318-00227-7

Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний (закономерностей).

В книге приводится объективный аналитический обзор методов и программных продуктов Data Mining. Подробно рассматриваются статистические пакеты, нейросети, эволюционные методы и алгоритмы поиска логических закономерностей. Описываются наиболее популярные инструментальные средства Data Mining. Разбираются практические примеры.

Для студентов, аспирантов, разработчиков интеллектуальных систем и широкой аудитории читателей, интересующихся проблемами анализа данных.

ISBN 5-318-00227-7

ЗАО «Питер Бук». 196105, Санкт-Петербург, Благодатная ул., д. 67.
Лицензия ИД № 01940 от 05.06.00.

Налоговая льгота – общероссийский классификатор продукции ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 13.07.01. Формат 70×100$^{1}/_{16}$. Усл. п. л. 29,67. Тираж 5000 экз. Заказ № 1160.

Отпечатано с готовых диапозитивов в ФГУП «Печатный двор» им. А. М. Горького
Министерства РФ по делам печати, телерадиовещания и средств массовых коммуникаций.
197110, Санкт-Петербург, Чкаловский пр., 15.

Краткое содержание

Содержание

От авторов

В связи с совершенствованием технологий записи и хранения данных на людей обрушились колоссальные потоки информационной руды в самых различных областях. Деятельность любого предприятия (коммерческого, производственного, медицинского, научного и т. д.) теперь сопровождается регистрацией и записью всех подробностей его деятельности. Без продуктивной переработки потоки сырых данных образуют никому не нужную свалку.

Специфика современных требований к такой переработке следующая:

- данные имеют неограниченный объем;
- данные являются разнородными (количественными, качественными, текстовыми);
- результаты должны быть конкретны и понятны;
- инструменты для обработки сырых данных должны быть просты в использовании.

Книга посвящена современной технологии обработки данных — интеллектуальному анализу данных (Data Mining). Data Mining — это процесс обнаружения в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний (закономерностей), необходимых для принятия решений в различных сферах человеческой деятельности. Практически все крупнейшие корпорации активно принимают участие в разработке Data Mining.

В книге приводится объективный аналитический обзор методов и программных продуктов Data Mining. Подробно рассматриваются разделы Data Mining: статистические пакеты, нейросети, эволюционные методы и алгоритмы поиска логических закономерностей. Описываются наиболее популярные инструментальные средства Data Mining. Разбираются практические примеры применения Data Mining.

Авторы выражают глубокую признательность за ценные советы директору Санкт-Петербургского института информатики и автоматизации РАН доктору технических наук, профессору Р. М. Юсупову и главному научному сотруднику этого института доктору технических наук, профессору Р. И. Полонникову.

Отдельные разделы книги по просьбе авторов подготовлены нашими коллегами. Это глава 3 (д. техн. наук, проф. А. В. Тимофеев, канд. техн. наук А. А. Богданов, канд. физ.мат. наук З. М Шибзухов) и глава 4 (С. Сотник, А. Кун). Авторы благодарны коллегам за оперативный отклик и качественный материал.

Книга адресована студентам, аспирантам, разработчикам интеллектуальных систем и широкой аудитории читателей, интересующихся вычислительной техникой, информатикой и проблемами компьютеризации.

Сопроводительный компакт-диск

К книге прилагается компакт-диск, на котором содержатся демонстрационные и свободные для распространения версии основных упоминаемых программных продуктов. Ограничения, наложенные на демо-версии, позволяют работать с ними неограниченное время. Они касаются, главным образом, объемов обрабатываемой информации и некоторых функций, не имеющих принципиального значения для освоения упомянутых программ.

Кроме того, компакт-диск содержит различные данные, которые используются при рассмотрении практических примеров в главах книги и в приложении. Эти данные представлены в виде таблиц Excel.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция) или непосредственно одному из авторов книги: v_duke@spiiras.nw.ru.

Мы будем рады узнать ваше мнение!

Подробную информацию о наших книгах вы найдете на Web-сайте издательства http://www.piter.com.

Общие представления о Data Mining

Что такое Data Mining?

Data Mining переводится как «добыча» или «раскопка данных». Нередко рядом с Data Mining встречаются слова «обнаружение знаний в базах данных» (knowledge discovery in databases) и «интеллектуальный анализ данных». Их можно считать синонимами Data Mining. Возникновение всех указанных терминов связано с новым витком в развитии средств и методов обработки данных.

До начала 90-х годов, казалось, не было особой нужды переосмысливать ситуацию в этой области. Все шло своим чередом в рамках направления, называемого прикладной статистикой (см., например, [1]). Теоретики проводили конференции и семинары, писали внушительные статьи и монографии, изобиловавшие аналитическими выкладками.

Вместе с тем, практики всегда знали, что попытки применить теоретические экзерсисы для решения реальных задач в большинстве случаев оказываются бесплодными. Но на озабоченность практиков до поры до времени можно было не обращать особого внимания — они решали, главным образом, свои частные проблемы обработки небольших локальных баз данных.

И вот прозвенел звонок. В связи с совершенствованием технологий записи и хранения данных на людей обрушились колоссальные потоки информационной руды в самых различных областях. Деятельность любого предприятия (коммерческого, производственного, медицинского, научного и т. д.) теперь сопровождается регистрацией и записью всех подробностей его деятельности. Что делать с этой информацией? Стало ясно, что без продуктивной переработки потоки сырых данных образуют никому не нужную свалку.

Специфика современных требований к такой переработке такова:

- данные имеют неограниченный объем;
- данные являются разнородными (количественными, качественными, текстовыми);
- результаты должны быть конкретны и понятны;
- инструменты для обработки сырых данных должны быть просты в использовании.

Традиционная математическая статистика, долгое время претендовавшая на роль основного инструмента анализа данных, откровенно спасовала перед лицом возникших проблем. Главная причина — *концепция усреднения по выборке*, приводящая к операциям над фиктивными величинами (типа средней температуры пациентов по больнице, средней высоты дома на улице, состоящей из дворцов и лачуг, и т. п.). Методы математической статистики оказались полезными главным образом для проверки заранее сформулированных гипотез (verification-driven data mining) и для «грубого» разведочного анализа, составляющего основу оперативной аналитической обработки данных (online analytical processing, OLAP).

В основу современной технологии Data Mining (discovery-driven data mining) положена концепция шаблонов (паттернов), отражающих *фрагменты* многоаспектных взаимоотношений в данных. Эти шаблоны представляют собой закономерности, свойственные *подвыборкам данных*, которые могут быть компактно выражены в понятной человеку форме. Поиск шаблонов производится методами, не ограниченными рамками априорных предположений о структуре выборки и виде распределений значений анализируемых показателей. Примеры заданий на такой поиск при использовании Data Mining приведены в табл. 1.1.

Таблица 1.1. Примеры формулировок задач при использовании методов OLAP и Data Mining

OLAP	Data Mining
Каковы средние показатели травматизма для курящих и некурящих?	Встречаются ли точные шаблоны в описаниях людей, подверженных повышенному травматизму?
Каковы средние размеры телефонных счетов существующих клиентов в сравнении со счетами бывших клиентов (отказавшихся от услуг телефонной компании)?	Имеются ли характерные портреты клиентов, которые, по всей вероятности, собираются отказаться от услуг телефонной компании?
Какова средняя величина ежедневных покупок по украденной и не украденной кредитной карточке?	Существуют ли стереотипные схемы покупок для случаев мошенничества с кредитными карточками?

Важное положение Data Mining — нетривиальность разыскиваемых шаблонов. Это означает, что найденные шаблоны должны отражать неочевидные, неожиданные (unexpected) регулярности в данных, составляющие так называемые скрытые знания (hidden knowledge). К обществу пришло понимание того, что сырые данные (raw data) содержат глубинный пласт знаний, при грамотной раскопке которого могут быть обнаружены настоящие самородки (рис. 1.1).

В целом технологию Data Mining достаточно точно определяет Григорий Пиатецкий-Шапиро — один из основателей этого направления.

Data Mining — это процесс обнаружения в сырых данных:

- ранее неизвестных;
- нетривиальных;
- практически полезных;
- доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Рис. 1.1. Уровни знаний, извлекаемых из данных

Кому это нужно?

Сфера применения Data Mining ничем не ограничена — она везде, где имеются какие-либо данные. Но в первую очередь методы Data Mining сегодня, мягко говоря, заинтриговали коммерческие предприятия, развертывающие проекты на основе информационных хранилищ данных (Data Warehousing). Опыт многих таких предприятий показывает, что отдача от использования Data Mining может достигать 1000 %. Например, известны сообщения об экономическом эффекте, в 10–70 раз превысившем первоначальные затраты от $350 до $750 тыс. [3]. Известны сведения о проекте в $20 млн, который окупился всего за 4 месяца. Другой пример — годовая экономия $700 тыс. за счет внедрения Data Mining в сети универсамов в Великобритании.

Data Mining представляют большую ценность для руководителей и аналитиков в их повседневной деятельности. Деловые люди осознали, что с помощью методов Data Mining они могут получить ощутимые преимущества в конкурентной борьбе. Кратко охарактеризуем некоторые возможные бизнес-приложения Data Mining [2].

Некоторые бизнес-приложения Data Mining

Розничная торговля

Предприятия розничной торговли сегодня собирают подробную информацию о каждой отдельной покупке, используя кредитные карточки с маркой магазина и компьютеризованные системы контроля. Вот типичные задачи, которые можно решать с помощью Data Mining в сфере розничной торговли:

- *Анализ покупательской корзины* (анализ сходства) предназначен для выявления товаров, которые покупатели стремятся приобретать вместе. Знание по-

купательской корзины необходимо для улучшения рекламы, выработки стратегии создания запасов товаров и способов их раскладки в торговых залах.

- *Исследование временных шаблонов* помогает торговым предприятиям принимать решения о создании товарных запасов. Оно дает ответы на вопросы типа: «Если сегодня покупатель приобрел видеокамеру, то через какое время он вероятнее всего купит новые батарейки и пленку?».
- *Создание прогнозирующих моделей* дает возможность торговым предприятиям узнавать характер потребностей различных категорий клиентов с определенным поведением, например, покупающих товары известных дизайнеров или посещающих распродажи. Эти знания нужны для разработки точно направленных, экономичных мероприятий по продвижению товаров.

Банковское дело

Достижения технологии Data Mining используются в банковском деле для решения следующих распространенных задач:

- *Выявление мошенничества с кредитными карточками.* Путем анализа прошлых транзакций, которые впоследствии оказались мошенническими, банк выявляет некоторые стереотипы такого мошенничества.
- *Сегментация клиентов.* Разбивая клиентов на различные категории, банки делают свою маркетинговую политику более целенаправленной и результативной, предлагая различные виды услуг разным группам клиентов.
- *Прогнозирование изменений клиентуры.* Data Mining помогает банкам строить прогнозные модели ценности своих клиентов и соответствующим образом обслуживать каждую категорию.

Телекоммуникации

В области телекоммуникаций методы Data Mining помогают компаниям более энергично продвигать свои программы маркетинга и ценообразования, чтобы удерживать существующих клиентов и привлекать новых. Среди типичных мероприятий отметим следующие:

- *Анализ записей о подробных характеристиках вызовов.* Назначение такого анализа — выявление категорий клиентов с похожими стереотипами пользования их услугами и разработка привлекательных наборов цен и услуг.
- *Выявление лояльности клиентов.* Data Mining можно использовать для определения характеристик клиентов, которые, один раз воспользовавшись услугами данной компании, с большой долей вероятности останутся ей верными. В итоге средства, выделяемые на маркетинг, можно тратить там, где отдача больше всего.

Страхование

Страховые компании в течение ряда лет накапливают большие объемы данных. Здесь обширное поле деятельности для методов Data Mining:

- *Выявление мошенничества.* Страховые компании могут снизить уровень мошенничества, отыскивая определенные стереотипы в заявлениях о выплате страхового возмещения, характеризующих взаимоотношения между юристами, врачами и заявителями.
- *Анализ риска.* Путем выявления сочетаний факторов, связанных с оплаченными заявлениями, страховщики могут уменьшить свои потери по обязательствам. Известен случай, когда в США крупная страховая компания обнаружила, что суммы, выплаченные по заявлениям людей, состоящих в браке, вдвое превышают суммы по заявлениям одиноких людей. Компания отреагировала на это новое знание пересмотром своей общей политики предоставления скидок семейным клиентам.

Другие приложения в бизнесе

Data Mining может применяться во множестве других областей:

- *Развитие автомобильной промышленности.* При сборке автомобилей производители должны учитывать требования каждого отдельного клиента, поэтому им нужны возможность прогнозирования популярности определенных характеристик и знание того, какие характеристики обычно заказываются вместе.
- *Политика гарантий.* Производителям нужно предсказывать число клиентов, которые подадут гарантийные заявки, и среднюю стоимость заявок.
- *Поощрение часто летающих клиентов.* Авиакомпании могут обнаружить группу клиентов, которых данными поощрительными мерами можно побудить летать больше. Например, одна авиакомпания обнаружила категорию клиентов, которые совершали много полетов на короткие расстояния, не накапливая достаточно миль для вступления в их клубы, поэтому она таким образом изменила правила приема в клуб, чтобы поощрять число полетов так же, как и мили.

Специальные приложения

Медицина

Известно много экспертных систем для постановки медицинских диагнозов. Они построены главным образом на основе правил, описывающих сочетания различных симптомов различных заболеваний. С помощью таких правил узнают не только, чем болен пациент, но и как нужно его лечить. Правила помогают выбирать средства медикаментозного воздействия, определять показания (противопоказания), ориентироваться в лечебных процедурах, создавать условия наиболее эффективного лечения, предсказывать исходы назначенного курса лечения и т. п. Технологии Data Mining позволяют обнаруживать в медицинских данных шаблоны, составляющие основу указанных правил.

Молекулярная генетика и генная инженерия

Пожалуй, наиболее остро и вместе с тем четко задача обнаружения закономерностей в экспериментальных данных стоит в молекулярной генетике и генной инженерии. Здесь она формулируется как определение так называемых маркеров, под которыми понимают генетические коды, контролирующие те или иные фенотипические признаки живого организма. Такие коды могут содержать сотни, тысячи и более связанных элементов.

На развитие генетических исследований выделяются большие средства. В последнее время в данной области возник особый интерес к применению методов Data Mining. Известно несколько крупных фирм, специализирующихся на применении этих методов для расшифровки генома человека и растений.

Прикладная химия

Методы Data Mining находят широкое применение в прикладной химии (органической и неорганической). Здесь нередко возникает вопрос о выяснении особенностей химического строения тех или иных соединений, определяющих их свойства. Особенно актуальна такая задача при анализе сложных химических соединений, описание которых включает сотни и тысячи структурных элементов и их связей.

Можно привести еще много примеров различных областей знания, где методы Data Mining играют ведущую роль. Особенность этих областей заключается в их сложной системной организации. Они относятся главным образом к надкибернетическому уровню организации систем [4], закономерности которого не могут быть достаточно точно описаны на языке статистических или иных аналитических математических моделей [5]. Данные в указанных областях неоднородны, гетерогенны, нестационарны и часто отличаются высокой размерностью.

Типы закономерностей

Выделяют пять стандартных типов закономерностей, которые позволяют выявлять методы Data Mining:

- ассоциация;
- последовательность;
- классификация;
- кластеризация;
- прогнозирование.

Ассоциация имеет место в том случае, если несколько событий связаны друг с другом. Например, исследование, проведенное в супермаркете, может показать, что 65 % купивших кукурузные чипсы берут также и «кока-колу», а при наличии скидки за такой комплект «колу» приобретают в 85 % случаев. Располагая сведениями о подобной ассоциации, менеджерам легко оценить, насколько действенна предоставляемая скидка.

Если существует цепочка связанных во времени событий, то говорят о *последовательности*. Так, например, после покупки дома в 45 % случаев в течение месяца приобретается и новая кухонная плита, а в пределах двух недель 60 % новоселов обзаводятся холодильником.

С помощью *классификации* выявляются признаки, характеризующие группу, к которой принадлежит тот или иной объект. Это делается посредством анализа уже классифицированных объектов и формулирования некоторого набора правил.

Кластеризация отличается от классификации тем, что сами группы заранее не заданы. С помощью кластеризации средства Data Mining самостоятельно выделяют различные однородные группы данных.

Основой для всевозможных систем *прогнозирования* служит историческая информация, хранящаяся в БД в виде временных рядов. Если удается найти шаблоны, адекватно отражающие динамику поведения целевых показателей, есть вероятность, что с их помощью можно предсказать и поведение системы в будущем.

Классы систем Data Mining

Data Mining является мультидисциплинарной областью, возникшей и развивающейся на базе достижений прикладной статистики, распознавания образов, методов искусственного интеллекта, теории баз данных и др. Отсюда обилие методов и алгоритмов, реализованных в различных действующих системах Data Mining. Многие из таких систем интегрируют в себе сразу несколько подходов. Тем не менее, как правило, в каждой системе имеется какой-то ключевой компонент, на который делается главная ставка. Ниже приводится классификация указанных ключевых компонентов на основе работы [6]. Выделенным классам дается краткая характеристика.

Предметно-ориентированные аналитические системы

Предметно-ориентированные аналитические системы очень разнообразны. Наиболее широкий подкласс таких систем, получивший распространение в области исследования финансовых рынков, носит название «технический анализ». Он представляет собой совокупность нескольких десятков методов прогноза динамики цен и выбора оптимальной структуры инвестиционного портфеля, основанных на различных эмпирических моделях динамики рынка. Эти методы часто используют несложный статистический аппарат, но максимально учитывают сложившуюся в своей области специфику (профессиональный язык, системы различных индексов и пр.). На рынке имеется множество программ этого класса. Как правило, они довольно дешевы (обычно $300–$1000).

Статистические пакеты

Последние версии почти всех известных статистических пакетов включают наряду с традиционными статистическими методами также элементы Data Mining. Но основное внимание в них уделяется все же классическим методикам — корреляционному, регрессионному, факторному анализу и др. Самый свежий детальный обзор пакетов для статистического анализа приведен на страницах Центрального экономико-математического института http://is1.cemi.rssi.ru/ruswin/publication/ep97001t.htm.

Недостатком систем этого класса считают требование к специальной подготовке пользователя. Также отмечают, что мощные современные статистические пакеты являются слишком «тяжеловесными» для массового применения в финансах и бизнесе. К тому же часто эти системы весьма дороги — от $1000 до $15 000.

Есть еще более серьезный принципиальный недостаток статистических пакетов, ограничивающий их применение в Data Mining. Большинство методов, входящих в состав пакетов, опираются на статистическую парадигму, в которой главными фигурантами служат усредненные характеристики выборки. А эти характеристики, как указывалось выше, при исследовании реальных сложных жизненных феноменов часто являются фиктивными величинами.

В качестве примеров наиболее мощных и распространенных статистических пакетов можно назвать SAS (компания SAS Institute), SPSS (SPSS), STATGRAPICS (Manugistics), STATISTICA, STADIA и др.

Нейронные сети

Это большой класс систем, архитектура которых имеет аналогию (как теперь известно, довольно слабую) с построением нервной ткани из нейронов. В одной из наиболее распространенных архитектур, многослойном перцептроне с обратным распространением ошибки, имитируется работа нейронов в составе иерархической сети, где каждый нейрон более высокого уровня соединен своими входами с выходами нейронов нижележащего слоя. На нейроны самого нижнего слоя подаются значения входных параметров, на основе которых нужно принимать какие-то решения, прогнозировать развитие ситуации и т. д. Эти значения рассматриваются как сигналы, передающиеся в следующий слой, ослабляясь или усиливаясь в зависимости от числовых значений (весов), приписываемых межнейронным связям. В результате на выходе нейрона самого верхнего слоя вырабатывается некоторое значение, которое рассматривается как ответ — реакция всей сети на введенные значения входных параметров. Для того чтобы сеть можно было применять в дальнейшем, ее прежде надо «натренировать» на полученных ранее данных, для которых известны и значения входных параметров, и правильные ответы на них. Тренировка состоит в подборе весов межнейронных связей, обеспечивающих наибольшую близость ответов сети к известным правильным ответам.

Основным недостатком нейросетевой парадигмы является необходимость иметь очень большой объем обучающей выборки. Другой существенный недостаток заключается в том, что даже натренированная нейронная сеть представляет собой «черный ящик». Знания, зафиксированные как веса нескольких сотен межнейронных связей, совершенно не поддаются анализу и интерпретации человеком (известные попытки дать интерпретацию структуре настроенной нейросети выглядят неубедительными – система «KINOsuite-PR»).

Примеры нейросетевых систем — BrainMaker (CSS), NeuroShell (Ward Systems Group), OWL (HyperLogic). Стоимость их довольно значительна: $1500–8000.

Системы рассуждений на основе аналогичных случаев

Идея систем case based reasoning — CBR — на первый взгляд крайне проста. Для того чтобы сделать прогноз на будущее или выбрать правильное решение, эти системы находят в прошлом близкие аналоги наличной ситуации и выбирают тот же ответ, который был для них правильным. Поэтому этот метод еще называют методом «ближайшего соседа» (nearest neighbour). В последнее время распространение получил также термин memory based reasoning, который акцентирует внимание на том, что решение принимается на основании всей информации, накопленной в памяти.

Системы CBR показывают неплохие результаты в самых разнообразных задачах. Главным их минусом считают то, что они вообще не создают каких-либо моделей или правил, обобщающих предыдущий опыт, — в выборе решения они основываются на всем массиве доступных исторических данных, поэтому невозможно сказать, на основе каких конкретно факторов CBR-системы строят свои ответы.

Другой минус заключается в произволе, который допускают системы CBR при выборе меры «близости». От этой меры самым решительным образом зависит объем множества прецедентов, которые нужно хранить в памяти для достижения удовлетворительной классификации или прогноза [7].

Примеры систем, использующих CBR, — KATE tools (Acknosoft, Франция), Pattern Recognition Workbench (Unica, США).

Деревья решений

Деревья решения (decision trees) являются одним из наиболее популярных подходов к решению задач Data Mining. Они создают иерархическую структуру классифицирующих правил типа «ЕСЛИ... ТО...» (if-then), имеющую вид дерева. Для принятия решения, к какому классу отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня. Вопросы имеют вид «значение параметра А больше х?». Если ответ положительный, осуществляется переход к правому узлу следующего уровня, если от-

рицательный — к левому узлу; затем снова следует вопрос, связанный с соответствующим узлом.

Популярность подхода связана как бы с наглядностью и понятностью. Но деревья решений принципиально не способны находить «лучшие» (наиболее полные и точные) правила в данных. Они реализуют наивный принцип последовательного просмотра признаков и «цепляют» фактически осколки настоящих закономерностей, создавая лишь иллюзию логического вывода.

Вместе с тем, большинство систем используют именно этот метод. Самыми известными являются See5/C5.0 (RuleQuest, Австралия, http://www.rulequest.com/), Clementine (Integral Solutions, Великобритания, http://www.spss.com/clementine/), SIPINA (University of Lyon, Франция), IDIS (Information Discovery, США), KnowledgeSEEKER (ANGOSS, Канада) (рис. 1.2). Стоимость этих систем варьируется от $1 до $10 тыс.

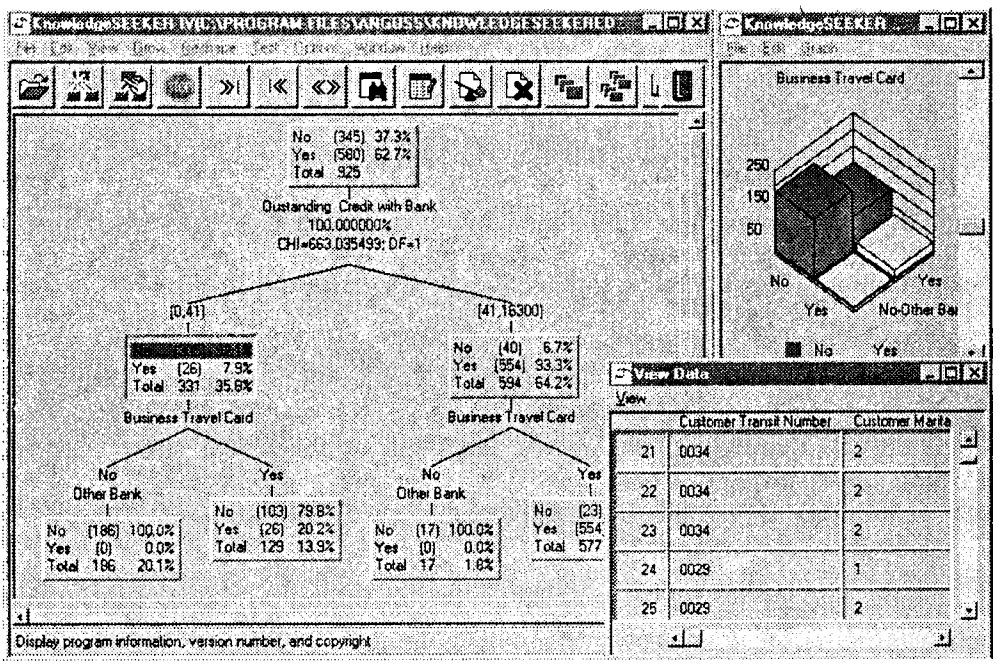


Рис. 1.2. Система KnowledgeSEEKER обрабатывает банковскую информацию

Эволюционное программирование

Проиллюстрируем современное состояние данного подхода на примере системы PolyAnalyst — отечественной разработки, получившей сегодня общее признание на рынке Data Mining. В данной системе гипотезы о виде зависимости целевой переменной от других переменных формулируются в виде программ на некото-

ром внутреннем языке программирования. Процесс построения программ строится как эволюция в мире программ (этим подход немного похож на генетические алгоритмы). Когда система находит программу, более или менее удовлетворительно выражающую искомую зависимость, она начинает вносить в нее небольшие модификации и отбирает среди построенных дочерних программ те, которые повышают точность. Таким образом система «выращивает» несколько генетических линий программ, которые конкурируют между собой в точности выражения искомой зависимости. Специальный модуль системы PolyAnalyst переводит найденные зависимости с внутреннего языка системы на понятный пользователю язык (математические формулы, таблицы и пр.).

Другое направление эволюционного программирования связано с поиском зависимости целевых переменных от остальных в форме функций какого-то определенного вида. Например, в одном из наиболее удачных алгоритмов этого типа — методе группового учета аргументов (МГУА) — зависимость ищут в форме полиномов. В настоящее время из продающихся в России систем МГУА реализован в системе NeuroShell компании Ward Systems Group.

Стоимость систем до $5000.

Генетические алгоритмы

Data Mining не основная область применения генетических алгоритмов. Их нужно рассматривать скорее как мощное средство решения разнообразных комбинаторных задач и задач оптимизации. Тем не менее, генетические алгоритмы вошли сейчас в стандартный инструментарий методов Data Mining, поэтому они и включены в данный обзор.

Первый шаг при построении генетических алгоритмов — это кодировка исходных логических закономерностей в базе данных, которые именуют хромосомами, а весь набор таких закономерностей называют популяцией хромосом. Далее для реализации концепции отбора вводится способ сопоставления различных хромосом. Популяция обрабатывается с помощью процедур репродукции, изменчивости (мутаций), генетической композиции. Эти процедуры имитируют биологические процессы. Наиболее важные среди них: случайные мутации данных в индивидуальных хромосомах, переходы (кроссинговер) и рекомбинация генетического материала, содержащегося в индивидуальных родительских хромосомах (аналогично гетеросексуальной репродукции), и миграции генов. В ходе работы процедур на каждой стадии эволюции получаются популяции со все более совершенными индивидуумами.

Генетические алгоритмы удобны тем, что их легко распараллеливать. Например, можно разбить поколение на несколько групп и работать с каждой из них независимо, обмениваясь время от времени несколькими хромосомами. Существуют также и другие методы распараллеливания генетических алгоритмов.

Генетические алгоритмы имеют ряд недостатков. Критерий отбора хромосом и используемые процедуры являются эвристическими и совсем не гарантируют на-

хождение «лучшего» решения. Как и в реальной жизни, эволюцию может «заклинить» на какой-либо непродуктивной ветви. И, наоборот, можно привести примеры, как два неперспективных родителя, которые будут исключены из эволюции генетическим алгоритмом, оказываются способными произвести высокоэффективного потомка. Это становится особенно заметно при решении высокоразмерных задач со сложными внутренними связями.

Примером может служить система GeneHunter фирмы Ward Systems Group. Ее стоимость — около $1000.

Алгоритмы ограниченного перебора

Алгоритмы ограниченного перебора были предложены в середине 60-х годов М. М. Бонгардом для поиска логических закономерностей в данных. С тех пор они продемонстрировали свою эффективность при решении множества задач из самых различных областей.

Эти алгоритмы вычисляют частоты комбинаций простых логических событий в подгруппах данных. Примеры простых логических событий: $X = a$; $X < a$; $X > a$; $a < X < b$ и др., где X — какой либо параметр, a и b — константы. Ограничением служит длина комбинации простых логических событий (у М. Бонгарда она была равна 3). На основании анализа вычисленных частот делается заключение о полезности той или иной комбинации для установления ассоциации в данных, для классификации, прогнозирования и т. п.

Наиболее ярким современным представителем этого подхода является система WizWhy предприятия WizSoft. Хотя автор системы Абрахам Мейдан не раскрывает специфику алгоритма, положенного в основу работы WizWhy, по результатам тщательного тестирования системы были сделаны выводы о наличии здесь ограниченного перебора (изучались результаты, зависимости времени их получения от числа анализируемых параметров и др.).

Автор WizWhy утверждает, что его система обнаруживает **ВСЕ** логические if-then-правила в данных. На самом деле это, конечно, не так. Во-первых, максимальная длина комбинации в if-then-правиле в системе WizWhy равна 6, и, во-вторых, с самого начала работы алгоритма производится эвристический поиск простых логических событий, на которых потом строится весь дальнейший анализ. Поняв эти особенности WizWhy, нетрудно было предложить простейшую тестовую задачу, которую система вообще не смогла решить. Другой момент — система выдает решение за приемлемое время только для сравнительно небольшой размерности данных.

Тем не менее, система WizWhy (рис. 1.3) является на сегодняшний день одним из лидеров на рынке продуктов Data Mining. Это не лишено оснований. Система постоянно демонстрирует более высокие показатели при решении практических задач, чем все остальные алгоритмы. Стоимость системы около $4000, количество продаж — 30 000.

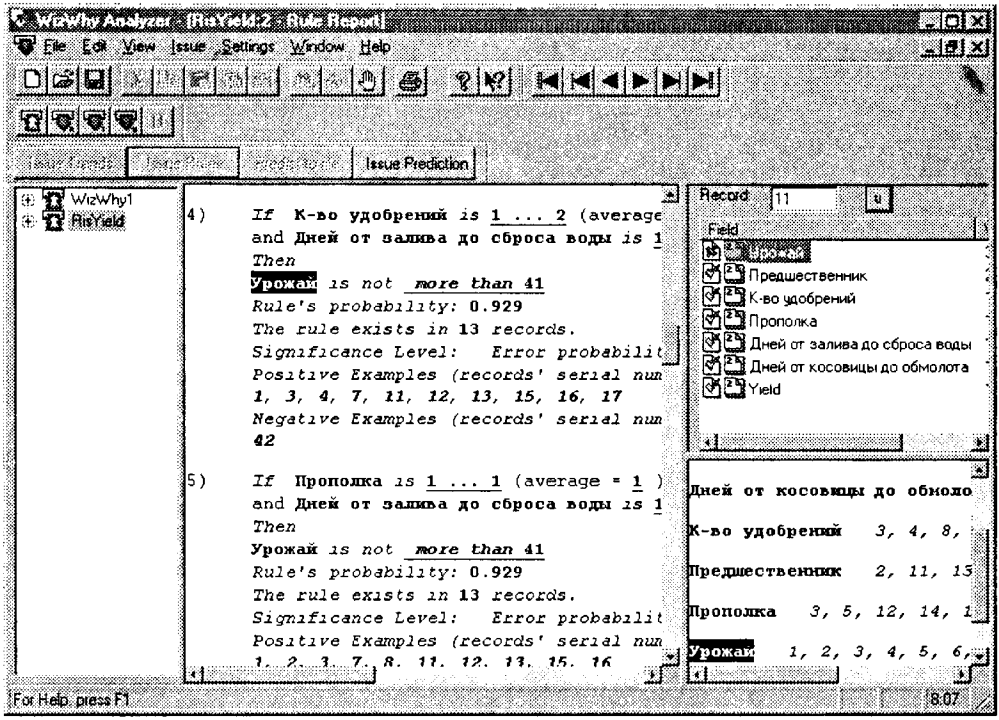


Рис. 1.3. Система WizWhy обнаружила правила, объясняющие низкую урожайность некоторых сельскохозяйственных участков

Системы для визуализации многомерных данных

В той или иной мере средства для графического отображения данных поддерживаются всеми системами Data Mining. Вместе с тем, весьма внушительную долю рынка занимают системы, специализирующиеся исключительно на этой функции. Примером здесь может служить программа DataMiner 3D (рис. 1.4) словацкой фирмы Dimension5 (5-е измерение).

В подобных системах основное внимание сконцентрировано на дружелюбии пользовательского интерфейса, позволяющего ассоциировать с анализируемыми показателями различные параметры диаграммы рассеивания объектов (записей) базы данных. К таким параметрам относятся цвет, форма, ориентация относительно собственной оси, размеры и другие свойства графических элементов изображения. Кроме того, системы визуализации данных снабжены удобными средствами для масштабирования и вращения изображений. Стоимость систем визуализации может достигать нескольких сотен долларов.

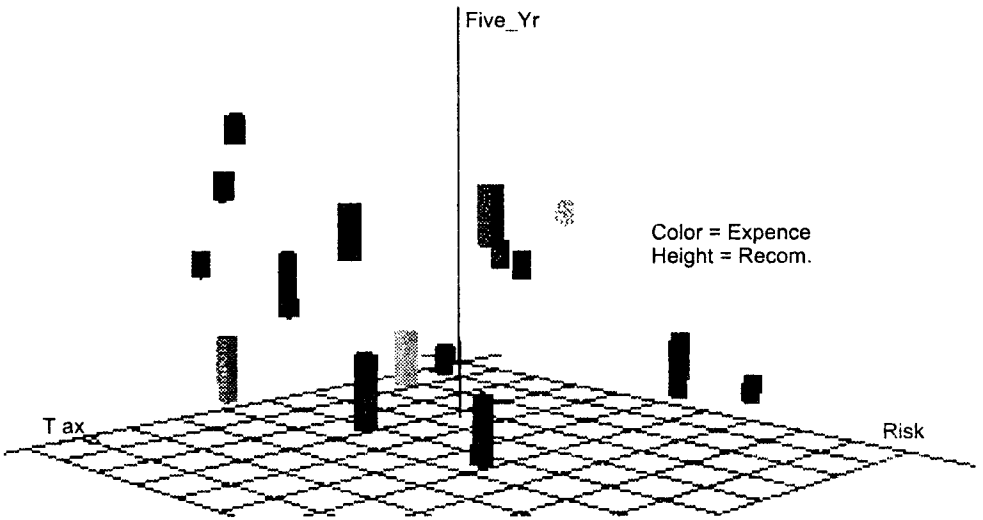


Рис. 1.4. Визуализация данных системой DataMiner 3D

Выводы

1. Рынок систем Data Mining экспоненциально развивается. В этом развитии принимают участие практически все крупнейшие корпорации (см., например, http://www.kdnuggets.com). В частности, Microsoft непосредственно руководит большим сектором данного рынка (издает специальный журнал, проводит конференции, разрабатывает собственные продукты).
2. Системы Data Mining применяются по двум основным направлениям:
   - Как массовый продукт для бизнес-приложений.
   - Как инструменты для проведения уникальных исследований (генетика, химия, медицина и пр.). В настоящее время стоимость массового продукта от $1000 до $10 000. Количество инсталляций массовых продуктов, судя по имеющимся сведениям, сегодня достигает десятков тысяч. Лидеры Data Mining связывают будущее этих систем с использованием их в качестве интеллектуальных приложений, встроенных в корпоративные хранилища данных.
3. Несмотря на обилие методов Data Mining, приоритет постепенно все более смещается в сторону логических алгоритмов поиска в данных if-then-правил. С их помощью решаются задачи прогнозирования, классификации, распознавания образов, сегментации БД, извлечения из данных «скрытых» знаний, интерпретации данных, установления ассоциаций в БД и др. Результаты таких алгоритмов эффективны и легко интерпретируются.
4. Вместе с тем, главной проблемой логических методов обнаружения закономерностей является проблема перебора вариантов за приемлемое время.

Известные методы либо искусственно ограничивают такой перебор (алгоритмы KOPA, WizWhy), либо строят деревья решений (алгоритмы CART, CHAID, ID3, See5, Sipina и др.), имеющих принципиальные ограничения эффективности поиска if-then-правил. Другие проблемы связаны с тем, что известные методы поиска логических правил не поддерживают функцию обобщения найденных правил и функцию поиска оптимальной композиции таких правил. Удачное решение указанных проблем может составить предмет новых конкурентоспособных разработок.

ПРИМЕЧАНИЕ Отметим очередное симптоматичное событие. 24 мая 1999 года компания Microsoft официально объявила об усилении своей активности в области Data Mining. Специальная исследовательская группа Microsoft, возглавляемая Усамой Файядом, и пять приглашенных партнеров (Angoss, Datasage, E.piphany, SAS, Silicon Graphics, SPSS) готовят совместный проект по разработке стандарта обмена данными и средств для интеграции инструментов Data Mining с базами и хранилищами данных.

Десять мифов интеллектуального анализа данных

Вокруг технологий Data Mining (интеллектуального анализа данных) ведутся активные дискуссии. Обширен спектр мнений по поводу этих технологий — от восторженных надежд на ожидаемые успехи до полного негативизма и отношения к ним как к преходящей моде. По-видимому, полезно сделать обзор указанных мнений и попытаться объективно разобраться, что здесь относится к областям фантазии и реальности. В основу такого обзора положена работа Р. Д. Смолла [9].

Миф 1. Интеллектуальный анализ данных позволяет получить неожиданные результаты, на основании которых стратегия принятия решений в той или иной области может быть кардинально изменена.

Разоблачение мифа. Как правило, применение интеллектуального анализа данных позволяет лишь усовершенствовать действующую и приносящую успех организационную схему. Это происходит в основном за счет небольших и постепенных изменений, а не революционных преобразований.

Вместе с тем, применение современной технологии способно приводить и к существенным переменам. Небольшие достижения, накапливаясь в течение продолжительного периода, могут вылиться в значительный отрыв от конкурентов. Кроме того, иногда интеллектуальный анализ данных позволяет обнаружить принципиально новые факты, радикально меняющие известные взгляды.

Миф 2. Технологии интеллектуального анализа данных настолько совершенны, что могут компенсировать недостаток знаний в предметной области или опыта по части построения моделей и их анализа.

Разоблачение мифа. Ни одна методика анализа не может заменить знание специалиста в своей области. Напротив, внедрение технологий интеллектуального анализа данных делает образование и опыт еще более важными факторами, чем раньше. В то время как опытным сотрудникам достаточно освоить одну-две новые аналитические методики, чтобы остаться на уровне требований дня и продолжить вносить свой вклад в повышение конкурентоспособности своего дела, от специалистов, не владеющих ничем, кроме техники анализа, нет абсолютно никакой пользы.

Чем меньше имеет знаний в предметной области специалист по интеллектуальному анализу, тем более он нуждается в тесном взаимодействии с людьми, которые такими знаниями обладают. В свою очередь, недостаток навыков и опыта в моделировании и применении соответствующих средств у экспертов в предметной области увеличивает их зависимость от поддержки специалистов по интеллектуальному анализу данных.

Например, предположим, что, рассматривая возможности повышения доходности вложений своего клиента, эксперт-финансист обращается к специалисту по интеллектуальному анализу для обработки большой и сложной базы данных, содержащей информацию о деятельности других клиентов. Пусть этот специалист выявляет определенную связь некоторых переменных с прибыльностью инвестирования. Но только финансист способен сказать, в каких пределах допускается законом изменение этих переменных.

Миф 3. Средства интеллектуальной обработки данных автоматически обнаруживают различные закономерности.

Разоблачение мифа. Многие средства такой обработки действительно позволяют автоматически выявлять закономерности в исследуемых данных. Тем не менее, ставить им конкретные цели необходимо. Например, если подать на вход список адресов клиентов и попытаться получить на выходе набор «профилей» покупателей, применение которых позволило бы повысить эффективность адресной рекламы, особенно на многое рассчитывать не стоит. В постановке целей важна конкретность. Без такой конкретности информация бывает перегружена мелкими, ненужными, отвлекающими и даже вредными деталями.

Миф 4. Интеллектуальный анализ данных может с пользой применяться только в определенных областях.

Разоблачение мифа. Практически любой процесс — от фармакологического производства до обслуживания клиентов — можно изучить, понять и улучшить с помощью методов интеллектуального анализа. Это могут быть такие разнообразные области: управление производственными процессами, кадровая работа, менеджмент предприятий общепита, медицина, социология, геология и др.

Интеллектуальный анализ данных станет полезен везде, где собраны данные. Конечно, в некоторых случаях расчет окупаемости может показать, что игра не стоит свеч.

Миф 5. Методы, используемые в средствах интеллектуального анализа данных, качественно отличаются от тех, которые применяются при традиционном построении количественных моделей.

Разоблачение мифа. Все методы, используемые в настоящее время для интеллектуального анализа данных, являются логическим развитием и обобщением аналитических подходов, известных уже на протяжении десятилетий. Нейронные сети — специальный случай градиентных методов определения параметров решающих функций — были предложены Ф. Розенблаттом в 40-х годах. Группа методов CART (classification and regression trees — деревьев классификации и регрессии) использовалась специалистами в области общественных наук в 60-х. Метод K-ближайшего соседа, специальный способ непараметрического оценивания плотности распределения, разработан Е. Фиксом и Дж. Ходжесом в 1951–52 годах для решения задач медицинской диагностики [10].

Все перечисленные и другие методы предназначены для построения моделей зависимости между набором определяющих переменных и результатом.

Новизна интеллектуального анализа информации заключается в расширении сферы применения указанных методов, которое стало возможно благодаря возросшей доступности данных и удешевлению вычислений.

Кроме того, из-за слабой связи между деловым миром и специалистами по анализу данных, большинство из которых принадлежат к академической среде, до недавнего времени не существовало программных реализаций указанных методик с дружественным интерфейсом пользователя. Наблюдающийся в последнее время рост интереса к средствам интеллектуального анализа данных объясняется отчасти именно усовершенствованиями в области интерфейса, которые сделали их доступными для использования различными прикладными специалистами.

Распространение мощных вычислительных методов интеллектуального анализа данных представляет собой значительный шаг вперед, однако не теряют своей ценности и применявшиеся ранее средства. Разнообразные регрессионные методы, дискриминантный анализ и даже простейшие графики также позволяют выявлять скрытые зависимости. Считается, что никакой один отдельно взятый метод не обеспечивает решения всех или хотя бы даже большинства задач. Чтобы преуспеть в интеллектуальном анализе данных, необходимо запастись достаточно широким набором инструментов, как старых, так и новых.

Миф 6. Интеллектуальный анализ данных представляет собой очень сложную процедуру.

Разоблачение мифа. Алгоритмы для интеллектуального анализа данных могут быть сложными, однако их применение, благодаря появлению новых программных средств, значительно упростилось. При этом часто не требуется обращаться к таким сложным алгоритмам, а достаточно использования относительно простых аналитических методов, табличных и графических представлений. Своей сложностью интеллектуальный анализ данных в значительной мере обязан тем же самым трудностям с организацией данных, которые характерны для любых методик моделирования. Это, в частности, работы по подготовке данных, такие как отбор переменных для включения в расчет и выбор способа их кодирования, а также интерпретация результата и принятие решения о путях его использования.

Миф 7. Применять интеллектуальный анализ имеет смысл только к базам данных больших объемов.

Разоблачение мифа. Действительно, некоторые из методов интеллектуального анализа данных были разработаны специально для применения к очень большим наборам данных, а многие использующие их приложения предназначены для обработки крупных массивов информации. Вместе с тем, полезные сведения можно извлекать и из наборов данных средних или малых размеров. Вообще, проблема необходимого и достаточного объема данных и по сей день остается открытой. Она решается отдельно для каждого конкретного случая.

Миф 8. Интеллектуальный анализ дает тем больший эффект, чем больше данных в него вовлечено, поэтому следует использовать в каждом случае все доступные данные.

Разоблачение мифа. Дополнительные данные приносят пользу, только если содержат новые сведения о рассматриваемых показателях или целях. В иных случаях их привлечение может оказаться не только бесполезным, но и вредным. Например, это происходит, если в данных содержится один из важных элементов информации, но нет других связанных с ним или не отражены взаимосвязи между такими элементами. Введение в процесс анализа данных, содержащих малую часть всей информации, может привести к снижению ценности получаемых решений, «зашумлять» информацию. Кроме того, эффективность применения средств интеллектуального анализа снижается в случае учета иррелевантной информации или дублирующих друг друга измерений одной и той же величины. Например, при использовании регрессионного анализа, если включить в число обрабатываемых признаков одновременно и возраст, и дату рождения, средство интеллектуального анализа обнаружит равную релевантность обоих факторов и понизит их вес.

Миф 9. Построение рабочей модели на основе выборки из базы неэффективно, так как информация, содержащаяся в базе данных, но не охваченная выборкой, оказывается потерянной для анализа.

Разоблачение мифа. Целью большинства усовершенствований методов формирования выборок является увеличение информационной эффективности по отношению к затраченным усилиям.

Любой набор данных уже представляет собой некую выборку из более мощной совокупности. Иногда просто не бывает иного выхода, как только обратиться к выборке. В некоторых случаях сбор полных данных оказывается невозможным. Но это ни в коей мере не снижает объективности грамотно проведенного анализа. В действительности даже относительно небольшая, но правильно составленная случайная выборка может дать великолепные результаты. В выборах президента США принимают участие более 60 млн граждан, имеющих право голоса, но последний предвыборный опрос, охватывающий две тысячных процента этого числа голосующих, редко дает ошибку прогноза более 2 %. Даже располагая базой данных обо всех 60 млн граждан с сотнями измерений по каждому из них, получить лучшую модель для предсказания исхода выборов было бы вряд ли возможно.

И в тех случаях, когда построение модели на основе полной БД вполне реально, часто бывает больше пользы от анализа нескольких моделей, основанных на выборках.

Миф 10. Интеллектуальный анализ данных — это еще одно веяние моды, которое уйдет так же скоро, как и пришло.

Разоблачение мифа. Название средств интеллектуального анализа данных может еще не раз измениться, но они сами навсегда останутся в числе важнейших инструментов. Внедрение методов интеллектуального анализа данных — очередной этап процесса, развивающегося с начала XX века. Бурный рост вычислительной мощности компьютеров в сочетании с появлением дешевых электронных методов сбора больших объемов данных логично вывели нас на этот этап.

Игнорировать интеллектуальный анализ данных невозможно. Применяемые для него методы многочисленны, а преимущества, открываемые в результате выявления новых знаний, — огромны. Предприятия, руководствующиеся в своих действиях в данной области «мифологией», окажутся в серьезном проигрыше по сравнению с организациями, использующими точно просчитанный рациональный подход, опирающийся на реальные факты.

В заключение приведем общие рекомендации специалистов по интеллектуальному анализу данных.

Шесть шагов к успеху в интеллектуальном анализе данных

1. Четкое представление о цели.
2. Сбор релевантных данных.
3. Выбор методов анализа.
4. Выбор программного средства.
5. Выполнение анализа.
6. Принятие решения об использовании результатов.

Литература

1. Айвазян С. А., Бухштабер В. М., Юнюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. — М.: Финансы и статистика, 1989.
2. Knowledge Discovery Through Data Mining: What Is Knowledge Discovery? — Tandem Computers Inc., 1996.
3. Кречетов Н. Продукты для интеллектуального анализа данных//Рынок программных средств, 1997. № 14–15, С. 32–39.

4. Boulding K. E. General Systems Theory – The Skeleton of Science//Management Science. 1956. № 2.
5. Дж. ван-Гик, Прикладная общая теория систем. – М.: Мир, 1981.
6. Киселев М., Соломатин Е., Средства добычи знаний в бизнесе и финансах// Открытые системы. 1997. № 4. С. 41–44.
7. Дюк В. А. Обработка данных на ПК в примерах. – СПб: Питер, 1997.
8. Дюк В. А. Data Mining – интеллектуальный анализ данных//Byte (Россия). 1999. № 9. С. 18–24.
9. Small R. D. Интеллектуальный анализ данных: мифы и факты//InfoWorld. 1997. № 22–23. С. 38–39.
10. Fix E., Hodges J. L Discriminatory analysis, nonparametric discrimination USA School of Medicine. – Texas: Rendolph Field, 1951–1952.

2 ГЛАВА Современные методы анализа данных

Обзор компьютерных средств анализа данных

Рынок компьютерных программ анализа данных обширен и разнообразен. На нем представлены продукты более тысячи наименований. Такое разнообразие отражает многоплановость задач анализа в различных областях человеческой деятельности. Обзоры указанных программ приводятся в специальных справочниках, где содержатся краткие описания их назначения, требования к техническим характеристикам компьютера, сведения о дополнительных сервисных возможностях, цены и адреса фирм-поставщиков. Это весьма объемные издания, публикуемые в западной прессе.

Информация о последних версиях программ регулярно помещается в популярных компьютерных журналах и газетах типа «PC Magazine», «PC World», «BYTE», «PC Week» и др. Известны аналогичные отечественные публикации. Они представлены, в основном, в журнале «Мир ПК» [21, 28, 35, 36, 38].

Ценные сведения о компьютерных системах обработки данных можно почерпнуть в [49, 50]. Кроме теории в этих книгах дается классификация программного обеспечения в области анализа данных, рассматриваются требования к статистическим пакетам общего назначения, характеризуются особенности российского рынка, приводится краткий обзор наиболее популярных программ и предлагаются рекомендации по их выбору. Также немало полезной информации содержится в словаре-справочнике «Информатика в статистике» [33]. Реальные примеры практически по всем основным разделам анализа данных разобраны в монографии [27].

Вместе с тем необходимо отметить, что значительная часть публикуемой информации быстро устаревает. Это связано со стремительными темпами развития отрасли. На рынке программного обеспечения в условиях жесткой конкуренции происходит процесс консолидации, и положение на сегодняшний день заметно

отличается от ситуации, скажем, трехлетней давности. Возглавляют процесс консолидации (как недавно сказал Jack Noonan, президент корпорации SPSS) те, кто может предложить наилучший продукт и сделать это быстрее всех. Для тех же, кто испытывает трудности при переходе к новым операционным системам, процесс объединения оказывается фатальным. Пример такой консолидации — слияние SPSS с фирмой SYSTAT и приобретение в январе 1996 года корпорацией SPSS одного из крупнейших конкурентов, фирмы BMDP Statistical Software Inc.

На рынке математического обеспечения в эпоху больших компьютеров лидировали несколько статистических пакетов — BMDP, SAS и SPSS. Это объяснялось тем, что фирмы-разработчики достаточно быстро реагировали на достижения в области анализа данных и ими был накоплен большой запас прочности, позволивший далеко оторваться от конкурентов. С появлением персональных компьютеров, новых языков программирования и технологий лидировавшим фирмам пришлось решать сложную задачу: создавать пакет для ПК заново или адаптировать уже существующую программу к требованиям «маломощных» компьютеров. Вместе с тем, богатые графические возможности ПК дали шанс менее известным фирмам сравнительно быстро создать новые, ныне очень популярные, программные средства анализа данных. В этот период появился пакет STATGRAPHICS (STATistical GRAPHICs System) фирмы Manugistics. Он настолько выигрышно отличался от других статистических пакетов удобством пользовательского интерфейса, что завоевал огромную популярность и в дальнейшем задал основные ориентиры для развития всей индустрии в целом. За последние годы, наконец, появились Windows-версии наиболее известных статистических систем. А корпорации SPSS и Manugistics выпустили версии для Windows 95/NT и продолжают наращивать огромный потенциал своих систем.

Таблица 2.1. Классификация статистических пакетов

Тип	Отечественные	Зарубежные
Профессиональные	Нет	SAS, BMDP
Универсальные	STADIA, Olymp	STATGRAPHICS, SPSS, STATISTICA, S-PLUS
Специализированные	Mesosaur, DataScope, «Класс-Мастер», «Эвриста», САНИ	Большое многообразие

Таблица 2.2. Зарубежные статистические пакеты

Название	Разработчик	Дилер в России
S-PLUS	Math. Soft Inc	Нет
SYSTAT	SPSS Inc.	Статистические системы и сервис
SPSS	SPSS Inc.	Статистические системы и сервис НКЦ «Тренд»
STATISTICA	Stat. Soft	Softline
STATGRAPHICS	Manugistic Inc.	«ИнфоСтрой»
SAS	SAS Inst.	ИНТУ
Visual Numerics	Visual Numerics	«СТАТ-ДИАЛОГ»

Таблица 2.3. Отечественные статистические пакеты

Название	Разработчик	Дилер
«Мезозавр» САНИ «Класс-мастер»	«СТАТ-ДИАЛОГ»	«СТАТ-ДИАЛОГ»
«Эвриста»	Центр статистических исследований МГУ	Центр статистических исследований МГУ
DataScope (СИГАМД)	«СтатПойнт»	«СтатПойнт»
Olymp	РОСЭКСПЕРТИЗА	РОСЭКСПЕРТИЗА
STADIA	«Информатика и компьютеры»	«Информатика и компьютеры»
SIGN	ИМТ МГУ	ИМТ МГУ
«Статистик-консультант»	«Тандем», Петрозаводск	ИМТ МГУ

Выбор пакета для анализа данных зависит от характера решаемых задач, объема обрабатываемого материала, квалификации пользователей, имеющегося оборудования и т. д. [49], [50].

Для пользователей, имеющих дело со сверхбольшими объемами данных или узкоспециализированными методами анализа, пока нет альтернативы профессиональным западным пакетам. Среди них самыми широкими возможностями обладает SAS. Для создания собственной системы обработки данных можно обратиться к библиотеке IMSL, содержащей сотни тщательно и квалифицированно составленных статистических подпрограмм.

Несколько меньшими возможностями обладают универсальные пакеты. Вместе с тем, их стоимость значительно ниже, чем профессиональных. При приобретении такого пакета не мешает, однако, лишний раз убедиться, что он содержит требуемые методы обработки.

Ниже предлагается краткий обзор некоторых популярных статистических пакетов, основанный на аналитической статье С. А. Айвазяна и В. С. Степанова «Программное обеспечение по статистическому анализу данных: методология сравнительного анализа и выборочный обзор рынка» (http://is1.cemi.rssi.ru/ruswin/publication/ep97001t.htm).

SAS

Общая информация

Система SAS развивается с 1976 года и работает на самых различных платформах под управлением одной из 12 операционных систем. Фирма-разработчик SAS принадлежит к числу ведущих разработчиков программных продуктов. В ней трудится более 3000 сотрудников, которые поддерживают более 3 миллионов пользователей в 120 странах.

SAS включает свыше 20 различных программных продуктов, объединенных друг с другом «средствами доставки информации» (Information Delivery System, или IDS, так что весь пакет иногда обозначается как SAS/IDS)[1]. Под понятием IDS

[1] *Wass J. A.* How Statistical Software Can Be Assessed. // Scientific Computing & Automation. 1996 (October). P. 14–24.

подразумевается, что пользователю SAS достаточно поставить на свой компьютер кроме ОС систему SAS и этим ограничиться для 100-процентной информатизации деятельности любой фирмы (все остальные функции типа задач, решаемых на основе Excel, Word, любой из СУБД и т. п., полностью возьмет на себя SAS/IDS).

Традиционно сложилось, что основными отечественными пользователями системы являются предприятия ВПК, крупные бизнесмены (некоторые банки, включая Центробанк, биржи, торговые фирмы), некоторые атомные станции, крупнейшие медицинские и геофизические центры, крупные государственные структуры.

Основным достоинством SAS является непревзойденная среди универсальных пакетов мощность по набору статистических алгоритмов. Кроме того, SAS предоставляет пользователю возможность подключения собственных оригинальных алгоритмов.

SAS/IDS — это интеграция весьма разнообразных возможностей доступа к данным и управления ими, средств анализа данных, способов представления информации и генерации отчетов. Система имеет модульную структуру и легко конфигурируется под специфические особенности ее пользователя.

Модули SAS, связанные с классификацией

Для классификации и снижения размерности в системе SAS/IDS функционируют следующие компоненты (модули системы):

- **BASE SAS** — ядро системы со встроенным языком программирования 4GL и языком работы с базами данных SQL, средства управления данными, поддержки индексов для баз данных, возможностями доступа к широкому набору форматов данных, процедуры описательной статистики и генерации отчетов.
- **FSP** обеспечивает полноэкранный доступ к данным, ввод, редактирование, преобразование данных, генерацию отчетов и деловую переписку.
- **GRAPH** содержит деловую, научную, рекламную графику, различные шрифты и карты.
- **STAT** включает в себя многофункциональный набор статистических процедур анализа данных.

Дополнительные модули, работающие под любой ОС

IML представляет собой интерактивный *матричный* язык программирования для выполнения углубленных математических, инженерных и статистических расчетов. Этот язык дает возможность математику легко программировать свои собственные процедуры, используя язык, близкий к языку линейной алгебры.

LAB предоставляет пользователю экспертную поддержку. В частности, здесь система подсказывает пользователю, выполняются или нет предположения, лежащие в основе того или иного метода анализа данных.

Модули, работающие, в частности, под Windows, OS/2

ASSIST служит средством для обеспечения интерактивного доступа пользователей к различным возможностям системы **SAS/IDS**.

EIS является меню-управляемым инструментом разработки и поддержки интерактивных исполняемых информационных систем методом *объектно-ориентированной технологии*. С помощью этого модуля легко настроить систему на собственные данные и формы представления результатов.

ACCESS дает возможность конструировать отдельные интерфейсы для связи SAS/IDS с самыми разнообразными СУБД (ADABAS, DB2, ORACLE, SQL/DS и др.).

INSIGHT представляет собой интерактивный инструмент для графического анализа данных.

Из описанных модулей SAS — «кирпичей» — можно строить любые «сколь угодно высокие дома». Однако следует заметить, что процесс освоения технологии строительства, самого строительства, а также получения лицензии на «право застройки» требует определенных интеллектуальных и материальных затрат.

Достоинства и недостатки пакета

Основными достоинствами SAS считают мощное интеллектуальное ядро, поддержку архитектур *клиент-сервер*, возможность доступа и интеграции данных из любых источников и наличие объектно-ориентированной технологии быстрой разработки приложений. При этом благодаря высокой гибкости и переносимости системы приложение, созданное в одной из ОС, может быть перенесено на любую из платформ, поддерживаемых SAS/IDS, начиная от суперЭВМ типа CRAY до Mainframe или рабочей станции (правда, при этом оно будет требовать для работы системную часть SAS).

Главными недостатками системы считают громоздкость, трудности в освоении, высокие требования к статистической квалификации пользователя, жесткие требования к аппаратной части ПЭВМ, большой объем занимаемого дискового пространства и дороговизну (свыше $800 за *каждый модуль*).

SPSS для Windows

Общие сведения

Пакет SPSS предназначен в первую очередь для статистиков-профессионалов. Он включает развитый аппарат статистического анализа, соизмеримый по мощности с SAS. Программу SPSS для Windows считают в настоящее время одним из лидеров среди универсальных статистических пакетов.

Вместе с тем, как и все мощные универсальные пакеты, SPSS «любит хорошее железо»: процессор должен быть 486DX-2 и выше, для его использования рекомендуется 16 Мбайт оперативной памяти, а на винчестере модули **Base** и

Professional Statistics для управления данными и с алгоритмами классификации требуют как минимум 65–80 Мбайт (вместе с файлами подкачки). Кроме того, цена полного комплекта системы SPSS (SPSS Base + набор из 7 модулей) достаточно внушительна ($4290 для версии 6.1 или 7.0).

Особенности версии 7.0

SPSS-7.0 имеет удобные графические средства (более 50 типов диаграмм), а также развитые средства подготовки отчетов. Эта версия отличается производительностью, скоростью вычислений и расширенным функциональным наполнением. Аналитические параметры отображаются на экране в виде простых и понятных меню и диалоговых окон.

Усовершенствование в процедуре обучения достигается введением специального средства Навигатор. Навигатор выполняет в SPSS интеллектуальную функцию, объясняя пользователю, какую статистику лучше применить в каждом конкретном случае или как ввести данные в данном подразделе. Во многом за счет этого средства можно сфокусировать свое внимание собственно на анализе данных, не заботясь о механизме его выполнения. Новая контекстно-ориентированная справочная система содержит пошаговые инструкции для наиболее важных операций.

Для эффективного применения пакета для классификации и снижения размерности, как минимум, нужны методы из модулей SPSS BASE и Profess. Statistics. Существенно же повысить точность и/или эффективность классификации и прогноза может применение части (или всех) из модулей Advanced Statistics, CHAID и Neural Connection.

Первый модуль содержит, в частности, модели логистической регрессии, а также ряд методов, смежных с классификацией. Модуль CHAID строит деревья решений. Последний модуль реализует классификацию с помощью нейросетей.

На основе DDE- и OLE-технологий фирмы Microsoft, а также стандарта ODBC в SPSS также решены вопросы обмена с другими Windows-приложениями и выполняется связь с большинством форматов баз данных. Так, можно, не выходя из среды WinWord, одновременно работать в среде SPSS и, наоборот, очень легко переносить полученные тестовые или графические результаты из SPSS в документ системы Word.

Достоинства и недостатки версий 6.1 и 7.0

По мнению разработчиков пакета, после SAS, в своей *полной конфигурации* SPSS для Windows является пакетом с наиболее высоким значением параметра **мощность**: он обладает *весьма полным* набором статистических (всего их более 60) и графических процедур, а также процедур создания отчетов. Также создатели пакета гордятся интерфейсом SPSS с пользователем, считая его очень простым и удобным. Кроме того, традиционно пакет отличается высокой точностью вычислений.

Однако за повышенные комфорт и мощность требуется заплатить немалые суммы. Чтобы сориентировать читателя в ценах на модули, приведем некоторые из них. Так, модули «Углубленная статистика» и CHAID стоят в США около $500 и $700 (и, соответственно, $550 и $740 при покупке у дистрибьютора в РФ); нейромодуль же стоит почти тысячу USD. Набор из модулей для решения задач классификации в составе SPSS BASE и Profess. Statistics для версии 6.1 или 7.0 будет стоить около $1100 в США [9] (и $1530 при покупке у российского дистрибьютора). Достаточно же полный комплект SPSS, как уже отмечалось, продается этим дистрибьютором за $4290.

Версия 7.5. В марте 1997 года фирма SPSS представила на российском рынке новую версию пакета 7.5 для Windows 95 (NT) [51]. В ней базовый модуль выполняет функции факторного, кластерного и дискриминантного анализа, а также дополнен инструментами вычисления близости между наблюдениями (или переменными).

Работа с версией 7.5 существенно облегчена. Поддержка сценариев позволяет настраивать интерфейс, связывать сценарии с пиктограммами панели инструментов (пунктами меню), активизировать заданные последовательности действий, интегрировать пакет с другими приложениями, разрабатывать новые приложения, создавать контекстно-зависимые справочные системы.

Благодаря средству ODBC расширено число приложений, с которыми версия 7.5 может обмениваться исходными данными. Кроме того, можно импортировать файлы SYSTAT, экспортировать таблицы и текст в формат ASCII. Пакет также может легко интегрироваться с Интернет-технологией.

SASS 7.5 Base поставляется с руководством на русском языке. Кроме того, есть русификация на уровне интерфейса пользователя и навигатора результатов.

Для работы с версией 7.5 требуется ПЭВМ с ОС Windows 95/NT (имеющая процессор 486 DX, 12 Мбайт RAM) и монитором VGA (или более мощный). Она занимает на диске 55 Мбайт и защищена аппаратной «заглушкой» (электронным ключом). Минимальная цена версии 7.5 составляет $980 за одно рабочее место. SPSS предлагает лицензии на год, три года и локальные лицензии для образовательных и научных учреждений. Приобретение последней дает право продавать копии пакета сотрудникам и студентам по любой цене. В числе дополнительных модулей оставлены модули Tables, Trends, Exact Tests, Nenral Connection и Diamod.

SYSTAT

Общая информация

Универсальная статистическая система SYSTAT разработана одноименной фирмой, которая с сентября 1994 года «поглощена» корпорацией SPSS. Она отличается от других универсальных систем типа SAS, SPSS, BMDP тем, что изначально спроектирована под платформу IBM PC. Главное достоинство пакета — исключительно широкий диапазон и глубина проработки функционального наполне-

ния. Здесь есть широкие возможности и для слабо подготовленного в статистике пользователя, и для достаточно искушенного статистика.

Фирма SYSTAT была совсем недавно одним из лидеров в области производства высококачественного статистического программного обеспечения. Поэтому сегодня более 150 учебных заведений во всем мире готовят у себя специалистов на основе наукоемких продуктов этой фирмы.

Ряд лет пакет считался одним из лучших среди универсальных пакетов углубленного статистического анализа. Однако с современных позиций просматривается его определенное отставание в графике в режиме «высокого разрешения» [10].

Windows-версия пакета 5.04 подробно описана в [12] и кратко в [44], а 6-я DOS-версия кратко анализируется в [4]. В приложении к эконометрике пакет разобран в работе [52]. Имеются учебные версии пакета, называемые MYSTAT и BUSINESS MYSTAT (см. о них The Economic Journal.1990. Vol. 100, June).

Последние 6-е версии пакета для среды MS-Windows (выпуск 1996 года) и для среды MS-DOS (выпуск 1995 года) являются первыми версиями пакета, с тех пор как фирму SYSTAT купила корпорация SPSS, Inc.

Разработчики пакета считают, что SYSTAT-6.0 для среды Windows хорошо сбалансирован по соотношению «мощность/удобство» (см. [9], с. 107).

Документация SYSTAT

Документация пакета включает в себя четыре тома. Это ясно и хорошо написанное руководство «Как начать работу», а также руководства по разделам «Графика», «Статистика». Есть также небольшое «Руководство по данным».

Второй и третий тома дают читателю углубленный взгляд на то, что можно делать и как это можно выполнить в среде пакета. Второй том описывает, как работать с графикой в пакете. Руководство по статистике начинается с обзора методов и включает ссылки на хорошо подобранную библиографию. В методическом плане руководство не уступает добротному учебному курсу по статистическому анализу данных. Инструкции по пошаговой работе, как правило, полезны и точно ведут к цели, а указатель очень полезен для пользователя, начинающего работу с SYSTAT.

Возможности SYSTAT по управлению данными

Пакет использует затабулированное окно для ввода данных и их редактирования. Верхняя строка таблицы с данными задает имена переменных, которые должны иметь не более чем 8 символов и оканчиваться на символ $, если данная переменная (признак) имеет неусловную природу. Ввод данных осуществляется без каких-либо ухищрений.

Имеются хорошие возможности по преобразованиям данных. Однако результаты таких преобразований являются статичными. Функциональные имена в окне преобразований являются кодами, например, XDF и XCF обозначают плотность и кумулятивную функцию распределения случайной величины хи-квадрат. Пропущенные символьные значения кодируются пробелами; пропуски в числовых

признаках кодируются отрицательными числами и появляются в редакторе как периодические вещественные числа.

Графика в пакете SYSTAT

Пакет обладает прекрасными возможностями отображения на экране исходных данных и полученных результатов разведочного анализа, имея в своем распоряжении около 30 различных способов графического отображения: гистограммы, ящики с «усами», стебли с листьями [47], значки, 2D- и 3D-диаграммы рассеяния и т. д. Кроме того, имеются матрицы диаграмм рассеяния, графики функций и географических карт.

Кроме того, пакет позволяет порождать и изображать сложные поверхности, что полезно для визуализации сложных функций. В версии 6.0 нажатием кнопки мыши легко вращать даже сложные 3D-графики с координатными осями по отношению к плоскости экрана [53]. Эта же возможность сохраняется и при обработке данных в режиме «реального времени».

Для многих графиков имеются специальные средства типа стрелки для исследования точек-выбросов, ключи с диапазонами для режима «лупа» или «лассо», для исследования выделенного фрагмента данных. Графика пакета достаточно гибкая, легко управляемая и объектно-ориентированная. Есть возможности интерактивных графических преобразований данных, что очень удобно при разведочном анализе. Также имеются средства разработки презентаций.

Функциональные достоинства SYSTAT

SYSTAT обладает хорошей и заслуженной репутацией в плане корректности применяемых алгоритмов. Он имеет обширное меню с функциональными алгоритмами, включая описательную и непараметрическую статистику, корреляцию, кластерный анализ, проверку многомерных гипотез для общей линейной модели (MGLH) и таблицы сопряженности. Пакет особенно силен в областях дисперсионного анализа и планирования экспериментов.

В версии 6.0 имеется множество дополнительных процедур для дискриминантного анализа, матричной алгебры, логлинейных моделей, планирования экспериментов, структурного анализа и карт контроля качества [4], [5]. Также были добавлены робастные (устойчивые) алгоритмы, дающие точные и корректные результаты при почти вырожденных данных. Кроме того, эта версия предоставляет пользователю наиболее широкие возможности анализа общей линейной статистической модели.

Критические замечания

По мнению авторов [44], определенное неудобство работы с пакетом связано с тем, что часть операций доступна лишь из командной строки. В качестве несущественного недостатка версии 5.04 пакета упоминается отсутствие хорошего редактора отчетов, так же как ограничение на число переменных в данных [12]

(в версии 5.04 их можно было иметь не более 256, но в версии 6.0 это ограничение уже было снято [53]). Иногда объяснение в руководстве пользователя дается для упрощенного варианта меню, а детали используемого статистического метода приведены только как инструкции к командной строке. Число десятичных разрядов вещественного числа нельзя фиксировать у данной переменной в рабочей таблице, а надо обязательно зафиксировать на всю таблицу целиком. При импорте файлов в версии 5.2.1. возникают проблемы, если в файле есть хотя бы одна переменная — признак с длиной имени, превышающей 8 символов [10].

Некоторые другие разделы меню содержат в себе меньше информации, чем это было бы нужно для оптимального дружественного интерфейса с пользователем, который характерен для некоторых графических редакторов. В частности, в [12] относительно версии 5.04 делается замечание по поводу надоедающего исчезновения главного меню при попытках управлять программой из определенных подменю.

Также несколько важных статистических методов решительно не являются дружественными к пользователю (в [12] приводится пример по непарному t-критерию и простому однофакторному дисперсионному анализу). Однако, по мнению [10], [12], несмотря на ряд этих небольших неудобств SYSTAT является превосходной и весьма ценной программой.

MINITAB

Пакет MINITAB развивается более 20 лет и широко известен в США, где он является одним из основных учебных пакетов. Во многом, правда, это объясняется не его исключительными свойствами, а тем, что пакет в свое время захватил определенный сегмент рынка. Сейчас распространяется версия 10.0 для среды MS-Windows и уже появилась улучшенная 32-разрядная версия 11.0 [12]. Кроме рассматриваемых платформ, пакет также работает на Macintosh [8] в среде MS-DOS, на рабочих станциях и других компьютерах.

MINITAB хорошо продуман по разделу описательной (дескриптивной) статистики, хорошо сконструирован и управляется с помощью удобного меню или, по желанию пользователя, через команды, составлять которые помогают диалоговые окна пакета. Часто используемые команды можно запускать по их первой букве. Общее число команд превышает 200 [8]. Можно составлять специальные макросы для выполнения последовательностей команд.

Импорт/экспорт данных из других Windows-приложений делается через стандартный буфер обмена. В пакете имеются разнообразные возможности по управлению данными.

Документация пакета включает в себя три тома: 28-страничное руководство для быстрого освоения, 240-страничное руководство пользователя и справочное руководство. Последнее содержит множество примеров и продуманные указатели.

Пользователь Minitab может легко и быстро научиться решать практически все типовые задачи, в основном из области одномерного анализа и анализа временных рядов. Фирмой Minitab, Inc. хорошо налажены поддержка пользователей и

обмен опытом через Группу пользователей MINITAB. Кроме того, фирма весьма недорого продает мини-руководство по пакету для тех студентов, которые, возможно, его не имеют, но хотят больше о нем узнать в рамках своих учебных курсов.

В области многомерного анализа 8-я версия пакета явно не являлась лидером. Тем не менее, она позволяет находить главные компоненты или же проводить стандартный линейный или даже квадратичный дискриминантный анализ. Однако многомерный анализ был усилен в более поздних версиях.

Так, в версии 10.0 были добавлены алгоритмы факторного и кластерного анализа. Кроме того, эта версия позволяет получать множество хороших и сложных полноцветных графиков. В плане характеристики мощности MINITAB-10.0 достаточно силен и разнообразен, поэтому говорят, что первые четыре буквы пакета скорее надо поменять на Maxi.

Недостатком пакета является отсутствие формул для статистик в справочном руководстве, что затрудняет анализ значимости влияния различных факторов на принимаемые решения. Но, с другой стороны, этот справочник изобилует ссылками на стандартные статистические учебники.

Полагают, что недостатки MINITAB не очень существенны и что он является пакетом с умеренным соотношением «качество/цена».

STATISTICA/W

Общая информация

По мнению авторов [44], пакет STATISTICA/W (ниже Statistica) не стоит использовать пользователю-новичку в статистике, так как он предполагает владение статистической терминологией. Тем не менее, на отечественном рынке этот пакет пользуется популярностью, по-видимому, благодаря высокой активности фирмы-разработчика Statsoft и дилера в России — Softline, способствующих популяризации пакета (см. например, [20]).

О мощности пакета Statistica/W

Ряд авторов считает, что пакет Statistica является хорошо сбалансированным по соотношению «мощность/удобство» [4], [5]. Наличие достаточно широкого спектра функциональных алгоритмов делает его достаточно привлекательным для статистиков-профессионалов. Однако существует точка зрения, что удобство работы с этим пакетом является невысоким [10]. В частности, Statistica по своей структуре как бы состоит из нескольких связанных между собой «мини-пакетов». Эти «мини-пакеты» взаимодействуют друг с другом, имея одинаковый формат системных файлов. Так, если нужен раздел линейной регрессии, то приходится покинуть окружение главного модуля СПП и выходить в окружение модуля («мини-пакета») линейной регрессии.

В плане функционального наполнения пакет, например, по сравнению с программой STATGRAPHICS, о которой будет сказано ниже, более разнообразен, вклю-

чая в себя и разделы анализа, которые STATGRAPHICS содержит лишь в дополнительных модулях (поставляемых за дополнительную цену). В частности, он включает в себя ряд непараметрических методов анализа, методы многомерного анализа: дискриминантного, факторного кластерного логлинейного и др.

Вместе с тем, в пакете Statistica отсутствуют методы планирования экспериментов, графика по методам контроля качества. В целом пакет Statistica по мощности уступает пакетам SAS, SPSS и SYSTAT.

Особенности управления пакетом

Средства манипулирования исходными данными в пакете Statistica хорошо развиты. Данные относительно легко отредактировать, можно создавать новые переменные («признаки»), выбирать отдельные наблюдения или «вырезать» подмножество данных по строкам и/или по столбцам таблицы «объект-признак». Благодаря обширной панели инструментов для выполнения большинства манипуляций достаточно несколько щелчков мыши, так как почти для всех функций пакета здесь имеются пиктограммы. Кроме того, щелчком правой кнопки мыши вызываются дополнительные подменю, которые существенно ускоряют работу с пакетом.

Полезной особенностью пакета является настройка функций под экран, открытый в текущий момент времени. Так, при загрузке программы в память машины в активном окне возникает список модулей («мини-пакетов»), доступных пользователю. Отсюда пользователь может самостоятельно решать, какого сорта анализ требуется выполнить. Список модулей и порядок их следования в окне могут быть определены пользователем, что обеспечивает дополнительные удобства и гибкость настройки.

Связь с другими Windows-приложениями

Благодаря поддержке DDE в пакете Statistica выполняются те или иные командные сценарии других приложений. Например, можно в Excel написать макрос, который запускает пакет Statistica. После добавления в макрос специальных SQL-команд можно импортировать в пакет данные.

В версии под Windows также, как для пакетов SPSS или STATGRAPHICS, использование OLE-технологии обмена между Windows-приложениями позволяет интегрировать результаты, например, WinWord и Statistica.

Графика и документация в пакете Statistica

Сильной стороной пакета являются графика и средства редактирования графических материалов. В пакете представлены сотни типов графиков 2D или 3D (имеются даже графики типа 4D), матрицы и пиктограммы. Предоставляется возможность разработки собственного дизайна графика.

Средства управления графиками позволяют работать одновременно с несколькими графиками, изменять размеры сложных объектов, добавлять художествен-

ную перспективу и ряд специальных эффектов, разбивку страниц и быструю перерисовку. Например, 3D-графики можно вращать, накладывать друг на друга, сжимать или увеличивать. Передовая анимационная техника, примененная в версии 5.0 и относящаяся скорее к области искусства, позволяет увидеть на графиках, какие точки там изменились под влиянием изменений в одной из переменных.

Пакет имеет трехтомную документацию в 3000 страниц и краткое руководство. В экранный справочник входит почти весь материал печатной документации. Содержащиеся в документации и экранном справочнике рекомендации полезны, но порой недостаточно полны, а порой — чересчур детальны. Кроме того, они не всегда стыкуются с иерархическим стилем пакета.

Все универсальные пакеты имеют много пересечений по составу статистических процедур. Кроме того, современные версии программ обладают, как правило, модульной структурой, что позволяет существенно экономить средства. Windows-интерфейс последних версий пакетов во многом унифицирует взаимодействие пользователя с аналитическими, графическими и системными процедурами. Основные отличия кроются главным образом в цене. Кроме того, по-разному организован диалог. Функциональное наполнение пакетов также может варьироваться. Здесь, по мнению авторов, с учетом всех аспектов в лучшую сторону отличается STATGRAPHICS *Plus* for Windows. Именно поэтому дальнейший материал основан на рассмотрении функций STATGRAPHICS. Вместе с тем, следует отметить, что такие же функции могут быть выполнены практически любым другим пакетом анализа данных, обладающим развитыми средствами интерактивной графики. Фактически, изучив принципы работы STATGRAPHICS *Plus* for Windows, не составляет особого труда перенести полученные навыки на другие пакеты.

Обзор методов анализа данных на примере пакета STATGRAPHICS

STATGRAPHICS Plus for Windows — общие и уникальные свойства

STATGRAPHICS *Plus* for Windows включает более 250 статистических и системных процедур, применяющихся в бизнесе, экономике, маркетинге, медицине, биологии, социологии, психологии, на производстве и в других областях. Каждой группе процедур соответствует собственное меню. В Базовой системе функционируют следующие процедуры:

- Меню Describe содержит статистические методы анализа по одной и множеству переменных, процедуры подбора распределений, средства табуляции и кросс-табуляции данных (рис. 2.1).

- Меню Compare включает методы сравнения двух и более выборок данных, процедуры одно- и многофакторного дисперсионного анализа.
- Меню Relate содержит процедуры простого, полиномиального и множественного регрессионного анализа.

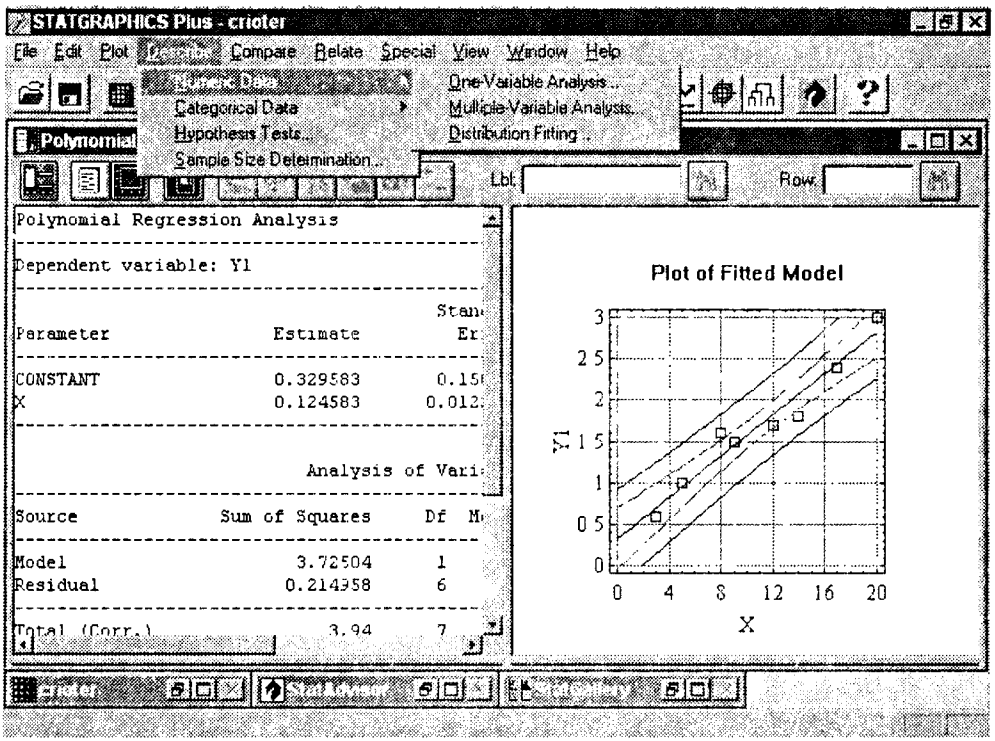


Рис. 2.1. Меню Describe содержит методы анализа по одной и множеству переменных, процедуры подбора распределений, табуляции и кросс-табуляции данных

Как видим, в Базовую систему включен достаточно полный набор наиболее часто встречающихся видов статистического анализа данных. В то же время, для расширения возможностей системы предлагаются дополнительные модули, инициализация которых осуществляется через меню Special. К ним относятся:

- Модуль **«Контроль качества»** предназначен для оценки эффективности всех звеньев производственного процесса и формирования соответствующих контрольных карт. В модуле прекрасно организованы процедуры для конструирования Парето-карт, анализа возможностей процесса и построения X- и R-контрольных карт. Тесная связь с базовой системой STATGRAPHICS *Plus* for Windows обеспечивает доступ к полному набору статистических методов. Представляется, что процедуры контроля качества реализованы наилучшим образом.

- Модуль **«Планирование эксперимента»** помогает сформулировать критерий оптимальности плана эксперимента, подобрать наилучший план, организовать сбор и обработку требуемой информации. При работе с этим модулем пользователю не стоит беспокоиться, много или мало ему известно о планировании эксперимента. В модуле предлагаются эффективные способы упрощения и интеграции знаний об исследуемом процессе. Процедура взаимодействия с модулем следующая: определение факторов; выбор плана; генерация рабочей таблицы для сбора и записи данных; подбор модели; интерпретация результатов. Все вместе позволяет уменьшить время исследования, снизить общие затраты и в целом повысить производительность.
- Модуль **«Анализ временных рядов»** содержит описательные методы, процедуры сглаживания рядов, сезонной декомпозиции и прогнозирования. Данный модуль помогает увидеть чистую картину динамических данных. Целесообразно начать работу с описательных методов, чтобы получить первое визуальное представление. Затем можно сделать более точное описание динамического ряда, учитывая сезонные эффекты, циклические изменения, тренды, ошибки, выбросы или точки излома в ваших данных. Результаты представляются в табличной форме или на удобных для восприятия графиках.

Если приходится иметь дело с данными из области финансов, STATGRAPHICS *Plus* for Windows предоставляет возможность определить оптимальное управление капиталом. А если требуется преобразовать данные для лучшей подгонки модели, то для этого существует широкий спектр встроенных функций, например, преобразования Бокса-Кокса. В модуле предусмотрена также возможность автоматического учета инфляционных факторов!

- Модуль **«Многомерные методы»** предназначен для изучения и раскрытия взаимоотношений множества факторов (переменных). Если пользователь занимается исследованиями в физике, социологии, медицине или других областях, где объекты исследования характеризуются большим числом признаков, данный модуль поможет сортировать и группировать данные, определять отношения между переменными, выдвигать и проверять различные гипотезы. Для этого в модуле функционирует пять мощных процедур, обеспечивающих проведение кластерного анализа, анализа по методу главных компонентов, факторного, дискриминантного и канонического корреляционного анализа.
- **Расширенный регрессионный анализ** кроме базисных процедур регрессионного анализа включает различные калибровочные модели, процедуры сравнения линий регрессии, отбора наилучших регрессионных моделей, нелинейную множественную регрессию, ридж-регрессию и логистическую регрессию. Требуется ли создать комплексную модель множественной регрессии или рассмотреть и оценить лабораторные методы или просто попытаться выбрать лучшую регрессионную модель — все это представлено в прекрасно организованном модуле расширенного регрессионного анализа.

Все перечисленные выше модули интегрируются в систему и полностью наследуют ее свойства. Модульная структура STATGRAPHICS *Plus* for Windows позволяет пользователю приобретать только то, что ему необходимо. Учитывая

сравнительно невысокую стоимость Базовой системы (например, по отношению к известному пакету SPSS), это позволяет существенно сэкономить средства.

Из множества свойств STATGRAPHICS *Plus* for Windows выделим и охарактеризуем следующие.

Гибкий импорт/экспорт данных

Система обеспечивает связь со всеми Windows-приложениями посредством OLE и DDE. Кроме того, файлы Windows- и DOS-версий полностью совместимы между собой, и система без проблем обменивается данными с другими программными продуктами, использующими Lotus-, dBASE-, DIF-, DBF- и ASCII-файлы.

Широкие возможности манипулирования данными

Управление данными организовано удобным и целесообразным способом. Легко доступные из редактора данных или из окна ввода данных опции преобразования предоставляют широкий набор возможностей сортировки данных и трансформации переменных, для чего предназначено более 100 операторов. При этом производимые манипуляции не изменяют содержимого исходных файлов.

STATGRAPHICS Plus for Windows Version 2

Коротко о продукте: Программный пакет для статистического анализа данных (рис. 2.2).

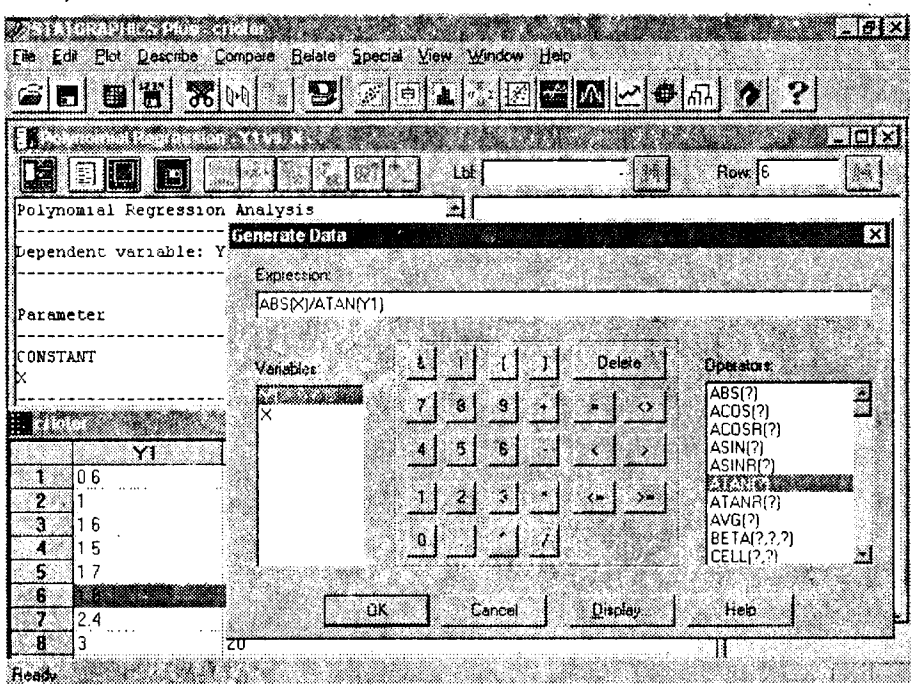


Рис. 2.2. STATGRAPHICS Plus for Windows располагает широкими возможностями сортировки и преобразования данных

Требования к оборудованию: ПК с процессором 386 и выше; 8 Мбайт ОЗУ; ОС Windows 3.x/95/NT, дисковое пространство — 14,5 Мбайт; SVGA графический адаптер; рекомендуется математический сопроцессор.

Цена:

Базовая система — $749.

Модули (Контроль качества, Планирование эксперимента, Анализ временных рядов, Многомерный анализ) — $449 каждый.

Вся система с полным набором модулей — $1699.

Manugistics, Inc.

Интегрированная графика

Каждая статистическая процедура в STATGRAPHICS *Plus* for Windows сопровождается интегрированной в систему отличной графикой. Щелкнув мышью на специальной пиктограмме, мы получаем меню, в котором предоставляется выбор графических отображений, соответствующих используемой процедуре. Все элементы графических отображений (масштабы, метки, цвета, надписи и пр.) могут быть подвергнуты коррекции и преобразованию (рис. 2.3). Для этого нужно выбрать требуемый элемент, щелкнув на нем левой кнопкой мыши, и затем щелкнуть правой кнопкой. Тогда на экране появится окно диалога, в которое вносятся необходимые изменения.

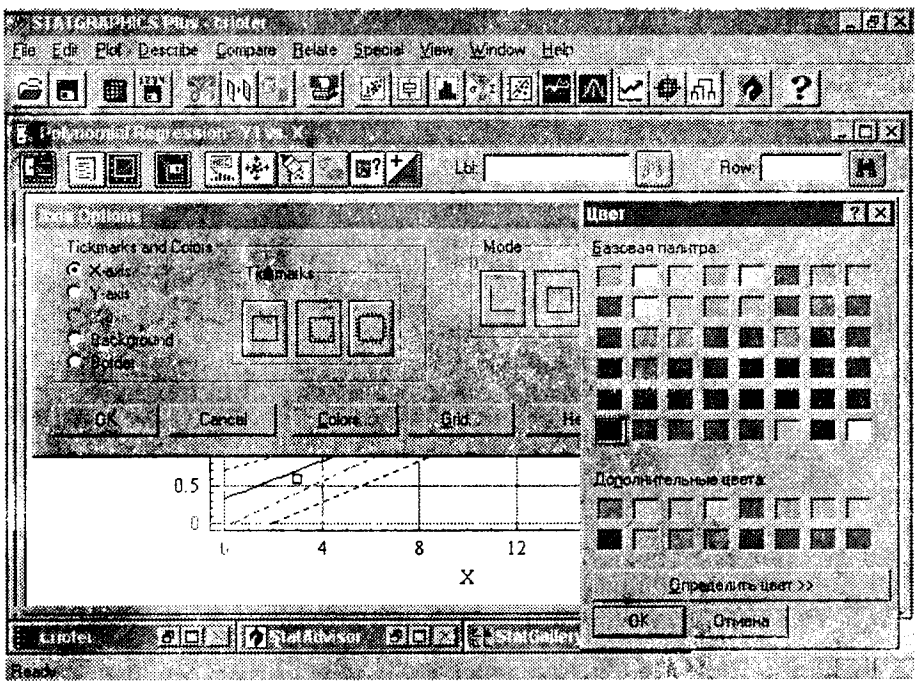


Рис. 2.3. Все элементы графических отображений результатов анализа могут быть преобразованы

Интерактивная графика

Интерактивная графика всегда была одной из самых сильных сторон STATGRAPHICS. Windows-интерфейс еще более повышает ее эффективность. Один щелчок мышью — и вы можете моментально идентифицировать точку на графическом отображении и выяснить ее местонахождение в файле данных. STATGRAPHICS *Plus* for Windows позволяет пользователю взаимодействовать с данными посредством графики любым мыслимым способом. Графика в системе становится аналитическим инструментом, а не только средством презентации. Например, можно вращать и рассматривать с разных сторон трехмерные изображения или осуществлять разгонку (jittering) точек на диаграммах рассеивания. Ценную возможность лучше «прочувствовать» структуру данных предоставляет функция окраски (brushing) точек на диаграммах рассеивания в соответствии со значениями какой-либо переменной (рис. 2.4). Быстрое и легкое исследование экспериментальных данных с помощью средств интерактивной графики делает процесс анализа увлекательным, стимулирующим интуицию и воображение.

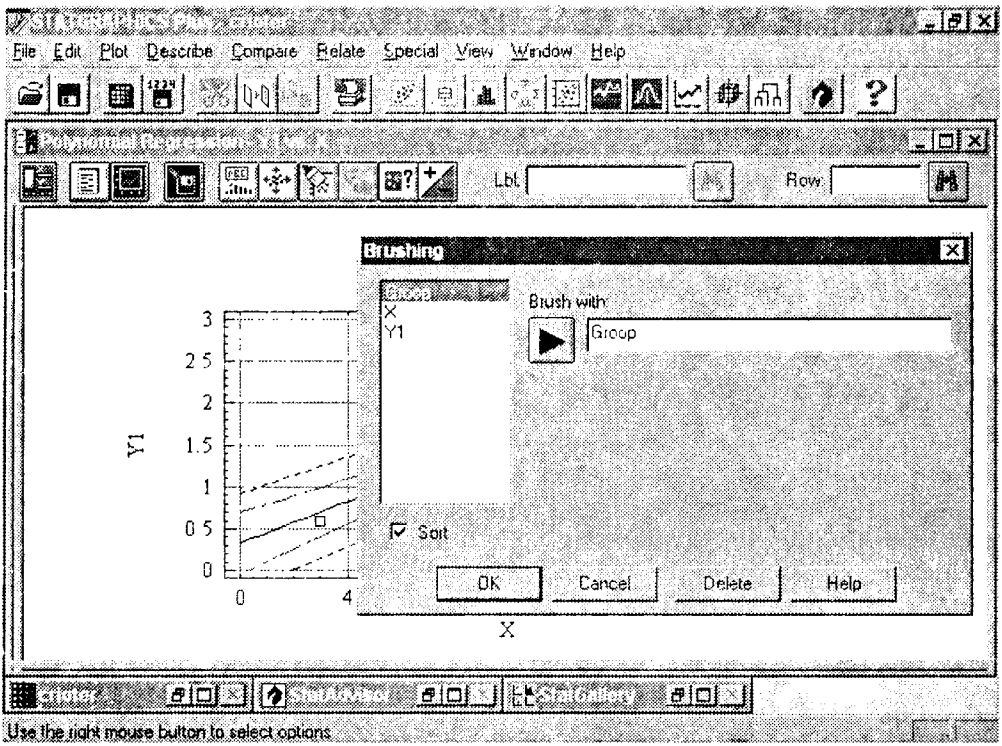


Рис. 2.4. Ценную возможность лучше прочувствовать структуру данных предоставляет функция окраски (brushing) точек на диаграммах рассеивания в соответствии со значениями какой-либо переменной

StatFolio — ваш собственный статистический проект

В STATGRAPHICS *Plus* for Windows реализовано уникальное средство для сохранения результатов работы и создания собственных статистических проектов. Представляется, что это не может быть сделано нагляднее и рациональнее. Все, что пользователь считает ценным в своем варианте анализа (выбранные методы, параметры статистических процедур, виды графических отображений результатов анализа, табличные формы, комментарии и т. п.), он может сохранить в виде нового файла StatFolio. Если возникает потребность в обработке другого множества данных по составленной схеме анализа, нужно просто загрузить новый файл данных. Результаты расчетов, таблицы и графики будут выданы автоматически.

StatFolio отображает технологическую цепочку анализа данных в виде набора пиктограмм. Отпала необходимость писать макросы, что значительно повышает продуктивность работы при подготовке и реализации статистических проектов. Статистический проект может быть обозначен в основных чертах искушенным профессионалом и затем передан менее опытному персоналу.

Всеобъемлющая статистическая консультация

В STATGRAPHICS *Plus* for Windows введено мощное средство, помогающее новичку работать на уровне эксперта, а специалисту — еще более повысить свое мастерство в анализе данных. Я имею в виду StatAdvisor (СтатКонсультант). Он предоставляет интерпретацию результатов, определяет значимые эффекты и выявляет возможные изъяны в проведенном анализе. Процедура получения консультации исключительно проста. Нужно щелкнуть мышью на интересующем графическом или табличном окне STATGRAPHICS и затем на пиктограмме StatAdvisor. Появляется консультационное окно, в котором содержатся исчерпывающие, легко воспринимаемые советы, разъяснения и рекомендации. Если к этому добавить высокий уровень документации STATGRAPHICS, написанной ясным языком, подробной, с тщательно разобранными примерами по всем видам анализа данных, то нужно сказать, что Windows-версия пакета может служить превосходным учебным пособием по основным разделам анализа данных.

Фактически StatAdvisor представляет собой интеллектуальную экспертную систему интерпретации результатов статистического анализа, аккумулирующую знания высококвалифицированных специалистов в этой тонкой и многогранной предметной области (рис. 2.5). Необходимость таких систем уже давно обсуждается в научной литературе [15]. Однако до сих пор существующим пакетам по прикладной статистике были присущи лишь слабо выраженные интеллектуальные свойства. Теперь можно констатировать, что в STATGRAPHICS *Plus* for Windows сделан важный шаг в данном направлении.

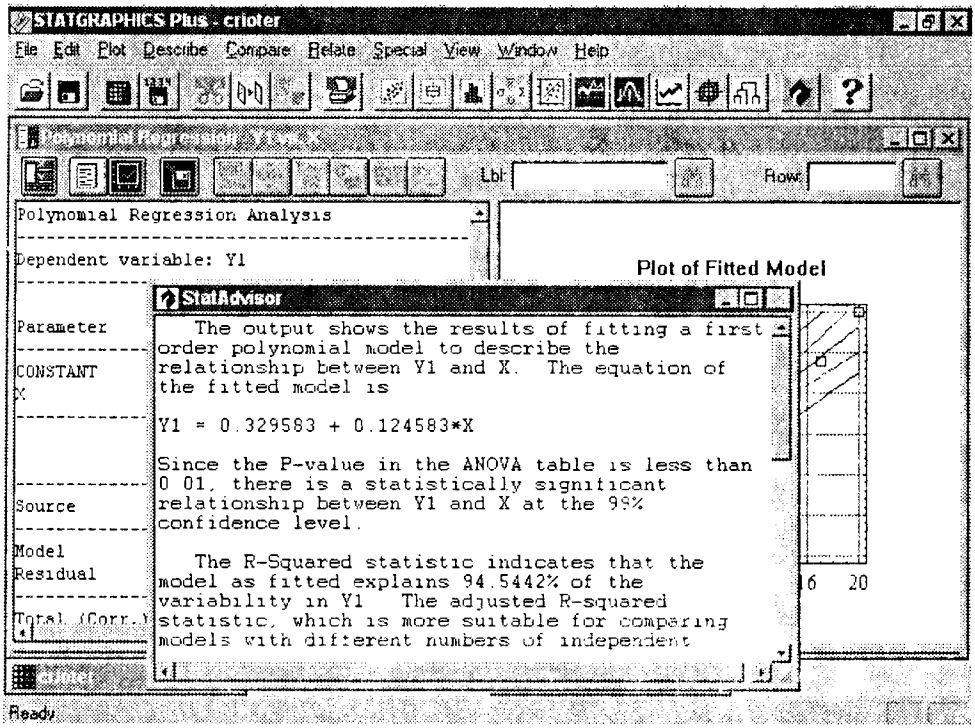


Рис. 2.5. StatAdvisor дает исчерпывающие консультации по всем видам статистического анализа данных

Комбинирование текста и графики для составления привлекательных статистических отчетов

Во всех версиях STATGRAPHICS большое внимание уделялось инструментам для составления отчетной документации. В последних версиях STATGRAPHICS *Plus* for Windows комбинирование текста и графики осуществляется с помощью специального нововведения — инструмента StatGallery. Теперь стало возможным произвольно располагать в одном окне или на одном листе до 9 различных фрагментов текста и графических иллюстраций. При этом трансформация и перемещение всех составляющих отчета производятся быстро и просто.

Поддержка последних технологий

STATGRAPHICS *Plus* for Windows способен работать не только в системе Windows 3.x, но также и под Windows 95 или Windows NT. В пакете активно используются все достижения и преимущества современных операционных систем, начиная от 32-разрядного кода, поддержки OLE и кончая длинными именами файлов. Это обеспечивает высокую скорость обработки и легкий обмен данными со всеми другими Windows-приложениями.

Подводя итог, следует отметить, что приведенные сведения далеко не полностью отражают все возможности анализа данных, которые предоставляет Windows-версия STATGRAPHICS и которые заслуживают более подробного рассмотрения. Но, думается, уже из кратких характеристик основных свойств становится ясно, что описываемый программный продукт имеет высокие качества. Работать с ним удобно, просто и эффективно. Ваш труд становится увлекательным и, что очень важно, стимулирующим творческое мышление и интуицию.

Базовая система статистических процедур

Основные характеристики

В Базовой системе STATGRAPHICS *Plus* for Windows функционируют следующие процедуры:

- меню Describe содержит статистические методы анализа по одной и множеству переменных, процедуры подбора распределений, средства табуляции и кросстабуляции данных;
- меню Compare включает методы сравнения двух и более выборок данных, процедуры одно- и многофакторного дисперсионного анализа;
- меню Relate содержит процедуры простого, полиномиального и множественного регрессионного анализа.

Ниже приводится подробный список доступных статистических и графических процедур.

Графические отображения данных

Диаграммы рассеивания

- Одномерные X–Y. К ним относятся: линии, диаграммы рассеивания, оцифрованная диаграмма, связанные диаграммы рассеивания, графики с наборами стандартных ошибок X–Y–Z, диаграмма рассеивания X–Y–Z, чертежный график, переплетенные графики;
- множественные X–Y: точки или линии;
- множественные X–Y–Z.

Разведочные графики

Здесь есть следующие категории:

- график «ящик с усами»: горизонтальный и вертикальный, усеченный, с внешними обозначениями, с маркерами средних;
- графики вероятностей;
- частотные гистограммы: относительные и кумулятивные, гистограмма или полигон.

Деловые карты

К ним относятся:

- графики: горизонтальные и вертикальные, множественные, кластерные, процентные;
- круговые диаграммы: с вырезанными частями, с надписями.

Описание данных

Анализ одной переменной

- Суммарные статистики: среднее, медиана, мода, среднее геометрическое, дисперсия, стандартное отклонение, стандартная ошибка, минимум, максимум, размах, нижний квартиль, верхний квартиль, межквартильный размах, коэффициент асимметрии, нормированный коэффициент асимметрии, коэффициент эксцесса, нормированный коэффициент эксцесса;
- процентили;
- табуляция частот: отношения или кумуляты, график «дерево с листьями», доверительные интервалы;
- проверка гипотез: о среднем и медиане, Т-тест, знаковый тест, знаковый ранговый тест;
- диаграмма рассеивания;
- график «ящик с усами»;
- гистограмма;
- квантильный график
- график нормального распределения;
- график плотности;
- симметричный график.

Анализ множества переменных

- Суммарные статистики;
- доверительные интервалы;
- корреляции;
- ранговые Спирмена;
- частные корреляции;
- ковариации;
- диаграммы рассеивания;
- график «звезда»;
- график «солнечные лучи».

Подбор распределения

- Встроенные распределения: экспоненциальное, экстремальных значений, логнормальное, нормальное, Вейбулла;
- проверка на нормальность: скорректированный хи-квадрат, тест Шапиро—Уилкса, тесты для малых выборок;
- тесты согласия: хи-квадрат, Колмогорова—Смирнова;
- площади остатков;
- критические значения;
- плотности;
- симметричные графики;
- график нормального распределения;
- график распределения Вейбулла;
- частотная гистограмма;
- функции распределения: плотность, распределение кумуляты, функция выживаемости, логарифм функции выживаемости, функция риска.

Табулирование

- Таблица частот: отношения и кумуляты;
- прямоугольные диаграммы;
- круговые диаграммы.

Кросстабуляция

- Таблица частот;
- критерий хи-квадрат;
- измерения связи: лямбда, коэффициенты неопределенности, R Пирсона, D Сомера, эта, коэффициент контингенции, V Крамера, условный Гамма, Тау Кендалла;
- прямоугольные диаграммы;
- мозаичные отображения: горизонтальные и вертикальные;
- трехмерная диаграмма: частот или процентов.

Сравнение данных

Сравнение двух выборок

- Суммарные статистики;
- сравнение средних: Т-тест, доверительные интервалы;

- сравнение стандартных отклонений: отношение дисперсий, F-тест, доверительные интервалы;
- сравнение медиан: тест Манна—Уитнея (Вилкоксона);
- тест Колмогорова—Смирнова;
- гистограммы частот;
- плотности распределения;
- сравнительные графики «ящик с усами»;
- графики Квантиле;
- графики Квантиль-Квантиль.

Сравнение множества выборок

- Суммарные статистики;
- таблица дисперсионного анализа: сумма квадратов, средний квадрат, F-отношение;
- таблица и график средних: стандартные ошибки, доверительные интервалы, наименьшие значимые различия (LSD), Тьюки HSD, Шеффе, Бонферони;
- множественные ранговые тесты: LSD, Тьюки HSD, Шеффе, Бонферони, Ньюмена—Кеулса, Дункана;
- соответствие дисперсий: тест Кокрена, тест Бартлетта, тест Хартлея;
- тест Краскала—Уоллиса;
- диаграммы рассеивания;
- сравнительные графики «ящик с усами»;
- остатки для выборок;
- остатки для прогнозов;
- остатки для наблюдений.

Однофакторный дисперсионный анализ

- Суммарные статистики;
- таблица дисперсионного анализа;
- таблица и графики средних;
- множественные ранговые тесты;
- анализ дисперсии;
- тест Краскалла—Уоллиса;
- диаграмма рассеивания;
- график «ящик с усами»;

- остатки и уровни фактора;
- остатки и описания;
- остатки и номер строки.

Многофакторный дисперсионный анализ

- Таблица дисперсионного анализа: сумма квадратов, тип I; сумма квадратов, тип III;
- таблица средних;
- множественные ранговые тесты;
- диаграмма рассеивания;
- графики средних;
- графики взаимодействий;
- остатки и уровни факторов;
- остатки и описания;
- остатки и номера строк.

Отношения данных

Простая регрессия

- Модели: линейная, экспоненциальная, обратная Y, обратная X, дважды обратная, логарифм X, мультипликативная, квадратный корень X, квадратный корень Y, S-кривая, логистическая, логарифм вероятности;
- T-статистики;
- анализ дисперсии: коэффициент корреляции, R-квадрат, стандартная ошибка оценки;
- прогнозы;
- сравнение альтернативных моделей;
- необычные остатки;
- точки влияния;
- график подобранной модели: описание и доверительные интервалы;
- наблюдения и описания;
- остатки и X: остатки, стьюдентизированные остатки;
- остатки и описание;
- остатки и номер строки.

Множественная регрессия

- Коэффициенты модели;
- T-статистики;
- анализ дисперсии: R-квадрат, скорректированный R-квадрат, стандартная ошибка, средняя абсолютная ошибка, статистика Дурбина—Ватсона, условная сумма квадратов, сумма квадратов, средний квадрат, F-отношение;
- доверительные интервалы;
- корреляционная матрица;
- отчеты: наблюдаемый Y, подогнанный Y, остатки, стьюдентизированные остатки, стандартные ошибки и прогнозы, доверительные границы;
- необычные остатки;
- точки влияния;
- компонентные эффекты;
- наблюдения и описания;
- остатки и X;
- остатки и описания;
- остатки и номер строки;
- интервальные графики: описываемые величины, средние, прогнозы, прогнозы средних.

Общие сведения о работе с базовой системой

Вид экрана после инициализации STATGRAPHICS *Plus* for Windows представлен на рис. 2.6.

Набор кнопок в верхней части окна предназначен для открытия готовых статистических проектов StatFolio и записи новых проектов, для открытия файлов данных и их сохранения, для вывода результатов статистического анализа на печать, а также для вызова некоторых статистических и графических процедур. Эти же операции можно осуществить, войдя в меню File, Edit, Plot, Describe, Compare, Relate и Special.

Внизу экрана расположен набор пиктограмм, связанных со следующими операциями (слева направо):

1. Работа с электронной таблицей.
2. Получение консультации у статистической экспертной системы StatAdvisor.
3. Вызов окна StatGallery.
4. Ввод комментариев к проводимому статистическому анализу.

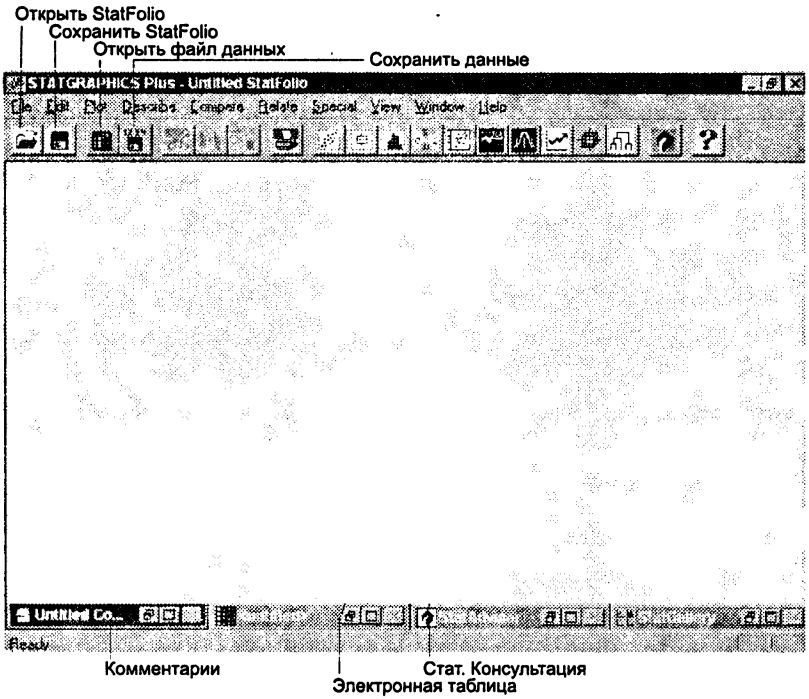


Рис. 2.6. Вид экрана STATGRAPHICS (начало работы)

Так как многие операции будут детально рассмотрены при разборе прикладных примеров, ниже излагаются только самые общие сведения о технологии взаимодействия с пакетом, которые, тем не менее, дают достаточную информацию для того, чтобы начать самостоятельную работу.

Ввод данных

Инициализируем новую электронную таблицу, задействовав соответствующую пиктограмму (Untitled) в левом нижнем углу рабочего поля (рис. 2.7).

Эта таблица организована таким образом, что ее строкам должны соответствовать объекты (наблюдения), а столбцам — признаки. В остальном работа с ней напоминает обращение с другими известными электронными таблицами для Windows типа Lotus, Excel и т. д. Вместе с тем, имеются определенные особенности, связанные со спецификой статистического анализа.

Для именования переменных (признаков) и задания их типа нужно маркировать требуемую колонку и щелкнуть правой кнопкой мыши. Появится контекстное меню, в котором следует выбрать команду Modify Column. Появится одноименное окно диалога (рис. 2.8).

Преобразование переменных и генерация новых признаков осуществляется аналогичным образом: маркируется необходимая колонка и щелчком правой кноп-

ки мыши вызывается контекстное меню, из которого выбирается Generate Data (рис. 2.8). В появившемся окне диалога можно производить арифметические, логические и другие манипуляции с переменными посредством более 100 предоставляемых операторов.

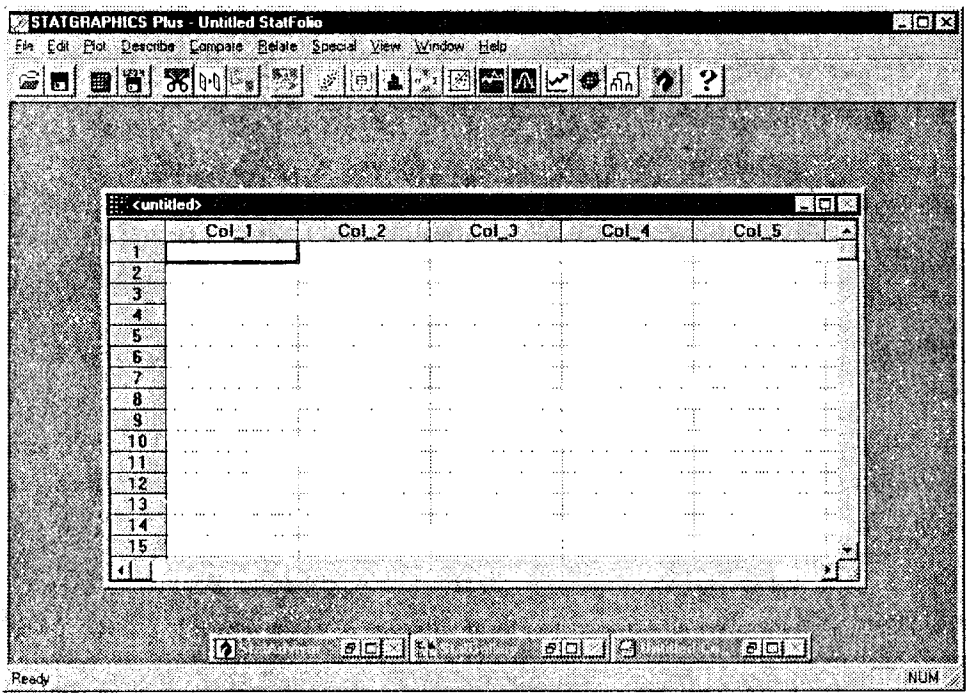


Рис. 2.7. Электронная таблица STATGRAPHICS Plus

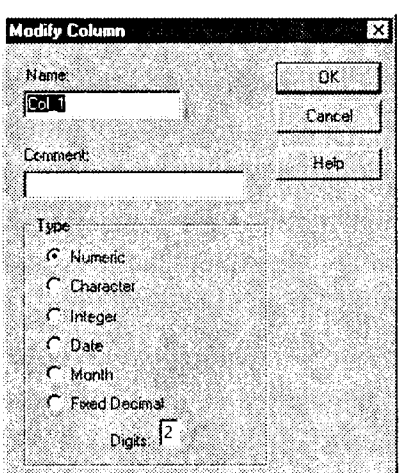


Рис. 2.8. Панель модификации колонки

Для импорта/экспорта данных из других электронных таблиц под Windows используется системный буфер обмена. При этом имеется одна существенная особенность: маркировка нужной области электронных таблиц должна осуществляться только путем буксировки мыши по диагонали выделяемой части таблицы.

После заполнения таблицы для задания имени и сохранения файла данных требуется выбрать команду File ▸ Save Data File As, ввести имя файла и нажать OK. После этой операции в заголовке таблицы появится указанное имя. Оно же будет использоваться в дальнейшем и на пиктограмме файла данных (рис. 2.9).

Будем считать, что данные подготовлены к статистическому анализу. Теперь любые манипуляции с ними будут отражаться в результатах обработки, но никоим образом не отразятся на содержимом сохраненных файлов данных.

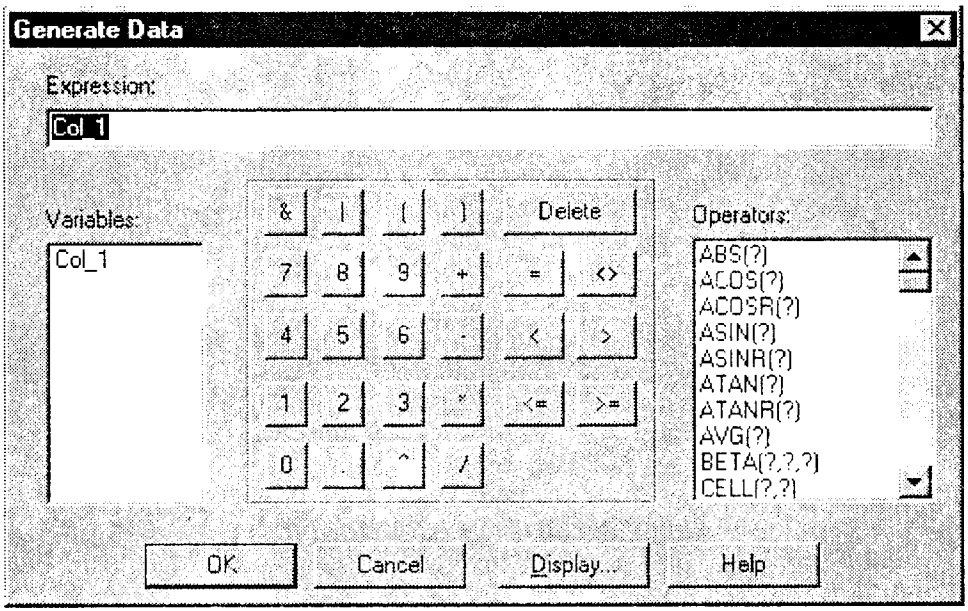


Рис. 2.9. Панель для генерации новых данных

Технология взаимодействия со статистическими и графическими процедурами

Технология взаимодействия с различными статистическими и графическими процедурами пакета во многом стандартизирована, что делает ее удобной для быстрого восприятия и обучения. Продемонстрируем это на простом примере анализа одной переменной.

Откроем файл данных Cardata, в котором представлены характеристики автомашин различных марок.

Выберем Describe ▸ Numeric Data ▸ One Variable Analysis. Появится окно для задания анализируемой переменной. Пусть это будет переменная **horsepower** — мощность автомобиля в лошадиных силах (рис. 2.10). Нажмем OK.

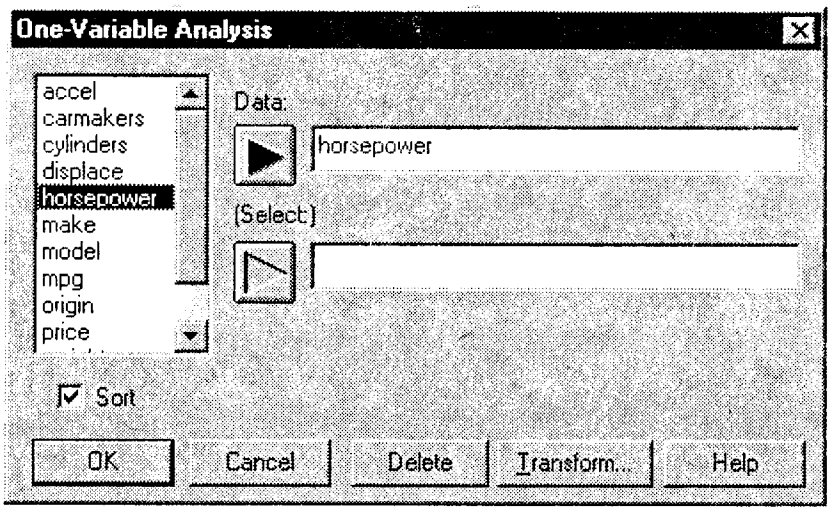


Рис. 2.10. Окно для задания переменных

На экране появится рабочее поле анализа одной переменной со сводкой, в которой констатируется, что изучается переменная horsepower; число наблюдений (объектов) равно 151; значения данной переменной распределены в пределах от 48 до 165. В верхней части рабочего поля расположены кнопки, с помощью которых можно изменять входные данные, выбирать табличные и графические опции и сохранять результаты анализа в файле данных. В нашем случае были установлены следующие флажки: Summary Statistics (общие статистики), Box-and-Whisker Plot (график «ящик с усами»), Frequency Histogram (гистограмма частот) (рис. 2.11).

Окна, в которых отображаются табличные и графические результаты, раскрываются на все рабочее поле двумя щелчками мыши. После раскрытия достаточно щелкнуть правой кнопкой мыши, чтобы получить доступ к специальным меню и задать новые параметры графических изображений или произвести какие-либо изменения и дополнения в текущем анализе данных. Например, в нашем случае, если раскрыть окно общих статистик и щелкнуть правой кнопкой мыши, то на экране возникает окно диалога (рис. 2.11), в котором можно заказать необходимые изменения в наборе выдаваемых статистик (рис. 2.12).

Для изменения элементов графических изображений нужно раскрыть требуемое графическое окно, выделить элемент, подвергаемый трансформации, и щелкнуть правой кнопкой мыши. Затем остается только заказать желаемое изменение в окне диалога.

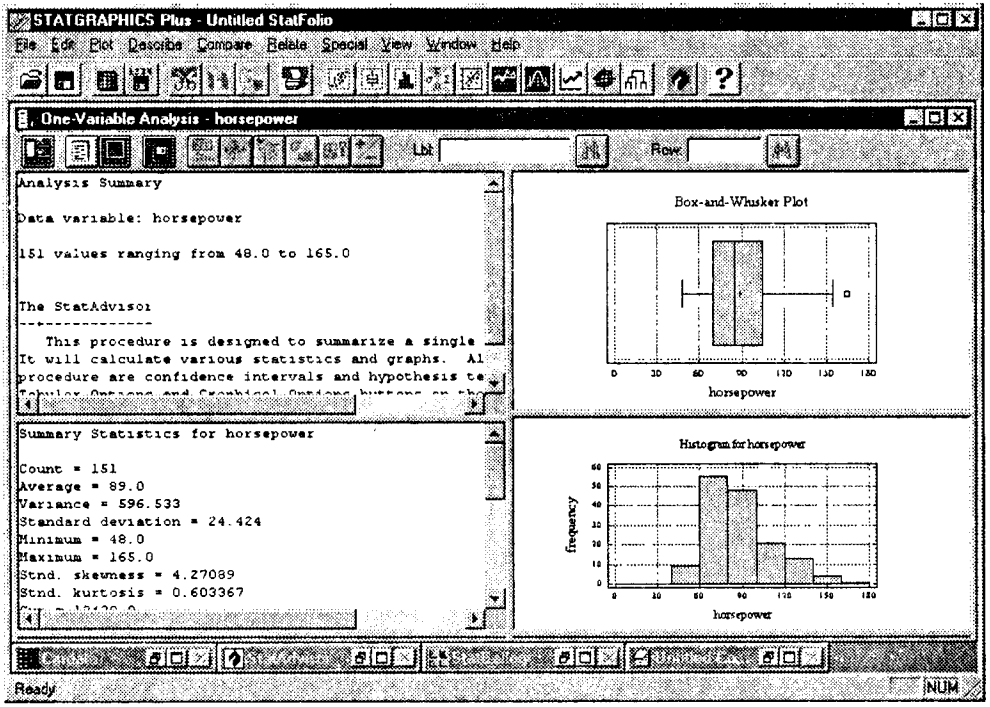


Рис. 2.11. Результаты анализа переменной horsepower

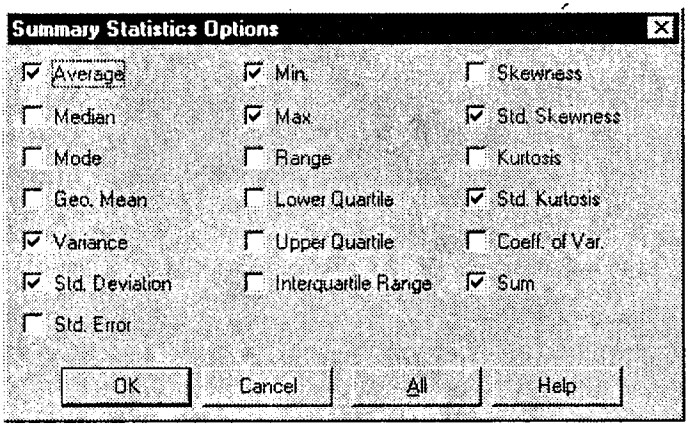


Рис. 2.12. Флажки для задания общих статистик

Таким образом, вся процедура статистического анализа данных находится, по выражению разработчиков STATGRAPHICS, как бы «на кончиках ваших пальцев». Все трансформации производятся быстро через удобные и понятные окна диалога. Это, в свою очередь, способствует включению игрового компонента и

делает увлекательной самую серьезную и ответственную работу по статистическому анализу.

Для того чтобы повторить весь проведенный анализ переменной **horsepower** на новом массиве данных, не прилагая никаких усилий по заданию табличных и графических опций, нужно сохранить анализ в виде файла StatFolio. Для этого производятся стандартные операции File ▸ Save StatFolio As (задание имени статистического проекта). Теперь остается только загрузить новый файл данных File ▸ Open Data File (имя файла данных) и вызвать записанный статистический проект File ▸ Open StatFolio (имя проекта). Все заданные таблицы и графические отображения будут выданы автоматически.

Продемонстрированная на простом примере анализа одной переменной технология взаимодействия со статистическими, графическими и системными процедурами пакета характерна и для других, более изощренных и сложных видов обработки информации. Она, конечно, может несколько различаться в зависимости от специфики применяемых процедур, но в целом достаточно стандартна. Подробности будут представлены в следующих разделах по мере необходимости.

Методы, использующие обучающую информацию

Группа методов анализа данных «с учителем» использует дополнительную информацию, которую несет так называемый внешний критерий. Этот критерий может быть представлен номинальным, ранговым или количественным показателем, привязанным к объектам анализируемой таблицы данных. Привязка номинального показателя означает разбиение исследуемых объектов на классы (группы). Ранговый показатель задает на множестве объектов отношение порядка. В случае количественного показателя отношения между объектами выражаются в какой-либо количественной шкале [43]. Указанный показатель будет в дальнейшем обозначаться «*z*».

Множественный регрессионный анализ

В регрессионном анализе критериальный показатель z рассматривается как «зависимая» переменная (как правило, ранговая или количественная), которая выражается функцией от «независимых» признаков $x_1, \ldots, x_p$. Линейная функция множественной регрессии записывается следующим образом: $z_i = w_0 + w^T \mathbf{x}_i + \varepsilon_i$ w_0 называется свободным членом, а элементы весового вектора $\mathbf{w} = (w_1, \ldots, w_p)^T$ называются коэффициентами регрессии.

Для оценки эффективности регрессионного уравнения вводится вектор остатков $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_N)^T$, который отражает влияние на z-совокупности неучтенных случайных факторов либо меру достижимой аппроксимации значений z_i.

Различают два подхода к определению параметров уравнения множественной регрессии в зависимости от происхождения матрицы данных. В первом считается, что признаки детерминированы и случайной величиной является только зависимая переменная z. Этот подход используется наиболее часто. Во втором подходе полагается, что и независимые признаки x_j и z — случайные величины, имеющие совместное распределение. В такой ситуации оценка уравнения регрессии есть оценка условного математического ожидания случайной величины z в зависимости от случайных величин x_j.

Каждый из приведенных подходов имеет свои особенности. Вместе с тем, показано, что они отличаются только статистическими свойствами оценок параметров уравнения регрессии, тогда как вычислительные аспекты этих моделей совпадают [24].

В уравнении множественной регрессии обычно полагают, что величины ε_i $(i = \overline{1,N})$ независимы и случайно распределены с нулевым средним и дисперсией σ_e^2, а оценка параметров w_0 и $\mathbf{w}$ производится с помощью метода наименьших квадратов (МНК). Ищется минимум суммы квадратов невязок:

$$\Delta^2 = \sum_{i=1}^{N}(z_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - w_0)^2$$

Это приводит к нормальной системе уравнений со следующим решением:

$$\mathbf{w} = \mathbf{S}^{-1}\mathbf{c}_{zx};$$

$$w_0 = m_z - \mathbf{w}^{\mathrm{T}}\mathbf{m}_x,$$

где $\mathbf{c}_{zx}$ — вектор оценок ковариации между внешним критерием z и признаками $x_1, \ldots, x_p$; m_z — оценка среднего значения z; $\mathbf{m}_x$ и $\mathbf{S}$ — вектор средних значений и матрица ковариаций признаков $x_1, \ldots, x_p$.

Основным показателем качества уравнения множественной регрессии является коэффициент детерминации (квадрат коэффициента множественной корреляции)

$$R^2 = \frac{N\sigma_z^2 - \Delta^2}{N\sigma_z^2},$$

σ_z^2 — оценка дисперсии прогнозируемой переменной $\sigma_z^2 = \frac{1}{N}\sum_{i=1}^{N}(z_i - m_z)^2$.

Статистический смысл коэффициента детерминации заключается в том, что он показывает, какая доля дисперсии зависимой переменной z объясняется построенной функцией регрессии. Например, при коэффициенте детерминации 0,49 регрессионная модель объясняет 49 % дисперсии внешнего критерия, остальные же 51 % считаются обусловленными факторами, не отраженными в регрессионное уравнение.

Еще одним важным показателем качества уравнения множественной регрессии является статистика $F = \frac{N-p-1}{p} \times \frac{R^2}{1-R^2}$. С помощью этой статистики проверяется гипотеза H_0: $w_1 = w_2 = \ldots = w_p = 0$, то есть гипотеза о том, что совокупность

признаков $x_1, \ldots, x_p$ не улучшает описания критериального показателя по сравнению с тривиальным описанием $z_i = m_z$. Если $F > f_{p, N-p-1}$, где $f_{p, N-p-1}$ — случайная величина, имеющая F-распределение с p и $N-p-1$ степенями свободы, то H_0 отклоняется (критерий Фишера).

Ниже приведено уравнение множественной регрессии, связывающее себестоимость продукции на 10 отечественных предприятиях (зависимая переменная) с валовым объемом и производительностью труда (независимые переменные — предикторы):

себестоимость = 0,8×(валовой объем) + 0,3×(производительность труда).

По статистическим критериям доверие к выявленной зависимости более 99 %. Вместе с тем, часто для иллюстрации полученного уравнения множественной регрессии используют график «наблюдения-предсказания» (рис. 2.13).

Рис. 2.13. Иллюстрация полученной модели множественной регрессии

При большем, чем в рассмотренном примере, числе переменных нередко приходится решать задачу отбора таких переменных в уравнение множественной регрессии. При этом наиболее популярными являются три алгоритма отбора: последовательного увеличения, последовательного уменьшения и алгоритм плюс l минус r, поочередно «работающий» то на добавление, то на удаление переменных. Все алгоритмы являются эвристическими и не гарантируют сходимости к оптимальному решению. Поэтому на практике попытки приблизиться к желаемому оптимуму всегда сопряжены с комбинированным применением указанных алгоритмов.

Пример пошаговой множественной регрессии: взаимосвязь психического и биологического

Пошаговая множественная регрессия применяется для минимизации количества независимых переменных, входящих в исследуемую модель. Известно много подходов к такой минимизации. В Базовой системе STATGRAPHICS *Plus* for Windows реализованы две наиболее популярные процедуры: последовательное увеличение и последовательное уменьшение группы независимых переменных.

Рассматриваемый пример относится к одной из наиболее важных и интересных проблем современных научных исследований.

Соотношение сознания и вещественного мира — одна из ключевых проблем современной науки. Представление о Вселенной как о гигантской супермашине, собранной из бесчисленных отдельных объектов и существующей независимо от наблюдателя, отошло в прошлое. Новые модели Вселенной предполагают, что связующим принципом в космической сети выступает сознание — первичный атрибут существования. Ряд известных физиков (Ю. Вигнер, Д. Бом, Дж. Чу, Ф. Капра, А. Янг и др.) высказываются за включение сознания в качестве неотъемлемой и главной части будущей глобальной теории материи.

На фоне эволюции идей о структуре мира биологические и психологические исследования выглядят, может быть, менее масштабно, но не менее весомо. Их основные темы концентрируются около фундаментальной задачи изучения взаимосвязей различных уровней биоорганизации: ген — клетка — организм — психика. Раскрытие данных взаимосвязей, кроме локальных целей, призвано в конечном счете дать ответ на вопрос о биологической обусловленности психики и о свободе воли как о важнейшей составляющей индивидуального сознания.

Известно много достижений, например, психофизиологии, описывающей связи физиологических процессов в организме с проявлениями психического, нейропсихологии (о зависимостях между особенностями функционирования нейронных ансамблей и психическими свойствами), генетической психологии (о наследовании различных черт характера, темперамента, психических заболеваний) и др. Вместе с тем накопленные сведения основаны на различных теоретических базах, экспериментальных технологиях, описаны на разных научных языках и пока не поддаются междисциплинарному обобщению, не говоря уже об интеграции с современными космологическими теориями.

Всеобщие законы подобия между предметами и явлениями умели находить восточные мыслители. Их философские концепции, ориентированные на модель «Человек во Вселенной», легли в основу теоретических и практических положений медицины, которые были изложены в многочисленных трактатах (например, китайские «Ней-цзин», «Хуай нань-цзы», тибетский медицинский трактат «Жуд-Ши» и др.). В этих трудах отмечается, что организм человека нужно рассматривать как единое целое, и между работой сердца, центральной нервной системой и внутренними органами человека существует тесная связь. Говорится о влиянии внутренних органов человека на его нравственные черты и описываются связи между некоторыми психическими свойствами и состоянием внутренних органов.

Таким образом, на качественном уровне связи элементов организма человека с его психическими свойствами были известны уже давно. Однако количественное выражение подобных связей стало возможным лишь в наши дни благодаря соединению ряда обстоятельств.

Во-первых, к ним относится модернизация восточных знаний по акупунктурной диагностике организма человека и развитие технических средств измерений в биологически активных точках (БАТ), в частности средств электропунктурной диагностики. Во-вторых, важным обстоятельством является современный уровень психодиагностических тестов, позволяющий с достаточно высокой точностью и надежностью проводить психологические измерения. И в-третьих, извлечение закономерностей из результатов электропунктурных и психологических

измерений стало возможным на основе развитой технологии компьютерного многомерного анализа данных.

Направление исследований взаимосвязей соматического профиля человека, определяемого методами электропунктурной диагностики, с его психологическими характеристиками получило название электропунктурной психодиагностики. Ниже приводятся сведения об одном из последних экспериментов в этой области.

Испытуемыми были студенты (мужского пола, возраст 20–22 года) Санкт-Петербургского государственного технического университета. У каждого из них проводилось измерение электрокожного сопротивления с помощью автоматизированного комплекса рефлексотерапевта «АКРО» в 24 биологически активных точках (12 слева и 12 справа), являющихся проекциями отдельных органов (табл. 2. 4). Кроме того, каждый студент тестировался по психологической методике Шмишека—Мюллера, диагностирующей 10 акцентуаций характера (табл. 2.5).

Таблица 2.4. Наименование и обозначение репрезентативных точек

Обозначения	Канал	Точка
P	Легкие	9-I тай-юань
GI	Толстый кишечник	5-II ян-си
E	Желудок	42-III чун-ян
RP	Селезенка	3-IV тай-бай
C	Сердце	7-V шэнь-мэнь
IG	Тонкий кишечник	4-VI вань-гу
V	Мочевой пузырь	65-VII шу-гу
R	Почки	3-VIII тай-си
MC	Перикард	7-IX да-лин
TR	Тройной обогреватель	4-X ян-чи
VB	Желчный пузырь	40-XI цю-сюй
F	Печень	3-XII тай-чун

Таблица 2.5. Диагностируемые психологические свойства

№	Название акцентуации	Краткая характеристика
1	Гипертимность	Активность, энергичность, оптимистичность, с высоким жизненным тонусом
2	Застревание	Длительное переживание одних и те же чувств, упрямство, сопротивление изменениям
3	Эмотивность	Богатство эмоциональных реакций, изменчивость настроения
4	Педантичность	Приверженность к определенному порядку, плохое переключение на новое в деятельности
5	Тревожность	Ощущение неблагополучия, внутренней напряженности
6	Циклотимность	Периоды ровного настроения чередуются с подъемами и субдепрессивными фазами

продолжение

Таблица 2.5 *(продолжение)*

№	Название акцентуации	Краткая характеристика
7	Демонстративность	Эгоцентричность, стремление постоянно быть в центре внимания
8	Возбудимость	Агрессивность, упрямство, самолюбие, обидчивость
9	Дистимность	Частые и длительные изменения настроения в сторону его снижения
10	Экзальтированность	Склонность приходить в состояние восторженного возбуждения по незначительным поводам и впадать в отчаяние под влиянием разочарования

Ввод и преобразование данных

Откроем окно таблицы данных и введем следующие результаты измерений электрокожного сопротивления в биологически активных точках (табл. 2.6) у 14 студентов.

Таблица 2.6. Результаты измерений электрокожного сопротивления в биологически активных точках справа (x — фоновые, y — после нагрузки)

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
1	80	107	44	87	85	214	47	91	69	77	54	86
2	94	93	72	106	130	254	75	97	130	69	99	76
3	95	115	81	186	73	79	95	111	96	129	114	110
4	99	126	87	74	69	107	61	121	65	176	85	78
5	117	53	94	122	139	113	101	144	108	59	127	148
6	78	80	58	112	66	195	96	94	86	79	144	142
7	91	95	120	82	131	142	62	140	106	50	76	116
8	104	52	45	113	136	385	65	53	114	180	82	84
9	55	54	65	140	77	173	117	117	82	100	90	107
10	130	109	51	34	129	302	37	126	123	126	60	62
11	64	66	70	186	61	204	157	129	66	86	99	69
12	81	82	94	167	78	112	153	130	85	75	138	149
13	58	144	49	53	53	289	55	115	82	135	105	81
14	91	144	49	49	82	293	64	88	113	124	90	60

y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	y11	y12
113	115	67	163	88	293	96	98	94	93	79	58
124	75	76	152	86	125	108	139	133	60	101	62
122	107	77	191	91	101	96	142	99	123	116	147
124	78	75	105	82	121	89	83	78	171	82	97
112	84	92	131	124	154	91	139	97	100	116	120
8758	90	221	60	137	127	227	109	79	159	119	
9774	88	94	113	191	61	125	111	95	85	75	

y1	Y2	y3	y4	y5	y6	y7	y8	y9	y10	y11	y12
111	43	84	116	88	284	91	70	108	147	124	71
70	100	63	132	99	209	104	141	91	130	79	97
123	74	67	74	66	194	61	115	118	106	100	81
76	76	42	231	62	156	140	130	72	79	117	66
75	59	96	170	86	94	119	119	89	63	126	116
69	115	77	72	48	274	89	100	67	146	99	74
80	83	66	56	70	235	70	88	117	107	89	71

Введем в эту же таблицу данные психологического тестирования.

Результаты тестирования (Z-акцентуации)

	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10
1	18	18	12	12	3	6	10	0	9	12
2	12	12	21	22	6	6	10	15	9	12
3	12	8	3	10	3	12	8	6	6	12
4	12	20	12	14	18	9	12	15	6	12
5	9	18	18	12	0	12	16	6	15	12
6	24	12	21	16	9	12	18	9	9	12
7	21	6	12	18	12	15	10	6	12	6
8	6	4	6	10	3	9	14	9	9	6
9	18	12	0	4	0	6	20	3	6	12
10	15	10	12	12	3	12	10	12	15	6
11	15	14	24	6	3	15	20	21	6	12
12	21	12	9	10	0	6	18	3	3	12
13	24	12	15	12	15	12	20	12	9	12
14	12	12	3	14	2	9	8	12	15	6

Как уже отмечалось, удалось построить статистически значимые регрессионные модели практически для всех акцентуаций характера. При этом в модели входили результаты измерения ЭКС, как фоновые, так и после нагрузки. В то же время значительный интерес представляют новые переменные, представляющие собой отношения фона к нагрузке. Для создания таких переменных произведем следующие операции.

Выделим новую колонку в таблице данных и щелкнем правой кнопкой мыши. В появившемся меню выберем Generate Data. В поле Expression диалогового окна Generate Data введем требуемое преобразование — отношение фонового замера к измерению после нагрузки (рис. 2.14). Нажмем OK.

Выберем в контекстном меню команду Modify Column. Зададим в поле Name имя новой переменной **rr1**. В поле Comment запишем комментарий «отношение фона к нагрузке» (рис. 2.15). Нажмем кнопку OK. Аналогичным образом сгенерируем остальные переменные **rr2–rr12**.

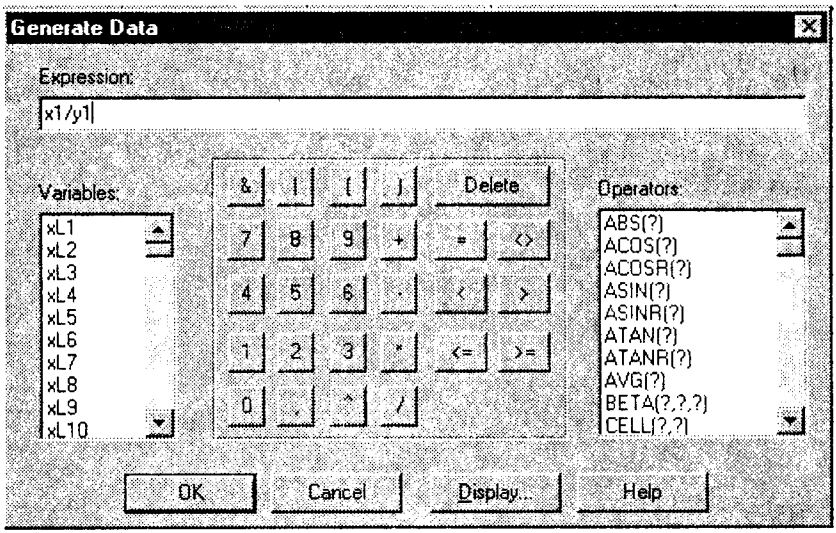


Рис. 2.14. Окно диалога для генерации новых переменных

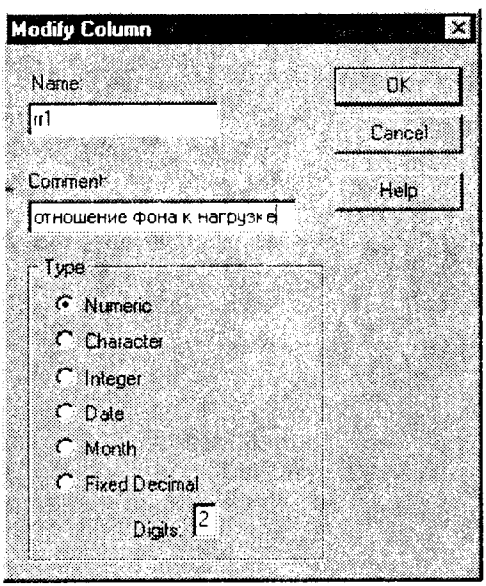


Рис. 2.15. Окно для наименования переменных, задания их типа и ввода комментариев

Построение модели множественной регрессии для всех переменных

Выберем из меню Relate пункт Multiple Regression. В окне диалога множественной регрессии с помощью кнопки со стрелкой активизируем поле Dependent Variable (зависимая переменная). Затем в списке переменных, находящемся слева, исполь-

зуя прокрутку, найдем требуемую переменную. Пусть в рассматриваемом случае это будет акцентуация характера z8 — «возбудимость». Дважды щелкнем на этой переменной, и она появится в активном поле.

Выделим в списке переменных из левого поля окна Multiple Regression переменные rr1–rr12 и нажмем кнопку со стрелкой, указывающую на поле Independent Variables (независимые переменные). Все маркированные переменные будут включены в анализ (рис. 2.16). Нажмем кнопку OK.

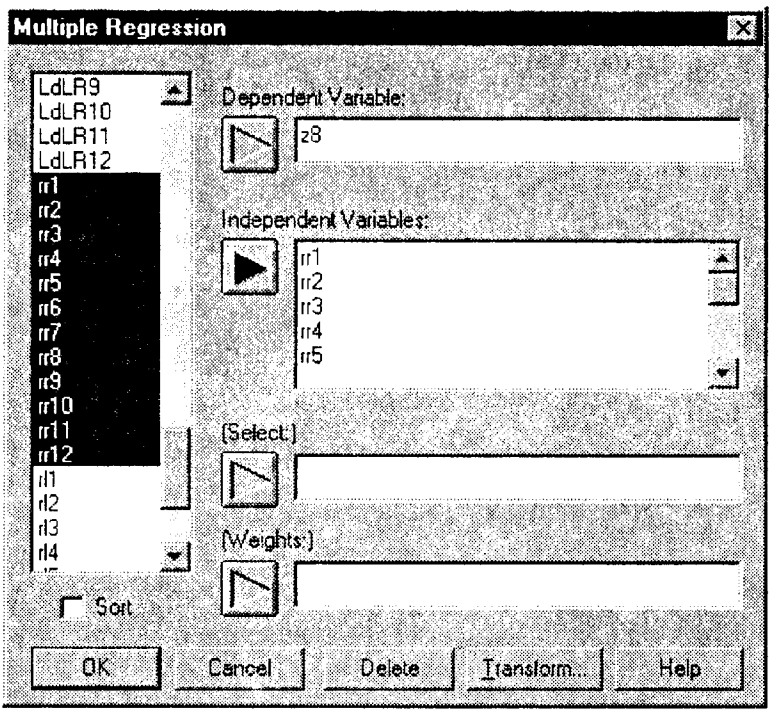


Рис. 2.16. Ввод данных в множественный регрессионный анализ

После того как мы нажали кнопку OK на экран выдается сводка проведенного анализа (рис. 2.17).

Из представленной сводки мы получаем сведения: об оценках величины константы и весовых коэффициентов в уравнении регрессии, о стандартных ошибках, Т-статистиках и р-значениях для полученных величин. Но главное, на что следует обратить внимание, — это высокое р-значение во второй таблице «Analysis of Varians» (Анализ дисперсии), где оценивается модель в целом. Оно составляет 0,4028, что говорит об очень низкой статистической значимости построенной модели. Это неудивительно, ведь, используя 12 переменных, мы имеем выборку объемом всего 14 человек. Путь, по которому следует идти в данном случае, — заключается в попытке снизить количество переменных в правой части уравнения регрессии, применив метод пошагового отбора.

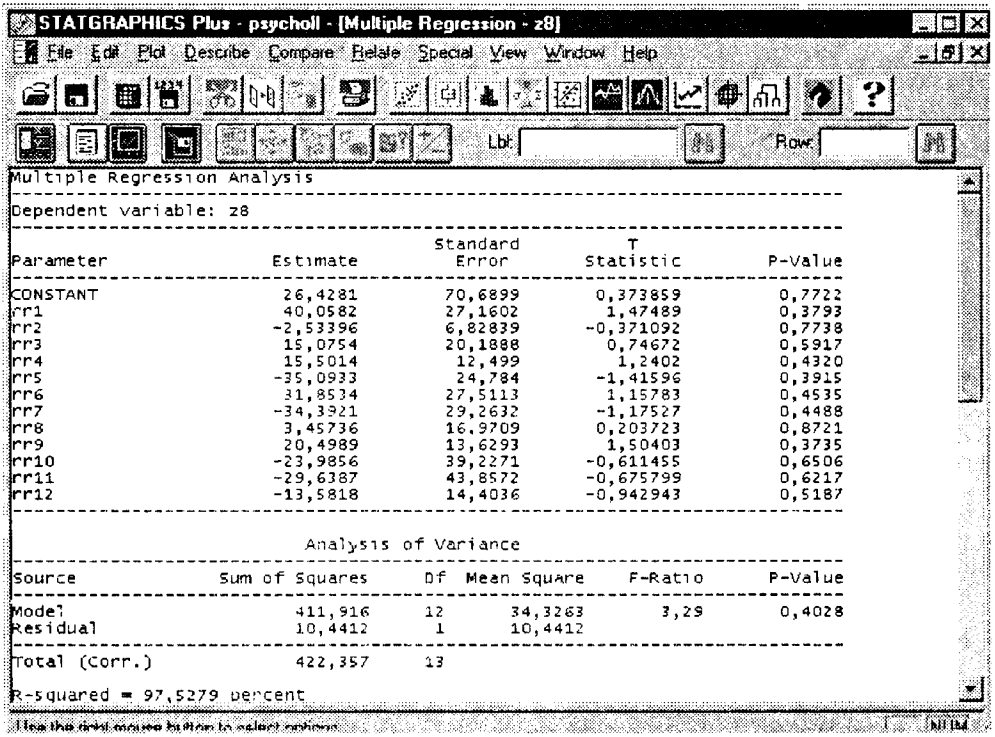


Рис. 2.17. Сводка множественного регрессионного анализа

Пошаговый отбор переменных

Щелкнем правой кнопкой мыши и выберем из появившегося меню пункт Analysis Options. В разделе Fit окна диалога установим переключатель в положение Forward Selection (алгоритм последовательного увеличения группы переменных). Все остальное оставим без изменений (рис. 2.18). Нажмем OK. Получаем новую сводку регрессионного анализа (рис. 2.19). Как видно из таблиц, построена регрессионная модель, обладающая высокой статистической значимостью и объясняющая почти 66,9 % дисперсии зависимой переменной z1.

Опробуем теперь процедуру с последовательным уменьшением группы переменных. Выберем Analysis Options в контекстном меню. Установим переключатель Fit в положение Backward Selection и снимем флажок Constant in Model. Остальные элементы управления оставим без изменений. Получим следующую сводку результатов работы процедуры (рис. 2.20).

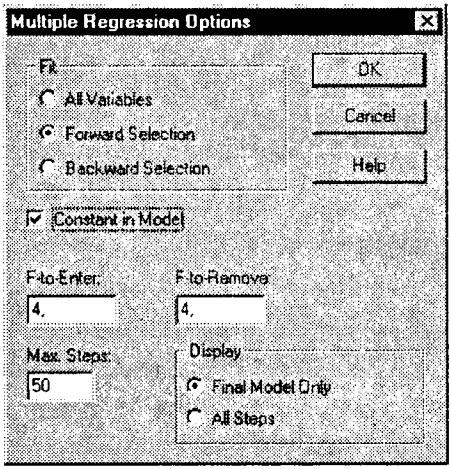


Рис. 2.18. Окно диалога для задания параметров процедуры пошаговой регрессии

STATGRAPHICS Plus - psycholl - [Multiple Regression - z8]

File Edit Plot Describe Compare Relate Special View Window Help

Lbl: Row:

Multiple Regression Analysis

Dependent variable: z8

Parameter	Estimate	Standard Error	T Statistic	P-Value
CONSTANT	0,905482	7,23613	0,125133	0,9029
rr3	8,45161	3,37965	2,50074	0,0314
rr6	8,77926	2,78849	3,14839	0,0104
rr12	-8,65032	4,18709	-2,06595	0,0657

Analysis of Variance

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
Model	282,411	3	94,1371	6,73	0,0092
Residual	139,946	10	13,9946		
Total (Corr.)	422,357	13			

R-squared = 66,8655 percent
R-squared (adjusted for d.f.) = 56,9252 percent
Standard Error of Est. = 3,74093

Use the right mouse button to select options NUM

Рис. 2.19. Сводка регрессионного анализа с пошаговым добавлением переменных

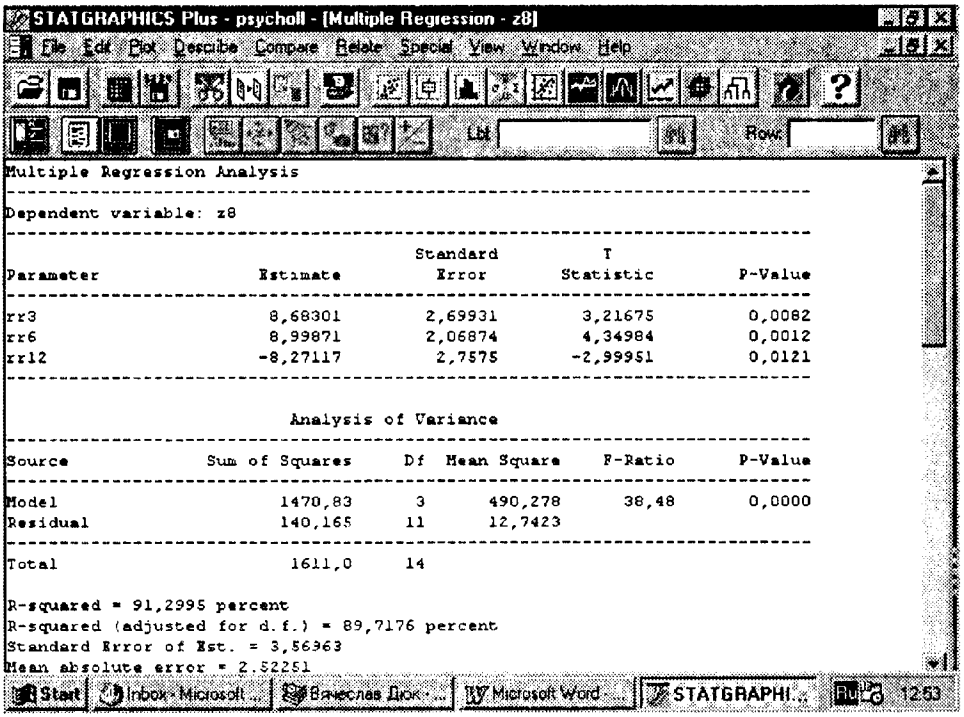


Рис. 2.20. Результаты работы процедуры последовательного уменьшения группы переменных

Видно, что построенная регрессионная модель обладает значительно лучшими свойствами, чем предыдущая. В нее вошли три переменные: **rr3** (желудок), **rr6** (тонкий кишечник) и с обратным знаком **rr12** (печень). Данная модель объясняет уже 91 % дисперсии зависимой переменной; также высок (89,7 %) коэффициент детерминации, скорректированный с учетом степеней свободы (adjusted R-squared). При этом взаимоотношения переменных, зафиксированные в модели, заслуживают почти 100-процентного доверия.

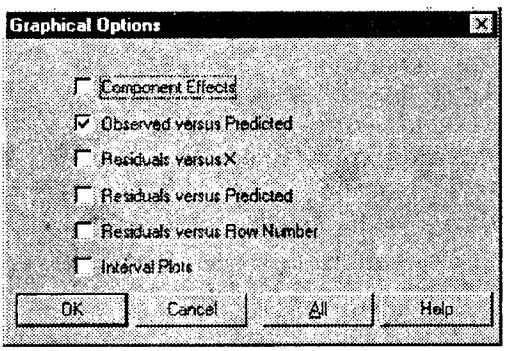


Рис. 2.21. Окно графических параметров

Отобразим графически полученные результаты. Для этого нажмем кнопку графических опций (третья слева в нижнем ряду) и в окне диалога установим флажок Observed versus Predicted (наблюдение — предсказание). Нажмем OK (рис. 2.21). На экране образуется второе окно с требуемым графическим отображением. Раскроем его на весь экран, щелкнув дважды левой кнопкой мыши на заголовке (рис. 2.22).

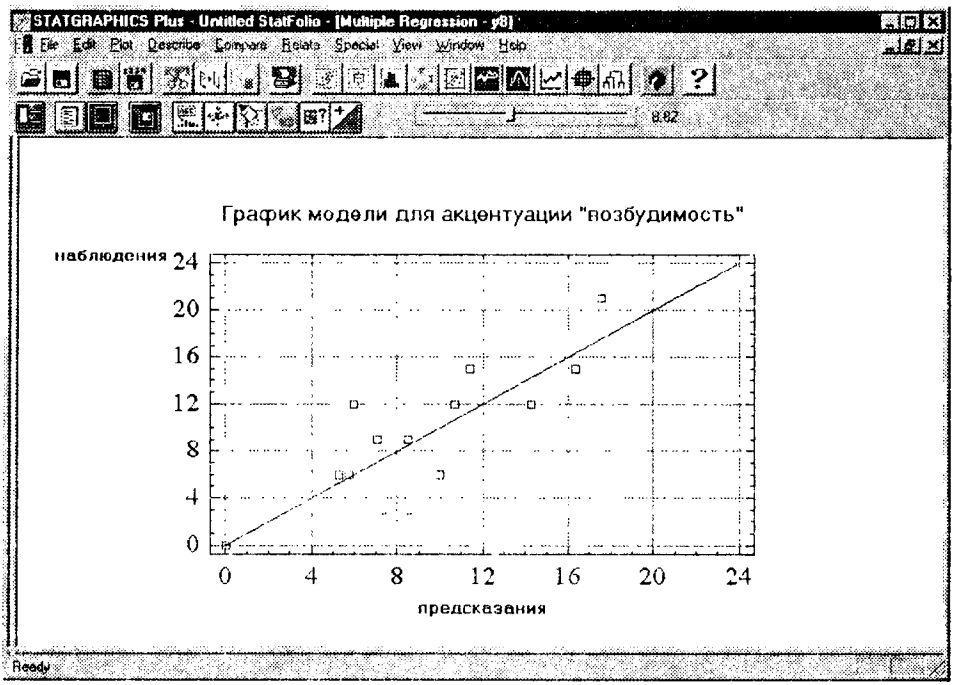


Рис. 2.22. Графическое отображение регрессионной модели

Резюме

Представленный пример демонстрирует эффективность процедур последовательного увеличения и уменьшения группы переменных при построении моделей множественной регрессии. Удается подбирать модели, содержащие гораздо меньше переменных по сравнению с исходным множеством и имеющие лучшие статистические характеристики. Незначительное количество переменных позволяет легко интерпретировать содержание регрессионных моделей. Так, в нашем случае уравнение регрессии после применения процедуры уменьшения группы переменных выглядит следующим образом:

z8 = 8,7(желудок) + 9,0(тонкий кишечник) – 8,3(печень)

Итак, можно предположить, что для людей с повышенной возбудимостью характерной реакцией на функциональную пробу является повышение активности желудка и тонкого кишечника при одновременном угнетении функции печени.

И наоборот, организм людей с пониженной возбудимостью реагирует на нагрузку снижением активности работы желудка и тонкого кишечника при одновременном увеличении активности работы печени.

Полученные данные в настоящее время подвергаются дальнейшим проверкам. Их окончательное подтверждение сулит значительные перспективы в следующих областях:

1. *Собственно психодиагностика.* Трудоемкая и громоздкая процедура психологического тестирования, сопровождаемая возможностями преднамеренных и непреднамеренных фальсификаций, заменяется в ряде случаев оперативной и объективной процедурой измерения электрокожного сопротивления в биологически активных точках с последующей релевантной обработкой.
2. *Психотерапия.* Раскрытие взаимосвязей особенностей функционирования органов человека с его психическими свойствами на количественном уровне создает предпосылки для создания эффективных методик психокоррекции.
3. *Психогигиена и психопрофилактика.* Электропунктурная психодиагностика вследствие объективности измерений позволяет обнаруживать нежелательные тенденции в психическом статусе на ранних стадиях.
4. *Мониторинг психического состояния.* Оперативная процедура электропунктурной психодиагностики дает возможность отслеживать изменения психического состояния в реальном времени.
5. *Соматическая медицина.* Раскрытие взаимосвязей психики и соматики создает предпосылки для разработки методик направленного соматического воздействия через создание определенных психических состояний с учетом межполушарной асимметрии мозговых процессов.

Приведенный список можно было бы продолжить вплоть до исследований генетической обусловленности психических особенностей человека и поиска границы, начиная с которой психическое становится самостоятельной сущностью. По-видимому, для этого настанет свое время.

Дискриминантный анализ

Если критериальный показатель z измерен в номинальной шкале или связь этого показателя с исходными признаками является нелинейной, то для раскрытия закономерностей в данных и построения решающих правил используются методы дискриминантного анализа. В этом случае объекты в соответствии с внешним критерием разбиваются на группы (классы) и пространство признаков рассматривается под углом зрения способности разделять (дискриминировать) выделенные классы.

Большая группа методов дискриминантного анализа основана на байесовской схеме принятия решений о принадлежности объектов к тем или иным классам. Байесовский подход базируется на предположении, что задача сформулирована в терминах теории вероятностей и известны все представляющие интерес величины: априорные вероятности $P(\omega_i)$ для классов ω_i $(i=\overline{1, K})$ и условные плотно-

сти распределения значений вектора признаков $P(\mathbf{x}/\omega_i)$. Правило Байеса заключается в нахождении апостериорной вероятности, $P(\omega_i/\mathbf{x})$, которая вычисляется следующим образом:

$$P(\omega_i/\mathrm{x}) = \frac{P(\mathrm{x}/\omega_i)P(\omega_i)}{P(\mathrm{x})},$$

где

$$P(\mathrm{x}) = \sum_{j=1}^{K} P(\mathrm{x}/\omega_i)P(\omega_i).$$

Решение о принадлежности объекта $\mathbf{x}_k$ к классу ω_j принимается при выполнении условия, обеспечивающего минимум средней вероятности ошибки классификации:

$$P(\omega_j/\mathrm{x}_k) = \max_{i=\overline{1,K}} P(\omega_i/\mathrm{x}_k).$$

Для бинарных признаков, с которыми часто приходится иметь дело, принимающих значение 0 либо 1, p-мерный вектор признаков $\mathbf{x}$ может принимать одно из 2^n дискретных значений $\mathrm{v}_1, \ldots, \mathrm{v}_n$. Функция плотности $P(\mathbf{x}/\omega_i)$ становится сингулярной и заменяется на $P(\mathrm{v}_k/\omega_i)$ — условную вероятность того, что $\mathbf{x} = \mathrm{v}_k$ при условии класса ω_i.

Другие подходы в дискриминантном анализе используют геометрические представления о разделении классов в пространстве признаков. Это следующие представления.

Совокупность объектов, относящихся к одному классу, образует «облако» в p-мерном пространстве, задаваемом исходными признаками. Для успешной классификации необходимо, чтобы [30]:

1. Облако из ω_i в основном было сконцентрировано в некоторой области D_i пространства признаков.
2. В область D_i попала незначительная часть «облаков» объектов из других классов.

Построение решающего правила рассматривают как задачу поиска K непересекающихся областей $D_i\ (i = \overline{1,K})$, удовлетворяющих условиям 1 и 2. Дискриминантные функции (ДФ) определяют эти области посредством описания их границ в пространстве признаков.

Если какой-либо объект попадает в область D, то будем считать, что принимается решение о его принадлежности к классу ω_i. Обозначим $P(\omega_i/\omega_j)$ — вероятность того, что объект из класса ω_j ошибочно попадает в область D_i. Тогда критерием правильного определения областей D_i будет

$$Q = \sum_{i=1}^{K-1} \sum_{j>i}^{K} P(\omega_i)P(\omega_i/\omega_j),$$

где $P(\omega_i)$ — априорная вероятность появления объекта из ω_i.

Приведенный критерий называют критерием средней вероятности ошибочной классификации. Его минимум достигается при использовании, в частности, рассмотренного выше байесовского подхода, который, однако, реализуется лишь при невысоких размерностях пространства признаков.

В теории анализа многомерных данных всесторонне разработаны процедуры построения линейных дискриминантных функций (ЛДФ), обеспечивающих при определенных предположениях минимум критерия средней вероятности ошибочной классификации. Так, для случая двух классов ω_1 и ω_2 методы построения (ЛДФ) опираются на два предположения.

Первое состоит в том, что области D_1 и D_2, в которых концентрируются объекты из двух классов, могут быть разделены $(p-1)$-мерной гиперплоскостью

$$y(\mathbf{x}) + w_0 = w_1 x_1 + w_2 x_2 + \ldots + w_p x_p = 0.$$

Коэффициенты ω_i в данном случае интерпретируются как параметры, характеризующие наклон гиперплоскости к координатным осям, а w_0 называются порогом и соответствует расстоянию от гиперплоскости до начала координат. Преимущественное расположение объектов одного класса, например ω_1, по одну сторону гиперплоскости выражается в том, что для них выполняется условие $y(\mathbf{x}) < 0$, а для объектов другого класса ω_2 — обратное условие $y(\mathbf{x}) > 0$.

Второе предположение касается критерия качества разделения областей D_1 и D_2 гиперплоскостью $\mathrm{y}(\mathbf{x})+w_0=0$. Наиболее часто предполагается, что разделение будет тем лучше, чем дальше отстоят друг от друга средние значения случайных величин $m_1 = \mathrm{E}\{y(\mathbf{x})\}$, $\mathbf{x} \in \omega_1$ и $m_2 = \mathrm{E}\{\mathrm{y}(\mathrm{x})\}$, $\mathrm{x} \in \omega_2$, где $\mathrm{E}\{\ ?\ \}$ — оператор усреднения.

В простейшем случае полагают, что классы ω_1 и ω_2 имеют одинаковые ковариационные матрицы $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}$. Тогда вектор оптимальных весовых коэффициентов $\mathbf{w}$ определяется следующим образом [18]:

$$\mathbf{w} = \mathbf{S}^{-1}(\mathbf{m}_1 - \mathbf{m}_2),$$

где $\mathbf{m}_i$ — вектор средних значений признаков для класса ω_i.

Для определения величины порога w_0 вводят предположение о виде законов распределения объектов. Если объекты каждого класса имеют многомерное нормальное распределение с одинаковой ковариационной матрицей $\mathbf{S}$ и векторами средних значений $\mathbf{m}_i$, то пороговое значение w_0, минимизирующее критерий средней вероятности ошибки, будет

$$\mathrm{w}_0 = \frac{1}{2}\mathrm{w}^{\mathrm{T}}(\mathrm{m}_1 + \mathrm{m}_2) + \ln\frac{P(\omega_1)}{P(\omega_2)}.$$

Для случая, когда число классов больше двух ($K > 2$), обычно определяются K дискриминантных векторов

$$\mathbf{w}_i = \mathbf{S}^{-1}\mathbf{m}_i \quad (i = \overline{1,K})$$

и пороговые величины

$$w_{0i} = -\frac{1}{2}\mathbf{w}_i^{\mathrm{T}}\mathbf{m}_i + \ln P(\omega_i)\ (i = \overline{1,K}).$$

Объект $\mathbf{x}_k$ относится к классу ω_i, если выполняется условие

$$g_i(\mathbf{x}_k) = \max_{j=\overline{1,K}} g_j(\mathbf{x}_k),$$

где $g_j(\mathbf{x}) = \mathbf{w}_j^{\mathrm{T}}\mathbf{x} - w_{0j}$.

В формулы вычисления пороговых значений входят величины априорных вероятностей $P(\omega_i)$. Априорная вероятность $P(\omega_i)$ соответствует доле объектов, относящихся к классу ω_i в большой серии наблюдений, проводящихся в некоторых стационарных условиях. Обычно $P(\omega_i)$ неизвестны. Поэтому при решении практических задач, не меняя дискриминантных векторов, эти значения задаются на основании субъективных оценок исследователя. Также нередко полагают эти значения равными или пропорциональными объемам обучающих выборок из рассматриваемых диагностических классов.

Известны другие подходы к построению дискриминантных функций. Широко распространен классический вариант дискриминантного анализа, основанный на определении канонических направлений в исходном пространстве признаков, удовлетворяющих следующему критерию:

$$J = \frac{\text{Дисперсия между классами}}{\text{Дисперсия внутри классов}} = \max.$$

Весовой вектор $\mathbf{w}$, удовлетворяющий данному критерию, исходя из геометрической интерпретации, задает новую координатную ось в исходном p-мерном пространстве признаков $y(\mathbf{x})=\mathbf{w}^{\mathrm{T}}\mathbf{x}$ ($\|\mathbf{w}\|=1$) с максимальной неоднородностью исследуемой совокупности объектов. Новой оси соответствует, по существу, первая главная компонента объединенной совокупности объектов, полученная с учетом дополнительной обучающей информации о принадлежности объектов различным классам.

Пример дискриминантного анализа: диагностика острого аппендицита

Данные заимствованы из книги [23].

Многие люди либо на собственном опыте, либо на случаях с родными и близкими воочию убедились, как бывает важно и вместе с тем, непросто установить точный диагноз острого аппендицита. Это достаточно тонкая проблема, нередко требующая привлечения знаний и умений врача самой высокой квалификации.

Таблица 2.7. Симптомы острого аппендицита и их коды

	Симптомы острого аппендицита	Выраженность	Код
x1	Боли в правой подвздошной области	Незначительные	1
		Выраженные	2
x2	Продолжительность болей	Свыше 2 суток	1
		25—48 часов	2
		1—-24 часа	3
		До 12 часов	4
x3	Частота пульса	До 80 уд/мин	1
		21—100 уд/мин	2
		Свыше 100 уд/мин	3
x4	Лейкоциты крови	До 8 тыс.	1
		8—14 тыс.	2
		Свыше 14 тыс.	3
x5	Изменения языка	Не обложен	0
		Обложен	1
x6	Симптом Щеткина—Блюмберга	Отсутствует	0
		Выражен	2
x7	Симптом Ровзинга	Отсутствует	0
		Выражен	2
x8	Защитное мышечное напряжение	Отсутствует	0
		Выражено	2

В качестве исходной информации использовались данные клиники, в которых зарегистрированы результаты обследования 103 человек с установленным диагнозом: группа 1 — гангренозный аппендицит (28 наблюдений), группа 2 — флегмонозный аппендицит (25 наблюдений), группа 3 — катаральный аппендицит (26 наблюдений) и группа 4 — неподтвержденный диагноз (24 наблюдения). Исходными признаками служили 8 симптомов, охарактеризованных в таблице.

Таблица 2.8. Результаты клинического обследования

group	x1	x2	x3	x4	x5	x6	x7	x8
1	2	3	1	2	1	2	2	2
1	2	2	2	2	1	2	0	2
1	2	3	1	3	1	2	2	2
1	2	2	3	1	1	0	2	2
1	2	3	2	2	1	2	2	0
1	2	3	1	3	0	0	2	2
1	2	2	2	2	1	2	0	2
1	2	4	1	3	1	2	2	2
1	1	2	2	3	1	2	2	2
1	2	3	2	2	1	2	2	2
1	2	1	1	3	1	2	2	0

group	x1	x2	x3	x4	x5	x6	x7	x8
1	2	3	2	2	1	2	2	2
1	2	2	1	3	0	2	0	2
1	2	3	2	2	1	0	2	2
1	2	4	2	2	1	2	2	2
1	2	2	1	3	1	2	2	2
1	2	3	3	2	1	2	0	2
1	1	1	2	2	0	2	2	2
1	2	3	2	3	1	2	2	2
1	2	1	1	3	1	0	2	2
1	2	3	3	2	1	2	2	2
1	2	3	2	3	1	2	2	0
1	2	2	1	2	1	2	0	2
1	2	3	2	2	0	2	2	2
1	2	3	1	2	1	2	2	2
1	2	3	2	3	1	2	2	2
1	2	3	1	3	1	2	2	2
1	2	3	1	2	1	2	2	2
2	2	3	1	2	1	2	2	0
2	1	4	2	1	0	2	0	2
2	2	3	1	3	1	0	2	2
2	1	4	2	2	1	2	2	2
2	2	4	1	2	0	2	2	2
2	2	4	2	2	1	2	0	0
2	1	2	1	2	1	2	2	2
2	2	4	2	3	0	0	2	2
2	1	3	1	1	1	2	0	2
2	2	4	1	2	1	2	2	0
2	2	4	1	3	0	2	2	2
2	1	2	1	2	1	0	0	2
2	2	3	1	3	1	2	2	0
2	1	4	1	1	1	2	2	2
2	2	4	1	2	0	2	0	2
2	2	3	1	2	1	0	2	0
2	1	4	2	2	1	2	2	2
2	2	4	1	3	0	2	2	2
2	2	3	1	2	1	2	0	0
2	1	4	2	1	1	0	2	2
2	2	3	1	2	0	2	2	2

продолжение ⇩

Таблица 2.8 *(продолжение)*

group	x1	x2	x3	x4	x5	x6	x7	x8
2	2	4	1	2	1	2	2	2
2	2	4	2	2	1	2	2	2
2	2	4	2	3	1	0	2	2
2	1	3	2	2	1	2	2	2
3	1	3	1	2	1	0	2	2
3	2	4	1	1	0	2	0	0
3	2	3	1	2	1	0	2	2
3	2	4	2	2	1	2	0	0
3	1	2	1	1	0	0	2	2
3	2	3	1	3	1	2	2	0
3	2	4	1	2	1	2	2	2
3	2	1	1	1	1	2	2	0
3	1	4	1	2	0	0	0	2
3	2	1	2	2	1	2	2	0
3	2	3	1	1	1	2	0	2
3	2	4	1	2	1	0	0	0
3	1	3	1	1	0	2	2	0
3	2	4	1	2	1	0	2	2
3	2	3	2	2	1	2	2	2
3	1	4	1	1	0	0	2	0
3	2	3	1	2	1	2	2	0
3	2	4	2	2	1	2	0	2
3	2	3	1	3	0	0	2	2
3	2	4	1	2	1	0	0	0
3	1	3	1	1	1	2	2	0
3	2	3	1	2	0	2	2	2
3	2	4	1	2	1	2	2	2
3	1	4	2	1	1	2	2	0
3	2	3	1	2	1	2	2	0
3	1	4	1	2	1	2	2	0
4	1	2	1	1	0	0	0	0
4	1	1	2	1	0	0	0	0
4	1	3	1	1	1	0	0	0
4	2	1	1	2	0	0	0	0
4	1	2	1	1	0	0	0	0
4	1	1	1	1	0	0	0	0
4	1	2	1	1	0	0	0	0
4	1	1	2	1	1	0	0	0
4	1	2	1	2	0	0	0	0
4	2	1	1	1	0	0	0	0

group	x1	x2	x3	x4	x5	x6	x7	x8
4	1	2	1	2	1	0	0	0
4	1	2	1	2	1	0	0	0
4	1	1	1	2	0	0	0	0
4	1	1	2	1	0	0	2	0
4	1	4	1	1	0	0	0	0
4	1	3	1	1	0	0	0	0
4	2	1	1	2	1	0	0	0
4	1	4	1	1	0	0	0	0
4	1	2	1	1	0	0	0	0
4	1	1	1	2	1	0	0	0
4	1	2	1	1	0	0	0	0
4	1	1	2	1	0	0	0	0
4	2	1	1	1	0	0	2	0
4	1	2	1	1	0	0	0	0

Вводим представленные данные в электронную таблицу STATGRAPHICS. Сохраняем их в файле под именем appendix.

Для проведения дискриминантного анализа выбираем Special ▸ Multivariate Methods ▸ Discriminant Analysis. Получаем окно диалога дискриминантного анализа и вводим в поле Classification Factor (классифицирующий фактор) переменную с именем group, в поле Data (данные) — переменные x1, x2, x3, x4, x5 x6, x7 и x8 (рис. 2.23).

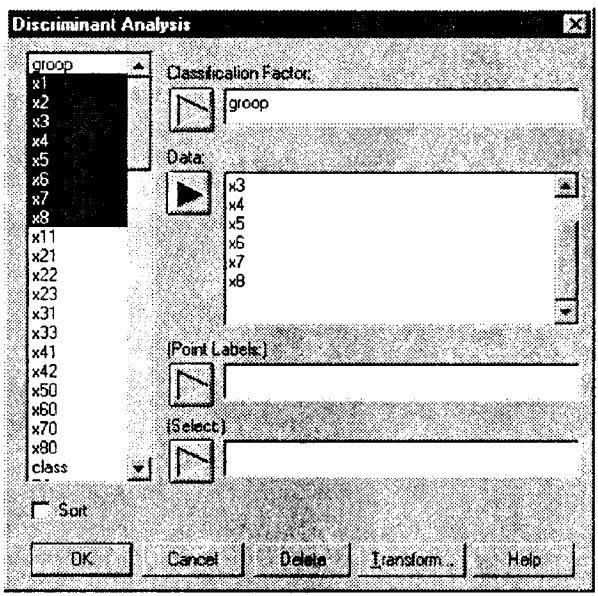


Рис. 2.23. Окно диалога дискриминантного анализа

Нажимаем OK. На экран выдается сводка дискриминантного анализа (рис. 2.24).

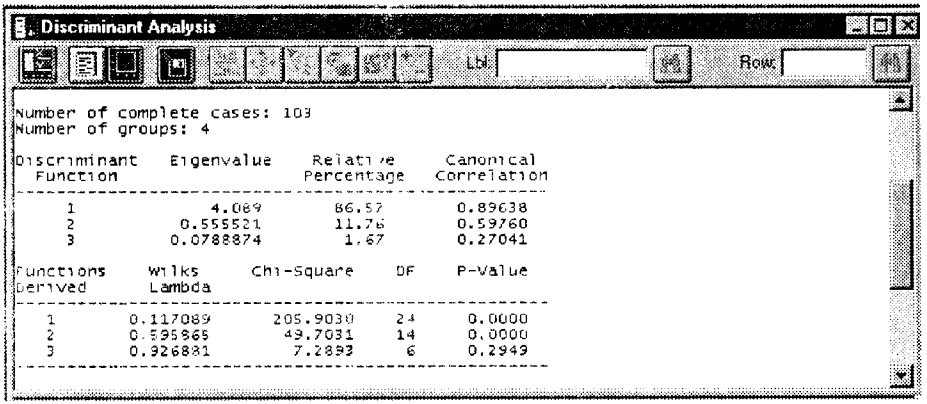


Рис. 2.24. Сводка дискриминантного анализа

Таблица содержит характеристики трех выделенных дискриминантных функций (Discriminant Function): собственные значения (Eigenvalue), вклад каждой функции в объяснение дисперсии симптомов (Relative Percentage) в процентах, канонические корреляции с классифицирующим фактором (Canonical Correlation) и оценки уровня значимости дискриминантных функций по критериям лямбда и Хи-квадрат. Как следует из приведенных цифр, для решения диагностической задачи достаточно применить две дискриминантные функции F1 и F2, на которые в сумме приходится 98,33 % дисперсии симптомов.

Нажмем кнопку табличных опций (вторая слева вверху) и установим флажок Discriminant Functions. Нажмем OK. Получаем таблицы, показанные на рис. 2.25.

Первая таблица содержит коэффициенты трех дискриминантных функций в стандартизированном виде. Для расчета по этим функциям в них следует подставлять стандартизированные значения исходных признаков. Вторая таблица включает константы и коэффициенты дискриминантных функций F1 и F2, в которые вводятся натуральные значения признаков:

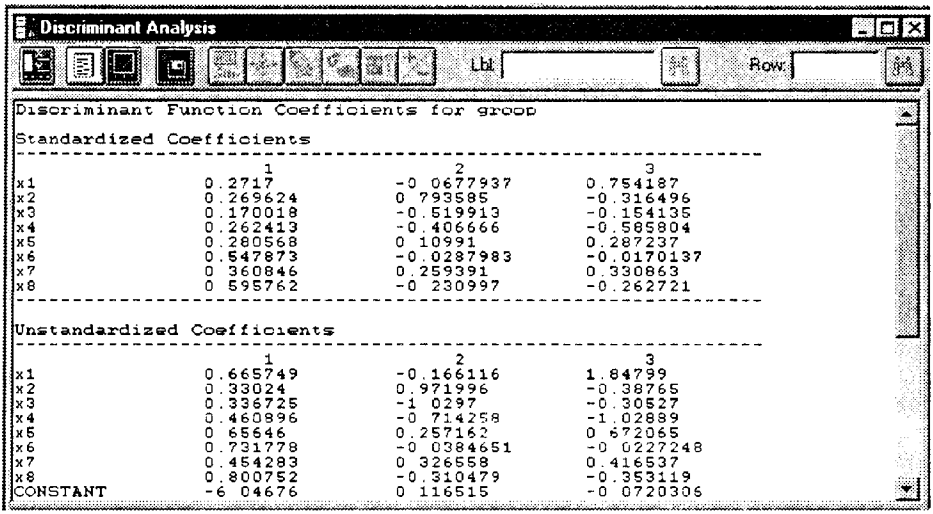
$$F1 = -6{,}05 + 0{,}67 \times x1 + 0{,}33 \times x2 + 0{,}34 \times x3 + 0{,}46 \times x4 + 0{,}66 \times x5 + 0{,}73 \times x6 + 0{,}45 \times x7 + 0{,}8 \times x8;$$

$$F2 = 0{,}12 - 0{,}17 \times x1 + 0{,}97 \times x2 - 1{,}03 \times x3 - 0{,}71 \times x4 + 0{,}26 \times x5 - 0{,}04 \times x6 + 0{,}33 \times x7 - 0{,}31 \times x8.$$

Вызовем еще раз окно табличных опций и попросим выдать на экран результаты расчета координат центроидов групп (рис. 2.26), а также групповых статистик. Результаты отображены на нижеследующих рисунках.

По данным таблицы на рисунке 2.26 можно уяснить, каковы средние значения симптомов в каждой группе больных и какова их вариация относительно средних. Видно, что по отдельно взятым разрозненным симптомам невозможно добиться постановки удовлетворительного диагноза. Здесь налицо многомерная диагностическая задача, когда только совокупное взаимодействие признаков

способно в той или иной степени отражать разбиение объектов на классы по актуальному критерию.



Discriminant Function Coefficients for groop

Standardized Coefficients

	1	2	3
x1	0.2717	-0 0677937	0.754187
x2	0.269624	0 793585	-0.316496
x3	0.170018	-0.519913	-0.154135
x4	0.262413	-0.406666	-0.585804
x5	0.280568	0 10991	0.287237
x6	0.547873	-0.0287983	-0.0170137
x7	0 360846	0.259391	0.330863
x8	0 595762	-0 230997	-0.262721

Unstandardized Coefficients

	1	2	3
x1	0.665749	-0.166116	1.84799
x2	0.33024	0.971996	-0.38765
x3	0.336725	-1 0297	-0.30527
x4	0.460896	-0 714258	-1.02889
x5	0 65646	0.257162	0 672065
x6	0.731778	-0 0384651	-0 0227248
x7	0.454283	0 326558	0.416537
x8	0.800752	-0.310479	-0.353119
CONSTANT	-6 04676	0 116515	-0 0720306

Рис. 2.25. Коэффициенты дискриминантных функций

Group Centroids for groop

	1	2	3
1	1 75963	-0 970723	0 097729
2	1.11673	0.530585	-0.415371
3	0 2162	0.849969	0 347939
4	-3.45038	-0 340983	-0 0582725

Рис. 2.26. Значения групповых центроидов

Для графического отображения результатов нажмем кнопку графических опций (третья слева вверху) и в предоставленном окне диалога закажем график дискриминантных функций (Discriminant Functions). Получаем рис. 2.27.

groop	1	2	3	4
COUNTS	28	25	26	24
MEANS				
x1	1.92857	1 64	1 69231	1 16667
x2	2 60714	3.52	3 26923	1.79167
x3	1.67857	1.36	1 19231	1.16667
x4	2.39286	2.08	1.76923	1.29167
x5	0.857143	0 72	0 730769	0 25
x6	1 71429	1 52	1 30769	0 0
x7	1 64286	1.52	1 46154	0.166667
x8	1.78571	1.52	0 923077	0.0
STD. DEVIATIONS				
x1	0.262265	0 489898	0 470679	0 380693
x2	0.785955	0.653197	0 874423	0.931533
x3	0.669636	0.489898	0 401918	0.380693
x4	0 566947	0.640312	0 58704	0 464306
x5	0.356348	0 458258	0 452344	0 442326
x6	0.712697	0.87178	0.970329	0 0
x7	0.780042	0.87178	0.904689	0.56466
x8	0 629941	0 87178	1 01678	0 0

Рис. 2.27. Значения групповых статистик

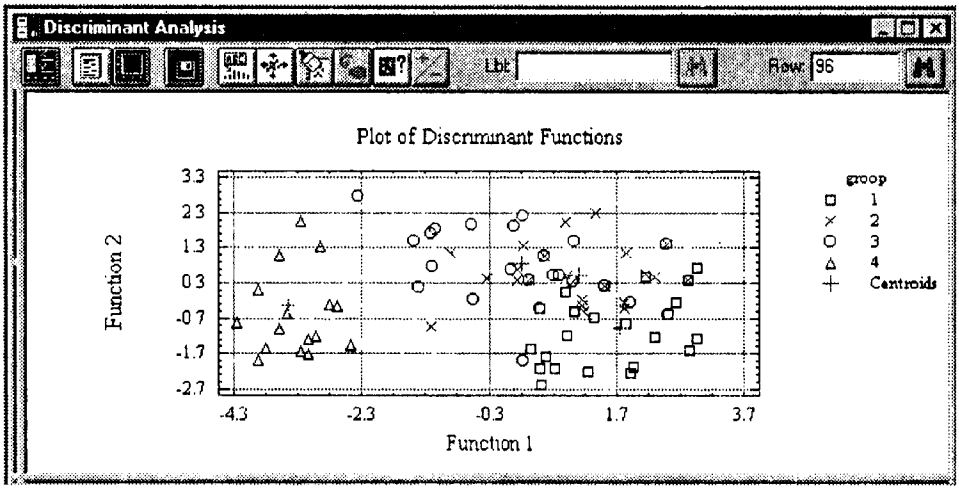


Рис. 2.28. Диаграмма рассеивания объектов на плоскости первых двух дискриминантных функций

На диаграмме рассеивания (рис. 2.28) хорошо видно, что объекты 4-го класса (неподтвержденный диагноз) образуют самостоятельную, четко выраженную группировку, не пересекающуюся с другими классами. В то же время, остальные классы имеют значительные пересечения в пространстве дискриминантных функций. В принципе, уже по этой картине можно составить диагностическое правило. Оно будет заключаться в вычислении расстояния от диагностируемого объекта до центроидов классов в пространстве канонических дискриминантных функций. Вместе с тем, более точные результаты диагностики дает применение линейных дискриминантных функций Фишера, параметры которых также определяются в рамках процедуры дискриминантного анализа STATGRAPHICS. Они здесь имеют название классифицирующих функций (Classification Functions) и были охарактеризованы выше.

Доступ к классифицирующим функциям осуществляется через окно диалога задания табличных опций. Ниже приводятся полученные параметры этих функций (рис. 2.29).

Classification Function Coefficients for groop

	1	2	3	4
X1	9.80432	8.17871	8.93672	5.94286
X2	3.75737	5.20323	4.92038	2.7094
X3	8.52742	6.92167	6.05655	6.17226
X4	6.57962	5.73891	4.31038	3.88906
X5	3.35299	2.97219	2.97616	-0.0100725
X6	3.68837	3.17182	2.4832	-0.14488
X7	2.39136	2.37583	2.38898	0.165203
X8	4.11553	3.31578	2.22598	-0.196831
CONSTANT	-41.0048	-35.7319	-29.899	-13.4048

Рис. 2.29. Коэффициенты классифицирующих функций

Для количественного выражения результатов применения классифицирующих функций обратимся снова к окну диалога задания табличных опций (нажав вторую слева кнопку) и установим флажок Classification Table. Нажимаем OK. Получаем две таблицы (рис. 2.30)

Classification Table

Actual groop	Group Size	Predicted groop 1	2	3	4
1	28	22	5	1	0
		(78.57%)	(17.86%)	(3.57%)	(0.00%)
2	25	1	16	7	1
		(4.00%)	(64.00%)	(28.00%)	(4.00%)
3	26	3	6	17	0
		(11.54%)	(23.08%)	(65.38%)	(0.00%)
4	24	0	0	0	24
		(0.00%)	(0.00%)	(0.00%)	(100.00%)

Percent of cases correctly classified: 76.70%

Рис. 2.30. Сводные результаты классификации

Row	Actual Group	Highest Prob. Group	Highest Value	2nd Highest Prob. Group	2nd Highest Value
1	1	*2	35.3337	1	35.3061
2	1	1	35.2934	2	32.3005
3	1	1	41.8857	2	41.0726
4	1	1	34.6472	2	31.8913
5	1	*3	36.19	2	35.6238
6	1	*2	31.7568	1	31.156
7	1	1	35.2934	2	32.3005
8	1	*2	46.2759	1	45.6431
9	1	1	36.8515	2	34.6124
10	1	1	43.8335	2	42.2554
11	1	1	26.1399	2	24.0346
12	1	1	43.8335	2	42.2554
13	1	1	29.9926	2	28.1456
14	1	1	36.4568	2	35.9118
15	1	1	47.5909	2	47.4586
16	1	1	38.1283	2	35.8694
17	1	1	47.5782	2	44.4254

Рис. 2.31. Детальный разбор результатов применения классифицирующих функций

Из верхней таблицы черпаем сведения об итоговых результатах диагностики острого аппендицита. Точность диагностики больных первой группы (гангренозный аппендицит) составляет 78,57 %, второй группы (флегмонозный аппендицит) — 64 % и третьей группы (катаральный аппендицит) — 65,38 %. Это не слишком точные результаты, которые, однако, могут в какой-то мере содействовать при вынесении клиническим специалистом окончательного заключения. Вместе с тем, констатация отсутствия острого аппендицита (группа 4 — неподтвержденный диагноз) осуществляется со 100-процентный надежностью, что следует считать определенным достижением в применении методов дискриминантного анализа для решения практически важных задач медицинской диагностики.

Во второй таблице (рис. 2.31) дается детальный разбор результатов диагностики посредством полученных классифицирующих функций. Для каждого объекта приведены значения двух наибольших дискриминантных функций и результат

отнесения к тому или иному классу. Неправильно классифицированные объекты помечены звездочкой. Это дает пищу для дополнительных размышлений о причине неудачных автоматических диагнозов.

Методы сравнения с образцом

В методах данной группы объекты рассматриваются как прецеденты и используется только одна операция — определение сходства (различия) этих прецедентов с неизвестным объектом. Сходство (различие) выражается геометрически через расстояние в p-мерном пространстве признаков. В зависимости от условий конкретной задачи роль отдельного прецедента может меняться в широких пределах от главной до весьма косвенного участия. Этим объясняется дальнейшее разделение данной группы методов на подклассы.

Метод сравнения с прототипом

Это самый простой метод (рис. 2.32). Он применяется тогда, когда классы объектов ω_i отображаются в пространстве признаков компактными геометрическими группировками. В таком случае обычно в качестве точки-прототипа выбирается центр геометрической группировки класса (или ближайший к центру объект), определяемый как

$$\mathbf{z}_i = (\mathbf{x}_1 + \mathbf{x}_2 + \ldots + \mathbf{x}_{N_i})/N_i$$

где N_i — количество объектов в классе ω_i.

Для классификации неизвестного объекта $\mathbf{x}$ находится ближайший к нему прототип, и объект относится к тому же классу, что и этот прототип.

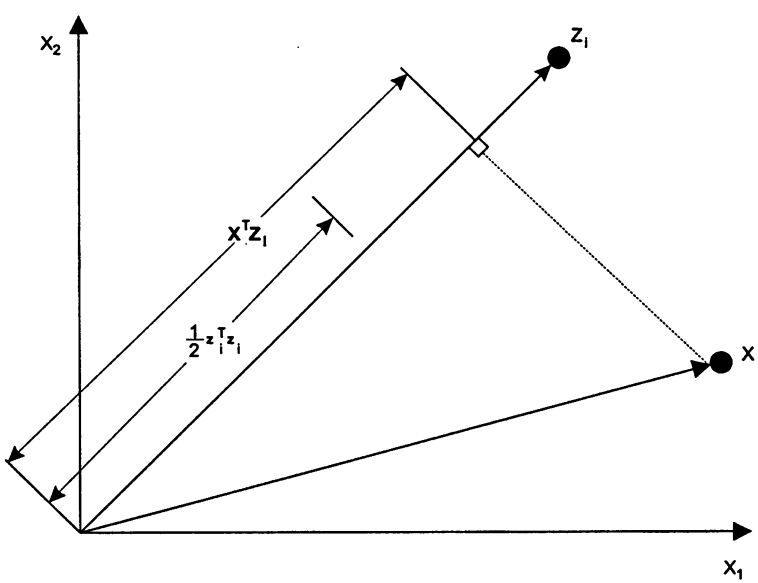


Рис. 2.32. Иллюстрация метода сравнения с прототипом

В качестве меры близости могут применяться различные меры расстояний. Например, для определения расстояния между объектом $\mathbf{x}$ и прототипом i-го класса $\mathbf{z}_i$ используют квадрат евклидова расстояния

$$d^2(\mathbf{x}, \mathbf{z}_i) = (\mathbf{x}-\mathbf{z}_i)^{\mathrm{T}}(\mathbf{x}-\mathbf{z}_i) = \mathbf{x}^{\mathrm{T}}\mathbf{x} - 2\mathbf{x}^{\mathrm{T}}\mathbf{z}_i + \mathbf{z}_i^{\mathrm{T}}\mathbf{z}_i .$$

Так как $\mathbf{x}^{\mathrm{T}}\mathbf{x}$ не зависит от класса, то этот член можно из приведенного выражения устранить. Тем самым, умножив оставшуюся часть на $-1/2$, получим правило классификации, эквивалентное линейной решающей функции (рис. 2.32):

$$y_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{z}_i - \frac{1}{2}\mathbf{z}_i^{\mathrm{T}}\mathbf{z}_i .$$

Указанный факт следует особо отметить. Он наглядно демонстрирует связь прототипной и признаковой репрезентации знаний о структуре данных. Пользуясь приведенным представлением, можно любую линейную решающую функцию (линейную модель) рассматривать как гипотетический прототип. В свою очередь, если анализ пространственной структуры классов позволяет сделать вывод об их геометрической компактности, то каждый из этих классов достаточно заменить одним прототипом, который эквивалентен линейной модели классификации объектов.

На практике, конечно, ситуация часто бывает отличной от описанного идеализированного примера. Перед аналитиком, намеревающимся применить метод классификации объектов БД, основанный на сравнении с прототипами классов, встают непростые проблемы. Это, в первую очередь, выбор меры близости (метрики), от которого может существенно измениться пространственная конфигурация распределения объектов. И во-вторых, самостоятельной проблемой является анализ многомерных структур данных. Обе проблемы особенно остро заявляют о себе в условиях высокой размерности и неоднородности пространства признаков.

Метод k-ближайших соседей

Как уже отмечалось ранее, метод k-ближайших соседей для решения задач дискриминантного анализа был впервые предложен в [2]. Он заключается в следующем.

При классификации неизвестного объекта $\mathbf{x}$ находится заданное количество k геометрически ближайших к нему объектов (ближайших соседей) с известной классификацией. Решение об отнесении объекта $\mathbf{x}$ к тому или иному классу принимается путем анализа информации об этой известной принадлежности его ближайших соседей, например, с помощью простого подсчета голосов.

Первоначально метод k-ближайших соседей (k-БС) рассматривался как непараметрический метод оценивания отношения правдоподобия в окрестности $\mathbf{x}$. Для этого метода получены теоретические оценки его эффективности в сравнении с оптимальным байесовским классификатором. Так, для случая $k = 1$ в [1] была доказана следующая теорема.

Пусть P_N — вероятность сделать по правилу первого ближайшего соседа (1-БС) в выборке $\mathbf{X}$ объема N. Тогда при распознавании двух классов в предположении, что из $\mathbf{X}$ делаются независимые случайные выборки с возвращением,

$$P^* \le P_\infty \le 2P^*(1-P^*),$$

где $P_\infty = \lim_{N\to\infty} P_N$; P^* — риск ошибочной классификации любого случайным образом выбранного объекта при использовании неизвестного байесовского метода оптимальной классификации.

В работе [25] приведен аналогичный результат для K классов

$$P^* \le P_\infty \le P^*\left(2 - \frac{K}{K-1}P^*\right).$$

Приведенные выражения показывают, что асимптотические вероятности ошибки для правила 1-БС превышают ошибки правила Байеса не более чем в два раза.

Для многих реальных задач искусственного интеллекта, когда объекты описываются большим количеством разнородных признаков (в том числе качественных и номинальных), удобно интерпретировать каждый объект обучающей выборки как отдельный линейный классификатор. Тогда тот или иной класс представляется не одним прототипом, а набором линейных классификаторов. Их совокупное взаимодействие дает в итоге кусочно-линейную поверхность, разделяющую классы в пространстве признаков. Вид такой поверхности, состоящей из кусков гиперплоскостей, может быть разнообразным и зависит от конфигурации классифицируемых совокупностей объектов (рис. 2.33).

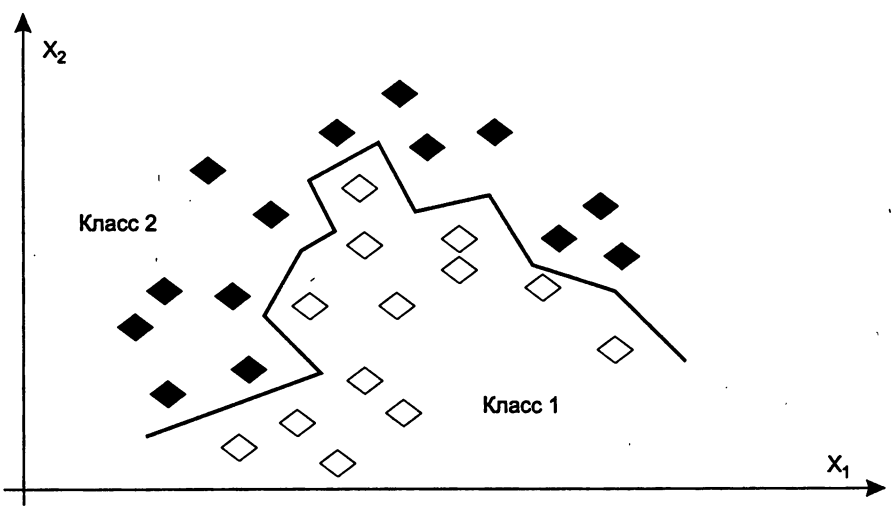


Рис. 2.33. Пример кусочно-линейной разделяющей границы для метода k-ближайших соседей

При использовании метода k-ближайших соседей для классификации объектов приходится решать сложную задачу выбора метрики. Эта проблема обостряется

в условиях высокой размерности данных вследствие достаточной трудоемкости метода. Поэтому здесь, как и в методе сравнения с прототипом, необходимо решать творческую задачу анализа многомерной структуры экспериментальных данных с целью минимизации числа объектов, представляющих свои классы.

Алгоритмы вычисления оценок

Принцип действия алгоритмов вычисления оценок (АВО) состоит в вычислении приоритетов (оценок сходства), характеризующих «близость» классифицируемого и эталонных объектов по системе ансамблей признаков, представляющей собой систему подмножеств заданного множества признаков [31].

В отличие от ранее рассмотренных методов алгоритмы вычисления оценок по-новому оперируют описаниями объектов. Для АВО объекты существуют одновременно в самых разных подпространствах пространства признаков.

Используемые подпространства (сочетания признаков) называют опорными множествами или множествами частичных описаний объектов. Объекты обучающей выборки в АВО называют эталонными. Сходство между классифицируемым и эталонными объектами определяется через так называемую обобщенную близость. Эта близость представляется комбинацией близостей, вычисленных на множестве частичных описаний.

Задача определения сходства и различия объектов в АВО формулируется как параметрическая. Выделен этап настройки АВО по обучающей выборке, на котором подбираются оптимальные значения введенных параметров. Критерием качества служит ошибка классификации, а параметризуется буквально все. Сюда относятся правила вычисления близости объектов по отдельным признакам, правила вычисления близости объектов в подпространствах признаков, степень важности того или иного эталонного объекта и значимость вклада каждого опорного множества признаков в итоговую оценку сходства классифицируемого объекта с каким-либо классом. Параметры АВО задаются в виде значений порогов и/или как веса указанных составляющих.

Теоретические возможности АВО не ниже возможностей любого другого алгоритма классификации, так как с помощью АВО могут быть реализованы все мыслимые операции с исследуемыми объектами. Но, как это обычно бывает, расширение потенциальных возможностей наталкивается на большие трудности их практического воплощения, особенно на этапе настройки АВО. Отдельные трудности отмечались при обсуждении метода k-ближайших соседей, который можно рассматривать как усеченный вариант АВО. Для алгоритмов вычисления оценок указанные трудности возрастают многократно.

Методы исследования структуры данных

Анализ многомерных данных без использования обучающей информации направлен на выяснение структуры взаимоотношений объектов и признаков ТЭД. В настоящее время накоплен обширный арсенал средств такого анализа. Наибо-

лее полное изложение применяемых здесь подходов, сопровождающееся подробными ссылками на ключевые работы, содержится в [14]. Классификация известных методов анализа структуры многомерных данных приведена в табл. 2.9.

Таблица 2.9. Классификация методов анализа структуры данных

Методы визуализации данных	Методы автоматического группирования
Линейные методы снижения размерности	Факторный анализ объектов и признаков
Нелинейные отображения	Кластерный анализ объектов и признаков
Многомерное шкалирование	Иерархическое группирование
Заполняющие пространство кривые	Определение «точек сгущения»

Разделение методов носит достаточно условный характер, так как различные методы имеют немало пересечений в отдельных приемах обработки информации. В основу приведенной классификации положен признак, отражающий степень участия экспериментатора в выделении особенностей взаимоотношений между исследуемыми объектами и признаками. Если в методах автоматического группирования это участие минимально, применение методов визуализации данных нацелено на поиск наиболее выразительных изображений совокупности исследуемых объектов для последующего максимального задействования потенциала зрительного анализатора экспериментатора. Рассмотрим указанные методы более подробно.

Методы визуализации данных

Основное назначение рассматриваемой группы методов — дать визуальное представление о структуре изучаемых данных. Визуализация данных предполагает получение тем или иным способом графического отображения совокупности объектов на числовую ось, на плоскость или в трехмерный объем, максимально отражающего особенности распределения этих объектов в многомерном пространстве.

Линейные методы снижения размерности

Линейные методы снижения размерности направлены на нахождение нового координатного пространства, в котором каждая координатная ось является линейной комбинацией исходных признаков. Популярность данного подхода объясняется тем, что линейные комбинации признаков хорошо интерпретируются — коэффициенты в уравнениях координатных осей трактуются, например, как веса или вклады признаков.

Всесторонне изученным является использование в качестве осей нового пространства первых главных компонент (ГК).

Метод главных компонент (МГК) был предложен Пирсоном в 1901 году и затем вновь открыт и детально разработан Хоттелингом в 1933 году. Ему посвящено

большое количество исследований, он широко представлен в литературных источниках. Обратим внимание на основные феномены МГК.

МГК осуществляет переход к новой системе координат $y_1, \ldots, y_p$ в исходном пространстве признаков $x_1, \ldots, x_p$, которая является системой ортонормированных линейных комбинаций [13, 14, 15]:

$$\begin{cases} y_j(\mathbf{x}) = w_{1j}(x_1 - m_1) + \ldots + w_{pj}(x_p - m_p); \\ \sum_{i=1}^{p} w_{ij}^2 = 1 \qquad (j = \overline{1, p}); \\ \sum_{i=1}^{p} w_{ij} w_{ik} = 0 \qquad (j, k = \overline{1, p}, j \neq k), \end{cases}$$

где m_i — математическое ожидание признака x_i.

Линейные комбинации выбираются таким образом, что среди всех возможных линейных нормированных комбинаций исходных признаков первая главная компонента $y_1(\mathbf{x})$ обладает наибольшей дисперсией. Геометрически это выглядит как ориентация новой координатной оси y_1 вдоль направления наибольшей вытянутости эллипсоида рассеивания объектов исследуемой выборки в пространстве признаков $x_1, \ldots, x_p$. Вторая главная компонента имеет наибольшую дисперсию среди всех оставшихся линейных преобразований, некоррелированных с первой главной компонентой. Она интерпретируется как направление наибольшей вытянутости эллипсоида рассеивания, перпендикулярное первой главной компоненте. Следующие главные компоненты определяются по аналогичной схеме.

Вычисление коэффициентов главных компонент w_{ij} основано на том факте, что векторы $\mathbf{w}_1=(w_{11}, \ldots, w_{p1})^T, \ldots, \mathbf{w}_p = (w_{1p}, \ldots, w_{pp})^T$ являются собственными (характеристическими) векторами корреляционной матрицы **S**. В свою очередь, соответствующие собственные числа этой матрицы равны дисперсиям проекций множества объектов на оси главных компонент.

Из ряда ценных свойств главных компонент с точки зрения визуализации многомерных данных выделяют свойства наименьшего искажения структуры исходных точек (объектов) при их проецировании в пространство меньшей размерности, «натянутое» на первые главные компоненты. Этими свойствами определяется полезность МГК при изучении структуры многомерных данных. Практически ни одно современное исследование такой структуры не обходится без того, чтобы не рассмотреть проекции объектов в пространстве, натянутом на первую, первые две и, реже, первые три главные компоненты. Нередко прибегают к анализу проекций объектов в пространстве, образованные комбинациями главных компонент более высокого порядка, например 3-й и 4-й ГК, 5-й и 6-й и т. п.

Пример применения метода главных компонент

Ниже рассматривается пример, относящийся к сравнительному оцениванию изделий, характеризующихся одновременно несколькими параметрами. Это — ав-

томобили. В таблице приводятся выборочные сведения о фирме-изготовителе автомобиля, названии модели, а также оценочные параметры — вес (переменная **weight**), число цилиндров (переменная **cylinders**), ускорение (переменная **accel**), объем двигателя (переменная **displace**) и мощность в лошадиных силах (переменная **horspower**).

Таблица 2.10. Исходные данные

Изготовитель	Модель	Вес	Количество цилиндров	Ускорение	Объем	Мощность
Volkswagen	Rabbit Dl	1985	4	21,5	90	48
Ford	Fiesta	1800	4	14,4	98	66
Mazda	GLC Deluxe	1985	4	19,4	78	52
Datsun	B210 GX	2070	4	18,6	85	70
Honda	Civic CVCC	1800	4	16,4	91	60
Oldsmobile	Cutlass	3365	8	15,5	260	110
Dodge	Diplomat	3735	8	13,2	318	140
Mercury	Monarch	3570	8	12,8	302	139
Pontiac	Phoenix	3535	6	19,2	231	105
Chevrolet	Malibu	3155	6	18,2	200	95
Ford	Fairmont A	2965	6	15,8	200	85
Ford	Fairmont M	2720	4	15,4	140	88
Plymouth	Volare	3430	6	17,2	225	100
AMC	Concord	3210	6	17,2	232	90
Buick	Century	3380	6	15,8	231	105
Mercury	Zephyr	3070	6	16,7	200	85
Dodge	Aspen	3620	6	18,7	225	110
AMC	Concord D1	3410	6	15,1	258	120
Chevrolet	MonteCarlo	3425	8	13,2	305	145
Buick	RegalTurbo	3445	6	13,4	231	165
Ford	Futura	3205	8	11,2	302	139
Dodge	Magnum XE	4080	8	13,7	318	140
Chevrolet	Chevette	2155	4	16,5	98	68
Toyota	Corona	2560	4	14,2	134	95
Datsun	510	2300	4	14,7	119	97
Dodge	Omni	2230	4	14,5	105	75
Toyota	Celica GT	2515	4	14,8	134	95
Plymouth	Sapporo	2745	4	16,7	156	105
Oldsmobile	Starfire	2855	4	17,6	151	85
Datsun	200-SX	2405	4	14,9	119	97
Audi	5000	2830	5	15,9	131	103
Volvo	264GL	3140	6	13,6	163	125
Saab	99GLE	2795	4	15,7	121	115
Peugeot	604SL	3410	6	15,8	163	133
Volkswagen	Scirocco	1990	4	14,9	89	71
Honda	Accord LX	2135	4	16,6	98	68

Изготовитель	Модель	Вес	Количество цилиндров	Ускорение	Объем	Мощность
Pontiac	Lemans V6	3245	6	15,4	231	115
Mercury	Zephyr 6	2990	6	18,2	200	85
Ford	Fairmont 4	2890	4	17,3	140	88
AMC	ConcordDL6	3265	6	18,2	232	90
Dodge	Aspen 6	3360	6	16,6	225	110
Chevrolet	Caprice Cl	3840	8	15,4	305	130
Ford	LTD Landau	3725	8	13,4	302	129
Mercury	GrandMarqs	3955	8	13,2	351	138
Dodge	St. Regis	3830	8	15,2	318	135
Buick	Estate SW	4360	8	14,9	350	155
Ford	Country SW	4054	8	14,3	351	142
Chevrolet	Malibu SW	3605	8	15	267	125
Chrysler	Lebaron SW	3940	8	13	360	150
Volkswagen	Rabbit Cus	1925	4	14	89	71
Mazda	GLC Deluxe	1975	4	15,2	86	65
Dodge	Colt Hatch	1915	4	14,4	98	80
AMC	Spirit DL	2670	4	15	121	80
Mercedes	300D	3530	5	20,1	183	77
Cadillac	Eldorado	3900	8	17,4	350	125
Peugeot	504	3190	4	24,8	141	71
Oldsmobile	Cutlass	3420	8	22,2	260	90
Plymouth	Horizon	2200	4	13,2	105	70
Plymouth	HorizonTC3	2150	4	14,9	105	70
Datsun	210	2020	4	19,2	85	65
Fiat	Strada Cus	2130	4	14,7	91	69
Buick	SkylarkLim	2670	4	16	151	90
Chevrolet	Citation	2595	6	11,3	173	115
Oldsmobile	Omega	2700	6	12,9	173	115
Pontiac	Phoenix	2556	4	13,2	151	90
Volkswagen	Rabbit	2144	4	14,7	98	76
Toyota	CorollaTer	1968	4	18,8	89	60
Chevrolet	Chevette	2120	4	15,5	98	70
Datsun	310	2019	4	16,4	86	65
Chevrolet	Citation	2678	4	16,5	151	90
Ford	Fairmont	2870	4	18,1	140	88
AMC	Concord	3003	4	20,1	151	90
Dodge	Aspen	3381	6	18,7	225	90
Audi	4000	2188	4	15,8	97	78
Toyota	Corona LB	2711	4	15,5	134	90
Mazda	626	2542	4	17,5	120	75
Datsun	510 Hatch	2434	4	15	119	92

продолжение

Таблица 2.10. *(продолжение)*

Изготовитель	Модель	Вес	Количество цилиндров	Ускорение	Объем	Мощность
Toyota	Corolla	2265	4	15,2	108	75
Mazda	GLC	2110	4	17,9	86	65
Dodge	Colt	2800	4	14,4	156	105
Datsun	210	2110	4	19,2	85	65
Volkswagen	Rabbit Dl	2085	4	21,7	90	48
Volkswagen	Dasher Dl	2335	4	23,7	90	48
Audi	5000S Dl	2950	5	19,9	121	67
Mercedes	240D	3250	4	21,8	146	67
Honda	Civic1500G	1850	4	13,8	91	67
Renault	LeCar Delx	1835	4	17,3	85	67
Subaru	DL	2145	4	18	97	62
Volkswagen	Rabbit	1845	4	15,3	89	132
Datsun	280-ZX	2910	6	11,4	168	100
Mazda	RX-7 GS	2420	3	12,5	70	88
Triumph	TR7 Coupe	2500	4	15,1	122	72
Ford	Must Cobra	2905	4	14,3	140	84
Honda	Accord	2290	4	17	107	84
Plymouth	Reliant	2490	4	15,7	135	92
Buick	Skylark	2635	4	16,4	151	110
Dodge	Aries SW	2620	4	14,4	156	84
Chevrolet	Citation	2725	6	12,6	173	58
Plymouth	Reliant	2385	4	12,9	135	64
Toyota	Starlet	1755	4	16,9	79	60
Plymouth	Champ	1875	4	16,4	86	67
Honda	Civic1300	1760	4	16,1	81	65
Subaru	210	2065	4	17,8	97	62
Datsun	Tercel	1975	4	19,4	85	68
Toyota	GLC 4	2050	4	17,3	89	63
Mazda	Horizon 4	1985	4	16	91	65
Plymouth	Escort 4W	2215	4	14,9	105	65
Ford	Escort 2H	2045	4	16,2	98	74
Ford	Jetta	2380	4	20,7	98	75
Volkswagen	18I	2190	4	14,2	105	75
Renault	Prelude	2320	4	15,8	100	100
Honda	Corolla	2210	4	14,4	107	74
Toyota	200SX	2350	4	16,8	108	80
Datsun	626	2615	4	14,8	119	110
Mazda	505S Dl	2635	4	18,3	120	76
Peugeot	900S	3230	4	20,4	141	116
Saab	Diesel	2800	4	15,4	121	120
Volvo	Cressida	3160	6	19,6	145	110

Изготовитель	Модель	Вес	Количество цилиндров	Ускорение	Объем	Мощность
Toyota	810 Maxima	2900	6	12,6	168	105
Datsun	Century	2930	6	13,8	146	88
Buick	Cutlass LS	3415	6	15,8	231	85
Oldsmobile	Granada GL	3725	8	19	350	88
Ford	Lebaron	3060	6	17,1	200	88
Chrysler	Cavalier	3465	6	16,6	225	88
Chevrolet	CavalierSW	2605	4	19,6	112	85
Chevrolet	Cavalier2D	2640	4	18,6	112	84
Chevrolet	1200 Hatch	2395	4	18	112	90
Pontiac	Aries SE	2575	4	16,2	112	92
Dodge	Phoenix	2525	4	16	135	74
Pontiac	Fairmont	2735	4	18	151	68
Ford	Concord Dl	2865	4	16,4	140	68
AMC	Rabbit L	3035	4	20,5	151	63
Volkswagen	GLC Cust l	1980	4	15,3	105	70
Mazda	GLC Custom	2025	4	18,2	91	88
Mazda	Horizon	1970	4	17,6	91	75
Plymouth	Lynx l	2125	4	14,7	105	70
Mercury	Stanza XE	2125	4	17,3	98	67
Nissan	Accord	2160	4	14,5	120	67
Honda	Corolla	2205	4	14,5	107	67
Toyota	Civic M	2245	4	16,9	108	110
Honda	Civic A	1965	4	15	91	85
Honda	310 GX	1965	4	15,7	91	92
Datsun	CenturyLmt	1995	4	16,2	91	112
Buick	Cutlass Dl	2945	6	16,4	181	96
Oldsmobile	Lebaron	3015	6	17	262	84
Chrysler	Granada l	2585	4	14,5	156	90
Ford	Celica GT	2835	6	14,7	232	86
Toyota	Charger2.2	2665	4	13,9	144	52
Dodge	Camaro	2370	4	13	135	84
Chevrolet	MustangGL	2950	4	17,3	151	79
Ford	Pickup	2790	4	15,6	140	82
Volkswagen	Rampage	2130	4	24,6	97	
Dodge	Ranger	2295	4	11,6	135	
Ford	S-10	2625	4	18,6	120	
Chevrolet		2720	4	19,4	119	

Введем эти данные в электронную таблицу STATGRAPHICS (в ней присутствуют также другие дополнительные параметры). Назовем файл данных cardata. Выберем Special ▸ Multivariate Methods ▸ Principal Components. Появляется окно диалога для задания анализируемых переменных (рис. 2.34).

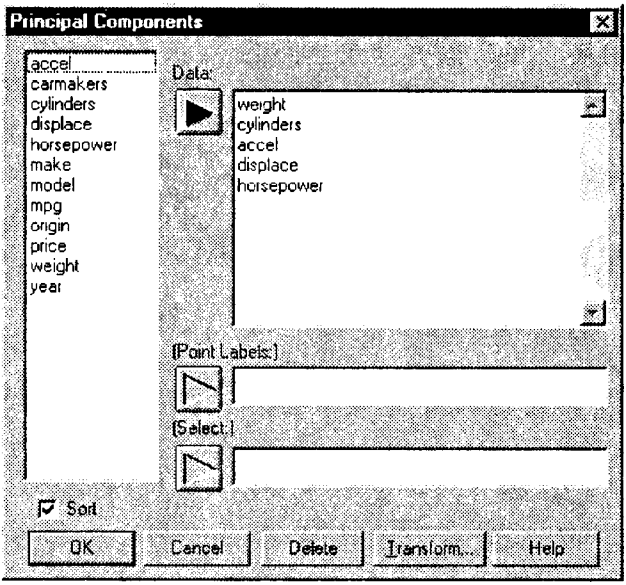


Рис. 2.34. Окно задания переменных для анализа по методу главных компонент

Нажимаем OK. Получаем исходную сводку анализа МГК (рис. 2.35).

Из полученной сводки заключаем, что анализу подвергаются переменные **weight**, **cylinders**, **accel**, **displace** и **horspower** и что число объектов составляет 151. Далее следует информация непосредственно МГК: собственные значения главных компонент, упорядоченные по величине (Eigenvalue); процент дисперсии, приходящийся на каждую выделенную главную компоненту (Percent of Variance); накопленный процент дисперсии (Cumulative Percentage).

Приведенные цифры говорят о том, что уже первые две главные компоненты описывают 93,4 % дисперсии исходных данных. Третья главная компонента добавляет еще приблизительно 4,2 % дисперсии, так что в сумме получается 97,6 % дисперсии.

Для более детального анализа нажмем кнопку табличных опций (вторая слева в верхнем ряду) и в соответствующем окне диалога (рис. 2.36) установим флажок компонентных весов (Component Weights). Получим следующую таблицу (рис. 2.37).

Как следует из полученных цифр, в первой главной компоненте примерно одинаковые по величине положительные коэффициенты имеют вес, количество цилиндров, объем двигателя и мощность в лошадиных силах. Вместе с тем, во второй главной компоненте превалирует только одна величина: ускорение. А в третьей главной компоненте наблюдается сочетание веса машины и ее мощности (с положительным знаком), которому противопоставляется количество цилиндров (с отрицательным знаком). Не углубляясь в интерпретацию полученных главных компонент, которая, конечно, может представлять интерес для

специалистов, перейдем к рассмотрению диаграммы рассеивания всей совокупности автомашин в пространстве выделенных трех первых главных компонент. Для этого щелкнем левой кнопкой мыши на кнопке графических опций и инициализируем данное трехмерное отображение (рис. 2.39).

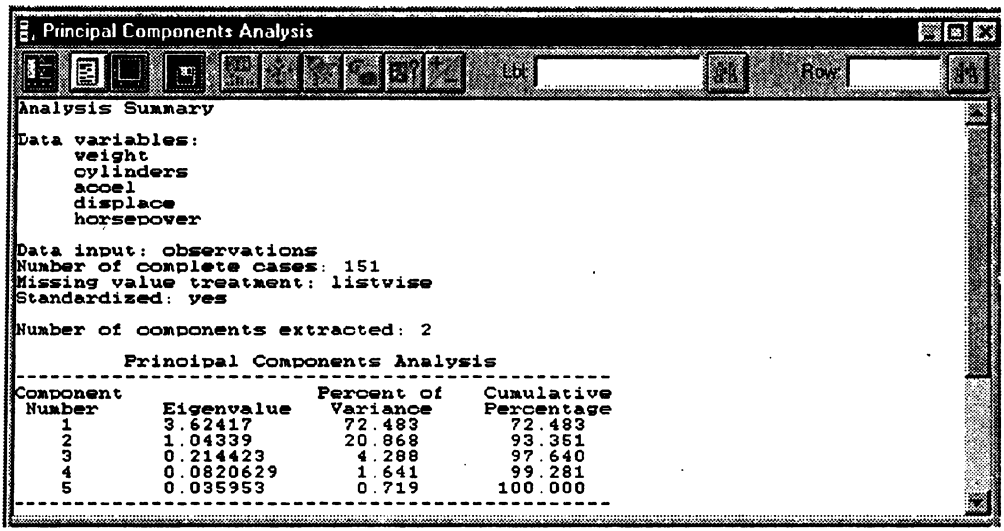


Рис. 2.35. Исходная сводка МГК

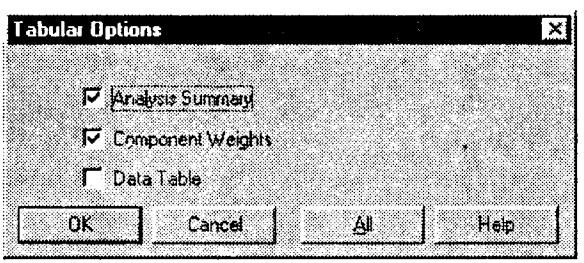


Рис. 2.36. Окно диалога табличных опций МГК

	Component 1	Component 2	Component 3
weight	0.484397	0.281143	0.426531
cylinders	0.489981	0.125914	-0.665775
accel	-0.178778	0.91435	0.130289
displace	0.507767	0.142972	-0.241578
horsepower	0.485273	-0.220516	0.547248

Рис. 2.37. Веса признаков в главных компонентах

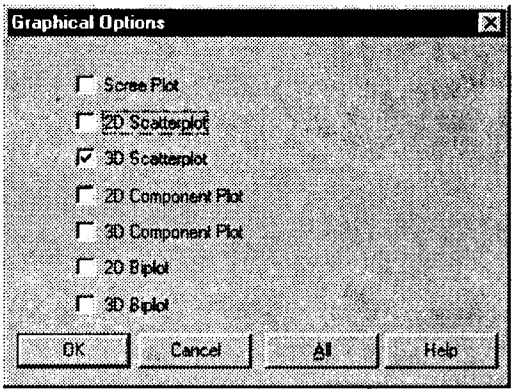


Рис. 2.38. Графические опции метода главных компонент

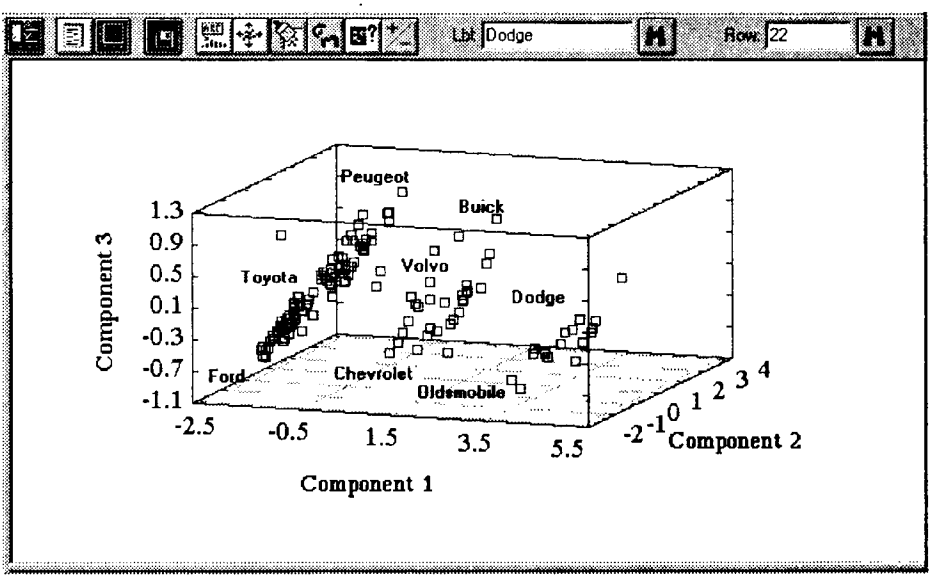


Рис. 2.39. Проекция исследуемых автомобилей в пространство первых трех ГК

На представленном рисунке хорошо видно, что вся исследуемая совокупность автомашин разделилась на три достаточно четко выраженные группы. Для большей выразительности на рисунке даны названия некоторых фирм, производящих автомобили, которые выдаются в специальных окнах STATGRAPHICS после нажатия пятой справа кнопки в верхнем ряду и маркировки интересующей точки.

Для первой, наиболее многочисленной группировки характерны сравнительно небольшие вес, количество цилиндров, мощность и объем двигателя (первая слева группа). Вместе с тем, большая доля автомашин этой группы обладают хорошим ускорением (высокие значения 2-й ГК) и высоким соотношением веса и мощности к количеству цилиндров (3-я ГК).

Вторая группа не столь многочисленна, но ей также свойственны указанные характеристики, хотя и менее ярко выраженные.

И наконец, третья группа автомашин (сравнительно малочисленная) имеет большие вес, мощность, количество, цилиндров. В то же время, показатели ускорения и соотношение веса и мощности к количеству цилиндров здесь (если говорить в целом) гораздо меньшие.

Таким образом, произведенный с помощью метода главных компонент анализ данных позволяет получить более «объемное» видение современного автомобильного рынка, что может способствовать лучшей ориентации как потребителей этой продукции, так и производителей с позиций оценки существующих тенденций.

Факторный анализ. В отличие от метода главных компонент факторный анализ основан не на дисперсионном критерии автоинформативности системы признаков, а ориентирован на объяснение имеющихся между признаками корреляций. Основная модель факторного анализа записывается следующей системой равенств [41]:

$$x_i = \sum_{j=1}^{m} l_{ij} f_j + \varepsilon_i; i = \overline{1,p}; m < p.$$

То есть полагается, что значения каждого признака x_i могут быть выражены взвешенной суммой латентных переменных (простых факторов) f_j, количество которых меньше числа исходных признаков, и остаточным членом ε_i с дисперсией $\sigma^2(\varepsilon_i)$, действующей только на x_i, который называют специфическим фактором.

Коэффициенты l_{ij} называются нагрузкой i-й переменной на j-й фактор, или нагрузкой j-го фактора на i-ю переменную. В самой простой модели факторного анализа считается, что факторы f_j взаимно независимы и их дисперсии равны единице, а случайные величины ε_i тоже независимы друг от друга и от какого-либо фактора f_j. Максимально возможное количество факторов m при заданном числе признаков p определяется неравенством

$$(p + m) < (p - m)^2,$$

которое должно выполняться, чтобы задача не вырождалась в тривиальную. Данное неравенство получается на основании подсчета степеней свободы, имеющихся в задаче [37]. Сумму квадратов нагрузок называют общностью соответствующего признака x_i, и чем больше это значение, тем лучше описывается признак x_i выделенными факторами f_j. Общность есть часть дисперсии признака, которую объясняют факторы. В свою очередь, она показывает, какая часть дисперсии исходного признака остается необъясненной при используемом наборе факторов, и данную величину называют специфичностью признака. Таким образом,

$$\text{дисперсия признака} = \text{общность}\left(\sum_{j=1}^{m} l_{ij}^2\right) + \text{специфичность}\left(\varepsilon_i^2\right).$$

Основное выражение факторного анализа показывает, что коэффициент корреляции любых двух признаков x_i и x_j можно выразить суммой произведения нагрузок некоррелированных факторов

$$r_{ij} = r(x_i, x_j) = l_{i1}l_{j1} + l_{i2}l_{j2} + \ldots + l_{im}l_{jm}$$

Задачу факторного анализа нельзя решить однозначно. Равенства в факторной модели не поддаются непосредственной проверке, так как p исходных признаков задается через $(p + m)$ других переменных — простых и специфических факторов. Поэтому представление корреляционной матрицы факторами, как говорят, ее факторизацию, можно произвести бесконечно большим числом способов. Если удалось произвести факторизацию корреляционной матрицы с помощью некоторой матрицы факторных нагрузок **F**, то любое линейное ортогональное преобразование **F** (ортогональное вращение) приведет к такой же факторизации [41]. Поэтому нередко в одном и том же пакете программ анализа данных реализовано сразу несколько версий методов факторизации, и у исследователей возникает закономерный вопрос, какой из них лучше. Здесь сошлемся на слова одного из основоположников современного факторного анализа Г. Хартмана: «Ни в одной из работ не было показано, что какой-либо один метод приближается к "истинным" значениям общностей лучше, чем другие методы... Выбор среди группы методов наилучшего производится в основном с точки зрения вычислительных удобств, а также склонностей и привязанностей исследователя, которому тот или иной метод казался более адекватным его представлениям об общности» [16].

В настоящее время одними из наиболее популярных являются три метода вращения факторов: варимакс, квартимакс и эквимакс. Вращение методом варимакс ставит целью упростить столбцы факторной матрицы, сводя все значения к 1 или 0. Вращение методом квартимакс ставит целью аналогичное упрощение только по отношению к строкам факторной матрицы. И наконец, эквимакс занимает промежуточное положение — при вращении факторов по этому методу одновременно делается попытка упростить и столбцы, и строки.

Кроме перечисленных трех методов нередко осуществляют вращение факторов до тех пор, пока не получатся результаты, поддающиеся содержательной интерпретации. Можно, например, потребовать, чтобы один фактор был нагружен преимущественно признаками одного типа, а другой — признаками другого типа. Или, скажем, можно потребовать, чтобы исчезли какие-то трудно интерпретируемые нагрузки с отрицательными знаками. Нередко исследователи идут дальше и рассматривают прямоугольную систему факторов как частный случай косоугольной, то есть ради содержания жертвуют условием некоррелированности факторов.

В целом по факторному анализу можно отметить следующее. С помощью такого анализа снижение размерности достигается за счет существования групп взаимосвязанных признаков, которые агрегируются в строящихся факторах. Как и при использовании метода главных компонент, полезные сведения о структуре данных можно почерпнуть на основании визуального анализа проектов объектов в одно-, двух- и трехмерных пространствах, образованных комбинациями различных факторов. Также ценную информацию о структуре исследуемой выборки могут дать результаты факторного анализа, проведенного раздельно в различных подгруппах объектов.

Другие методы линейного проецирования данных развиваются в рамках направления, получившего название разведочного анализа данных [47]. Современные

методы проецирования, в частности методы целенаправленного проецирования, являются естественным обобщением охарактеризованных выше классических методов анализа данных. Их систематизация и характеристики представлены в [14].

Пример применения факторного анализа

Факторный анализ широко применяется в экономике, социологии, медицине для выявления скрытых закономерностей в данных. Но, может быть, наиболее широко он используется в психологии, из которой, собственно, идут корни факторной статистической техники. Этим объясняется выбор нижеследующего примера, связанного с изучением структуры интеллекта на основе данных, полученных с помощью психологического тестирования.

Настоящий пример адаптирован по данным, приведенным в отчете об изучении пожилых людей [6]. Испытуемые были разделены с помощью теста Векслера на две полярные группы. Для первой группы характерно наличие признаков старения, для второй такие признаки отсутствуют.

В нашем случае будут рассмотрены 37 человек, у которых признаки старения выражены. Мы выделим на основе экспериментальных данных факторы и проинтерпретируем их.

Откроем файл данных с названием Senile.sf.

Таблица 2.11. Таблица с экспериментальными данными

№	Info	similars	arith	picture	№	info	similars	arith	Picture
1	7	5	9	8	20	10	10	15	8
2	8	8	5	6	21	14	7	11	5
3	16	18	11	9	22	16	11	12	11
4	8	3	7	9	23	10	7	14	6
5	6	3	13	9	24	10	10	9	6
6	11	8	10	10	25	10	7	10	10
7	12	7	9	8	26	7	6	5	9
8	8	11	9	3	27	15	12	10	6
9	14	12	11	4	28	17	15	15	8
10	13	13	13	6	29	16	13	16	9
11	13	9	9	9	30	13	10	17	8
12	13	10	15	7	31	13	10	17	10
13	14	11	12	8	32	19	12	16	10
14	15	11	11	10	33	19	15	17	11
15	13	10	15	9	34	13	10	7	8
16	10	5	8	6	35	15	11	12	8
17	10	3	7	7	36	16	9	11	11
18	17	13	13	7	37	14	13	14	9
19	10	6	10	7					

Получение и интерпретация сводки анализа

Выберем Special ▸ Multivariate Methods ▸ Factor Analysis. Система выдаст окно диалога для задания переменных.

Введем в поле анализа переменные **arith** (арифметический тест), **info** (информационный тест), **picture** (тест дополнения картинок) и **similars** (тест на подобие).

В поле Select запишем **first(37)** — первые 37 объектов, тогда как полная матрица данных содержит больше объектов. Заполненное окно диалога ввода переменных в анализ показано на рис. 2.40.

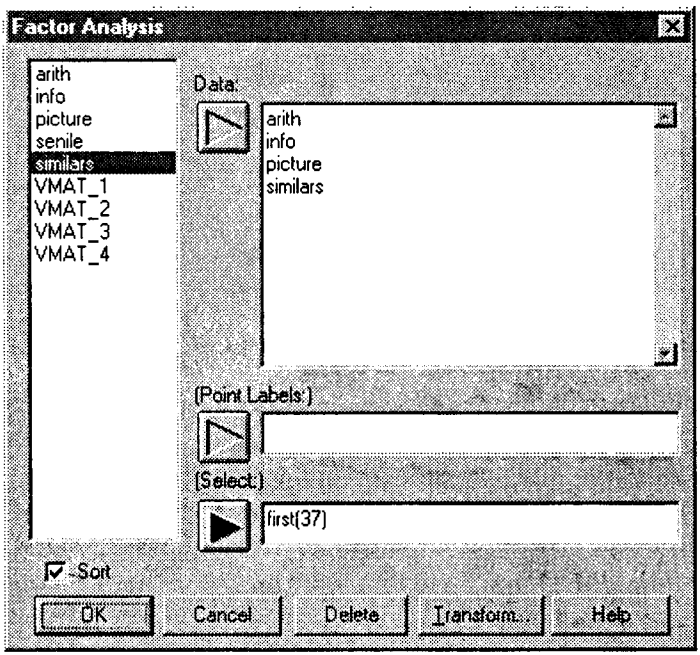


Рис. 2.40. Заполненное окно диалога ввода данных для проведения факторного анализа

Нажмем кнопку OK. Система выдаст первичную сводку факторного анализа (рис. 2.41).

Из полученной сводки следует, что на первые три фактора приходится 95 % дисперсии.

Передвинем курсор на окно первичной сводки и щелкнем правой кнопкой мыши. Система предоставит окно диалога для задания опций факторного анализа.

Оставим в неприкосновенности переключатели, указывающие на Listwise, Principal Components (тип факторизации) и Varimax (метод вращения факторов).

Снимем флажок Standartize, так как мы имеем дело с уже стандартизированными психологическими данными, измеренными в определенных шкалах.

Установим переключатель в положение Number of Factors (количество факторов) и в соответствующем поле изменим 4 на 3. Нажмем кнопку OK (рис. 2.42). Система произведет необходимые расчеты и выдаст новую сводку факторного анализа (рис. 2.43).

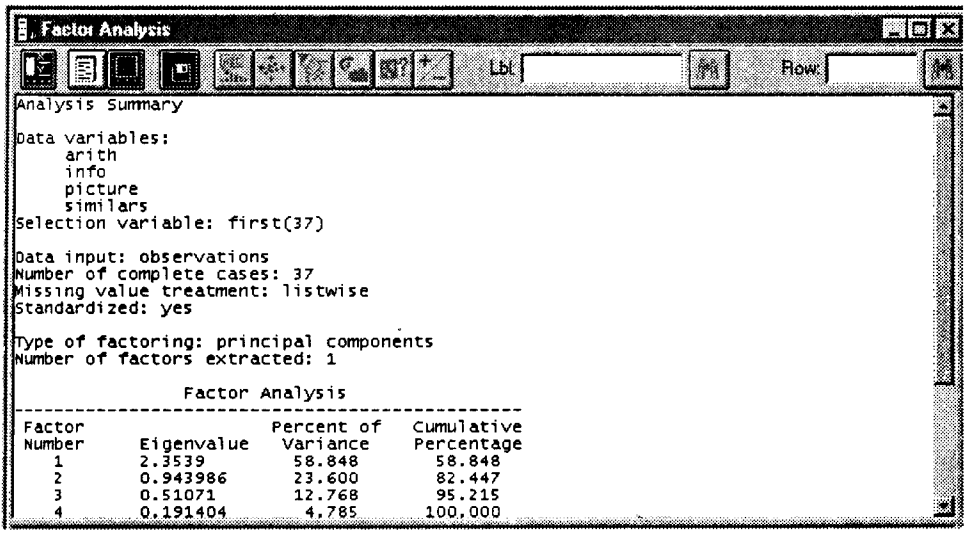


Рис. 2.41. Первичная сводка факторного анализа

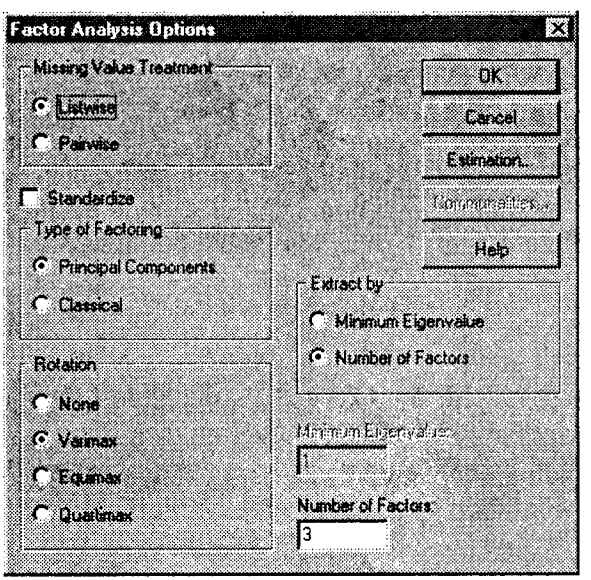


Рис. 2.42. Окно диалога для задания параметров факторного анализа

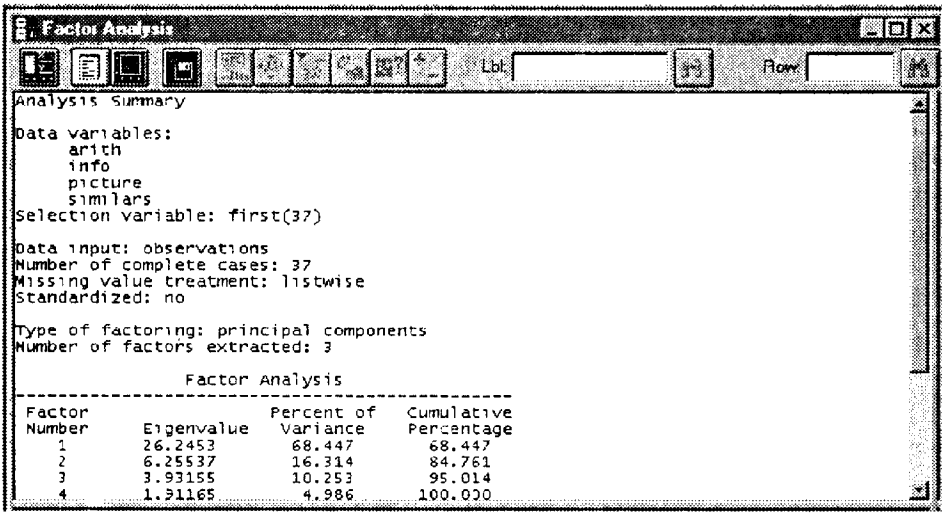


Рис. 2.43. Вторая сводка факторного анализа

Получение и интерпретация табличных результатов

Нажмем кнопку табличных опций (вторая слева в верхнем ряду). Система выдаст соответствующее окно диалога.

Щелкнем мышью на кнопке All (все) и тем самым выберем все имеющиеся виды численных представлений результатов факторного анализа. Нажмем кнопку OK. Система выдаст на рабочее поле экрана четыре окна с табличными результатами.

Дважды щелкнем левой кнопкой мыши на табличном окне Extraction Statistics (выделенные статистики). Окно займет все рабочее поле экрана (рис. 2.44).

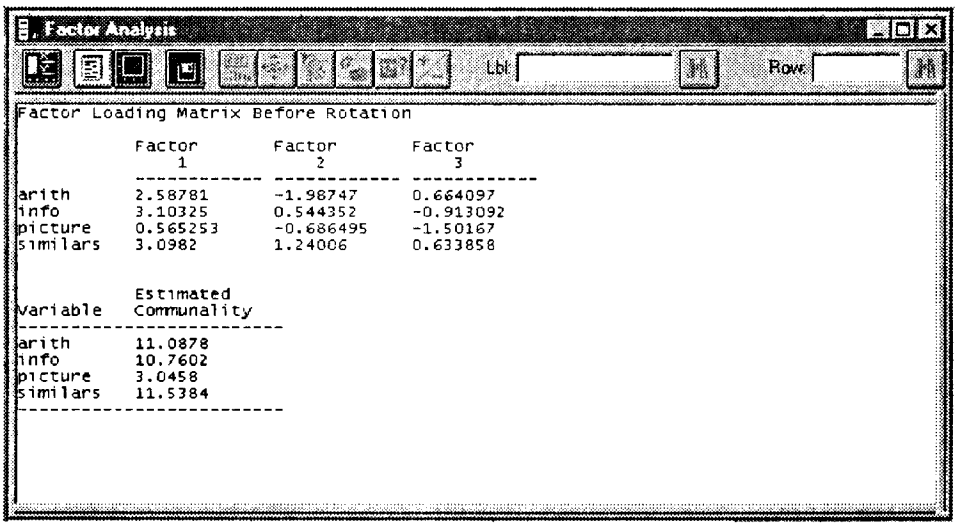


Рис. 2.44. Результаты факторизации до вращения факторов

В таблице приведены значения факторных нагрузок до применения процедуры вращения факторов. Но так как вращение факторов нередко помогает получить более полезные сведения о структуре экспериментальных данных, рассмотрим значения факторных нагрузок после проведения такого вращения.

Дважды щелкнем на раскрытом окне левой кнопкой мыши, минимизируя его размеры.

Произведем двойной щелчок на окне Rotation Statistics (нагрузки после проведения вращения) — развернем его на все рабочее поле (рис. 2.45).

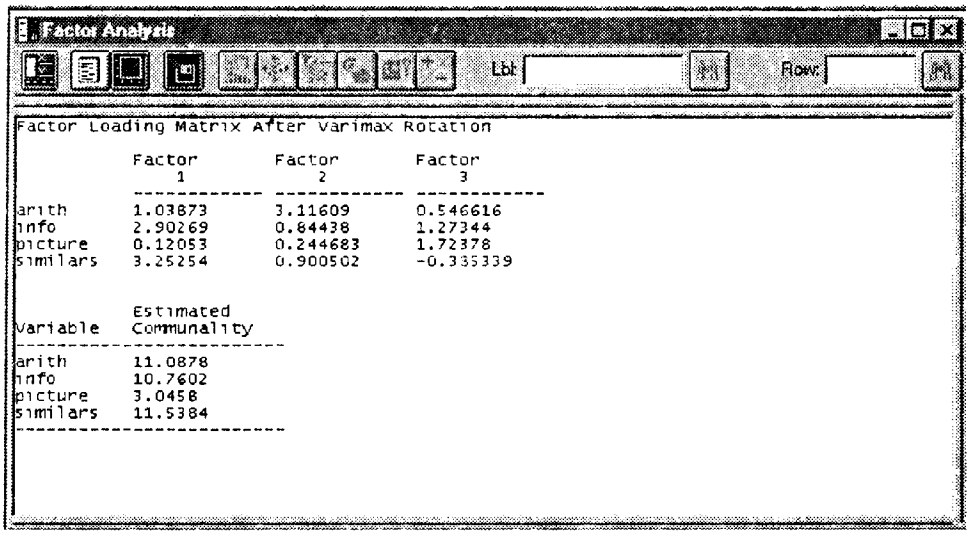


Рис. 2.45. Матрица факторных нагрузок после проведения вращения

Нетрудно видеть, что после применения процедуры вращения в факторе 2 гораздо более высокое значение имеет факторная нагрузка для переменной **arith**, которая отражает способность испытуемых к проведению арифметических действий в уме. Вместе с тем, в факторе 1 высокие величины нагрузок наблюдаются для переменных **similars** и **info**, в то время как у переменной **picture** нагрузка мала. Это говорит о том, что фактор 1 отражает различия людей по так называемому основному интеллекту.

Получение и интерпретация графических отображений

Нажмем кнопку графических опций (третья слева). Появится соответствующее окно диалога.

Щелкнем на кнопке All, задействуя все графические опции. Система добавит на рабочее поле 5 окон с различными графическими отображениями результатов факторного анализа (рис. 2.46).

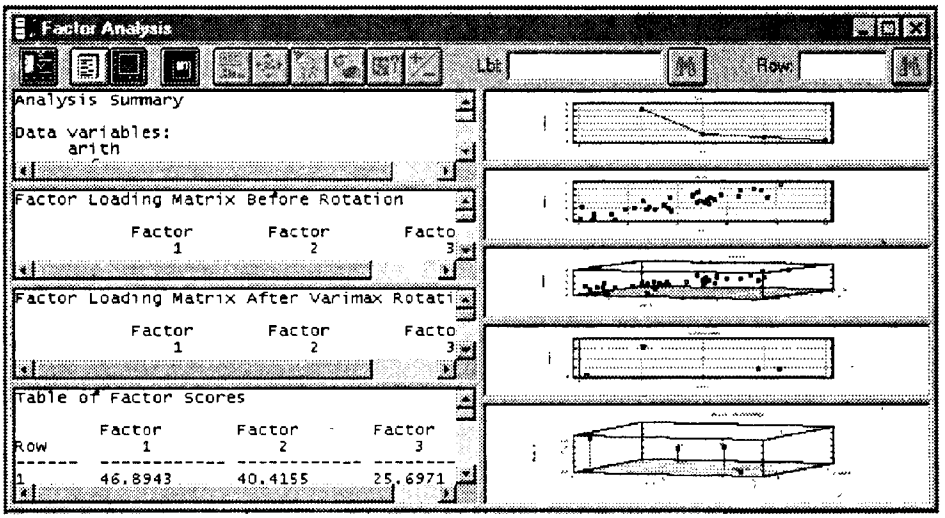


Рис. 2.46. Все табличные и графические окна факторного анализа

Раскроем сначала окно с названием Scree Plot. Этот график в исходном построении иллюстрирует собственные значения для каждого фактора. Но, предположим, нам хочется иметь выражения величины собственных значений в процентах.

Щелкнем на графике правой кнопкой мыши и получим окно диалога для задания опций данного вида отображения результатов факторизации. Установим флажок Percent of Variance (процент дисперсии) вместо Eigenvalues (собственные значения). Заметим, что фактор 1 имеет весьма высокое и, конечно, самое большое значение процента дисперсии. На фактор 2 приходится менее 20 % дисперсии, а фактор 4 и вовсе малозаметен по этому показателю (рис. 2.47).

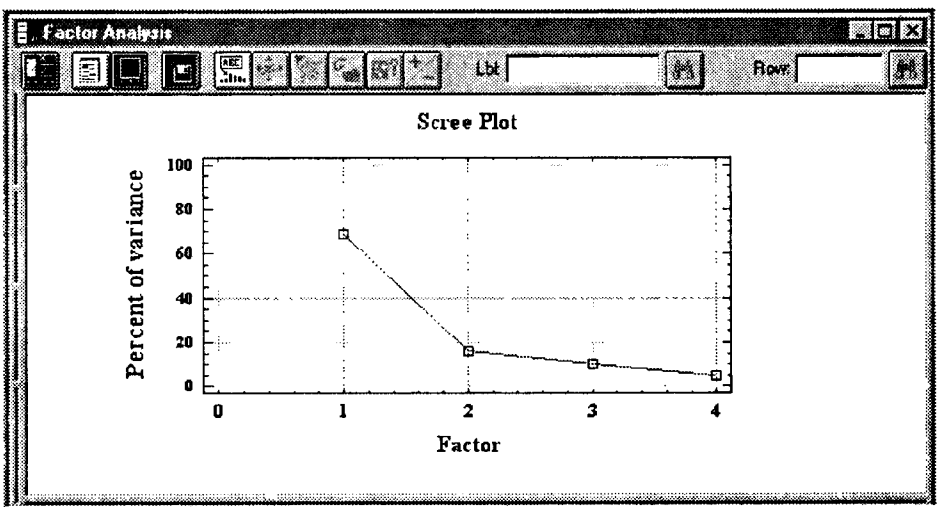


Рис. 2.47. Иллюстрация процента дисперсии для выделенных факторов

Минимизируем размеры рассмотренного графика.

Дважды щелкнем левой кнопкой мыши на втором графическом окне 2D Scatterplot (двухмерная диаграмма рассеивания). На полученном рисунке показана проекция исследуемых объектов на плоскость, образованную первым и вторым факторами. Судя по конфигурации облака точек, первый и второй факторы сильно коррелируют. То есть в нашем случае, применительно к пожилым людям с выраженными признаками старения, видно что общий интеллект у них тесно связан со способностью к произведению в уме арифметических действий (рис. 2.48).

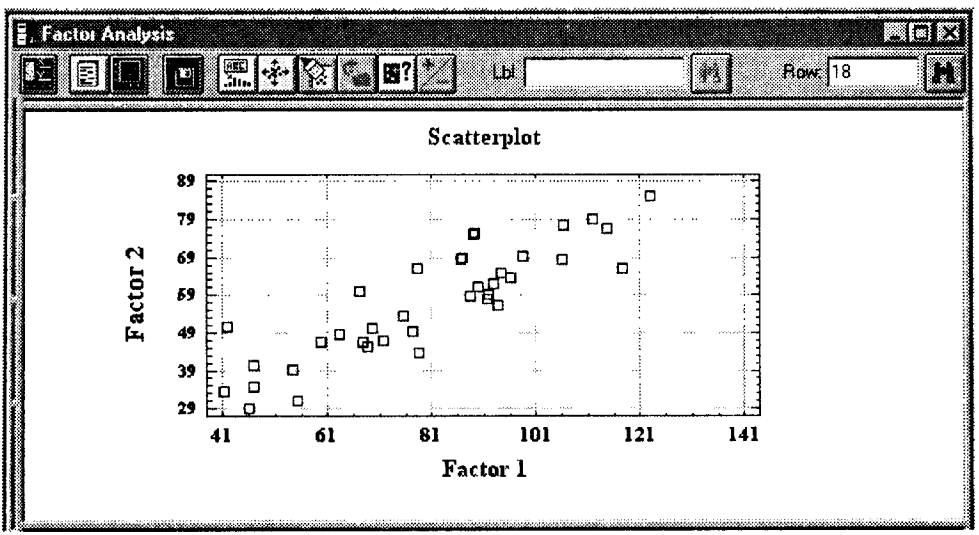


Рис. 2.48. Проекция объектов на плоскость первого и второго факторов

Вернем двухмерной диаграмме рассеивания прежние минимальные размеры, дважды щелкнув на ней левой кнопкой мыши.

Максимизируем размеры графического отображения **3D Scatterplot**. Этот график представляет собой проекцию объектов в трехмерное пространство, образованное первыми тремя факторами.

Свернем рассмотренное отображение.

Дважды щелкнем левой кнопкой мыши на графике **2D Factor Plot** (двухмерное отображение факторных нагрузок), раскрывая график на все рабочее поле (рис. 2.49).

На графике хорошо видно, что переменная **arith** имеет значение как для первого, так и для второго факторов. Вместе с тем, у переменной **picture** малые нагрузки на все факторы, а тестовые измерения **info** и **similars** имеют большие нагрузки только на первый фактор.

Свернем раскрытое окно до минимальных размеров, дважды щелкнув на нем левой кнопкой мыши.

Раскроем до максимальных размеров окно 3D Factor Plot (трехмерный факторный график). График изображает факторные нагрузки уже в пространстве трех факторов (рис. 2.50).

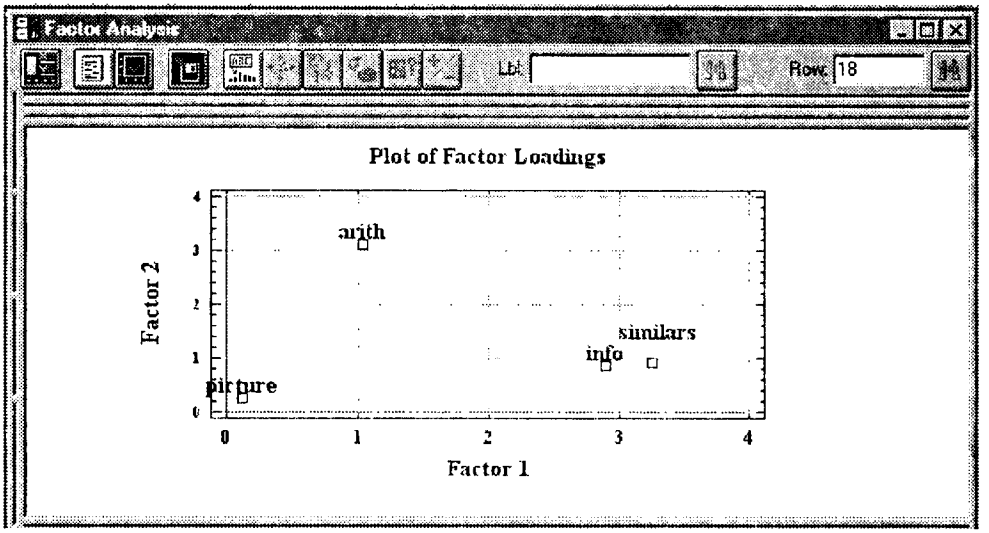


Рис. 2.49. Графическое изображение факторных нагрузок

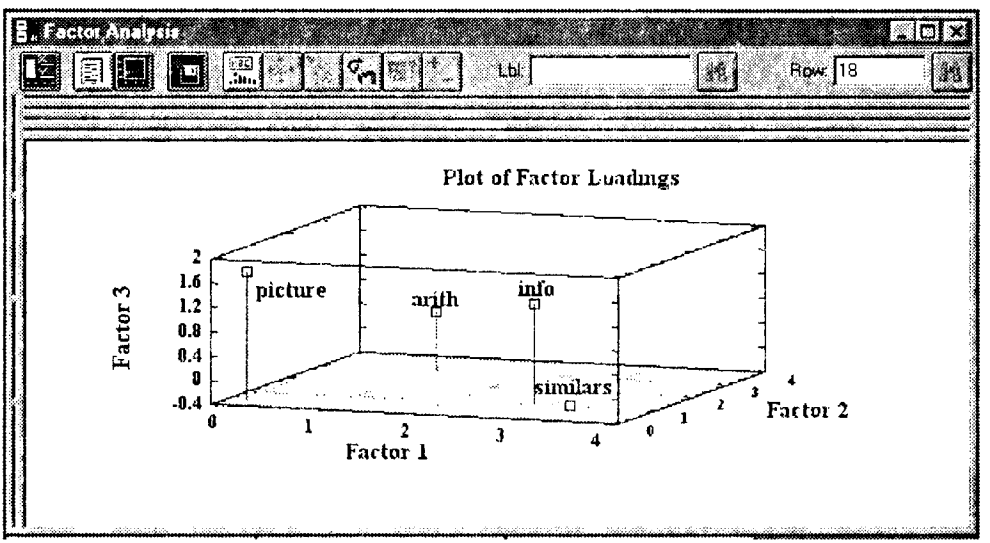


Рис. 2.50. Отображение факторных нагрузок в пространстве трех факторов

Из приведенного рисунка видно, что весьма высокую нагрузку имеет переменная **picture**. Однако эта нагрузка приходится на третий выделенный фактор, а как нам известно из предыдущих данных, третий фактор играет несущественную роль при описании рассмотренных экспериментальных наблюдений.

Нелинейные отображения

Нелинейные методы отображения данных в пространство меньшей размерности, как правило, пренебрегают аналитическим выражением преобразования исходного пространства признаков в новые координатные оси, позволяющие интерпретировать новое координатное пространство. Они не скованы никакими ограничениями на вид допустимых преобразований. Все операции подчинены одной главной цели: построить графическое изображение совокупности данных, дающее наиболее наглядное представление об особенностях их структуры. Причем особенности структуры понимаются достаточно широко. Это приводит к тому, что алгоритмы нелинейного отображения могут быть направлены не обязательно на минимальное искажение всех попарных расстояний между объектами выборки в исходном пространстве признаков, а, например, на максимально точное отображение только сравнительно больших расстояний или, наоборот, только малых. Такая гибкость методов нелинейного отображения позволяет настраивать их на тот или иной интересующий аспект структуры данных и как бы целенаправленно зондировать изучаемую выборку.

Для получения нелинейных отображений $y(\mathbf{x})$ задается некоторый критерий (мера) искажения структуры данных $J\{y(\mathbf{x})\}$ и решается задача на определение минимума J. Большинство мер искажения основано на сравнении попарных расстояний между объектами в исходном пространстве и пространстве отображения. Например, используется мера, предложенная в [7] и являющаяся аналогом критерия «стресса», применяемого в многомерном шкалировании,

$$J\{y(\mathbf{x})\} = \frac{1}{\sum_{i>j}^{N-1}(d_{ij}^*)^a}\sum_{i>j}^{N-1}(d_{ij}^* - d_{ij})^2 (d_{ij}^*)^a,$$

где d_{ij}^* — расстояние между i-м и j-м объектами в исходном пространстве R^p; d_{ij} — евклидово расстояние между отображениями этих объектов в $R^{p'}$.

Если в приведенном критерии принять $a < 0$, то он станет более чувствительным к ошибкам отображения малых расстояний и менее чувствительным к искажению больших расстояний. При $a > 0$, наоборот, точнее отображаются большие расстояния и загрубляются малые, так как критерий начинает сильнее реагировать на ошибки в передаче больших расстояний. Обычно результаты, полученные для $a < 0$, лучше, чем для $a > 0$ [14].

Несколько более разнообразные возможности предоставляет использование двухпараметрического критерия, предложенного в [45]:

$$a = \begin{cases} a_1, \text{ если } d_{ij} > d_{ij}^* \\ a_2, \text{ если } d_{ij} \le d_{ij}^* \end{cases}.$$

Данный критерий может оказаться полезным, если при отображении объектов в $R^{p'}$ требуется большие расстояния еще больше увеличить, а малые — еще сильнее уменьшить. Этот эффект будет получен, если принять $a_1 < 0$ и $a_2 > 0$.

Поиск отображений объектов в пространство меньшей размерности, минимизирующих значение критерия J, осуществляется, как правило, с помощью различ-

ных градиентных процедур. Большой выбор таких процедур для решения данной задачи, а также разнообразные варианты критерия J предлагаются, например, в [42], [45]. В качестве начального приближения для новых координат объектов часто используются их проекции на первые главные компоненты. Размерность пространства для визуального анализа данных в R^p', допустимое количество итераций в градиентной процедуре и точность отображения задаются исследователем. В зависимости от выбранного критерия J могут получаться различные конфигурации точек в R^p' и может существенно варьироваться время работы алгоритма отображения, которое также во многом определяется типом применяемой градиентной процедуры. Известны другие, менее распространенные разновидности методов нелинейного отображения объектов. Они рассматриваются, например, в [14].

Многомерное шкалирование

Многомерное шкалирование — совокупность методов, позволяющих по заданной информации о мерах различия (близости) между объектами рассматриваемой совокупности приписывать каждому из этих объектов вектор характеризующих его количественных показателей. При этом размерность искомого координатного пространства задается заранее, а «погружение» в него анализируемых объектов производится таким образом, чтобы структура взаимных различий (близостей) между ними, измеренных с помощью приписываемых им вспомогательных координат, в среднем наименее отличалась бы от заданной в смысле того или иного функционала качества [14]. Процедуры многомерного шкалирования отличаются от описанных выше методов линейного и нелинейного проецирования данных в пространство меньшей размерности в основном тем, что исходной информацией для них служит только матрица различий (близостей) между исследуемыми объектами и не требуется знания значений признаков для этих объектов. Когда информация задана в виде матрицы попарных расстояний между объектами, используются методы так называемого метрического шкалирования. Если же элементы матрицы выражают порядковые отношения между объектами, то применяются методы неметрического шкалирования. Ниже охарактеризован классический подход к решению задачи метрического шкалирования.

Обычно, хотя и не обязательно, пространство предполагается евклидовым. Для этого случая справедливы следующие преобразования, которые необходимы для перехода от матрицы расстояний $\mathbf{D} = (d_{ij})$ к координатам объектов в пространстве для визуального анализа $x_1, \ldots, x_p$.

Метод определения координат точек $\mathbf{x}_1, \ldots, \mathbf{x}_N$ (с точностью до ортогонального вращения) и заодно размерности пространства, в которое они отображаются, основан не на непосредственном использовании матрицы $\mathbf{D}$, а на преобразовании ее в матрицу $\mathbf{B}$ скалярных произведений центрированных векторов:

$$b_{ij} = (\mathbf{x}_i - \mu)^{\mathrm{T}}(\mathbf{x}_j - \mu),$$

где μ — вектор средних значений.

Между элементами матрицы $\mathbf{B}$ и расстояниями d_{ij} установлено следующее соотношение.

$$b_{ij} = \left(-d_{ij}^2 + \frac{1}{N}\sum_{i=1}^{N} d_{ij}^2 + \frac{1}{N}\sum_{j=1}^{N} d_{ij}^2 - \frac{1}{N^2}\sum_{i,j} d_{ij}^2 \right).$$

Процедура перехода от **D** к **B** называется двойным центрированием **D**. Матрица **B** размером $N \times N$ обладает следующими свойствами:

1. Неотрицательно определена.
2. Ранг матрицы **B** равен размерности искомого пространства отображения.
3. Ненулевые собственные числа матрицы **B**, упорядоченные в порядке убывания, совпадают с соответствующими собственными числами матрицы $\mathbf{S}=\mathbf{X}\mathbf{X}^T$, где **X** — центрированная матрица данных (неизвестная нам). Матрица $\mathbf{S}/N$ есть матрица ковариаций для **X**.
4. Пусть $\mathbf{u}_r$ есть r-й собственный вектор матрицы **S**, соответствующий r-му собственному числу λ_r. Тогда вектор значений r-й главной компоненты будет $\mathbf{z}_r = \mathbf{X}^T\mathbf{u}_r$.

В то же время, пусть $\mathbf{y}_r$ — r-й собственный вектор матрицы **B**, соответствующий тому же самому собственному значению λ_r, то есть

$$\mathrm{B}\, \mathrm{y}_r = \lambda_r \mathrm{y}_r.$$

Тогда

$$\mathrm{z}_r = \sqrt{\lambda_r}\, \mathrm{y}_r.$$

Из свойства 4 следует, что, решая задачу собственных чисел и собственных векторов для матрицы **B** и ограничиваясь ненулевыми собственными числами $\lambda_1, \ldots, \lambda_p$ получаем координатное представление точек в пространстве главных компонент, основываясь на приведенных формулах.

Элементы матрицы **B** могут быть представлены в виде

$$b_{ij} = \sum_{r=1}^{p'} z_{ir} z_{jr}.$$

Очевидно, решение **Z** является линейной функцией **X** и определяется лишь с точностью до ортогонального преобразования, поскольку, применяя к матрице **Z** преобразование вращения, получим, что преобразованная матрица $\mathbf{Z}^*$ столь же точно восстанавливает матрицу **B**, как и матрица **Z**. Поэтому такое шкалирование называют линейным.

Подробно с классическим подходом к многомерному шкалированию можно ознакомиться в работах [11], [26], [45]. Решение задачи шкалирования, полученное классическим линейным методом, часто используется как начальное приближение в процедурах нелинейного многомерного шкалирования, которые строятся аналогично рассмотренным выше процедурам нелинейного проецирования данных в пространство меньшей размерности. Особенности этих процедур описаны в приведенной литературе по многомерному шкалированию.

Заполняющие пространство кривые

Суть данной группы методов состоит в заполнении пространства признаков гиперкривой таким образом, чтобы близкие в пространстве объекты оказались по возможности близкими и на этой кривой. Визуальному анализу подвергаются гистограммы распределений объектов на построенной гиперкривой. Весь процесс называется разверткой пространства признаков.

В [17] описан рекурсивный алгоритм порождения кривой, заполняющей многомерный интервал, дальнейшее исследование которого проводилось, например, в [22]. Для построения заполняющей пространство кривой (ЗПК) табличные данные приводятся к единичному p-мерному гиперкубу. Стороны этого гиперкуба разбивают на части и получают квантованное p-мерное. ЗПК порождается путем задания рекурсивного правила обхода данных квантов.

Главными свойствами отображений объектов на ЗПК являются взаимная однозначность, сходимость по разбиениям и квазинепрерывность. Взаимная однозначность выражает строго определенное соответствие каждого кванта p-мерного пространства какому-либо участку ЗПК. Сходимость по разбиениям означает, что при очередном дроблении пространства на ЗПК сохраняется закон принадлежности отображений новых, более мелких квантов старому — более крупному. Под квазинепрерывностью понимается то, что два соседних отображения квантов на ЗПК обязательно являются также соседними в многомерном пространстве (обратное условие может не выполняться).

Приведенные свойства дают основание считать ЗПК полезным инструментом исследования структуры данных. Развертки пространства признаков с помощью ЗПК хорошо дополняют информацию о взаимном расположении объектов выборки, которую экспериментатор получает, рассматривая проекции объектов, например, на плоскости главных компонент или выделенных факторов. Если для таких проекций справедливо правило: далекие на проекции объекты обязательно далеки в исходном пространстве R^p, но близкие на проекции объекты могут быть далекими в R^p, — то для ЗПК все происходит наоборот: близкие на развертке объекты обязательно близки в R^p, но далекие на развертке объекты могут быть близкими в R^p. Разновидность ЗПК, не требующая квантования пространства признаков и названная адаптивной разверткой пространства описания, описана в [29].

Методы автоматического группирования

Факторный анализ объектов

При факторном анализе объектов используется формальный аппарат факторного анализа, изначально предназначавшийся для агрегирования взаимосвязанных признаков. Этому аппарату была дана характеристика в предыдущем подразделе. Отличие состоит в том, что в факторном анализе объектов таблица экспериментальных данных поворачивается на 90^0 (транспонируется), то есть объекты и признаки меняются местами. Если при факторном анализе признаков ищутся группы близких (коррелированных) признаков на основе корреляционной мат-

рицы, то для транспонированных данных аналогом корреляционной матрицы является матрица, описывающая попарные коэффициенты корреляции (сходства) объектов. Она вводится в алгоритм формального факторного анализа, и в результате получаются факторы, описывающие уже не группы коррелированных признаков, а группы сходных объектов [16]. Особенности данной процедуры подробно рассмотрены в [13].

Кластерный анализ

Этот анализ предназначен для разбиения множества объектов на заданное или неизвестное число классов на основании некоторого математического критерия качества классификации (*cluster* (англ.) — гроздь, пучок, скопление, группа элементов, характеризуемых каким-либо общим свойством). Критерий качества кластеризации в той или иной мере отражает следующие неформальные требования [40]:

1. Внутри групп объекты должны быть тесно связаны между собой.
2. Объекты разных групп должны быть далеки друг от друга.
3. При прочих равных условиях распределения объектов по группам должны быть равномерными.

Требования пунктов 1 и 2 выражают стандартную концепцию компактности классов разбиения [19]; требование пункта 3 состоит в том, чтобы критерий не навязывал объединения отдельных групп объектов.

Узловым моментом в кластерном анализе считается выбор метрики (или меры близости объектов), от которого решающим образом зависит окончательный вариант разбиения объектов на группы при заданном алгоритме разбиения [14]. В каждой конкретной задаче этот выбор производится по-своему, с учетом главных целей исследования, физической и статистической природы исследуемой информации и т. п.

Другой важной величиной в кластерном анализе является расстояние между целыми группами объектов. Приведем примеры наиболее распространенных расстояний, характеризующих взаимное расположение отдельных групп объектов (рис. 2.51).

Пусть ω_l — l-я группа (класс, кластер) объектов, N_l — число объектов, образующих группу, вектор μ_l — среднее арифметическое объектов, входящих в ω_l (другими словами, «центр тяжести» l-й группы), а $\rho(\omega_l, \omega_m)$ — расстояние между группами ω_l и ω_m.

Расстояние ближайшего соседа есть расстояние между ближайшими объектами кластеров:

$$\rho_{\min}(\omega_l, \omega_m) = d(\mu_i, \mu_j).$$

Расстояние центров тяжести равно расстоянию между центральными точками кластеров:

$$\rho_{\min}(\omega_l, \omega_m) = \min_{x_i \in \omega_l, x_j \in \omega_m} d(x_i, x_j).$$

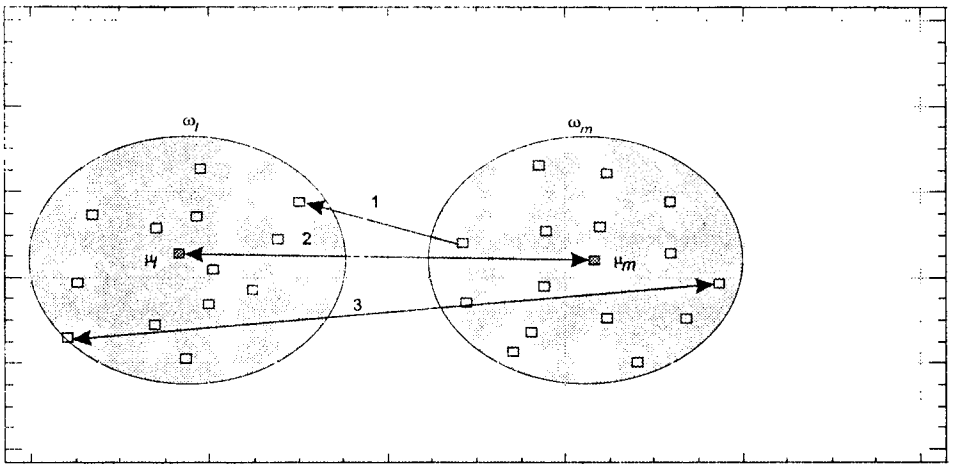


Рис. 2.51. Различные способы определения расстояния между кластерами: 1 — по ближайшим объектам, 2 — по центрам тяжести, 3 — по самым дальним объектам

Расстояние дальнего соседа — расстояние между самыми дальними объектами кластеров:

$$\rho_{\max}(\omega_l, \omega_m) = \max_{x_i \in \omega_l, x_j \in \omega_m} d(x_i, x_j).$$

Выбор той или иной меры расстояния между кластерами влияет, главным образом, на вид выделяемых алгоритмами кластерного анализа геометрических группировок объектов в пространстве признаков. Так, алгоритмы, основанные на расстоянии ближайшего соседа, хорошо работают в случае группировок, имеющих сложную, в частности, цепочечную структуру. Расстояние дальнего соседа применяется, когда искомые группировки образуют в пространстве признаков шаровидные облака. И промежуточное место занимают алгоритмы, использующие расстояние центров тяжести и средней связи, которые лучше всего работают в случае группировок эллипсоидной формы.

Нацеленность алгоритмов кластерного анализа на определенную структуру группировок объектов в пространстве признаков может приводить к неоптимальным или даже неправильным результатам, если гипотеза о типе группировок неверна. В случае отличия реальных распределений от гипотетических указанные алгоритмы часто «навязывают» данным не присущую им структуру и дезориентируют исследователя. Поэтому экспериментатор, учитывающий данный факт в условиях априорной неопределенности, прибегает к применению батареи алгоритмов кластерного анализа и отдает предпочтение какому-либо выводу на основании комплексной оценки результатов работы этих алгоритмов.

Алгоритмы кластерного анализа отличаются большим разнообразием. Это могут быть, например, алгоритмы, реализующие полный перебор сочетаний объектов или осуществляющие случайные разбиения множества объектов. Вместе с тем, большинство таких алгоритмов состоит из двух этапов. На первом этапе задается начальное (возможно, искусственное или даже произвольное) разбие-

ние множества объектов на классы и определяется некоторый математический критерий качества автоматической классификации. Затем, на втором этапе, объекты переносятся из класса в класс до тех пор, пока значение критерия не перестанет улучшаться.

Многообразие алгоритмов кластерного анализа обусловлено также множеством различных критериев, отражающих те или иные аспекты качества автоматического группирования. Простейший критерий качества непосредственно базируется на величине расстояния между кластерами. Однако такой критерий не учитывает «населенность» кластеров — относительную плотность распределения объектов внутри выделяемых группировок. Поэтому другие критерии основываются на вычислении средних расстояний между объектами внутри кластеров. Но наиболее часто применяют критерии в виде отношений показателей «населенности» кластеров к расстоянию между ними. Это, например, может быть отношение суммы межклассовых расстояний к сумме внутриклассовых (между объектами) расстояний или отношение общей дисперсии данных к сумме внутриклассовых дисперсий и дисперсий центров кластеров.

Функционалы качества и конкретные алгоритмы автоматической классификации (группирования) достаточно полно и подробно рассмотрены в [14].

Иерархическое группирование

Процедуры иерархического типа предназначены для получения наглядного представления о стратификационной структуре всей исследуемой совокупности объектов. Эти процедуры основаны на последовательном объединении кластеров (агломеративные процедуры) и на последовательном разбиении (дивизимные процедуры). Наибольшее распространение получили агломеративные процедуры. Они выглядят следующим образом.

На первом шаге все объекты считаются отдельными кластерами. Затем на каждом шаге два ближайших кластера объединяются в один. Каждое объединение уменьшает число кластеров на один так, что в конце концов все объекты объединяются в один кластер. Наиболее подходящее разбиение выбирает чаще всего сам исследователь, которому предоставляется дендрограмма, отображающая результаты группирования на всех шагах алгоритма (рис. 2.52). Могут одновременно использоваться также и математические критерии качества группирования.

Различные варианты определения расстояния между кластерами дают различные варианты иерархических процедур. Учитывая их специфику, для задания расстояния между кластерами оказывается достаточным указать порядок пересчета расстояний между кластером k и кластером (i, j), являющимся объединением двух других кластеров i и j по расстояниям d_{ki}, d_{kj} и d_{ij}. Для этого используется широко известная формула

$$d_{k(ij)} = a_i d_{ki} + a_j d_{kj} + b d_{ij} + c|d_{ki} - d_{kj}|,$$

где a_i, a_j, b, c — параметры, которыми определяется тот или иной вид расстояния между кластерами. Например, при $a_i = a_j = \frac{1}{2}$, $b = 0$, $c = -\frac{1}{2}$ приходим к расстоянию, измеряемому по принципу ближайших соседей между двумя кластера-

ми; $a_i = a_j = \frac{1}{2}$, $b = 0$, $c = \frac{1}{2}$ дает расстояние, измеряемое по принципу дальнего соседа и т. д.

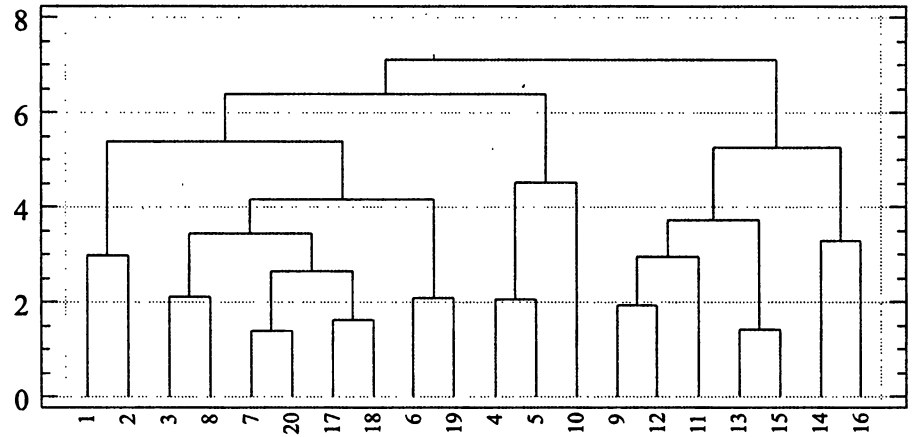


Рис. 2.52. Результаты работы иерархической агломеративной процедуры группирования объектов, представленные в виде дендрограммы (количество объектов — 20, расстояние между кластерами — дальний сосед, евклидова метрика)

В отличие от оптимизационных кластерных алгоритмов, предоставляющих исследователю конечный результат группирования объектов, иерархические процедуры позволяют проследить процесс выделения кластеров, образующихся на разных шагах агломеративного или дивизимного алгоритма. Это стимулирует воображение исследователя и помогает ему привлекать для оценки структуры данных дополнительные формальные и неформальные представления.

Определение «точек сгущения»

Алгоритмы автоматической классификации, основанные на определении «точек сгущения», или, в другой терминологии, на описании классов «ядрами», представляют обширную группу алгоритмов, непосредственно опирающихся на выделение в многомерном пространстве компактных совокупностей точек. В несколько упрощенном виде методы определения «точек сгущения» можно охарактеризовать следующим образом.

Сначала несколько объектов, выбранных по каким-либо соображениям (возможно, даже произвольно), назначаются центрами кластеров. Затем поочередно все остальные объекты относятся к тем кластерам, центры которых к ним ближе всего. Центры кластеров пересматриваются либо после включения каждого нового объекта, либо после отнесения всех объектов. Некоторые кластеры могут быть ликвидированы, если для них не выполняется заданный критерий качества. Также могут образовываться новые кластеры. Описанная процедура повторяется до получения устойчивой классификации.

К наиболее известным представителям данной группы алгоритмов относятся алгоритмы Форель [32], [46], методы динамических сгущений и менее сложные

эвристические алгоритмы, такие как k эталонов, «взаимного поглощения» и другие, рассмотренные в [14].

Сравнительный анализ различных алгоритмов автоматического группирования был проведен в работе Дж. Меззиха [34]. Сравнивались следующие алгоритмы: факторный анализ объектов, один оптимизационный кластерный алгоритм, два алгоритма поиска «точек сгущения» и три варианта иерархической процедуры. По результатам исследования на четырех наборах реальных данных сделан вывод, что наилучшим образом зарекомендовали себя два алгоритма (из трех самых распространенных) иерархического группирования. Собственный опыт и опыт других авторов (например, [16]) подтверждает этот вывод. Для практического исследования структуры многомерных данных с помощью алгоритмов автоматического группирования часто оказывается достаточным иметь в пакете прикладных программ один иерархический алгоритм группирования объектов по принципу ближайшего соседа и один алгоритм, группирующий объекты по принципу центров тяжести или средней связи.

Пример кластерного анализа

В качестве примера рассмотрим интересующую многих, пока в основном в западных странах, задачу о рынке ценных бумаг, в частности проблему оценки различных фондов, оперирующих этими бумагами.

Несмотря на неспокойность мирового рынка ценных бумаг, инвесторы сегодня вкладывают в него свои средства и имеют к нему повышенный интерес. Например, даже несмотря на то, что большинство фондов ценных бумаг в 1993 и 1994 годах функционировали без особого блеска, американцы в этот период вложили в них рекордное количество денег.

В рассматриваемом примере будут исследованы 16 известных инвестиционных фондов для оценки их состояния. В качестве переменных используются следующие характеристики (большинство из них описывается в условных единицах): доходность за пятилетний период — переменная **Five_Yr**, риск — переменная **Risk**, ежегодный процент дохода (performance) (для каждого года) — **Perf90**, **Perf91**, **Perf92**, **Perf93**, **Perf94**, расходная часть — переменная **Expence** и налоговые рейтинги — переменная **Tax**. Ниже приводится таблица (табл. 2.12) с исходными данными по исследуемым фондам. В первом столбце указано наименование фонда, а в последнем — рекомендации экспертов по операциям с ценными бумагами этих фондов. Данные заимствованы из руководства по применению STATGRAPHICS *Plus* for Windows.

Исследование приведенных данных состоит их трех частей. На первом этапе, излагаемом в настоящем разделе, будут изучаться многомерные группировки общественных фондов, полученные методами кластерного анализа STATGRAPHICS. Второй и третий этапы представлены в разделе «Практикумы» руководства по применению STATGRAPHICS *Plus* for Windows. При изложении второго этапа приводятся результаты построения линейных дискриминантных функций для разделения фондов на группы в соответствии с рекомендациями экспертов по операциям с ценными бумагами. Третья часть отведена задаче формирования базы знаний методами локальной геометрии для решения той же проблемы.

Таблица 2.12. Анализируемые данные

Fund	Five_Yr	Risk	Perf 90	Perf 91	Perf 92	Perf 93	Perf 94	Expence	Tax	Recom.
F. Chip	16476	2	10	25	6	55	4	1.22	89	Buy
F. Contra	15476	2	−1	21	16	55	4	1.03	90	Buy
F. Destiny	14757	3	4	26	15	39	−3	0.7	69	Buy
Vista A	15145	4	−1	20	13	71	−6	1.49	96	Hold
Berger 100	15596	5	−7	21	9	89	−6	1.7	95	Hold
Gab. Assett	13640	1	0	22	15	18	−6	1.33	85	Buy
Neub. Focus	14081	3	1	16	21	25	−6	0.85	75	Buy
F. Magellan	13827	3	−2	25	7	41	−5	0.96	73	Buy
Janus	13187	2	−1	11	7	43	−1	0.91	85	Sell
L. Mason Value	13029	4	1	12	11	35	−17	1.82	92	Hold
Gabelli Growth	12301	3	−3	11	4	34	−2	1.41	80	Buy
Franklin Growth	11793	2	3	7	3	27	2	0.77	90	Sell
Janus 20	12441	4	−7	3	2	69	1	1.02	95	Sell
AARP Capital	11728	4	−10	16	5	41	−16	0.97	68	Sell
Kemper Growth A	11386	4	−6	2	−2	67	4	1.09	86	Sell
20th Cent. Growth	11258	4	−8	15	−4	32	0	1	60	Buy

Введем приведенные данные в электронную таблицу STATGRAPHICS и сохраним их в файле с именем growth. Выберем Special ▸ Multivariate Methods ▸ Cluster Analysis. Система отобразит окно диалога для ввода данных в кластерный анализ (рис. 2.53).

Дважды щелкнем левой кнопкой мыши на переменных **Expence**, **Five_Yr**, **Perf90**, **Perf91**, **Perf92**, **Perf93**, **Perf94**, **Risk** и **Tax** для задействования их в анализе.

Введем характеристику Fund в поле Point Labels и оставим поле данных Select пустым. На рис. 2.53 показан пример заполнения окна диалога для ввода информации в кластерный анализ.

Нажмем OK. Система выдаст окно с первичной сводкой кластерного анализа.

Так как в нашем случае желательно, чтобы кластерный алгоритм хорошо работал с небольшим количеством наблюдений (у нас их всего 16) и был нацелен на выделение кластеров с приблизительно равным числом членов, остановим свой выбор на методе Варда (**Wards method**).

Щелкнем правой кнопкой мыши — на экране появляется окно диалога для выбора параметров кластерного анализа.

Установим флажок Wards, а все остальные оставим в прежнем положении (рис. 2.54).

Нажмем OK; на экране отобразится сводка кластерного анализа для выбранного метода.

Нажмем кнопку для задания графических опций (третья слева в верхнем ряду окна анализа). Система предоставит специальное окно диалога.

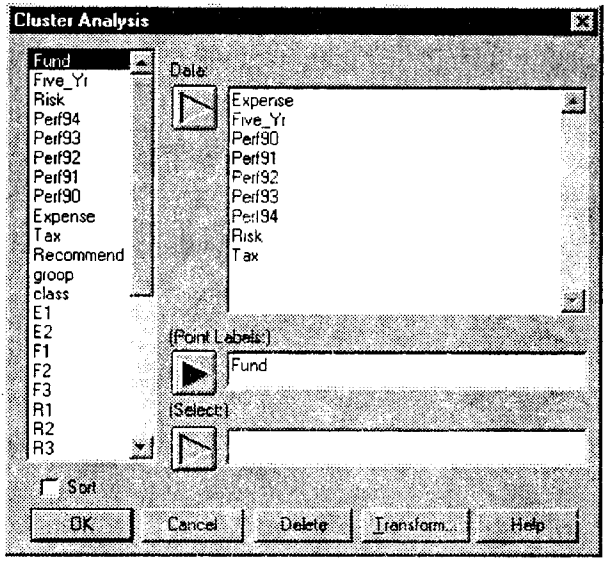


Рис. 2.53. Пример заполнения окна диалога ввода данных для кластерного анализа

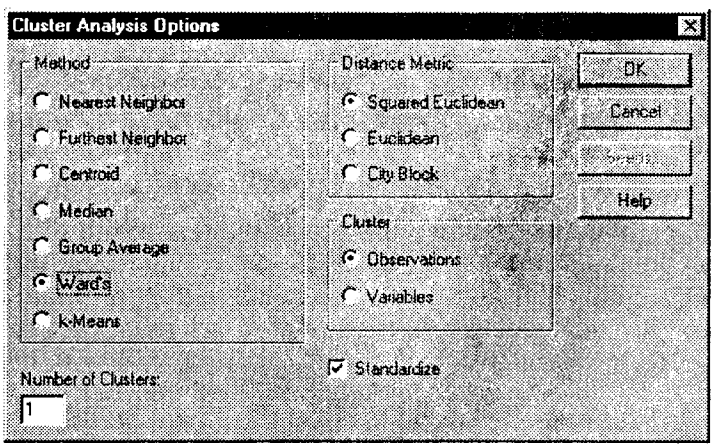


Рис. 2.54. Пример заполнения окна диалога для выбора параметров кластерного анализа

Выберем отображение в виде дендрограммы (**Dendrogram**) и нажмем кнопку OK. Система добавит к табличному окну графическое окно.

Дважды щелкнем на дендрограмме для максимального раскрытия окна (рис. 2.55).

Дендрограмма отображает иерархическую структуру группирования инвестиционных фондов. На ней отчетливо видны как минимум три группировки: одна заканчивается на фонде Gabelli Growth, вторая заканчивается на фонде Legg Mason Value и третья, достаточно плотная группировка, — на фонде 20th Century Growth. Отсюда следует, что для более подробного рассмотрения группировок следует задать их количество равным 3.

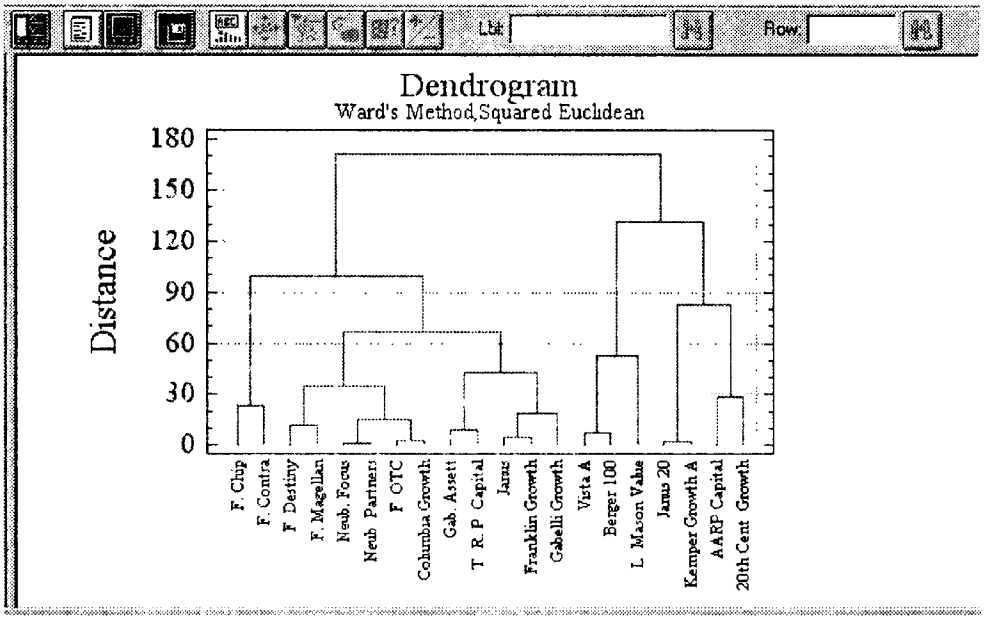


Рис. 2.55. Дендрограмма, полученная методом Варда для одного кластера

Дважды щелкнем на рисунке для минимизации размеров окна.

Щелкнем правой кнопкой мыши на окне сводки кластерного анализа — появится окно диалога для задания параметров проводимого исследования.

Изменим количество кластеров (**Number of Clusters**) с 1 до 3.

Нажмем кнопку OK. В соответствии с введенными изменениями будут произведены табличные преобразования (рис. 2.56 и 2.57).

В сводке кластерного анализа прежде всего указываются: имена переменных, участвующих в анализе, количество полных образцов (наблюдений без пропусков), использованный метод кластерного анализа и принятая метрика. Затем в сводке описываются: число кластеров, количество объектов в каждом кластере (населенность) и соответствующий процент населенности. Кроме того, в нижней части сводки приводится важная дополнительная информация.

Например, по координатам центроидов (рис. 2.57) можно судить о том, какие переменные играют наиболее важную роль в каждом кластере. В частности, в первом кластере видно, что расходы были разумными: несмотря на низкие доходы в 1990 году, заметно, что в другие годы состояние фондов 1-го кластера постоянно улучшалось. Также в первом кластере индицируется самый низкий рейтинг риска среди всех кластеров, а налоговые сборы были тоже достаточно невысокими.

Переменные, представляющие кластер 2, говорят о том, что здесь имелись наибольшие расходы, хотя за пятилетний период доходы оставались самыми высокими. Оценка риска и налоговые сборы являются максимальными среди всех кластеров.

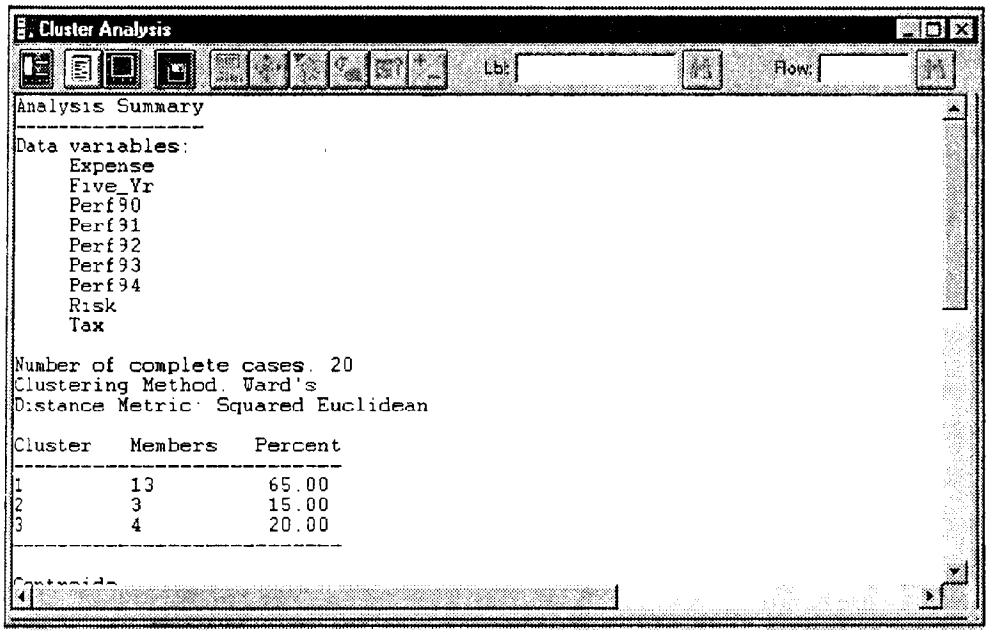


Рис. 2.56. Сводка кластерного анализа (верхняя часть)

О третьем кластере можно сказать, что он занимает второе место по расходам относительно к доходам за пятилетний период. Оценка риска была самая высокая, однако налоговые сборы существенно ниже, чем у первого кластера.

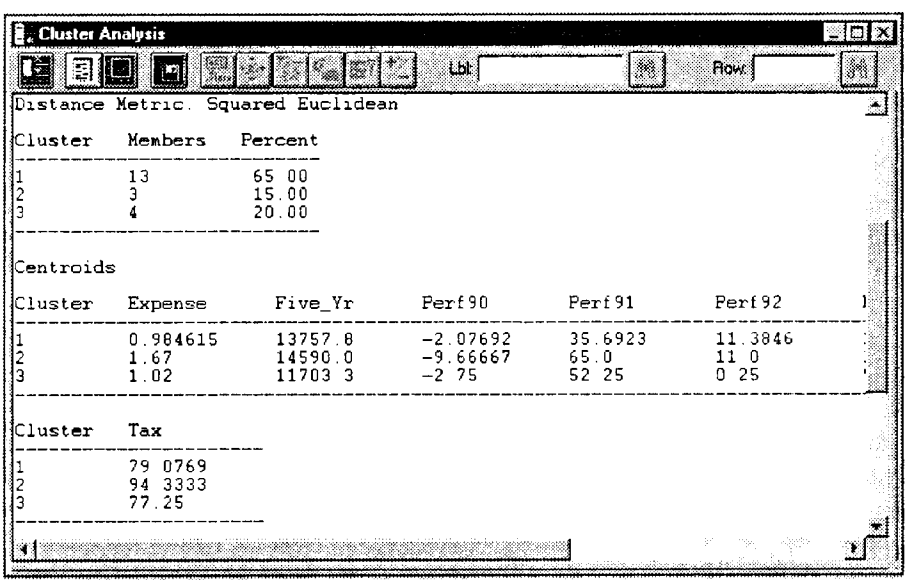


Рис. 2.57. Сводка кластерного анализа (нижняя часть)

Нажмем кнопку табличных опций (вторая слева в верхнем ряду). Система предоставит соответствующее окно диалога.

Установим Membership Table (таблица принадлежности наблюдений), затем нажмем кнопку OK.

Дважды щелкнем левой кнопкой мыши на таблице населенности для максимального раскрытия окна.

В данной таблице описаны выбранные параметры кластерного анализа и затем дается полный список всех наблюдений, их имена и номера кластеров, в которые входят указанные наблюдения (рис. 2.58).

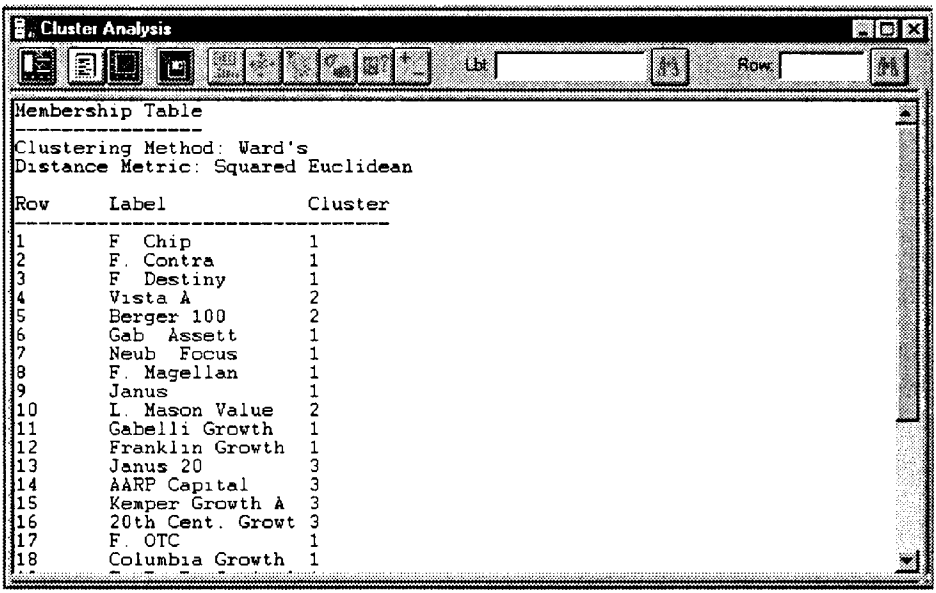


Рис. 2.58. Таблица принадлежности наблюдений к кластерам

Создание двухмерной диаграммы рассеивания

Нажмем кнопку графических опций (третью слева в верхней части окна анализа). Появится окно диалога для задания соответствующих параметров.

Установим флажок 2D Scatterplot (двухмерная диаграмма рассеивания).

Нажмем кнопку OK — система отобразит еще одно графическое окно.

Дважды щелкнем левой кнопкой мыши на окне дендрограммы, чтобы развернуть его.

На дендрограмме видны три дерева (рис. 2.59). По вертикальной оси отложено расстояние для каждого шага работы агломеративного иерархического алгоритма кластеризации. На горизонтальной оси показаны наблюдения, скомбинированные в соответствии с проведенным анализом. Дендрограмма позволяет увидеть отчетливую картину трех группировок и имена наблюдений (инвестиционных фондов), вошедших в выделенные кластеры.

Дважды щелкнем на дендрограмме и тем самым вновь минимизируем ее.

Дважды щелкнем левой кнопкой мыши на двухмерной диаграмме рассеивания (рис. 2.60).

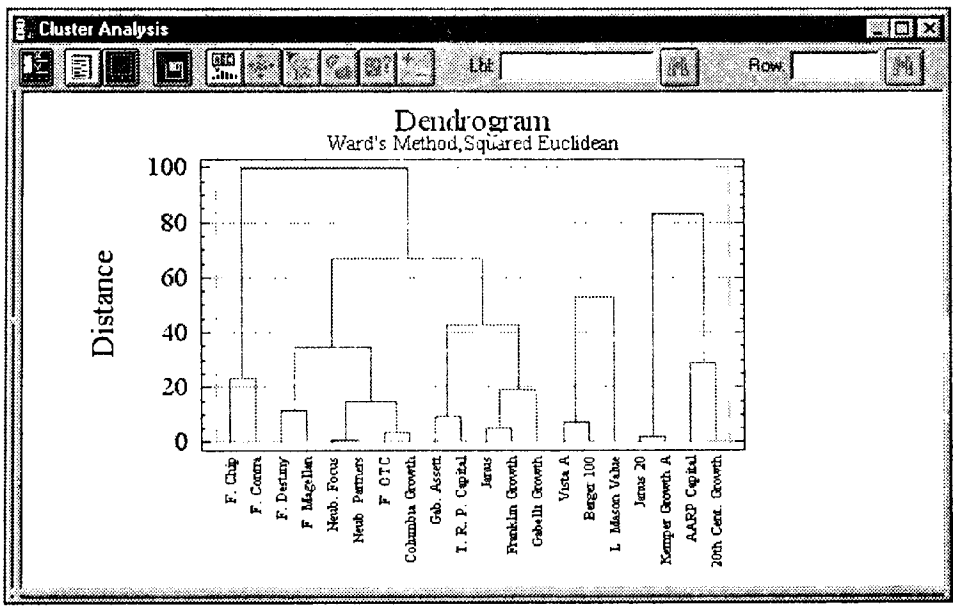


Рис. 2.59. Дендрограмма для трех кластеров

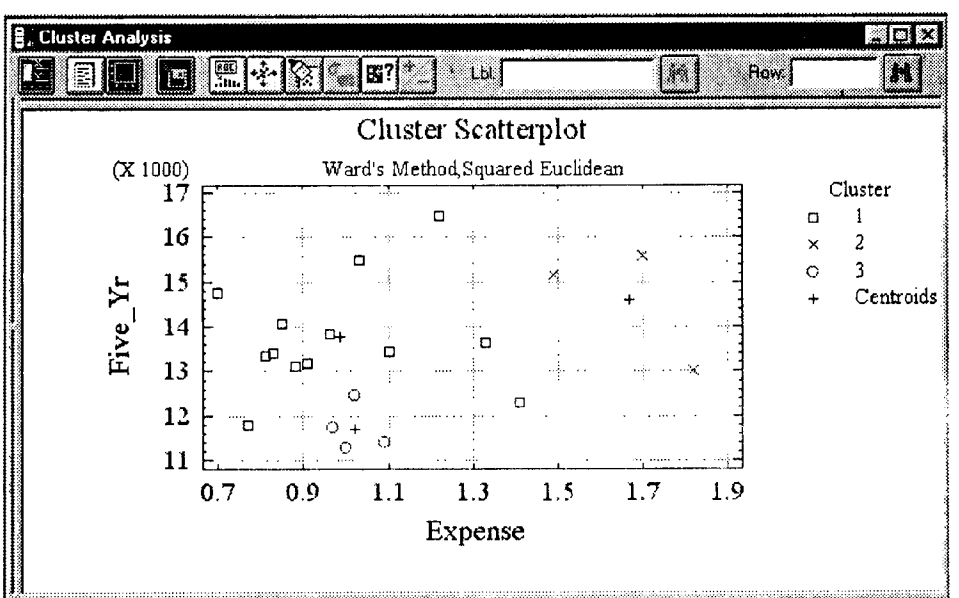


Рис. 2.60. Двухмерная диаграмма рассеивания

Диаграмма показывает, как группируются исследуемые наблюдения на плоскости двух переменных **Expence** и **Five_Yr** (рис. 2.60). Каждый кластер представлен на диаграмме собственным символом, а если бы это было в цветном исполнении, то и цветом. Из графика следует, что первый кластер имеет низкие относительные расходы; видно, как распределены доходы фондов за пятилетний период. В кластере 2 наблюдаются самые высокие расходы, но и максимальные пятилетние доходы. В кластере 3 низкие расходы сопровождаются и невысокими пятилетними доходами.

Для того чтобы отобразить другие диаграммы рассеивания, достаточно щелкнуть правой кнопкой мыши и, получив в распоряжение соответствующее окно диалога, выбрать интересующие пары переменных.

Литература

1. Cover T., Hart P. Nearest neighbour pattern classification//IEEE Trans. Inform. Theory. 1967. Vol. IT–13, P. 21–27.
2. Fix E., Hodges J. L. Discriminatory analysis, nonparametric discrimination USA School of Medicine. — Texas: Rendolph Field, 1951–1952.
3. Fridlund A.J. CTI Catalogue of Economics Software: STATISTICAL ANALYSIS. P. 21.
4. Fridlund A. J. Powerful SYSTAT Limited by Outdated Interface.//InfoWorld. 1995. Vol. 17, № 40. P. 99.
5. Fridlund A. J., Sophisticated STATISTICA Is a Slick Jack-of-all-trades.//InfoWorld. 1995. 30-th Oct. P. 106.
6. Morrison D. F. Multivariate Statistical Methods. — New York: McGraw-Hill, 1990.
7. Sammon J. W. A nonlinear mapping for Data Structure Analysis//IEEE Trans. Comput 1969. — Vol. C—18. № 5. P. 401–409.
8. Schervish M. J. MINITAB//CHANCE: New Directions for Statistics and Computing. 1993. vol. 6, № 1. P. 54–61.
9. SciTECH: Software for Science. Vol. 27.
10. Stein P. G., Matey J. R., Pitts K. A. Review of Statistical Software for the Apple Macintosh. — The American Statistician, 1997. Vol. 51, № 1. P. 67–82.
11. Torgerson W. S., Multidimensional Scaling. Theory and Method//Psychometrika, 1952.Vol. 17, № 4.
12. Wass J. A. How Statistical Software Can Be Assessed. — Scientific Computing & Automation. 1996 (October). P. 14–24.
13. Айвазян С. А., Бежаева З. И., Староверов О. В. Классификация многомерных наблюдений. — М.: Статистика, 1974.
14. Айвазян С. А., Бухштабер В. М. Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности. — М.: Финансы и статистика, 1989.

15. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Статистическое оценивание зависимостей. — М.: Финансы и статистика, 1985.
16. Александров В. В., Алексеев А. И., Горский Н. Д., Анализ данных на ЭВМ (на примере системы СИТО). — М.: Финансы и статистика, 1990.
17. Александров В. В., Лачинов В. И., Поляков А. О. Рекурсивная алгоритмизация кривой, заполняющей многомерный интервал//Изв. АН СССР. «Техн. кибернетика», 1978. № 1. С. 192–197.
18. Андерсон Т. Введение в многомерный статистический анализ. — М.: Физматгиз, 1963.
19. Аркадьев А. Г., Браверман Э. М. Обучение машины классификации объектов. — М.: Наука, 1971.
20. Боровиков В. П., Популярное введение в программу STATISTICA, Компьютер Пресс, 1998.
21. Векслер Л. С., Статистический анализ на персональном компьютере//Мир ПК. 1992. № 2.
22. Горский Н. Д. Рекурсивный метод отображения многомерного пространства при решении задач хранения и обработки данных в автоматизированных системах научных исследований. — Автореф. на соиск. уч. степ. канд. техн. наук./ ЛИАН. Л., 1981.
23. Григорьев С. Г., Перфилов А. М., Левандовский В. В., Юнкеров В. И. STATGRAPHICS на персональном компьютере. — СПб, 1992.
24. Демиденко Е. З. Линейная и нелинейная регрессия. — М.: Финансы и статистика, 1981.
25. Дуда Р., Харт П. Распознавание образов и анализ сцен. — М.: Мир, 1976.
26. Дэйвисон М. Многомерное шкалирование. Методы наглядного представления данных. — М.: Финансы и статистика, 1988.
27. Дюк В. А. Обработка данных на ПК в примерах. — СПб.: Питер, 1997.
28. Дюк В. А. Мирошников А. И. Эволюция STATGRAPHICS//Мир ПК. 1995. № 12.
29. Дюк В. А. Компьютерная психодиагностика. — СПб.: Питер, 1994.
30. Енюков И. С. Методы, алгоритмы, программы многомерного статистического анализа: Пакет ППСА. — М.: Финансы и статистика, 1986.
31. Журавлев Ю. И., Гуревич И. Б. Распознавание образов и анализ изображений/ Искусственный интеллект.: В 3 кн. Кн. 2. Модели и методы: Справ./Под ред. Д. А. Поспелова. — М.: Радио и связь, 1990.
32. Загоруйко Н. Г., Елкина В. Н., Лбов Г. С. Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск: Наука, 1985.
33. Информатика в статистике: Словарь-справочник. — М.: Финансы и статистика, 1994.
34. Классификация и кластер/Под ред. Дж. Вэн Райзин. — М.: Мир, 1980.
35. Кулаичев А. П. Пакеты для анализа данных//Мир ПК. 1995. № 1.

36. Кулаичев А. П. Средства и программные системы анализа данных//Мир ПК. 1994. № 10.

37. Лоули Д., Максвелл А. Факторный анализ как статистический метод. — М.: Мир, 1967.

38. Макаров А. А. STADIA против STATGRAPHICS, или Кто ваш лоцман в море статистических данных//Мир ПК. 1992. № 3.

39. Международная конференция «Статистическое образование в современном мире: идеи, ориентации, технологии», 3—5 июля 1996. Тез. докл. — СПб.: Изд-во СПбУЭФ, 1996.

40. Миркин Б. Г. Анализ качественных признаков и структур. — М.: Статистика, 1980.

41. Налимов В. В. Теория эксперимента. — М.: Наука, 1971.

42. Попечителев Е. П., Романов С. В. Анализ числовых таблиц в биотехнических системах обработки экспериментальных данных. — Л.: Наука, 1985.

43. Пфанцагль И. Теория измерений. — М.: Мир, 1976.

44. Статистические и математические системы//Тысячи программных продуктов: Каталог. 1995. № 2. С. 88–92.

45. Терехина А. Ю. Анализ данных методами многомерного шкалирования. — М.: Наука, 1986.

46. Ту Дж., Гонсалес Р. Принципы распознавания образов. — М.: Мир, 1978.

47. Тьюки Дж. У. Анализ результатов наблюдений/Пер. с англ. — М.: Мир, 1981.

48. Тьюки Дж. Анализ результатов наблюдений. Разведочный анализ. — М.: Мир, 1981.

49. Тюрин Ю. Н., Макаров А. А. Анализ данных на компьютере. — М.: ИНФРА-М, Финансы и статистика, 1995.

50. Тюрин Ю. Н., Макаров А. А. Анализ данных на компьютере. — М.: ИНФРА-М, Финансы и статистика, 1997.

51. Шанчев Р. SPSS-7.5 прокладывает курс в океане данных.//PC Week. 1997. № 12 (86). С. 6.

52. White B. SYSTAT //Economic Journal. 1992. Vol. 102. № 415.

53. SciTECH: Software for Science. — Vol. 33.

3 ГЛАВА Нейросетевое представление неизвестных знаний и закономерностей[1]

Искусственные нейронные сети (ИНС) и **нейрокомпьютеры** (НК) являются основным средством обработки информации в нейроинформатике. По своей структуре и функционированию они являются искусственными аналогами биологических нейронных систем человека и животных. ИНС представляют новую парадигму обработки информации, базирующуюся на той или иной упрощенной математической модели биологических нейронных систем.

Попытки реализации нейронных сетей для вычислений, имитирующих интеллектуальные процессы в человеческом мозге, такие как распознавание образов, принятие решений, управление движением и многие другие, стали возможны с появлением первых компьютеров. Можно сказать, что они, в некотором смысле, стимулировали их появление. В то же время, ИНС представляют новый подход в методологии вычислений, отличающийся последовательной архитектурой от вычислений на цифровых компьютерах с традиционной фон-неймановской архитектурой.

Цифровые компьютеры функционируют под управлением программы, точно заданной последовательности операций, и запоминают информацию в специально организованной памяти. Весь процесс вычислений организуется сложными процессорными устройствами.

ИНС организуют свою работу путем распределения процесса обработки информации между простыми локальными процессорными элементами — нейроэлементами, связанными между собой посредством специальных соединений — синаптических связей. Запоминаемая информация распределяется по сети в виде весовых параметров этих соединений, а развитие возможностей ИНС осуществляется не путем программирования, как в цифровых компьютерах, а путем обучения ИНС.

---

[1] Раздел подготовлен д-ром техн. наук А. В. Тимофеевым, канд. техн. наук А. А. Богдановым и канд. физ.-мат. наук З. М. Шибзуховым.

Первоначально ИНС развивались для исследования и воспроизведения таких видов человеческой деятельности по обработке информации, как речь, зрение и обработка знаний. Однако они оказались эффективными также в задачах классификации и приближения функций, восстановления скрытых закономерностей по экспериментальным базам данных. ИНС показали высокий потенциал при решении таких сложных задач, как оптимальное управление и адаптивная идентификация систем, сжатие данных, распознавание образов и диагностика состояний.

Существует много различных точек зрения на технологию ИНС. Часто их называют конекционистскими системами, акцентируя внимание на большом числе настраиваемых связей между нейропроцессорными элементами. Иногда ИНС относят к адаптивным системам или самообучающим системам вследствие способности НС адаптировать значения весовых параметров в процессе обучения. ИНС называют также параллельными распределенными системами, отражая способ организации обработки информации между процессорными элементами. Но в целом они обладают всеми перечисленными свойствами.

Теория нейронных сетей развивается на стыке различных дисциплин: нейрофизиологии, математического моделирования, информатики и искусственного интеллекта, а также физики, психологии и лингвистики. В целом все эти теории в конечном итоге направлены на создание искусственных интеллектуальных систем, имитирующих интеллектуальную деятельность и мышление человека.

Структура искусственных нейронных сетей

Общепринятое представление об ИНС как об обучаемой или самообучающейся системе обработки информации сформировалось под влиянием теории параллельных и распределенных вычислений, а также теории адаптивных систем.

В *параллельных* системах в любой момент времени в активном состоянии могут находиться несколько процессов. В *распределенных* системах каждый из процессов может независимо обрабатывать локальные данные и принимать решения, а отдельные процессы обмениваются информацией между собой и внешней информационной средой через каналы связей. В *адаптивных* системах процессы обработки информации организуются таким образом, чтобы достигать требуемых целей при неопределенных факторах; причем при изменении последних система способна адаптироваться.

Таким образом, ИНС представляет собой *параллельную*, *распределенную*, *адаптивную* систему, которая восстанавливает скрытые закономерности и развивает свои способности по обработке информации в результате *обучения*.

Она обладает следующими отличительными особенностями:

- ИНС состоит из простых **нейропроцессорных элементов** (НЭ), или нейроэлементов, являющихся искусственными аналогами биологических нейронов;
- НЭ связаны между собой направленными **информационными каналами** (ИК), по которым распространяются информационные сигналы, закодированные в скалярной форме;

- каждый НЭ может быть связан посредством входных ИК с множеством других НЭ;
- каждый НЭ имеет единственный выходной ИК, который впоследствии может разветвляться;
- каждый НЭ может обладать собственной внутренней памятью (в основном в виде весовых параметров соединений) и может осуществлять локальную обработку приходящей к нему информации;
- обработка информации нейроэлементом осуществляется локально: она зависит только от значений, поступающих по входным ИК, и значений, хранящихся в его внутренней памяти.

В противоположность традиционным структурам обработки информации, которые развиваются путем прямого программирования, ИНС развиваются и адаптируются в процессе обучения по примерам. Технология обучения подразделяется на две категории:

- *обучение с учителем*, при котором имеется множество примеров, для которых отклик или поведение НС известно;
- *обучение без учителя*, *самообучение* или *самоорганизация*, при котором процесс обучения НС происходит автономно: по мере поступления новой информации находятся некоторые ее свойства и закономерности, и НС обучается отражать их на выходе.

Нейропроцессорные элементы

Каждый НЭ выполняет относительно простую функцию: получает информационные сигналы от других НЭ или внешних источников, формирует выходной информационный сигнал и передает его другим.

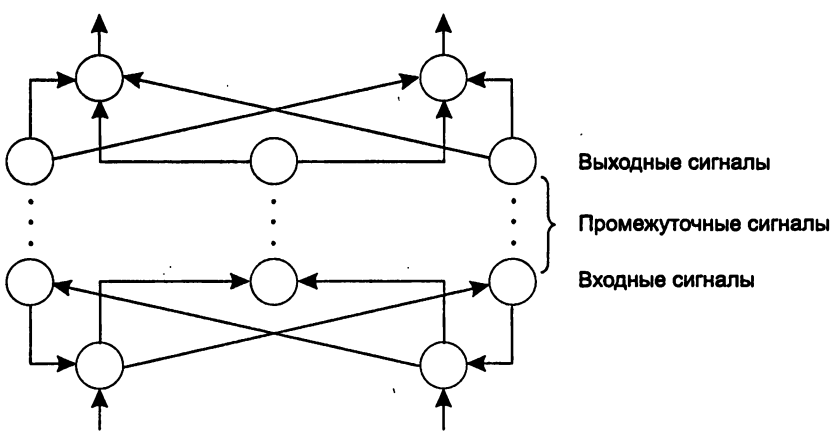


Рис. 3.1. Общая схема многослойной искусственной нейронной сети

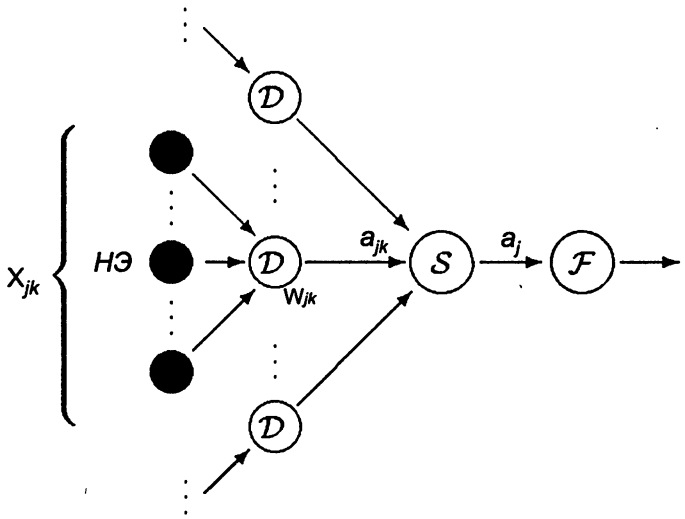


Рис. 3.2. Схема преобразования сигналов в НЭ

Внутри многослойной ИНС нейропроцессорные элементы подразделяются на три типа (рис. 3.1):

- входные, которые получают сигналы от внешних источников;
- промежуточные, которые принимают сигналы от одних НЭ, формируют сигналы и передают их другим;
- выходные, которые формируют выходные сигналы.

Для описания функционирования НЭ в ИНС необходимо ввести ряд характеристик и правил (рис. 3.2):

- каждый j-й НЭ характеризуется скалярной величиной a_j, называемой активацией, которая соответствует уровню его активности;
- передача сигналов от одних НЭ к другим осуществляется посредством взвешенных соединений с настраиваемыми весовыми параметрами;
- каждое k-е соединение характеризуется своей величиной вектора весов $\mathbf{w}_{jk}$;
- взвешенное *дендритное преобразование* $a_{jk} = D(\mathbf{w}_{jk}; \mathbf{x}_{jk})$ определяет величину активации a_{jk}, производимой k-м соединением по вектору $\mathbf{x}_{jk}$ выходных сигналов НЭ, участвующих в его образовании;
- правило суммирования активаций $a_j = S(..., a_{jk}, ...)$ определяет величину a_j активации j-го НЭ в зависимости от активаций a_{jk}, производимых взвешенными соединениями, где индекс k пробегает все соединения j-го НЭ;
- функция выхода $x_j = F(a_j)$ определяет значение выходного сигнала x_j НЭ в зависимости от величины его активации a_j.

Функциональную зависимость $a_j = A_j(\mathbf{x}_j)$ величины активации a_j от вектора $\mathbf{x}_j$ значений выходных сигналов НЭ, связанных с j-м НЭ посредством соединений, называют *функцией активации* НЭ. Функциональную зависимость $x_j = N_j(\mathbf{x}_j)$ величины выходного сигнала x_j НЭ от вектора $\mathbf{x}_j$ значений выходных сигналов НЭ, связанных с j-м НЭ посредством соединений, называют *функцией преобразования* НЭ.

В динамических моделях функционирования ИНС все величины — x_i, x_j, $\mathbf{x}_{jk}$, $\mathbf{w}_{jk}$, a_{jk}, a_j — являются функциями, зависящими от дискретного или непрерывного фактора времени t. В этом случае функция выхода может также зависеть от значений активации a_j и выходного значения x_j в моменты времени $\tau < t$.

В процессе функционирования ИНС нейроэлементы могут изменять свое состояние синхронно или асинхронно:

- при синхронном функционировании нейроэлементы изменяют свое состояние активации одновременно;
- при асинхронном функционировании нейроэлементы могут изменять свое состояние не одновременно, причем только один нейроэлемент может изменить свое состояние активации в одно время.

В некоторых случаях асинхронная модель функционирования нейроэлементов может иметь преимущества.

Функции активации нейроэлементов

Правила суммирования активаций

В большинстве случаев предполагают, что величина активации НЭ может быть вычислена как результат простого арифметического суммирования величин активации каждого соединения, то есть преобразование S имеет вид

$$a_j = \sum_k a_{jk} + \theta_j ,$$

где индекс k пробегает соединения НЭ, θ_j — величина так называемого смещения или порога.

Функция активации с таким правилом суммирования принимает вид

$$a_j(\vec{x}_j) = \sum_k D(\vec{x}_{jk}; \vec{w}_{jk}) + \theta_j ,$$

где $\vec{x}_j$ — вектор выходных сигналов НЭ, связанных с j-м НЭ посредством взвешенных соединений, а D — дендритная функция активации.

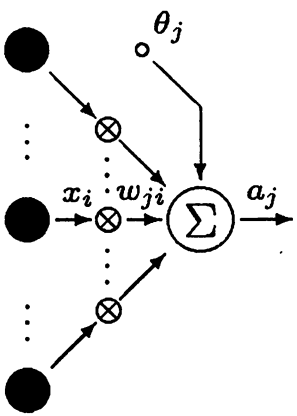
Линейная функция активации

В большинстве случаев предполагают, что каждый нейроэлемент, от которого поступает информационный сигнал, вносит линейный вклад в a_j.

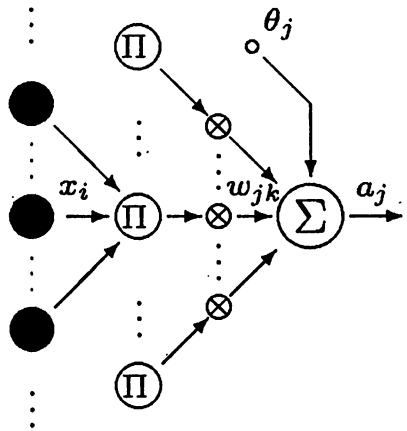
В простейшем случае величина активации j-го НЭ вычисляется как взвешенная сумма значений выходных сигналов от других НЭ и значения смещения:

$$a_j = \sum_i w_{ji} x_i + \theta_j,$$

где i — индексы НЭ, от которых посредством синаптических связей с весами w_{ji} соответственно, поступают информационные сигналы к j-му НЭ. Если $w_{ji} > 0$, то входной сигнал от j-го НЭ является возбуждающим; если $w_{ji} < 0$ — тормозящим. НЭ с линейным правилом распространения сигналов называются **Σ-элементами**. Исторически это самое первое и самое распространенное правило.

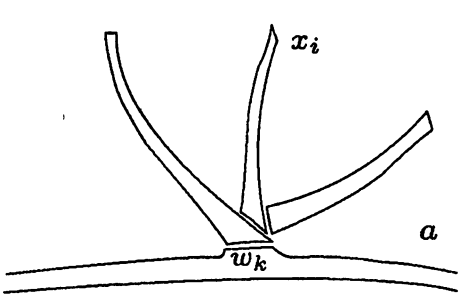


Полилинейная функция активации



Линейная функция активации

Рис. 3.3. Правила распространения сигналов для Σ– и Σ-Π-элементов



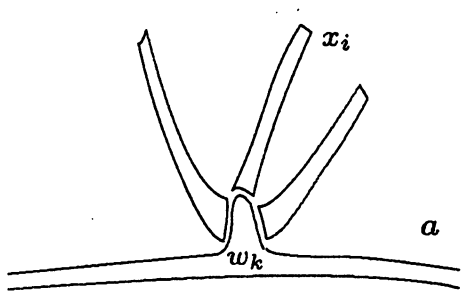


Рис. 3.4. Сложные синаптические соединения

Полилинейная и полиномиальная функции активации

Σ-правило соответствует предположению о том, что каждый НЭ, от которого поступает информационный сигнал, вносит линейный и независимый вклад в величину активации. Это бывает в случае простых синаптических связей. Однако вклад нейроэлементов, участвующих в образовании более сложных связей (пресинаптические соединения, синаптические кластеры, несинаптические соединения в пучках аксонов и дендритов) (рис. 3.4), уже не может быть независимым.

В простейшем случае можно предположить, что каждый нейроэлемент при неизменной активности остальных нейроэлементов, участвующих в образовании синаптических связей, вносит независимый линейный вклад в величину активации. Такому предположению соответствует полилинейное правило, известное как правило распространения сигналов для **Σ-П-элементов**:

$$a_j = \sum_i w_{jk} \prod_{i \in I_{ik}} x_i + \theta_j$$

или

$$a_j = \sum_i w_{jk} \prod_{i \in I_{ik}} (x_i - d_{jki}) + \theta_j,$$

где индекс k пробегает синаптические соединения, индексы $i \in I_{ik}$ — различны и пробегают НЭ, участвующие в текущем k-м соединении. Таким образом, Σ-П-элементы описываются полилинейными отображениями. Если предположить, что некоторые входы Σ-П-элемента могут повторяться, то он будет описываться уже полиномиальным отображением, а правило распространения будет называться *полиномиальным правилом.*

Хотя полилинейное и полиномиальное правила распространения в литературе по нейроинформатике встречаются реже, чем линейное правило, однако биологически и математически более оправдано применение именно таких правил, а в ряде случаев их применение дает более эффективные результаты.

Функции выхода

Нелокализованные функции выхода

Первые ИНС, предложенные еще в 40-х годах Мак-Каллоком и Питсом в 1956 году и основанные на логических НЭ порогового типа, использовали в качестве функции выхода функцию Хевисайда:

$$\theta(x) = \begin{cases} 1, \text{ если } x > 0, \\ 0, \text{ если } x \le 0. \end{cases}$$

Использование такой функции было связано с развитием логического анализа вычислительных сетей и попытками представить человеческий мозг в виде компьютера. В принципе, произвольные вычисления действительно можно организовать с использованием только логических нейронов. Вещественные значения

можно представлять последовательностями битов, а используемые для этих целей логические НЭ можно обучать. Главными достоинствами логических НЭ являются высокая скорость вычислений и относительная простота реализации некоторых функций. В ряде случаев вместо функции $\theta(x)$ используется функция

$$\gamma(x) = \begin{cases} 1, \text{ если } x > 1, \\ x, \text{ если } 0 \le x \le 1, \\ 0, \text{ если } x \le 0. \end{cases}$$

Позднее в качестве функции выхода стали использовать *сигмоидальные функции*:

$$\sigma(x;\ s) = \frac{1}{1 + e^{-x/s}}.$$

Константа s определяет величину наклона сигмоидальной функции в окололинейной области. Функция $\sigma(x;\ s)$ принимает значения в интервале (0, 1), поэтому в ряде случаев она заменяется функциями $\operatorname{arctg}(x/s)$ и $\operatorname{th}(x;\ s)$, принимающими значения в интервале (−1, 1):

$$\operatorname{th}(x;\ s) = \frac{1 - e^{-x/s}}{1 + e^{-x/s}}.$$

Для увеличения скорости вычислений можно использовать другие функции:

$$s_1(x;\ s) = \frac{x}{|x| + s},$$

$$s_2(x;\ s) = \frac{\sqrt{1 + s^2 x^2} - 1}{sx}.$$

Сигмоидальные функции имеют нелокальное поведение: они отличны от нуля в бесконечной области. Для них известны результаты о существовании универсального нейросетевого аппроксиматора, содержащего единственный слой сигмоидальных НЭ.

Другой класс функций, используемый в теории аппроксимации, включает в себя так называемые радиальные функции. Среди них есть нелокализованные функции, однако большинство из них — локализованные. ИНС, построенные с использованием радиальных функций, тоже являются универсальными аппроксиматорами.

Локализованные функции выхода

В ряде случаев в адаптивных системах, играющих роль классификаторов, более предпочтительными на роль функций выхода оказывались локализованные функции. Попытки использования локализованных функций в адаптивных системах, в частности, для распознавания образов, предпринимались давно. Локализованные функции использовались для построения вещественнозначных отображений и обучения классификаторов.

Наиболее простыми с вычислительной точки зрения являются следующие функции:

$$g_2(x; t, s) = \frac{1}{1 + \|x - t\|^2 / s^2},$$

$$g_4(x; t, s) = \frac{1}{1 + \|x - t\|^4 / s^2}.$$

Радиальные функции

Радиальные функции в качестве функций выхода использовались во многих моделях ИНС. Функции данного типа также использовались в теории аппроксимации и в распознавании образов, правда,. под другими именами (потенциальные функции). Существует несколько типов локализованных радиальных функций. Среди них — гауссовы функции вида

$$h_1(x; t, b) = e^{-\|x - t\|2/b},$$

а также радиальная координатная, мультиквадратичные и тонколистные сплайн-функции:

$$h_2(x; t) = \|x - t\|^2;$$
$$h_3(x; t, b) = (b^2 + \|x - t\|^2)^{-a},\ a > 0;$$
$$h_4(x; t, b) = (b^2 + \|x - t\|^2)^{b},\ 0 < b < 1;$$
$$h_5(x; t, b) = (b\|x - t\|)^2 \ln(b\|x - t\|).$$

В большинстве случаев функция выхода F является неубывающей функцией. Чаще всего используются следующие функции:

$$\text{thereshold}(x) = \begin{cases} 1, \text{если } x \geq 0, \\ 0, \text{если } x < 0; \end{cases}$$

$$\text{sigmoidal}(x; w) = \frac{1}{1 + e^{-wx}};$$

$$\text{ramp}(x) = \begin{cases} 1, \text{если } x \geq 1, \\ x, \text{если } 0 < x < 1, \\ 0, \text{если } x < 0; \end{cases}$$

$$\text{sign} = \begin{cases} 1, \text{если } x > 0, \\ 0, \text{если } x = 0, \\ -1, \text{если } x < 0; \end{cases}$$

$$\text{th}(x;\, w) = \frac{1 - e^{-wx}}{1 + e^{-wx}};$$

$$s_1(x;\, w) = \frac{x}{|x| + w};$$

$$s_2(x;\, w) = \frac{\sqrt{1 + w^2 x^2} - 1}{wx}.$$

Получили также распространение следующие функции выхода:

$$\text{Gaussian}_1(x, x_0, w) = e^{(x - x_0)^2 / w^2};$$

$$\text{Gaussian}_2(x, x_0, \alpha) = \frac{1}{1 + (x - x_0)^{\alpha / w^2}};$$

$$\text{BiRadial}(x; x_0; w_1; \vec{b}_1; \vec{s}) = \prod_{i=1}^{N} \text{sigmoidal}(e^{A_i}(x - x_0 + e^{b_i}); w) \times$$

$$\times (1 - \text{sigmoidal}(e^{A_i}(x - x_0 + e^{b_i} e; w).$$

Топология нейронных сетей

В зависимости от топологии соединений нейроэлементов нейронные сети подразделяются на две основные категории.

Нейронные сети прямого распространения

Структура соединений нейроэлементов в таких ИНС такова, что в ней нет обратных связей, то есть выходной сигнал нейроэлемента не может в последующем возвратиться ему на вход.

Рекуррентные нейронные сети

Для сетей подобного вида выходные сигналы нейроэлемента могут в последующем возвратиться ему на вход, то есть в ИНС имеются обратные связи (или циклы).

Архитектура различных искусственных нейронных сетей

Элементарный перцептрон Ф. Розенблатта

В 1957 году американским нейрофизиологом Ф. Розенблаттом была предложена одна из первых моделей ИНС, названная **перцептроном** [2]. Ее отличительная черта — способность обучаться распознаванию простых зрительных образов. Вскоре перцептрон был реализован в виде модели НС Mark 1. С тех пор началось активное развитие нейрокомпьютеров.

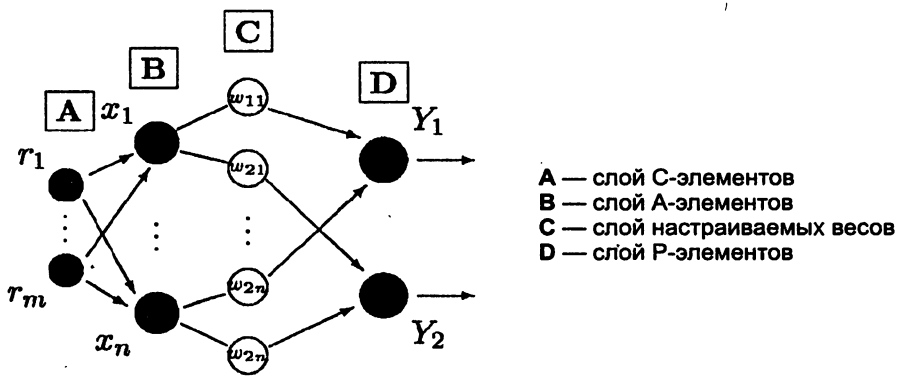


Рис. 3.5. Архитектура перцептрона Ф. Розенблатта

Перцептрон (рис. 3.5) состоит из детерминированных *пороговых* НЭ трех типов:

- *сенсорные* НЭ (С-элементы) $r_1, \ldots, r_n$ («сетчатка»), которые воспринимают входные образы;
- *ассоциативные* НЭ (А-элементы) $x_1, \ldots, x_n$, связанные с сенсорными НЭ, выполняют роль детекторов признаков;
- *решающие* НЭ (Р-элементы), связанные настраиваемыми связями с ассоциативными НЭ, служат для принятия решений.

Обычно число Р-элементов выбирают равным количеству классов, на которое необходимо разбить предъявляемые перцептрону образы.

Рассмотренная модель *ИНС* относится к классу сетей *с прямыми связями* (feedforward), один из слоев которых является модифицируемым.

В простейшем случае, когда $n=m$ и $x_i = r_i$, $i = 1 \ldots n$, детекторы признаков могут рассматриваться как входной слой. Тогда перцептрон вырождается в один *пороговый* НЭ с m входами, называемый **элементарным перцептроном**.

Перцептрон реагирует на входной образ (вектор) генерацией сигнала на выходе Р-элемента. Он функционирует в двух режимах:

- в *режиме обучения* перцептрона в пространстве признаков x_1, ..., x_m формируется гиперплоскость, которая делит это пространство на два класса;
- в *режиме распознавания* перцептрон определяет, к какому из классов относится входной образ. Так производится бинарная классификация входных образов.

Решение перцептроном конкретной задачи распознавания обеспечивается с помощью *настройки весов связей* в режиме обучения.

Основными недостатками элементарного перцептрона являются:

- ограниченность круга задач, решаемых этой ИНС (из-за линейности разделяющей классы поверхности);
- большая длительность процесса обучения (из-за необходимости многократного предъявления всей обучающей выборки).

Многослойный перцептрон

Многослойный перцептрон — это ИНС с прямыми связями (feedforward), в которой имеется несколько слоев НЭ с настраиваемыми весами связей.

Архитектура трехслойного перцептрона, получившего наибольшее распространение, приведена на рис. 3.6.

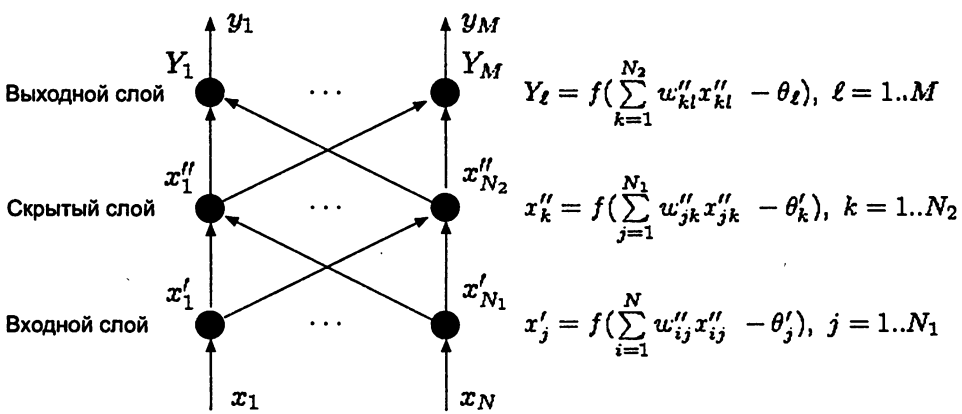
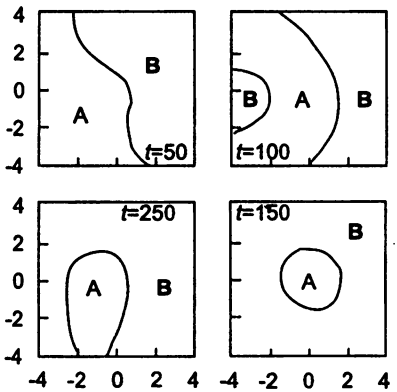


Рис. 3.6. Архитектура трехслойного перцептрона



Входные сигналы, соответствующие классам ***A*** и ***B***, попеременно подавались на вход перцептона. Элементы класса ***A*** равномерно распределены внутри окружности, а элементы класса ***B*** расположены вне ее

Рис. 3.7. Пример области, описываемой трехслойным перцептроном

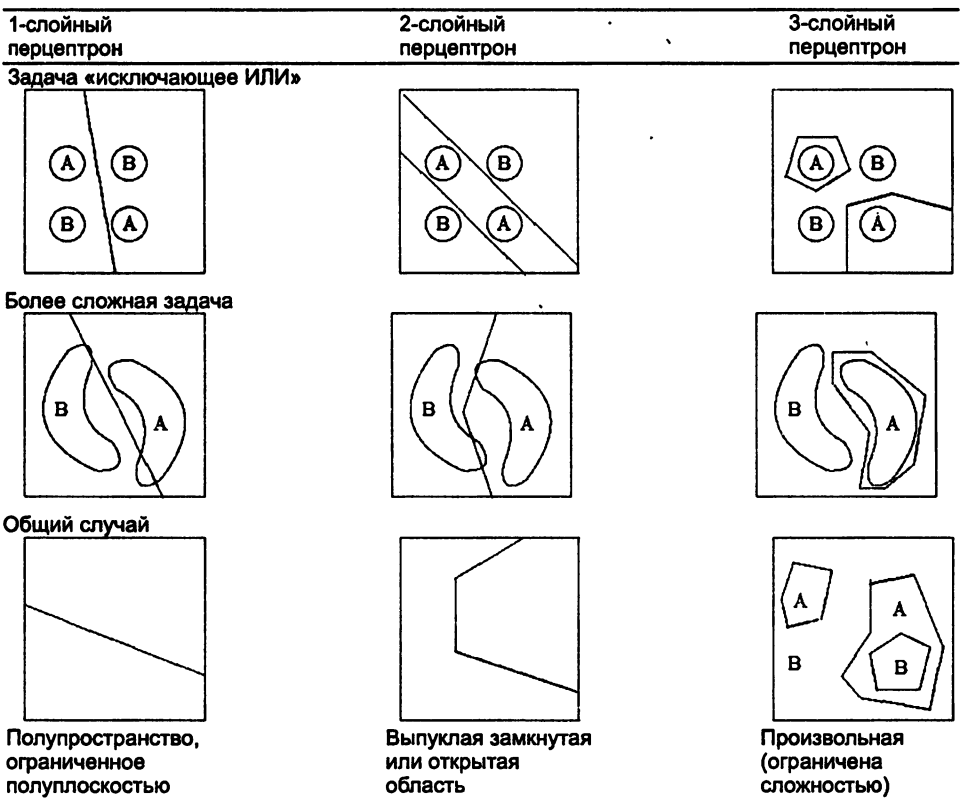


Рис. 3.8. Вид областей решений для перцептронов

В отличие от элементарного перцептрона, формирующего классифицирующую гиперплоскость, многослойный перцептрон может формировать в режиме обучения сложные нелинейные гиперповерхности (кусочно-линейные, полиномиальные и т. п.) (рис. 3.7). Благодаря этому многослойный перцептрон способен распознавать произвольное число классов, имеющих сложную структуру в пространстве первичных признаков. При этом классы могут представлять собой выпуклые, невыпуклые и многосвязные объекты (рис. 3.8).

Таким образом, многослойный перцептрон с достаточным количеством внутренних НЭ и соответствующей матрицей связей принципиально способен осуществлять любое преобразование <вход-выход>, аппроксимировать любую решающую (распознающую) функцию с любой наперед заданной точностью. Для этого необходимо:

- выбрать архитектуру перцептрона, а именно: число слоев, число НЭ и их тип;
- настроить веса связей НЭ с помощью некоторого алгоритма обучения.

Нейросети Хопфилда

Сети Хопфилда представляют собой широкий класс ИНС, включающий в себя многие типы ИНС в качестве частных подклассов. Это обусловлено тем, что сеть Хопфилда является абсолютно однородной структурой без какой-либо внутренней специализации ее НЭ.

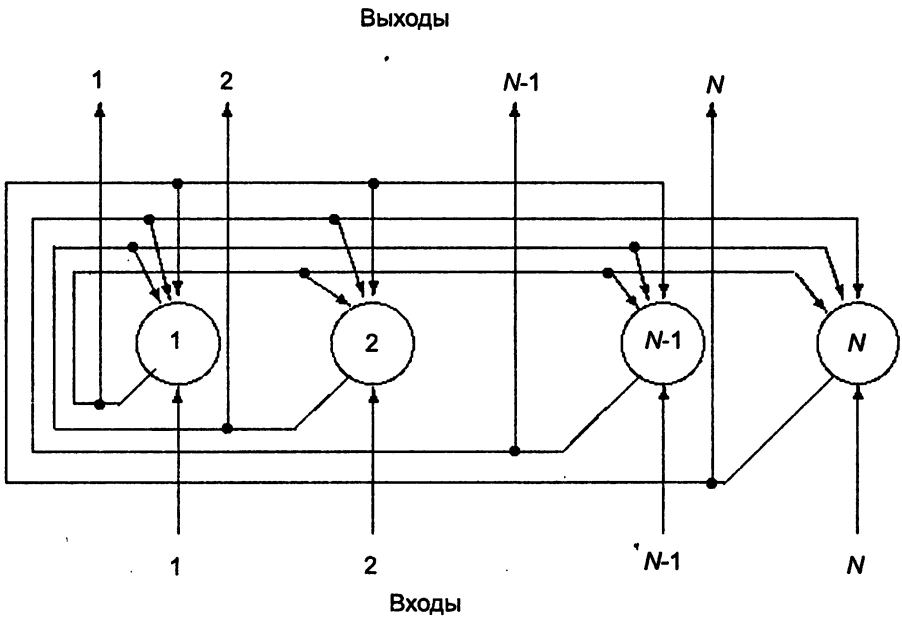


Рис. 3.9. Архитектура ИНС Хопфилда

Выход каждого из НЭ сети связан *настраиваемыми связями* с входами всех других НЭ. Отсутствуют только связи собственного выхода с собственным входом (рис. 3.9).

Сеть Хопфилда предназначена для запоминания нескольких устойчивых выходных состояний (эталонных образов) и может работать в качестве устройства ассоциативной памяти.

Для этого необходимо перед началом работы рассчитать симметричную матрицу связей НЭ сети.

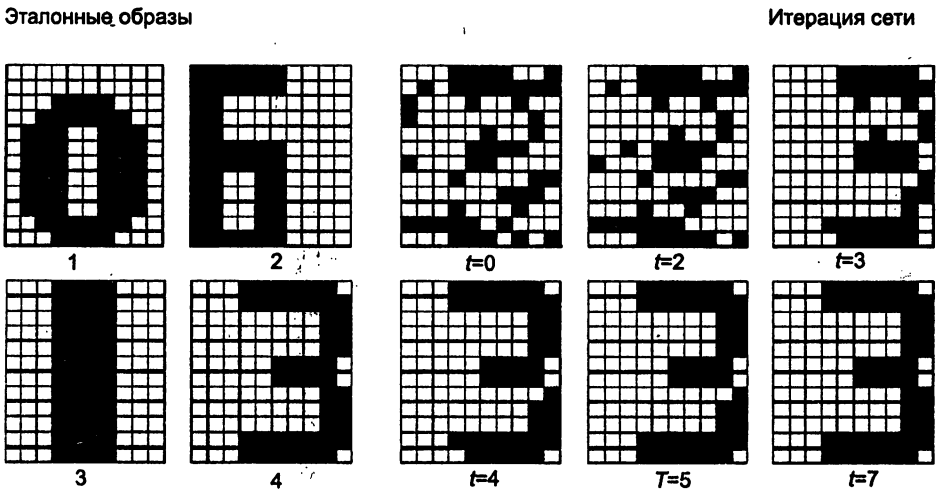


Рис. 3.10. Ассоциативные свойства ИНС Хопфилда

Записывая в сеть некоторый входной образ (например, частично искаженный эталон), можно сформировать правильный выходной образ (эталон без искажений) после достижения сетью стабильного состояния (рис. 3.10).

Разновидности сетей Хопфилда

Существуют две основные разновидности ИНС Хопфилда:

- ИНС с пороговыми НЭ;
- ИНС с сигмоидальными НЭ.

Рис. 3.11. Механическая аналогия поведения ИНС Хопфилда

ИНС с *пороговыми* НЭ функционируют в *асинхронном режиме*, то есть каждый НЭ в случайные моменты времени с некоторой средней частотой определяет свое состояние. В качестве порогового уровня для всех НЭ обычно выбирается ноль.

Благодаря этому оказалось возможным описать динамику ИНС Хопфилда в рамках энергетического подхода. При этом оказывается полезной механическая аналогия между поведением сети и движением шарика по некоторому вязкому рельефу под действием силы тяжести (рис. 3.11).

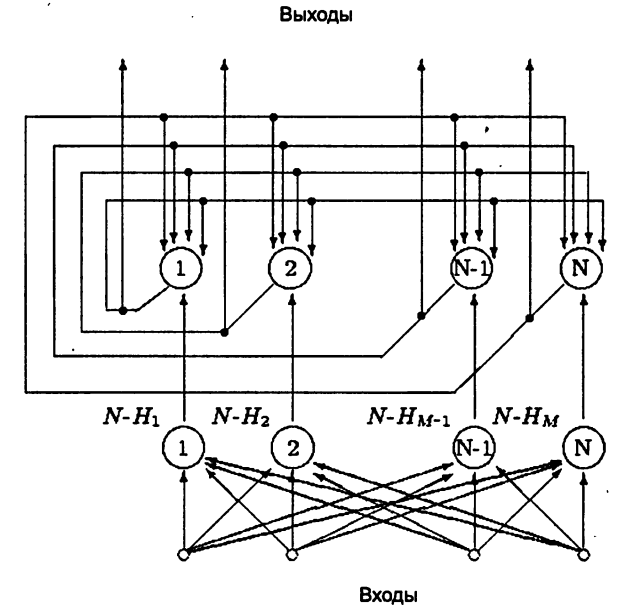
Поведение *ИНС* с *сигмоидными* НЭ описывается системой нелинейных дифференциальных уравнений, учитывающих синхронность и непрерывность переключения НЭ.

Основными недостатками ИНС Хопфилда являются:

1) неэкономное использование памяти (количество случайных образов, которое сеть может запомнить с вероятностью последующего восстановления, равной 1, не превышает 15 % от общего количества НЭ);
2) чувствительность к искажениям распознаваемых образов (сеть имеет весьма скромные возможности распознавания, так как допустимо лишь искажение образов ограниченным аддитивным шумом);
3) неинвариантность к геометрическим преобразованиям образов (например, к аффинным или проективным преобразованиям распознаваемых изображений).

Нейросети Хемминга

ИНС Хемминга реализует оптимальный в смысле минимума ошибки *алгоритм классификации*, используемый для решения задач в области связи (восстановление искаженного случайным шумом эталонного сигнала).



1) H_i – расстояние Хемминга между входным сигналом и i-ым эталоном;

2) вид выходной функции:

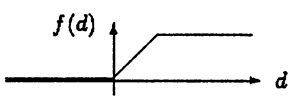


Рис. 3.12. Архитектура ИНС Хемминга

Архитектура ИНС Хемминга показана на рис. 3.12.

Сеть представляет собой двухуровневую структуру, в каждом слое которой расположены НЭ с насыщением. Число НЭ в верхнем и нижнем слоях одинаково и равно числу классов образов. Веса связей и пороги НЭ задаются при обучении ИНС Хемминга.

После задания весов связей и порогов НЭ на входы ИНС подается двоичный N-мерный вектор (образ). Он должен сохраняться на входах в течение времени, достаточного для *параллельного срабатывания* всех НЭ нижнего слоя и инициализации НЭ верхнего слоя. Каждый НЭ нижнего слоя вычисляет следующую величину $N–H_i$, где H_i — расстояние Хемминга между входным вектором и i-м эталоном.

После удаления входного вектора с входов ИНС состояние НЭ нижнего слоя не изменяется, а в верхнем слое происходит *синхронное срабатывание* НЭ.

Верхний слой осуществляет выбор НЭ с наибольшим уровнем возбуждения.

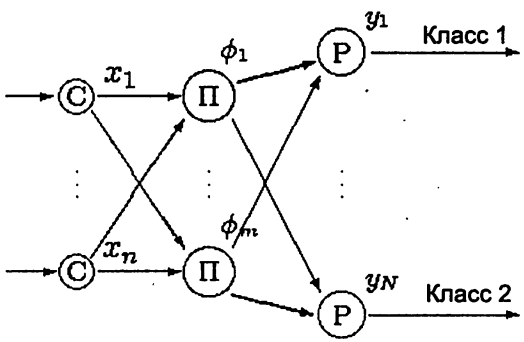
Критерием окончания процесса классификации служит отсутствие выходных сигналов на выходах всех НЭ верхнего слоя, за исключением одного. Номер этого НЭ соответствует классу, к которому принадлежит искаженный шумом входной образ.

По сравнению с ИНС Хопфилда ИНС Хемминга:

1) требует почти на порядок меньшего числа соединений между НЭ при одинаковой информационной емкости сети;
2) в процессе работы всегда «сходится» (эволюционирует) к некоторому эталонному образу.

Порогово-полиномиальные нейросети

Порогово-полиномиальная ИНС предназначена для распознавания сложных (линейно не разделимых) классов образов, заданных в n-мерном пространстве двоичных признаков [4], [5].



$$y_k = \text{sign}\left(\sum_{j=1}^{m} w_j \phi_j(x)\right), \quad k = 1 \ldots N, \quad w_j \text{ — целочисленные веса}$$

$$\phi_j(x^{(h)}) = \prod_{i=1}^{n} [x_i^{(h)}]^{x_i^{(j)}}, \quad j = 1 \ldots m$$

Рис. 3.13. Архитектура порогово-полиномиальной ИНС

Схема порогово-полиномиальной ИНС изображена на рис. 3.13.

На входе ИНС имеется сенсорный блок пороговых НЭ (С-элементов), фиксирующий признаки объектов $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ в виде двоичных кодов.

Полученный входной образ (двоичный вектор $\mathbf{x}$) поступает на *блок полиномиальных преобразователей* (А-элементов), формирующий m-мерный вектор вторичных (полиномиальных) признаков $\mathbf{z} = (\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x}))$.

Эти вторичные признаки определяют m-мерное пространство полиномиальных $\phi_j(\mathbf{x}), j = 1...m$, признаков, называемое *спрямляющим пространством*. Явный вид функций $\phi_j(\mathbf{x})$ выбирается адекватно решаемой задаче непосредственно по обучающей выборке.

В выходном слое расположены *решающие* пороговые НЭ (Р-элементы), каждый из которых «отвечает» за распознавание одного из классов объектов.

Методы обучения знаниям искусственных нейронных сетей

Обучение элементарного перцептрона

Предложены различные методы и правила обучения перцептрона Ф. Розенблатта.

Один из методов называется методом *процедурой сходимости перцептрона*. Он основывается на одноименной теореме.

Теорема. Для любого данного набора входных векторов и любой требуемой их классификации алгоритм обучения через конечное число шагов приведет к вычислению требуемого набора весов, если таковой существует.

Это утверждение было сформулировано Ф. Розенблаттом.

Принцип действия алгоритма обучения перцептрона иллюстрируется на рис. 3.14.

Обучение происходит под контролем «учителя», который сообщает перцептрону правильный ответ $d(t)$ для любого входного вектора из *обучающей выборки*.

Перцептрон многократно проходит *цикл обучения*, состоящий:

- из предъявления вектора входных данных;
- вычисления ответа перцептрона $y(t)$;
- сообщения ему правильного ответа и корректировки весов связей.

Алгоритм обучения продолжает работу до тех пор, пока все входные векторы не будут правильно классифицированы.

Процесс обучения зависит от параметра адаптации μ. Задавая этот параметр внутри интервала (0; 1), можно регулировать скорость обучения, связанную с минимизацией общей ошибки функционирования прецептрона.

Алгоритм обучения элементарного перцептрона может быть записан следующим образом:

1. Инициализировать веса и пороговые значения как случайные числа из диапазона [−0.1; 0.1];
2. Выбрать очередной входной вектор $\mathbf{x}(t)$ из обучающей выборки
3. Рассчитать выходной сигнал перцептрона:

$$y(t) = f\left(\sum_{i=1}^{m} w_i x_i(t) - \theta\right).$$

4. Выполнить коррекцию весов связей:

$$w_i(t+1) = w_i(t) + \mu[d(t) - y(t)]x_i(t), \qquad \mu \in (0,1);$$

$$d(t) = \begin{cases} +1, \text{ если выходной вектор из класса A,} \\ -1, \text{ если выходной вектор из класса B.} \end{cases}$$

5. Если не все векторы из обучающей выборки классифицированы правильно, то перейти к шагу 2.
6. Конец.

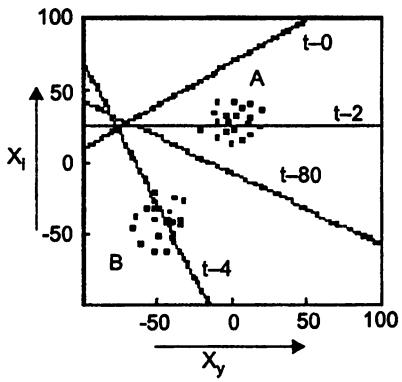


Рис. 3.14. Иллюстрация алгоритма обучения перцептрона

Алгоритмы обучения многослойного перцептрона

В последнее время широко используется процедура обучения многослойного перцептрона, получившего название *алгоритма обучения с обратным распространением ошибки* (error backpropagation). Этот алгоритм является обобщением на произвольное число слоев одной из процедур обучения элементарного перцептрона, известный как правило Уидроу—Хоффа (*дельта-правило*).

Перед обучением перцептрона связям присваиваются небольшие случайные значения. Каждая итерация процедуры состоит из двух стадий. На первой стадии на сеть подается входной вектор, и он распространяется по сети, порождая некоторый выходной вектор. Для работы алгоритма обучения требуется, чтобы передаточная *характеристика* (характеристика «вход-выход») была непрерывно дифференцируемой и неубывающей функцией с ограниченной производной. Обычно для этого используют сигмоидные НЭ, удовлетворяющие указанному требованию.

На второй стадии полученный выходной вектор сравнивается с требуемым. Если они совпадают, то обучения не происходит. В противном случае вычисляется ошибка между фактическими и требуемыми выходными значениями. Затем ошибка передается обратно последовательно от выходного слоя к входному (backpropagation).

Данный алгоритм основан на *минимизации ошибки* функционирования сети D методом градиентного спуска в пространстве весов связей:

$$D = \frac{1}{2}\sum_p \sum_k (Y_{kp} - d_{kp})^2.$$

Здесь — фактический выходной сигнал k-го НЭ в выходном слое ИНС, — требуемый выходной сигнал, p-индекс p-й пары входного и выходного векторов из ОВ.

Обучение продолжается до тех пор, пока ошибка D не уменьшится до заданной величины.

Алгоритм в общем виде может быть записан следующим образом:

1. Инициализация порогов и весов НЭ как случайных величин из диапазона [0,1; 0,1].
2. Представить очередные p-е входной и выходной вектора из ОВ.
3. Рассчитать выходной вектор.
4. Рассчитать сигнал ошибки выходного слоя по формуле

$$q_{jp} = Y'_{jp}(d_{jp} - Y_{jp}), \quad Y(a) = \frac{1}{1 + e^{-(a-\theta)}},$$

$$Y'_{jp} = Y_j(1 - Y_j).$$

5. Выполнить коррекцию весов связей для выходного слоя по формуле

$$W_{ij}(t + 1) = w_{ij}(t) + \mu q_{ip}\varphi_{ip} + \alpha(w_{ij}(t) - w_{ij}(t - 1)), \quad 0 < \alpha < 1.$$

6. далее цикл распространения ошибки по слоям до выходного включительно:

(а) Рассчитать сигнал ошибки для очередного слоя по формуле

$$q_{jp} = \varphi'_j \sum q_k w_{jk},$$

где φ'_j — производная нелинейной функции НЭ для очередного слоя, k — индекс НЭ смежного слоя, ближнего к выходному;

(б) Выполнить коррекцию весов связей для очередного слоя:

$$w_{ij}(t + 1) = w_{ij}(t) + \mu q_{ip}\varphi_{ip} + \alpha(w_{ij}(t) - w_{ij}(t - 1)), \quad 0 < \alpha < 1,$$

где φ_{ip} — выходной сигнал НЭ смежного слоя, ближнего к выходному.

7. Если пройдена не вся обучающая выборка, то прейти к шагу 2.
8. Если ошибка $D \geq \varepsilon$, то перейти к шагу 2.
9. Конец.

Недостатками алгоритма с обратным распространением ошибки являются:

- возможность преждевременной остановки из-за попадания в область локального минимума D, в которой эффективность метода градиентного спуска резко падает;

ПРИМЕЧАНИЕ В отличие от элементарного перцептрона, для которого ошибки функционирования имеют единственный минимум, многослойный перцептрон может иметь несколько минимумов с приблизительно равными областями притяжения.

- необходимость многократного (сотни и тысячи раз) предъявления всей ОВ для получения заданного качества распознавания.

Предложены различные модификации алгоритма обучения backpropagation, позволяющие повысить скорость обучения и улучшить точность воспроизведения требуемого отображения «вход—выход».

В одной из модификаций backpropagation используется алгоритм обучения, названный алгоритмом виртуального импеданса (он основан на аналогии поведения ИНС и механической колебательной системы).

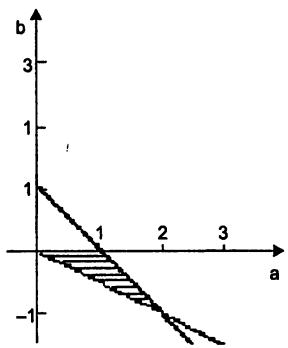


Рис. 3.15. Область выбора оптимальных значений параметров a и b (pseudo impedance control)

Алгоритм настройки весов связей имеет следующий вид:

$$dw_{ij}(t) = \mu\, q_j y_i + a dw_{ij}(t-1) + b dw_{ij}(t-2),$$

где $dw_{ij}(t)$ — приращение веса связи от i-го НЭ к j-му на t-м вычислительном цикле, μ — параметр, определяющий скорость обучения ($0 < \mu_{opt} < 1$, $\mu = 0{,}6...0{,}8$), a — инерционный член, сглаживающий изменение весов, b — параметр, позволяющий снизить возможность попадания ИНС в зону локального минимума D.

Область рационального выбора значений a и b приведена на рис. 3.15.

Другая модификация алгоритма backpropagation осуществляет на каждом шаге обучения выбор из ОВ такого подмножества вектров, для которых ошибка функционирования находится в пределах $0,5 \le |y_k - 5d_k| < 1 - e$, где y_k — сигнал k-го НЭ в выходном слое ИНС, d_k — требуемый выходной сигнал, e — параметр значимости ($e \in (0, 1)$).

В дальнейшем обучение ИНС ведется до тех пор, пока отобранное подмножество будет полностью исчерпано.

Алгоритмы обучения многослойного перцептрона с пороговыми НЭ тесно связаны с методами синтеза перцептрона, исходя из условий решаемой задачи.

Согласно алгоритму вначале определяется число слоев ИНС и распределение НЭ по слоям. Затем во входном слое выбираются группы ИНС, «отвечающие» за построение разделяющих гиперплоскостей для некоторого класса. Далее определяются веса связей этих НЭ со входами сети.

Пороги и веса связей во втором и выходном слоях определяются по следующим формулам:

Второй слой: $\theta_i = N_{вх} - e$, $0 < e < 1, w_i = 1$, где $N_{вх}$ — число НЭ, «отвечающих» за данную выпуклую область выбранного класса.

Выходной слой: $\theta_i = 0,5$, $w_i = 1$.

Обучение нейросетей Хопфилда

Обучение ИНС Хопфилда сводится к предварительному заданию матрицы связей ИНС. Приведенное для матрицы связей выражение описывает вычисление элементов матрицы кросс-корелляций между элементами эталона, усредненной по всем запоминаемым образам.

Обучение сетей Хемминга

Обучение ИНС Хемминга, также как и ИНС Хопфилда, заключается в предварительном задании весов связей и порогов НПЭ обоих слоев сети.

Веса связей и пороги НЭ нижнего (входного) слоя ИНС определяются по следующим формулам:

$$w_{ij} = x_i^j / 2,\ \theta_j = N / 2,\ i = \overline{1,N},\ j = \overline{1,M},$$

где x_i^j — состояние i-го входа *ИНС* для j-го эталона.

Для верхнего слоя формула задания весов связей имеет вид

$$t_k = \begin{cases} -e, & \text{если } k \neq 1, \\ 1, & \text{если } k = 1, \end{cases}$$

где e — параметр, выбираемый из условия $e < 1/M$.

Веса порогов всех НЭ верхнего слоя равны нулю.

Связи между соответствующими НЭ нижнего и верхнего слоев *ИНС* имеют единичные положительные веса.

Обучение и минимизация сложности порогово-полиномиальных и диофантовых сетей

Обучение порогово-полиномиальной ИНС заключается в определениии таких значений весов связей решающих НЭ, которые бы обеспечивали [4, 5]:

— безошибочное распознавание ОВ;

— целочисленные значения весов.

Желательно в процессе обучения *минимизировать* число ненулевых весов. Это позволяет синтезировать архитектуру порогово-полиномиальной ИНС *минимальной сложности.*

Удовлетворение этих требований связано с выбором *A*-элементов в виде полиномов от компонент входных пороговых *C*-элементов и самонастройкой весов связей порогово-полиномиальной ИНС.

Таблица 3.1 иллюстрирует основные этапы обучения порогово-полиномиальной ИНС и минимизацию сложности ее архитектуры.

На первом этапе элементы ОВ $x=(x_1, \ldots, x_n)$ ранжируются в порядке возрастания их норм.

На втором этапе формируется матрица A, строки которой состоят из полинома $\phi_j^{(h)}(x)$, представляющего собой компонент вектора **x**, возведенный в степень соответствующих компонент вектора $x_j^{(h)}(x)$.

В силу упорядоченности ОВ матрица A получается нижнетреугольной, что значительно снижает плотность вычисления весов связей.

Под минимизацией сложности ИНС, реализующей решающие правила $S^i(x, w)$, понимается обнуление наибольшего возможного числа весов w_j в процессе самонастройки весов связей. Это означает, что среди весов должно быть как можно больше таких, что $w_j = 0$. При этом соответствующие синаптические связи аннулируются как избыточные.

Повторяя процедуру последовательно для объектов всех классов из ОВ (путем изменения содержимого столбца S_j^i), можно рассчитать все значения весов связей для решающих НЭ и тем самым обучить ИНС оптимальному (безошибочному) распознаванию ОВ.

Таблица 3.1. Иллюстрация основных этапов обучения порогово-полиномиальной *ИНС*

	ОВ				S_j^i	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	A_{10}	a_{11}	a_{12}	ω_j^i
						1	x_4	x_3	x_3	x_1	x_1	x_2	x_2	x_1	X_1	x_1	x_1	
									x_4	x_2	x_4	x_4	x_3	x_2	x_3	x_2	x_2	
														x_3	x_4	x_4	x_3	
																	x_4	
m	x_1	x_2	x_3	x_4														
1	0	0	0	0	−1	1	0	0	0	0	0	0	0	0	0	0	0	−1
2	0	0	0	1	−1	1	1	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	−1	1	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0	1	1	−1	1	1	1	1	0	0	0	0	0	0	0	0	0
5	1	0	1	0	−1	1	0	1	0	1	0	0	0	0	0	0	0	0
6	1	0	0	1	−1	1	1	0	0	0	1	0	0	0	0	0	0	0
7	0	1	0	1	+1	1	1	0	0	0	0	1	0	0	0	0	0	2
8	0	1	1	0	+1	1	0	1	0	0	0	0	1	0	0	0	0	2
9	1	1	1	0	+1	1	0	1	0	0	0	0	1	1	0	0	0	0
10	1	0	1	1	−1	1	1	1	1	1	1	0	0	0	1	0	0	0
11	1	1	0	1	−1	1	1	0	0	0	1	1	0	0	0	1	0	0
12	1	1	1	1	−1	1	1	1	1	1	1	1	1	1	1	1	1	0

$$w_1^i = S_1^i / a_1 \ ,$$

$$w_j^i = \begin{cases} \dfrac{1}{a_k}(S_j^i - \sum_{k=1}^{j-1} a_k w_k^i), \text{ если } S_j^i \sum_{k=1}^{j-1} a_k w_k^i \le 0, \\ 0, \text{ если } S_j^i \sum_{k=1}^{j-1} a_k w_k^i > 0. \end{cases}$$

В заключение отметим, что все описанные ИНС по существу служат для формирования и нейросетевого представления знаний. Эти знания автоматически извлекаются из экспериментальных или статистических баз данных, предъявляемых ИНС в режиме обучения. После обучения ИНС аппроксимирует с требуемой точностью заранее неизвестные функциональные зависимости (интерполяция и экстраполяция функций), решающие правила (распознавание образов, диагностика состояний, прогнозирование ситуаций и т. п.) и другие «скрытые» закономерности. Таким образом, ИНС извлекает (в процессе обучения) знания из обучающих выборок данных и аннулируют их в своей архитектуре с настроенными синаптическими весами и связями.

Искусственные нейронные сети в задачах идентификации и управления

Рассмотрим особенности ИНС, используемых для моделей динамики и представления законов управления. Ради определенности ограничимся задачами нейросетевой идентификации неизвестных уравнений динамики механических систем и синтеза различных законов нейроуправления. Для решения этих задач часто используется ИНС, получившая название CMAC (Cerebellar Model Articulation Computer — мозжечковая модель суставного регулятора) [7].

CMAC представляет собой ассоциативную НС, в которой только небольшой участок сети оказывает влияние на выходной сигнал в текущий момент времени, и этот участок определяется входным вектором НС. Архитектура CMAC представлена на рис. 3.16.

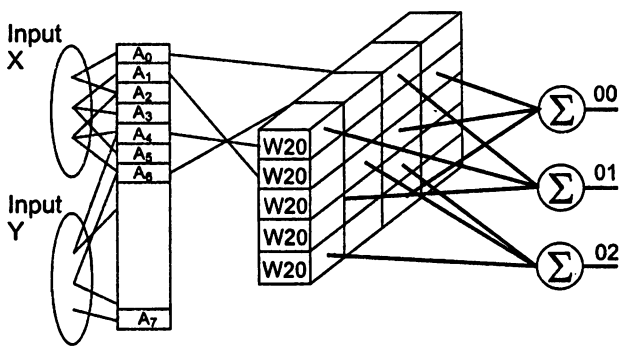


Рис. 3.16. Архитектура нейронной сети CMAC

На рисунке:

X, Y — входные данные;

$A_0 \ldots A_n$ — детекторы пространства состояния (логическое «или»);

W — весовые коэффициенты;

а — выходные сумматоры

Ассоциативное отображение, заложенное в CMAC, обеспечивает локальное обобщение: похожие входные сигналы вызывают схожую реакцию сети, то есть объединяются в кластер (подкласс), тогда как сильно различающиеся входные сигналы ведут к практически независимым сигналам на выходе. В идеальном случае ассоциативное отображение в CMAC гарантирует, что близлежащие точки в пространстве входов обобщаются, тогда как далеко расположенные друг от друга точки в пространстве входов не приводят к такому обобщению. Мерой близости

является сумма (по всем компонентам) абсолютных значений разностей каждой компоненты входного вектора. CMAC-сети обладают следующими свойствами:

- Компоненты входного вектора квантуются, но число уровней квантования может быть сколь угодно большим, так что достижима любая заданная степень точности.
- CMAC использует правило модификации весов Уидроу и Хоффа, основанное на методе наименьших квадратов [7]. Этот алгоритм эквивалентен градиентному спуску и сходится к единственному минимуму.
- CMAC подчиняется принципу суперпозиции в пространстве выходов. Например, если CMAC можно обучить аппроксимации многомерных синусоид с различными пространственными частотами (гармониками), то ее можно обучить и аппроксимации целого класса функций. Эти функции остаются нелинейными, так как суперпозиция действует только в пространстве выходов.
- CMAC можно аппаратно реализовать на массивах логических элементов.

Работу с ИНС типа CMAC можно разделить на две фазы: *обучение* и *идентификацию* (*восстановление*) или *управление*. Если ИНС уже обучена, то есть схема связей и веса ИНС известны, то фаза идентификации (восстановления) сводится к последовательному изменению значений активации нейроэлементов всех слоев ИНС с целью выработки соответствующих выходных (идентификационных или управляющих) сигналов сети.

Существуют три типа процедур обучения:

1) обучение с супервизором;

2) закрепленное обучение;

3) обучение без супервизора.

Обучение с супервизором требует наличия обучающей выборки. Для каждого элемента обучающей выборки (или обучающей базы данных) супервизору («учителю») известна желаемая реакция ИНС. По этой информации определяется сигнал ошибки после каждого предъявления элемента обучающей выборки.

Процедура закрепленного обучения похожа на обучение с супервизором, но вместо желаемой реакции сети передается лишь качественная оценка выполнения заданной задачи в течение времени (используется своего рода «критика», то есть оценка вида «правильно—неправильно»). Таким образом, веса связей закрепляются для правильно выполненных действий и изменяются при неправильной реакции ИНС. Такая процедура обучения обычно применяется для задач классификации данных и распознавания образов.

Обучение без супервизора не использует знания «учителя» о правильных реакциях ИНС. Поэтому процесс самообучения ИНС сводится к кластеризации данных, то есть построению внутренних кластеров.

Алгоритмы обучения ИНС можно разделить на две главные группы: правила обучения без супервизора и правила обучения с супервизором. В задачах идентификации и управления обычно используются только алгоритмы обучения с

супервизором. Алгоритмы самообучения, то есть обучения без супервизора, предназначены для кластеризации данных по сходству их признаков.

Задачи идентификации и при неполной информации об объекте управления сводятся к оцениванию неизвестных параметров и функций. С другой стороны, задача обучения ИНС также сводится к оцениванию неизвестных синаптических параметров ИНС в некотором нейросетевом базисе. Например, обучение распознаванию образов заключается в настройке значений синаптических параметров ИНС по набору обучающих примеров, называемых обучающей выборкой. Поскольку многослойные ИНС показывают хорошие результаты в задачах классификации и аппроксимации неизвестных функций, то естественно ожидать, что они могут оказаться полезными и в решении прикладных задач оптимального и адаптивного управления. Однако эти задачи имеют ряд специфических особенностей. Поэтому нейросетевое представление знаний о динамике и законах управления наталкивается на значительные трудности. Обзор основных подходов к использованию ИНС в задачах управления можно найти в [6], [7], [9], [11], 12]. Приведем краткое описание некоторых из этих подходов.

Нейросетевое управление с супервизором

При обучении с супервизором (например, копировании существующего регулятора) искусственная НС обучается имитировать («копировать») некоторый закон управления. Этот закон управления известен человеку-оператору или управляющему устройству и обеспечивает требуемое качество управления. Цель обучения ИНС в этом случае заключается в нейросетевой аппроксимации этого «супервизорного» закона управления в соответствии со схемой, изображенной на рис. 3.17.

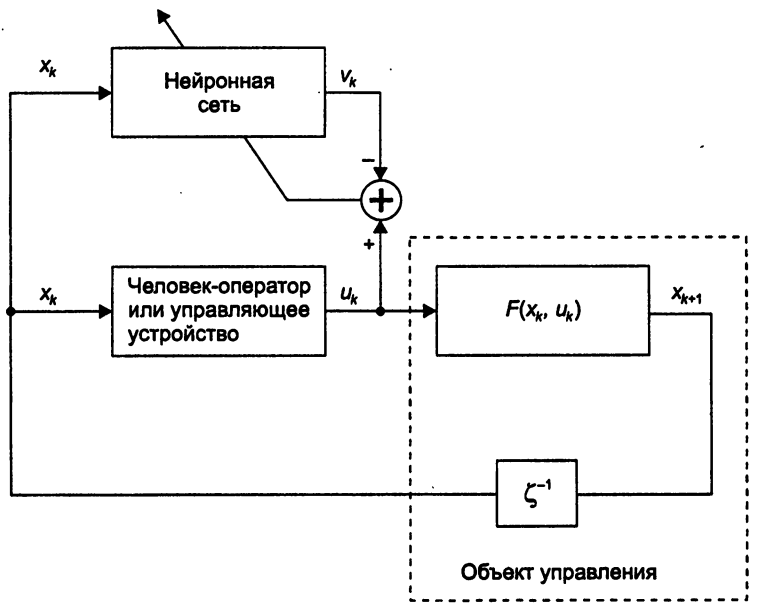


Рис. 3.17. Нейроуправление с супервизором

Возникает естественный вопрос, зачем создавать новое устройство управления на основе НС, если эффективное устройство уже существует. Во-первых, существующее управляющее устройство может оказаться непрактичным или неприменимым в опасных для жизни человека условиях. Во-вторых, ИНС способна формировать эффективное управление на основании более простого нейросетевого представления о динамике объекта управления по сравнению с существующим устройством управления.

В качестве недостатка описанного нейроуправления с супервизором следует отметить то, что после обучения ИНС представляет собой лишь «копию» существующего устройства управления и при изменении параметров объекта управления или среды может потребоваться полное переобучение сети.

Нейросетевая идентификация объектов управления

Информация для обучения ИНС для идентификации (аппроксимации) модели динамики объекта управления (ОУ) получается путем наблюдения сигналов «вход—выход» (рис. 3.18). Вход ИНС подвергается тому же самому управляющему воздействию u_k, что и ОУ, при этом выход ОУ y_k (или его вектор состояния x_k) является желаемым выходным сигналом для ИНС.

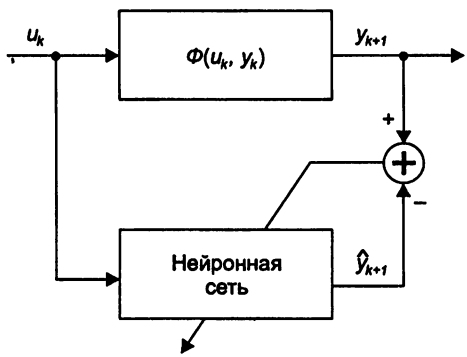


Рис. 3.18. Идентификация объектов управления

Для идентификации непрерывных динамических ОУ необходимо подавать на входы ИНС вектор переменных состояния x и его производную, так как выход ОУ зависит не только от сигнала управления u, но и от состояния x ОУ. В случае дискретной системы управления необходимо дополнить ИНС некоторым количеством линий задержки (ЛЗ), как это видно из рис. 3.19.

Количество необходимых линий задержки сигналов определяется порядком объекта управления. Линии задержки могут быть заменены внутренними соединениями сети (рекуррентные НС), которые в этом случае используют элементы памяти.

Часто для простых (то есть линейных и слабо нелинейных) ОУ классические методы идентификации сходятся быстрее и более точно, чем использовавшиеся нейросетевые методы. Однако с помощью ИНС можно идентифицировать существенно нелинейные ОУ, имея весьма малые знания о классе, структуре и параметрах ОУ. В этом случае классические алгоритмы идентификации часто не дают удовлетворительных результатов.

С другой стороны, считается, что структурно-параметрическая идентификация на базе ИНС приводит к моделям типа «черного ящика». Из этого делается ошибочный вывод, что к ним нельзя применить известные аналитические методы теории управления. Однако это не всегда так. В частности, в [12] разработан метод сочетания традиционного и нейросетевого управления для существенно нелинейного объекта (манипуляционного робота) с перекрестными связями. В основе этого метода лежит идея использования обратной динамической модели, которая аппроксимируется ИНС.

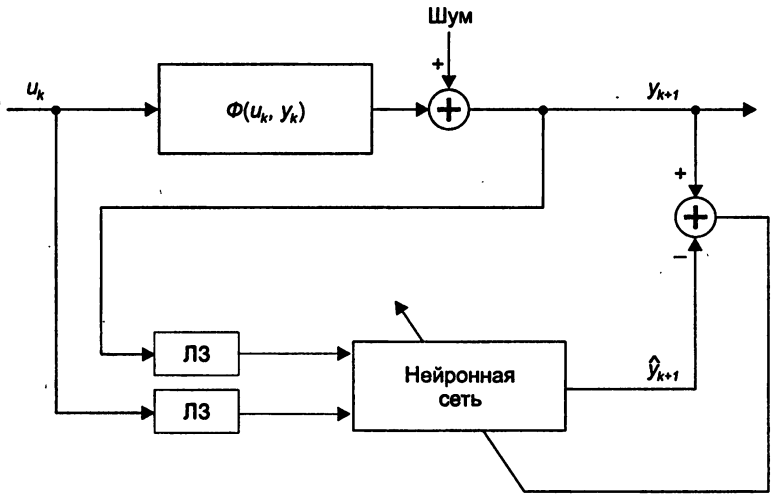


Рис. 3.19. Идентификация дискретных динамических ОУ с линиями задержки

Схемы обучения управлению с использованием обратной модели динамики объекта

Методы идентификации на базе ИНС могут быть использованы для нейросетевой аппроксимации обратной модели динамики ОУ. В литературе [6], [7] упоминаются два вида подключения ИНС к ОУ при обучении.

В схеме непрямого (косвенного) обучения (рис. 3.20) желаемый выход ОУ d_k подается на вход ИНС, выход u_k которой подается на ОУ. При этом выход ОУ y_{k+1} используется как вход другой копии ИНС — ее дубликата. Разность управ-

ления u_k и выхода второй ИНС v_k используется для корректировки весов связей ИНС и ее копии.

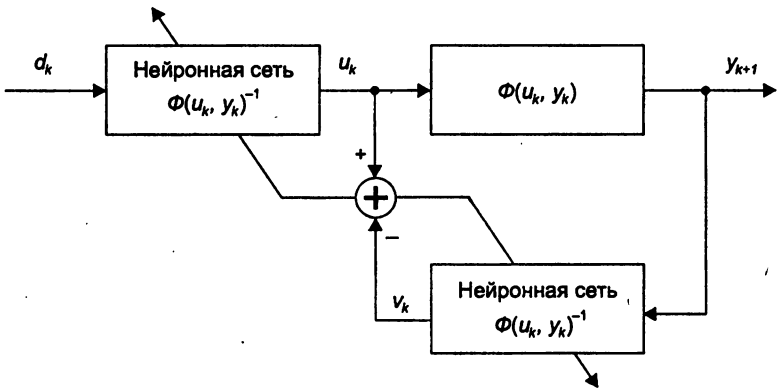


Рис. 3.20. Схема непрямого (косвенного) обучения (обе ИНС идентичны и представляют обратную модель ОУ)

Однако иногда эта схема не работает по той причине, что ИНС может отображать большое количество различных входных сигналов d_k на одно и то же значение u_k («вырожденное отображение»). Поэтому ошибка $u_k - v_k$, используемая для модификации синаптических весов, может оказаться нулевой, хотя общая ошибка $d_k - y_{k+1}$ не равна нулю.

Обобщенная схема обучения (рис. 3.21) решает эту проблему путем непосредственной подачи сгенерированного управления u_k на вход ОУ.

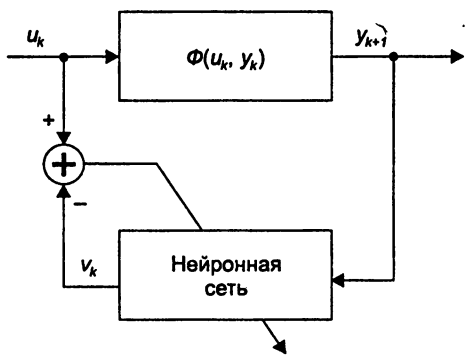


Рис. 3.21. Обобщенная схема обучения ИНС

Схема специализированного обучения ИНС

В схеме специализированного обучения ИНС желаемый выход ОУ d_k подается на вход ИНС, а выход сети u_k подается на вход ОУ (рис. 3.22). Разность реакции системы y_{k+1} и d_k является информацией для изменения весов ИНС. Нейросетевой регулятор, соответствующий этой схеме, может быть обучен во время работы системы. Однако для настройки его весов необходимо пропускать вектор ошибки $d_k - y_{k+1}$ обратно через ОУ, что обычно возможно только приближенно. Для этой задачи рекомендуется применить ИНС, эмулирующую ОУ. Нейросетевой эмулятор ОУ можно обучать в режиме тестирования объекта перед обучением нейросетевого регулятора в режиме работы системы, используя ИНС эмулятора с «замороженными» весами для обратного распространения ошибки.

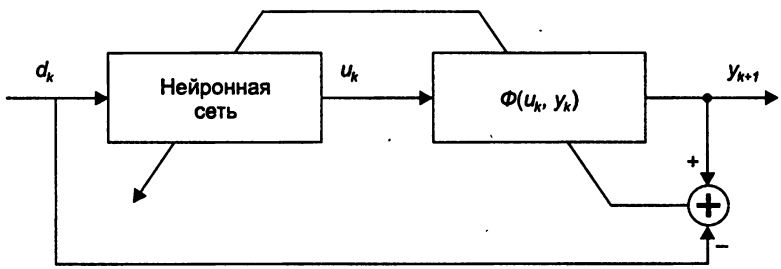


Рис. 3.22. Схема специализированного обучения ИНС

Схема специализированного обучения и схемы обучения управлению с использованием обратной модели ОУ ранее в основном предназначались для решения обратной задачи кинематики при управлении манипуляторами. Эта задача состоит в определении обобщенных координат для достижения определенного положения в рабочем пространстве робота (в системе координат, связанной с основанием). Динамика ОУ в таких задачах управления учитывалась другими способами.

Для управления динамическими ОУ схему специализированного обучения необходимо дополнить обратными связями по переменным состояния и их производным, которые подаются на вход ИНС. Например, для манипуляторов компонентами вектора состояния являются обобщенные координаты и скорости звеньев, которые легко измеряются. Измерить обобщенные ускорения звеньев также не составляет большой проблемы. Поэтому в схеме обобщенного обучения (см. рис. 3.21) необходимо на вход ИНС подавать не только выходной сигнал ОУ, но и вектор переменных состояния и его производную.

Для управления роботами с априорно заданным характером затухания переходных процессов важно знать обратную модель динамики робота. Поэтому описанная выше модификация схемы имеет большое значение в теории нейроуправления.

Метод обратного распространения ошибки в задачах управления

Одним из наиболее удачных методов обучения ИНС для управления является метод обратного распространения ошибки, а также его модификации [6], [7], [9]. В задачах управления возможно использовать on-line- или off-line-обучение.

В первом случае при управлении объектом с помощью традиционного регулятора (или просто с помощью некоторых тестовых сигналов) в определенные моменты времени с датчиков считываются пары «вход—выход» (управление → вектор состояний, его производная) и с помощью алгоритма обратного распространения ошибки производится настройка весов ИНС. В результате ИНС обучается представлению обратной зависимости (то есть вектор состояний, его производная → управление) [12]. При этом на вход ИНС подаются вектор состояний ОУ и его производная, и на выходе ИНС появляется сигнал, соответствующий управлению. Разность между выходом ИНС и реальным значением управления является ошибкой. Далее производится настройка весов ИНС в направлении градиента целевой функции, представляющей собой функционал минимума ошибки. Процедура повторяется до достижения требуемой точности.

При off-line-обучении с помощью существующего регулятора или специальных тестов предварительно создается обучающая выборка, и НС обучается на ней.

Литература

1. Мак-Каллон У. С., Питтс В. Логическое исчисление идей, относящихся к нервной активности//Автоматы. —М.: Изд-во иностр. л-ры, 1956. С. 362–402.
2. Розенблатт Ф. Принципы нейродинамики. Перцептроны и теория механизмов мозга., М. — Мир, 1965.
3. Тимофеев А. В. Методы синтеза диофантовых нейросетей минимальной сложности//Доклады АН, 1995. Т. 301. № 3. С. 1106–1109.
4. Тимофеев А. В., Каляев А. В. Методы обучения и минимизации сложности когнитивных нейромодулей супер-макро-нейрокомпьютера с программируемой архитектурой//Доклады АН. 1994. Т. 337. № 2. С. 180–183.
5. Тимофеев А. В., Шибзухов З. М. Методы синтеза и минимизации сложности диофантовых нейронных сетей над конечным полем//Автоматика и телемеханика. 1997. № 4. С. 204–212.
6. Уоссерман Ф. Нейрокомпьютерная техника. — М.: Мир, 1992.
7. Aleksander I. Morton H. An Introduction to Neural Computing. — London, U.K.: Chapman & Hall, 1990.
8. Bogdanov A., Popov I., Tarnovsky S. Neurob — Software for Simulation and Comparative Analysis of Traditional and Neural Methods of Robots Control//Proceedings of International Conference on Informatics and Control. 1997. June 9–13. P. 621–628.

9. Hecht-Nielsen R. Neurocomputing. — Redwood City, CA: Addison-Wesley Pub. Co., 1990.
10. Sprecher D. A. On the structure of continious functions of several variables// Transactions of the American Mathematical Society. 1965. Vol. 115. P. 340–355.
11. Timofeev A. V. Learning and Complexity minimization of diophantine and spline neural networks for control//Proc. of the 2-nd Russian-Swedish Control Conf. (RSCC'95). 1995. August 29–31, Saint-Petersburg. 1995. — P. 47–52.
12. Timofeev A. V., Bogdanov A. A. Synthesising Principles of Neural Controllers for Robots Optimal Control//Proc. of the 2nd International Symposium on Neuroinformatics and Neurocomputers. 1995. — Rostov-on-Don, Russia, 1995. P. 189–193.
13. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения//Доклады АН СССР. 1957. Т. 114, № 5. С. 953–956.

4 ГЛАВА Эволюционные алгоритмы[1]

История эволюционных алгоритмов

Прежде всего необходимо упомянуть, что отнюдь не все ученые признают наличие эволюции. Многие религиозные течения (например, свидетели Иеговы) считают учение об эволюции живой природы ошибочным. Не хотелось бы сейчас вдаваться в полемику относительно доказательств «за» и «против» по одной простой причине. Даже если авторы не правы в своих взглядах, объясняя эволюционные алгоритмы как аналоги процессов, происходящих в живой природе, никто не сможет сказать, что эти алгоритмы неверны. Несмотря ни на что, они находят огромное применение в современной науке и технике и показывают подчас просто поразительные результаты.

Первые случаи применения генетических алгоритмов находят в образцах, возраст которых более 1 млрд лет. Конечно же, речь идет о живых организмах, исследование процессов размножения которых и позволило разработать данный алгоритм глобальной оптимизации.

Первой ласточкой, послужившей предвестницей применения естественных алгоритмов в повседневной деятельности человека, явилась работа Чарльза Дарвина «Происхождение видов», написанная в 1859 году. Именно в этой работе четко обозначены три столпа, на которых зиждятся современные генетические алгоритмы, — наследственность, изменчивость и отбор. Однако, если отбор — процесс отбраковки нежизнестойких видов, в общем-то был понятен, то механизм, который отвечал за сохранение в потомках черт предков и обеспечивал способность к приспособлению под новые условия окружающей среды, стал понятен гораздо позднее — в 1944 году О. Эйвери, К. Маклеод и М. Маккарти опубликовали результаты своих исследований, доказывавших, что за наследственные процессы ответственна «кислота дезоксирибозного типа». Однако о том, как работает ДНК, весь мир узнал еще позднее — 27 апреля 1953 года в номере журнала «Nature» вышла статья Уотсона и Крика, впервые предложивших модель двухцепочечной спирали ДНК.

---

[1] Глава подготовлена Сергеем Сотником с участием Александра Кука (NeuroPower AG, Германия).

Однако многие моменты в работе ДНК не ясны до сих пор. Относительно новыми сведениями являются, например, данные о том, что разные участки ДНК мутируют с разной частотой.

Раскрытие механизмов, отвечающих за создание и работу таких сложных систем, как живые организмы, конечно же вдохновило многих исследователей на моделирование этих процессов при помощи компьютеров. В настоящее время существует огромное количество различных алгоритмов, в той или иной степени моделирующих естественные процессы. В качестве основных направлений можно назвать генетические алгоритмы и классификационные системы Голланда (Holland) [5], опубликованные в начале 60-х годов и получившие всеобщее признание после выхода в свет книги, ставшей классикой в этой области, — «Адаптация в естественных и искусственных системах» [5]. В 70-е годы Растригиным Л. А. в рамках теории случайного поиска был предложен ряд алгоритмов, которые моделировали различные аспекты поведения живых организмов. Дальнейшее развитие эти идеи получили в работах Букатовой И. Л., посвященных эволюционному моделированию. Развивая идеи Цетлина М. Л. о целесообразном и оптимальном поведении стохастических автоматов, Неймарк Ю. И. предложил осуществлять поиск глобального экстремума на основе коллектива независимых автоматов, моделирующих процессы развития и элиминации особей. Большой вклад в развитие эволюционного программирования внесли Фогел (Fogel) и Уолш (Walsh). При всей разнице в подходах каждая из этих «школ», взяв за основу ряд принципов, существующих в природе, упростила их до такой степени, чтобы их можно было реализовать на компьютере.

Из основных особенностей эволюционных алгоритмов можно отметить их некоторую сложность в плане настройки основных параметров (вырождение либо неустойчивость решения). Поэтому, экспериментируя с ними и получив не очень хорошие результаты, попробуйте не объявлять сразу алгоритм неподходящим, а попытаться опробовать его при других настройках. Данный недостаток следует из основной эвристики — можно «уничтожить» предка самого лучшего решения, если сделать селекцию слишком «жесткой» (не зря ведь биологам давно известно, что если осталось меньше десятка особей исчезающего вида, то этот вид сам по себе исчезнет из-за вырождения).

Генетический алгоритм

Генетический алгоритм (ГА) является самым известным на данный момент представителем эволюционных алгоритмов и по своей сути является алгоритмом для нахождения глобального экстремума многоэкстремальной функции. ГА представляет собой модель размножения живых организмов.

Для начала представим себе целевую функцию от многих переменных, у которой необходимо найти глобальный максимум или минимум:

$f(x_1, x_2, x_3, \ldots, x_N)$.

Для того чтобы заработал ГА, нам необходимо представить независимые переменные в виде хромосом — цепочек символов, с которыми и работает ГА. Как это делается?

Как создать хромосомы?

Первым вашим шагом будет преобразование независимых переменных в цепочки бит, которые будут содержать всю необходимую информацию о каждой создаваемой особи. Имеется два варианта кодирования параметров:

- в двоичном формате;
- в формате с плавающей запятой.

В случае если мы используем двоичное кодирование, мы используем N бит для каждого параметра, причем N может быть различным для каждого параметра. Если параметр может изменяться между минимальным значением MIN и максимальным MAX, используем следующие формулы для преобразования:

$r = g \times (\text{MAX} - \text{MIN})/(2^N - 1) + \text{MIN};$

$g = (r - \text{MIN})/(\text{MAX} - \text{MIN}) \times (2^N - 1),$

где g — целочисленные двоичные гены, r — эквивалент генов в формате с плавающей запятой.

Хромосомы в формате с плавающей запятой создаются при помощи размещения закодированных параметров один за другим.

Если сравнивать эти два способа представления, то более хорошие результаты дает вариант представления в двоичном формате (особенно при использовании кодов Грея). Правда, в этом случае мы вынуждены мириться с постоянным кодированием/декодированием параметров.

Как работает генетический алгоритм?

В общем, генетический алгоритм работает следующим образом. В первом поколении все хромосомы генерируются случайно. Определяется их «полезность». Начиная с этой точки, ГА может начинать генерировать новую популяцию. Обычно размер популяции постоянен.

Репродукция состоит из двух шагов:

1. Селекции.
2. Генетических операторов (порядок применения не важен), из которых самыми важными и принципиально необходимыми являются:
   - кроссовер;
   - мутация.

Роль и значение селекции мы уже рассмотрели в обзоре эволюционных алгоритмов.

Кроссовер является наиболее важным генетическим оператором. Он генерирует новую хромосому, объединяя генетический материал двух родительских. Существует несколько вариантов кроссовера. Наиболее простым является одноточечный. В этом варианте просто берутся две хромосомы и перерезаются в случайно выбранной точке. Результирующая хромосома получается из начала одной и конца другой родительских хромосом:

001100101110010|11000 → 001100101110010 *11100*

*11010110110100*0|*11100*

Мутация представляет собой случайное изменение хромосомы (обычно простым изменением состояния одного из битов на противоположное). Данный оператор позволяет более быстро находить локальные экстремумы ГА с одной стороны и позволяет «перескочить» на другой локальный экстремум с другой:

0011001011100**1**011000 → 0011001011100**0**111000

Примером дополнительных операторов может служить инверсия. Инверсия изменяет порядок бит в хромосоме путем циклической перестановки (случайное количество раз). Многие модификации ГА обходятся без данного генетического оператора:

0011001011100**10**11000 → **11000**001100101110010

Очень важно понять, за счет чего ГА на несколько порядков превосходит по скорости случайный поиск во многих задачах? Дело здесь, видимо, в том, что большинство систем имеют довольно независимые подсистемы. Вследствие этого при обмене генетическим материалом часто может встретиться ситуация, когда от каждого из родителей берутся гены, соответствующие наиболее удачному варианту определенной подсистемы (остальные «уродцы» постепенно вымирают). Другими словами, ГА позволяет накапливать удачные решения для систем, состоящих из относительно независимых подсистем (большинство современных сложных технических систем и все известные живые организмы). Соответственно, можно предсказать и когда ГА скорее всего даст сбой (или, по крайней мере, не покажет особых преимуществ перед методом Монте-Карло), — системы, которые сложно разбить на подсистемы (узлы, модули), а также в случае неудачного порядка расположения генов (рядом расположены параметры, относящиеся к различным подсистемам), при котором преимущества обмена генетическим материалом сводятся к нулю.

Другим важным моментом, которым отличаются различные генетические алгоритмы, является процесс селекции. Здесь мы находим и простой допуск к размножению определенного количества лучших объектов, и сложные турнирные схемы, элитизм (дарование жизни самым лучшим особям — элите, и в следующем поколении), разделение популяции на несколько частей (имитация эволюции на островах Океании) с нечастыми обменами генетическим материалом между ними.

Общими принципами применения ГА являются следующие (табл. 4.1):

Таблица 4.1. Общие принципы применения ГА

Увеличивают скорость нахождения решения	Улучшают качество работы (поисковые способности) ГА
Увеличение прессинга естественного отбора	Уменьшение прессинга естественного отбора
• Уменьшение количества особей, допускаемых к размножению	• Увеличение объема генетического материала
• Использование элитизма	• Диплоидия (также увеличивает количество генетического материала) [1]
• Уменьшение общего количества генерируемого потомства	Разбиение популяции на части (кстати говоря, это еще и легкий путь распараллелить алгоритм)
Выполнение алгоритма параллельно на нескольких компьютерах (процессорах)	

Генетическое программирование

Данные, которые закодированы в генотипе, могут представлять собой команды какой-либо виртуальной машины. В таком случае мы говорим об эволюционном или генетическом программировании. В простейшем случае мы можем ничего не менять в генетическом алгоритме. Однако в таком случае длина получаемой последовательности действий (программы) получается не отличающейся от той (или тех), которую мы поместили как затравку. Современные алгоритмы генетического программирования распространяют ГА для систем с переменной длиной генотипа.

Для того чтобы работать с генетическим материалом переменной длины, генетическое программирование (ГП) работает с отличающимися от стандартного ГА формами представления геномов и, соответственно, алгоритмами применения к ним генетических операторов. Исторически первой, предложенной Н. Крамером [4] и Дж. Коза [7], была древообразная форма генома (рис. 4.1).

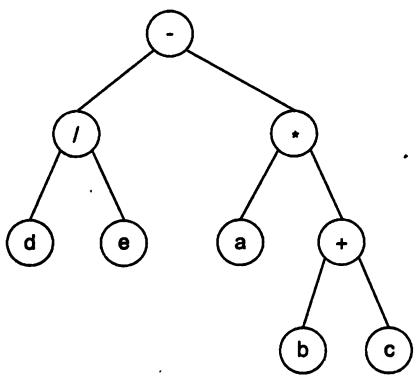


Рис. 4.1. Представленный в виде дерева алгоритм вычисления функции $d/e-(b+c)\times a$

Из других, распространенных в настоящее время, можно назвать линейную и сетевую (графовую) формы (рис. 4.2, 4.3).

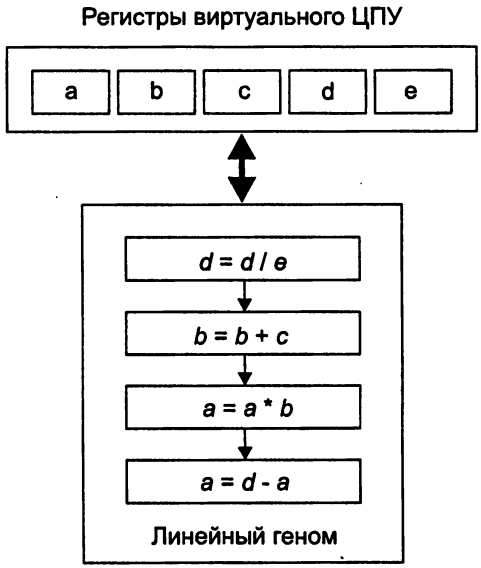


Рис. 4.2. Алгоритм вычисления функции $d/e-(b+c)\ldots a$, закодированный в линейном геноме

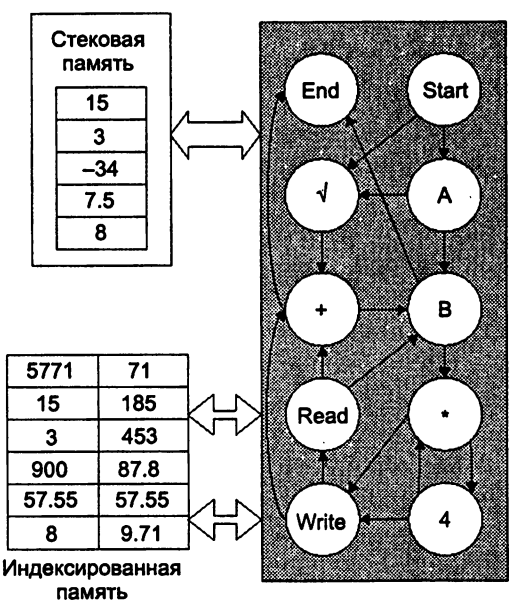


Рис. 4.3. Пример небольшой программы, закодированной в виде графового генома

При использовании древовидного кодирования алгоритма кроссовер в простейшем случае представляет собой просто обмен двумя случайно выбранными поддеревьями между родителями (рис. 4.4).

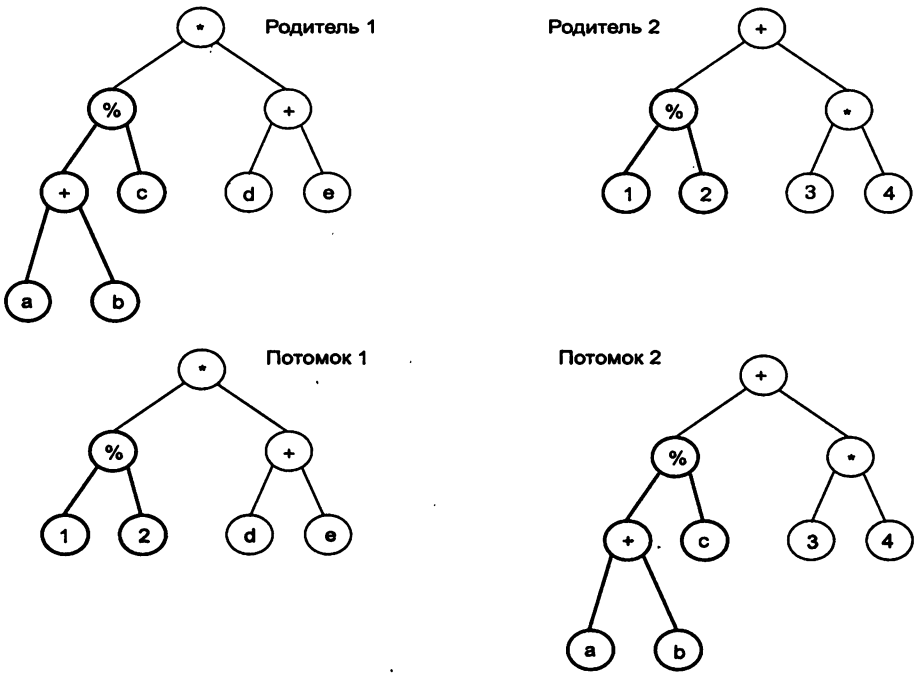


Рис. 4.4. Схема обмена генетической информацией (кроссовер) между двумя родителями при древообразном способе хранения алгоритма

Также сравнительно несложным является линейный кроссовер (рис. 4.5).

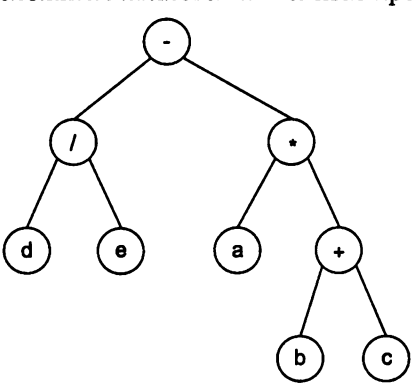


Рис. 4.5. Линейный кроссовер

Самой сложной является схема сетевого кроссовера. Другим принципиально важным генетическим оператором является мутация. В различных вариантах ГП она выполняется по-разному, но общая схема остается прежней — случайным

образом меняется случайно выбранный элемент. Например, в случайно выбранной трехадресной команде (приемник и два источника), часто используемой при линейном ГП, меняется либо приемник, либо источник, либо тип выполняемой операции. Кроме того, случайным образом может либо вставляться новый оператор, либо удаляться уже имеющийся.

В связи с тем, что для рассмотренных схем хранения алгоритмов большинство случаев применения оператора кроссовера являются разрушительными, программы, сгенерированные при помощи ГП, имеют тенденцию к накоплению интронов — ничего не делающих участков кода, единственным смыслом существования которых является уменьшение разрушительных последствий применения генетических операторов. Типичные примеры интронов (древовидная структура записана в символьном виде) [8]:

- (NOT (NOT X))
- (AND...(OR X X))
- (+...(–X X))
- (+X 0)
- (* X 1)
- (* ... (DIV X X))
- (MOVE-LEFT MOVE-RIGTH)
- (IF (2=1) ... X)
- (A := A)

На самом деле интроны не являются изобретением генетических алгоритмов, моделируемых человеком. Такие инертные последовательности можно найти и в естественных генах.

Постепенно, по мере генерации новых поколений, количество интронов в сгенерированных программах может достигать 60 и более процентов. В связи с этим многие новые алгоритмы имеют те или иные особенности, которые увеличивают средний процент выживающих после кроссовера особей-программ. Одним из таких способов может быть введение гомологичности (тоже, кстати, заимствованной у природы) и т. п.

Метод группового учета аргументов

Краткая история

Впервые метод группового учета аргументов (МГУА) появился в 1968 году, и его появление связывают с именем академика Алексея Григорьевича Ивахненко — одного из самых известных украинских ученых в области computer science. Плодовитость этого ученого поражает — под его руководством было защищено около 220 диссертаций (из них 27 на докторскую степень). Он является автором около

15 монографий и более чем 300 исследовательских статей в области математического моделирования и распознавания образов сложных систем. Кроме того, имеет 5 детей, 7 внуков, 2 правнука. Несмотря на свои более чем 85 лет, до сих пор продолжает активную научную деятельность.

За пределами бывшего СССР первые публикации по МГУА начинают появляться с 1972 года. Наибольшую активность в данном направлении демонстрируют ученые Польши, Японии, Германии, предложившие ряд усовершенствованных вариантов первоначального алгоритма. За последние годы появилась целая плеяда МГУА-подобных алгоритмов, использующих основные базовые принципы МГУА.

Многослойный итеративный МГУА

Начать знакомство с базовыми принципами МГУА лучше всего на примере многослойного итеративного варианта (МИМГУА).

На рис. 4.6 показано трехмерное пространство, двумя координатами которого являются признаки x_1 и x_2, а третьей — значение некоторой функции $a = f(X)$, или $f(x_1, x_2)$. Нашей задачей будет являться построение функции, наиболее точно аппроксимирующей данную поверхность.

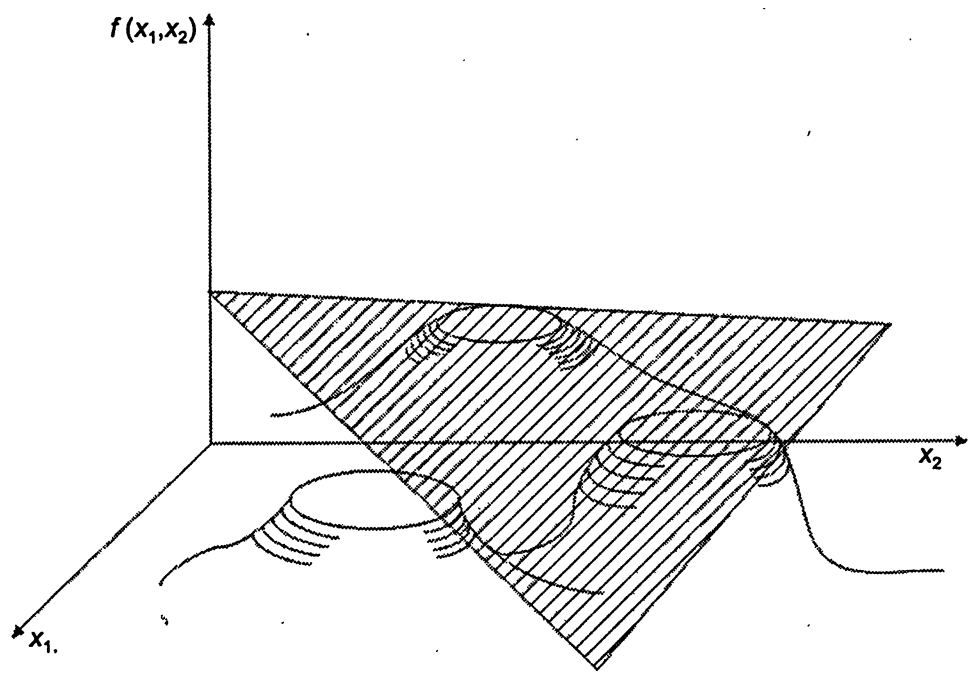


Рис. 4.6. Функция $f(x_1, x_2)$

Подобного вида задача очень часто возникает при распознавании образов или при построении модели объекта. Как известно, при любом характере поверхности функции f(X) теоретически мы сможем построить сколь угодно точный аппроксимирующий полином вида

$$f(X) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \gamma_{ijk} x_i x_j x_k + \ldots \quad (1)$$

Задача аппроксимации легко решалась бы в приведенной постановке, если бы мы заранее знали, какие члены полинома (1) используются в каждом конкретном случае. Однако даже в такой благоприятной постановке задачи нас ожидают огромные трудности, связанные с тем, что даже если учитывать только четыре признака и построить для них полный полином (1) с максимальной степенью три, то мы вынуждены будем оперировать матрицами размерностью 40×40 элементов. Если же нам придет в голову мысль построить таким методом аппроксимирующую поверхность для функции 20 элементов, учитывая члены вплоть до степени 10, то надо приготовиться обрабатывать матрицы размерностью $5{,}38947 \cdot 10^{11} \times 5{,}38947 \cdot 10^{11}$ элементов. Еще одной трудностью для нас будет являться тот факт, что для нахождения такого количества неизвестных при помощи, например, метода наименьших квадратов длина обучающей последовательности также должна быть очень большой.

Предложенный А. Г. Ивахненко метод позволяет заменить одномоментное построение полинома вида (1) его последовательным (итеративным) синтезом из сравнительно простых элементарных функций (классификаторов). Большинство задач распознавания образов, решаемых с помощью МГУА, сводится к восстановлению функции по небольшим последовательностям. Представим полином (1) в виде

$$F = F(x_1, x_2, \ldots, x_3) \quad (2)$$

Априори, характер функции F неизвестен. Описание (2) заменяется несколькими частными описаниями:

$$y_1 = f(x_1, x_1),\ y_2 = f(x_1, x_2),\ y_3 = f(x_1, x_3),\ \ldots,\ y_{s-1} = f(x_{n-1}, x_n),\ y_s = f(x_n, x_n), \quad (3)$$

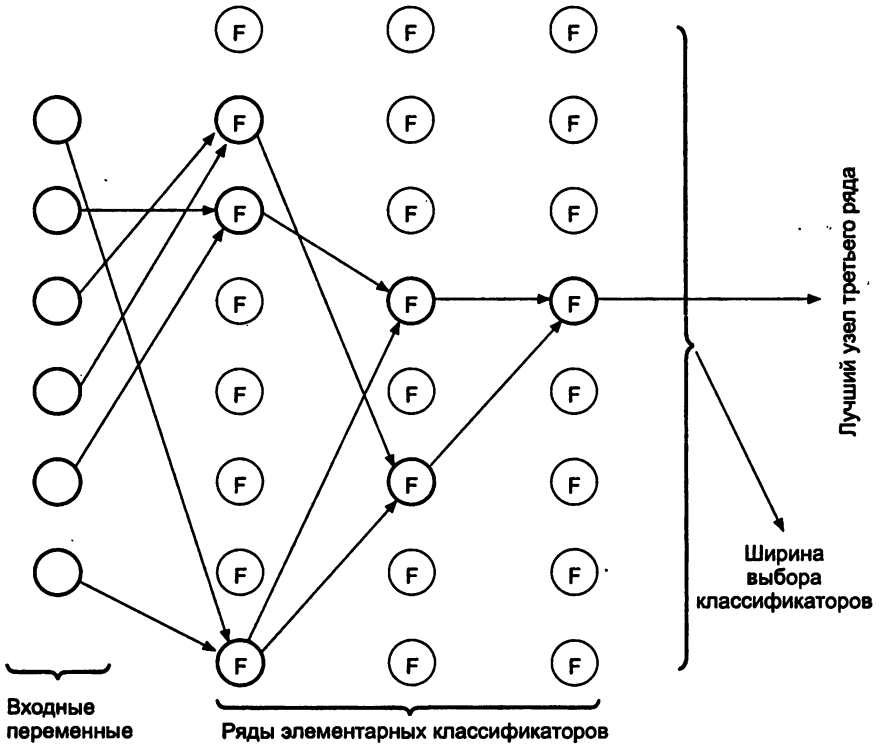
где $s = n^2$, функция f — везде одинаковая.

Каждое уравнение в (3) представляет собой элементарный классификатор. Для определения коэффициентов у элементарных классификаторов используются данные обучающей последовательности. После этого происходит отбор (селекция) некоторого количества лучших по выбранному критерию (о критериях чуть ниже), которые теперь представляют собой входные переменные для следующего ряда классификаторов:

$$z_1 = f(y_1, y_1),\ z_2 = f(y_1, y_2),\ z_3 = f(y_1, y_3),\ \ldots,\ z_{s-1} = f(y_{n-1}, y_n),\ z_s = f(y_n, y_n), \quad (4)$$

Со вторым рядом элементарных классификаторов мы поступаем точно так же, как и с первым, — определяем коэффициенты, определяем качество полученного классификатора, производим отбор. После этого мы можем начать строить третий ряд классификаторов.

Графически данный процесс можно представить следующим образом (рис. 4.7):



Различные варианты частных описаний:

Линейное с ковариациями: $F(x_1, x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x$

Квадратичное: $F(x_1, x_2) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x$

Дробно-полиномиальное: $F(x_1, x_2) = \dfrac{a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2}{1 + b_1x_1 + b_2x_2 + b_3x_1x_2 + b_4x_1^2 + b_5x_2^2}$

Рис. 4.7. Схема построения многорядного итеративного МГУА

Читателям, знакомым с работой многослойных нейронных сетей, нетрудно будет заметить некоторые общие черты МИМГУА и НС:

- анализируемый объект представляет собой «черный ящик»;
- сложные передаточные функции составляются путем соединения большого количества элементарных и сравнительно простых элементов.

Базовые различия между МИМГУА, являющимся одним из наиболее ярких представителей сетей со статистическим обучением (Statistical Learning Networks – SLN), и НС заключаются в следующих принципах:

- кибернетический принцип самоорганизации наиболее объективного создания сети без использования различных субъективных разбиений данных;
- принцип внешнего дополнения, допускающий целевой выбор модели оптимальной сложности;
- принцип регуляризации плохо изложенных задач.

Что это за принципы? Начнем с того, как определяется оптимальная сложность (количество слоев) сетевой структуры МИМГУА. Зачастую качество полученной модели ассоциируется со среднеквадратичной ошибкой между данными, получаемыми при помощи модели, и данными реальной модели. Если мы для проверки будем использовать те же данные, что и при обучении, то согласно теореме «о невозрастании среднеквадратичной ошибки на обучающей последовательности в алгоритмах МГУА» она будет монотонно убывать. Для того чтобы определить оптимальное количество рядов, сверх которого происходит переусложнение модели (говоря языком селекционеров — инцухт, вырождение), нам необходим принцип внешнего дополнения. Оказывается, если мы часть данных не будем использовать для обучения (настройки) элементарных классификаторов, а будем на них подсчитывать среднеквадратичное отклонение реальных данных от получаемой модели, то получим примерно следующую картину (рис. 4.8).

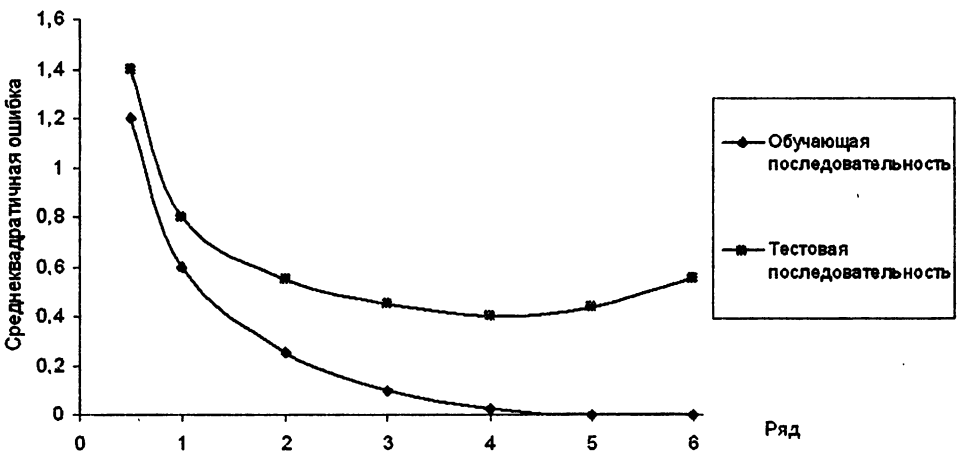


Рис. 4.8. Графики изменения среднеквадратичной ошибки

Естественно предположить, что именно в точке минимума следует остановить наращивание рядов, поскольку последующие ряды уже не прибавят модели точности на данных, на которых она не обучалась. На самом деле в большинстве случаев наращивание рядов желательно прекратить даже немного раньше — точ-

ность от этого практически не страдает, а получаемая модель оказывается проще и более надежно отражает физическую модель объекта.

Однако не только среднеквадратичная ошибка на последовательности, не вошедшей в обучающую серию, имеет подобную особенность (минимум при получении модели оптимальной сложности). В качестве внешнего критерия также может быть использован коэффициент корреляции, максимальная ошибка, средняя ошибка, критерий баланса переменных и много других критериев. Одним из часто применяемых в последнее время является критерий несмещенности коэффициентов:

$$n_{sh} = \frac{\sum_{i=0}^{i=k} (c_i^{*} - c_i^{**})}{k}, \quad n_{sh} \to 0 \qquad (5)$$

где c^* — коэффициенты модели, полученные на первой последовательности, c^{**} — коэффициенты, полученные на второй последовательности (причем, две последовательности могут пересекаться в случае малого количества данных), k — количество параметров. Один и тот же критерий может быть выбран как для отбора самых лучших представителей определенного ряда (слоя), так и для объективного выбора момента для прекращения процесса наращивания сложности модели. Кроме того, в последнее время зачастую также применяется и двухуровневая селекция, когда множество сгенерированных элементарных классификаторов вначале «урезается» при помощи первого, дискриминационного критерия, после чего они сортируются уже при помощи второго критерия.

Процесс выбора критерия представляет собой, пожалуй, самый субъективный момент в работе МГУА, поскольку осуществляется человеком в зависимости от того, какую модель он хочет получить.

Из других процедур, часто применяемых при работе многорядного итеративного МГУА, следует отметить ортогонализацию данных для каждого ряда, а также вариант с протекцией входным переменным. При протекции входным переменным они используются при построении каждого нового слоя наряду с выходными переменными предыдущего слоя. Такой способ, как, впрочем, и ортогонализация, позволяет несколько уменьшить влияние ошибок многорядности. Кроме того, протекция входным переменным позволяет более гладко наращивать сложность модели, что положительно сказывается на поисковых способностях МГУА.

Конечно, достоинства МИМГУА не могут не оборачиваться некоторыми недостатками. Как и любой алгоритм, который использует вместо полного перебора его усеченный вариант, МИМГУА может и не найти правильную модель, однако даже в этом случае обычно получается модель, которая отражает некоторые зависимости эталонного объекта. Риск пропуска оптимальной модели уменьшается при увеличении ширины выбора элементарных классификаторов.

В качестве преимуществ МИМГУА можно назвать высокую скорость обучения, отсутствие зацикливания на локальных аттракторах или при симметричной инициализации весовых коэффициентов (что характерно для нейронных сетей с обратным распространением ошибки). Очень важным свойством МИМГУА яв-

ляется его хорошая способность к «отбрасыванию» малоинформативных входных переменных, что позволяет использовать его как фильтр, выделяющий самые информативные из них для дальнейшей обработки другими алгоритмами [2]. Еще одним свойством МГУА, которое выделяет его из остальных алгоритмов, является способность к построению очень сложных моделей по очень коротким сериям данных, что связано с простотой элементарных функций, которые составляют результирующую передаточную функцию построенной модели.

Спектр алгоритмов и методов МГУА

На сайте разработчиков МГУА [3] можно найти следующую классификацию алгоритмов, использующих одинаковые с МИМГУА принципы в зависимости от имеющихся данных и целей их обработки.

Переменные	МГУА-алгоритмы	
	Параметрические	**Непараметрические**
Непрерывные	Combinatorial (COMBI) — комбинаторный;	Objective Computer Clusterization (OCC) — объективная компьютерная кластеризация;
	Multilayered Iterative (MIA) — многослойный итеративный;	«Pointing Finger» (PF) clusterization algorithm — кластеризация по методу «указательного пальца»;
	Objective System Analysis (OSA) — объективный системный анализ;	Analogues Complexing (AC) — усложняющиеся аналогии
	Harmonical — гармонический;	
	Two-level (ARIMAD) — двухуровневый;	
	Multiplicative-Additive (MAA)	
Дискретные, или бинарные	Harmonical Rediscretization — гармоническая редискретизация	Algorithm on the base of Multilayered Theory of Statistical Decisions (MTSD) — алгоритм, базирующийся на многослойной теории статистических решений

Многослойный итеративный алгоритм был рассмотрен в предыдущем разделе, а сейчас кратко рассмотрим некоторые другие.

Комбинаторный МГУА — COMBI

Исторически COMBI был первым представителем МГУА-алгоритмов (http://inf.kiev.ua/GMDH-home 1). COMBI при генерации модели использует все возможные сочетания входных переменных, вследствие чего наилучшая модель не может быть упущена. Как и у МИМГУА, для выбора оптимальной для данного уровня шумов модели используется критерий селекции. К сожалению, полный

перебор приводит к тому, что на практике практически невозможно строить модели по данным, содержащим более 30 входных переменных [6], в то время как типичные сложные системы, для моделирования и анализа которых и нужны компьютеры, могут иметь до нескольких сотен входных переменных. В связи с этим в 1995 году А. Г. Ивахненко предложил новый вариант комбинаторного МГУА, имеющего, как и МИМГУА, многослойную итеративную структуру.

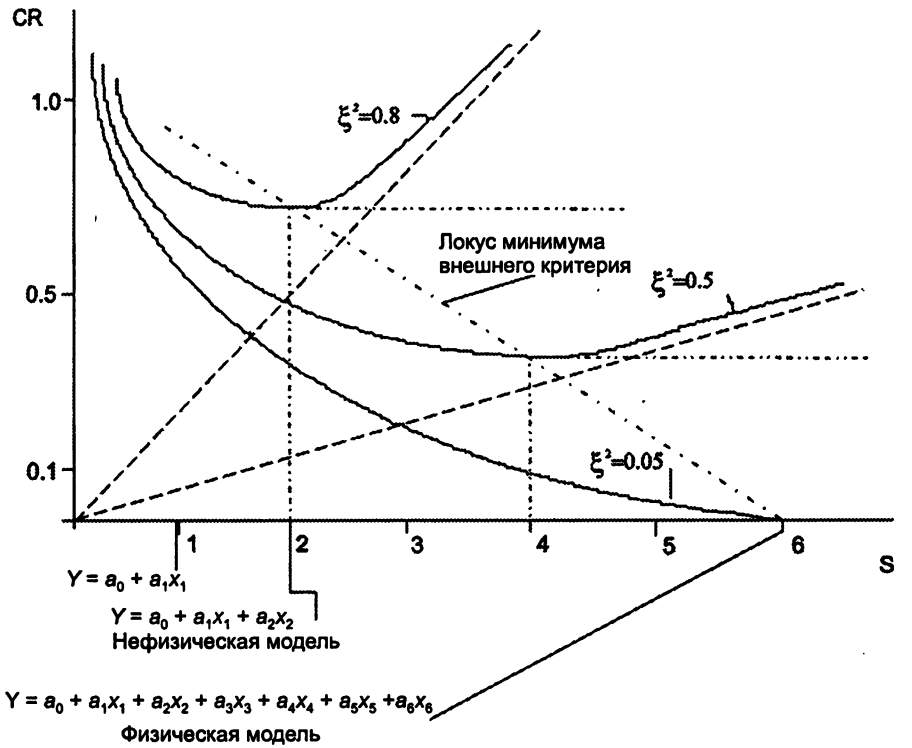


Рис. 4.9. Минимальное значение критерия качества получаемой модели (внешний критерий) и получаемой сложности модели в зависимости от различного уровня шума (помехи) ξ^2

Для того чтобы перебор был полным, нам необходимо получить как можно более гладкий график критерия селекции (рис. 4.9), используемого как индикатор прекращения процесса усложнения модели. Достигается это путем очень плавного наращивания сложности структуры. В первом слое все модели имеют простейшую структуру:

$$y = a_0 + a_1x_i, \quad i = 1, 2, ..., M \tag{6}$$

Здесь M — количество входных переменных. F наилучших моделей остаются после селекции моделей первого уровня. На следующем слое генерируются модели с

более сложной структурой. Они составляются из входных переменных, селектированных на первом уровне:

$$y = a_0 + a_1 x_i + a_2 x_k, \quad i = 1, 2, \ldots, F; \quad j = i+1, \ldots, F; \quad F \le M \qquad (7)$$

После селекции F наилучших моделей, их входных переменных, в этих моделях представленных, составляются новые, еще более сложные модели:

$$y = a_0 + a_1 x_i + a_2 x_k + a_3 x_k, \quad i = 1, 2, \ldots, F; \quad j = i+1, \ldots, F; \quad k = j+1, \ldots, F; \quad F \le M \qquad (8)$$

И так далее, до достижения минимума внешнего критерия. В случае, когда F=M, алгоритм выполняет полный перебор возможных моделей в полиномиальной форме.

Объективная компьютерная кластеризация

Что же такого объективного в данном варианте кластеризации? Дело в том, что в большинстве алгоритмов кластеризации количество кластеров, в которые объединяются наборы входных данных, задается человеком, выполняющим анализ, и задается субъективно. В то же время, если применить обычную для всех МГУА-алгоритмов методику, а именно разделение входных наборов данных на два, то и здесь мы можем получить вполне объективный критерий, позволяющий сказать, какое количество кластеров будет оптимально для конкретных данных при конкретном уровне шумов (помех).

Пусть мы имеем M наблюдений за N входными переменными (или же нам необходимо кластеризовать N объектов по имеющимся M переменным). Для этого мы делим наблюдения на две группы, после чего применяем к ним процедуру иерархического группирования, как показано на схеме (рис. 4.10).

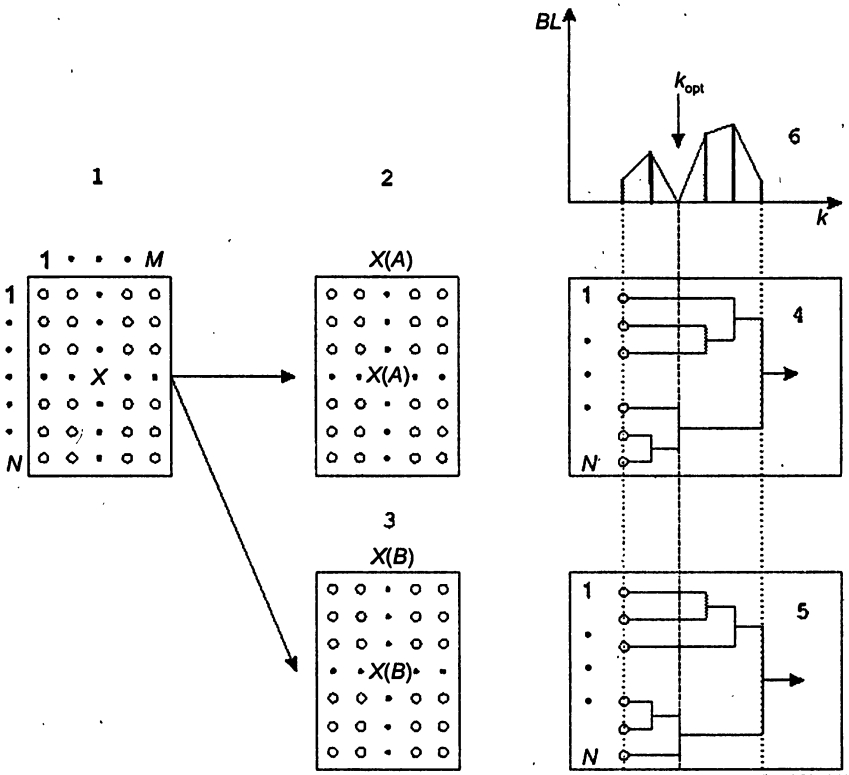
На каждом этапе иерархического группирования данных мы подсчитываем критерий баланса переменных, который для ОСС выглядит следующим образом:

$$BL = \frac{k - \Delta k}{k} \to \min, \qquad (9)$$

где k — количество кластеров, а Δk — количество подобных кластеров.

Где лучше всего использовать ОСС? Как уже было сказано, ОСС оптимальным для данного конкретного набора данных способом подбирает количество кластеров. Что здесь имеется в виду? Реальные объекты можно представить в виде двух частей — детерминированной, для которой мы можем построить вполне определенную модель, которая будет имитировать ее поведение с некоторой точностью, и стохастической части, поведение которой мы можем предсказать только в среднем. Алгоритм ОСС объединяет объекты, различающиеся только «шумовой» составляющей в один кластер, поэтому мы в дальнейшем вполне можем при исследовании сложного объекта, содержащего большое количество переменных, отбросить большинство из них, оставив так называемые репрезентативные переменные, что, наряду с радикальным уменьшением трудоемкости дальнейшего анализа (например, при помощи того же МИМГУА), также приводит к повышению его точности.

Такой подход хорошо показал себя, например, в области моделирования состояния экономики, когда вместо экономических параметров многих фирм используются параметры наиболее характерной (находящейся ближе всего к центру кластера) фирмы [6].



1 — набор входных данных	4, 5 — первое и второе деревья иерархического группирования данных
2, 3 — два подмножества входных данных	6 — график изменения критерия баланса переменных

Рис. 4.10. Схема объективной компьютерной кластеризации (ОСС)

Нейронные сети с активными нейронами

Впервые появившиеся в 1943 году в работе Мак-Каллока и Питта нейроны представляли собой очень простые компоненты, которые могли иметь два или три

состояния. С той поры появилось много других, более сложных конструкций нейронов, каждая из которых имеет свои преимущества и недостатки. Другой проблемой, стоящей при работе с нейронными сетями, является построение оптимальной топологии их соединений.

Одной из возможных конструкций нейрона для случая, когда мы можем указать для него желаемое значение на выходе в зависимости от значений на его входах, является «упакованный» в него параметрический МГУА-алгоритм. Поскольку МГУА-алгоритмы обладают свойством отбрасывания малозначащих входных переменных, что выглядит как изменение топологии связей между слоями, такие нейронные сети называются сетями с активными нейронами.

Примером такой сети является дважды многослойная нейронная сеть (Twice-Multilayered Neural Nets — TMNN). Это обычная многослойная сеть с прямым распространением сигнала, у которой каждый из нейронов «строит» внутри себя одну из многослойных МГУА-структур (МИМГУА, или многослойный COMBI).

Самоорганизованное построение нечетких правил

Еще одним достаточно новым, но при этом интересным вариантом является самоорганизованное построение нечетких правил (Self-organizing Fuzzy Rule Induction). Внешне данный алгоритм очень похож на МИМГУА за исключением того, что элементарные классификаторы работают с нечеткими переменными и реализуют передаточную функцию вида:

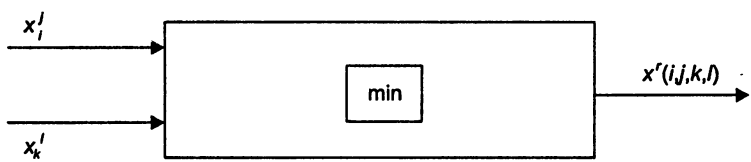


Рис. 4.11. Элементарный классификатор для самоорганизованного нечеткого моделирования

Из показанных на рис. 4.11 элементарных классификаторов по аналогии с МИМГУА составляются слои элементарных классификаторов. В качестве критерия селекции может быть выбран, например, следующий:

$$Q(i,j,k,l)=\sum_{t=1}^{N}(y_t(i,j,k,l)-y_t^r)^2 \quad \text{или} \quad Q(i,j,k,l)=\sum_{t=1}^{N}\left|y_t(i,j,k,l)-y_t^r\right| \qquad (10)$$

Процедура наращивания слоев прекращается тогда, когда критерий селекции начинает увеличиваться по сравнению со своим значением на предыдущем слое.

Описанная схема работает и для случая, когда нечеткие переменные принимают только два значения — «ложь» и «истина».

Литература

1. Вороновский Г. А., Махотило К. В., Петрашев С. Н., Сергеев С. А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. — Харьков: Основа, 1997. В Сети на http://www.neuropower.de/rus/.
2. Ивахненко А. Г., Долгосрочное прогнозирование и управление сложными системами. — Киев: Техника, 1975.
3. Киевский институт кибернетики, домашняя страничка разработчиков МГУА — http://inf.kiev.ua/GMDH-home/.
4. Cramer, N. L. A representation for the adaptive generation of simple sequential programs. //Proceedings of an International Conference on Genetic Algorithms and the Applications. 1985. P. 183–187.
5. Holland J. Adaptation in natural and artificial systems. — Cambridge: MIT Press, MA, 1992.
6. Mueller J.-A., Lemke F. Self-organizing Data Mining. — Berlin, Dresden, 1999.
7. Koza J. R. Hierarchical genetic algorithms operating on populations of computer programs//Proceedings of the Eleventh International Joint Conference on Artificial Intelligence IJCAI-89. Vol. 1, P. 768–774. Morgan Kaufmann, San Francisco, CA,1989.
8. Banzhaf W., Nordin P., Keller R. E. Francone F. D. GENETIC PROGRAMMING. An Introduction. — San Francisco: Morgan Kaufmann Publishers, Inc. 1998.

5 ГЛАВА Обнаружение логических закономерностей в данных

Рассмотрим рис. 5.1. На нем схематично изображены лица людей. Эти лица по каким-то причинам, может быть, важным, разделены на два класса. Ставится задача найти закономерности проведенного разделения.

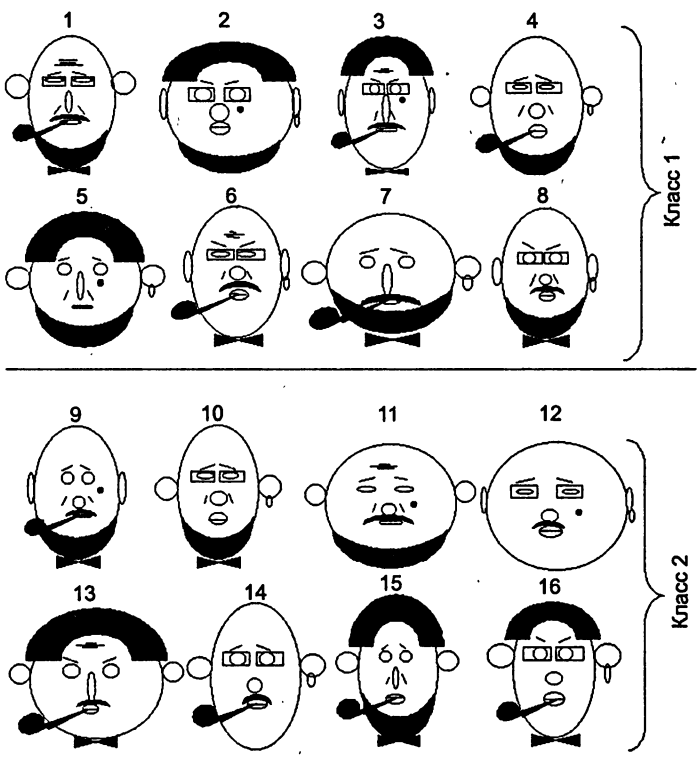


Рис. 5.1. Изображения лиц людей

Попробуйте визуально определить, чем лица разных классов отличаются друг от друга и что объединяет лица одного класса. Сразу заметим — решение существует. Но ваш визуальный анализ, скорее всего, не даст ответа на поставленный вопрос. Обычный человеческий разум не в состоянии решить даже такую, на первый взгляд простую, задачу обнаружения скрытых закономерностей. Здесь необходимо применение компьютерных методов анализа данных.

Можно ли решить задачу обнаружения знаний с помощью классических многомерных методов?

Попытаемся решить поставленную задачу с помощью одного из классических многомерных методов — дискриминантного анализа, содержащегося во всех статистических пакетах. Мы будем здесь использовать пакет STATGRAPHICS *Plus* for Windows.

Прежде всего выделим признаки, характеризующие изображенные лица. Это следующие характеристики:

x_1 (голова) — круглая — 1, овальная — 0;

x_2 (уши) — оттопыренные — 1, прижатые — 0;

x_3 (нос) — круглый — 1, длинный — 0;

x_4 (глаза) — круглые — 1, узкие — 0;

x_5 (лоб) — с морщинами — 1, без морщин — 0;

x_6 (складка) — носогубная складка есть — 1, носогубной складки нет — 0;

x_7 (губы) — толстые — 1, тонкие — 0;

x_8 (волосы) — есть — 1, нет — 0;

x_9 (усы) — есть — 1, нет — 0;

x_{10} (борода) — есть — 1, нет — 0;

x_{11} (очки) — есть — 1, нет — 0;

x_{12} (родинка) — родинка на щеке есть — 1, родинки на щеке нет — 0;

x_{13} (бабочка) — есть — 1, нет — 0;

x_{14} (брови) — подняты кверху — 1, опущены книзу — 0;

x_{15} (серьга) — есть — 1, нет — 0;

x_{16} (трубка) — курительная трубка есть — 1, нет — 0.

Исходная матрица данных, соответствующая изображенным лицам, представлена в табл. 5.1. Строки соответствуют объектам ($N = 16$), столбцы — выделенным бинарным признакам ($p = 16$). Объекты с номерами 1–8 относятся к классу 1, а с номерами 9–16 — к классу 2.

Вводим данные табл. 5.1 в электронную таблицу STATGRAPHICS. Сохраняем их в файле под именем face.

Для проведения дискриминантного анализа выбираем Special ▸ Multivariate Methods ▸ Discriminant Analysis. Получаем окно диалога дискриминантного анализа и вводим в поле Classification Factor (классифицирующий фактор) переменную с именем Class, в поле Data (данные) — переменные x_1–x_{16} (рис. 5.2).

Таблица 5.1. Исходная матрица данных

№ п/п	Голова	Уши	Нос	Глаза	Лоб	Складка	Губы	Волосы	Усы	Борода	Очки	Родинка	Бабочка	Брови	Серьга	Трубка	Class
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	
1	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	1	1
2	1	0	1	1	0	0	1	1	0	1	1	1	0	0	1	0	1
3	0	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1	1
4	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1
5	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1
6	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1
7	1	1	0	1	0	0	0	0	1	1	0	0	1	1	1	1	1
8	0	0	1	1	0	1	1	0	1	1	1	0	1	0	1	0	1
9	0	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1	2
10	0	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0	2
11	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	0	2
12	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1	0	2
13	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	1	2
14	0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	1	2
15	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0	1	2
16	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	1	2

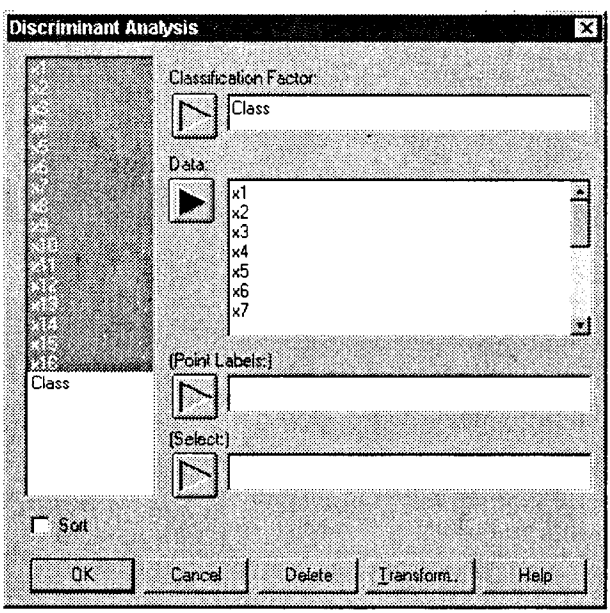


Рис. 5.2. Окно диалога дискриминантного анализа

Нажимаем OK. На экран выдается сводка дискриминантного анализа, в которой сообщается, что анализ не может быть проведен, так как переменные являются линейно зависимыми.

Для преодоления возникшего препятствия щелкнем правой кнопкой мыши и вызовем окно диалога для заданий параметров дискриминантного анализа (рис. 5.3). В поле Fit выберем Backward Selection (метод уменьшения группы признаков).

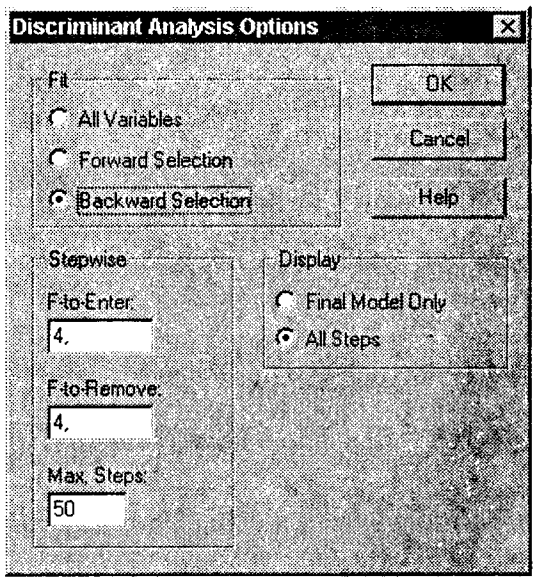


Рис. 5.3. Окно диалога для задания опций дискриминантного анализа

Нажимаем OK. Получаем сообщение, что для продолжения анализа переменная x_9 должна быть удалена. Вызываем окно ввода данных (левая верхняя кнопка) и исключаем эту переменную. Получаем новое сообщение о необходимости удалить из анализа x_{14}. Повторяем операцию по исключению переменной. Затем появляются аналогичные сообщения относительно переменных x_{15} и x_{16}. Исключаем эти переменные. Получаем сводку дискриминантного анализа, проведенного в пространстве признаков x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{10}, x_{11}, x_{12} и x_{13} с применением метода последовательного уменьшения группы признаков.

Нажимаем кнопку табличных опций (вторая слева вверху) и устанавливаем флажки Discriminant Functions (дискриминантные функции) и Classification Table (таблица классификаций). Нажимаем OK. Получаем таблицы, показанные на рис. 5.4 и 5.5.

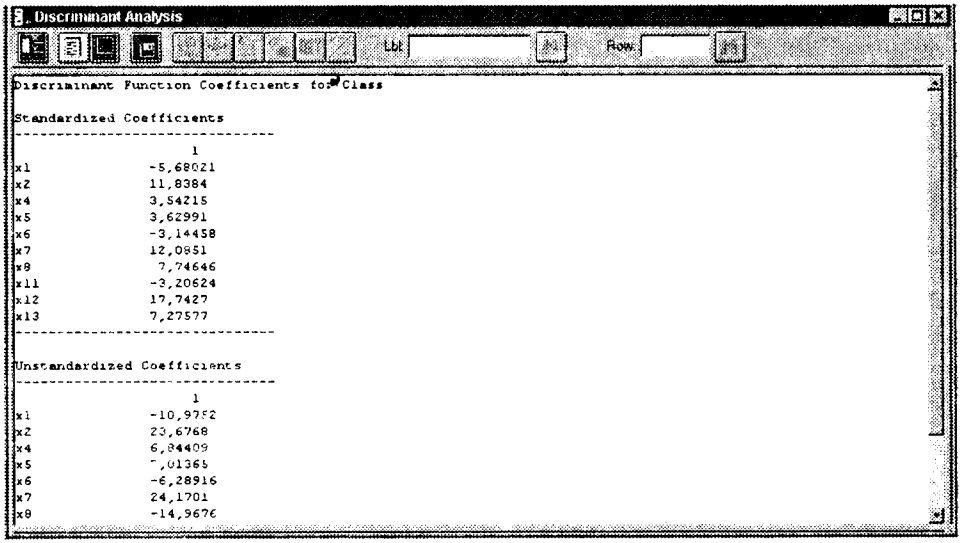


Рис. 5.4. Коэффициенты дискриминантных функций

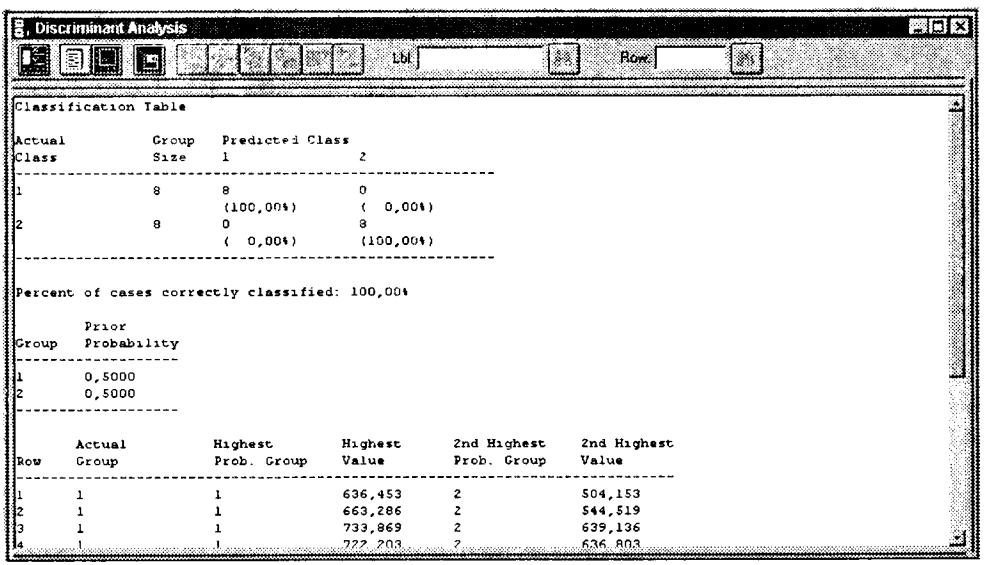


Рис. 5.5. Таблица классификаций

Из таблиц следует, что построена дискриминантная функция, обеспечивающая 100 % правильной классификации исследуемых объектов. Это следующая функция:

$$F = -40{,}1 - 11{,}0 \times x_1 + 23{,}7 \times x_2 + 6{,}8 \times x_4 - 7{,}0 \times x_5 - 6{,}3 \times x_6 + \\ + 24{,}2 \times x_7 - 15{,}0 \times x_8 - 6{,}4 \times x_{11} + 34{,}3 \times x_{12} + 14{,}1 \times x_{13}.$$

На рис. 5.6 приведены гистограммы распределения значений построенной дискриминантной функции в двух сравниваемых группах объектов.

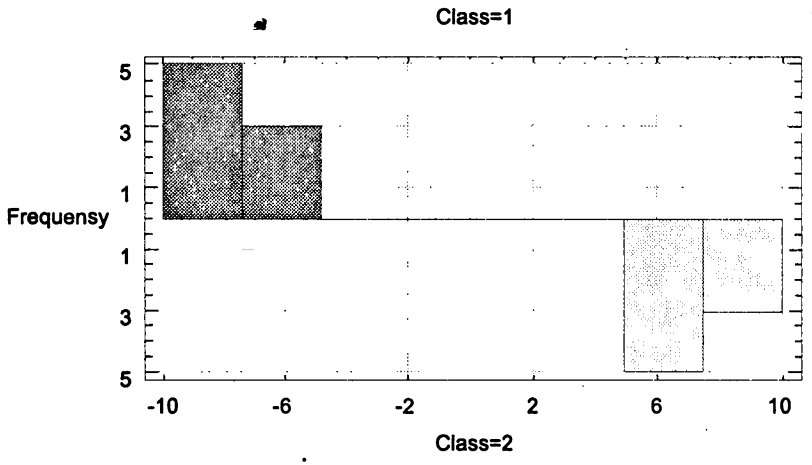


Рис. 5.6. Гистограммы распределения значений дискриминантной функции

Казалось бы, мы достигли желаемой цели — правило классификации построено. Но вряд ли такое правило способно удовлетворить разработчика интеллектуальной системы. Оно формально и не дает нового знания. Глядя на это правило, мы можем лишь перечислить признаки, вошедшие в дискриминантную функцию, и сказать, что данные признаки необходимы для разделения двух классов объектов.

Попытка дать интерпретацию весовым коэффициентам в дискриминантной функции вообще приводит к нелепым результатам. Непонятно, например, почему вес ушей (признак x_2) более чем в три раза превышает вес носа (признак x_3), и т. д.

Отсюда возникает недоверие к построенной дискриминантной функции и растет подозрение, что используемый математический аппарат многомерного анализа «подгоняет» результат.

Так, собственно говоря, и есть. Классическая теория многомерного анализа, являющаяся разделом математической статистики, никогда не претендовала на решение задач, подобных рассмотренной. Но именно такие и гораздо более сложные задачи часто ставит перед нами жизнь.

Основное требование к математическому аппарату обнаружения закономерностей в данных (кроме, конечно, требования эффективности) заключается в интерпретируемости результатов. Правила, выражающие найденные закономерности, должны формулироваться на простом и понятном человеку языке логических высказываний. Например, **ЕСЛИ {(событие 1) и (событие 2) и ... и (событие *N*)} ТО ...** Иными словами, это должны быть логические правила.

Так, классификация лиц в рассмотренном примере может быть произведена с помощью четырех логических правил:

1. **ЕСЛИ** {(голова овальная) и (есть носогубная складка) и (есть очки) и (есть трубка)} **ТО** (Класс 1).

2. **ЕСЛИ** {(глаза круглые) и (лоб без морщин) и (есть борода) и (есть серьга)} **ТО** (Класс 1).
3. **ЕСЛИ** {(нос круглый) и (лысый) и (есть усы) и (брови подняты кверху)} **ТО** (Класс 2).
4. **ЕСЛИ** {(оттопыренные уши) и (толстые губы) и (нет родинки на щеке) и (есть бабочка)} **ТО** (Класс 2).

Математическая запись этих правил выглядит следующим образом:

$$[(x_1 = 0) \wedge (x_6 = 1) \wedge (x_{11} = 1) \wedge (x_{16} = 1)] \vee [(x_4 = 1) \wedge (x_5 = 0) \wedge (x_{10} = 1) \wedge (x_{15} = 1)] \Rightarrow \omega_1,$$

$$[(x_3 = 1) \wedge (x_8 = 0) \wedge (x_9 = 1) \wedge (x_{14} = 1)] \vee [(x_2 = 1) \wedge (x_7 = 1) \wedge (x_{12} = 0) \wedge (x_{13} = 1)] \Rightarrow \omega_2.$$

Здесь значки $\vee$ — коньюнкция (и), $\wedge$ — дизъюнкция (или), $\Rightarrow$ — импликация (если, то).

Под правило 1 подпадают первое, третье, четвертое и шестое лица из первого класса; под правило 2 — второе, пятое, седьмое и восьмое лицо из первого класса; под правило 3 –девятое, одиннадцатое, двенадцатое и четырнадцатое лицо из второго класса; под правило 4 — десятое, тринадцатое, пятнадцатое и шестнадцатое лицо из второго класса.

В этом и следующем разделах вы узнаете, как находить в данных логические закономерности и на их основе строить логические правила.

Логические правила в нашей жизни

Приведенный выше пример, в котором требуется найти правила для классификации человеческих лиц, является сравнительно простым и, можно сказать, «игрушечным». Тем не менее, он дает представление о специфике задачи поиска логических закономерностей в многомерных данных. Подобные задачи являются одними из самых распространенных и полезных для практики.

Логические правила дают возможность прогнозировать и помогают связывать разные стороны жизни в единое целое. Они объясняют связи, которые нередко бывают довольно запутаны. Нет ни одной стороны жизни и области человеческой деятельности, где не применялись бы логические правила. Рассмотрим несколько примеров.

Правила в социологии

Поведение людей в определенных обстоятельствах часто предсказать трудно или невозможно. Но в некоторых случаях социальное поведение все же поддается прогнозу. Объяснения, лежащие в основе прогноза, всегда имеют вид логических правил. Они связывают поступки с мотивами, ориентациями, демографическими характеристиками социальных групп и обстоятельствами жизни.

Правила в экономике и управлении финансами

Какая-то доля рынка непредсказуема. Некоторые специалисты даже говорят, что, например, рынок ценных бумаг — это сфера религии. Но существуют отдельные сегменты рынка, события которых можно уверенно прогнозировать. Это касается как краткосрочного, так и долгосрочного прогнозирования. Примером этого служит популярность многочисленных программных продуктов для управления финансами. Особую ценность представляют системы, использующие логические правила и дающие обоснование своему прогнозу. Это позволяет контролировать принимаемые решения и повышает доверие к ним.

Правила в медицине

Известно много экспертных систем для постановки медицинских диагнозов. Они построены главным образом на основе логических правил. С помощью таких правил узнают не только, чем болен пациент, но и как нужно его лечить. Правила помогают выбирать средства медикаментозного воздействия, определять показания (противопоказания), ориентироваться в лечебных процедурах, создавать условия наиболее эффективного лечения, предсказывать исходы назначенного курса лечения и т. п.

Правила в молекулярной генетике и генной инженерии

Пожалуй, наиболее остро и вместе с тем четко задача обнаружения логических закономерностей стоит в молекулярной генетике и генной инженерии. Здесь она формулируется как определение так называемых маркеров, под которыми понимают генетические коды, контролирующие те или иные фенотипические признаки живого организма. Такие коды могут содержать сотни, тысячи и более связанных элементов.

Можно привести еще много примеров различных областей знания, где логические правила играют ведущую роль. Особенность этих областей заключается в их сложной системной организации. Они относятся главным образом к надкибернетическому уровню организации систем [7], закономерности которого не могут быть достаточно точно описаны на языке статистических или иных аналитических математических моделей [2]. Данные в указанных областях неоднородны, гетерогенны, нестационарны и часто отличаются высокой размерностью.

Точность и полнота правил

Прежде чем перейти к описанию способов поиска логических правил, рассмотрим их общие характеристики.

Будем рассматривать логические правила следующего вида:

$$\textbf{IF}\ \underbrace{(\text{условие 1})\ \text{и}\ (\text{условие 2})\ \text{и}\ \ldots\ (\text{условие } N)}_{\textbf{A}}\ \textbf{THEN}\ \underbrace{(\text{условие } M)}_{\textbf{B}}$$

Примеры условий: $X=C_1$; $X<C_2$; $X>C_3$; $C_4<X<C_5$ и др., где X — какой-либо параметр (поле базы данных), C_i — константы.

Любое правило в виде условного суждения ЕСЛИ (А) ТО (В) имеет две основные характеристики — точность и полноту [5].

Точность правила — это доля случаев, когда правило подтверждается, среди всех случаев его применения (доля случаев В среди случаев А).

Полнота правила — это доля случаев, когда правило подтверждается, среди всех случаев, когда имеет место объясняемый исход В (доля случаев А среди случаев В).

Правила могут иметь какие угодно сочетания значений точности и полноты. Исключение составляет лишь один случай: если точность равна нулю, то равна нулю и полнота (и наоборот).

Примеры правил

Примеры иллюстрируют правила вида ЕСЛИ (А) ТО (В) с различным содержанием А и В. Приведенные примеры демонстрируют четыре правила со значениями точности и полноты, близкими или равными единице либо нулю: 1) точное, но неполное, 2) неточное, но полное, 3) точное и полное, 4) неточное и неполное.

Пример 1. Точное, но неполное правило

Люди смертны (А = «человек», В = «смертен»).

Известно, что все люди смертны. Это значит, что правило «Люди смертны» предельно точное (точность равна единице), оно не имеет исключений. Вместе с тем, среди смертных существ люди составляют весьма скромную долю. Это значит, что полнота правила «Люди смертны» заведомо невелика.

Пример 2. Неточное, но полное правило

Курильщик рано или поздно заболевает раком легких (А = «Курильщик», В = «рано или поздно заболевает раком легких»).

Доля заболевающих раком легких среди курильщиков составляет около 6 %. Это значит, что точность приведенного правила равна примерно 0,06. В то же время, доля курильщиков среди болеющих раком легких составляет 95 %. Таким образом, правило «Курильщик рано или поздно заболевает раком легких» обладает

очень высокой полнотой — 0,95. Часто пропаганда против курения, использующая такого рода правила, делает упор на их полноту, тогда как курильщики ориентируются на точность, которая весьма мала, и продолжают курить, не видя в этом большой угрозы для себя.

Пример 3. Правило точное и полное

В прямоугольном треугольнике из трех углов имеется два, сумма которых составляет прямой угол (A = «прямоугольный треугольник», B = «в треугольнике из трех углов имеется два, сумма которых составляет прямой угол»).

В мире не слишком больших масштабов, где справедлива геометрия Евклида, указанное правило имеет точность, равную единице (среди прямоугольных треугольников все обладают свойством B). Полнота правила также равна единице (среди треугольников, которые обладают свойством B, все прямоугольные).

Пример 4. Правило неточное и неполное

Если у человека родинка на щеке, то он альбинос (A = «человек имеет родинку на щеке», B = «альбинос»).

Среди людей, у которых родинка на щеке, доля альбиносов заведомо невелика. Среди альбиносов также, по всей видимости, не так много имеют родинку на щеке. Это означает, что и точность, и полнота такого правила значительно меньше единицы.

Традиционные методы обнаружения логических закономерностей

Методы поиска логических закономерностей в данных апеллируют к информации, заключенной не только в отдельных признаках, но и в сочетаниях значений признаков. Это одна из причин, по которой классические методы многомерного анализа в ряде случаев, аналогичных рассмотренному выше, не могут конкурировать с логическими методами.

В методах поиска логических закономерностей значения какого-либо признака x_i рассматриваются как элементарные события T. Например, для признаков, измеренных в номинальных шкалах, элементарными событиями называют события $x_i = a$ или $x_i \neq a$, где a — одно из возможных значений x_i. Если же шкала порядковая или количественная, то элементарными событиями могут служить события вида $a < x_i < b$, $x_i < a$, $x_i > a$.

За время развития теории анализа многомерных данных было предложено много различных методов поиска логических закономерностей. Как показала жизнь, большинство из них, в том числе весьма математически изощренные методы, не стали популярными. В настоящее время приоритет принадлежит прагматическим алгоритмам, имеющим прозрачную подоплеку. Можно сказать, что это алгоритмы так называемого здравого смысла.

Алгоритм «Кора»

Алгоритм «Кора» был предложен М. М. Бонгардом [1] в 1967 году. С тех пор за три десятилетия он зарекомендовал себя удачным в ряде прикладных областей. В алгоритме «Кора» анализируются все возможные конъюнкции вида

$$T_{i_1} \wedge T_{i_2} \wedge \ldots \wedge T_{i_l} \quad (l \le l_0),$$

где T — элементарные события, а l_0 — некоторое наперед заданное число (первоначально в алгоритме «Кора» это число было равно трем).

Среди конъюнкций выделяются те, которые характерны (верны на обучающей выборке чаще, чем некоторый порог $1 - \varepsilon_1$) для одного из классов и не характерны для другого (верны реже, чем в доле случаев ε_2). Если коэффициент корреляции между какими-либо двумя выделенными конъюнкциями по модулю более $1 - \varepsilon_3$, то оставляется «наилучшая» из них с точки зрения различения классов, а если конъюнкции эквивалентны, то более короткая (имеющая меньшее l) или просто отобранная ранее. Параметры ε_1, ε_2 и ε_3 подбираются так, чтобы общее число отобранных (информативных) конъюнкций не превосходило некоторого числа n. Чтобы классифицировать новое наблюдение $\mathbf{x}$, для него подсчитывается n_i — число характерных для i-го класса отобранных конъюнкций, которые верны в точке $\mathbf{x}$. Если n_i является максимальным из всех, то принимается решение о принадлежности объекта i-му классу.

Рассмотренный в начале раздела пример по обнаружению закономерностей группировок лиц людей может быть решен с помощью алгоритма «Кора» для $l_0 = 4$ и $\varepsilon_1 > 0{,}5$. Вместе с тем, нужно хорошо представлять, что алгоритм «Кора» является очень трудоемким, так как основан на полном переборе вариантов. Поэтому он хорошо работает только при сравнительно небольших размерностях пространства признаков и невысоких значениях l_0.

Деревья решений

Деревья решений (decision trees) являются самым распространенным в настоящее время подходом к выявлению и изображению логических закономерностей в данных. Видные представители этого подхода — системы CHAID (chi square automatic interaction detection), CART (classification and regression trees) и ID3 (Interactive Dichotomizer — интерактивный дихотомайзер). Частично эти системы уже упоминались в главе 4, посвященной интеллектуальному анализу дан-

ных. Рассмотрим более подробно процедуру построения деревьев решений на примере системы ID3.

В основе системы ID3 лежит алгоритм CLS [6]. Этот алгоритм циклически разбивает обучающие примеры на классы в соответствии с переменной, имеющей наибольшую классифицирующую силу. Каждое подмножество примеров (объектов), выделяемое такой переменной, вновь разбивается на классы с использованием следующей переменной с наибольшей классифицирующей способностью и т. д. Разбиение заканчивается, когда в подмножестве оказываются объекты лишь одного класса. В ходе процесса образуется дерево решений. Пути движения по этому дереву с верхнего уровня на самые нижние определяют логические правила в виде цепочек конъюнкций.

Для иллюстрации работы алгоритма CLS обратимся к примеру по классификации лиц людей (см. рис. 5.1 и табл. 5.1).

На первом шаге алгоритма определяется признак с наибольшей дискриминирующей силой. В нашем случае одинаковой и максимальной силой обладают сразу 7 признаков — x_2, x_3, x_6, x_{10}, x_{11}, x_{14} и x_{15} (табл. 5.2). Поэтому здесь мы принимаем волевое решение и назначаем первым признаком, например, x_6.

Таблица 5.2. Отношение единиц (1) в разных классах объектов для разных признаков

Признаки	x1	**x2**	**x3**	x4	x5	**x6**	x7	x8	x9	**x10**	**x11**	x12	x13	**x14**	**x15**	x16
Класс 1/Класс 2	3/3	**4/6**	**4/6**	5/5	3/3	**6/4**	4/6	3/3	5/5	**6/4**	**6/4**	3/3	5/5	**4/6**	**4/6**	5/5

От первого признака отходят две ветви. Первая для значения x_6=0, а вторая — x_6= 1. В табл. 5.3 и 5.4 содержатся данные, соответствующие этим ветвям.

Таблица 5.3. Таблица данных, соответствующая ветви x_6 = 0

	Признаки															
Объекты	x1	x2	x3	x4	x5	x6	x7	X8	x9	**x10**	x11	x12	x13	x14	x15	x16
2	1	0	1	1	0	0	1	1	0	**1**	1	1	0	0	1	0
7	1	1	0	1	0	0	0	0	1	**1**	0	0	1	1	1	1
12	1	0	1	0	1	0	1	0	1	**0**	1	1	0	1	1	0
13	1	1	0	1	1	0	1	1	1	**0**	0	0	1	0	0	1
14	0	1	1	1	0	0	1	0	1	**0**	1	0	0	1	1	1
16	0	1	1	1	0	0	1	1	0	**0**	1	0	1	0	1	1
Класс 1/Класс 2	2/2	1/3	1/3	2/3	2/2	2/4	1/4	1/2	1/3	**2/0**	1/3	1/1	1/2	1/2	2/3	1/3

Для ветви x_6 = 0 окончательное решение дает признак x_{10}. Он принимает значение 1 на объектах 2 и 7 из первого класса и значение 0 на объектах 12, 13, 14 и 16 из второго класса.

Таблица 5.4. Таблица данных, соответствующая ветви $x_6 = 1$

	Признаки															
Объекты	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
1	0	1	0	0	1	1	0	0	1	1	1	0	1	1	0	1
3	0	0	0	1	1	1	0	1	1	0	1	1	1	0	0	1
4	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
5	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1	0
6	0	0	1	0	1	1	1	0	1	0	1	0	1	0	1	1
8	0	0	1	1	0	1	1	0	1	1	1	0	1	0	1	0
9	0	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1
10	0	1	1	0	0	1	1	0	0	1	1	0	1	1	1	0
11	1	1	1	0	1	1	0	0	1	1	0	1	0	1	0	0
15	0	1	0	1	0	1	1	1	0	1	0	0	1	1	0	1
Класс 1/Класс 2	1/1	3/3	3/3	3/2	3/1	6/4	3/2	2/1	4/2	4/4	5/1	2/2	4/3	3/4	4/1	4/2

Ветвь $x_6 = 1$ устроена более сложно. На этой ветви наибольшей дискриминирующей силой обладает признак x_{11} (табл. 5.4). Он имеет значение 0 у объекта 5 из первого класса и объектов 9, 11, 15 из второго класса; и значение 1 у объектов 1, 3, 4, 6 из первого класса и объекта 10 из второго класса. Таким образом, требуется дополнительное ветвление, которое осуществляется с помощью признаков x_{15}, x_{16} и x_2. Полное дерево приведено на рис. 5.7.

Как следует из рисунка, дерево логического вывода, выросшее из признака x_6 (носогубная складка), имеет 6 исходов. Только два из этих исходов включают по четыре объекта (полнота 4/8). Один исход группирует три объекта своего класса (полнота 3/8), один исход — два объекта (полнота 2/8) и три исхода включают по одному объекту (полнота 1/8). Естественно, нельзя считать логической закономерностью путь по дереву, для которого исход захватывает столь малое относительное число объектов одного класса (обладает малой полнотой). Как видим, решение, даваемое алгоритмом CLS, далеко от требуемого.

Если попытаться вырастить дерево решений из любого другого признака, например, x_2, x_3, x_{10}, x_{11}, x_{14} или x_{15}, то результат также будет далек от оптимального.

Алгоритм CLS способен приводить к качественным решениям задачи поиска логических закономерностей только в случае независимых признаков. В противном случае он нередко направляет ход логического вывода по ложному пути и создает лишь иллюзию правильного рассуждения.

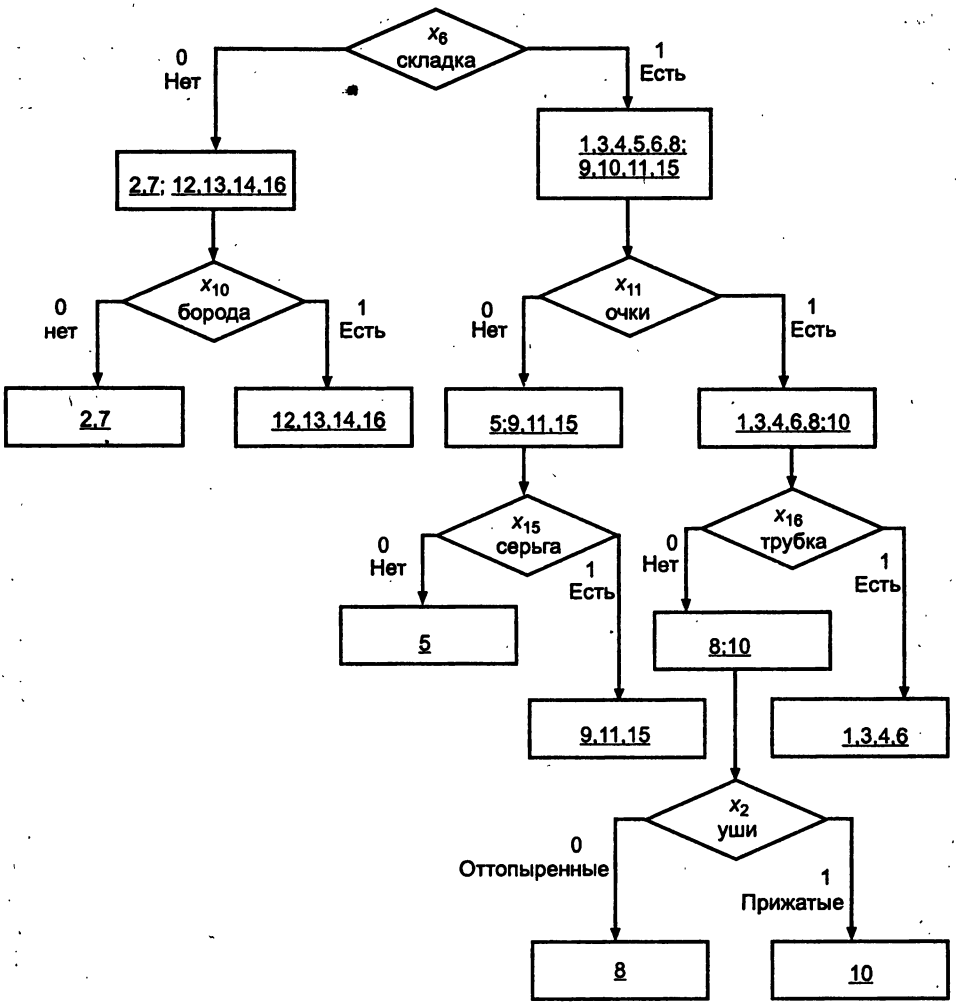


Рис. 5.7. Дерево логического вывода

Случайный поиск с адаптацией

Алгоритм случайного поиска с адаптацией (СПА) был предложен Г. С. Лбовым в 1965 году для работы в условиях зависимых признаков [4]. Он заключается в следующем.

Имеется множество возможных событий $\mathrm{T} = \{T_i\}$, $i = \overline{1, p}$. Из этого множества требуется отобрать цепочки конъюнкций $T_{i_1} \wedge T_{i_2} \wedge \ldots \wedge T_{i_l}$ заданной длины l, максимизирующие некоторый критерий J.

Прежде всего, проводится серия опытов по случайному определению состава цепочек конъюнкций. Затем для этих цепочек вычисляются значения критерия J. Цепочка с максимальным значением критерия поощряется увеличением вероятности выбора вошедших в нее событий в следующих опытах. Цепочка с наименьшей величиной критерия наказывается соответствующим образом. Вся процедура повторяется до тех пор, пока события отчетливо не поляризуются по вероятности их выбора для испытаний.

Алгоритм СПА избегает полного перебора событий, требующего просмотра $Q = C_p^l$ их комбинаций в цепочке. Его трудоемкость, однако, зависит от задаваемых условий: количества испытаний, мер поощрения и наказания событий, — которые существенно влияют на скорость сходимости алгоритма. Экспериментально на одних и тех же данных показано, что алгоритм СПА давал оптимальное либо близкое к оптимальному решение за число шагов, сравнимое с величиной [3]:

$$Q = \sum_{i=1}^{l} (p - i).$$

Пример с обнаружением логических закономерностей в группах изображений лиц людей (см. рис. 5.1) может быть решен с помощью алгоритма СПА, если задать $l = 4$. Вместе с тем, этот пример является достаточно простым — число объектов и признаков в таблице данных мало. В более сложных ситуациях алгоритм СПА остается все-таки весьма трудоемким. Кроме того, необходимо задавать длину цепочек конъюнкций l, что также ограничивает его применение.

Инструментальные средства обнаружения знаний в данных

Рынок программных продуктов в области Data Mining и Knowledge Discovery бурно развивается. Фактически каждый месяц в Интернете появляются анонсы новых инструментов для обнаружения знаний в базах. Помочь разобраться в том, какой из продуктов является наиболее подходящим, призваны специальные обзорные страницы Интернета (например, http://www.kdnuggest.com), на которых приводятся каталоги разработок, рассказывается о фирмах-разработчиках, ведутся дискуссии, сравниваются характеристики различных программ и т. д.

В данном разделе приводится описание двух систем для обнаружения знаний в данных. Первая система — See5 — относится к наиболее представительному и популярному направлению, связанному с построением деревьев решений. А вторая система — WizWhy — интересна тем, что ее разработчики утверждают, будто она способна обнаружить **ВСЕ** if-then-правила в данных. Это утверждение подкрепляется сообщением о весьма большом количестве коммерческих структур, использующих WizWhy (более 30 000). Давайте сами убедимся в полезных свойствах предлагаемых инструментов.

Построение деревьев решений — система See5/C5.0

Система See5/C5.0 (Windows 95/98/NT) компании RuleQuest (http://www.rulequest.com) предназначена для анализа больших баз данных, содержащих до сотни тысяч записей и до сотни числовых или номинальных полей. Результат работы See5 выражается в виде деревьев решений и множества if-then-правил. Система проста в обращении и не требует от пользователя специфических знаний в области прикладной статистики. Стоимость See5 — $740, некоммерческая версия для обучения ограничена количеством анализируемых записей (до 200).

Проиллюстрируем процесс работы See5 на реальном примере из области медицинской диагностики. Исходные данные в рассматриваемом случае относятся к задаче дифференциальной диагностики заболеваний почек. Данные были получены в Российской медицинской академии (Хитрова А. Н. Дифференциальная диагностика кист почечного синуса и гидронефрозов методом комплексного ультразвукового обследования: Диссертация на соискание ученой степени кандидата медицинских наук. Москва, 1996).

Фрагмент исходных данных приведен в табл. 5.5. Это как раз тот вид данных, для обработки которых более всего подходит See5. Каждый объект (пациент) здесь принадлежит к одному из небольшого числа классов (здоров, множественные кисты, гидронефроз) и описывается одиннадцатью разнотипными признаками.

Задача See5 состоит в предсказании диагностического класса какого-либо объекта по значениям его признаков. При этом, как мы увидим, See5 конструирует классификатор в виде дерева решений, которому, в свою очередь, может быть поставлено в соответствие некоторое множество логических правил.

Таблица 5.5. Фрагмент исходных данных по дифференциальной диагностике заболеваний почек

Признак	Объект 1	Объект 2	...
Состояние почки *diagnosis*	Множественные кисты	Гидронефроз	...
Возраст пациента (число полных лет) *Age*	46	52	...
Пол пациента *Sex*	Женщина (F)	Мужчина (M)	...
Правая или левая почка *LR*	Правая почка (R)	Левая почка (L)	...
Длина почки (мм) *Length*	112	136	...
Ширина почки (мм) *Width*	68	69	...
Толщина почки (мм) *Thickness*	88	72	...
Толщина паренхимы (мм) *Thickpar*	18	18	...
Средняя скорость кровотока (см/с) *Speed*	2,3	12	...
Индекс резистентности *Index*	0,584	0,614	...
Ускорение артериального потока в систолу ($см/с^2$) *Accel*	459	291	...

Подготовка данных для See5

Каждой задаче, решаемой в системе See5, требуется присвоить свое собственное имя. Пусть в нашем случае это имя будет USR (UltraSonic Research). В процессе решения See5 использует и формирует несколько файлов с одинаковым именем и различными расширениями. Важно точно соблюдать правила записи имен и расширений (система различает строчные и прописные буквы). Кроме того отметим, что See5 поддерживает только латинские шрифты.

Файл имен переменных

Для работы See5 самыми необходимыми и существенными являются два файла — имен переменных и данных. В файле имен переменных с расширением *.names даются названия используемых признаков и классов.

Среди признаков различают две важные подгруппы:

- номинальные признаки (discrete attribute), количественные признаки (continuous attribute) и метки;
- явно определенные признаки, значения которых берутся непосредственно из файла данных, и неявно определенные признаки, задаваемые формулами (чаще всего употребляются явно определенные признаки).

Файл имен переменных USR.names в нашей задаче выглядит следующим образом:

```
diagnosis.          | the target attribute
diagnosis:          1, 2, 3
Age:                continuous
Sex:                F, M
LR:                 L, R
Length:             continuous
Width:              continuous
Thickness:          continuous
Thickpar:           continuous
Speed:              continuous
Index:              continuous
Accel:              continuous
```

Целевой признак **diagnosis** принимает три значения: 1 — в классе «здоровая почка»; 2 — в классе «множественные кисты» и 3 — в классе «гидронефроз». Признаки **Age** (возраст), **Length** (длина почки), **Width** (ширина почки), **Thickness** (толщина почки), **Thickpar** (толщина паренхимы), **Speed** (средняя скорость кровотока), **Index** (индекс резистентности) и **Accel** (ускорение артериального потока в систолу) являются количественными. Признак **Sex** (пол пациента) может иметь два значения F (female) и M (male), а признак **LR** (левая или правая почка) принимает значения L или R. Порядок записи имен переменных должен соответствовать их порядку в файле данных.

При подготовке файла имен переменных следует иметь в виду, что пробелы, пустые строки и знаки табуляции игнорируются системой (кроме, конечно, случаев, когда они применяются в именах переменных). Вертикальная черта «|» предназначена для записи напоминаний или комментариев.

После имени каждой явно определенной переменной вставляется двоеточие «:», а затем следует характеристика этой переменной. Возможны следующие характеристики:

- continuous — количественный признак;
- список значений переменной, разделенных запятой (для дискретной, номинальной переменной);
- максимальное значение N для дискретной переменной (эту характеристику рекомендуется применять очень осторожно, так как здесь исключается дополнительная проверка данных при их вводе в анализ);
- ignore — для признака, исключаемого из анализа;
- label — метка для идентификации отдельного объекта.

После имени каждой неявно определенной переменной также следует двоеточие и далее записывается формула. В формуле используются, где необходимо, скобки, а дискретные признаки ограничиваются кавычками. Ниже приведены доступные операторы:

- +, -, *, /, % (mod), ^ (возведение в степень);
- >, >=, <, <=, =, <> или != (не равно);
- and, or;
- sin(...), cos(...), tan(...), log(...), exp(...), int(...) (целая часть от).

В зависимости от применяемой формулы конечный результат может быть как количественным, так и давать логическое значение **true/false**.

Файл данных

Вторым файлом, необходимым для работы See5, является файл данных. Он имеет расширение *.data. В нашем случае это файл USR.data.

Каждому объекту в файле данных соответствует собственная строка. Если значение целевой переменной находится вверху файла имен переменных, строка начинается со значения этой целевой переменной. Затем через запятую следуют значения всех остальных признаков. Неизвестные значения переменных кодируются вопросительным знаком «?», после вертикальной черты «|» можно писать невоспринимаемые системой комментарии.

Ниже приводится полностью весь файл данных USR.data, который мы будем использовать для демонстрации возможностей See5.

```
1, 62, F, R, 127, 52, 43, 14, 13.3, 0.698, 140
1, 43, M, L, 103, 44, 49, 15, 16.3, 0.634, 291
1, 58, M, R, 103, 58, 46, 17, 16.5, 0.704, 143
1, 37, M, L, 112, 53, 51, 18, 18.2, 0.562, 189
1, 21, M, L, 126, 62, 45, 14, 18.5, 0.613, 116
1, 74, M, R, 115, 57, 49, 16, 19.1, 0.69, 85
1, 62, M, R, 103, 66, 45, 16, 19.2, 0.657, 65
1, 43, M, R, 104, 54, 46, 14, 19.3, 0.574, 629
```

1, 34, F, L, 110, 52, 42, 19, 19.3, 0.686, 152
1, 68, F, R, 112, 52, 42, 17, 19.4, 0.593, 258
1, 37, M, R, 119, 51, 41, 14, 19.8, 0.65, 101
1, 38, M, L, 107, 59, 56, 12, 20.9, 0.572, 279
1, 67, F, R, 113, 57, 47, 17, 20.9, 0.681, 115
1, 46, F, L, 107, 60, 36, 15, 21.8, 0.678, 352
1, 67, F, R, 135, 70, 67, 27, 24, 0.583, 51
1, 52, M, L, 111, 59, 50, 21, 26.6, 0.644, 379
1, 42, M, R, 82, 47, 42, 14, 26.8, 0.651, 538
1, 47, F, L, 114, 63, 51, 17, 27.8, 0.645, 589
1, 23, F, L, 128, 39, 56, 21, 30.6, 0.62, 255
1, 35, F, L, 114, 50, 41, 14, 31.2, 0.622, 445
1, 64, F, R, 97, 57, 39, 16, 34.7, 0.68, 368
1, 56, M, L, 125, 74, 72, 2, 46.6, 0.638, 685
2, 46, F, R, 112, 68, 88, 18, 2.3, 0.584, 459
2, 58, M, L, 129, 67, 58, 18, 12.5, 0.686, 151
2, 69, F, R, 115, 69, 44, 14, 13.2, 0.657, 231
2, 69, M, L, 126, 59, 49, 13, 14.1, 0.652, 282
2, 54, M, R, 98, 87, 41, 24, 14.3, 0.637, 352
2, 67, M, R, 111, 59, 47, 18, 14.6, 0.742, 242
2, 70, F, R, 108, 58, 37, 11, 14.6, 0.663, 139
2, 67, M, L, 129, 64, 58, 17, 15.2, 0.693, 382
2, 55, M, L, 125, 59, 48, 18, 15.4, 0.674, 330
2, 65, F, R, 111, 54, 51, 13, 15.4, 0.727, 257
2, 70, F, L, 108, 65, 51, 14, 16.3, 0.724, 250
2, 63, F, R, 121, 67, 55, 16, 16.4, 0.653, 148
2, 56, F, R, 99, 55, 47, 16, 16.8, 0.693, 103
2, 60, F, R, 105, 56, 46, 14, 17.7, 0.526, 219
2, 54, M, L, 107, 57, 43, 14, 17.9, 0.651, 254
2, 48, F, R, 100, 56, 44, 16, 18, 0.667, 114
2, 74, M, L, 104, 56, 56, 16, 18.2, 0.643, 88
2, 62, F, L, 118, 56, 41, 14, 18.3, 0.611, 347
2, 60, F, L, 108, 56, 50, 14, 18.6, 0.546, 216
2, 63, F, L, 110, 61, 56, 14, 19, 0.645, 216
2, 64, M, R, 123, 61, 57, 20, 19.1, 0.632, 173
2, 54, M, R, 105, 58, 43, 16, 21, 0.62, 230
2, 47, F, L, 116, 73, 56, 17, 21.4, 0.636, 178
2, 47, F, L, 109, 58, 48, 11, 21.5, 0.544, 266
2, 66, F, R, 98, 64, 56, 19, 22.3, 0.655, 111
2, 67, F, L, 103, 44, 42, 14, 22.6, 0.682, 656
2, 54, M, L, 119, 50, 48, 20, 23.5, 0.65, 188
2, 70, F, R, 105, 56, 41, 13, 23.5, 0.663, 242
2, 66, F, L, 111, 61, 50, 16, 24.7, 0.689, 189
2, 56, F, L, 99, 58, 47, 16, 25, 0.686, 196
2, 47, F, R, 99, 62, 50, 12, 26.2, 0.544, 235
2, 69, F, L, 125, 66, 51, 18, 26.2, 0.667, 275
2, 37, M, L, 120, 89, 61, 16, 27.9, 0.566, 218
2, 48, F, L, 113, 77, 49, 22, 30.5, 0.686, 210
2, 45, F, L, 115, 56, 52, 17, 34.2, 0.587, 410
2, 70, F, L, 108, 48, 42, 19, 36.1, 0.69, 219
2, 67, F, L, 122, 72, 64, 20, 43.3, 0.674, 229

```
3, 42, M, L, 104, 69, 51, 24, 14.5, 0.729, 211
3, 21, M, R, 144, 101, 49, 6, 16.3, 0.707, 194
3, 67, F, L, 99, 52, 52, 8, 16.3, 0.744, 332
3, 34, F, R, 105, 54, 46, 19, 19.4, 0.704, 98
3, 47, F, R, 116, 85, 64, 24, 19.5, 0.804, 416
3, 38, M, R, 118, 45, 60, 12, 19.9, 0.701, 354
3, 56, M, R, 118, 55, 43, 22, 21, 0.735, 165
3, 37, M, R, 114, 64, 52, 18, 21.8, 0.717, 225
3, 62, M, L, 134, 67, 53, 16, 23, 0.76, 321
3, 68, F, L, 106, 48, 46, 18, 24.5, 0.693, 224
3, 43, M, L, 136, 65, 57, 22, 25.6, 0.731, 351
3, 35, F, R, 124, 41, 67, 20, 26.3, 0.692, 286
3, 23, F, R, 127, 78, 62, 17, 39.2, 0.714, 349
3, 52, M, R, 125, 59, 44, 27, 39.3, 0.703, 545
3, 67, F, R, 114, 55, 36, 9, 41.5, 0.73, 310
```

Файлы тестовых данных (необязательные)

Для проверки качества построенного дерева решений и соответствующего множества логических правил в системе See5 предусмотрена возможность работы со специальными файлами, в которых содержатся дополнительные тестовые данные.

Третий вид файла, используемый системой See5, содержит новые тестовые объекты. Это то, что еще принято называть контрольной выборкой. Данный файл USR.test является необязательным и, если используется, имеет формат уже описанного файла USR.data.

Следующий вспомогательный файл USR.cases также является необязательным. Он содержит объекты с неизвестной классификацией.

Файл стоимости

Последний вид файла, обозначаемый USR.costs, содержит информацию о стоимости различных ошибок классификации. Заполнение этого файла является необязательным. Вместе с тем, назначение штрафов за ошибки может оказаться весьма полезным при разработке некоторых приложений.

Интерфейс пользователя

В главном окне See5 располагается пять кнопок (рис. 5.8). Перечислим их слева направо.

С помощью кнопки Locate Data (местонахождение данных) вызывается окно для просмотра доступных файлов данных и их загрузки в систему.

Нажатием кнопки Construct Classifier (построение классификатора) производится обращение к окну диалога для выбора типа классификатора и установки его параметров.

Кнопка Stop предназначена для останова процеса построения дерева решений.

Кнопка Use Classifier (использование классификатора) запускает процесс интерактивной классификации одного или более объектов.

С помощью кнопки Cross-Reference (перекрестная ссылка) вызывается окно, в котором наглядно раскрываются связи между объектами обучающей выборки и найденными правилами их классификации.

Все перечисленные функции доступны также из меню File. В свою очередь, в меню Edit предоставляется возможность редактирования файла имен данных и файла стоимости ошибок классификации.

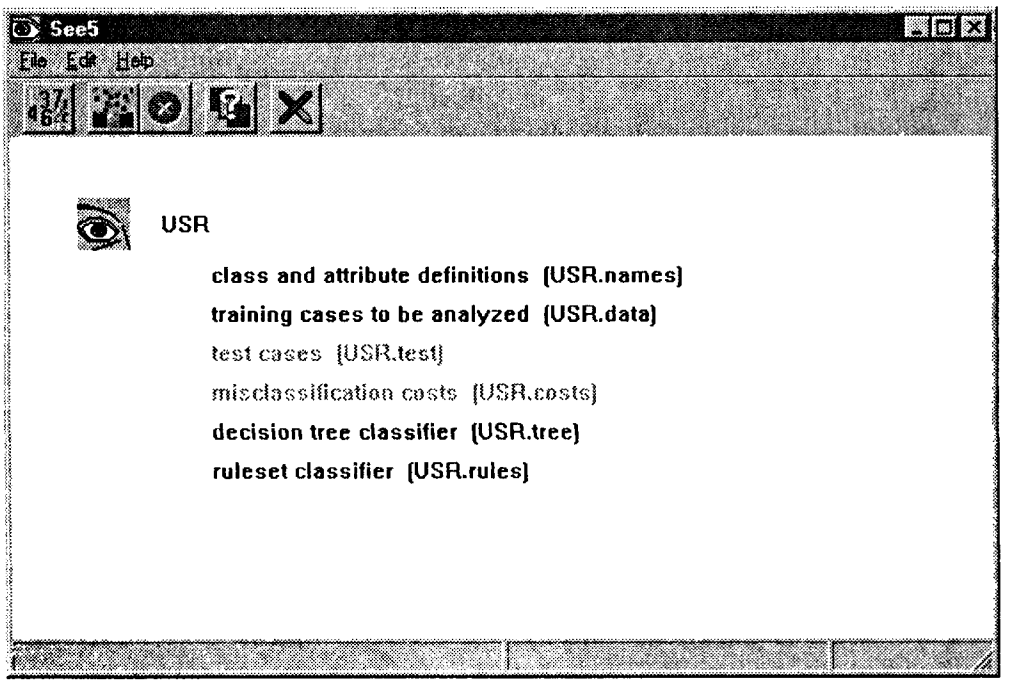


Рис. 5.8. Главное окно системы See5

Построение дерева решений

На первом этапе обработки данных обычно используются параметры системы, установленные по умолчанию. Нажимаем кнопку Construct Classifier и затем в появившемся окне диалога (рис. 5.9) сразу нажимаем OK (предполагается, что файл данных USR.data уже загружен). Система выдает окно результатов, которые выглядят следующим образом (рис. 5.10).

В первой строке отчета о результатах дается информация об используемой версии системы See5 и текущее время. Затем в следующих двух строках говорится о том, что классифицирующей переменной служит **diagnosis** и прочтенный файл данных USR.data содержит 74 объекта, каждый из которых описан одиннадцатью признаками.

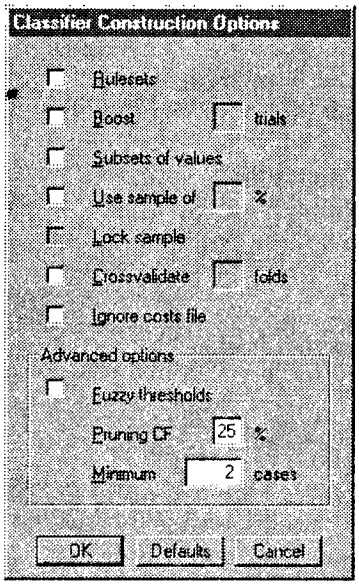


Рис. 5.9. Окно диалога для задания параметров алгоритма конструирования классификатора

В следующих строках отчета отображено построенное дерево решений. Его можно проинтерпретировать следующим образом:

ЕСЛИ **Index** больше 0,69 и **Speed** больше 18, ТО класс № 3, иначе

ЕСЛИ **Index** больше 0,69 и **Speed** не больше 18 и **Thickness** не больше 46, ТО класс № 1 и т. д.

Каждая ветка дерева заканчивается указанием номера класса, к которому она приводит. Сразу за номером следует запись вида (*n*) или (*n/m*). Например, самая первая ветка заканчивается записью (12,0). Это означает, что данной ветке соответствует 12 объектов из определенного (третьего) класса. Последняя ветка заканчивается записью 1 (6,0/1,0), из чего следует, что эта ветка описывает класс № 1 и сюда попадают 6 объектов, из которых 1 попадает ошибочно. Величины *n* или *m* могут оказаться дробными в случае, когда на какую-либо ветку придется некоторое число объектов с неизвестными значениями признаков.

В следующем разделе отчета приводятся характеристики сконструированного классификатора, оцениваемые на обучающей выборке. Здесь мы видим, что построенное дерево решений имеет 9 веток (size = 9), а ошибка классификации наблюдается на 5 объектах, что составляет 6,8 %.

В завершающей части отчета дается таблица с детальным разбором результатов классификации. Исходя из данных этой таблицы, можно сказать, что из 1-го класса (здоровые почки) правильно классифицируются 20 объектов, а 2 объекта ошибочно относятся к классу 2; среди объектов 2-го класса (множественные кисты) 35 диагностируются правильно и 2 ошибочно признаются здоровыми; все объекты 3-го класса (гидронефроз) классифицируются правильно за исключением одного объекта, попадающего в класс № 2.

```
See5 INDUCTION SYSTEM [Release 1.10]      Thu Apr 15 12:15:18 1999
-----------------------------------
Class specified by attribute 'diagnosis'
Read 74 cases (11 attributes) from USR.data
Decision tree:

Index > 0.69:
:...Speed > 18: 3 (12.0)
:   Speed <= 18:
:   :...Thickness <= 46: 1 (2.0)
:       Thickness > 46:
:       :...Age <= 48: 3 (2.0)
:           Age > 48: 2 (6.0/1.0)
Index <= 0.69:
:...Age <= 43: 1 (11.0/1.0)
    Age > 43:
    :...Accel <= 85: 1 (3.0)
        Accel > 85:
        :...Accel <= 349: 2 (29.0/2.0)
            Accel > 349:
            :...Index <= 0.637: 2 (3.0)
                Index > 0.637: 1 (6.0/1.0)


Evaluation on training data (74 cases):

          Decision Tree
        ----------------
        Size      Errors

           9    5( 6.8%)   <<


         (a)   (b)   (c)    <-classified as
        ----  ----  ----
          20     2          (a): class 1
           2    35          (b): class 2
                 1    14     (c): class 3


Time: 0.5 secs
```

Рис. 5.10. Результаты построения начального дерева решений

В заключение система See5 выдает сообщение о затраченном на решение времени. В нашем случае оно составило 0,5 с. Здесь надо отметить вообще очень высокую скорость работы алгоритма See5, позволяющую оперативно обрабатывать высокоразмерные массивы информации, содержащие тысячи и десятки тысяч записей.

Можно еще более подробно разобрать результаты нашей классификации. Для этого нажмем в главном окне See5 кнопку Cross-Reference (перекрестная ссылка). Система выдаст окно, в левой половине которого нарисовано построенное дерево решений, а в правой половине перечисляются объекты, попавшие на ту или

иную ветвь дерева. Чтобы выделить интересующую ветвь, нужно щелкнуть по ней левой кнопкой мыши (справа от ветви появится темный круг — на рис. 5.11 на него указывает стрелка). Кроме того, если щелкнуть мышью по номеру какого-либо объекта из правого поля, то система выдаст еще одно окно с именем Case, в котором приводятся значения признаков и выделенного объекта. В случае, показанном на рисунке, нас заинтересовала ветвь (**Index** <=0.69 и **Age** <=43), на которой находятся 10 объектов из 1-го класса и 1 объект из 2-го класса.

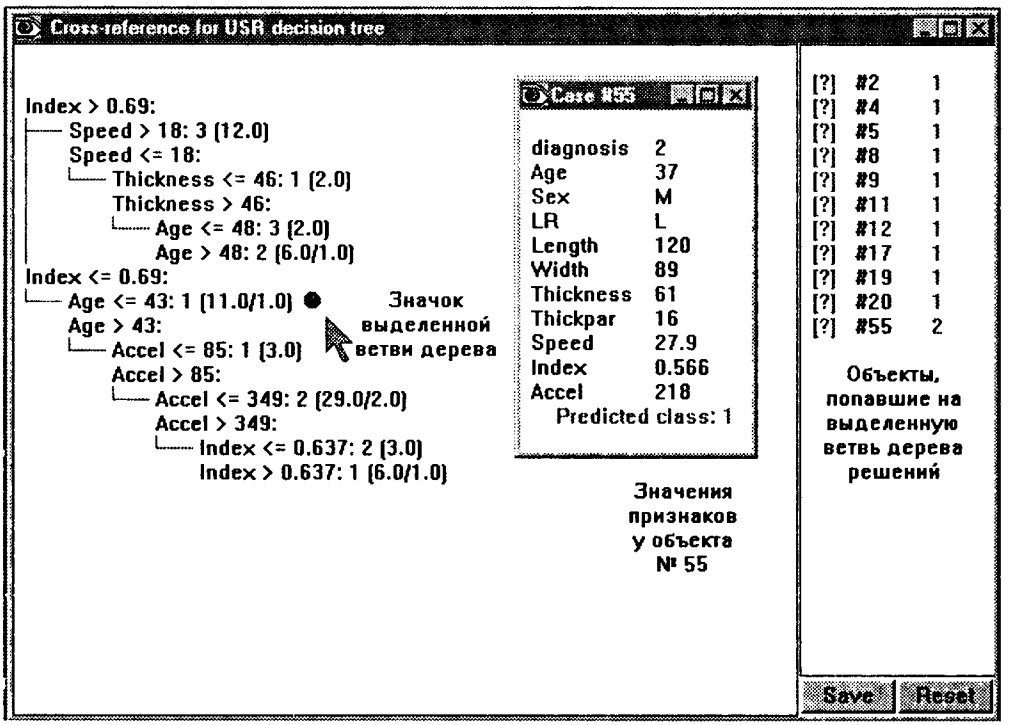


Рис. 5.11. Отображение результатов классификации в окне перекрестных ссылок

Преобразование дерева решений в набор правил

В ряде случаев полученное дерево решений может оказаться слишком сложным для восприятия. Например, при решении задач высокой размерности для неоднородных данных дерево нередко получается кустистое и довольно запутанное. Вместо того чтобы «ползать» по каждой полученной ветке, в системе See5 предусмотрена возможность преобразования дерева решений в набор правил IF ... THEN. Для этого требуется вызвать окно диалога для заданий параметров конструируемого алгоритма (см. рис. 5.9) и поставить флажок в поле **Rulesets** (набор правил). После проведения такой операции система добавляет в окно отчета список правил, соответствующих рассчитанному дереву решений. Применительно к рассматриваемым данным по ультразвуковой диагностике это будет следующий список:

```
Rule 1: (cover 11)
                Age <= 43
                Index <= 0.69
                ->  class 1  [0.846]
Rule 2: (cover 10)
                Speed > 19
                Index <= 0.69
                Accel > 310
                ->  class 1  [0.750]
Rule 3: (cover 14)
                LR = R
                Speed > 19
                Index <= 0.69
                ->  class 1  [0.625]
Rule 4: (cover 5)
                Age <= 63
                Speed <= 18
                Index > 0.69
                ->  class 1  [0.429]
Rule 5: (cover 15)
                Age > 43
                LR = L
                Index <= 0.69
                Accel <= 310
                ->  class 2  [0.941]
Rule 6: (cover 15)
                Age > 43
                Speed <= 19
                Index <= 0.69
                ->  class 2  [0.941]
Rule 7: (cover 8)
                Age > 63
                Speed <= 18
                ->  class 2  [0.800]
Rule 8: (cover 17)
                Age > 43
                Length <= 108
                Index <= 0.69
                ->  class 2  [0.789]
Rule 9: (cover 12)
                Speed > 18
                Index > 0.69
                ->  class 3  [0.929]
```

Каждое правило состоит из следующих фрагментов:

- номер правила;
- количество объектов обучающей выборки, подпадающих под действие правила (cover «n»);
- одно или несколько элементарных логических событий, входящих в состав правила (сложного логического высказывания);
- номер класса, которому соответствует данное правило;
- величина, принимающая значение от 0 до 1, которая выражает степень доверия к правилу (характеристика точности правила).

Для более детального рассмотрения множества правил, подобно тому как это делалось с деревом решений, можно обратиться к окну перекрестных ссылок (Cross-Reference).

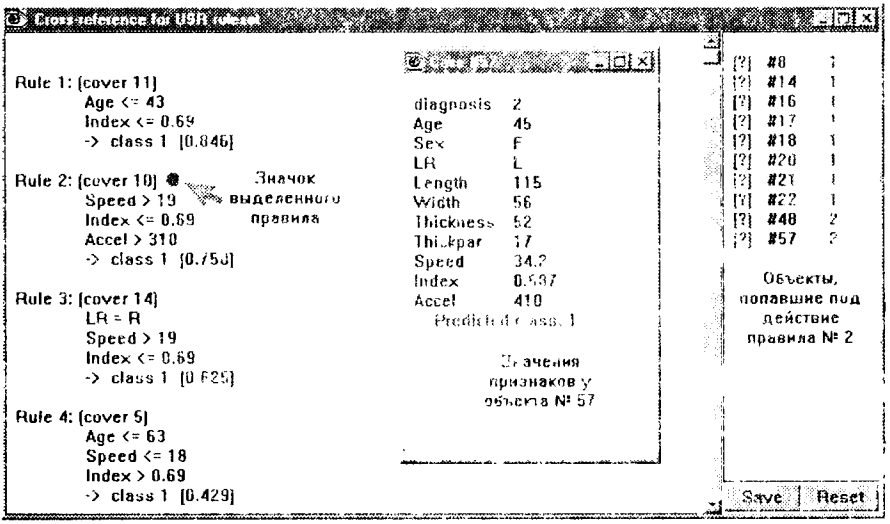


Рис. 5.12. Детальный разбор полученных правил в окне перекрестных ссылок

В целом, как уже говорилось, результаты в виде набора правил являются более простыми и понятными, чем в виде деревьев решений. Каждое правило ясным образом описывает связь между набором значений признаков и идентификатором класса. Более того, количество правил, сгенерированных из дерева решений, нередко оказывается несколько меньшим, чем число веток на дереве, а результат может оказаться более точным (в нашем случае этого, правда, не произошло). Вместе с тем, для больших баз данных генерация множества правил требует ощутимых временных затрат.

Усиление решения (Boosting)

Идея усиления решения заключается в конструировании не одного, как в рассмотренном выше случае, а сразу нескольких деревьев решений. При этом главное требование к таким деревьям решений заключается в том, чтобы они как

можно меньше дублировали друг друга. В системе See5 данная идея реализуется следующим образом.

На первом шаге конструируется начальное дерево решений (такое дерево применительно к данным по ультразвуковой диагностике почек было рассмотрено выше). Как следует из представленных результатов, классификатор, построенный на основе начального дерева, дает ошибки на некоторых объектах. Так, в нашем случае наблюдается 5 ошибок на 74 объектах обучающей выборки (см. рис. 5.10).

На втором шаге при конструировании следующего дерева делается попытка избежать ранее сделанных ошибок. Следствием такой попытки считается существенное отличие второго дерева от начального. Полученное дерево также будет приводить к ошибочным решениям, но уже на других объектах. На следующем шаге работы алгоритма очередное дерево строится с учетом ошибок всех предыдущих деревьев решений.

Для запуска процесса усиления решения требуется установить флажок Boost в диалоговом окне для задания параметров работы алгоритма (см. рис. 5.9). Кроме того, в этом же окне нужно задать общее число строящихся деревьев решений. Это число проставляется в поле trials.

Понятно, что построение множества деревьев решений требует дополнительного времени. Но временные издержки способны вполне окупиться — точность классификации, как правило, значительно повышается.

Разработчики See5 утверждают, что при использовании 10 деревьев решений ошибки классификации снижаются в среднем на 25 %. Посмотрим, как это будет выглядеть на наших числовых данных. Установим флажок Boost и в поле trials запишем цифру 3 (попробуем построить 3 дерева решений). Нажимаем OK и получаем окно отчета с информацией о результатах решения (рис. 5.13).

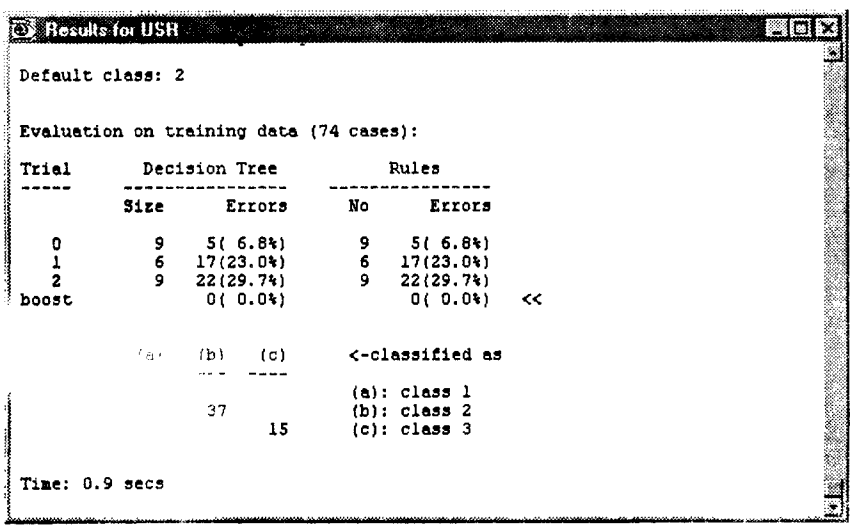


Рис. 5.13. Окно отчета о результатах построения трех деревьев решений

Как следует из отчета, второе дерево решений классифицирует данные с ошибкой 23 %, а для третьего дерева эта ошибка составляет 29,7 % (вообще говоря, нумерация деревьев начинается с цифры 0). Но все три дерева решений вместе классифицируют данные без ошибок (запись в строке «boost»). Для достижения такого безошибочного результата, как видно из отчета, потребовалось использование 9 + 6 + 9 = 24 правил.

Использование правил для принятия решений

Построенное множество правил применяется для принятия решения о принадлежности того или иного объекта какому-либо классу. При этом бывают ситуации, когда один и тот же объект подпадает под действие сразу нескольких правил, в том числе правил, описывающих разные классы. Подобные внутренние конфликты могут быть разрешены двумя способами. В первом способе предпочтение отдается одному правилу, имеющему более высокую степень доверия (более высокую точность). Второй способ связан с обобщением результатов разных правил для принятия окончательного решения.

В системе See5 принят второй способ — каждое сработавшее правило подает голос для отнесения какого-либо объекта к изучаемым классам. Голоса суммируются с весами, равными вычисленным степеням доверия, и объект считается принадлежащим к классу, для которого набирается наибольшая взвешенная сумма голосов.

Смягчение порогов

В системе See5 предусмотрена еще одна возможность улучшения качества классификации. Но на сей раз эта возможность касается не столько точности результатов, сколько повышения их устойчивости к возможным флуктуациям значений признаков. Она связана с введением нечетких (мягких) порогов, на сравнении с которыми основывается выбор той или иной ветви дерева решений.

В диалоговом окне для задания параметров алгоритма See5 (см. рис. 5.9) имеется специальный параметр для смягчения порогов. Это параметр fuzzy thresholds (размытые пороги). При обращении к нему вместо одного порога задается три значения — нижняя граница **LB**, верхняя граница **UB** и центральное значение **T**. Если значение переменной лежит ниже **LB** или выше **UB**, то исследуются соответствующие единственные ветви дерева. Если же значение переменной попадает между **LB** и **UB**, то исследуются одновременно две ветви дерева и выбирается наиболее правдоподобный результат классификации. Значения **LB**, **UB** и **T** система определяет автоматически.

Пример дерева решений с размытыми порогами приведен на рис. 5.14. Здесь каждый порог представлен в виде **<= LB (T)** или **>= UB (T)**.

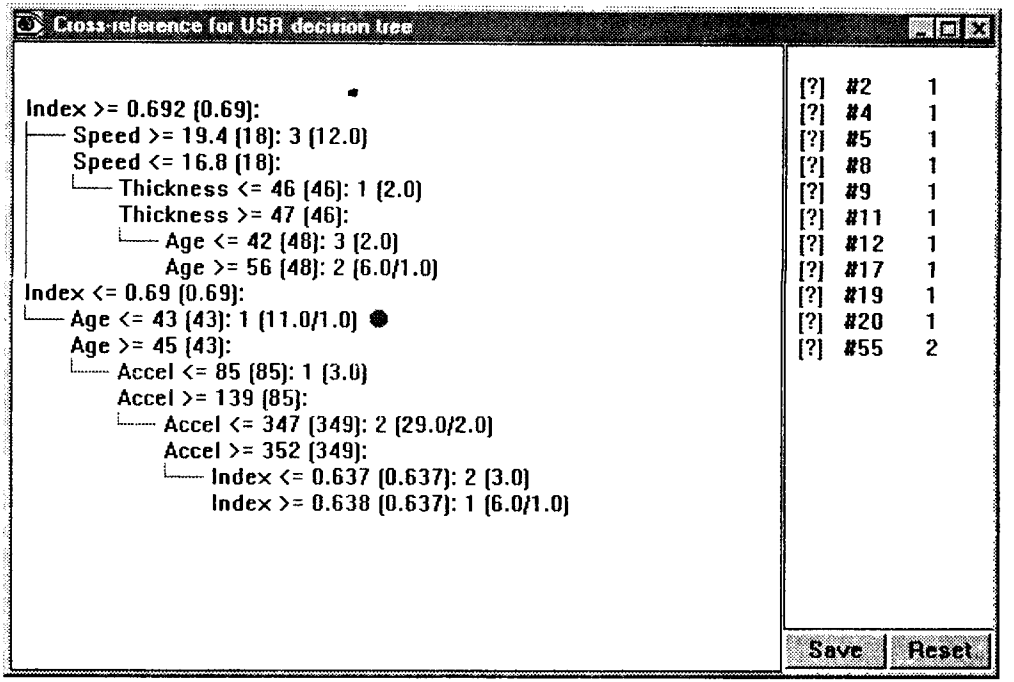


Рис. 5.14. Дерево решений с размытыми порогами

Дополнительные настройки алгоритма

В системе See5 предусмотрены параметры для дополнительной настройки алгоритма построения деревьев решений. Они предназначены для пользователя, желающего поэкспериментировать и, возможно, попытаться улучшить найденный результат.

Во-первых, сюда относится параметр **pruning CF** (ConFidence level — уровень доверия), предназначенный для отсечения статистически несостоятельных ветвей дерева решений. По умолчанию система выставляет значение уровня доверия 25 % (см. рис. 5.2). Изменение этого значения приводит к соответствующему изменению размера дерева решений.

Во-вторых, на точность классификации может существенно влиять параметр **Minimum...cases**. В поле этого параметра выставляется число, ограничивающее минимальное количество объектов на ветке дерева решений. Чем меньше будет это число, тем более «кустистым» станет дерево и тем точнее производится «подгонка» дерева под требуемую классификацию.

Перекрестная проверка

Для получения надежных оценок качества построенных классификаторов в системе See5 используется так называемая перекрестная проверка. Она осуществляется следующим образом.

Вся выборка объектов разбивается на *m* блоков примерно одного размера и с одинаковым распределением классов. Затем последовательно каждый блок используется как контрольный набор объектов для тестирования классификатора, построенного на основе внешних для данного блока объектов. Число блоков вводится в поле crossvalidate диалогового окна для задания опций алгоритма конструирования классификатора. Результат перекрестной проверки отображается в нижней части окна отчета.

Выборка из больших наборов данных

Несмотря на высокое быстродействие системы See5, конструирование классификаторов на полном наборе исходных данных при их большом количестве может занимать довольно много времени. Это становится особенно заметно при использовании дополнительных параметров алгоритма, например параметра для усиления решения (**boosting**).

See5 имеет возможность работы не с полным набором данных, а с некоторой выборкой из исходного набора. Для этого предусмотрен специальный параметр **Use sample of X %** (см. рис. 5.9). При использовании указанного параметра осуществляются две операции. Во-первых, из исходного набора случайным образом извлекается X % объектов и на их основе конструируется классификатор. И, во-вторых, производится тестирование построенного классификатора на другой непересекающейся выборке объема X % (если X < 50 %) либо на всех оставшихся объектах (если X > 50 %).

При очередном обращении к параметру **Use sample of X %** будет сделана новая случайная выборка из исходных данных, построен и протестирован новый классификатор. Но в системе See5 имеется также возможность зафиксировать выборку. Для этого необходимо поставить флажок в поле **Lock sample**.

На рис. 5.15 приведен результат построения дерева решений на выборке половинного объема от исходных данных. На обучающей выборке достигнут неплохой эффект классификации – ошибка составляет всего 5,4 %. Вместе с тем, на контрольной выборке, объем которой равен половине объема исходных данных, процент правильной классификации резко падает до 35,1 %. Это заставляет задуматься о том, насколько построенное дерево решений и соответствующие if-then-правила отражают объективную реальность, и, скорее всего, продолжить поиск более устойчивого варианта решения.

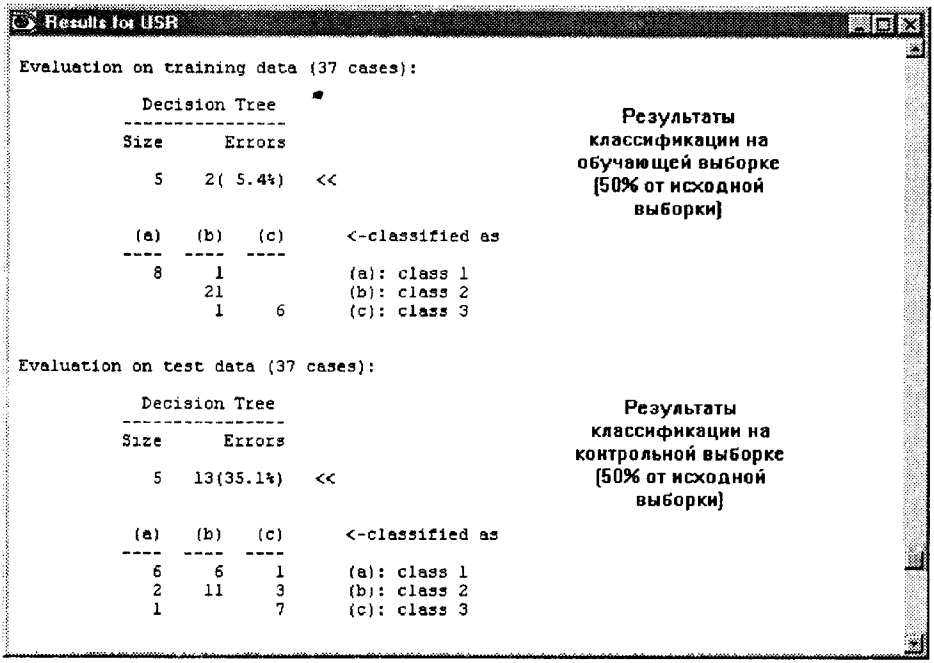


Рис. 5.15. Результаты классификации данных ультразвуковой диагностики на обучающей и контрольной выборках

Учет стоимости различных ошибок классификации

До сего момента, анализируя данные по ультразвуковой диагностике заболеваний, мы считали все виды ошибок классификации эквивалентными. Мы давали оценку качества построенного классификатора, просто подсчитывая общее число ошибок. Но в реальной жизни стоимость различных ошибок может быть разной. Например, если мы ошибочно сочтем здорового человека больным и направим его на дополнительное обследование, это будет не так страшно, как в случае ошибочного отнесения больного к группе здоровых. Соответственно, при оценке качества построенного дерева решений часто бывает необходимо вводить в анализ веса различных ошибок.

В системе See5 для учета стоимости различных ошибок классификации создается специальный файл *.costs. Он содержит строки следующего вида:

предсказанный класс, истинный класс: стоимость ошибки,

где «стоимость ошибки» — неотрицательное действительное число.

Число строк, характеризующих комбинации «предсказанный класс — истинный класс», в этом файле может быть любым. Если стоимость какой-либо ошибки не определена явно, то система назначает эту стоимость равной 1.

Предположим, что стоимость ошибочного отнесения больных почек к классу здоровых в нашем случае будет равна 10, а стоимость всех остальных видов

ошибок равна 5. Тогда файл для учета различной стоимости ошибок USR.costs может выглядеть следующим образом:

```
| costs file for USR
1, 2: 10    | стоимость ошибочного отнесения класса 2 к классу 1
1, 3: 10    | стоимость ошибочного отнесения класса 3 к классу 1
2, 1: 5     | стоимость ошибочного отнесения класса 1 к классу 2
2, 3: 5     | стоимость ошибочного отнесения класса 3 к классу 2
3, 1: 5     | стоимость ошибочного отнесения класса 1 к классу 3
3, 2: 5     | стоимость ошибочного отнесения класса 2 к классу 3
```

Результаты обработки данных с разделением на обучающую и контрольную выборки (по 50 %) и с учетом стоимости различных ошибок приведены на рис. 5.16.

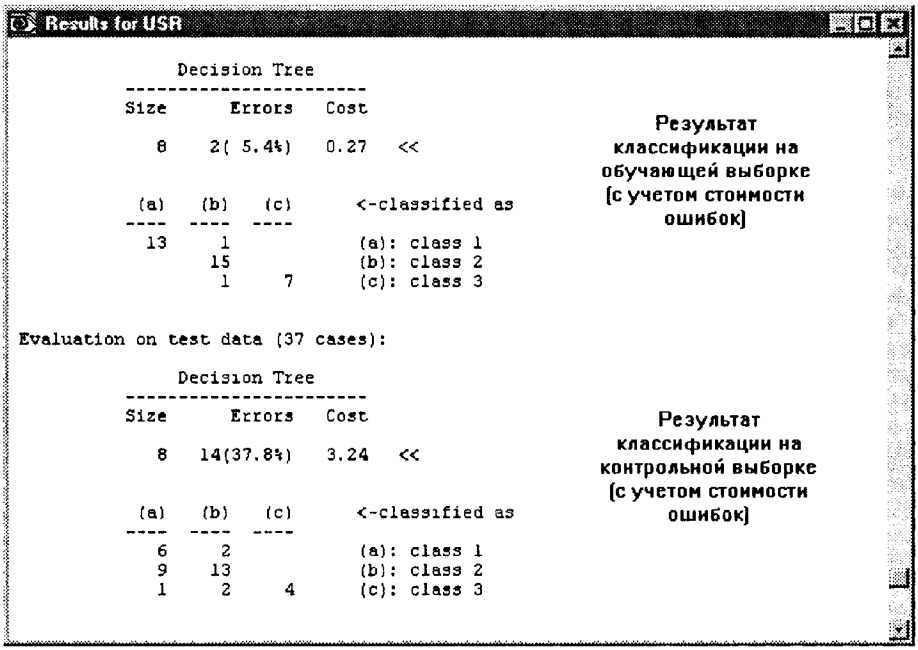


Рис. 5.16. Результаты классификации с учетом стоимости различных ошибок

Для редактирования файла стоимости различных ошибок классификации следует его вызвать из меню Edit ▸ costs file и внести необходимые изменения в автоматически инициализированном редакторе WordPad. Можно исключить учет стоимости ошибок, если поставить флажок Ignore costs file в окне диалога для задания параметров алгоритма построения деревьев решений (см. рис. 5.9).

Использование классификаторов

После того как пользователь признает какой-либо вариант дерева решений удовлетворительным, ему предоставляется возможность испытать этот вариант в интерактивном режиме на новых данных. Нажимаем в главном окне See5 кноп-

ку Use Classifier и тем самым активизируем алгоритм классификации данных, соответствующий самому последнему варианту дерева решений. На экран выдается специальное окно для ввода значений поочередно предъявляемых признаков (см. рис. 5.10). Количество таких признаков может быть разным, ведь в зависимости от ответов реализуется та или иная ветвь дерева решений. После ввода всех затребованных значений на экран выдается окно, в котором указываются предсказанный класс и уровень доверия к результату классификации (рис. 5.18).

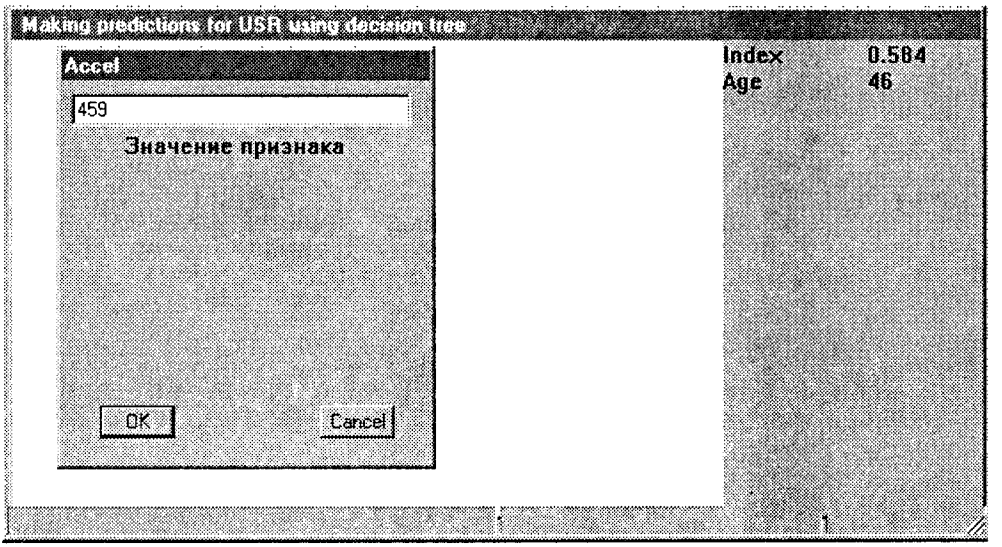


Рис. 5.17. Интерактивный режим классификации данных

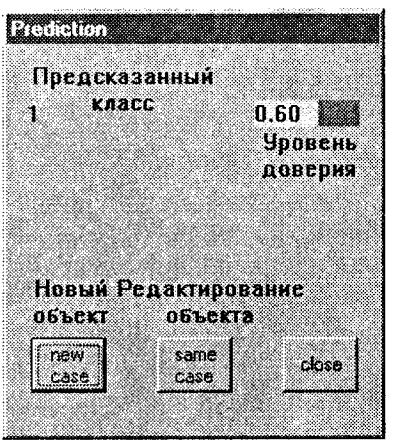


Рис. 5.18. Результат интерактивной классификации

Детальная проверка и сохранение результатов

Завершающая стадия работы See5 обычно заключается в детальном просмотре результатов работы построенного классификатора в окне перекрестных ссылок. После нажатия соответствующей кнопки (Cross-Reference) на экране появляется диалоговое окно, в котором предлагается выбрать файл с данными для классификации (рис. 5.19). Это может быть исходный файл данных (в нашем случае USR.data), файл с тестовыми данными (USR.test) или файл, содержащий объекты с неизвестной классификацией (USR.cases).

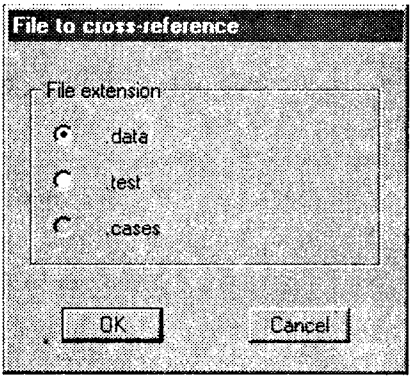


Рис. 5.19. Выбор файла данных для классификации

Выбрав требуемый файл, нажимаем OK. На экране появляется окно перекрестных ссылок, в левой половине которого сначала изображено полное дерево решений, а в правой представлен список объектов, подвергнутых классификации. Некоторые возможности работы с окном перекрестных ссылок обсуждались выше. Здесь остановимся еще на двух возможностях.

Первая заключается в возможности поэлементного просмотра для выбранного объекта ветви построенного дерева решения. Для этого нужно щелкнуть левой кнопкой мыши в правом поле окна перекрестных ссылок на требуемом объекте — в левом поле автоматически отобразится соответствующая ветка. Так, в случае, показанном на рис. 5.20, для изучения был выбран объект № 4 (около него появился темный кружок). Как видим, с этим объектом соотносится достаточно короткая ветка решения (**Index≤0,69&Age>43&Accel≤85**). Аналогичным образом можно разобрать результаты классификации всех других доступных объектов (нажатием кнопки Reset возвращается исходное изображение полного дерева решений).

Вторая возможность заключается в сохранении полученных результатов. Причем здесь существенным является *выборочное* сохранение. А именно, после нажатия кнопки Save, расположенной в правом нижнем углу окна перекрестных ссылок, сохраняться в текстовом формате будут только результаты, относящиеся к текущему отображению дерева решений (целиком или его части).

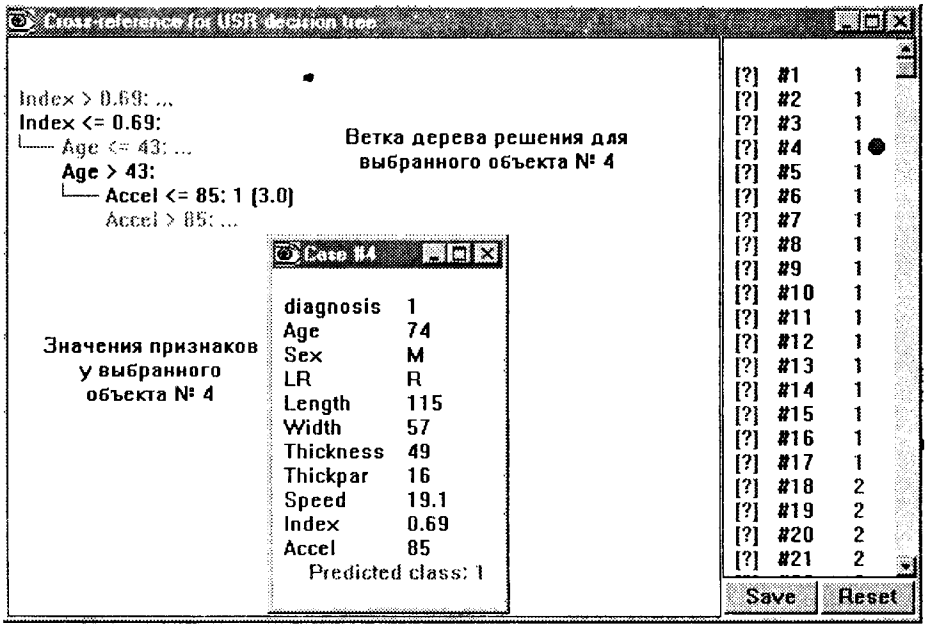


Рис. 5.20. Просмотр результатов классификации в окне перекрестных ссылок

WizWhy — система поиска логических правил в данных

Система WizWhy предприятия WizSoft (http://www.wizsoft.com) является современным представителем подхода, реализующего ограниченный перебор. Хотя авторы системы не раскрывают специфику алгоритма, положенного в основу работы WizWhy, вывод о наличии здесь ограниченного перебора был сделан по результатам тщательного тестирования системы (изучались результаты, зависимости времени их получения от числа анализируемых параметров и т. п.). Правда, по-видимому, в WizWhy ограниченный перебор используется в модифицированном варианте с применением дополнительного алгоритма «Apriori», исключающего из анализа логические события с низкой частотой.

Как уже отмечалось ранее, алгоритмы ограниченного перебора были предложены в середине 60-х годов М. М. Бонгардом [1] для поиска логических закономерностей в данных. С тех пор они продемонстрировали свою эффективность при решении множества задач из самых различных областей.

Эти алгоритмы вычисляют частоты комбинаций простых логических событий в подгруппах данных. Примеры простых логических событий: $X = a$; $X < a$; $X > a$; $a < X \le b$ и др., где X — какой-либо параметр, a и b — константы. Ограничением служит длина комбинации простых логических событий (у М. Бонгарда [1] она была равна 3). На основании анализа вычисленных частот делается заключение о полезности той или иной комбинации для установления ассоциации в данных, для классификации, прогнозирования и т. п.

Авторы системы WizWhy утверждают, что она автоматически извлекает из данных **ВСЕ** if-then-правила. На самом деле это, конечно, не так. Во-первых, максимальная длина комбинации в правиле if-then в системе WizWhy равна 6, и, во-вторых, с самого начала работы алгоритма производится эвристический поиск простых логических событий, на которых потом строится весь дальнейший анализ. Тем не менее, система WizWhy является на сегодняшний день одним из лидеров на рынке продуктов Data Mining. Это не лишено оснований. Система демонстрирует более высокие показатели при решении ряда практических задач, чем все остальные алгоритмы. Стоимость системы около $4000, количество пользователей ≈ 30 000. Демонстрационная версия WizWhy ограничена только количеством анализируемых записей — 1000 объектов.

Общие свойства системы WizWhy

Авторы WizWhy акцентируют внимание на следующих общих свойствах системы:

- выявление BCEX if-then-правил;
- вычисление вероятности ошибки для каждого правила;
- определение наилучшей сегментации числовых переменных;
- вычисление прогностической силы каждого признака;
- обобщение полученных правил и зависимостей;
- выявление необычных феноменов в данных;
- использование обнаруженных правил для прогнозирования;
- выражение прогноза в виде списка релевантных правил;
- вычисление ошибки прогноза;
- прогноз с учетом стоимости ошибок.

В качестве достоинств WizWhy дополнительно отмечают такие:

- на прогнозы системы не влияют субъективные причины;
- пользователям системы не требуется специальных знаний в прикладной статистике;
- более точные и быстрые вычисления, чем у других методов Data Mining.

Для большей убедительности авторы WizWhy противопоставляют свою систему нейросетевому подходу и алгоритмам построения деревьев решений и утверждают, что WizWhy, обладая более высокими характеристиками, вытесняет другие программные продукты с рынка Data Mining.

Загрузка и управление данными

Первое, что нужно сделать при работе с WizWhy, это загрузить анализируемый файл данных. Здесь имеется несколько возможностей:

- Вы можете подготавливать и читать файлы ASCII.
- Вы можете напрямую работать с файлами dBase (*.dbf), MS Access (*.mdb), Oracle и таблицами MS SQL.
- Вы можете воспринимать наборы данных посредством ODBC (Open Database Connectivity).
- Для начала работы с процедурой загрузки следует прежде всего обратиться к закладке Basic Data в окне диалога с именем текущего проекта (рис. 5.21). Здесь в поле Open Data of Type нужно указать тип загружаемых данных.

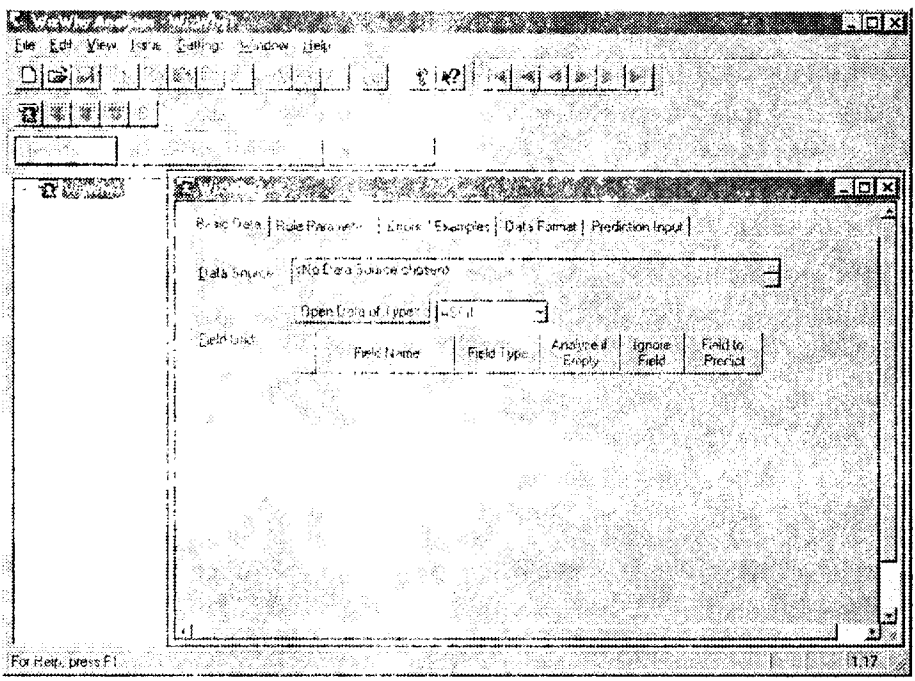


Рис. 5.21. Начало работы с системой WizWhy

- Для примера возьмем таблицу с данными по ультразвуковой диагностике почек в текстовом формате ASCII (разделителями колонок является знак табуляции, в первой строке таблицы данных записаны имена переменных). Укажем требуемый тип данных и в появившемся окне диалога выберем файл USR.txt — на экран выдается окно диалога системы WizWhy для редактирования и преобразования текстовых файлов (рис. 5.22).

В поле Record Type (тип записи) устанавливаем переключатель в положение Delimited (данные с разделителем) и ставим флажок в позиции First record for fields names, говорящий о том, что имена переменных располагаются в первой строке таблицы данных. В поле Field delimiters (разделитель) ставим флажок в позиции Tab (знак табуляции). Нажимаем кнопку Parse, после чего система производит

автоматический грамматический разбор наших данных. Просматриваем результаты этого разбора и при необходимости вносим коррективы — в поле Column (field) предоставляются возможности для изменения имен и типов переменных, а также для отказа от импорта каких-либо колонок. Нажимаем OK. Система импортирует данные для дальнейшей обработки, что отражается в диалоговом окне для управления данными Basic Data (рис. 5.23).

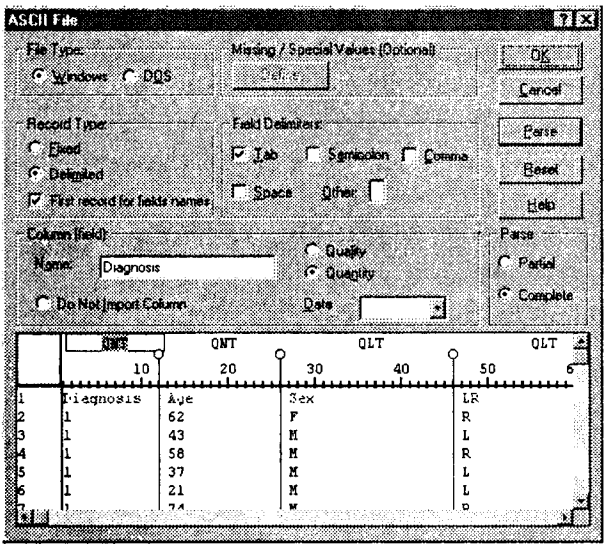


Рис. 5.22. Диалоговое окно для чтения данных в текстовом формате

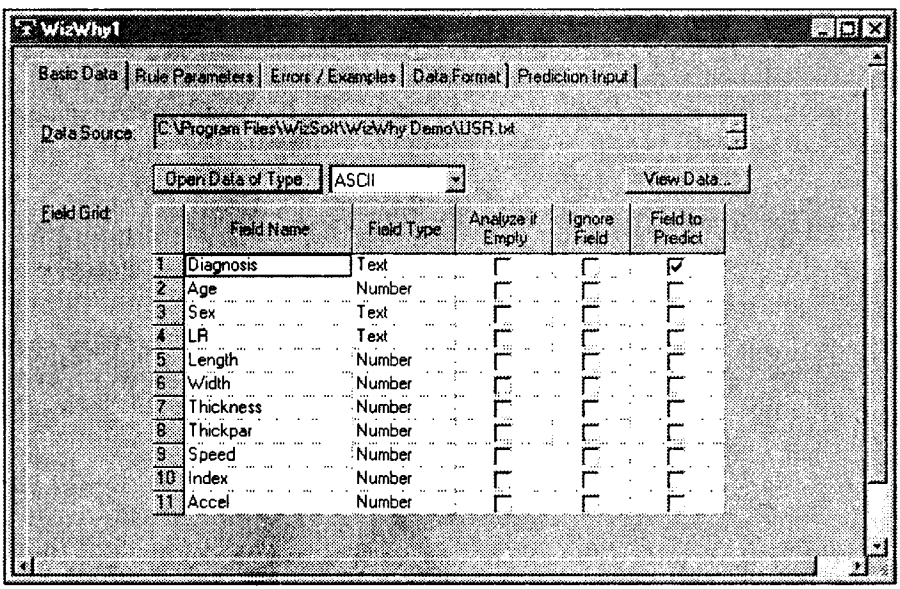


Рис. 5.23. Окно диалога для управления данными

В поле Data Source указываются местоположение и имя файла, из которого были импортированы данные. Кнопка View Data предназначена для вызова окна для просмотра загруженных данных (в нем демонстрируются 100 первых строк таблицы данных). В поле Field Grid отображаются имена и типы введенных переменных и предоставляются возможности проведения следующих операций:

- Назначение целевой, или так называемой зависимой (dependent) переменной. Это переменная, значения которой будут связываться с помощью if-then-правил со значениями так называемых независимых (independent) переменных. В нашем случае такой целевой переменной является **Diagnosis** — выставляем флажок в соответствующей позиции колонки Field to Predict.
- Модификация переменных. В колонке Field Name можно редактировать имена переменных. Для этого нужно щелкнуть на соответствующей позиции и ввести новое имя. Кроме того, в позициях колонки Field Type можно изменять тип переменной. Например, заменить тип Text (текстовый, номинальный) на Number (количественный) или Date (дата) в формате День-Месяц-Год (Год-Месяц-День) и т. п. Здесь заметим, что в зависимости от выбранного типа данных в дальнейшем к переменной применяются различные процедуры обработки.

В системе WizWhy предусмотрен также случай, когда пропуски в таблице данных (пустые ячейки) представляют собой самостоятельные информативные события. Для учета подобных пропусков в значениях какой-либо переменной ставится флажок против нее в колонке Analyze if Empty. В свою очередь, если имеется необходимость исключить переменную из анализа, нужно выставить флажок в колонке Ignore Field.

В нашем примере текстовый формат имеют три переменные — целевой признак **Diagnosis**, признак **Sex** (пол пациента) и признак **LR** (левая или правая почка). Остальные переменные **Age** (возраст), **Length** (длина почки), **Width** (ширина почки), **Thickness** (толщина почки), **Thickpar** (толщина паренхимы), **Speed** (средняя скорость кровотока), **Index** (индекс резистентности) и **Accel** (ускорение артериального потока в систолу) являются количественными.

Задание параметров процедуры поиска правил

В системе WizWhy целевой признак разделяет все множество объектов на две части. Это делается следующим образом.

Если целевая переменная является текстовой (номинальной), WizWhy просматривает все объекты (записи) и отбирает те из них, для которых целевая переменная имеет выбранное значение. Отобранные таким образом объекты составляют первую группу. Правила, характерные для данной группы, называются if-then-правилами. Оставшиеся объекты составляют вторую группу, и для этой группы характерные правила обозначаются как if-then-NOT-правила.

Если целевой признак является количественным, пользователь должен указать область значений этого признака. Правила if-then будут определяться для этой

указанной области. В свою очередь, if-then-NOT-правила будут описывать объекты, не попавшие в выделенную область.

В рассматриваемом нами практическом примере целевой признак **Diagnosis** номинальный. Он принимает три значения: 1 — в классе «здоровая почка»; 2 — в классе «множественные кисты» и 3 — в классе «гидронефроз». Будем искать в данных if-then-правила для объектов с диагнозом «множественные кисты». Для этого с помощью закладки Rule Parameters (параметры правил) войдем в соответствующее окно диалога и в поле Predicted Value выставим значение «2» (рис. 5.24).

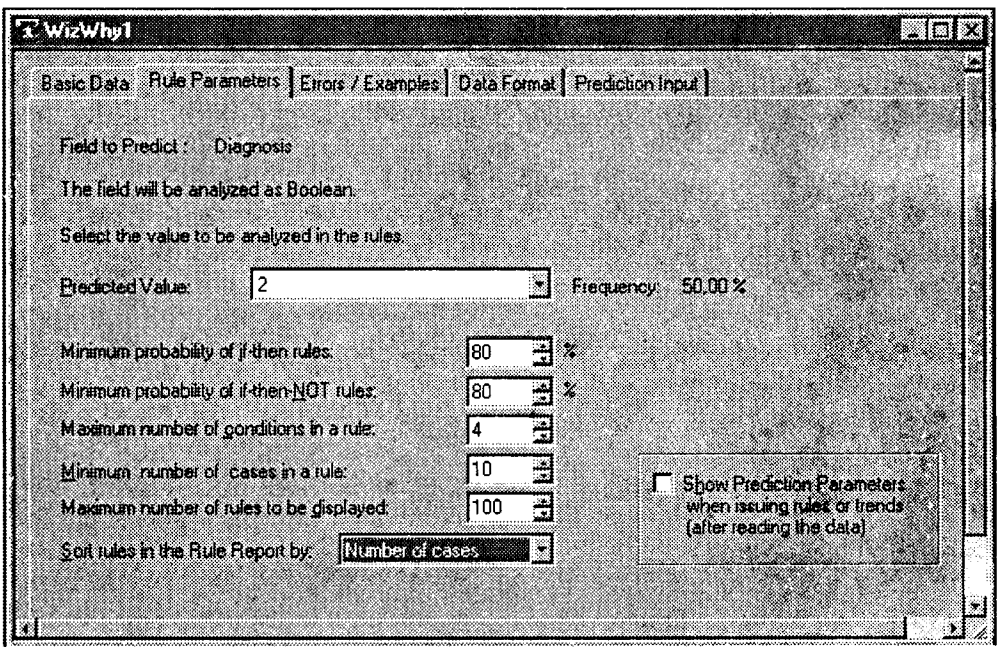


Рис. 5.24. Окно диалога для задания параметров процедуры поиска логических правил

После задания области значений целевой переменной или, как в нашем случае, ее одного значения система WizWhy читает данные и вычисляет простые статистики, которые могут быть использованы в дальнейшем анализе. Так, например, справа от поля Predicted Value система выводит значение частоты, с которой в анализируемых данных встречается значение Diagnosis=2. Как указывают авторы, чтение больших наборов данных способно занимать много времени. Пользователь может прекратить процесс чтения, нажав кнопку Cancel на специальной панели. В этом случае дальнейшему исследованию подвергается только та информация, которая успела прочитаться. Но при желании процесс чтения данных можно повторить.

Следующим шагом является задание собственно параметров правил, которые будут искаться в прочитанных данных. Сюда прежде всего относятся **Minimum probability of if-then rules** (минимальная вероятность if-then-правил) и **Minimum**

probability of if-then-NOT rules (минимальная вероятность if-then-NOT-правил). Эти параметры есть не что иное, как **точность** правила, охарактеризованная в предыдущей главе. Проставим в соответствующих полях окна диалога одинаковые значения указанных вероятностей 80 %. Это означает, что системе WizWhy формулируется требование обнаруживать правила, которые будут ошибаться не более чем в 20 % случаев (имеются в виду ошибки на анализируемой выборке).

В принципе, можно задавать любые значения минимальных вероятностей от 0 до 100 %. Но следует хорошо представлять, что, задав слишком низкий уровень точности, мы получим большое количество правил, среди которых будет много малоинформативных компонентов. В свою очередь, выставив требование 100 %, мы, скорее всего, не получим вообще ничего.

Еще одним важным параметром служит Maximum number of conditions in a rule (максимальное число условий в правиле). Это максимальное количество элементарных логических событий в одном правиле. Хотя авторы системы ничего не говорят о предельном значении данного параметра, установлено, что оно равно 6.

Следующим параметром, который необходимо задать для работы процедуры поиска правил, является Minimum number of cases in a rule (минимальное число объектов в правиле). Выставим здесь значение 10, обозначив тем самым наше желание обнаружить в данных правила, «покрывающие» не менее 10 объектов. Нижний предел составляет 4 объекта.

Последние операции в работе с рассматриваемым окном диалога касаются способов выдачи результатов. Во-первых, нужно ввести параметр Maximum number of rules to be displayed (максимальное количество отображаемых правил). Этот параметр не влияет на работу процедуры поиска правил. Он предназначен только для ограничения количества правил, выдаваемых в отчет (Rule Report). Далее следует указать способ сортировки правил в отчете (по уровню значимости Significance Level, по точности %Probability, по количеству объектов No. of Cases in a Rule).

Наконец, в правом нижнем углу окна диалога для задания параметров процедуры поиска правил можно поставить флажок, если имеется желание перед стартом процедуры дополнительно просматривать и корректировать ее параметры. Полностью подготовленное окно диалога для нашего примера по ультразвуковой диагностике почек приведено на рис. 5.24.

Работа с окном диалога Ошибки/Примеры (Errors/Examples)

Окно диалога Ошибки/Примеры показано на рис. 5.25. Оно разделено на два поля: Стоимость ошибок прогноза (Prediction Error Costs) и Представить примеры (Present examples where).

В поле Стоимость ошибок прогноза требуется ввести соответствующие значения по отдельности для двух видов ошибок: пропуска объекта (Cost of a miss) и ложной тревоги (Cost of a false alarm). По умолчанию эти значения равны 1. Но, как обсуждалось в предыдущем разделе, учет различной стоимости указанных ошибок может оказаться весьма ценным при решении практических задач.

В поле Представить примеры можно выразить желание просмотреть примеры работы выявляемых правил. Если поставить флажок в позиции Rule is in effect, то система будет формировать в отчете для каждого правила список номеров объектов, для которых правило не ошибается. Длина списка ограничивается заданным числом. Соответственно, флажок в позиции Rule is not in effect запрашивает у системы выдачу списка номеров объектов, на которых какое-либо правило работает с ошибкой.

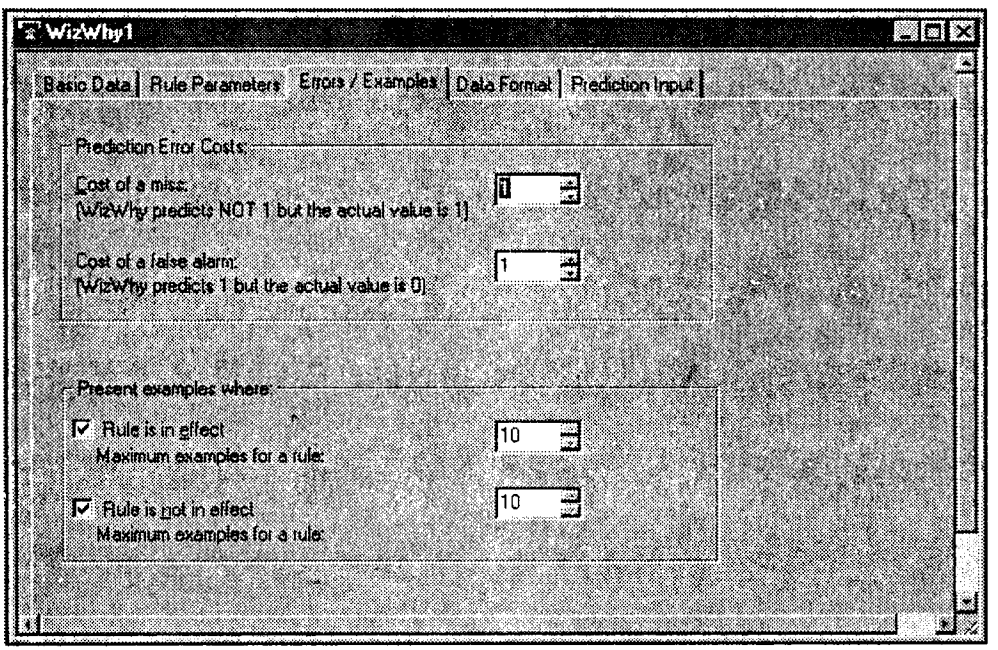


Рис. 5.25. Окно диалога Ошибки/Примеры

Работа с другими окнами диалога

Окно диалога Data Format предназначено для задания и корректировки формата информации, с которой работает WizWhy (рис. 5.26). Прежде всего сюда относится формат данных. В поле Number and Currency Format имеется возможность задавать количество цифр и виды разделителей в числовых и денежных данных, а в поле Data Format выбирать формат для записи дат.

Кроме того, в нижней части окна диалога предусмотрены параметры, выбор которых определяет место выдачи отчета о результатах работы системы (на принтер, на экран, в текстовый файл, в RTF-файл). В поле Subheading заносится подзаголовок отчета. Нажатием кнопки Font в правом нижнем углу вызывается окно диалога для выбора используемых шрифтов.

Последнее окно диалога — Prediction Input — предназначено для ввода, просмотра и коррекции внешних данных, на которых требуется проверить действие най-

денных правил. Оно изображено на рис. 5.27. Работа с этим окном аналогична работе с уже рассмотренным окном диалога Basic Data.

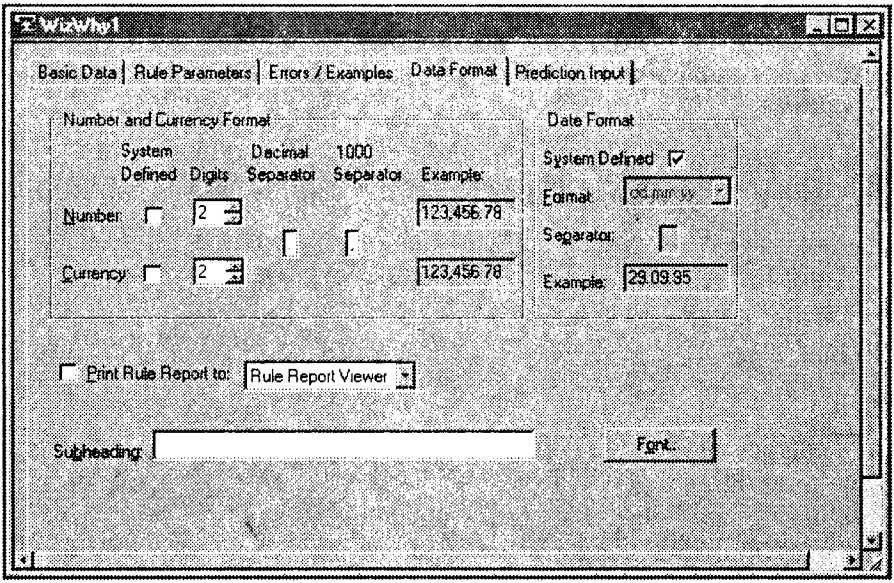


Рис. 5.26. Окно диалога для изменения формата информации

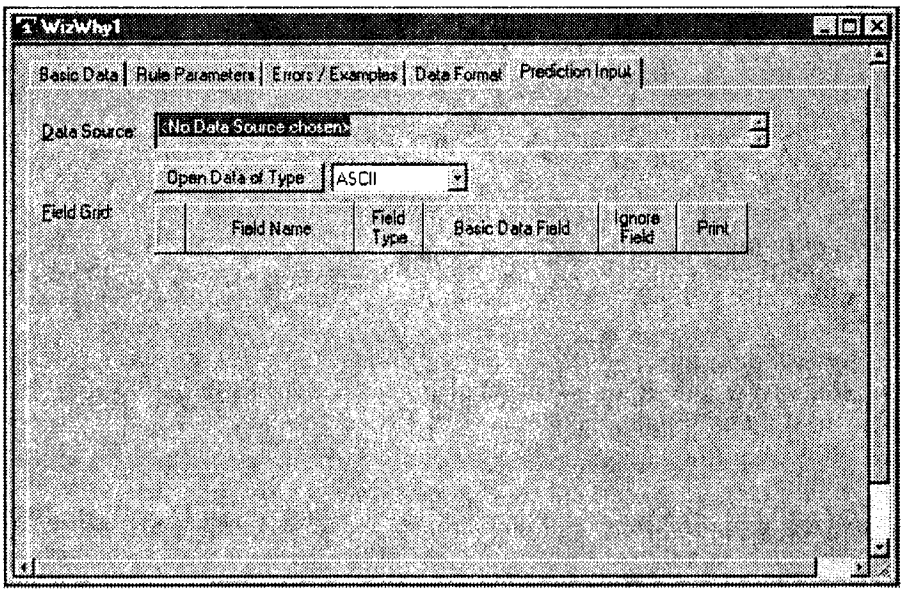


Рис. 5.27. Окно диалога для ввода внешних данных

Результаты работы системы

После внесения необходимой информации в рассмотренные выше окна диалога можно приступить к поиску правил в загруженных данных. Для этого нужно нажать кнопку Issue Rules (выдача правил) — система WizWhy выдает три отчета:

1. Отчет о правилах (Rule Report), в котором перечисляются обнаруженные правила с указанием их характеристик.
2. Отчет о трендах (Trend Report), в котором представляются результаты сегментации отдельных признаков.
3. Отчет о неожиданных правилах (Unexpected Rule Report).

Рассмотрим указанные отчеты более подробно.

Отчет о правилах

Отчет о правилах размещен в трех окнах (рис. 5.28):

1. Левое окно — список правил (Rule List);
2. Правое верхнее окно — содержание записи в деталях (Record Details Grid);
3. Правое нижнее окно — индекс признака (Field Index).

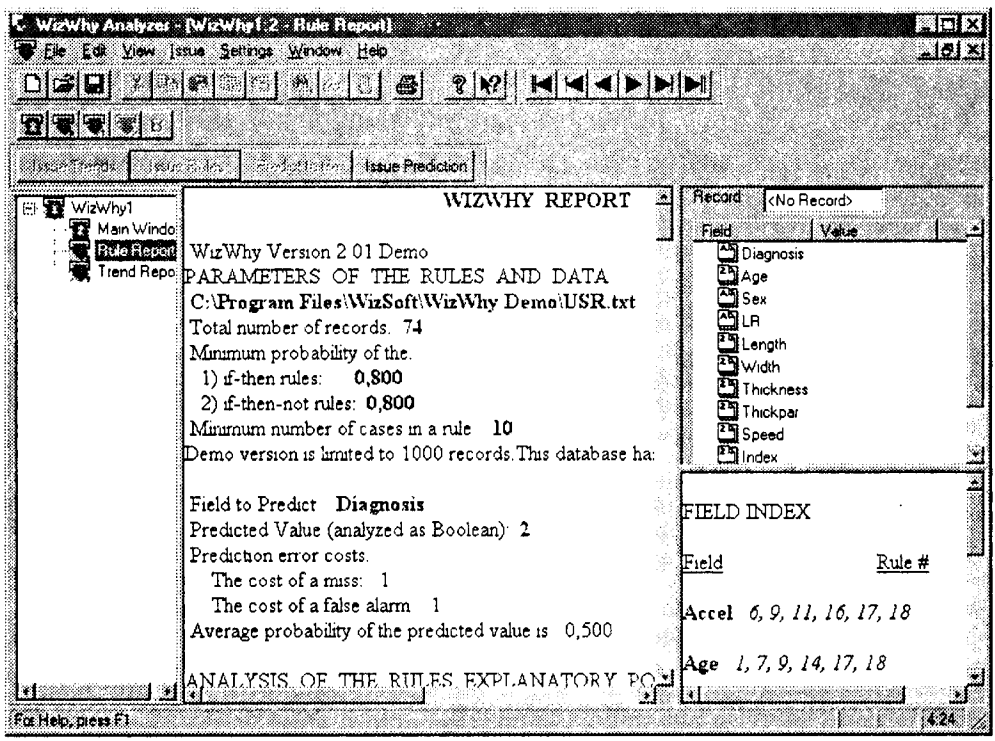


Рис. 5.28. Общий вид отчета о правилах в системе WizWhy

Список правил

Список правил предваряется информацией о заданных параметрах поиска. Здесь на примере данных по ультразвуковой диагностике почек, как видим, говорится, что общее число обработанных записей (объектов) составляет 74, минимальные вероятности правил if-then и if-then-NOT равны по 0,8, минимальное количество объектов для правил — 10. Затем подтверждается, что правила находятся для переменной **Diagnosis**, конкретно для значения этой переменной, равного 2. Также указывается, что стоимости ошибок в виде пропусков и ложных тревог составляют 1, а средняя вероятность (априорная вероятность) прогнозируемого значения переменной равна 0,5.

Далее система выдает следующий блок общей информации об обнаруженных правилах:

```
ANALYSIS OF THE RULES EXPLANATORY POWER
 Decision point: Predict 2 when conclusive probability is more than      0,460
 Number of misses:       2
 Number of false alarms: 5
 Total number of errors: 7
 Total cost of errors:   7
 Success rate when predicting 2:       0,868
 Success rate when predicting NOT 2:   0,931
 Number of records with no relevant rules:    7
 Average cost (per record):     0,104
 Expected average cost (per record):   0,500
 Improvement Factor:     4,786
```

Из приведенного блока можно почерпнуть сведения о значениях некоторых служебных параметров — `Decision point` (точка решения), `Average cost` (средние потери на запись), `Expected average cost` (ожидаемые средние потери) и `Improvement Factor` (выигрыш), представляющий собой отношение ожидаемых средних потерь к реальным потерям на запись. Кроме того, в блоке содержатся сведения о прогнозирующей способности всей совокупности обнаруженных правил — количество пропусков при прогнозировании (`Number of misses`), число ложных тревог (`Number of false alarms`), общее количество ошибок (`Total number of errors`), общие потери (`Total cost of errors`), вероятность успешного прогнозирования для класса 2 (`Success rate when predicting 2`), вероятность успешного прогнозирования альтернативного класса (`Success rate when predicting NOT 2`) и количество объектов, не охваченных выделенными правилами (`Number of records with no relevant rules`).

Список правил состоит из правил, упорядоченных по заданному критерию (в нашем случае по числу объектов, описываемых правилом). В данных по ультразвуковой диагностике почек при установленных параметрах процедуры система WizWhy обнаружила 19 правил. Они приведены в следующем листинге:

```
1.  If Age is 21,00...43,00 (average=34,89)
Then
Diagnosis is not 2
Rule's probability: 0,947
The rule exists in 18 records.
Significance Level: Error probability <0.1
Positive Examples (records' serial numbers):
```

2, 4, 5, 8, 9, 11, 12, 17, 19, 20
Negative Examples (records' serial numbers):
55
2. *If* **Index** is *0,70...0,80* (average=*0,73*)
Then
Diagnosis is not *2*
Rule's probability: 0,833
The rule exists in 15 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 3, 60, 61, 62, 63, 64, 65, 66, 67
Negative Examples (records' serial numbers):
28, 32, 33
3. If **Width** is *39,00...53,00* (average=*48,00*)
Then
Diagnosis is not *2*
Rule's probability: 0,813
The rule exists in 13 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 4, 9, 10, 11, 17, 19, 20, 62
Negative Examples (records' serial numbers):
48, 49, 58
4.**Speed** is *16,30...41,50* (average=*23,97*)
and **Index** is *0,70...0,80* (average=*0,73*)
Then
Diagnosis is not *2*
Rule's probability: 0,929
The rule exists in 13 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
3, 61, 62, 63, 64, 65, 66, 67, 68, 70
Negative Examples (records' serial numbers):
33
5. If **Index** is *0,65...0,67* (average=*0,66*)
Then
Diagnosis is *2*
Rule's probability: 0,800
The rule exists in 12 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 26, 29, 31, 34, 37, 38, 47, 49, 50
Negative Examples (records' serial numbers):
7, 11, 17
6. If **Accel** is *210,00...242,00* (average=*224,47*)
Then
Diagnosis is *2*
Rule's probability: 0,800
The rule exists in 12 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 28, 36, 41, 42, 44, 50, 53, 55, 56
Negative Examples (records' serial numbers):
60, 67, 69

7. If **Age** is *48,00...70,00* (average=*62,77*)
and **Index** is *0,65...0,67* (average=*0,66*)
Then
Diagnosis is *2*
Rule's probability: **0,923**
The rule exists in **12** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 26, 29, 31, 34, 37, 38, 47, 49, 50
Negative Examples (records' serial numbers):
7
8. If **Sex** is *F*
and **LR** is *L*
and **Width** is *56,00...77,00* (average=*63,00*)
Then
Diagnosis is *2*
Rule's probability: **0,857**
The rule exists in **12** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
33, 40, 41, 42, 45, 46, 51, 52, 54, 56
Negative Examples (records' serial numbers):
14, 18
9. If **Age** is *47,00...70,00* (average=*61,92*)
and **Accel** is *210,00...242,00* (average=*226,08*)
Then
Diagnosis is *2*
Rule's probability: **0,917**
The rule exists in **11** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 28, 36, 41, 42, 44, 50, 53, 56, 58
Negative Examples (records' serial numbers):
69
10. If **Width** is *56,00...72,00* (average=*62,42*)
and **Index** is *0,65...0,67* (average=*0,66*)
Then
Diagnosis is *2*
Rule's probability: **0,917**
The rule exists in **11** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 26, 29, 31, 34, 37, 38, 47, 50, 54
Negative Examples (records' serial numbers):
7
11. If **Width** is *56,00...89,00* (average=*65,23*)
and **Accel** is *210,00...242,00* (average=*224,92*)
Then
Diagnosis is *2*
Rule's probability: **0,846**
The rule exists in **11** records.

Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
25, 28, 36, 41, 42, 44, 50, 53, 55, 56
Negative Examples (records' serial numbers):
60, 67
12. If **Sex** is *M*
and **LR** is *R*
and **Speed** is *16,30...39,30* (average=*21,47*)
Then
Diagnosis is not *2*
Rule's probability: **0,846**
The rule exists in **11** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
3, 6, 7, 8, 11, 17, 61, 65, 66, 67
Negative Examples (records' serial numbers):
43, 44
13. If **Speed** is *2,30...15,40* (average=*13,28*)
Then
Diagnosis *is 2*
Rule's probability: **0,833**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
23, 24, 25, 26, 27, 28, 29, 30, 31, 32
Negative Examples (records' serial numbers):
1, 60
14. If **Age** is *46,00...70,00* (average=*62,00*)
and **Speed** *is 2,30...15,40* (average=*13,17*)
Then
Diagnosis is *2*
Rule's probability: **0,909**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
23, 24, 25, 26, 27, 28, 29, 30, 31, 32
Negative Examples (records' serial numbers):
1
15. If **Width** is *54,00...87,00* (average=*64,82*)
and **Speed** is *2,30...15,40* (average=*13,28*)
Then
Diagnosis is *2*
Rule's probability: **0,909**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
23, 24, 25, 26, 27, 28, 29, 30, 31, 32
Negative Examples (records' serial numbers):
60
16. If **Speed** is *17,70...43,30* (average=*25,84*)
and **Accel** is *210,00...242,00* (average=*223,58*)
Then

```
Diagnosis is 2
Rule's probability: 0,833
The rule exists in 10 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
36, 41, 42, 44, 50, 53, 55, 56, 58, 59
Negative Examples (records' serial numbers):
67, 69
17. If Age is 23,00...43,00 (average=36,30)
and Accel is 255,00...629,00 (average=377,70)
Then
Diagnosis is not 2
Rule's probability: 1,000
The rule exists in 10 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
2, 8, 12, 17, 19, 20, 65, 70, 71, 72
18. If Age is 47,00...70,00 (average=60,50)
and Width is 56,00...77,00 (average=62,60)
and Accel is 210,00...242,00 (average=227,00)
Then
Diagnosis is 2
Rule's probability: 1,000
The rule exists in 10 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
25, 28, 36, 41, 42, 44, 50, 53, 56, 59
19. If LR is R
and Speed is 16,30...41,50 (average=25,44)
and Index is 0,70...0,80 (average=0,72)
Then
Diagnosis is not 2
Rule's probability: 1,000
The rule exists in 10 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
3, 61, 63, 64, 65, 66, 67, 72, 73, 74
```

Для пояснения полученных результатов рассмотрим более подробно, например, правило № 19:

```
19. If LR is R
and Speed is 16,30...41,50 (average=25,44)
and Index is 0,70...0,80 (average=0,72)
Then
Diagnosis is not 2
Rule's probability: 1,000
The rule exists in 10 records.
Significance Level:  Error probability <0,1
Positive Examples (records' serial numbers):
3, 61, 63, 64, 65, 66, 67, 72, 73, 74
```

Это правило представляет собой конъюнкцию трех элементарных высказываний. Первое — LR is *R* — говорит о том, что правило относится только к правой поч-

ке. Второе — Speed is *16.30...41.50* — определяет диапазон значений для средней скорости кровотока, и третье — Index is *0.70...0.80* — описывает интервал значений индекса резистентности. Высказывание Diagnosis is not *2* означает, что правило характерно для объектов, не имеющих диагноз «множественные кисты».

Запись Rule's probability: 1.000 означает, что точность правила в данном случае равна 1. Следующая запись — The rule exists in 10 records — характеризует объем множества объектов, для которых справедливо рассматриваемое правило, а другая запись — Significance Level: Error probability <0.1 — касается статистической оценки уровня значимости полученного правила (как видим, доверие к правилу превышает 90 %). Последняя запись — Positive Examples (records' serial numbers) — означает «положительные примеры», которые затем представлены как номера записей (объектов) в наборе данных.

Система WizWhy предоставляет возможность визуализации полученного правила. Для этого нужно щелкнуть на правиле левой кнопкой мыши и затем с помощью правой кнопки вызвать контекстное меню, в котором выбрать диаграмму правила Rule Chart (рис. 5.29). Эта диаграмма иллюстрирует отдельные компоненты правила и дает графическое отображение совокупного взаимодействия переменных.

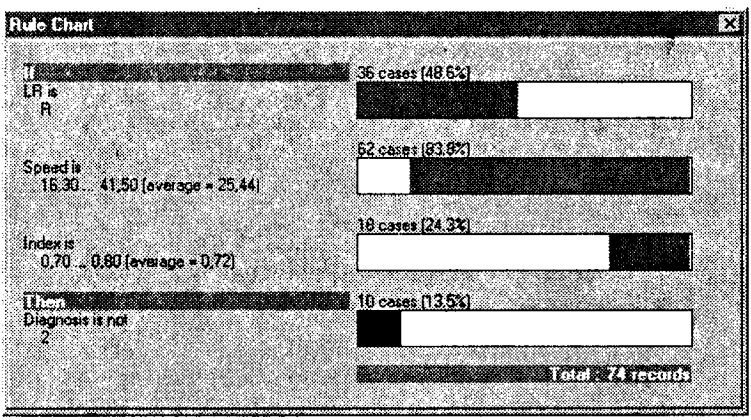


Рис. 5.29. Диаграмма выделенного правила № 19

Содержание записи в деталях

Окно «Содержание записи в деталях» позволяет просмотреть значения признаков для каждого объекта. Для этого требуется ввести номер объекта в поле Record и нажать клавишу Enter. Пример для объекта № 25 приведен на рис. 5.30.

Другая возможность состоит в том, что если дважды щелкнуть левой кнопкой мыши на номере объекта в списке правил, который там приведен в качестве положительного или отрицательного примера, соответствующие значения признаков отобразятся в рассматриваемом окне. При этом целевая переменная будет отмечена специальным значком красного цвета, а все остальные — значками зеленого цвета. Кроме того, на значках, расположенных сразу слева от названия признаков, указываются типы данных признаков.

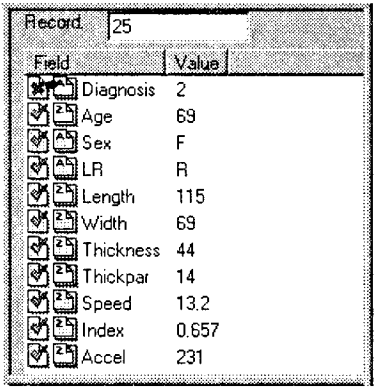


Рис. 5.30. Содержание записи в деталях

Индекс признака

В окне «Индекс признака», расположенном в правом нижнем углу, отображаются порядковые номера правил, в которых появляются те или иные признаки (рис. 5.31). Можно просмотреть все окно, используя прокрутку. Также в системе предусмотрена другая возможность — если в списке правил дважды щелкнуть мышью на каком-либо признаке в любом из правил, то этот признак будет автоматически выделен в окне «Индекс признака». По представляемой информации удобно выносить суждения о полезности признаков (о коэффициенте использования признаков) для классификации данных и прогнозирования. В свою очередь, если дважды щелкнуть в окне «Индекс признака» по любому номеру правила, то это правило моментально будет выделено в списке правил.

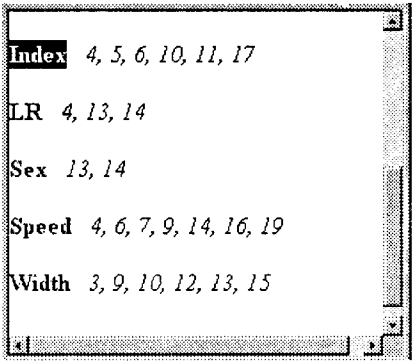


Рис. 5.31. Индекс признака

Распечатка и экспорт правил

Для распечатки правил или их экспорта в другой файл требуется нажать соответствующую кнопку печати в главном окне WizWhy — на экране появится специальное окно диалога Print Rules (рис. 5.32).

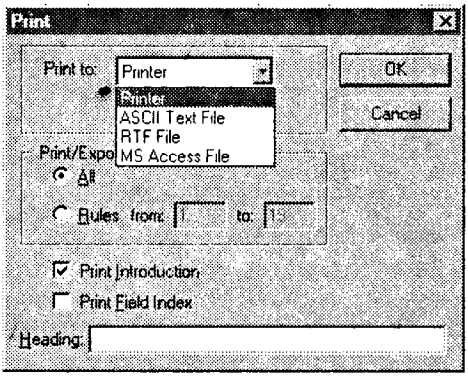


Рис. 5.32. Окно диалога для распечатки и экспорта выделенных правил

В поле Print to указывается адрес, по которому направляется результирующая информация. Это может быть принтер, ASCII- или RTF-файлы, а также файл VS Access.

В поле Print/Export Range указывается диапазон порядковых номеров правил, которые должны быть распечатаны или экспортированы. В нижней части окна диалога проставляются по необходимости флажки для распечатки или экспорта введения к списку правил Print Introduction и содержимого окна «Индекс признака». Кроме того, в поле Heading можно ввести заголовок для результирующей информации.

В системе WizWhy предусмотрена также возможность экспорта и сохранения полученных правил в виде операторов SQL. Для этого необходимо войти в меню Issue ▸ SQL Statement... — система выдает окно диалога, показанное на рис. 5.33. В этом окне можно редактировать совокупность операторов и с помощью переключателей адресовать данную совокупность в текстовый файл либо в буфер обмена.

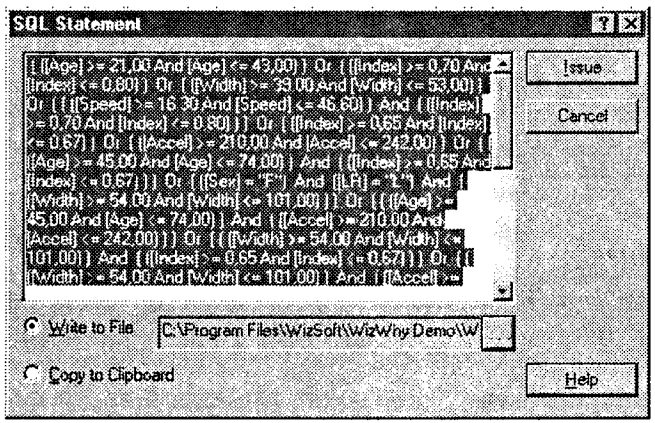


Рис. 5.33. Результаты работы WizWhy в виде операторов SQL

Отчет о трендах

Отчет о трендах представляет результаты сегментации отдельных признаков. Окно данного отчета разделено на три области (рис. 5.34).

В области, расположенной в левом верхнем углу, мы задаем анализируемый признак (Field to be analyzed). Здесь можно не только выбирать требуемый признак, но и сортировать признаки по какому-либо критерию (в алфавитном порядке, по номеру поля, по информативности).

Другие две области предназначены для отражения отношений между значениями признака и зависимой переменной. В верхней правой области окна отчета приводятся статистические характеристики сегментов выделенного признака. И, наконец, в нижней области отчета приводится графическая иллюстрация информативности каждого сегмента.

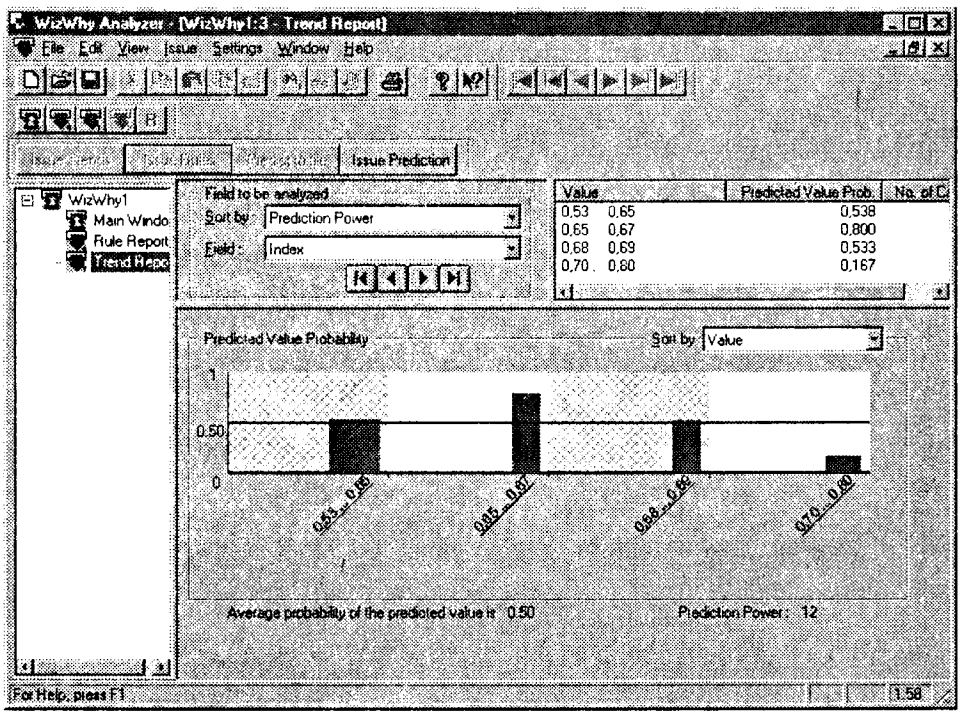


Рис. 5.34. Отчет о трендах

На графике по горизонтальной оси располагаются сегменты, на которые выбранный признак автоматически разбивается системой WizWhy. По вертикальной оси откладывается отношение количества объектов класса if-then к общему количеству объектов, попадающих в сегмент. Таким образом, высота столбиков на графике отражает информативность сегментов. Если столбик выше горизонтальной черты, значит, в данный сегмент чаще попадают объекты класса if-then, а если

ниже горизонтальной черты — класса if-then-NOT. В свою очередь, ширина столбиков пропорциональна количеству объектов, относящихся к данному сегменту.

Отчет о неожиданных правилах

В системе WizWhy введено представление о так называемых неожиданных правилах (unexpected rules). Под неожиданными понимаются правила в виде конъюнкции двух и более простых высказываний, комбинация которых дает точность и полноту прогноза выше, чем это можно было бы ожидать при независимости простых высказываний. Это представление, по-видимому, имеет цель дополнительно заинтриговать конечного пользователя возможностью открывать в данных нетривиальные закономерности.

В нашем случае система не обнаружила таких неожиданных правил. Однако можно попытаться это сделать, если мы изменим задание на поиск правил. Например, уменьшим минимальную вероятность if-then- и if-then-NOT-правил с 80 до 70 % в окне Rule Parameters (см. рис. 5.25). Проделаем указанную операцию и нажмем кнопку Issue Rules — теперь система обнаружит в данных по ультразвуковой диагностике 38 правил, и среди них окажется четыре неожиданных, отчет о которых выдается в специальном окне (рис. 5.35).

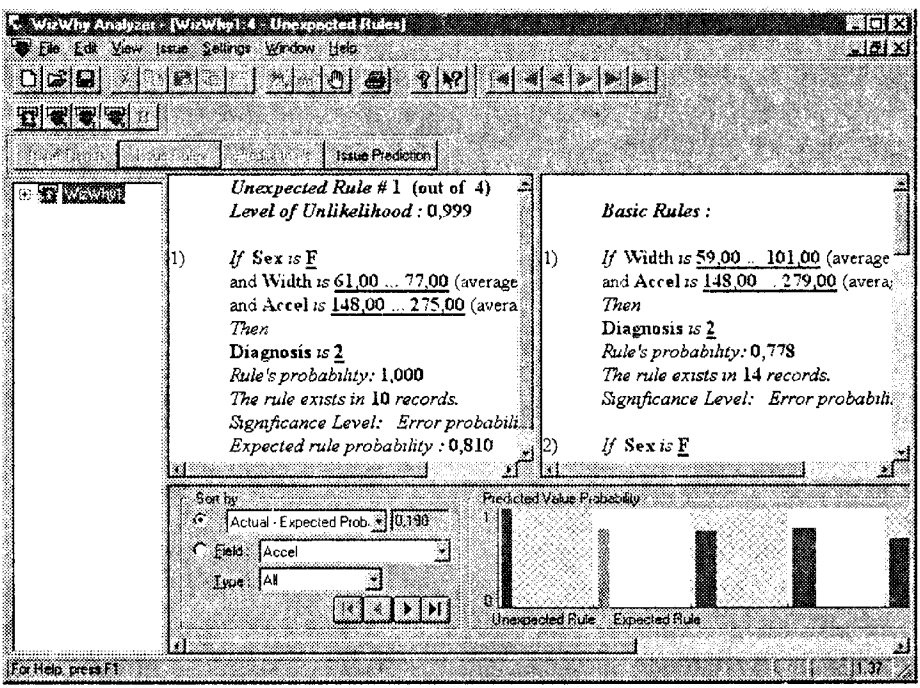


Рис. 5.35. Отчет о неожиданных правилах

Окно отчета о неожиданных правилах разделено на три секции. В левой верхней секции отображается в стандартной форме найденное неожиданное правило.

Правая верхняя секция содержит информацию об элементах, из которых составлено неожиданное правило. И наконец, нижняя секция предназначена для сортировки неожиданных правил и графического представления результатов.

Так, в нашем случае первое неожиданное правило, изображенное на рис. 5.28, расшифровывается следующим образом: если (пол женский) и (ширина почки в интервале от 61 до 77) и (ускорение кровотока от 148 до 275), то диагноз «множественные кисты». Данное правило вместе с рассчитанными характеристиками приведено ниже. Здесь по сравнению с ранее рассмотренными характеристиками выдаются две новые — уровень неожиданности (Level of Unlikelihood) и ожидавшаяся вероятность правила (Expected rule probability). Как видим, за счет взаимосвязи элементов правила точность целого правила составила 1 и оказалась значительно выше ожидавшейся (0,81).

```
Unexpected Rule # 1 (out of 4)
Level of Unlikelihood: 0.999
If Sex is F
and Width is 61,00...77,00 (average=67,30)
and Accel is 148,00...275,00 (average=216,10)
Then
Diagnosis is 2
Rule's probability: 1.000
The rule exists in 10 records.
Significance Level: Error probability <0.1
Expected rule probability: 0,810
Actual minus Expected probability: 0,190
```

В правой верхней секции приводится статистический разбор компонентов, из которых состоит неожиданное правило. Он состоит из двух частей (табл. 5.6).

Базисные правила (Basic Rules) представляют собой комбинации простых событий, входящих в неожиданное правило. В нашем случае, так как неожиданное правило состоит из трех простых событий, число таких комбинаций также составит 3.

Базисные тренды (Basic Trends) — это статистический разбор сегментов анализируемых переменных, составляющих собственно простые логические события.

Таблица 5.6. Разбор компонентов неожиданного правила

Basic Rules	Basic Trends
1. If Width is 59,00...101,00 (average=69,00)	4. If Accel is 148,00...279,00 (average=217,18)
and Accel is 148,00...279,00 (average=214,11)	Then
Then	Diagnosis is 2
Diagnosis is 2	Rule's probability: 0,706
Rule's probability: 0,778	The rule exists in 24 records
The rule exists in 14 records	Significance Level: Error probability <0,3
Significance Level: Error probability <0,2	5. If Sex is F
2. If Sex is F	Then
and Accel is 148,00...275,00 (average=222,14)	Diagnosis is 2
Then	Trend's probability: 0,595

Basic Rules	Basic Trends
Diagnosis is 2	The trend exists in 25 records
Rule's probability: 0,810	6. If Width is 59,00...101,00
The rule exists in 17 records	Then
Significance Level: Error probability <0,2	Diagnosis is 2
3. If Sex is F	Trend's probability: 0,556
and Width is 60,00...85,00 (average=68,29)	The trend exists in 20 records
Then	
Diagnosis is 2	
Rule's probability: 0,706	
The rule exists in 12 records	
Significance Level: Error probability <0,3	

Как видим из таблицы, все компоненты неожиданного правила по отдельности имеют точность существенно ниже 1 — самое высокое значение точности наблюдается у базисного правила № 2, представляющего собой комбинацию двух простых событий (`Sex is F`) и (`Accel is` *148,00:..275,00*).

Нижняя секция отчета о неожиданных правилах разделена на две части (рис. 5.36). В левой части располагаются элементы управления для сортировки этих правил. По умолчанию правила проранжированы по величине разности между реальной и ожидавшейся точностями правила. Если установить переключатель в поле Field и выбрать из списка какой-либо признак, то будут отображаться только те неожиданные правила, в которых встречается указанный признак. В свою очередь, в поле Type можно выбрать один из трех типов фильтров правил: All (все правила), if-then-правила и if-then-NOT.

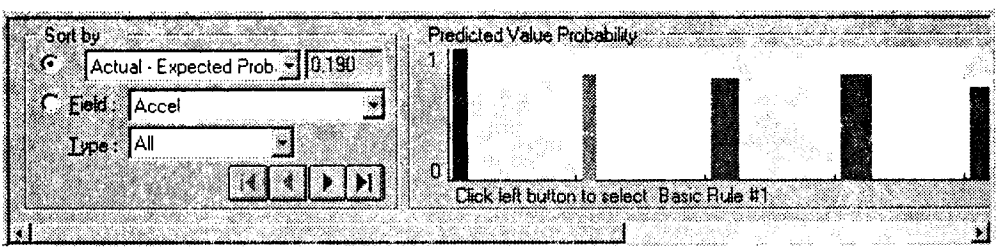


Рис. 5.36. Сортировка и визуализация неожиданных правил

В правой части нижней секции отчета о неожиданных правилах дается графическое представление характеристик правил и их составляющих. Первый слева столбик относится к найденному неожиданному правилу — его высота равна точности, а ширина пропорциональна количеству покрываемых объектов. Следующий столбик отображает ожидавшиеся характеристики правила, а остальные столбики соответствуют описанным выше базисным правилам и трендам. Если щелкнуть левой кнопкой мыши по какому-либо столбику, то система WizWhy автоматически изменит содержание верхних окон отчета о неожиданных прави-

лах. Можно также щелкнуть на столбике правой кнопкой мыши — появляется контекстное меню, в котором можно заказать иллюстрацию в виде диаграммы правила (Rule chat).

Предсказание на основе полученных правил

В системе WizWhy предусмотрены две возможности использования обнаруженных правил для предсказания значений целевого показателя на новом материале.

Первая возможность заключается в ручном вводе значений признаков и обработке нового одиночного объекта (записи). Она реализуется следующим образом.

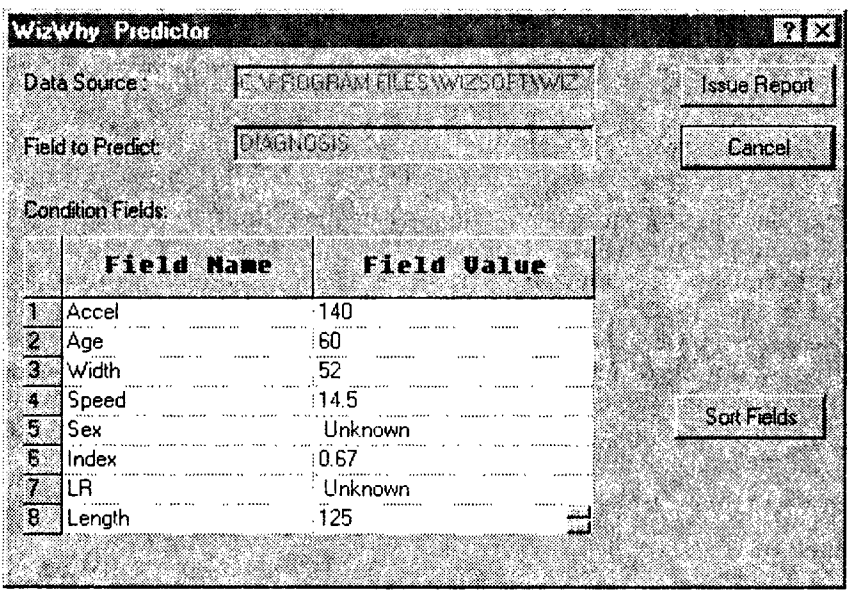


Рис. 5.37. Окно диалога для ручного ввода значений признаков

Нажимаем кнопку Issue Prediction — на экран выдается окно диалога для ручного ввода значений признаков (рис. 5.37). После заполнения окошек предложенной таблицы (здесь возможны пропуски) нажимаем кнопку Issue Report — система находит релевантные правила и создает отчет, в котором подробно описываются как конечный результат предсказания, так и характеристики каждого отдельного правила, сработавшего для данного объекта. Этот отчет приводится ниже.

```
WIZWHY PREDICTION REPORT
Condition Fields:
Age=60.00
Length=125.00
Width=52.00
Speed=14.50
Index=0.67
```

```
Accel=140.00
Field to Predict: Diagnosis
Subject for Prediction: Diagnosis is 2
Prediction's significance level: Error probability=0.259
Primary Prediction's probability: 0.500
Conclusive Prediction's probability: 0.549
Prediction: No 2
Relevant rules:
1. If Age is 56.00...74.00 (average=64.81)
and Speed is 12.50...18.30 (average=15.59)
Then
Diagnosis is 2
Rule's probability: 0.813
The rule exists in 13 records.
Significance Level: Error probability <0.2
2. If Age is 56.00...74.00 (average=66.31)
and Index is 0.63...0.67 (average=0.65)
Then
Diagnosis is 2
Rule's probability: 0.846
The rule exists in 11 records.
Significance Level: Error probability <0.2
3. If Length is 112.00...128.00 (average=118.64)
and Width is 39.00...55.00 (average=49.36)
Then
Diagnosis is not 2
Rule's probability: 0.909
The rule exists in 10 records.
Significance Level: Error probability <0.2
```

Как видим, в нашем конкретном случае система выдала предсказание, что рассматриваемый объект не относится к классу № 2 (Prediction: No 2). Это решение система приняла на основании трех релевантных правил. Хотя два первых правила говорят, что объект является представителем класса № 2 (диагноз «множественные кисты»), но их «побеждает» третье правило, имеющее более высокую точность (0,909).

Вторая возможность использования множества правил заключается в обработке сразу большого массива новой информации. Для этого сначала, перейдя к закладке Prediction Input в окне диалога для ввода данных (см. рис. 5.37), нужно указать файл, в котором записана новая информация. Пусть в нашем случае это будет тот же самый файл с обучающей выборкой USR.txt. Затем требуется задать имя файла, в который будут записываться результаты предсказания. Данная операция осуществляется с помощью кнопки Print result to... И наконец, нажимается кнопка Predict to file — система производит необходимые расчеты и сообщает, что результаты успешно записаны в указанный файл результатов, который приведен в табл. 5.7.

Таблица 5.7. Содержимое файла результатов предсказания

№ п/п	«Diagnosis»	«Sign_Level»	«Concl_Prob»	«Prediction»
1	«1»	0.492	0.481	No 2
2	«1»	0.559	0.089	No 2
3	«1»	0.406	0.526	No 2
4	«1»	0.689	0.088	No 2
5	«1»	0.020	0.030	No 2
6	«1»	0.198	0.072	No 2
7	«1»	0.374	0.532	No 2
8	«1»	0.000.	0.032	No 2
9	«1»	0.511	0.459	No 2
10	«1»	0.186	0.068	No 2
11	«1»	0.173	0.051	No 2
12	«1»	0.433	0.450.	No 2
13	«1»	0.272	0.077	No 2
14	«1»	0.466	0.514	No 2
15	«1»	0.330	0.525	No 2
16	«1»	0.200	0.528	No 2
17	«1»	0.243	0.059	No 2
18	«1»	0.152	0.610	No 2
19	«1»	0.588	0.460	No 2
20	«1»	0.002	0.016	No 2
21	«1»	0.489	0.454	No 2
22	«1»	0.145	0.624	2
23	«2»	0.361	0.743	2
24	«2»	0.097	0.952	2
25	«2»	0.009	0.995	2
26	«2»	0.000	0.960	2
27	«2»	0.384	0.737	2
28	«2»	0.097	0.952	2
29	«2»	0.000	0.963	2
30	«2»	0.210	0.788	2
31	«2»	0.275	0.759	2
32	«2»	0.285	0.745	2
33	«2»	0.009	0.995	2
34	«2»	0.009	0.995	2
35	«2»	0.191	0.613	2
36	«2»	0.000	0.962	2
37	«2»	0.000	0.961	2
38	«2»	0.000	0.961	2
39	«2»	0.001	0.854	2

№ п/п	«Diagnosis»	«Sign_Level»	«Concl_Prob»	«Prediction»
40	«2»	0.228	0.789	2
41	«2»	0.205	0.933	2
42	«2»	0.009	0.995	2
43	«2»	0.097	0.952	2
44	«2»	0.705	0.497	No 2
45	«2»	0.097	0.952	2
46	«2»	0.181	0.770	2
47	«2»	0.000	0.857	2
48	«2»	0.230	0.422	No 2
49	«2»	0.194	0.626	2
50	«2»	0.000	0.975	2
51	«2»	0.009	0.995	2
52	«2»	0.001	0.965	2
53	«2»	0.097	0.952	2
54	«2»	0.009	0.995	2
55	«2»	0.377	0.503	No 2
56	«2»	0.097	0.952	2
57	«2»	0.715	0.498	No 2
58	«2»	0.240	0.930	2
59	«2»	0.009	0.995	2
60	«3»	0.446	0.458	No 2
61	«3»	0.097	0.048	No 2
62	«3»	0.610	0.512	No 2
63	«3»	0.000	0.024	No 2
64	«3»	0.097	0.048	No 2
65	«3»	0.097	0.048	No 2
66	«3»	0.097	0.048	No 2
67	«3»	0.097	0.048	No 2
68	«3»	0.097	0.048	No 2
69	«3»	0.297	0.774	2
70	«3»	0.097	0.048	No 2
71	«3»	0.000	0.022	No 2
72	«3»	0.097	0.048	No 2
73	«3»	0.097	0.048	No 2
74	«3»	0.097	0.048	No 2

Литература

1. Бонгард М. М. Проблема узнавания. — М.: Наука, 1967.
2. Дж. ван Гик. Прикладная общая теория систем. М.: Мир, 1981.

3. Загоруйко Н. Г. Методы распознавания и их применение. — М.: Сов. радио, 1972.
4. Лбов Г. С. Выбор эффективной системы зависимых признаков//Тр. Сиб. отд. АН СССР: Вычислительные системы. 1965. Вып. 19.
5. Чесноков С. В. Детерминационный анализ. — Материалы Internet, 1997.
6. Экспертные системы. Принципы работы и примеры/Пер. с англ.; Под ред. Р. Форсайта. — М.: Радио и связь, 1987.
7. Boulding K. E. General Systems Theory — The Skeleton of Science//Management Science. 1956. 2.

ПРИЛОЖЕНИЯ

1 ПРИМЕР

Выяснение причин неурожайности сельскохозяйственных участков

Исходные данные

Исходные данные заимствованы из книги (Кильдишев Г. С., Аболенцев Ю. И. Многомерные группировки. — М.: Статистика, 1978).

На 43 опытных участках по возделыванию риса был получен различный урожай. Агротехника возделывания культуры характеризовалась следующими признаками:

x_1 — предшественник (в баллах);

x_2 — количество удобрений (ц на 1 га);

x_3 — прополка (раз);

x_4 — число дней от залива до сброса воды;

x_5 — число дней от косовицы до обмолота.

Экспериментальные данные представлены в табл. П.1.

Таблица П.1. Значения признаков для участков с различной урожайностью риса

№ п/п	Урожайность, ц с 1 га	Предшественник, баллы	Кол-во удобрений, ц на 1 га	Прополка, раз	Число дней от залива до сброса воды	Число дней от косовицы до обмолота
	Y	x_1	x_2	x_3	x_4	x_5
			Группа 1			
1	36	2,8	1,47	1,2	115	8
2	36,1	3	1,23	1,3	117	7
3	36,1	2,7	1,31	1,4	114	9
4	36,2	3	1,5	1,5	119	10
5	36,4	3,2	1,14	1,6	120	7
6	36,9	2,8	1,22	1,6	121	11
7	37,5	2,7	1,3	1,3	122	8

№ п/п	Урожайность, ц с 1 га	Предшественник, баллы	Кол-во удобрений, ц на 1 га	Прополка, раз	Число дней от залива до сброса воды	Число дней от косовицы до обмолота
	Y	x_1	x_2	x_3	x_4	X_5
			Группа 1			
8	37,8	3,3	1,24	1,3	118	10
9	38,2	2,8	1,16	1,9	119	7
10	38,6	2,7	1,22	1,6	117	9
11	38,9	2,8	1,35	1,2	119	10
12	39	2,9	1,4	1,4	115	8
13	39	3,1	1,36	1,3	120	11
14	39,2	2,8	1,23	1,6	114	10
15	39,4	2,7	1,3	1,4	118	9
16	39,5	3	1,41	1,3	117	8
17	39,7	2,9	1,28	1,4	120	12
18	39,7	3,1	1,36	1,2	121	9
19	39,8	2,8	1,32	1,4	118	7
20	40	2,9	1,4	1,5	118	10
			Группа 2			
21	41,2	3,2	1,05	1,5	109	9
22	41,4	2,8	1,1	1,2	108	10
23	41,6	2,9	1,2	1,6	118	10
24	41,8	3	1,12	1,3	110	14
25	41,9	3,3	1,08	1,4	112	12
26	42,2	2,7	1,13	1,5	111	15
27	42,5	3	1,18	1,7	112	12
28	42,8	3,1	1,22	1,3	113	14
29	43,1	3,3	1,25	1,8	112	13
30	43,1	2,9	1,1	1,7	113	10
31	43,2	2,8	1,2	1,8	112	15
32	43,6	3,2	1,26	1,6	113	9
33	43,7	3,4	1,28	1,8	110	12
34	43,8	3,5	1,22	1,9	114	13
35	43,8	3	1,19	1,7	108	16
36	43,9	2,8	1,29	1,7	108	12
37	43,9	2,9	1,24	1,6	112	10
38	44,2	3	1,17	1,8	114	9
39	44,6	3,3	1,25	1,3	115	11
40	44,8	3,4	1,27	1,7	112	12
41	44,9	3,5	1,26	1,5	111	14
42	44,9	3,1	1,3	1,5	119	11
43	45	3,2	1,24	1,6	110	13

В первоисточнике на основании специальных критериев математической статистики делается вывод, что рассматриваемая совокупность из 43 опытных участков по возделыванию риса не может считаться однородной. На этом обсуждение экспериментального материала фактически заканчивается. Главный вопрос о взаимосвязях между агротехническими мероприятиями и урожайностью рассматриваемой сельскохозяйственной культуры остался не раскрытым.

Ниже представлены результаты комплексного исследования экспериментальных данных из табл. П.1. На первом этапе эти данные были обработаны рядом традиционных методов прикладной статистики с помощью пакета программ STATGRAPHICS *Plus* for Windows (version 2.2). А на втором этапе поиска ответа на сформулированный главный вопрос применялись методы поиска логических закономерностей.

Используемые обозначения: **yield** — урожайность; **predec** — предшественник, **fertil** — количество удобрений; **weeding** — прополка; **water** — число дней от залива до сброса воды; **trashing** — число дней от косовицы до обмолота.

Комплексная обработка данных традиционными методами

Сравнение средних значений признаков

На рис. П.1 приведены совмещенные гистограммы распределений значений всех анализируемых признаков (**class**=0 — объекты с низкой урожайностью, **class**=1 — с высокой).

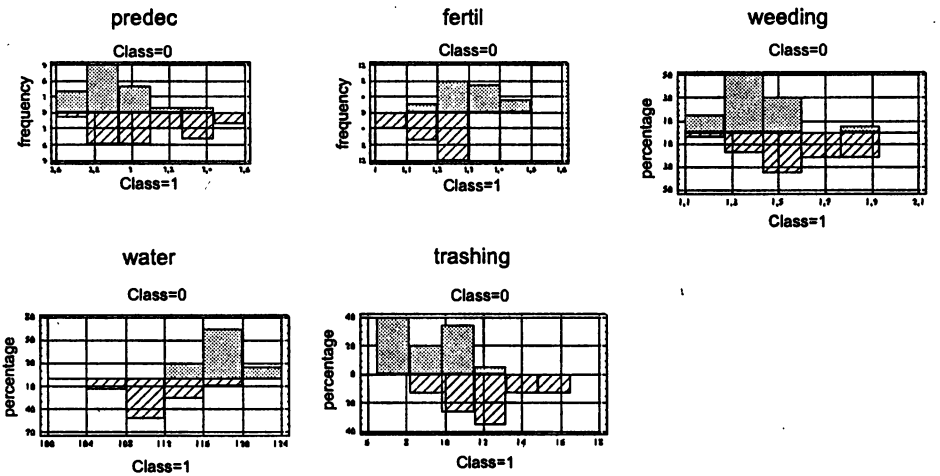


Рис. П.1. Гистограммы распределения значений признаков

Из гистограмм следует, что более высокая урожайность чаще встречается на участках с более высокой балльной оценкой предшественника, со средними и ниже количествами внесенных удобрений, с более частой прополкой, с более ранним сбросом воды и более поздним обмолотом после косовицы. Эти выводы подтверждаются результатами Т-тестов для сравнения средних значений на уровне значимости 0,001.

Вместе с тем, более глубокие выводы делаются на основании многомерного анализа экспериментальных данных, в котором учитывается совокупное взаимодействие признаков.

Метод главных компонент

В результате применения метода главных компонент (МГК) оказалось, что сильное разделение двух классов сельскохозяйственных участков имеют проекции объектов на 1-ю главную компоненту, на которую приходится 45 % дисперсии анализируемой выборки. Весовые коэффициенты для этой главной компоненты приведены в табл. П.2.

Таблица П.2. Веса признаков в первой главной компоненте

Признак	Predec	Fertil	Weeding	Water	Trashing
Вес	0,32	−0,43	0,40	−0,54	0,50

Представление о разделении проекций классов на 1-ю главную компоненту дает рис. П.2.

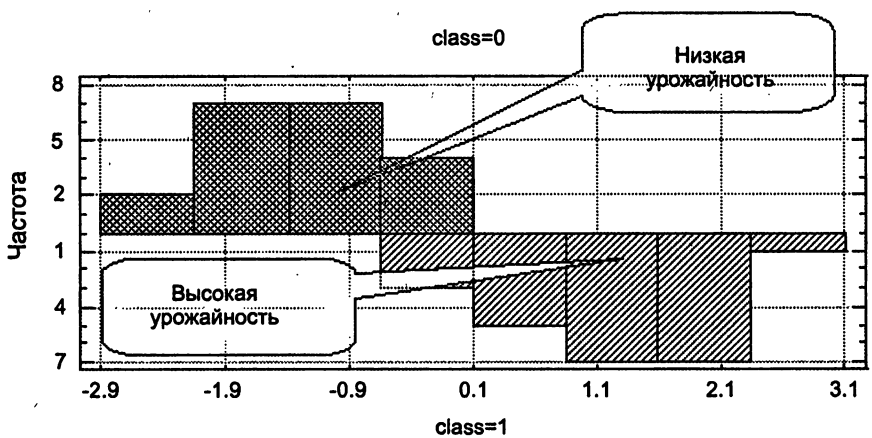


Рис. П.2. Проекция объектов на 1-ю главную компоненту

Из рисунка следует, что на проекции имеется сравнительно небольшая область неопределенности, в которую попадают 3 участка с высокой урожайностью и 4 с низкой. В целом смысл, который имеет 1-я главная компонента, определяемый по весам входящих в нее признаков, совпадает с интерпретацией одномерного

анализа — повышение урожайности риса положительно связано с балльной оценкой предшественника, с частотой прополки и с временным интервалом от косовицы до молотьбы и отрицательно связано с количеством внесенных удобрений и числом дней от залива до сброса воды.

Множественный регрессионный анализ

Сводка множественного регрессионного анализа, в котором независимой переменной выступает урожайность участков **yield**, а предикторами служат вышеописанные признаки x_1, ..., x_5, приведена ниже. При построении регрессионной модели применялся алгоритм последовательного уменьшения группы признаков.

Как следует из полученной сводки, регрессионная модель заслуживает более 99 % доверия. Однако коэффициент детерминации сравнительно невысок и составляет 57 %, а средняя абсолютная ошибка слишком велика, чтобы модель могла претендовать на точный прогноз урожайности, и равна 1,5. Фактически эта модель пригодна для ориентировочного осмысления статистической связи независимой переменной и предикторов. Из модели следует, что рост урожайности риса положительно связан с балльной оценкой предшественника и числом дней между косовицей и обмолотом и отрицательно связан с числом дней от залива до сброса воды.

Листинг (табличный) П.3. Сводка множественного регрессионного анализа

```
Dependent variable: yield
                                        Standard          T
Parameter                Estimate         Error       Statistic        P-
Value
CONSTANT                  55,4292        12,6433        4,38407
0,0001
predec                    3,71752        1,40207        2,65144
0,0115
water                    -0,262156       0,09313       -2,81495
0,0076
trashing                  0,424119       0,161582         2,6248
0,0123

                        Analysis of Variance
Source              Sum of Squares     Df  Mean Square    F-Ratio      P-
Value
Model                      208,339      3      69,4465      17,40
0,0000
Residual                   155,661     39       3,9913
Total (Corr.)                364,0     42
R-squared = 57,2361 percent
R-squared (adjusted for d.f.) = 53,9466 percent
Standard Error of Est. = 1,99782
Mean absolute error = 1,50813
Durbin-Watson statistic = 1,04477
```

Дискриминантный анализ

Применялся классический вариант дискриминантного анализа, основанный на определении канонических направлений в исходном пространстве признаков. Обучающая информация задавалась переменной **class**, которая принимает значение 0 для объектов с низкой урожайностью и значение 1 в группе объектов с высокой урожайностью. Дискриминантный анализ проводился с применением алгоритма последовательного уменьшения группы признаков. Результаты приведены в табл. П.4 и П.5. Гистограмма распределения значений дискриминантной функции показана на рис. П.3.

Таблица П.4. Стандартизированные коэффициенты дискриминантной функции

Predec	Fertil	Water	Trashing
−0,459109	0,476217	0,680748	−0,446469

Таблица П.5. Таблица классификаций

Номер класса	Объем группы	Предсказанный класс	
		0	1
0	20	20 (100 %)	0 (0 %)
1	23	2 (8,7 %)	21 (91,3 %)
Процент правильной классификации — 95,35 %			

По формальному эффекту дискриминантная функция обеспечивает несколько лучшее разделение групп участков с различной урожайностью риса по сравнению с проекциями объектов на 1-ю главную компоненту. Здесь только два участка с высокой урожайностью ошибочно относятся к группе с низкой урожайностью, а коэффициент канонической корреляции дискриминантной функции с переменной **class** составляет 0,87.

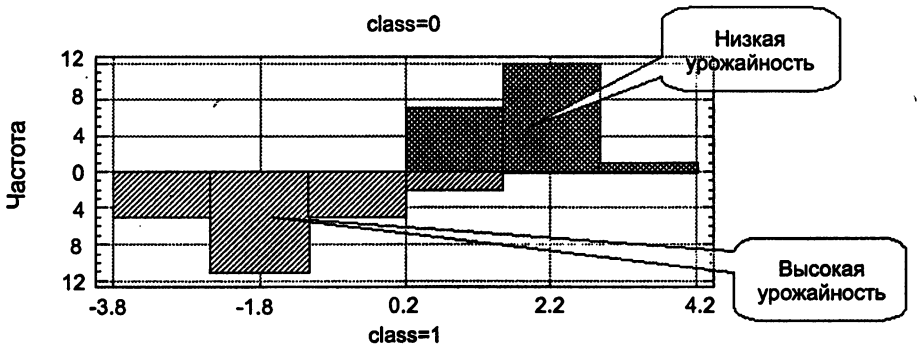


Рис. П.3. Гистограмма распределения значений дискриминантной функции

Вместе с тем содержательная сторона проведенного дискриминантного анализа, как и в предыдущих видах анализа, носит качественный характер. А именно, можно лишь утверждать, что повышение урожайности положительно связано с балльной оценкой предшественника **predec** и числом дней от косовицы до обмолота **trashing** и отрицательно связано с количеством внесенных удобрений **fertil** и временным интервалом от залива до сброса воды **water**.

В целом результаты комплексной обработки агротехнических характеристик участков с различной урожайностью риса с применением аппарата одномерного и многомерного статистического анализа дают основание лишь для выявления общих тенденций. Кроме того, наблюдаются некоторые расхождения результатов различных методов. Так, метод главных компонент говорит о важности учета всех пяти агротехнических признаков, модель дискриминантного анализа не испытывает необходимости учета прополки, а регрессионный анализ, кроме всего этого, показывает, что не требуется привлекать для объяснений агротехнических закономерностей количество удобрений. Это происходит, главным образом, из-за внутригрупповой неоднородности экспериментальных данных.

Агротехническая система имеет высокий уровень сложности, и для более глубокого и полного проникновения в ее суть должен применяться аппарат, умеющий выявлять и учитывать структурные неоднородности данных об этой системе. Такими свойствами обладают методы индуктивного логического вывода, позволяющие находить в ограниченном наборе экспериментальных фактов логические закономерности.

Результаты обработки данных системой See5

Decision tree:

```
(дней от залива до сброса воды) <= 113: 0 (18.0)
(дней от залива до сброса воды) > 113:
:...(дней от косовицы до обмолота) <= 10: 1 (19.0/2.0)
    (дней от косовицы до обмолота) > 10:
    :...(дней от залива до сброса воды) <= 119: 0 (3.0)
        (дней от залива до сброса воды) > 119: 1 (3.0)
```

Extracted rules:

```
Rule 1: (cover 18)
                    (дней от залива до сброса воды) <= 113
                    ->  высокая урожайность  [0.950]
Rule 2: (cover 16)
                    (дней от залива до сброса воды) <= 119
                    (дней от косовицы до обмолота) > 10
                    ->  высокая урожайность  [0.944]
Rule 3: (cover 6)
                    (дней от залива до сброса воды) > 119
```

```
                -> низкая урожайность [0.875]
Rule 4: (cover 19)
                (дней от залива до сброса воды) > 113
                (дней от косовицы до обмолота) <= 10
                -> низкая урожайность [0.857]
```

Ошибки классификации на обучающей выборке:

```
        Decision Tree          Rules
      --------    --------
      Size      Errors      No      Errors
         4    2( 4.7%)       4    2( 4.7%)   <<

       (a)   (b)      <-classified as
      ----  ----
        21     2      (a): высокая урожайность
               20     (b): низкая урожайность
```

Результаты обработки данных системой WizWhy

Ниже представлен отчет системы WizWhy для следующих установочных параметров поиска логических правил в экспериментальных данных:

- Целевая переменная — **урожай**.
- Минимальная вероятность if-then-правила — 0,7.
- Минимальная вероятность if-then-NOT-правила — 0,7.
- Минимальное количество объектов, покрываемых правилом, — 10.

```
PARAMETERS OF THE RULES AND DATA
C:\Program Files\WizSoft\WizWhy Demo\Data\RisYield.txt
Total number of records: 43
Minimum probability of the:
1.                    if-then rules: 0,700
2.                    if-then-not rules: 0,700
Minimum number of cases in a rule: 10

Field to Predict: Урожай
Predicted Value (analyzed as Boolean): more than 40,93
Prediction error costs:
The cost of a miss: 1
The cost of a false alarm: 1
Average probability of the predicted value is 0,535

ANALYSIS OF THE RULES EXPLANATORY POWER
Decision point: Predict more than 40,93 when conclusive probability is more than 0,484
Number of misses: 1
Number of false alarms: 2
```

Total number of errors: 3
Total cost of errors: 3
Success rate when predicting more than 40,93: 0,917
Success rate when predicting NOT more than 40,93: 0,947
Number of records with no relevant rules: 0
Average cost (per record): 0,070
Expected average cost (per record): 0,488
Improvement Factor: 7,000

IF-THEN RULES:
1. If **Дней от залива до сброса воды** is *108,00...113,00* (average=*110,89*)
Then
Урожай is more than *40,93*
Rule's probability: 1,000
The rule exists in 18 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
21, 22, 24, 25, 26, 27, 28, 29, 30, 31

2. If **Предшественник** is *2,70...2,80* (average=*2,76*)
and **Дней от залива до сброса воды** is *114,00...122,00* (average=*117,70*)
Then
Урожай is not *more than 40,93*
Rule's probability: 1,000
The rule exists in 10 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 3, 6, 7, 9, 10, 11, 14, 15, 19

3. If **К-во удобрений** *is 1,28...1,47* (average=*1,35*)
and **Прополка** is *1,20...1,40* (average=*1,32*)
Then
Урожай is not more than *40,93*
Rule's probability: 1,000
The rule exists in 11 records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 3, 7, 11, 12, 13, 15, 16, 17, 18

4. If **К-во удобрений** is *1,28...1,50* (average=*1,36*)
and **Дней от залива до сброса воды** *is 114,00...122,00* (average=*118,21*)
Then
Урожай is not more than *40,93*
Rule's probability: 0,929
The rule exists in 13 records.

Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 3, 4, 7, 11, 12, 13, 15, 16, 17
Negative Examples (records' serial numbers):
42

5. If **Прополка** is *1,20...1,40* (average=*1,31*)
and **Дней от залива до сброса воды** is *114,00...122,00* (average=*117,79*)
Then
Урожай *is not more than 40,93*
Rule's probability: **0,929**
The rule exists in **13** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 2, 3, 7, 8, 11, 12, 13, 15, 16
Negative Examples (records' serial numbers):
39

6. If **Дней от залива до сброса воды** *is 114,00...122,00* (average=*117,46*)
and **Дней от косовицы до обмолота** is *7,00...9,00* (average=*8,08*)
Then
Урожай is not *more than 40,93*
Rule's probability: **0,923**
The rule exists in **12** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 2, 3, 5, 7, 9, 10, 12, 15, 16
Negative Examples (records' serial numbers):
38

7. If **Дней от залива до сброса воды** is *114,00...122,00* (average=*117,68*)
Then
Урожай is not *more than 40,93*
Rule's probability: **0,800**
The rule exists in **20** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Negative Examples (records' serial numbers):
23, 34, 38, 39, 42

8. If **К-во удобрений** is *1,28...1,50* (average=*1,35*)
Then
Урожай is not *more than 40,93*
Rule's probability: **0,813**
The rule exists in **13** records.

Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 3, 4, 7, 11, 12, 13, 15, 16, 17
Negative Examples (records' serial numbers):
33, 36, 42

9. If **Дней от косовицы до обмолота** is *7,00...9,00* (average=*8,20*)
Then
Урожай is not *more than 40,93*
Rule's probability: **0,800**
The rule exists in 12 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 3, 5, 7, 9, 10, 12, 15, 16
Negative Examples (records' serial numbers):
21, 32, 38

10. If **К-во удобрений** is *1,08...1,27* (average=*1,20*)
and **Дней от косовицы до обмолота** is *10,00...16,00* (average=*12,25*)
Then
Урожай is *more than 40,93*
Rule's probability: **0,850**
The rule exists in 17 records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
22, 23, 24, 25, 26, 27, 28, 29, 30, 31
Negative Examples (records' serial numbers):
6, 8, 14

11. If **Предшественник** is *3,20...3,50* (average=*3,32*)
Then
Урожай *is more than 40,93*
Rule's probability: **0,833**
The rule exists in 10 *records*.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
21, 25, 29, 32, 33, 34, 39, 40, 41, 43
Negative Examples (records' serial numbers):
5, 8

12. If **Прополка** is *1,50...1,90* (average=*1,65*)
and **Дней от косовицы до обмолота** is *10,00...16,00* (average=*12,05*)
Then
Урожай is *more than 40,93*
Rule's probability: **0,789**
The rule exists in 15 records.

Significance Level: Error probability <0.2
Positive Examples (records' serial numbers):
23, 26, 27, 29, 30, 31, 33, 34, 35, 36
Negative Examples (records' serial numbers):
4, 6, 14, 20

13. If **Предшественник** is *2,70...2,80* (average=*2,76*)
Then
Урожай is not *more than 40,93*
Rule's probability: 0,714
The rule exists in 10 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
1, 3, 6, 7, 9, 10, 11, 14, 15, 19
Negative Examples (records' serial numbers):
22, 26, 31, 36

14. If **Прополка** is *1,20...1,40* (average=*1,31*)
Then
Урожай is not *more than 40,93*
Rule's probability: 0,722
The rule exists in 13 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
1, 2, 3, 7, 8, 11, 12, 13, 15, 16
Negative Examples (records' serial numbers):
22, 24, 25, 28, 39

15. If **К-во удобрений** is *1,05...1,27* (average=*1,19*)
Then
Урожай is *more than 40,93*
Rule's probability: 0,741
The rule exists in 20 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
21, 22, 23, 24, 25, 26, 27, 28, 29, 30
Negative Examples (records' serial numbers):
2, 5, 6, 8, 9, 10, 14

16. If **Прополка** is *1,50...1,90* (average=*1,65*)
Then
Урожай is *more than 40,93*
Rule's probability: 0,720
The rule exists in 18 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
21, 23, 26, 27, 29, 30, 31, 32, 33, 34
Negative Examples (records' serial numbers):

4, 5, 6, 9, 10, 14, 20

17. If Дней от косовицы до обмолота is *10,00...16,00* (average=*11,89*)
Then
Урожай is *more than 40,93*
Rule's probability: 0,714
The rule exists in 20 records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
22, 23, 24, 25, 26, 27, 28, 29, 30, 31
Negative Examples (records' serial numbers):
4, 6, 8, 11, 13, 14, 17, 20

Полученная система 17 логических правил обладает следующими характеристиками:

- вероятность правильного предсказания высокого урожая (больше 40,93 ц/га) — 0,917;
- вероятность правильного предсказания низкого урожая (меньше 40,93 ц/га) — 0,947.

Фактически полученные правила представляют собой инструкцию агротехнику с указанием конкретных значений факторов и их комбинаций, влияющих на урожайность рисовых участков. Это довольно обстоятельная и длинная инструкция, но, по-видимому, иначе вряд ли возможно описать сложную систему многофакторного взаимодействия с удовлетворительной точностью.

При сравнении результатов двух систем — See5 и WizWhy — бросается в глаза, что система See5 обнаружила всего 4 правила, но этого оказалось достаточно для описания рассмотренной агротехнической ситуации (только 2 ошибки на обучающей выборке). Вместе с тем, одно из четырех обнаруженных правил в системе See5 обладает малой полнотой (охватывает 6 случаев). Это, по-видимому, несколько снижает его ценность.

2 ПРИМЕР

Сравнение структуры интеллекта «физиков» и «лириков»[1]

В рассматриваемом примере исследуются экспериментальные данные, представляющие собой результаты психологического тестирования учащихся специализированных школ Санкт-Петербурга с физико-математическим и гуманитарным уклоном. Сначала эти данные будут обработаны традиционными методами статистического анализа. Затем мы применим к экспериментальному материалу технологию поиска логических закономерностей.

Нам предстоит на практике убедиться, что к исследованиям людей далеко не всегда следует подходить с позиций классической математической статистики. Это приводит к малосодержательным и нередко бесполезным выводам. В то же время, применение аппарата поиска в данных логических высказываний способно привести к раскрытию ценных многоаспектных знаний о людях.

Общая характеристика данных

Объектами исследования являются 76 учащихся специализированных школ:

- 38 человек — учащиеся 10-х классов физико-математической школы № 30. В дальнейшем они будут называться «физики»;
- 38 человек — учащиеся 10-х классов гуманитарных школ (21 учащийся литературной школы № 27 и 17 учащихся художественной школы № 363). Назовем их «лирики».

Целью исследования являлось определение различий в структуре интеллекта «физиков» и «лириков». Естественно было заранее предположить, что у «физиков» более развит «левополушарный» вербальный, а у «лириков» — «правопо-

[1] Пример подготовлен совместно с сотрудниками Санкт-Петербургского государственного института психологии и социальной работы А. В. Паронько, А. В. Ямаевой и А. В. Пономаревым.

лушарный» невербальный интеллект. Исходя из этого выбирался инструмент психологического исследования, позволяющий оценивать как вербальную, так и невербальную стороны мышления. Конкретно использовались субтесты IN, AN, GE, AR и PL популярного теста Р. Амтхауэра. Ниже этим субтестам дается краткая характеристика:

- Тест № 1 (IN) — «дополнение предложений». Он предназначен для оценки способности к рассуждению, здравого смысла, сложившейся самостоятельности мышления (для юношеского возраста). По отношению к этому тесту будем использовать термин «здравомыслие».
- Тест № 3 (AN) — «аналогии». Этот тест отражает способность комбинировать, подвижность и непостоянство мышления.
- Тест № 4 (GE) — «обобщение». Тест предназначен для определения способности к абстрактному мышлению, образованию понятий, умению словесно выражать мысль.
- Тест №5 (AR) — «арифметические задачи». Данный тест отражает развитость практического численного мышления.
- Тест № 7 (PL) — «выбор геометрического образца». Тест направлен на оценку воображения, богатства представлений, наглядного целостного мышления. Сопоставим этому тесту термин «воображение».

Результаты психологического тестирования «физиков» и «лириков» представлены в табл. П.6. В этой же таблице указан пол испытуемых. Среди «физиков» 12 девушек и 26 юношей, среди лириков 20 девушек и 18 юношей.

Таблица П.6. Результаты психологического тестирования «физиков» и «лириков»

№ п/п	Тест 1 (здравомыслие)	Тест 3 (аналогии)	Тест 4 (обобщение)	Тест 5 (численное мышление)	Тест 7 (воображение)	Пол
			«Физики»			
1	15	16	19	13	8	Мужской
2	16	11	16	20	8	Мужской
3	11	16	14	15	12	Мужской
4	15	10	9	18	10	Мужской
5	8	11	10	16	10	Женский
6	11	13	13	7	11	Женский
7	7	9	13	11	13	Женский
8	11	15	14	12	8	Женский
9	12	15	18	10	10	Женский
10	14	10	16	12	10	Мужской
11	8	4	16	14	8	Мужской
12	9	13	12	15	6	Мужской

№ п/п	Тест 1 (здравомыслие)	Тест 3 (аналогии)	Тест 4 (обобщение)	Тест 5 (численное мышление)	Тест 7 (воображение)	Пол
			«Физики»			
13	10	11	18	12	10	Мужской
14	10	11	16	14	9	Мужской
15	11	10	14	13	15	Мужской
16	10	15	17	19	11	Мужской
17	12	13	13	11	8	Мужской
18	11	12	16	17	10	Мужской
19	9	12	16	14	11	Женский
20	11	14	11	17	13	Мужской
21	11	14	15	10	7	Мужской
22	15	17	16	12	12	Мужской
23	11	12	16	12	6	Мужской
24	13	7	17	6	8	Мужской
25	13	15	13	7	10	Мужской
26	18	15	17	7	11	Женский
27	18	17	15	9	8	Мужской
28	16	18	23	6	10	Мужской
29	15	18	23	15	11	Мужской
30	14	10	19	10	13	Женский
31	13	10	10	6	7	Мужской
32	14	15	19	8	7	Женский
33	18	15	18	9	10	Женский
34	10	12	18	10	11	Мужской
35	14	12	16	7	13	Мужской
36	17	17	24	13	9	Женский
37	10	12	22	9	14	Мужской
38	11	15	19	11	8	Женский
			«Лирики»			
39	6	7	10	4	10	Мужской
40	5	7	15	2	7	Женский
41	9	10	15	4	9	Женский
42	8	14	15	5	14	Женский
43	10	7	11	8	6	Мужской
44	12	8	9	8	7	Мужской
45	6	8	8	7	9	Мужской

продолжение

Таблица П.6 *(продолжение)*

№ п/п	Тест 1 (здравомыслие)	Тест 3 (аналогии)	Тест 4 (обобщение)	Тест 5 (численное мышление)	Тест 7 (воображение)	Пол
			«Лирики»			
46	10	12	20	14	11	Мужской
47	14	4	14	7	11	Мужской
48	6	6	12	11	9	Мужской
49	8	10	19	3	5	Женский
50	8	9	20	3	5	Женский
51	4	7	12	5	10	Женский
52	11	12	16	7	6	Женский
53	7	6	5	3	4	Мужской
54	10	3	9	5	6	Женский
55	15	14	19	7	6	Женский
56	10	11	12	8	15	Мужской
57	6	10	12	9	15	Мужской
58	10	9	12	5	10	Женский
59	10	9	12	5	10	Мужской
60	11	12	11	5	11	Мужской
61	10	11	13	4	11	Мужской
62	7	10	12	7	10	Женский
63	10	12	12	6	5	Мужской
64	7	11	10	5	8	Женский
65	9	10	16	4	12	Женский
66	10	8	18	4	7	Женский
67	12	10	19	6	13	Женский
68	15	4	14	5	9	Женский
69	12	9	18	4	9	Женский
70	11	6	18	5	8	Мужской
71	12	15	20	6	8	Мужской
72	13	14	25	15	13	Женский
73	15	8	14	14	4	Женский
74	13	15	19	7	8	Женский
75	10	12	18	6	11	Мужской
76	13	9	14	8	10	Женский

Сравнение средних значений результатов тестирования в группах «физиков» и «лириков»

Сравнение средних значений каких-либо показателей с помощью статистических Т-критериев считается важнейшим видом анализа экспериментально-психологических данных. Применим этот метод к нашим данным. В табл. П.7 приведены средние значения и стандартные отклонения результатов субтестов по группам «физиков» и «лириков».

Таблица П.7. Средние значения и стандартные отклонения результатов субтестов

	Тест 1		Тест 3		Тест 4		Тест 5		Тест 7	
	Физики	Лирики	Физики	Лирики	Физики	Лирики	Физики	Лирики	Физики	Лирики
Среднее	12,4	9,8	12,9	9,4	16,1	14,4	11,8	6,3	9,9	9,0
Стандарт. отклонение	2,9	2,9	3,1	3,0	3,5	4,2	3,8	9,1	2,2	2,9

На основании статистических критериев по данным таблицы можно сделать следующие выводы:

1. Среднее «здравомыслие» в группе «физиков» выше среднего «здравомыслия» в группе «лириков» с 99 % достоверности.
2. Средняя «способность к аналогии» в группе «физиков» выше средней «способности к аналогии» в группе «лириков» с 99 % достоверности.
3. Средняя «способность к обобщению» в группе «физиков» выше средней «способности к обобщению» в группе «лириков» с 95 % достоверности.
4. Средняя «способность к «численному мышлению» в группе «физиков» выше средней «способности к численному мышлению» в группе «лириков» более чем с 99 % достоверности.

На рис. П.4, где приведены гистограммы распределения тестовых оценок, невооруженным глазом видно, что особенно сильное различие между «физиками» и «лириками» наблюдается в способности к численному мышлению (тест 5).

Полученные результаты классического статистического анализа экспериментальных данных достаточно тривиальны. Можно было заранее без проведения специального исследования предположить, что у учащихся физико-математических школ окажутся в среднем более высокие способности к установлению аналогий, обобщению (абстрактному мышлению) и умению решать арифметические задачи. Хотя слегка удивляют их более высокая способность к самостоятельному мышлению (результаты анализа по тесту 1), а также низкие способности гуманитариев решать простые арифметические задачи (тест 5).

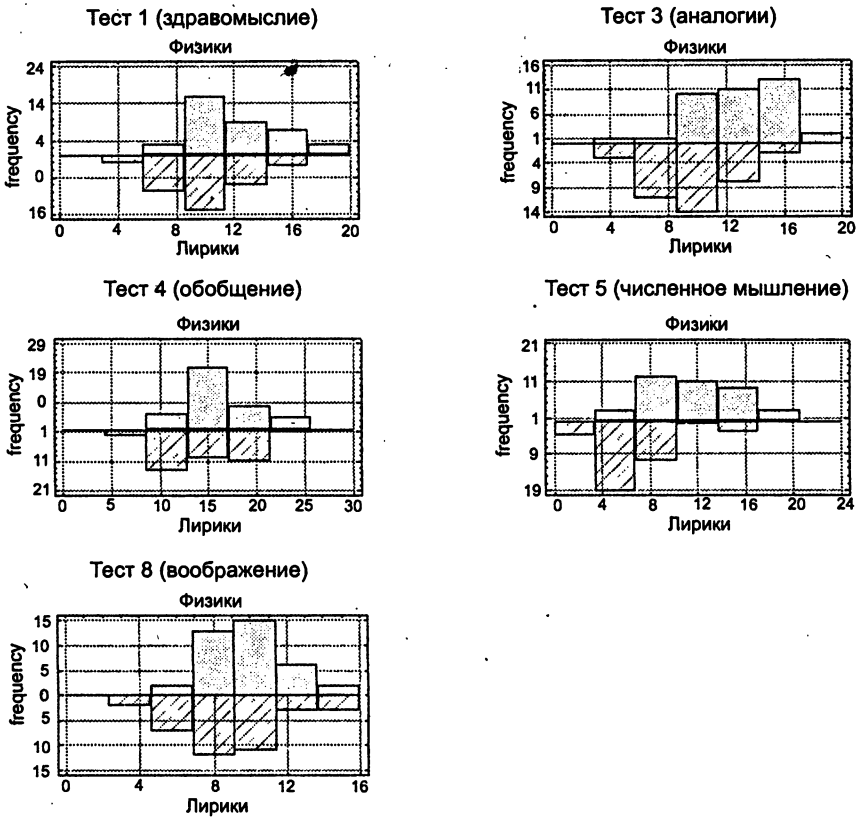


Рис. П.4. Сравнительные гистограммы распределения тестовых оценок

Вместе с тем, представленный экспериментальный материал дает основание для попыток установления более глубоких и содержательных отличий гуманитариев от естественников, чем это выявляется с помощью классического статистического аппарата, оперирующего усредненными характеристиками совокупностей объектов. Вообще усреднение характеристик объектов с высоким уровнем сложности системной организации является во многом фиктивной операцией, не имеющей смысла (по выражению Б. С. Ястремского, не имеет смысла средняя высота дома на улице, состоящей из дворцов и лачуг). Совокупности таких объектов являются принципиально неоднородными, полиморфными, и для описания закономерностей таких совокупностей должен использоваться аппарат, умеющий выявлять и учитывать указанные неоднородности.

Такими свойствами обладают методы индуктивного логического вывода, позволяющие находить в ограниченном наборе экспериментальных данных логические закономерности, рассмотренные в главе 5. Мы будем искать логические закономерности вида

(событие 1) и (событие 2) и ... и (событие N),

характерные для каждой из исследуемых групп учащихся. Под событием здесь понимается, что значения какого-либо определенного теста попадают в некоторый заданный интервал значений. Таким образом, логическая закономерность в данном случае представляет собой «комплекс психологических характеристик, часто встречающийся у одной группы испытуемых и редко у другой».

Поиск логических закономерностей системой WizWhy

Ниже применяются следующие обозначения:

T1 — тест 1; T3 — тест 3; T4 — тест 4; T5 — тест 5; T7 — тест 7; Sex — пол (1 — мужской, 0 — женский); значок «&» — логическая связка «и».

```
PARAMETERS OF THE RULES AND DATA
D:\Old\Knowledge\Школа
Total number of records: 76
Minimum probability of the:
                        1) if-then rules: 0,700
                        2) if-then-not rules: 0,700
Minimum number of cases in a rule: 10
Demo version is limited to 1000 records.This database has more than 1000 records.
Calculation will be cancelled.

Field to Predict: Class
Predicted Value (analyzed as Boolean): Physics
Prediction error costs:
The cost of a miss: 1
The cost of a false alarm: 1
Average probability of the predicted value is 0,500

ANALYSIS OF THE RULES EXPLANATORY POWER
Decision point:  Predict Physics when conclusive probability is more than  0,454
Number of misses:   2
Number of false alarms:   9
Total number of errors:   11
Total cost of errors:   11
Success rate when predicting Physics :   0,800
Success rate when predicting  NOT Physics :   0,935
Number of records with no relevant rules :   0
Average cost (per record):   0,145
Expected average cost (per record) :   0,500
Improvement Factor :   3,455

IF-THEN RULES:

1. If Здравомыслие is 4,00...10,00 (average=8,29)
and Арифметич. задачи is 2,00...6,00 (average=4,29)
Then
```

Class is not Physics
Rule's probability: **1,000**
The rule exists in **17** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
39, 40, 41, 42, 49, 50, 51, 53, 54, 58

2. If **Арифметич. задачи** is *2,00...6,00* (average=*4,69*)
Then
Class is not *Physics*
Rule's probability: **0,885**
The rule exists in **23** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
39, 40, 41, 42, 49, 50, 51, 53, 54, 58
Negative Examples (records' serial numbers):
24, 28, 31

3. If **Здравомыслие** is *4,00...10,00* (average=*7,68*)
and **Аналогии** *is 3,00...10,00* (average=*7,84*)
Then
Class is not *Physics*
Rule's probability: **0,895**
The rule exists in **17** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
39, 40, 41, 43, 45, 48, 49, 50, 51, 53
Negative Examples (records' serial numbers):
7, 11

4. If **Аналогии** is *12,00...18,00* (average=*14,46*)
and **Арифметич. задачи** *is 7,00...19,00* (average=*11,67*)
and **Воображение** *is 8,00...14,00* (average=*10,46*)
Then
Class is *Physics*
Rule's probability: **0,875**
The rule exists in **21** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
1, 3, 6, 8, 9, 16, 17, 18, 19, 20
Negative Examples (records' serial numbers):
46, 72, 74

5. If **Обобщение** *is 15,00...17,00* (average=*16,09*)
and **Арифметич. задачи** *is 7,00...20,00* (average=*13,18*)
and **Воображение** *is 8,00...13,00* (average=*10,09*)
Then
Class is *Physics*
Rule's probability: **1,000**

The rule exists in **11** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
2, 10, 11, 14, 16, 18, 19, 22, 26, 27

6. If **Здравомыслие** is *4,00...10,00* (average=*7,45*)
and **Аналогии** *is 3,00...10,00* (average=*7,45*)
and **Обобщение** *is 5,00...12,00* (average=*10,45*)
Then
Class is not *Physics*
Rule's probability: **1,000**
The rule exists in **11** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
39, 43, 45, 48, 51, 53, 54, 57, 58, 59

7. If **Аналогии** is *12,00...18,00* (average=*14,23*)
and **Арифметич. задачи** is *7,00...19,00* (average=*11,30*)
Then
Class is *Physics*
Rule's probability: **0,833**
The rule exists in **25** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 3, 6, 8, 9, 12, 16, 17, 18, 19
Negative Examples (records' serial numbers):
46, 52, 55, 72, 74

8. If **Обобщение** is *15,00...17,00* (average=*16,00*)
and **Арифметич. задачи** is *7,00...20,00* (average=*12,43*)
Then
Class is *Physics*
Rule's probability: **0,929**
The rule exists in **13** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
2, 10, 11, 14, 16, 18, 19, 21, 22, 23
Negative Examples (records' serial numbers):
52

9. If **Здравомыслие** is *14,00...18,00* (average=*15,62*)
and **Арифметич. задачи** *is 7,00...20,00* (average=*11,69*)
and **Воображение** *is 8,00...13,00* (average=*10,31*)
Then
Class is *Physics*
Rule's probability: **0,923**
The rule exists in **12** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 4, 10, 22, 26, 27, 29, 30, 33

Negative Examples (records' serial numbers):
47
10. If **Здравомыслие** is *14,00...18,00* (average=*15,91*)
and **Аналогии** is *12,00...18,00* (average=*15,82*)
Then
Class is *Physics*
Rule's probability: **0,909**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 22, 26, 27, 28, 29, 32, 33, 35, 36
Negative Examples (records' serial numbers):
55
11. If **Здравомыслие** is *14,00...18,00* (average=*15,60*)
and **Воображение** is *8,00...13,00* (average=*10,20*)
Then
Class is *Physics*
Rule's probability: **0,867**
The rule exists in **13** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 4, 10, 22, 26, 27, 28, 29, 30
Negative Examples (records' serial numbers):
47, 68
12. If **Здравомыслие** is *4,00...10,00* (average=*7,88*)
and **Обобщение** *is 5,00...12,00* (average=*10,69*)
Then
Class is not *Physics*
Rule's probability: **0,875**
The rule exists in **14** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
39, 43, 45, 48, 51, 53, 54, 56, 57, 58
Negative Examples (records' serial numbers):
5, 12
13. If **Здравомыслие** is *4,00...10,00* (average=*7,27*)
and **Обобщение** *is 8,00...12,00* (average=*11,09*)
and **Воображение** *is 8,00...15,00* (average=*10,55*)
Then
Class is not *Physics*
Rule's probability: **0,909**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
39, 45, 48, 51, 56, 57, 58, 59, 62, 64
Negative Examples (records' serial numbers):
5

14. *If* **Аналогии** *is 4,00...10,00* (average=*8,00*)
and **Арифметич. задачи** *is 4,00...6,00* (average=*4,82*)
and **Воображение** *is 8,00...13,00* (average=*9,82*)
Then
Class is not *Physics*
Rule's probability: **0,909**
The rule exists in **10** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
39, 41, 51, 58, 59, 65, 67, 68, 69, 70
Negative Examples (records' serial numbers):
24

15. If **Арифметич. задачи** is *7,00...20,00* (average=*11,63*)
and **Воображение** is *8,00...15,00* (average=*10,63*)
Then
Class is *Physics*
Rule's probability: **0,756**
The rule exists in **31** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Negative Examples (records' serial numbers):
45, 46, 47, 48, 56, 57, 62, 72, 74, 76

16. If **Аналогии** is *3,00...10,00* (average=*7,86*)
and **Обобщение** is *5,00...12,00* (average=*10,21*)
Then
Class is not *Physics*
Rule's probability: **0,857**
The rule exists in **12** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
39, 43, 44, 45, 48, 51, 53, 54, 57, 58
Negative Examples (records' serial numbers):
4, 31

17. If **Здравомыслие** is *14,00...18,00* (average=*15,44*)
Then
Class is *Physics*
Rule's probability: **0,778**
The rule exists in **14** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 4, 10, 22, 26, 27, 28, 29, 30
Negative Examples (records' serial numbers):
47, 55, 68, 73

18. If **Аналогии** is *3,00...10,00* (average=*8,00*)
Then

Class is not *Physics*
Rule's probability: **0,758**
The rule exists in **25** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
39, 40, 41, 43, 44, 45, 47, 48, 49, 50
Negative Examples (records' serial numbers):
4, 7, 10, 11, 15, 24, 30, 31

19. If **Здравомыслие** is *14,00...18,00* (average=*15,44*)
and **Арифметич. задачи** is *7,00...20,00* (average=*11,31*)
Then
Class is *Physics*
Rule's probability: **0,813**
The rule exists in **13** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 2, 4, 10, 22, 26, 27, 29, 30, 32
Negative Examples (records' serial numbers):
47, 55, 73

20. If **Аналогии** is *12,00...18,00* (average=*14,41*)
and **Воображение** is *8,00...14,00* (average=*10,52*)
Then
Class is *Physics*
Rule's probability: **0,759**
The rule exists in **22** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
1, 3, 6, 8, 9, 16, 17, 18, 19, 20
Negative Examples (records' serial numbers):
42, 46, 60, 71, 72, 74, 75

21. If **Обобщение** is *15,00...17,00* (average=*16,00*)
and **Воображение** *is 8,00...14,00* (average=*10,27*)
Then
Class is *Physics*
Rule's probability: **0,800**
The rule exists in **12** records.
Significance Level: Error probability <0,2
Positive Examples (records' serial numbers):
2, 10, 11, 14, 16, 18, 19, 22, 24, 26
Negative Examples (records' serial numbers):
41, 42, 65

22. If **Аналогии** is *12,00...18,00* (average=*14,17*)
Then
Class is *Physics*
Rule's probability: **0,722**
The rule exists in **26** records.

Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
1, 3, 6, 8, 9, 12, 16, 17, 18, 19
Negative Examples (records' serial numbers):
42, 46, 52, 55, 60, 63, 71, 72, 74, 75

23. If **Обобщение** is *5,00...12,00* (average=*10,52*)
Then
Class is not *Physics*
Rule's probability: **0,762**
The rule exists in **16** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
39, 43, 44, 45, 48, 51, 53, 54, 56, 57
Negative Examples (records' serial numbers):
4, 5, 12, 20, 31

24. If **Арифметич. задачи** is *7,00...20,00* (average = *11,32*)
Then
Class is *Physics*
Rule's probability: **0,700**
The rule exists in **35** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Negative Examples (records' serial numbers):
43, 44, 45, 46, 47, 48, 52, 55, 56, 57

25. If **Обобщение** is *8,00...12,00* (average=*10,93*)
and **Воображение** *is 8,00...15,00* (average=*10,71*)
Then
Class is not *Physics*
Rule's probability: **0,786**
The rule exists in **11** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
39, 45, 48, 51, 56, 57, 58, 59, 60, 62
Negative Examples (records' serial numbers):
4, 5, 20

26. If **Здравомыслие** is *4,00...10,00* (average=*8,44*)
Then
Class is not *Physics*
Rule's probability: **0,706**
The rule exists in **24** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
39, 40, 41, 42, 43, 45, 46, 48, 49, 50
Negative Examples (records' serial numbers):
5, 7, 11, 12, 13, 14, 16, 19, 34, 37

27. If **Обобщение** is *15,00...17,00* (average=*15,89*)
Then
Class is *Physics*
Rule's probability: **0,737**
The rule exists in **14** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
2, 10, 11, 14, 16, 18, 19, 21, 22, 23
Negative Examples (records' serial numbers):
40, 41, 42, 52, 65

28. If **Обобщение** is *18,00...25,00* (average=*20,00*)
and **Арифметич. задачи** *is 7,00...15,00* (average=*10,87*)
Then
Class is *Physics*
Rule's probability: **0,733**
The rule exists in **11** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
1, 9, 13, 29, 30, 32, 33, 34, 36, 37
Negative Examples (records' serial numbers):
46, 55, 72, 74

29. If **Воображение** is *4,00...7,00* (average=*5,94*)
Then
Class is not *Physics*
Rule's probability: **0,706**
The rule exists in **12** records.
Significance Level: Error probability <0,3
Positive Examples (records' serial numbers):
40, 43, 44, 49, 50, 52, 53, 54, 55, 63
Negative Examples (records' serial numbers):
12, 21, 23, 31, 32

Найденные логические закономерности демонстрируют значительную неоднородность объектов исследования — людей. Как мы видим, потребовалось 29 разнообразных высказываний, чтобы с разных сторон осветить особенности многоплановой структуры интеллекта «физиков» и «лириков». При этом остро проявились определенные слабости мыслительной деятельности у учащихся гуманитарных школ. Буквально все высказывания, относящиеся к лирикам, говорят об их средних и низких тестовых оценках, в то время как у физиков оценки всегда средние и высокие (наиболее яркие высказывания выделены в таблице). Возможно, такая картина связана с ограниченностью данного психологического эксперимента. Но в любом случае полученные результаты могут служить интересными аргументами в известном споре о физиках и лириках.

Влияние возраста и стажа работников на производительность труда

Встречаясь с объявлениями о приеме на работу, нередко можно заметить, что в этих объявлениях имеются ограничения на возраст сотрудников и пожелания относительно их стажа работы по специальности. В рассматриваемом примере нам предстоит разобраться, насколько бывают обоснованы подобные претензии работодателей.

Исходные данные заимствованы из книги «Информатика в статистике: Словарь-справочник» (М.: Финансы и статистика, 1994). Это результаты обследования 60 работников производства, у которых фиксировалась средняя часовая выработка в натуральных единицах продукции. Данные обследования отражены в табл. П.8.

Таблица П.8. Данные обследования работников производства

Стаж	Возраст		
	От 25 до 35 лет	От 35 до 45 лет	От 45 до 55 лет
От 1 до 4 лет	19 20 20 20 22	19 20 20 23 25	18 19 20 21 21
От 4 до 7 лет	30 31 32 32 34	20 29 30 31 31	19 25 25 26 26
От 7 до 10 лет	35 35 39 40 41	36 40 41 42 45	24 24 24 25 25
Свыше 10 лет	40 40 41 41 42	28 31 35 36 40	20 24 25 31 32

Наш анализ будет состоять из двух частей. В первой части мы его проведем по классической статистической схеме с использованием аппарата многофакторного дисперсионного анализа. Во второй части к анализируемым данным будет применена технология обнаружения логических закономерностей.

Дисперсионный анализ

Дисперсионный анализ применяется для обнаружения влияния выделенного (контролируемого) набора факторов на результативный признак. Факторы обычно измеряются в неколичественной шкале, а результативный признак выражается числом или вектором с числовыми компонентами.

Идея дисперсионного анализа состоит в разложении общей дисперсии результативного признака на части, обусловленные влиянием контролируемых факторов, и остаточную дисперсию, объясняемую неконтролируемым влиянием или случайными обстоятельствами. Выводы о существенности влияния контролируемых факторов на результат производится путем сравнения частей общей дисперсии при выполнении требования нормальности распределения результативного признака.

Известно много моделей дисперсионного анализа. Они классифицируются, с одной стороны, по математической природе факторов (детерминированные, случайные и смешанные) и, с другой стороны — по числу контролируемых факторов (однофакторные и многофакторные модели). Модели с более чем одним фактором дают возможность исследовать влияние на результат не только отдельных контролируемых факторов (главные влияния), но и их наложения (взаимодействия). По способу организации исходных данных выделяют полные и неполные m-факторные планы, полные и неполные блочные планы и рандомизированные (случайные) блочные планы.

Для проведения дисперсионного анализа будем использовать пакет STATGRAPHICS *Plus* for Windows, в котором реализованы все перечисленные выше модели дисперсионного анализа.

Раскроем электронную таблицу STATGRAPHICS и введем в нее значения результативного признака **output** (производительность) и закодированные значения градаций контролируемых факторов **age** (возраст) и **record** (стаж), как это показано на рис. П.5.

Выберем Compare ▸ Analysis of Variance ▸ Multifactor ANOVA. Заполним окно многофакторного дисперсионного анализа (рис. П.6).

Нажмем OK. На экране появится сводка множественного дисперсионного анализа, в которой подтверждается, что к обработке принято 60 наблюдений, для которых зафиксированы значения двух факторов. Внизу под этими сообщениями включено сообщение StatAdvisor (СтатКонсультанта) с рекомендациями по проведению дальнейшего анализа.

	output	age	record	Col_4	Col_5	Col_6	Col_7
1	19	1	1				
2	20	1	1				
3	20	1	1				
4	20	1	1				
5	22	1	1				
6	19	2	1				
7	20	2	1				
8	20	2	1				
9	23	2	1				
10	25	2	1				
11	18	3	1				
12	19	3	1				
13	20	3	1				
14	21	3	1				
15	23	3	1				
16	30	1	2				
17	31	1	2				
18	32	1	2				
19	32	1	2				
20	34	1	2				
21	20	2	2				
22	29	2	2				
23	30	2	2				
24	31	2	2				
25	31	2	2				
26	19	3	2				

Рис. П.5. Результаты обследования работников производства

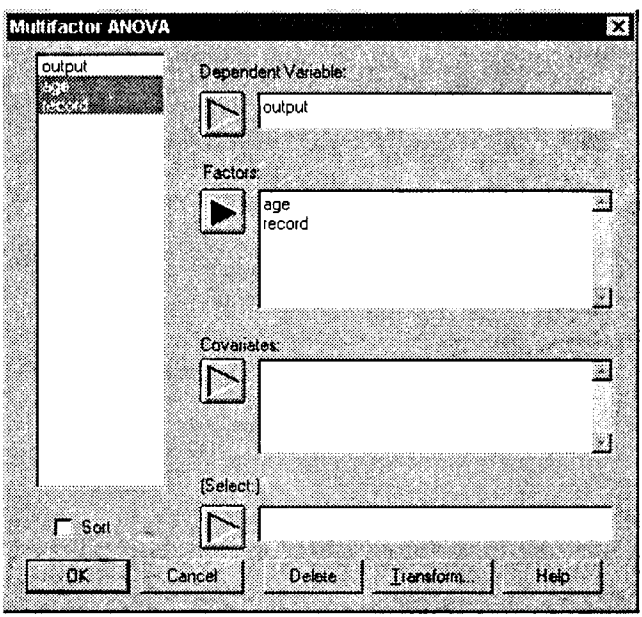


Рис. П.6. Окно диалога многофакторного дисперсионного анализа

Вызовем окно табличных опций, нажав вторую слева кнопку в нижнем ряду кнопок (рис. П.7). Установим флажок ANOVA Table (таблица дисперсионного ана-

лиза) и нажмем OK. Щелкнув дважды на окне с этой таблицей, раскроем его на все рабочее поле (рис. П.8).

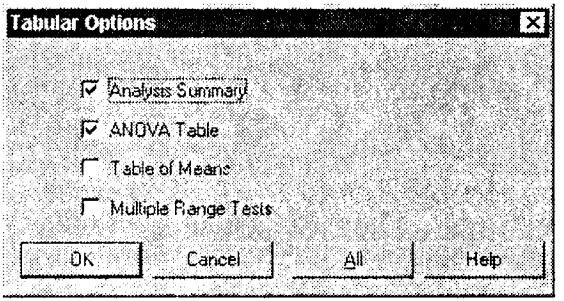


Рис. П.7. Табличные окна дисперсионного анализа

Multifactor ANOVA - output

Lbl Row

Analysis of Variance for output - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:age	900,4	2	450,2	24,64	0,0000
B:record	1842,53	3	614,178	33,89	0,0000
RESIDUAL	978,667	54	18,1235		
TOTAL (CORRECTED)	3721,6	59			

All F-ratios are based on the residual mean square error.

Рис. П.8. Исходная таблица дисперсионного анализа

На основании табличных чисел (а также по сообщению StatAdvisor) делаем заключение, что на производительность труда оказывают влияние оба фактора по отдельности — и возраст работника, и его трудовой стаж. Доверие к такому выводу — более 99 %. Можно, кроме того, оценить и совместное влияние двух факторов.

Щелкнем правой кнопкой мыши на табличном окне и выберем Analysis Options. Появится окно диалога для ввода различных взаимодействий факторов и задания их порядка (рис. П.9).

Введем порядок взаимодействия, равный 2, и нажмем OK. В таблицу многофакторного дисперсионного анализа будут добавлены оценки статистической значимости совместного влияния возраста и стажа работников на их производительность труда (рис. П.10).

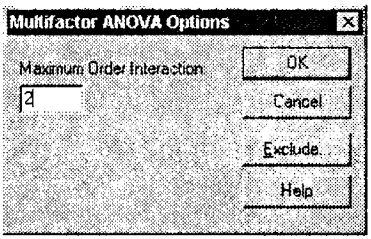


Рис. П.9. Окно диалога для задания порядка взаимодействия факторов

Multifactor ANOVA - output

Ltd Row

Analysis of Variance for output - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:age	900,4	2	450,2	48,98	0,0000
B:record	1842,53	3	614,178	66,82	0,0000
INTERACTIONS					
AB	537,467	6	89,5778	9,75	0,0000
RESIDUAL	441,2	48	9,19167		
TOTAL (CORRECTED)	3721,6	59			

All F-ratios are based on the residual mean square error.

Рис. П.10. Таблица дисперсионного анализа с оценкой значимости совокупного влияния возраста и стажа работников на производительность труда

Как следует из полученных цифр, на производительность труда изучаемой совокупности работников существенно влияют совместно действующие возраст и стаж. Уровень доверия к такому выводу выше 99 %. Можно еще более углубить проводимое исследование, воспользовавшись многосторонними оценками различных компонентов факторного взаимодействия и дополнительными статистическими тестами, реализованными в процедуре дисперсионного анализа STATGRAPHICS. Но, как говорится, лучше один раз увидеть, чем сто раз услышать. Поэтому воспользуемся графическими возможностями отображения результатов анализа.

Нажмем кнопку графических опций (третья слева в нижнем ряду кнопок) и установим флажки Means Plot (график средних) и Interactions Plot (график взаимодействий). Нажмем OK (рис. П.11).

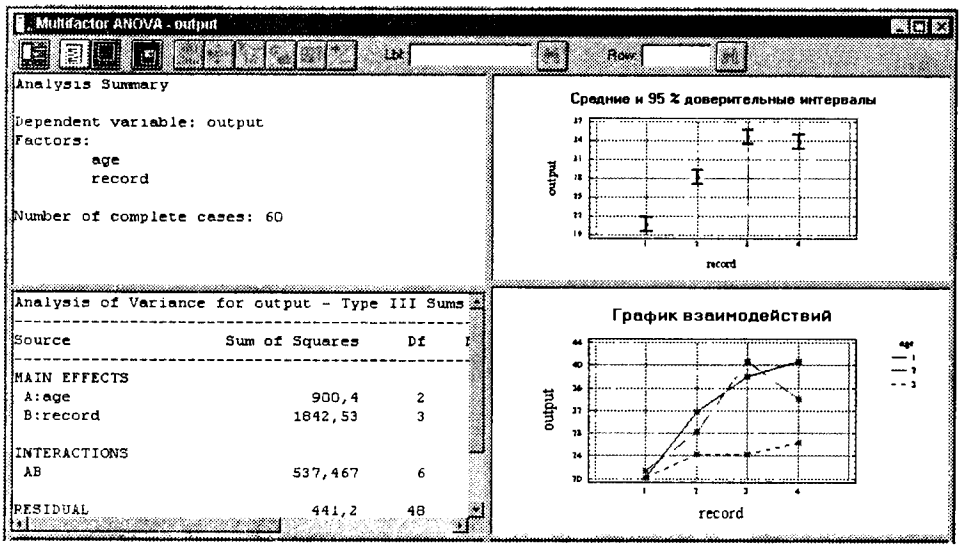


Рис. П.11. Табличные и графические отображения результатов

В верхнем графическом окне показан график зависимости средних значений производительности труда от стажа и очерчены доверительные интервалы для этих средних. Хорошо видно, что стаж несомненно влияет на результативный признак. Вместе с тем, похоже, производительность достигает своего пика у работников со стажем от 7 до 10 лет, а затем начинает снижаться.

Полученная картина проясняется, если взглянуть на нижнее графическое окно, где приведена картинка, иллюстрирующая взаимодействие возраста и стажа. Из нее следует, что производительность труда постоянно увеличивается с ростом стажа у молодых работников (25–35 лет). Для второй возрастной группы (35–40 лет) такой рост наблюдается только для тех работников, стаж которых не превышает 10 лет. Затем производительность у них резко падает. Для третьей возрастной группы (45–55 лет) характерна вообще самая низкая производительность труда, значение которой остается почти на одном и том же уровне независимо от стажа работы.

Отобразим результаты дисперсионного анализа в ином ракурсе. Для этого будем щелкать правой кнопкой на каждом графическом окне, выбирая из контекстного меню пункт Pane Options, и заменять в соответствующих окнах диалога фактор **record** (стаж) на фактор **age** (возраст). Теперь на всех графиках по оси абсцисс будут отображаться возрастные категории. Пример одного из окон диалога приведен на рис. П.12.

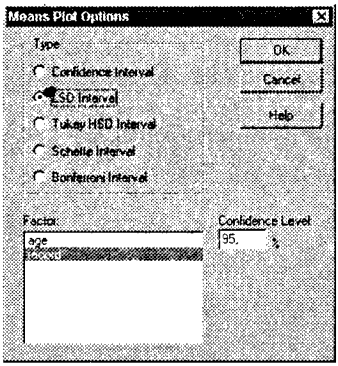


Рис. П.12. Пример окна диалога для задания параметров графических отображений результатов дисперсионного анализа

Раскроем полученные графические окна двумя щелчками левой кнопки мыши. Получим следующие картинки (рис. П.13 и П.14).

Рис. П.13. Влияние возраста работников на производительность труда

Первый график показывает уменьшение производительности труда с возрастом. Из второго следует, что максимальная производительность труда наблюдается у первой возрастной группы (25–35 лет) со стажем свыше 4–7 лет и у второй возрастной группы (35–45 лет) со стажем 7–10 лет. Также видно, что при незначительном стаже, независимо от возраста, производительность труда всегда является самой низкой. Кроме того, можно еще раз заметить, что в третьей возрастной группе (45–55 лет) производительность труда существенно снижается.

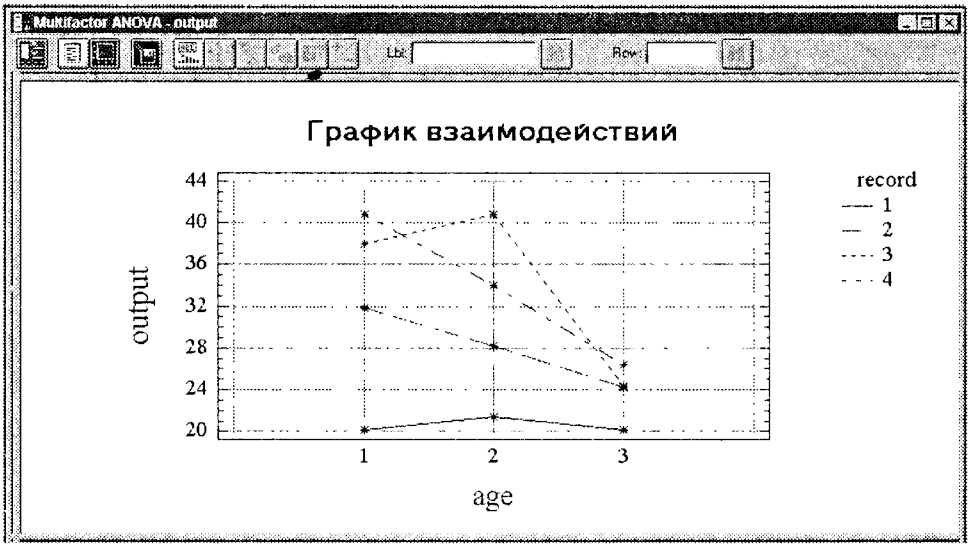


Рис. П.14. Влияние взаимодействия возраста и стажа на производительность труда

На этих выводах, как правило, анализ данных завершается. Мы вроде бы удовлетворили свое любопытство и можем теперь аргументированно со ссылкой на статистические критерии обосновать желание работодателей иметь дело с молодыми специалистами, имеющими, однако, достаточно большой стаж.

Вместе с тем наша аргументация основана на анализе усредненных характеристик и высвечивает лишь общие тенденции в рассматриваемом вопросе. А это вряд ли уместно, когда речь идет о конкретных людях, по отношению к которым требуется принимать административное решение. Как мы убедимся позже, с помощью технологии обнаружения логических закономерностей в данных можно сделать гораздо более ответственные выводы по анализируемой ситуации.

Обработка данных системой WizWhy

Система WizWhy после обработки данных о производительности труда выдала следующий отчет.

```
Total number of records: 60
Minimum probability of the:
1) if-then rules: 0,730
2) if-then-not rules: 0,630
Minimum number of cases in a rule: 8
Field to Predict: Производительность
Predicted Value (analyzed as Boolean): more than 25
Prediction error costs:
The cost of a miss: 1
The cost of a false alarm: 1
Average probability of the predicted value is 0.550
```

ANALYSIS OF THE RULES EXPLANATORY POWER
Decision point: Predict more than 25 when conclusive probability is more than **0,706**
Number of misses: 4
Number of false alarms: 1
Total number of errors: 5
Total cost of errors: 5
Success rate when predicting more than 25: 0,967
Success rate when predicting NOT more than 25: 0,.867
Number of records with no relevant rules: 0
Average cost (per record): 0,083
Expected average cost (per record): 0,450
Improvement Factor: 5,400

IF-THEN RULES:

1. If **Стаж** is *1*
Then
Производительность is not *more than 25*
Rule's probability: **1,000**
The rule exists in **15** records.
Significance Level: Error probability < 0.1
Positive Examples (records' serial numbers):
1, 2, 3, 4, 6, 7, 8, 9, 10, 11

2. If **Возраст** is *1*
and **Стаж** *is 2...4* (average=*3*)
Then
Производительность *is more than 25*
Rule's probability: **1,000**
The rule exists in **15** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
32, 34, 39, 40, 42, 43, 44, 48, 49, 51

3. If **Возраст** is *2*
and **Стаж** *is 2...4* (average=*3*)
Then
Производительность *is more than 25*
Rule's probability: **0,933**
The rule exists in **14** records.
Significance Level: Error probability <0,1
Positive Examples (records' serial numbers):
30, 31, 33, 35, 36, 37, 45, 46, 47, 50
Negative Examples (records' serial numbers):
12

4. If **Возраст** is *3*
Then
Производительность is not *more than 25*
Rule's probability: **0,800**
The rule exists in **16** records.
Significance Level: Error probability <0.2
Positive Examples (records' serial numbers):
1, 4, 5, 11, 13, 14, 17, 18, 19, 20
Negative Examples (records' serial numbers):
28, 29, 38, 41

5. If Стаж is *2...4* (average=*3*)
Then

```
Производительность is more than 25
Rule's probability: 0,733
The rule exists in 33 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
28, 29, 30, 31, 32, 33, 34, 35, 36, 37
Negative Examples (records' serial numbers):
5, 12, 13, 18, 19, 20, 21, 23, 24, 25
6. If Возраст is 1
Then
Производительность is more than 25
Rule's probability: 0,750
The rule exists in 15 records.
Significance Level: Error probability <0.3
Positive Examples (records' serial numbers):
32, 34, 39, 40, 42, 43, 44, 48, 49, 51
Negative Examples (records' serial numbers):
2, 6, 7, 8, 15
```

Из найденных 6 правил 4 описывают группу работников со средней и высокой производительностью труда, а 2 относятся к низкопроизводительным работникам. Рассмотрим эти правила раздельно по группам.

Группа с высокой производительностью

В первую очередь обращает на себя внимание правило № 2. Оно расшифровывается следующим образом: если возраст от 25 до 35 лет и стаж больше 4 лет, то высокая производительность труда. Это правило безошибочно и описывает 15 работников.

Другое правило, № 3, также имеет достаточно высокую точность и апеллирует к лицам от 35 до 45 лет, имеющим стаж также более 4 лет. Данное правило описывает 14 работников и делает всего одну ошибку.

Оставшиеся два правила обладают меньшей точностью и представляет собой высказывания отдельно по возрасту и стажу. Так, правило № 5 с точностью 0,73 говорит о том, что средняя и высокая производительность наблюдается у 33 работников, имеющих стаж более 4 лет. Правило № 6 с точностью 0,75 утверждает, что 15 высокопроизводительных работников отличает сравнительно молодой возраст (от 25 до 35 лет).

Группа с низкой производительностью

Правило № 1 здесь говорит о том, что при небольшом стаже никогда не следует ожидать хорошей производительности от работников. Это правило описывает 15 случаев низкой производительности со 100-процентной точностью.

И наконец, правило № 4 состоит в том, что 80 % людей в возрасте от 45 лет и выше не показывают удовлетворительных результатов работы в условиях рассмотренного производства.

Поиск правил для прогноза длительности ремиссий при алкоголизме

Исходным материалом для исследования служили исторические данные о 266 пациентах, проходивших лечение в отделении лечения больных алкоголизмом Психоневрологического института им. В. М. Бехтерева. Эти данные собирались на основе «Прогностической карты ремиссий при алкоголизме» (Ерышев О. Ф., Рыбакова Т. Г., Балашова Т. Н., Дубинина Л. А., 1990), которая включает более 400 признаков, отражающих анамнестические сведения о больном и его социально-психологическую характеристику, а также клинические и социально-психологические данные о динамике ремиссий.

Общая характеристика данных

Обработке подвергались данные о клинико-психологических особенностях больных алкоголизмом на начальном этапе формирования ремиссии. Это признаки x130 — x144 из прогностической карты за исключением признака x131 (данный признак не рассматривался, так как в его измерениях было довольно много пропущенных значений). Кроме того, в анализ был введен признак x125, характеризующий продолжительность спонтанных ремиссий в прошлом. Всего 15 признаков. Количество больных — 191. Все больные были разделены на две группы: 1-я группа — с продолжительностью ремиссии до 1 года (84 чел.); 2-я группа — с продолжительностью ремиссии больше 1 года (107 чел.). Ниже раскрывается содержание значений используемых признаков.

- x125 — **спонтанные ремиссии в прошлом**: 1 — нет, 2 — продолжительностью до 6 мес., 3 — до 1 года, 4 — более 2 лет;
- x130 — **влечение к алкоголю**: 1 — нет, 2 — эпизодическое, 3 — постоянное;

- x132 — **тревога**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x133 — **внутреннее напряжение**: 1 — нет, 2 — слабо-умеренно-выраженное, 3 — выраженное;
- x134 — **снижение настроения**: 1 — нет, 2 — слабо-умеренно-выраженное, 3 — выраженное;
- x135 — **дисфория**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x136 — **апатия**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x137 — **эйфория**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x138 — **дистимия**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x139 — **астенические растройства**: 1 — нет, 2 — слабо-умеренно-выраженные, 3 — выраженные;
- x140 — **неврозоподобные расстройства**: 1 — нет, 2 — слабо-умеренно-выраженные, 3 — выраженные;
- x141 — **психопатоподобные расстройства**: 1 — нет, 2 — слабо-умеренно-выраженные, 3 — выраженные;
- x142 — **психоорганические нарушения**: 1 — нет, 2 — слабо-умеренно-выраженные, 3 — выраженные;
- x143 — **критика к болезни**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная;
- x144 — **установка на трезвость**: 1 — нет, 2 — слабо-умеренно-выраженная, 3 — выраженная.

Исходные данные приведены в табл. П.9.

Таблица П.9. Исходные данные

Group	x 125	x 130	x 132	x 133	x 134	x 135	x 136	x 137	x 138	x 139	x 140	x 141	x 142	x 143	x 144
1	1	2	2	2	1	1	2	1	1	2	1	2	2	2	2
1	1	2	2	1	1	2	1	2	2	2	1	2	1	2	2
1	1	3	2	2	3	2	3	1	3	2	3	2	3	2	3
1	1	2	3	2	1	2	1	1	1	1	1	2	1	2	2
1	1	2	2	2	2	3	2	1	3	3	1	3	3	3	2
1	1	3	2	2	2	3	1	1	2	2	1	3	1	3	3
1	1	2	2	2	2	2	1	2	1	1	1	2	1	2	2
1	1	1	1	1	1	2	2	1	2	1	1	2	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
1	1	1	1	1	1	1	1	2	1	2	1	1	1	2	1
1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	2
1	1	2	2	2	1	2	1	1	1	1	1	1	1	2	2
1	2	1	2	2	1	1	1	1	1	1	2	1	1	1	2
1	1	2	1	1	1	1	1	1	1	1	1	1	1	2	2
1	1	2	2	1	1	2	2	1	1	2	2	1	1	1	1
1	1	1	2	1	1	1	1	1	1	2	1	1	1	2	2
1	1	2	1	2	1	2	1	1	2	1	1	1	1	2	1

Group	x 125	x 130	x 132	x 133	x 134	x 135	x 136	x 137	x 138	x 139	x 140	x 141	x 142	x 143	x 144
1	2	1	2	2	2	2	1	2	2	2	2	1	1	1	1
1	1	1	1	1	1	2	1	1	1	1	1	2	1	3	2
1	2	1	1	1	1	1	2	1	1	1	1	1	1	2	2
1	1	2	1	1	1	2	1	1	1	2	1	2	2	2	2
1	1	2	2	2	1	2	1	1	1	1	1	1	1	3	2
1	1	2	2	3	2	2	1	1	2	1	2	2	1	1	2
1	1	2	3	1	1	2	1	1	1	2	1	2	1	2	2
1	1	2	1	1	2	1	2	1	2	2	1	1	3	2	2
1	1	3	2	2	2	1	3	1	1	2	0	1	2	2	2
1	1	3	2	2	2	3	3	1	2	2	1	2	2	2	2
1	1	2	2	2	2	2	2	1	2	2	2	3	2	2	2
1	1	2	2	2	1	1	1	2	1	1	1	2	1	2	2
1	1	2	2	2	1	2	1	2	1	2	1	2	1	2	2
1	4	2	2	1	1	1	1	1	1	2	2	1	1	2	1
1	1	2	2	1	2	1	2	1	1	2	2	2	2	2	2
1	1	2	1	2	1	1	1	1	1	2	1	1	1	2	2
1	1	2	2	2	2	2	1	1	1	1	1	2	1	2	2
1	1	2	2	2	1	3	1	1	2	2	1	3	1	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
1	1	2	1	1	1	3	1	1	1	2	1	2	2	2	2
1	1	1	2	1	1	2	1	1	1	1	1	2	1	2	2
1	1	2	1	1	1	2	1	2	1	1	1	1	1	2	1
1	1	2	1	2	1	2	1	2	2	1	1	2	2	3	2
1	1	2	1	1	2	2	2	1	1	2	1	2	2	3	3
1	1	2	1	2	2	1	2	1	1	2	1	3	1	2	2
1	1	1	2	1	1	1	1	2	1	1	1	1	1	2	1
1	2	1	1	1	1	1	1	2	1	1	1	1	1	2	2
1	1	2	2	1	1	2	1	2	1	1	1	2	2	2	2
1	2	2	2	1	1	2	1	2	1	2	1	2	2	2	1
1	1	2	2	2	2	2	1	1	1	2	1	2	1	2	2
1	1	2	1	1	1	2	1	2	1	1	1	2	1	2	2
1	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1
1	1	2	2	2	1	1	1	1	1	2	2	2	2	2	2
1	1	2	2	1	1	1	1	1	1	1	1	1	2	2	2
1	2	1	2	2	2	1	1	1	1	1	1	1	1	2	1
1	1	1	2	2	1	2	1	1	1	1	1	2	2	2	2
1	1	2	2	1	1	1	1	1	1	1	1	2	2	2	2
1	1	2	2	2	2	2	2	1	1	2	1	2	1	2	2
1	1	2	2	2	1	1	1	2	1	1	1	1	1	2	2
1	1	2	2	2	1	2	1	2	1	2	1	1	1	2	1
1	1	2	1	2	2	2	1	1	1	1	2	3	1	2	2
1	1	2	1	1	1	1	2	1	1	2	1	1	2	2	2
1	1	2	1	2	1	2	1	2	2	1	1	3	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
1	1	2	1	1	1	1	1	2	1	1	1	1	1	2	1
1	1	2	1	1	1	1	1	2	1	1	1	1	2	2	1
1	1	2	1	2	1	2	2	1	2	2	1	2	2	3	2

(продолжение)

Таблица П.9 *(продолжение)*

Group	x 125	x 130	x 132	x 133	x 134	x 135	x 136	x 137	x 138	x 139	x 140	x 141	x 142	x 143	x 144
1	1	1	2	1	1	2	1	1	1	2	1	1	2	2	2
1	1	1	2	2	2	2	1	1	2	1	1	2	1	2	2
1	1	1	1	2	1	2	2	1	1	1	1	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
1	1	1	1	1	1	1	1	1	1	2	1	1	1	2	2
1	1	2	1	2	1	1	1	1	1	2	2	1	1	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	2	1	1	2	1	1	1	2	1	2	2	2	2
1	2	1	1	1	1	1	1	1	1	2	1	1	2	2	2
1	1	1	1	1	1	2	1	2	1	2	1	3	2	2	2
1	1	1	1	1	1	2	1	1	1	1	1	2	1	2	2
1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1
1	1	2	2	2	1	2	1	1	2	1	2	2	1	2	2
1	1	1	1	2	2	1	1	1	2	2	2	1	1	2	1
1	1	2	2	1	1	2	1	2	1	2	1	1	2	2	2
1	2	1	1	2	1	2	1	1	2	1	1	2	1	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
1	1	2	1	1	1	1	1	2	1	1	1	2	1	2	2
1	1	1	1	1	1	1	1	2	1	1	1	2	1	2	1
1	2	1	2	2	1	1	1	1	1	1	2	1	1	2	1
2	3	1	1	2	1	1	1	1	2	1	2	1	1	2	1
2	1	1	1	2	1	2	1	1	1	1	1	2	1	2	1
2	1	1	2	2	2	1	1	1	1	2	2	1	1	1	1
2	1	2	2	2	1	1	1	1	2	2	1	2	1	1	1
2	1	1	1	1	1	1	1	2	1	1	1	1	1	2	2
2	1	2	2	2	2	2	1	1	2	1	1	2	1	2	2
2	1	2	2	2	1	3	1	2	1	2	1	3	2	2	1
2	1	2	2	2	1	2	1	2	2	2	1	2	2	2	2
2	2	2	2	2	2	1	2	1	1	2	1	2	1	2	1
2	2	2	2	2	1	1	0	1	1	2	2	1	2	1	1
2	1	2	1	2	1	1	1	1	1	1	1	2	1	2	1
2	1	2	2	1	1	1	1	1	1	1	1	1	1	2	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	2	1	1	1	1	1	1	1	2	1	1	1	1
2	2	1	1	1	1	2	1	2	1	1	1	1	1	2	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	2	2	2	2	1	1	1	1	1	1	1	2	2
2	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1
2	1	3	2	2	2	2	2	1	2	2	1	2	1	2	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	2	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	1	1	1	1	1	2	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Group	x 125	x 130	x 132	x 133	x 134	x 135	x 136	x 137	x 138	x 139	x 140	x 141	x 142	x 143	x 144
2	3	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	2	1	1	1	2	1	2	1
2	2	1	1	1	2	2	1	1	1	1	1	2	3	2	2
2	2	1	1	1	1	1	1	1	1	2	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	1	2	2	2	1	1	2	2	1	2	3	2	1
2	2	1	1	2	1	2	1	1	1	1	1	2	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	1	1	1	1	1	2	2	1	1	2	2
2	1	2	2	2	1	1	1	1	1	1	1	2	1	2	2
2	1	2	2	2	1	1	1	1	1	1	1	2	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	1	2	2	2	1	1	1	2	1	1	1	1	1
2	1	1	1	2	1	2	2	1	1	1	1	1	1	2	1
2	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	1	1	1	1	1	1	1	1	2	1	2	2
2	1	2	1	1	1	2	1	2	1	1	1	2	1	2	1
2	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	2	1	1	2	1	1	1	2	1	2	1	2	2
2	1	1	2	2	1	1	1	1	2	1	2	1	1	2	1
2	1	1	1	1	1	1	1	2	2	1	1	1	2	2	1
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	2	2	2	2	1	1	2	2	2	1	1	2	1
2	1	1	1	2	2	2	1	1	2	1	1	2	1	2	2
2	1	2	2	1	1	2	1	1	1	1	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	3	2	2	2	2	3	1	2	2	1	2	1	2	2
2	4	2	2	2	2	2	2	1	2	1	1	1	2	2	1
2	1	3	3	3	2	1	1	1	2	3	3	1	2	1	1
2	3	2	2	2	2	2	1	1	1	2	2	1	1	1	1
2	1	2	2	3	1	3	1	1	2	1	1	2	2	1	1
2	1	1	1	1	1	1	1	2	1	1	1	1	1	2	1
2	1	1	2	2	1	1	1	1	1	1	1	1	1	2	1
2	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	2	2	2	1	1	2	1	1	2	1	3	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	1	1	1	2	1	1	2	1	1	2	2	2
2	1	2	1	1	1	2	1	1	1	1	1	1	1	2	2
2	1	1	2	2	1	1	1	1	1	2	1	1	1	2	1
2	1	2	2	1	1	1	1	1	1	2	1	1	1	1	1
2	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	1	1	1	1	1	1	1	1	2	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	2	2	1	3	1	2	1	1	1	2	2	2	2
2	1	2	2	2	1	1	1	1	1	2	1	1	1	1	1

(продолжение)

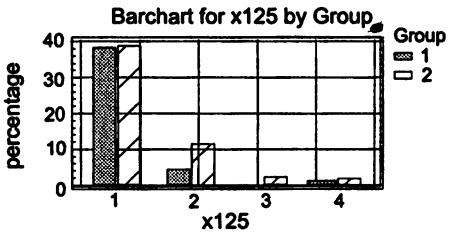
Таблица П.9 *(продолжение)*

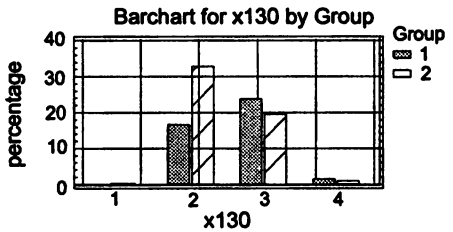
Group	x 125	x 130	x 132	x 133	x 134	x 135	x 136	x 137	x 138	x 139	x 140	x 141	x 142	x 143	x 144
2	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	2	2	1	2	1	1	2	2	1	2	1	2	2
2	3	2	1	2	1	2	1	2	1	1	1	2	1	2	1
2	4	2	2	1	2	2	1	2	1	1	1	1	1	2	1
2	2	1	2	2	1	1	1	1	1	2	2	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	4	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	2	1	1	1	2	1	1	1	1	1	1	1
2	4	1	1	1	1	2	1	2	1	1	1	1	1	2	1
2	1	2	1	1	1	1	1	1	1	2	1	2	2	1	1
2	1	1	2	2	1	2	2	1	1	2	1	2	1	2	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
2	1	2	1	1	2	1	2	1	1	2	1	1	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	2	2	2	1	1	1	1	2	2	1	2	2	2
2	1	1	1	2	1	1	2	1	1	2	1	1	2	2	12
2	1	1	1	1	2	1	2	1	2	1	1	1	2	2	2
2	1	1	1	1	1	1	1	1	2	1	1	2	2	1	2
2	1	1	1	1	1	2	1	1	1	1	1	1	1	1	2
1	1	1	2	1	2	2	1	1	2	1	1	2	2	1	2
1	2	1	2	2	1	2	1	1	2	1	1	1	2	2	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2
3	1	1	1	1	1	1	1	1	1	1	1	1	2	1	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	2	1	1	1	1	1	2	2	2
2	2	2	1	1	1	1	1	1	2	1	1	1	2	1	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	2	1	2	1	1	1	2	1	2	1	
2	1	1	1	2	1	1	1	1	1	1	1	1	1	2	1

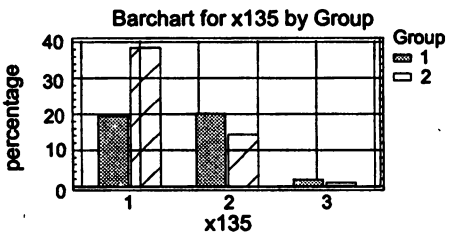
Частотный анализ признаков

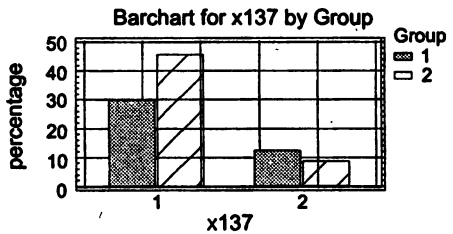
В таблице П.9 приведены графические иллюстрации результатов частотного анализа. В эту таблицу включены признаки, которые могут считаться прогностически важными по критерию хи-квадрат ($p < 0,05$).

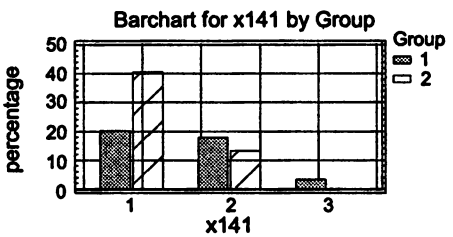
Таблица П.9. Гистограммы распределения значений информативных признаков

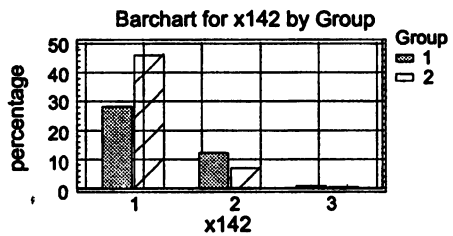


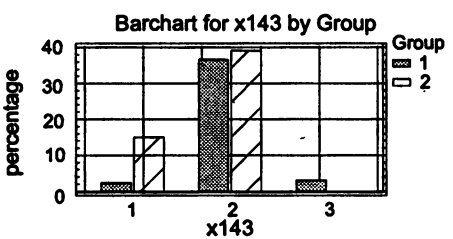


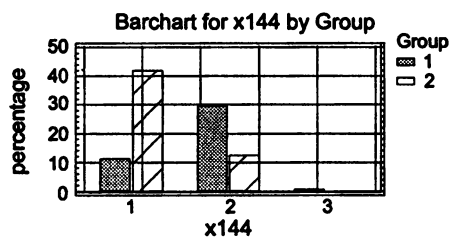












Дискриминантный анализ

Приведенные выше результаты частотного анализа дали основание полагать, что исходное пространство признаков в значительной степени способно отражать значение целевого показателя — длительности ремиссии (большое число признаков по отдельности имеет статистически значимую связь с целевым показателем). Это подтвердили последующие результаты дискриминантного анализа, который проводился по классической схеме, дополненной алгоритмом последовательного уменьшения группы признаков. Получена следующая дискриминантная функция:

$$F = -4{,}9 - 0{,}4 \times 125 + 0{,}9 \times 137 + 0{,}7 \times 140 + 0{,}6 \times 143 + 1{,}7 \times 144.$$

Дискриминантная функция обеспечивает 74,4 % правильной классификации, что иллюстрируется рис. П.15.

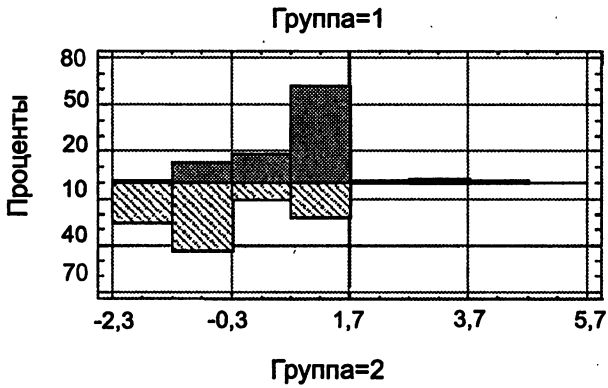


Рис. П.15. Сравнительные гистограммы распределения значений дискриминантной функции

Как видим, гистограммы распределения значений дискриминантной функции в сравниваемых классах все же достаточно сильно пересекаются. «Хорошие» решения получаются только на краях распределения.

Подобная картина вообще характерна для медико-психологических данных. Линейная модель не способна описывать сложную и неоднородную структуру классов исследуемых объектов. Она отражает только одну самую общую «грубую» тенденцию.

Результаты обработки данных системой WizWhy

Те же самые данные были подвергнуты обработке системой WizWhy с целью обнаружения логических правил IF ... THEN и дальнейшего их использования для осуществления прогнозирования продолжительности воздержания больных алкоголизмом от употребления спиртных напитков.

Всего система WizWhy обнаружила 327 логических правил. Ниже приводится выдержка из ее отчета (50 первых логических правил).

```
WIZWHY REPORT
Total number of records: 191
Minimum probability of the:
   1. if-then rules:      0,800
   2. if-then-not rules:  0,800
Minimum number of cases in a rule:   15
```

Field to Predict: **Group**
Predicted Value (analyzed as Boolean): 2
Prediction error costs:
The cost of a miss: 1
The cost of a false alarm: 1
Average probability of the predicted value is 0.560

ANALYSIS OF THE RULES EXPLANATORY POWER
Decision point: Predict 2 when conclusive probability is more than **0.830**
Number of misses: 19
Number of false alarms: 18
Total number of errors: 37
Total cost of errors: 37
Success rate when predicting 2 : 0.818
Success rate when predicting NOT 2: 0.768
Number of records with no relevant rules : 10
Average cost (per record): 0.204
Expected average cost (per record): 0.440
Improvement Factor: 2.151
IF-THEN RULES:

1. If **x125** is *1*
and **x135** is *2*
and **x138** is *1*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0.941**
The rule exists in **16** records.
Significance Level: Error probability <0.1

2. If **x137** is *1*
and **x138** is *1*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0.919**
The rule exists in **57** records.
Significance Level: Error probability <0.1

3. If **x134** is *1*
and **x135** is *1*
and **x137** is *1*
and **x140** is *1*
and **x144** is *1*

Then
Group is *2*
Rule's probability: **0,926**
The rule exists in **50** records.
Significance Level: Error probability <0.1

4. If **x125** is *1*
and **x135** is *2*
and **x138** is *1*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,880**
The rule exists in **22** records.
Significance Level: Error probability <0.1

5. If x125 is *1*
and **x138** is *1*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,852**
The rule exists in **23** records.
Significance Level: Error probability <0.1

6. If **x134** is *1*
and **x138** is *1*
and **x142** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,882**
The rule exists in **15** records.
Significance Level: Error probability <0.1

7. If **x135** is *1*
and **x137** is *1*
and **x143** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,963**
The rule exists in **26** records.
Significance Level: Error probability <0.1

8. If **x125** is *1*
and **x137** is *1*
and **x142** is *2...3* (average=*2*)

and x144 is *2*
Then
Group is not *2*
Rule's probability: **0,857**
The rule exists in **18** records.
Significance Level: Error probability <0,1

9. If **x137** is *1*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,897**
The rule exists in **61** records.
Significance Level: Error probability <0,1

10. If **x136** is *1*
and **x137** is *1*
and **x143** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,964**
The rule exists in **27** records.
Significance Level: Error probability <0,1

11. *If* **x135** is *2*
and **x138** is *1*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,889**
The rule exists in **16** records.
Significance Level: Error probability <0,1

12. If **x134** is *1*
and **x142** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,857**
The rule exists in **18** records.
Significance Level: Error probability <0,1

13. If **x125** is *1*
and **x134** is *1*
and **x135** is *2*
and **x144** is *2*
Then

Group is not *2*
Rule's probability: 0,846
The rule exists in 22 records.
Significance Level: Error probability <0,1

14. If **x137** is *1*
and **x139** is *2*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: 1,000
The rule exists in 15 records.
Significance Level: Error probability <0,1

15. If **x130** is *2*
and **x138** is *1*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: 0,850
The rule exists in 17 records.
Significance Level: Error probability <0,1

16. If **x133** is *1*
and **x137** is *1*
and **x138** is *1*
and **x139** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: 0,900
The rule exists in 36 records.
Significance Level: Error probability <0,1

17. If **x134** is *1*
and **x135** is *2*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: 0,850
The rule exists in 17 records.
Significance Level: Error probability <0,1

18. If **x137** is *1*
and **x140** is *1*
and **x143** is *1*
Then

Group is *2*
Rule's probability: **0,958**
The rule exists in **23** records.
Significance Level: Error probability <0,1

19. If **x125** is *2*
and **x134** is *1*
and **x140** is *1*
and **x142** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **17** records.
Significance Level: Error probability <0,1

20. If **x125** is *2*
and **x134** is *1*
and **x135** is *1*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

21. If **x125** is *2*
and **x133** is *1*
and **x135** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

22. If **x125** is *2*
and **x133** is *1*
and **x142** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

23. If **x133** is *1*
and **x136** is *1*
and **x137** is *1*

and x144 is *1*
Then
Group is *2*
Rule's probability: **0,891**
The rule exists in **41** records.
Significance Level: Error probability <0,1

24. If **x125** is *2*
and **x130** is *1*
and **x133** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

25. If **x137** is *1*
and **x138** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,873**
The rule exists in **62** records.
Significance Level: Error probability <0,1

26. If **x138** is *1*
and **x141** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,821**
The rule exists in **23** records.
Significance Level: Error probability <0,1

27. If **x125** is *2*
and **x134** is *1*
and **x140** is *1*
and **x141** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

28. If **x125** is *2*
and **x132** is *1*
and **x144** is *1*
Then

Group is *2*
Rule's probability: **1,000**
The rule exists in **17** records.
Significance Level: Error probability <0,1

29. If **x134** is *1*
and **x135** is *2*
and **x138** is *1*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,850**
The rule exists in **17** records.
Significance Level: Error probability <0,1

30. If **x135** is *2*
and **x138** is *1*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,815**
The rule exists in **22** records.
Significance Level: Error probability <0,1

31. If **x137** is *1*
and **x143** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,935**
The rule exists in **29** records.
Significance Level: Error probability <0,1

32. If **x125** is *1*
and **x142** is *2...3* (average=*2*)
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,821**
The rule exists in **23** records.
Significance Level: Error probability <0,1

33. If **x134** is *1*
and **x135** is *2*
and **x144** is *2*
Then
Group is not *2*
Rule's probability: **0,821**
The rule exists in **23** records.

Significance Level: Error probability <0,1

34. If x125 is *2*
and x134 is *1*
and x137 is 1
and x140 is *1*
and x144 is *1*
Then
Group is *2*
Rule's probability: 1,000
The rule exists in 16 records.
Significance Level: Error probability <0,1

35. If x133 is *1*
and x135 is *1*
and x137 is *1*
and x144 is *1*
Then
Group is *2*
Rule's probability: 0,891
The rule exists in 41 records.
Significance Level: Error probability <0,1

36. If x137 is *1*
and x142 is *2...3* (average=*2*)
and x144 is *2*
Then
Group is not *2*
Rule's probability: 0,826
The rule exists in 19 records.
Significance Level: Error probability <0,1

37. If x137 is *1*
and x144 is *1*
Then
Group is *2*
Rule's probability: 0,854
The rule exists in 70 records.
Significance Level: Error probability <0,1

38. If x134 is *1*
and x137 is *1*
and x138 is *1*
and x139 is *1*
and x144 is *1*
Then
Group is *2*
Rule's probability: 0,900
The rule exists in 45 records.
Significance Level: Error probability <0,1

39. If **x134** is *1*
and **x136** is *1*
and **x137** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: 0,875
The rule exists in 56 records.
Significance Level: Error probability <0,1

40. *If* **x132** is *1*
and **x137** is *1*
and **x138** is *1*
and **x144** is *1*
Then
Group *is 2*
Rule's probability: 0,896
The rule exists in 43 records.
Significance Level: Error probability <0,1

41. If **x125** is *1*
and **x132** is *2*
and **x136** is *1*
and **x137** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: 1,000
The rule exists in 17 records.
Significance Level: Error probability <0,1

42. *If* **x132** *is 2*
and **x134** *is 1*
and **x137** is *1*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: 1,000
The rule exists in 15 records.
Significance Level: Error probability <0,1

43. If **x142** is *2...3* (average=*2*)
and **x144** is *2*
Then
Group is not *2*
Rule's probability: 0,800
The rule exists in 24 records.
Significance Level: Error probability <0,1

44. If **x125** is *2*
and **x133** is *1*
and **x141** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **16** records.
Significance Level: Error probability <0,1

45. If **x134** is *1*
and **x135** is *1*
and **x137** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,887**
The rule exists in **55** records.
Significance Level: Error probability <0,1

46. If **x125** is *2*
and **x130** is *1*
and **x134** is *1*
and **x140** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **17** records.
Significance Level: Error probability <0,1

47. If **x135** is *1*
and **x143** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **0,931**
The rule exists in **27** records.
Significance Level: Error probability <0,1

48. If **x125** is *2*
and **x133** is *1*
and **x137** is *1*
and **x144** is *1*
Then
Group is *2*
Rule's probability: **1,000**
The rule exists in **15** records.
Significance Level: Error probability <0,1

```
49. If x130 is 1
and x134 is 1
and x137 is 1
and x144 is 1
Then
Group is 2
Rule's probability: 0,887
The rule exists in 47 records.
Significance Level: Error probability <0.1
50. If x135 is 1
and x137 is 1
and x143 is 1
Then
Group is 2
Rule's probability: 0,929
The rule exists in 26 records.
Significance Level: Error probability <0.1
```

Результаты обработки данных системой See5 (decision trees)

```
Trial 0:-
Rule 0/1: (cover 86)
                        x144 > 1
                        -> class 1 [0.705]
Rule 0/2: (cover 105)
                        x144 <= 1
                        -> class 4  [0.776]
Trial 1:-
Rule 1/1: (cover 23.5)
                        x137 <= 1
                        x142 > 1
                        x144 > 1
                        -> class 1  [0.726]
Rule 1/2: (cover 21.0)
                        x135 > 1
                        x137 <= 1
                        x138 <= 1
                        x144 > 1
                        -> class 1  [0.696]
Rule 1/3: (cover 47.1)
                        x137 > 1
                        -> class 1  [0.593]
```

```
Rule 1/4: (cover 76.3)
                    x137 <= 1
                    x144 <= 1
                    ->  class 4  [0.758]
Rule 1/5: (cover 70.1)
                    x135 <= 1
                    x137 <= 1
                    x138 <= 1
                    x142 <= 1
                    ->  class 4  [0.716]
Rule 1/6: (cover 14.0)
                    x137 <= 1
                    x138 > 1
                    x142 <= 1
                    x144 > 1
                    ->  class 4  [0.624]
Trial 2:-
Rule 2/1: (cover 27.0)
                    x140 > 1
                    ->  class 1  [0.626]
Rule 2/2: (cover 164.0)
                    x140 <= 1
                    ->  class 4  [0.576]
Trial 3:-
Rule 3/1: (cover 9.5)
                    x125 > 1
                    x132 > 1
                    x143 > 1
                    ->  class 1  [0.640]
Rule 3/2: (cover 131.0)
                    x125 <= 1
                    x143 > 1
                    ->  class 1  [0.566]
Rule 3/3: (cover 25.5)
                    x125 > 1
                    x132 <= 1
                    ->  class 4  [0.812]
Rule 3/4: (cover 30.7)
                    x143 <= 1
                    ->  class 4  [0.726]

Trial 4:-
Rule 4/1: (cover 12.3)
                    x125 <= 1
                    x133 <= 1
```

```
                    x137 > 1
                    x141 <= 1
                    x144 <= 1
                    ->  class 1  [0.688]
Rule 4/2: (cover 19.4)
                    x133 > 1
                    x138 <= 1
                    x140 <= 1
                    x144 > 1
                    ->  class 1  [0.672]
Rule 4/3: (cover 21.6)
                    x125 <= 1
                    x133 <= 1
                    x141 > 1
                    ->  class 1  [0.667]
Rule 4/4: (cover 19.1)
                    x125 <= 1
                    x130 <= 1
                    x140 <= 1
                    x144 > 1
                    ->  class 1  [0.647]
Rule 4/5: (cover 26.3)
                    x140 > 1
                    ->  class 1  [0.568]
Rule 4/6: (cover 36.2)
                    x133 <= 1
                    x137 <= 1
                    x140 <= 1
                    x141 <= 1
                    x144 <= 1
                    ->  class 4  [0.793]
Rule 4/7: (cover 30.3)
                    x125 > 1
                    x140 <= 1
                    ->  class 4  [0.733]
Rule 4/8: (cover 12.4)
                    x130 > 1
                    x133 <= 1
                    x141 <= 1
                    x144 > 1
                    ->  class 4  [0.634]
Rule 4/9: (cover 54.8)
                    x125 <= 1
                    x133 > 1
                    x140 <= 1
                    ->  class 4  [0.568]
```

```
Trial 5:-
Rule 5/1: (cover 15.3)
                    x133 <= 1
                    x136 <= 1
                    x142 > 1
                    ->  class 1  [0.701]
Rule 5/2: (cover 51.1)
                    x141 > 1
                    x144 > 1
                    ->  class 1  [0.663]
Rule 5/3: (cover 28.4)
                    x136 > 1
                    ->  class 1  [0.569]
Rule 5/4: (cover 91.1)
                    x136 <= 1
                    x144 <= 1
                    ->  class 4  [0.675]
Rule 5/5: (cover 121.7)
                    x141 <= 1
                    ->  class 4  [0.603]

Trial 6:-
Rule 6/1: (cover 12.4)
                    x134 <= 1
                    x138 > 1
                    x141 > 1
                    ->  class 1  [0.642]
Rule 6/2: (cover 12.4)
                    x134 <= 1
                    x138 <= 1
                    x141 > 1
                    x142 > 1
                    ->  class 1  [0.640]
Rule 6/3: (cover 31.9)
                    x130 > 1
                    x134 <= 1
                    x141 <= 1
                    ->  class 1  [0.629]
Rule 6/4: (cover 22.1)
                    x134 <= 1
                    x138 <= 1
                    x141 > 1
                    x142 <= 1
                    ->  class 4  [0.645]
Rule 6/5: (cover 70.7)
                    x130 <= 1
```

```
                    x134 <= 1
                    x141 <= 1
                    ->  class 4  [0.615]
Rule 6/6: (cover 41.5)
                    x134 > 1
                    ->  class 4  [0.547]
Trial 7:-
Rule 7/1: (cover 18.8)
                    x130 <= 1
                    x132 <= 1
                    x134 <= 1
                    x144 > 1
                    ->  class 1  [0.719]
Rule 7/2: (cover 11.7)
                    x125 <= 1
                    x132 > 1
                    x136 > 1
                    ->  class 1  [0.690]
Rule 7/3: (cover 23.7)
                    x130 > 1
                    x132 <= 1
                    x134 <= 1
                    ->  class 1  [0.645]
Rule 7/4: (cover 15.1)
                    x125 > 1
                    x132 > 1
                    ->  class 1  [0.577]
Rule 7/5: (cover 49.4)
                    x130 <= 1
                    x132 <= 1
                    x134 <= 1
                    x144 <= 1
                    ->  class 4  [0.669]
Rule 7/6: (cover 60.4)
                    x125 <= 1
                    x132 > 1
                    x136 <= 1
                    ->  class 4  [0.582]
Rule 7/7: (cover 11.9)
                    x132 <= 1
                    x134 > 1
                    ->  class 4  [0.537]
```

Отчет системы See5

```
Evaluation on training data (191 cases):

Trial                Decision Tree          Rules
--                   -------------    -------------
                     Size    Errors     No    Errors

   0                  2      48(25.1%   2     48(25.1%)
   1                  6      52(27.2%)  6     52(27.2%)
   2                  9      57(29.8%)  2     81(42.4%)
   3                  4      70(36.6%)  4     70(36.6%)
   4                 10      51(26.7%)  9     51(26.7%)
   5                  9      53(27.7%)  5     52(27.2%)
   6                  8      60(31.4%)  6     69(36.1%)
   7                  7      63(33.0%)  7     63(33.0%)
boost                          40(20.9%)        40(20.9%)   <<


          (a)   (b)      <-classified as
          --    --
           57    27      (a): class 1
           13    94      (b): class 2
```

Таблица П.10. Сводная таблица результатов обработки клинико-психологических данных различными методами

Метод	Ошибка прогноза для 1-й группы	Ошибка прогноза для 2-й группы	Отказ от прогноза	Количество правил
See5	32,1 %	12,1 %	—	39
WizWhy	23,2 %	19,2 %	5,2 %	327

Виды знаний и способы их представления

Существуют различные определения понятия «знания». В одном из наиболее емких определений [3] под знаниями понимают формализованную информацию, на которую ссылаются или которую используют в процессе решения задачи. Знание о предметной области включает описание объектов, их окружения, необходимых явлений, фактов, а также отношений между ними [9]. Общение с ЭВМ на уровне знания предопределяет возможность ввода в машину и использование ею некоторой совокупности взаимосвязанной информации. Сложность понятия «знание» заключена в множественности его носителя и неразрывности с понятием «данные». Выделяют несколько уровней формализации знания о предметной области: знания в памяти человека; знания в форме языковой модели предметной области, используемые человеком и зафиксированные на физических носителях с использованием контекстно-зависимых языков, графических образов и т. п.; знания, формализованные для их представления при использовании в ЭВМ; фактографические сведения или данные.

Виды знаний

Фактические и стратегические знания

В [4] знания определяются как «...основные закономерности предметной области, позволяющие человеку решать конкретные производственные, научные и другие задачи, то есть факты, понятия, взаимосвязи, оценки, правила, эвристики (иначе *фактические знания*), а также стратегии принятия решения в этой области (иначе *стратегические знания*)».

Факты и эвристики

Некоторые авторы разделяют знания на две большие категории: факты и эвристики. Первая категория (факты) указывает на хорошо известные в той или иной предметной области обстоятельства. Такие знания еще называют текстовыми, имея

в виду достаточную их освещенность в специальной литературе и учебниках. Вторая категория знаний (эвристики) основывается на индивидуальном опыте специалиста (эксперта) в предметной области, накопленном в результате многолетней практики. Эта категория нередко играет решающую роль при построении интеллектуальных программ. Сюда относятся такие знания, как «способы удаления бесполезных гипотез», «способы использования нечеткой информации», «способы разрешения противоречий» и т. п.

Декларативные и процедурные знания

Под декларативными знаниями подразумевают знания типа «А это В», и они характерны для баз данных. Это, например, такие факты, как «в час пик на улице много машин», «зажженная плита — горячая», «скарлатина — инфекционное заболевание»...

К процедурным знаниям относятся сведения о способах оперирования или преобразования декларативных знаний.

Интенсиональные и экстенсиональные знания

Интенсиональные знания — это знания о связях между атрибутами (признаками) объектов данной предметной области. Они оперируют абстрактными объектами, событиями и отношениями.

Экстенсиональные знания представляют собой данные, характеризующие конкретные объекты, их состояния, значения параметров в пространстве и времени.

Глубинные и поверхностные знания

В *глубинных знаниях* отражается понимание структуры предметной области, назначение и взаимосвязь отдельных понятий (глубинные знания в фундаментальных науках — это законы и теоретические основания). *Поверхностные знания* обычно касаются внешних эмпирических ассоциаций с каким-либо феноменом предметной области.

Например, для разговора по телефону требуется лишь поверхностное знание того, что, сняв трубку и правильно набрав номер, мы соединимся с нужным абонентом. Большинство людей не испытывает необходимости в глубинных представлениях о структуре телефонной связи, конструкции телефонного аппарата, которыми, безусловно, пользуются специалисты по телефонии.

В [4] отмечается, что большинство экспертных систем основано на применении поверхностных знаний. Это, однако, нередко не мешает достигать вполне удовлетворительных результатов. Вместе с тем, опора на глубинные представления помогает создавать более мощные, гибкие и интеллектуальные адаптивные системы. Наглядным примером может служить медицина. Здесь молодой и недостаточно опытный врач часто действует по поверхностной модели: «Если кашель — то пить таблетки от кашля, если ангина — то эритромицин» и т. п. В то же время

опытный врач, основываясь на глубинных знаниях, способен порождать разнообразные способы лечения одной и той же болезни в зависимости от индивидуальных особенностей пациента, его состояния, наличия доступных лекарств в аптечной сети и т. д.

Глубинные знания являются результатом обобщения первичных понятий предметной области в некоторые более абстрактные структуры. Степень глубины и уровень обобщенности знаний непосредственно связаны с опытом экспертов и могут служить показателем их профессионального мастерства.

Жесткие и мягкие знания

Жесткие знания позволяют получать однозначные четкие рекомендации при заданных начальных условиях. *Мягкие знания* допускают множественные, «размытые» решения и различные варианты рекомендаций (рис. П.16).

Характеристика различных предметных областей по глубине и жесткости дает возможность проследить тенденцию развития интеллектуальных систем [6].

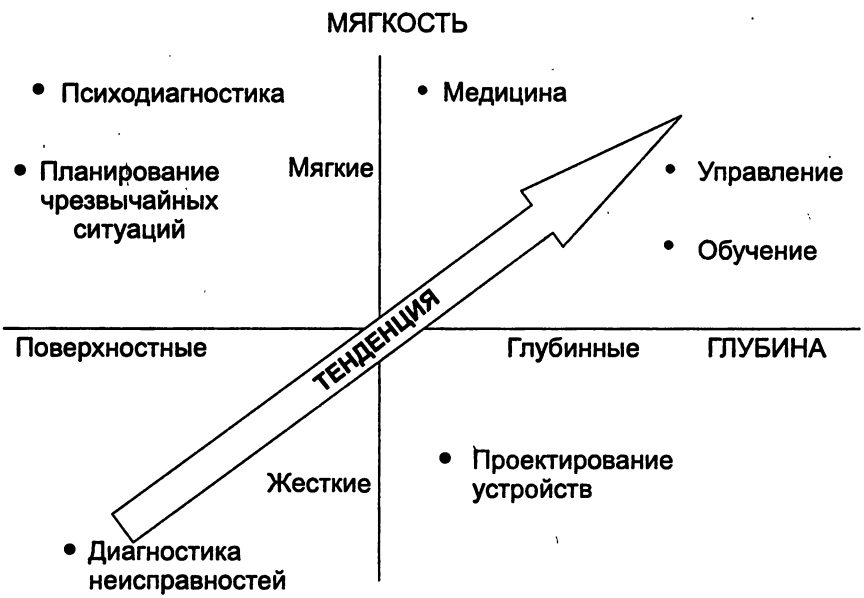


Рис. П.16. Тенденция развития интеллектуальных систем

Как видно из рисунка, область практического применения интеллектуальных систем все более смещается в сферу задач с преобладанием глубинных и мягких знаний. Такие задачи еще называют трудно формализуемыми. Для них характерна одна или несколько следующих особенностей:

- задача не может быть определена в числовой форме (требуется символьное представление);

- алгоритмическое решение задачи не известно (хотя, возможно, и существует) или не может быть использовано из-за ограниченных ресурсов (памяти компьютера, быстродействия);
- цели задачи не могут быть выражены в терминах точно определенной целевой функции или не существует точной математической модели задачи.

Системы, основанные на знаниях, не отвергают и не заменяют традиционных подходов к решению формализованных задач. Они отличаются тем, что ориентированы на решение трудно формализуемых задач. Интеллектуальные системы особенно важны там, где наука не может создать конструктивных определений, область определений меняется, ситуации зависят от контекстов и языковая (описательная) модель доминирует над алгоритмической.

Модели представления знаний

Наиболее распространенными моделями представления знаний являются:

- продукционные системы;
- логические модели;
- фреймы;
- семантические сети.

Продукционные системы

В *продукционных системах* знания представляются в виде совокупности специальных информационных единиц, имеющих следующую структуру [2]:

```
Имя продукции: Сфера
Предусловие
Условие
Если А, то В
Постусловие
Пример продукции:
47: Интерпретация результатов психологического тестирования
Использовать в первую очередь
Шкала «лжи» L<70 Т-баллов
Если (шкала ошибок F – шкала коррекции К) <-11, то вывести сообщение: «Результаты
тестирования недостоверны»
Закончить интерпретацию результатов
```

Из приведенного примера видно, как устроена одна продукция. При большом количестве продукций (их еще называют продукционными правилами) сфера позволяет анализировать только правила, относящиеся к делу, не обращая внимания на большинство правил из иных сфер. Предусловия устанавливают на множестве правил из интересующей сферы некоторый порядок, приоритет их использования. Условия определяют возможность применения того или иного

правила. Ядро продукции «Если А, то В» описывает преобразование, которое составляет суть продукционного правила. Наконец, постусловие говорит о том, что надо делать, когда данное продукционное правило сработало.

В общем случае продукционная система включает следующие компоненты:

- базу данных, содержащую множество фактов;
- базу правил, содержащую набор продукций;
- интерпретатор (механизм логического вывода) или правила работы с продукциями.

База правил и база данных образуют базу знаний. Факты в базе данных представляют собой краткосрочную информацию и в принципе могут изменяться в ходе работы продукционной системы по мере накопления опыта. Правила являются более долговременной информацией и предназначены для порождения гипотез (новых фактов) из того, что уже известно.

Продукционные системы делят на два типа — с прямыми и обратными выводами. При прямом выводе рассуждение ведется от данных к гипотезам, а при обратном производится поиск доказательства или опровержения некоторой гипотезы. Часто используются комбинации прямой и обратной цепи рассуждений.

Продукции по сравнению с другими формами представления знаний имеют следующие преимущества [15]:

- модульность;
- единообразие структуры (основные компоненты продукционной системы могут применяться для построения интеллектуальных систем с различной проблемной ориентацией);
- естественность (вывод заключения в продукционной системе во многом аналогичен процессу рассуждений эксперта);
- гибкость родовидовой иерархии понятий, которая поддерживается только как связи между правилами (изменение правила влечет за собой изменение в иерархии).

Однако продукционные системы не свободны от недостатков:

- процесс вывода менее эффективен, чем в других системах, поскольку большая часть времени при выводе затрачивается на непроизводительную проверку применимости правил;
- этот процесс трудно поддается управлению;
- сложно представить родовидовую иерархию понятий.

Представление знаний с помощью продукций иногда называют «плоским», так как в продукционных системах отсутствуют средства для установления иерархии правил. Объем базы знаний продукционных систем растет линейно, по мере включения в нее новых фрагментов знаний, в то время как в традиционных алгоритмических системах, использующих деревья решений, зависимость между объемом базы знаний и количеством собственно знаний является логарифмической.

Логические модели

Логические модели представления знаний реализуются средствами логики предикатов.

Предикатом называется функция, принимающая только два значения — истина и ложь — и предназначенная для выражения свойств объектов или связей между ними. Выражение, в котором утверждается или отрицается наличие каких-либо свойств у объекта, называется *высказыванием. Константы* служат для именования объектов предметной области. Логические предложения или высказывания образуют *атомарные формулы. Интерпретация предиката* — это множество всех допустимых связываний переменных с константами. Связывание представляет собой подстановку констант вместо переменных. Предикат считается общезначимым, если он истинен на всех возможных интерпретациях. Говорят, что высказывание логически следует из заданных посылок, если оно истинно всегда, когда истинны посылки.

Наиболее простым языком логики является *исчисление высказываний*, в котором отсутствуют переменные. Любому высказыванию можно приписать значение *истинно* или *ложно*. Отдельные высказывания могут соединяться связками И, ИЛИ, НЕ, которые называются булевыми операторами. Основу исчисления высказываний составляют правила образования сложных высказываний из атомарных. В качестве примеров сложных (составных) высказываний можно привести следующие:

```
A - ИСТИННО и B - ЛОЖНО.
A и B ЛОЖНО.
A или B ИСТИННО.
```

Здесь переменные обозначают логические высказывания, о которых можно сказать, что они истинны или ложны. Логические операторы имеются в большинстве языков программирования. Однако исчисление высказываний — недостаточно выразительное средство для обработки знаний, поскольку в нем не могут быть представлены предложения, включающие переменные с кванторами.

Исчисление предикатов с *кванторами* (логика предикатов) является расширением исчисления высказываний, в котором для выражения отношений между объектами предметной области могут использоваться предложения, включающие не только константы, но и переменные.

В общем случае модели, основанные на логике предикатов, описываются формальной системой, которая задается четверкой:

M = (T, P, A, П),

где T — множество базовых элементов или алфавит формальной системы;

P — множество синтаксических правил, с помощью которых можно строить синтаксически корректные предложения;

A — множество аксиом или некоторых синтаксически правильных предложений, заданных априорно;

П — правила продукций (правила вывода или семантические правила), с помощью которых можно расширять множество А, добавляя в него синтаксически правильные предложения.

Главное преимущество логических моделей представления знаний заключается в возможности непосредственно запрограммировать механизм вывода синтаксически правильных высказываний. Примером такого механизма служит, в частности, процедура вывода, построенная на основе метода резолюций [8]. Однако с помощью правил, задающих синтаксис языка, нельзя установить истинность или ложность того или иного высказывания. Причем это распространяется абсолютно на все языки. Высказывание может быть построено синтаксически правильно, но оказаться совершенно бессмысленным.

Логические модели представления и манипулирования знаниями были особенно популярны в 70-х годах. Тогда казалось, что с появлением языков программирования типа ПРОЛОГ процедуры логического вывода в исчислении предикатов будут достаточны для решения всех типов задач в интеллектуальных системах. Вместе с тем, по мере того как в поле зрения исследователей включались все новые интеллектуальные задачи, стало ясно, что говорить о доказательном выводе можно только в небольшом числе случаев, когда проблемная область, в которой решается задача, формально описана и полностью известна. Но большинство задач, где интеллект человека позволяет находить нужные решения, связано с областями, где знания принципиально неполны, неточны, некорректны и характеризуются еще немалым числом характеристик, начинающихся с частицы «не» [2].

При таких условиях речь может идти только о правдоподобном выводе, при котором окончательный результат получается лишь с некоторой оценкой уверенности в его истинности. Кроме того, специалисты, работающие в плохо формализованных областях (например, в медицине), рассуждают совсем не так, как представители точных наук. Для них весомым аргументом в пользу принятия какого-либо положения может быть мнение ряда признанных в этих областях авторитетов или, например, сходство доказываемого положения с другим, для которого решение уже известно. Поэтому дальнейшее развитие баз знаний пошло по пути работ в области индуктивных логик, логик «здравого смысла», логик веры и других логических систем, имеющих мало общего с классической математической логикой.

Фреймы

Фрейм чаще всего определяют как структуру данных для представления стереотипных ситуаций. Модель представления знаний на основе фреймов использует концепцию организации памяти, понимания и обучения человека, предложенную М. Минским (1979). Фрейм (дословно — «рамка») — это единица представления знаний, детали которой могут изменяться в соответствии с текущей ситуацией. Фрейм в любой момент может быть дополнен различной информацией, касающейся способов применения данного фрейма, последствий этого применения и т. п.

Структура фрейма состоит из характеристик описываемой стереотипной ситуации и их значений, которые называются, соответственно, *слотами* и *заполнителями слотов.*

```
Имя фрейма:     Имя первого слота, значение первого слота
                Имя второго слота, значение второго слота
                .....
                .....
                .....
                Имя К-го слота, значение К-го слота
```

Незаполненный фрейм называется протофреймом, а заполненный — экзофреймом. Роль протофрейма как оболочки в экзофрейме весьма важна. Эта оболочка позволяет осуществлять процедуру внутренней интерпретации, благодаря которой данные в памяти системы не безлики, а имеют вполне определенный, известный системе смысл.

Слот может содержать не только конкретное значение, но и имя процедуры, позволяющей вычислить его по заданному алгоритму, а также одну или несколько продукций (эвристик), с помощью которых это значение определяется. В слот может входить не одно, а несколько значений. Иногда этот слот включает компонент, называемый *фасетом*, который задает диапазон или перечень его возможных значений. Фасет указывает также граничные значения заполнителя слота.

Как уже отмечалось, помимо конкретного значения в слоте могут храниться процедуры и правила, которые вызываются при необходимости вычисления этого значения. Среди них выделяют *процедуры-демоны* и *процедуры-слуги.* Первые запускаются автоматически при выполнении некоторого условия, а вторые активизируются только по специальному запросу. Если, например, фрейм, описывающий человека, включает слоты ДАТА РОЖДЕНИЯ и ВОЗРАСТ и в первом из них находится некоторое значение, то во втором слоте может стоять имя процедуры-демона, вычисляющей возраст по дате рождения и текущей дате и активизирующейся при каждом изменении текущей даты.

Совокупность фреймов, моделирующая какую-либо предметную область, представляет собой иерархическую структуру, в которую фреймы собираются с помощью родовидовых связей. На верхнем уровне иерархии находится фрейм, содержащий наиболее общую информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать значения характеристик своих родителей, находящихся на более высоком уровне иерархии. Эти значения могут передаваться по умолчанию фреймам, находящимся ниже них в иерархии, но если последние содержат собственные значения данных характеристик, то в качестве истинных принимаются именно они. Это обстоятельство позволяет без затруднений учитывать во фреймовых системах различного рода исключения.

Различают статические и динамические системы фреймов. В системах первого типа фреймы не могут быть изменены в процессе решения задачи, а в системах второго типа это допустимо.

О системах программирования, основанных на фреймах, говорят, что они являются объектно-ориентированными. Каждый фрейм соответствует некоторому объекту предметной области, а слоты содержат описывающие этот объект дан-

ные, то есть в слотах находятся значения признаков объектов. Фрейм может быть представлен в виде списка свойств, а если использовать средства базы данных, то в виде записи.

Наиболее ярко достоинства фреймовых систем представления знаний проявляются в том случае, если родовидовые связи изменяются нечасто и предметная область насчитывает немного исключений. Во фреймовых системах данные о родовидовых связях хранятся явно, как и знания других типов. Значения слотов представляются в системе в единственном экземпляре, поскольку включаются только в один фрейм, описывающий наиболее общие понятия из всех тех, которые содержит слот с данным именем. Такое свойство систем фреймов обеспечивает экономное размещение базы знаний в памяти компьютера. Еще одно достоинство фреймов состоит в том, что значение любого слота может быть вычислено с помощью соответствующих процедур или найдено эвристическими методами. То есть фреймы позволяют манипулировать как декларативными, так и процедурными знаниями.

К недостаткам фреймовых систем относят их относительно высокую сложность, что проявляется в снижении скорости работы механизма вывода и увеличении трудоемкости внесения изменений в родовидовую иерархию. Поэтому большое внимание при разработке фреймовых систем уделяют наглядным способам отображения и эффективным средствам редактирования фреймовых структур.

Семантические сети

Семантическая сеть описывает знания в виде сетевых структур. В качестве вершин сети выступают понятия, факты, объекты, события и т. п., а в качестве дуг сети — отношения, которыми вершины связаны между собой. Так, семантическая сеть, представляющая знания об автомобиле гр. Васильева, показана на рис. П.17.

Семантические сети часто рассматривают как общий формализм для представления знаний. Частным случаем таких сетей являются сценарии, в которых в качестве отношений выступают каузальные отношения или отношения типа «цель — средство».

Вершины сети соединяются дугой, если соответствующие объекты предметной области находятся в каком-либо отношении. Самыми распространенными являются следующие типы отношений:

БЫТЬ ЭЛЕМЕНТОМ КЛАССА (ЯВЛЯТЬСЯ) — означает, что объект входит в состав данного класса, например: ВАЗ 2106 является автомобилем;

ИМЕТЬ — позволяет задавать свойства объектов, например: жираф имеет длинную шею;

ЯВЛЯТЬСЯ СЛЕДСТВИЕМ — отражает причинно-следственные связи, например: астеническое состояние является следствием перенесенного простудного заболевания;

ИМЕТЬ ЗНАЧЕНИЕ — задает значение свойств объектов, например: пациент может иметь двух братьев.

Как и в системе, основанной на фреймах, в семантической сети могут быть представлены родовидовые отношения, которые позволяют реализовывать наследование свойств от объектов-родителей. Это обстоятельство приводит к тому, что семантические сети приобретают все недостатки и достоинства представления знаний в виде фреймов. Преимущества заключаются в простоте и наглядности описания предметной области. Однако последнее свойство с усложнением семантической сети теряется и, кроме того, существенно увеличивается время вывода. Также к недостаткам семантических сетей относят сложность обработки различного рода исключений.

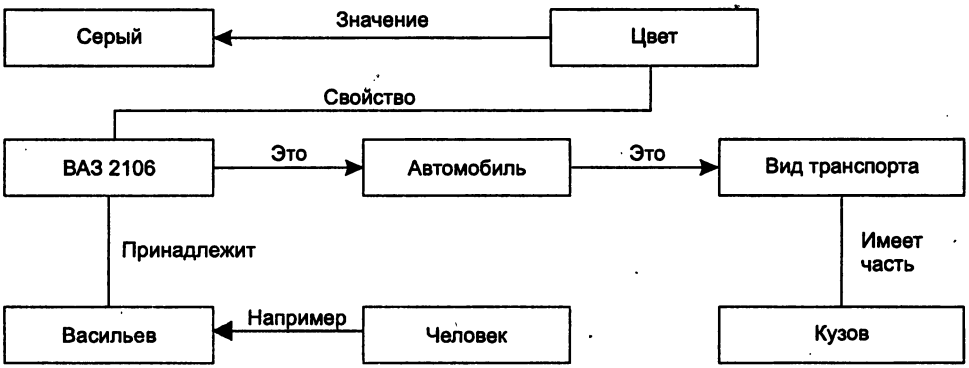


Рис. П.17. Пример семантической сети

Другие методы представления знаний

Из других методов представления знаний популярностью пользуется *представление знаний по примерам* [15]. Работая с системой такого типа, пользователь задает ей несколько примеров решения задач из актуальной предметной области. На основе этих примеров система самостоятельно строит базу знаний, которая затем применяется для решения других задач. При создании базы знаний пользователь имеет возможность в любой момент вызвать на экран дисплея матрицу, состоящую из примеров задач и их решений, с тем чтобы установить в ней наличие пустых мест, которые необходимо заполнить недостающими примерами «задача—решение».

Знания в такой системе могут храниться в различной форме. Это может быть, например, интенсиональная форма, когда пользователь вводит в систему правила операций с атрибутами объектов предметной области, приводящие к требуемому решению. Также это может быть экстенсиональная форма, при которой каждый пример детально описывается пользователем и представляется в памяти компьютера в виде совокупности значений выделенных атрибутов. Возможно сочетание и той, и другой форм. В результате получается матрица примеров, которая может быть расширена или изменена лишь путем корректировки примеров, содержащихся в матрице, или их добавлением.

Основным достоинством представления знаний по примерам является простота данного способа, поскольку пользователь может не иметь ни малейшего представления о продукционных правилах, исчислении предикатов, фреймах и семантических сетях. Вместе с тем, в качестве недостатков метода представления знаний по примерам отмечают отсутствие гибкости процесса построения интеллектуальной системы. Пользователь оказывается отстраненным от собственно создания базы знаний и поэтому не может контролировать связи между содержащимися в ней понятиями.

Выбор способа представления знаний осуществляется инженером по знаниям после того, как им достигнуто понимание природы данных моделируемой области. При решении сложных задач возможны ситуации, когда источники знаний различаются по типам, и, соответственно, представление таких знаний требует использования разных способов (смешанное представление). Тогда для продуктивного функционирования интеллектуальной системы нередко применяют *принцип доски объявлений*, с помощью которого реализуется взаимодействие различных независимых источников знаний.

Литература

1. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности. — М.: Финансы и статистика, 1989.
2. Будущее искусственного интеллекта. — М.: Наука, 1991.
3. Вассерман Л. И., Дюк В. А., Иовлев Б. В., Червинская К. Р. Психологическая диагностика и новые информационные технологии. — СПб.: СЛП, 1997.
4. Гаврилова Т. А., Червинская К. Р. Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992.
5. Десять лет спустя (интервью с Д. Мичи)//Будущее искусственного интеллекта. — М.: Наука, 1991. — С. 213–215.
6. Коов М. И., Мацкин М. Б., Тыугу Э. Х. Интеграция концептуальных и экспертных знаний в САПР//Изв. АН СССР. Техн. кибернетика. — 1988. — № 5. — С. 108–118.
7. Минский М. Фреймы для представления знаний. — М.: Мир, 1979. Попов Э. В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ. — М.: Наука.
8. Назаретов В. М., Ким Д. П. Техническая имитация интеллекта. В 9 кн: Кн. 6. Робото-техника и гибкие автоматизированные производства./Под ред. И. М. Макарова. — М.: Высш. шк. 1986.
9. Поляков А. О. Технология интеллектуальных систем: Учеб. пособие. — СПб.: СПбГТУ, 1995.
10. Попов Э. В. Особенности разработки и использования экспертных систем// Искусственный интеллект. Кн. 1. Системы общения и экспертные системы. — М.: Радио и связь, 1990.

11. Поспелов Д. А. Данные и знания. Представление знаний//Искусственный интеллект. Кн. 2. Модели и методы: Справочник/Под ред. Д. А. Поспелова. — М.: Радио и связь, 1990. — С. 7–13.

12. Поспелов Д. А. Моделирование рассуждений. Опыт анализа мыслительных актов. — М.: Радио и связь, 1989.

13. Поспелов Д. А. Ситуационное управление. — М.: Наука, 1986.

14. Представление и использование знаний/Под ред. К. Уэно, М. Исидзука. — М.: Мир, 1989.

15. Таунсенд К., Фохт Д. Проектирование и программная реализация экспертных систем на персональных ЭВМ. — М.: Финансы и статистика, 1990.

16. Франселла Ф., Баннистер Д. Новый метод исследования личности. — М.: Прогресс, 1987.

17. Davis R. TEIRESIAS: Applications of mete-level knowledge//Knowledge-based Systems in Artifical Intelligence). — N. Y.: McGraw-Hill, 1982.

18. Osgood Ch. E., Susi G. E., Tannenbaum P. N. The Measurement of Meaning. — Urbana: Univ. I 11. Press, 1957.

19. Shortliffe E. Computer based medical consultations: MYCIN .— N.-Y.: American Elsevier, 1976.

20. Waterman D. A. Guide to expert Systems. — N.-Y.: Addison-Welse, 1986.

Системы, основанные на знаниях, и особенности их разработки

Области применения и решаемые задачи

Области применения систем, основанных на знаниях, весьма разнообразны: бизнес, производство, военные приложения, медицина, социология, геология, космос, сельское хозяйство, управление, юриспруденция и т. д.

Типы решаемых задач:

- интерпретация символов или сигналов — составление смыслового описания по входным данным;
- диагностика — определение неисправностей (заболеваний) по симптомам;
- мониторинг — наблюдение за изменяющимся состоянием объекта и сравнение его показателей с установленными или желаемыми;
- проектирование — разработка объекта с заданными свойствами при соблюдении установленных ограничений;
- прогнозирование — определение последствий, наблюдаемых ситуаций;
- планирование — определение последовательности действий, приводящих к желаемому состоянию объекта;
- управление — воздействие на объект для достижения желаемого поведения;
- обучение — объяснение или консультации в той или иной области знаний.

Типы систем, основанных на знаниях

Интеллектуальные информационно-поисковые системы (ИИПС)

Эти системы отличаются от предыдущего поколения информационно-поисковых систем не только гораздо более обширным справочно-информационным фондом,

но и важнейшей способностью формировать адекватные ответы на запросы пользователя даже тогда, когда запросы не носят прямого характера. Иными словами, ИИПС достаточно «умны» для того, чтобы понять недостаточно четко сформулированные вопросы. Другой особенностью ИИПС является их способность «переваривать» огромные количества информации из разнообразных источников, осуществляя ее автоматическое реферирование и проводя анализ состояний противоречивости и неполноты тех или иных фрагментов знания.

Экспертные системы

Наиболее известным практическим примером достижений в области искусственного интеллекта могут служить экспертные системы, которые представляют собой интеллектуальные программы, способные делать логические выводы на основании знаний в конкретной предметной области и обеспечивающие решение специфических задач на профессиональном уровне. Общая структура экспертной системы приведена на рис. П.18.

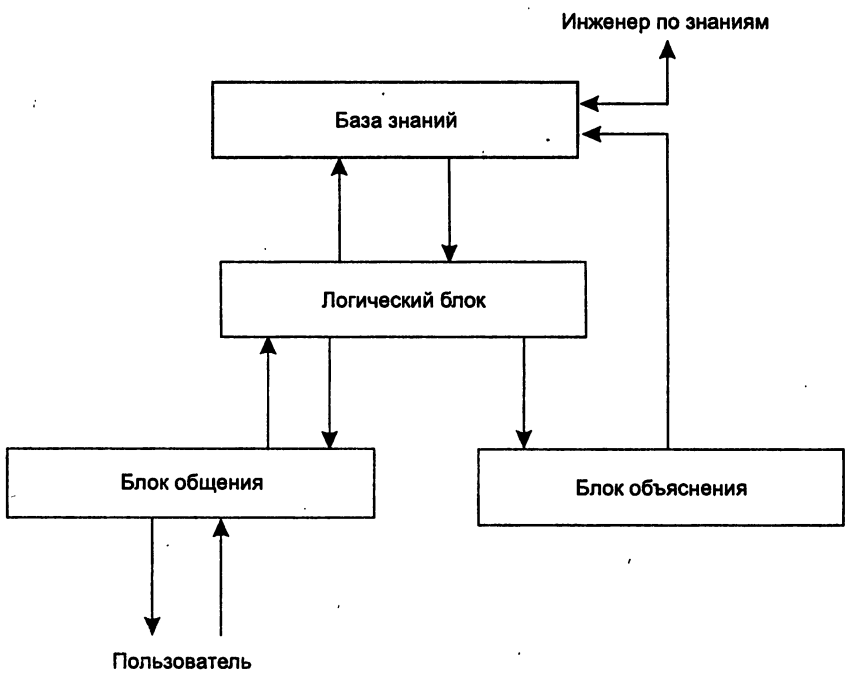


Рис. П.18. Структурная схема экспертной системы

Основу экспертной системы составляет база знаний, под которой понимают совокупность знаний, относящихся к некоторой предметной области и формально представленных таким образом, чтобы на их основе можно было осуществлять рассуждения. Знания специальным образом структурированы (методы представления знаний будут рассмотрены несколько позже), и за наполнение их базы отвечает инженер по знаниям.

Вторая часть любой экспертной системы — логический блок, или решатель. В нем реализуются, например, процедуры достоверного вывода, алгоритмы правдоподобных рассуждений и другие процедуры, предназначенные для выработки экспертных заключений.

Третий блок — блок общения, или интеллектуальный интерфейс, — организует взаимодействие пользователя с экспертной системой в удобной форме. В блоке общения используются достижения искусственного интеллекта, касающиеся понимания текстов на естественном языке, а также представления результатов работы экспертной системы в наглядном и выразительном виде.

Наконец, четвертый блок экспертной системы — это блок объяснения. Его функция состоит в выдаче информации, объясняющей и иллюстрирующей путь получения того или иного вывода, если он интересует пользователя. Например, пользователь может сомневаться в предпочтительности одного заключения перед другим. Тогда по запросу на этот счет пользователя экспертная система с развитым блоком объяснения должна аргументированно обосновать тот или иной выбор в качестве правдоподобного решения.

Любой из перечисленных блоков экспертной системы строится на базе глубоких исследований различных сторон восприятия, представления и анализа информации компьютером и человеком.

Экспертные системы классифицируют по разным основаниям [2].

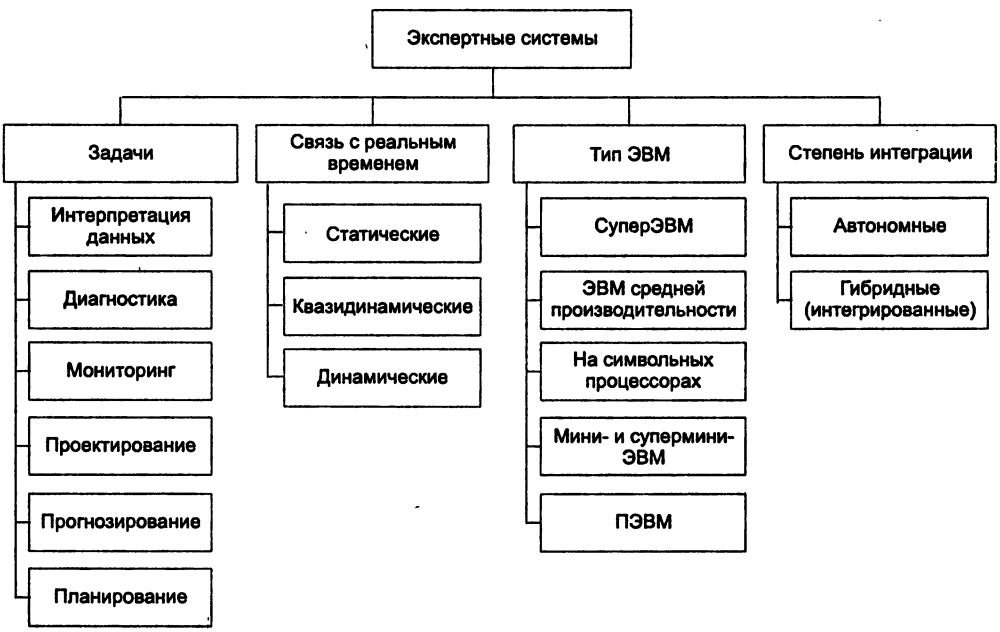


Рис. П.19. Классификация экспертных систем

Классификация экспертных систем по решаемой задаче

Экспертные системы *интерпретации данных* предназначены для определения семантики данных. Результаты интерпретации должны быть согласованными и корректными. В таких системах нередко используется многовариантный анализ данных. В настоящее время эти системы развиваются в рамках направления, получившего название Data Mining — «добыча» или «заготовка» данных.

Диагностические экспертные системы выполняют функцию отнесения объектов к определенным классам. Область применения таких систем широка — от определения неисправностей в технических системах (техническая диагностика) до распознавания заболеваний живых организмов, а также социальных и природных аномалий.

Экспертные системы *мониторинга* выполняют задачу интерпретации данных в реальном масштабе времени и сигнализируют о выходе тех или иных параметров за допустимые пределы.

Экспертные системы *прогнозирования* выводят вероятные следствия из заданных ситуаций. В прогнозирующих системах часто используются параметрические модели, в которых значения параметров «подгоняются» под анализируемую ситуацию. Кроме того, особенно в последнее время для решения задачи нередко стали применяться другие подходы. Сюда относятся, например, нейрокомпьютерный подход и различные алгоритмы поиска логических закономерностей в структурах многомерных данных.

Экспертные системы для *планирования* относятся к объектам, способным выполнять некоторые функции. В таких системах используются модели поведения реальных объектов с тем, чтобы логически вывести последствия планируемой деятельности.

Более крупно экспертные системы разделяют на системы, решающие задачи *анализа*, и на системы, решающие задачи *синтеза*. Задачи анализа отличаются от задач синтеза главным образом тем, что в них множество решений может быть перечислено и включено в систему. В задачах синтеза множество решений потенциально и строится из решений компонентов или подпроблем.

Классификация экспертных систем по связи с реальным временем

Статические экспертные системы применяются для решения задач, в которых база знаний и интерпретируемые данные не изменяются со временем. Эти системы стабильны. Сюда относится, например, диагностика неисправностей в автомобиле определенной марки.

Квазидинамические экспертные системы работают в ситуациях, которые не меняются на некотором фиксированном интервале времени. Сюда относятся, в частности, экспертные системы в микробиологии, где лабораторные измерения технологического процесса осуществляются один раз в 4—5 часов (например, производство лизина) и анализируется динамика полученных показателей по отношению к предыдущим измерениям.

Динамические экспертные системы функционируют в условиях постоянно изменяющихся данных, часто в сопряжении с датчиками объектов, иногда в режиме реального времени с непрерывной интерпретацией поступающих данных. Примеры — гибкие производственные системы, мониторинг в реанимационных палатах и т. п.

Классификация по типу ЭВМ

Эта классификация, по-видимому, не нуждается в пояснениях.

Классификация по степени интеграции с другими программами

Автономные экспертные системы применяются для решения «экспертных» задач в режиме консультации, когда не требуется привлекать дополнительные методы обработки данных (расчеты, моделирование и др.).

Гибридные экспертные системы представляют программные комплексы, объединяющие стандартные пакеты прикладных программ (например, пакеты для анализа данных, линейного программирования и т. п.). Они представляют собой интеллектуальные надстройки и выполняют функции монитора по отношению к известному программному обеспечению. Считается, что разработка гибридной системы является гораздо более сложной, чем создание автономной системы. Это связано не столько с техническими трудностями, сколько с необходимостью стыковки разных методологий, что порождает ряд теоретических трудностей.

Обучающие системы

Обучающие системы нередко называют тьюторами (англ. tutor — учитель, наставник). Они являются разновидностью экспертных систем. Тьюторы обладают повышенной способностью давать обоснованные, методически эффективные для обучения объяснения с адаптивной степенью детализации по рассматриваемым решениям. Эти системы применяют прежде всего для профессионального обучения будущих специалистов. При создании таких систем на первый план выходят знания о методе.

Этапы разработки экспертных систем

Разработка интеллектуальных систем отличается от создания обычного программного продукта. Опыт разработки ранних экспертных систем показал, что использование традиционной технологии программирования либо чрезмерно затягивает процесс разработки, либо вообще приводит к отрицательному результату [5]. Это связано главным образом с необходимостью модифицировать принципы и способы построения по мере того, как увеличивается знание разработчиков о проблемной области.

В настоящее время сложилась последовательность действий при разработке экспертных систем. Она включает следующие этапы [2], [5], [9]: идентификация, получение знаний, концептуализация, формализация, выполнение, тестирование и опытная эксплуатация.

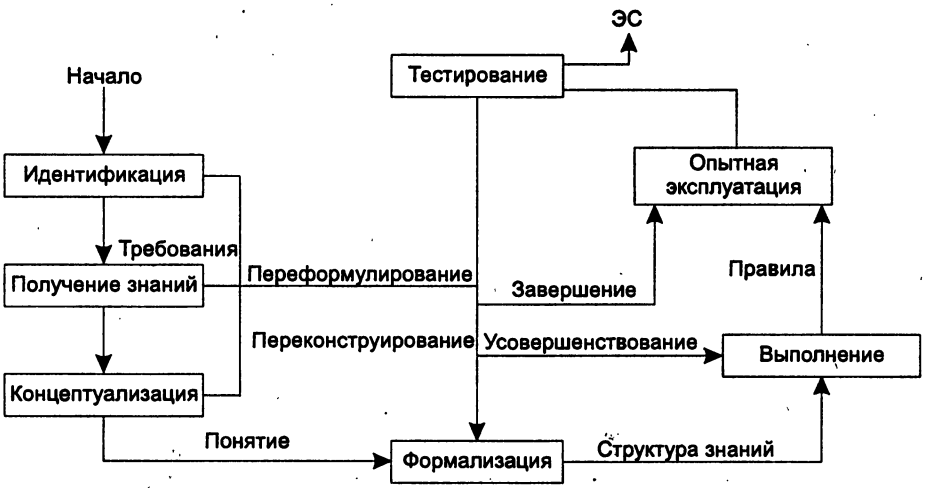


Рис. П.20. Технология разработки экспертных систем

Идентификация

Этап идентификации связан прежде всего с осмыслением тех задач, которые предстоит решать будущей экспертной системе, и формированием требований к ней. На этом этапе планируется ход разработки прототипа системы, определяются источники знаний (книги, эксперты, методики), цели (распространение опыта, автоматизация рутинных действий), классы решаемых задач и т. д. Результатом этапа идентификации является ответ на вопрос, что надо сделать и какие ресурсы необходимо задействовать.

Получение знаний

При решении проблемы получения знаний выделяют три стратегии: приобретение знаний, извлечение знаний и обнаружение знаний.

Под *приобретением* (acquisition) знаний понимается способ автоматизированного наполнения базы знаний посредством диалога эксперта и специальной программы. Автоматизированные средства приобретения знаний получают готовые фрагменты знаний в соответствии со структурами, заложенными разработчиками систем. Большинство таких инструментальных средств ориентировано на конкретные экспертные системы с жестко обозначенной предметной областью и моделью представления знаний. Например, система TEIRESIAS [6] предназначена

для пополнения базы знаний системы MYCIN или ее дочерних ветвей, построенных на оболочке EMYCIN [8] в области медицинской диагностики с использованием продукционной модели представления знаний. При попытке использования систем приобретения знаний в других областях разработчикам нередко приходится сталкиваться со следующими трудностями:

- неудачный способ приобретения, не совпадающий со структурой знаний в данной области;
- неадекватная модель представления знаний;
- отсутствие целостной системы знаний в результате приобретения только «фрагментов»;
- упрощение и уплощение «картины мира» и пр.

Извлечением (elicitation) знаний называют процедуру взаимодействия инженера по знаниям с источником знаний (экспертом, специальной литературой и др.) без использования вычислительной техники. Это длительная и трудоемкая процедура, в которой инженеру по знаниям, владеющему методами когнитивной психологии, системного анализа, математической логики и пр., нужно воссоздать модель предметной области, используемой экспертами для принятия решений.

Актуальность задачи извлечения знаний при разработке интеллектуальных систем обусловлена следующими причинами. Во-первых, значительная часть знаний эксперта является результатом многочисленных наслоений, ступеней опыта, и эксперт нередко не способен самостоятельно проанализировать все детали в цепи своих умозаключений. Во-вторых, диалог инженера по знаниям и эксперта служит наиболее естественной формой «раскручивания» лабиринтов памяти эксперта, в которых хранятся знания, часто носящие невербальный характер. И в-третьих, многочисленные причинно-следственные связи реальной предметной области образуют сложную систему, скелет которой иногда более доступен для восприятия аналитика, владеющего системной методологией и не обремененного знанием большого количества подробностей.

Термины *«обнаружение знаний»* (knowledge discovery), а также Data Mining связывают с созданием компьютерных систем, реализующих методы автоматического получения знания. На сегодняшний день это наиболее перспективное направление инженерии знаний, предполагающее, что в результате автоматизации процесса обучения система «сможет» самостоятельно раскрыть закономерности предметной области и сформировать необходимые знания на основе имеющегося эмпирического материала (баз данных). В настоящее время специалистам стало ясно, что инженер по знаниям с помощью одного лишь диалога с экспертом в какой-то конкретной области не способен добыть все нужные для разработки интеллектуальной системы сведения. Требуется еще и множество примеров, на которых удастся обучить машину [3].

Концептуализация

На этапе концептуализации проводится содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи, определяются методы решения задач. Этот этап завершается созданием модели предметной области, включающей основные концепты и отношения. Модель представляется в виде графа, таблицы, диаграммы или текста.

Формализация

На этапе формализации все ключевые понятия и отношения выражаются на некотором формальном языке, который выбирается из числа уже существующих либо создается заново. Другими словами, на данном этапе определяются состав средств и способы представления декларативных и процедурных знаний, осуществляется это представление, и в итоге создается описание решения задачи экспертной системы на выбранном формальном языке.

Выполнение (реализация)

На этапе выполнения создается один или несколько реально работающих прототипов экспертной системы. Для ускорения этого процесса в настоящее время широко применяются различные *инструментальные средства*, характеристика которым дается ниже.

Тестирование

На данном этапе оценивается и проверяется работа программы-прототипа с целью приведения ее в соответствие с реальными запросами пользователей. Прототип проверяется по следующим основным позициям:

- удобство и адекватность интерфейсов ввода/вывода (характер вопросов в диалоге, связность выводимого текста результата и др.);
- эффективность стратегии управления (порядок перебора, использование нечеткого вывода и т. д.);
- корректность базы знаний (полнота и непротиворечивость правил).

Задача стадии тестирования — выявление ошибок и выработка рекомендаций по доводке прототипа экспертной системы до промышленного образца.

Опытная эксплуатация

На этапе опытной эксплуатации проверяется пригодность экспертной системы для конечного пользователя. Пригодность определяется в основном удобством и полезностью разработки. Под полезностью понимается способность экспертной системы определять в ходе диалога потребности пользователя, выявлять и

устранять причины неудач в работе, а также удовлетворять указанные потребности пользователя (решать поставленные задачи). В свою очередь, удобство работы подразумевает естественность взаимодействия с экспертной системой, гибкость (способность системы настраиваться на различных пользователей, а также учитывать изменения в квалификации одного и того же пользователя) и устойчивость системы к ошибкам (способность не выходить из строя при ошибочных действиях пользователя).

После успешного завершения этапа опытной эксплуатации экспертная система классифицируется как коммерческая система, пригодная не только для собственного использования, но и для продажи различным потребителям.

В ходе разработки экспертной системы всегда осуществляется ее модификация. Выделяют следующие виды такой модификации: переформулирование понятий и требования, переконструирование представления знаний в системе и усовершенствование прототипа (см. рис. П.20). Усовершенствование прототипа производится в процессе циклического прохождения через этапы выполнения и тестирования для отладки правил и процедур вывода. Циклы повторяются до тех пор, пока система не будет вести себя ожидаемым образом. Изменения, осуществляемые при усовершенствовании, зависят от выбранного способа представления знаний и класса решаемых задач. Если в процессе усовершенствования желаемое поведение не достигается, то производят более серьезные модификации архитектуры системы и используемой базы знаний.

Возврат от этапа тестирования на этап формализации приводит к пересмотру ранее выбранного способа представления знаний. Данный цикл называют переконструированием. Если возникшие проблемы еще более серьезны, то после неудачи на этапе тестирования может потребоваться возврат на этапы концептуализации и даже идентификации. В этом случае речь идет о переформулировании системы понятий, метапонятий и семантических отношений, то есть о проектировании всей системы заново.

Инструментальные средства

Раньше на проектирование и создание экспертных систем требовалось 20—30 человеко-лет. [1]. В настоящее время имеются средства, ускоряющие этот процесс. Их называют инструментальными средствами, или просто инструментарием. Иными словами, под инструментальными средствами понимают совокупность аппаратного и программного обеспечения, позволяющего создавать прикладные системы, основанные на знаниях.

Среди программных инструментальных средств выделяют следующие большие группы:

- символьные языки программирования (например, LISP, INTERLISP, SMALLTALK);
- языки инженерии знаний, то есть языки программирования, позволяющие реализовать один из способов представления знаний (например, OPS-5, LOOPS, Пролог, KES);

- оболочки экспертных систем (или пустые экспертные системы), то есть системы, не содержащие знаний ни о какой предметной области (например, EMYCIN, ЭКО, ЭКСПЕРТ);
- среды или окружение (environment) для разработки экспертных систем, то есть системы, автоматизирующие разработку (проектирование) систем (например, KEE, ART, TEIRESIAS).

При использовании инструментальных средств первого и второго типа в задачу разработчика входит программирование всех компонентов экспертной системы, а при использовании инструментальных средств третьего и четвертого типа разработчик, как правило, полностью освобождается от написания программ.

В настоящее время наблюдается отход от инструментария первого и второго типа и широкое использование инструментальных средств четвертого типа.

Вместе с тем, следует отметить, что к выбору инструментария необходимо подходить очень осторожно, так как управляющие стратегии, вложенные в процедуры вывода инструментальных сред и оболочек, могут не соответствовать методам решения, которые использует эксперт, взаимодействующий с данной системой, что способно приводить к неэффективным, а возможно, и неправильным решениям.

Литература

1. Вассерман Л. И., Дюк В. А., Иовлев Б. В., Червинская К. Р. Психологическая диагностика и новые информационные технологии. — СПб.: СЛП, 1997.
2. Гаврилова Т. А., Червинская К. Р. Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992.
3. Десять лет спустя (интервью с Д. Мичи)//Будущее искусственного интеллекта. — М.: Наука, 1991. — С. 213—215.
4. Попов Э. В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ. — М.: Наука, 1987.
5. Попов Э. В. Особенности разработки и использования экспертных систем// Искусственный интеллект. Кн. 1. Системы общения и экспертные системы. — М.: Радио и связь, 1990.
6. Davis R. TEIRESIAS: Applications of meta-level knowledge//Knowledge-based Systems in Artifical Intelligence. — N.-Y.: McGraw-Hill, 1982.
7. Osgood Ch. E., Susi G. E., Tannenbaum P. N. The Measurment of Meaning. — Urbana: Univ. I 11. Press, 1957.
8. Shortliffe E. Computer based medical consultations: MYCIN. N.-Y.: American Elsevier, 1976.
9. Waterman D. A. Guide to expert Systems. — N.-Y.: Addison.-Welse, 1986.

Извлечение знаний из памяти эксперта

В этом разделе рассмотрены особенности процедуры взаимодействия инженера по знаниям с источником знаний (экспертом), позволяющей сделать явными рассуждения специалистов при принятии решений и структуру их представлений о предметной области.

Процедура взаимодействия инженера по знаниям с экспертом

Известен парадоксальный факт Джонсона о том, что по мере накопления опыта специалист-эксперт все больше утрачивает умение словесно выражать свои знания. Имеются достаточно убедительные психологические доказательства того, что люди далеко не всегда в состоянии достоверно описать свои мыслительные процессы. Теоретик искусственного интеллекта Марвин Минский писал, что «самосознание — это сложная, но тщательно сконструированная иллюзия...» и что «...только как исключение, а не как правило, человек может объяснить то, что он знает».

Другое психологическое положение состоит в том, что опыт эксперта — это интуиция, которая трудно поддается выражению в форме правил типа «ЕСЛИ — ТО».

Широко известно высказывание Лао-Цзы («старого учителя»): «Кто скажет, тот не знает, кто знает, тот не скажет».

Тем не менее, инженерия знаний предлагает определенные методы (приемы, способы) работы с экспертами. Эти методы направлены на «раскручивание» лабиринтов памяти экспертов, в которых хранятся знания, часто имеющие невербальный характер.

Классификация методов работы с экспертами

В основу излагаемого материала положена классификация коммуникативных методов работы инженера по знаниям, представленная в [3].

Под коммуникативными методами понимают все виды контактов инженера по знаниям с живым источником знаний — экспертом. Среди этих методов выделяют две большие группы: активные и пассивные (рис. П.21).

Пассивные методы подразумевают, что ведущая роль в процедуре извлечения знаний принадлежит эксперту. При этом инженер по знаниям главным образом протоколирует рассуждения и действия эксперта.

В *активных* методах инициатива полностью в руках инженера по знаниям. Он ведет с экспертом беседу, предлагает различные «игры», организует «круглый стол» и т. д.

Пассивные методы на первый взгляд просты. Вместе с тем, они требуют от инженера по знаниям умения анализировать «поток сознания» эксперта и выделять в нем ценные фрагменты знания.

Активные методы разделяют на две группы в зависимости от числа экспертов, участвующих в процедуре извлечения знаний. В групповых методах большое значение имеет дискуссия между экспертами, в которой нередко выявляются нетривиальные аспекты знаний. В то же время, ведущую роль на сегодняшний день играют индивидуальные методы. В значительной степени это связано с деликатностью процедуры «отъема знаний».

Рис. П.21. Классификация методов работы с экспертами

Пассивные методы

Наблюдения

Метод наблюдения является единственным «чистым» методом, где инженер по знаниям не вмешивается в процесс работы эксперта и не навязывает ему

какие-либо собственные представления. Выделяют две разновидности наблюдений:

- Наблюдение за реальным процессом.
- Наблюдение за имитацией процесса.

Сначала обычно применяют первую разновидность и наблюдают за реальным процессом на рабочем месте эксперта. Это помогает глубже понять предметную область и отметить все внешние особенности процедуры принятия решений, необходимые для проектирования интерфейса пользователя.

На втором этапе эксперт имитирует процесс. В таком режиме он менее напряжен и работает на «два фронта» — ведет профессиональную деятельность и одновременно демонстрирует ее.

Сеансы наблюдений предъявляют к инженеру по знаниям следующие требования:

- Владение техникой стенографии.
- Знакомство с методиками хронометрирования для четкого структурирования производственного процесса во времени.
- Развитые навыки «чтения по глазам», то есть наблюдательность к жестам, мимике и другим невербальным компонентам общения.
- Предварительное знакомство с предметной областью.

Протоколы наблюдений после проведения сеансов тщательно расшифровываются, а затем обсуждаются с экспертом.

Анализ протоколов «мыслей вслух»

При протоколировании «мыслей вслух» эксперта просят раскрыть всю цепочку рассуждений, объясняющих его действия и решения. При таком протоколировании считается важным зафиксировать не только весь «поток сознания» эксперта, но даже паузы и междометия в речи эксперта. Иногда данный метод называют «вербальными отчетами».

При протоколировании «мыслей вслух» эксперт может проявить себя максимально ярко. Он ничем не скован, ему никто не мешает, он как бы свободно парит в потоке собственных рассуждений и умозаключений, может блеснуть своей эрудицией и продемонстрировать глубину познаний. Для большого числа экспертов это самый приятный и лестный способ извлечения знаний.

Вместе с тем, как отмечалось выше, далеко не каждый специалист, даже из числа умеющих произносить впечатляющие монологи о своей работе, оказывается в состоянии формализовать и структурировать рассуждения. Однако существуют люди, склонные к рефлексии, способные к конструктивному изложению мыслей. Такие люди — находка для инженера по знаниям.

Лекции

Лекторский дар встречается нечасто. Опытный лектор хорошо структурирует свои знания и ход рассуждений. Но бывает, некоторые люди обладают лекторским

даром, но не подозревают о его присутствии. В любом случае инженеру по знаниям стоит попробовать озадачить эксперта подготовкой лекции на интересующую тему. Если эксперт сумеет преодолеть специфический психологический барьер и войти в образ педагога, это может оказаться весьма эффективным для решения задачи извлечения знаний.

Хороший вопрос инженера по знаниям по ходу лекции имеет важное значение. Серьезные, глубокие и интересные вопросы, с одной стороны, стимулируют творческое воображение лектора, и с другой — повышают авторитет инженера по знаниям.

Продолжительность лекций рекомендуется стандартная — от 40 до 50 минут, и через 5–10 минут — еще столько же. Весь курс должен занимать, как правило, от двух до пяти лекций.

Метод извлечения знаний в форме лекций, как и все пассивные методы, применяют в начале многоэтапной процедуры извлечения знаний из памяти эксперта. Он способствует быстрому погружению инженера по знаниям в предметную область.

Таблица П.11. Сравнительные характеристики пассивных методов извлечения знаний

Пассивный метод извлечения знаний	Достоинства	Недостатки	Требования к эксперту (типы и основные качества)	Требования к инженеру по знаниям (типы и основные качества)	Характеристика предметной области
Наблюдения	Отсутствие влияния аналитика и его субъективной позиции. Максимальное приближение аналитика к предметной области	Отсутствие обратной связи	Мыслитель или практик (способность к комментариям + рефлективность + дружелюбие)	Мыслитель (наблюдательность + полезависимость)	Слабо- и среднеструктурированные среднедокументированные
«Мысли вслух»	Свобода самовыражения эксперта. Обнаженность структур рассуждений. Отсутствие влияния аналитика и его субъективной позиции	Отсутствие обратной связи. Возможность ухода «в сторону» в рассуждениях эксперта. «Зашумленность» деталями	Собеседник или мыслитель (способность к вербализации мыслей + открытость + рефлективность)	Мыслитель или собеседник (контактность + поленезависимость)	То же
Лекции	Структурированное изложение. Высокая концентрация знаний. Отсутствие аналитика и его субъективной позиции	Слабая обратная связь. Недостаток хороших лекторов среди практиков	Мыслитель (лекторские способности)	Мыслитель (поленезависимость + способность к обобщению)	Слабодокументированные и слабоструктурированные

ПРИМЕЧАНИЕ Полезависимость отражает способность человека концентрировать внимание лишь на тех аспектах проблемы, которые необходимы для решения конкретной задачи, и умение отбрасывать лишнее.

Активные индивидуальные методы

Анкетирование

Анкетирование является наиболее стандартизированным методом. Составление анкеты — достаточно тонкий и ответственный момент. Вот несколько рекомендаций:

- анкета не должна быть монотонной и однообразной, чтобы не вызывать скуку и усталость. Для этого вопросы должны варьироваться, тематика меняться. Кроме того, нередко в анкету вставляют специальные вопросы-шутки и игровые вопросы;
- анкета должна быть приспособлена к языку экспертов;
- следует учитывать, что вопросы влияют друг на друга. Поэтому последовательность вопросов должна быть хорошо продумана;
- анкета должна иметь «хорошие манеры». Ее нужно излагать ясным, понятным и предельно вежливым языком. Методическим мастерством составления анкеты можно овладеть только на практике.

Процедура анкетирования может проводиться двумя способами. В первом аналитик вслух задает вопросы и сам заполняет анкету по ответам эксперта. Во втором эксперт заполняет анкету самостоятельно после предварительного инструктирования.

Выбор способа зависит от ряда условий (в частности от оформления анкеты, ее понятности, готовности эксперта). Вместе с тем, второй способ представляется предпочтительным, так как у эксперта появляется неограниченное время на обдумывание вопросов и снижается так называемый эффект присутствия.

Интервью

Перед проведением интервью неплохо спросить себя: «А умеем ли мы задавать вопросы?» В философии эта проблема обсуждается с древности. Рассмотрим классификацию вопросов (рис. П.22).

Открытый вопрос обозначает тему или предмет, предоставляя эксперту свободу по форме и содержанию ответа.

При *закрытом вопросе* эксперту предлагается набор ответов, среди которых он должен сделать выбор.

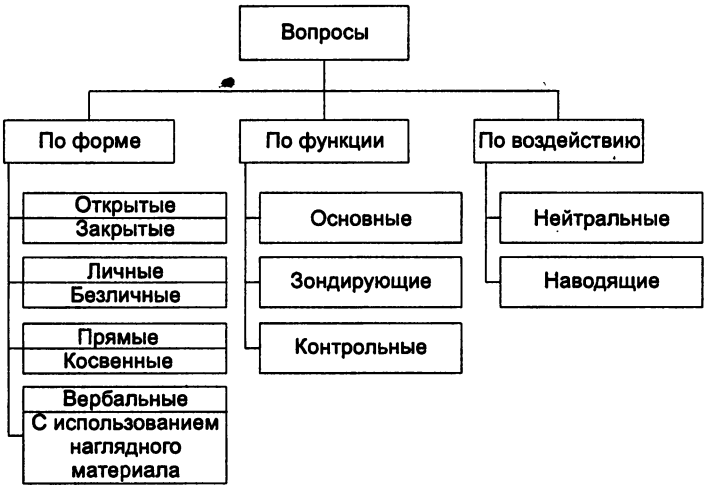


Рис. П.22. Классификация вопросов

Закрытые вопросы легче обрабатываются, но они в определенной мере «программируют» ответ эксперта и «закрывают» ход его рассуждений. Поэтому при составлении сценария интервью обычно чередуют открытые и закрытые вопросы и особенно тщательно продумывают «меню» и содержание закрытых вопросов.

Личный вопрос апеллирует к индивидуальному опыту эксперта. Личные вопросы обычно активизируют мышление эксперта, «играют» на его самолюбии, украшают интервью.

Безличный вопрос нацелен на выявление наиболее распространенных и общепринятых закономерностей предметной области.

При подготовке вопросов учитывают, что языковые возможности эксперта, как правило, ограничены. Кроме того, имеют в виду, что из-за замкнутости, скованности и робости отдельные эксперты не могут сразу высказать свое мнение и предоставить требуемые знания. Поэтому часто используют не *прямые* вопросы, которые непосредственно указывают на предмет или тему, а *косвенные*, опосредованно направляющие внимание на актуальную проблему. Иногда в интересах дела приходится задавать несколько косвенных вопросов вместо одного прямого.

Вербальные вопросы — это традиционные устные вопросы.

Вопросы с использованием наглядного материала разнообразят интервью и снижают утомляемость эксперта. В качестве наглядного материала используют фотографии, рисунки и карточки.

Разделение вопросов по функции на основные, зондирующие и контрольные связано с тем, что нередко эксперт по каким-то причинам уходит в сторону от вопроса и *основные* вопросы интервью оказываются непродуктивными. Тогда аналитик применяет *зондирующие* вопросы, концентрирующие внимание эксперта в нужном направлении. *Контрольные* вопросы используют для проверки достоверности и объективности полученной информации.

Нейтральные вопросы носят беспристрастный характер. В то же время, *наводящие вопросы* заставляют эксперта прислушаться или даже принять во внимание позицию интервьюера.

Кроме приведенных в классификации на рис. П.22, полезно различать и включать в интервью следующие виды вопросов [5]:

- *контактные* («ломающие лед» между аналитиком и экспертом);
- *буферные* (для разграничения различных тем интервью);
- *оживляющие* память экспертов (для реконструкции отдельных случаев из практики);
- «*провоцирующие*» (для получения спонтанных, неподготовленных ответов).

Свободный диалог

При свободном диалоге инженера по знаниям с экспертом отсутствует какой-либо регламентированный план. Однако эта форма извлечения знаний требует самой серьезной предварительной подготовки. На рис. П.23 показана одна из рекомендуемых схем такой подготовки.

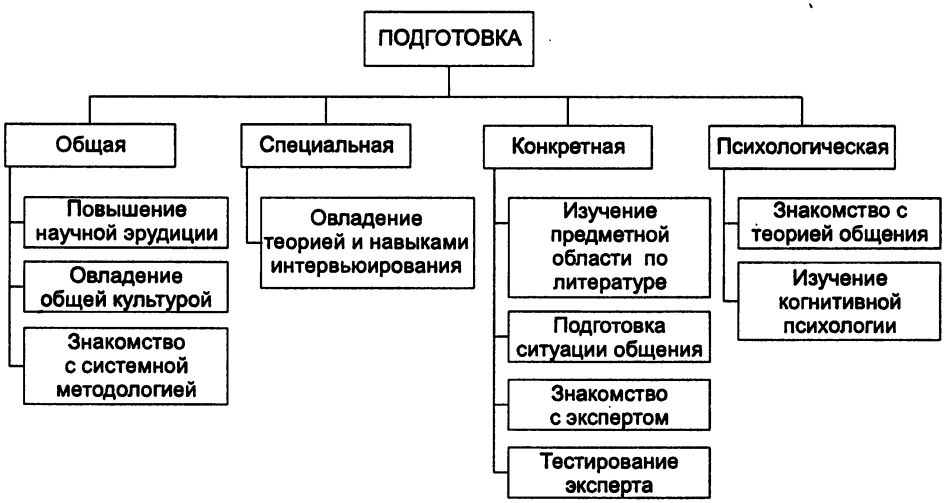


Рис. П.23. Схема подготовки к интервью и свободному диалогу

Квалифицированная подготовка к диалогу — подлинная драматургия. В ее сценарии предусматривают плавное развитие процедуры извлечения знаний от приятного впечатления в начале беседы к профессиональному контакту через пробуждение интереса и завоевание доверия эксперта.

Для обеспечения желания эксперта продолжать беседу обычно производят «поглаживания» типа: «Я Вас понимаю...», «...Это очень интересно» и т. п. При этом поведение аналитика должно быть искренним, ведь давно известно, что лучшая уловка — избегать всяких уловок и относиться к собеседнику с истинным уважением и настоящим интересом.

Существует каталог свойств идеального интервьюера [5]: «Он должен выглядеть здоровым, спокойным, уверенным, внушать доверие, быть искренним, веселым, проявлять интерес к беседе, быть опрятно одетым, ухоженным».

Для ведения продуктивного диалога полезно ознакомиться с мнением классика отечественного литературоведения М. М. Бахтина (цит. по [3]):

«Диалог — столкновение разных умов, разных истин, несходных культурных позиций, составляющих единый ум, единую истину, общую культуру.

...Диалог предполагает:

- уникальность каждого партнера и их принципиальное равенство друг другу;
- различие и оригинальность их точек зрения;
- ориентацию каждого на понимание и на активную интерпретацию его точки зрения партнером;
- ожидание ответа и его предвосхищение в собственном высказывании;
- взаимную дополнительность позиций участников общения, соотнесение которых и является целью диалога».

Сравнительные характеристики активных индивидуальных методов извлечения знаний представлены в табл. П.12.

Таблица П.12. Сравнительные характеристики активных индивидуальных методов

Метод	Достоинства	Недостатки	Требования к эксперту	Требования к аналитику	Характеристика предметной области
Анкетирование	Возможность стандартизированного опроса нескольких экспертов. Не требует особенного напряжения от аналитика во время процедуры анкетирования	Требует умения и опыта составления анкет. Отсутствие контакта между экспертом, нет обратной связи. Вопросы анкеты могут быть неправильно поняты экспертом	Практик и мыслитель	Мыслитель (педантизм в обработке и составлении анкет, внимательность)	Слабоструктурированные, слабо- и среднедокументированные
Интервью	Наличие обратной связи (возможность уточнений и разрешения противоречий)	Требует значительного времени на подготовку вопросов интервью	Собеседник или мыслитель	Собеседник (журналистские навыки, умение слушать)	То же
Свободный диалог	Гибкость. Сильная обратная связь. Возможность изменения сценария и формы сеанса	Требует от аналитика высочайшего напряжения. Отсутствие формальных методик. Трудность протоколирования	Собеседник или мыслитель	Собеседник (наблюдательность, умение слушать, обаяние)	То же

Активные групповые методы

Активные групповые методы сами по себе не могут служить источником более или менее полного знания. Они выступают как дополнительные и служат хорошей «приправой» к индивидуальным методам извлечения знаний, активизирующей мышление и поведение экспертов.

«Круглый стол»

Метод круглого стола предполагает равноправное обсуждение интересующей проблемы несколькими экспертами. Задача дискуссии — коллективно, с разных точек зрения, под разными углами исследовать спорные проблемы предметной области. Для остроты на «круглый стол» приглашают представителей различных научных направлений и поколений. Число участников дискуссии обычно колеблется от трех до пяти-семи.

Перед началом дискуссии ведущему (инженеру по знаниям) необходимо убедиться, что все участники правильно понимают задачу. Затем нужно установить регламент и четко сформулировать тему.

По ходу дискуссии важно проследить, чтобы слишком эмоциональные и разговорчивые эксперты не подменяли тему и чтобы критика позиций друг друга была обоснованной. Определенные усилия ведущий должен приложить для уменьшения «эффекта фасада», когда у участников превалирует желание произвести впечатление на других и они говорят совсем не то, что сказали бы в нормальной обстановке.

Научная плодотворность дискуссий делает их привлекательными не только для инженера по знаниям, но и для самих экспертов, особенно для тех, кто знает меньше. Как отмечал Эпикур: «При философской дискуссии больше выигрывает побежденный — в том отношении, что он умножает знания» (цит. по [3]).

«Мозговой штурм»

«Мозговой штурм» или «мозговая атака» — один из наиболее популярных методов раскрепощения и активизации человеческого мышления. Впервые этот метод был использован в 1939 году А. Осборном в США для генерации новых идей. Основное положение штурма — отделение процедуры генерации идей в замкнутой группе специалистов от процесса их анализа и оценки.

Обычная продолжительность штурма — порядка 40 минут. Количество участников — до 10 человек. Этим участникам предлагается высказать на заданную тему любые мысли, в том числе шутливые, фантастические и ошибочные. Критика запрещена. Регламент — до 2 минут на выступление.

Из опыта известно, что число высказанных идей часто превышает 50. Наиболее существенный момент штурма — наступление пика (ажиотажа), когда идеи начинают буквально «фонтанировать». Последующий анализ, который проводит группа сторонних экспертов, как правило, показывает, что всего лишь 10–15 % идей разумны, но среди них встречаются весьма оригинальные.

Искусство инженера по знаниям, проводящего «мозговой штурм», заключается в способности задавать вопросы аудитории, «подогревая» аудиторию. Вопросы служат своеобразным «крючком», которым извлекаются идеи.

Достоинства и недостатки активных групповых методов охарактеризованы в табл. П.13.

Таблица П.13. Характеристики активных групповых методов

Метод	Достоинства	Недостатки	Требования к эксперту	Требования к аналитику	Область
«Круглый стол»	Позволяет получить более объективные фрагменты знаний. Оживляет процедуру извлечения. Позволяет участникам обмениваться знаниями	Требует больших организационных затрат. Отличается сложностью проведения	Собеседник или мыслитель (искусство полемики)	Собеседник (дипломатические способности)	Слабоструктурированные и слабодокументированные с наличием спорных проблем
«Мозговой штурм»	Активизирует мышление экспертов. Позволяет выявлять глубинные пласты знаний (на уровне бессознательного). Позволяет получать новое знание (гипотезы)	Возможен только для новых интересных проблем. Не всегда эффективен (довольно низкий процент продуктивных идей)	Мыслитель (креативность, то есть способность к творчеству)	Собеседник или мыслитель (быстрая реакция и чувство юмора)	То же + наличие перспективных «белых пятен»

Экспертные игры

Плодотворность моделирования реальных ситуаций в играх сегодня подтверждена практически во всех областях науки и техники. Ниже рассмотрены различные виды экспертных игр в соответствии с классификацией, введенной в [3].

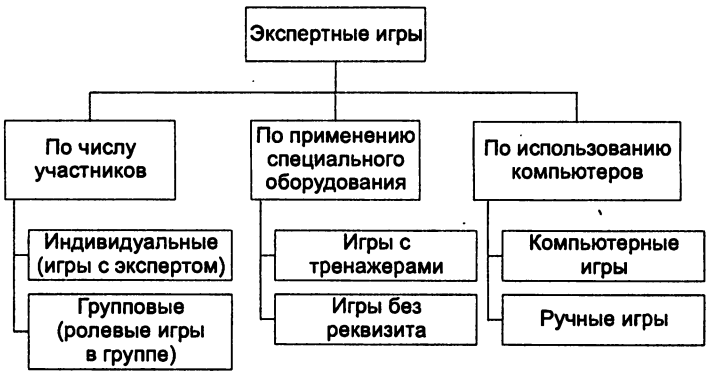


Рис. П.24. Классификация экспертных игр

Игры с экспертом

В играх с экспертом инженер по знаниям берет на себя какую-либо роль в моделируемой ситуации. Например, это может быть роль Ученика, который на глазах у эксперта (Учителя), поправляющего Ученика, выполняет работу на заданную тему. Такая игра — хороший способ разговорить застенчивого эксперта.

Другой пример — игра в Специалиста (инженер по знаниям) и Консультанта (эксперт). Эта игра дает иногда впечатляющие результаты. В частности, интересный случай описан в [4]. Здесь эксперт выступал в роли врача, хорошо знающего больного, а консультант задавал вопросы и делал прогноз о целесообразности применения того или иного вида лечения. Такая игра позволила установить, что требуется всего 30 вопросов для успешного прогноза, в то время как первоначальный вариант вопросника, составленного медиками, содержал 170 вопросов.

Для выявления скрытых пластов знания применяется игра, в которой специалист делает прогнозы в профессиональных ситуациях и дает им обоснования. Затем по истечении определенного времени специалисту предъявляют его собственные обоснования и просят произвести по ним прогнозы. Как оказывается, такой простой прием нередко позволяет обнаружить пропущенные шаги в рассуждениях эксперта.

В игре «фокусировка на контексте» эксперт выполняет роль экспертной системы, а инженер по знаниям — роль пользователя. Моделируется ситуация консультации. Первые реакции эксперта концентрируются вокруг наиболее значимых понятий и самых важных аспектов проблемы.

В целом по играм с экспертом даются следующие основные советы инженеру по знаниям:

- Играйте смелее, придумывайте игры сами.
- Не навязывайте игру эксперту, если он не расположен.
- Не «давите» на эксперта, не забывайте цели игры.
- Не забывайте о времени и о том, что игра утомительна для эксперта.
- Играйте весело, нешаблонно.

Ролевые игры в группе

В каждой групповой игре заранее составляется сценарий, распределяются роли, готовятся портреты-описания ролей и разрабатывается система оценивания игроков.

Известны различные способы проведения ролевых игр. В одних играх участники придумывают себе новые имена и выступают под ними. В других все игроки переходят на «ты». В третьих роли выбирают игроки, в четвертых для распределения ролей вытягивается жребий.

Обычно в игре, предназначенной для получения знания, принимают участие от трех до шести экспертов. В случае большего числа экспертов они разбиваются на группы, между которыми организуется состязание: чей диагноз окажется ближе

к истинному, чей план рациональнее использует ресурсы, кто быстрее определить неисправность в техническом блоке и т. п.

Создание игровой обстановки требует фантазии и выдумки от инженера по знаниям. Главное, чтобы эксперты в игре максимально погрузились в ситуацию, действительно «заиграли», раскрепостились и «раскрыли свои карты».

Игры с тренажерами

Тренажеры широко применяются для обучения профессиям, требующим динамического реагирования на изменяющуюся производственную ситуацию. Сюда относятся профессии летчиков, судоводителей, операторов атомных станций и др. Применение тренажеров для извлечения знаний позволяет фиксировать фрагменты так называемых летучих знаний. Эти знания сиюминутны и, как правило, трудно воспроизводимы и выпадают из памяти в обычной обстановке при выходе из моделируемой ситуации.

Компьютерные экспертные игры

Компьютерные игры разделяют на следующие классы [6]:

- позиционные игры (шахматы, шашки, го);
- динамические игры, связанные с сенсомоторными реакциями;
- зрелищные или диалоговые фильмы, где пользователь может влиять на сюжет;
- обучающие игры.

Экспертные игры сочетают в себе элементы всех перечисленных выше классов. Сравнительные характеристики этих игр представлены в табл. П.14.

Таблица П.14. Сравнительные характеристики экспертных игр

Экспертные игры	Достоинства	Недостатки	Требования к эксперту	Требования к аналитику	Области
Индивидуальные	Дают возможность сравнительно быстро получать качественную картину принятия решения. Позволяют выяснять, какую информацию и как использует эксперт	Отсутствие методик и стандартного набора игр. Высокие профессиональные требования к аналитику	Собеседник или практик (раскованность и актерское мастерство)	Собеседник (режиссерские способности, умение создавать сценарий, актерское мастерство)	Средне- и слабоструктурированные и слабодокументированные
Групповые	Реалистично воссоздают атмосферу конкретной задачи. Раскрепощают экспертов. «Групповые» знания более объективны. Выявляют логику и аргументацию экспертов	Требуют от аналитика знания основ игротехники. Сложность создания игр для конкретных предметных областей	То же	То же + способность к ведению конферанса	То же

Экспертные игры	Достоинства	Недостатки	Требования к эксперту	Требования к аналитику	Области
Компьютерные	Вызывают интерес у эксперта. Привлекают дизайном и динамикой	Сложность и высокая стоимость создания специализированных игр по конкретной предметной области	Практик без психологического барьера по отношению к компьютеру	Мыслитель	То же

Структурирование знаний

Проблема структурирования знаний тесно связана с проблемой извлечения, но рассматривается как бы под другим углом зрения. Вопрос заключается не в том, как надо взаимодействовать с экспертом, а в том, что надо получить в результате такого взаимодействия и что конкретно построить для решения данной задачи. Структурирование знаний иногда называют также концептуальным анализом знаний.

Целью концептуального анализа знаний является построение модели предметной области, которая представляет собой проблемно-ориентированные и системно-независимые структуры.

Любая модель предметной области включает в себя *систему понятий*, *семантические отношения* и *стратегии принятия решений* [2].

Система понятий

Под *системой понятий* подразумевают совокупность единиц смысловой информации, отражающих реальные явления, процессы, факты, объекты и т. д. предметной области. Инженерия знаний предлагает ряд практических методов работы с экспертами для формирования системы понятий [9]:

- метод локального представления;
- метод вычисления коэффициента использования;
- метод формирования перечня понятий;
- составление списка элементарных действий;
- составление оглавления учебника;
- текстологический метод.

Цель этих методов заключается в том, чтобы сформировать систему понятий, обладающую свойствами полноты, уникальности, достоверности и непротиворечивости.

Метод локального представления

При построении системы понятий с помощью *метода локального представления* эксперта просят разбить задачу на подзадачи для перечисления целевых состояний описания общих категорий цели. Далее для каждого разбиения (локального представления) эксперт формулирует информационные факты и дает им четкое наименование (название). Считается, что для успешного решения задачи построения модели предметной области число информационных фактов в каждом локальном представлении, которыми человек способен одновременно манипулировать, должно быть примерно равно семи.

Метод вычисления коэффициента использования

Этот метод основан на следующей гипотезе. Информационный факт может являться понятием, если:

1) он используется в большом числе подзадач;
2) используется вместе с большим числом других информационных фактов;
3) редко используется совместно с другими информационными фактами по сравнению с общим числом его использования во всех подзадачах (это и есть коэффициент использования). Полученные значения коэффициента использования служат критерием для классификации информационных фактов и, таким образом, для формирования системы понятий.

Метод формирования перечня понятий

Данный метод заключается в том, что экспертам (желательно, чтобы их было больше двух) дается задание составить список понятий, относящихся к исследуемой предметной области. Понятия, выделенные всеми экспертами, включаются в систему понятий, остальные подлежат обсуждению.

Метод составления списка элементарных действий

Здесь эксперту дается задание составить список элементарных действий при решении задачи в произвольном порядке.

Метод составления оглавления учебника

В данном случае эксперту предлагается представить ситуацию, в которой его попросили написать учебник. Необходимо составить на бумаге перечень предполагаемых глав, разделов, параграфов, пунктов и подпунктов книги.

Текстологический метод

Этот метод формирования системы понятий заключается в том, что эксперту дается задание выписать из руководств (книг по специальности) некоторые элементы, представляющие собой единицы смысловой информации.

Семантические отношения

Семантические отношения представляют собой взаимосвязи между понятиями. Известно около 200 базовых отношений, остальное многообразие является комбинацией базовых [7]. Примерами базовых отношений служат: отношения классификации (иметь имя, класс — подкласс, элемент — класс, часть — целое), признаковые или атрибутивные отношения (иметь признак или иметь значение признака), количественные отношения (иметь меру, иметь значение меры), отношения сравнения (больше — меньше, похож — не похож), временные отношения, пространственные отношения, каузальные отношения (причина — следствие, быть целью, быть мотивом), инструментальные отношения (служить для, быть инструментом, способствовать), модальные отношения (проблематичность, необходимость, возможность) и т. д.

В инженерии знаний разработаны определенные методы для выявления семантических отношений. Среди них выделяют две группы (рис. П.25): неформальные (осуществляются путем непосредственной работы с экспертами) и формальные (требующие применения формальных алгоритмов).

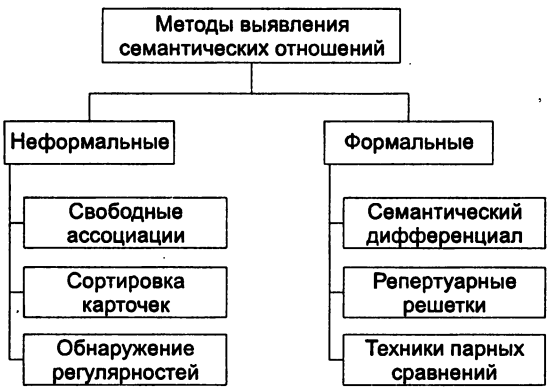


Рис. П.25. Методы выявления семантических отношений

Метод свободных ассоциаций

Эффект свободных ассоциаций заключается в следующем. Испытуемого просят отвечать на заданное слово первым пришедшим на ум словом. Как правило, реакция большинства испытуемых (если слова не были слишком необычными) оказывается одинаковой. Количество переходов в цепочке служит мерой «смыслового расстояния» между двумя понятиями. Многочисленные опыты подтверждают, что для двух любых слов (понятий) существует ассоциативная цепочка, состоящая не более чем из семи слов.

Метод свободных ассоциаций основан на психологическом эффекте, описанном выше. Эксперту предъявляется понятие с просьбой назвать как можно быстрее

первое пришедшее на ум понятие из сформированной ранее системы понятий. Далее производится анализ полученной информации.

«Сортировка карточек»

В методе «сортировка карточек» исходным материалом служат выписанные на карточке понятия. Применяют два варианта метода. В первом эксперту задают некоторые глобальные критерии предметной области, которыми он должен руководствоваться при раскладывании карточек на группы. Во втором случае, когда сформулировать глобальные критерии невозможно, эксперту дают задание разложить карточки на группы в соответствии с интуитивным пониманием семантической близости предъявляемых понятий.

«Обнаружение регулярностей»

Метод обнаружения регулярностей основан на положении, что элементы цепочки понятий, которые человек вспоминает с определенной регулярностью, имеют тесную ассоциативную взаимосвязь.

Для эксперимента произвольным образом отбирается 20 понятий. Эксперту предъявляется одно из них с просьбой построить ассоциативную цепочку из числа отобранных. Процедура повторяется 20 раз, причем каждый раз начальные понятия должны быть разными. Затем инженер по знаниям анализирует полученные цепочки с целью нахождения постоянно повторяющихся звеньев цепочек. Внутри выделенных таким образом группировок понятий устанавливаются ассоциативные взаимосвязи.

Метод семантического дифференциала

Метод семантического дифференциала разработан Ч. Осгудом [10]. Он широко используется в психологии и предназначен для измерения различий в интерпретации понятий испытуемыми. Исследуемый объект, в качестве которого может выступить слово, понятие, символ в вербальной или невербальной форме, оценивается путем соотнесения с одной из фиксированных точек шкалы, заданной полярными по значению признаками. Весь континуум шкалы разбивается, как правило, на 7 интервалов, и оцениваемый признак может принимать значения от –3 до +3.

Полученные на основании процедуры семантического дифференциала количественные данные изображаются в виде так называемого семантического профиля исследуемого понятия. Точность отражения понятия зависит от числа заданных осей. Вместе с тем, Ч. Осгудом показано, что от 50 до 65 % дисперсии результатов объясняется всего тремя факторами — фактором оценки, фактором силы и фактором активности.

Применяя технику семантического дифференциала для оценки множества объектов (понятий) одним экспертом или одного объекта множеством экспертов, на

выходе получают числовые таблицы вида «объект — признак». Эти таблицы затем подвергают многомерному анализу с целью выявления группировок как объектов, так и признаков. Проведенный анализ позволяет выявлять особенности понятийной структуры эксперта.

Техника репертуарных решеток

Техника репертуарных решеток предложена Г. Келли в 1955 году. Она направлена на изучение индивидуально-личностных конструктов, опосредующих восприятие и самовосприятие при анализе личностного смысла понятий [1].

По Г. Келли, «конструкт можно представить себе как референтную ось, основной параметр оценки... На поведенческом уровне его можно рассматривать как открытый человеком способ поведения...» [8]. Описание конструкта, как замечает Келли, удобнее всего проводить в биполярных понятиях. При этом конструкт становится тем, «чем два или несколько объектов сходны между собой и, следовательно, отличны от третьего объекта или нескольких других объектов». Биполярность конструктов дает возможность получить матрицу взаимоотношений между ними «конструкт — конструкт» и применить для выявления структуры смысловых параметров, лежащих в основе восприятия данным человеком объектов и отношений, алгоритмы анализа многомерных данных (факторный и кластерный анализ, неметрическое шкалирование и пр.).

Процедуру определения семантических отношений с использованием техники репертуарных решеток проводят в два этапа.

Сначала эксперту дают задание найти в собственном представлении существенные признаки, по которым два любых понятия из предметной области являются схожими и чем они оба отличаются от какого-либо третьего понятия. К найденным существенным признакам находят полярные по значению признаки. Так образуются индивидуальные конструкты.

На втором этапе, как и в методе семантического дифференциала, все понятия предметной области оцениваются экспертом по собственным выделенным конструктам. Считается, что расстояния между понятиями в пространстве индивидуальных конструктов отражают семантические отношения.

Техники парных сравнений

Техники парных сравнений применяют для изучения различных феноменов психического отражения. Они хороши тем, что не нуждаются в четкой привязке к каким-либо оценочным признакам. Эксперту требуется дать интуитивные оценки попарного сходства или различия понятий исследуемой предметной области. Семантическое пространство по результатам парного сравнения реконструируется с использованием аппарата многомерного метрического и неметрического шкалирования, получившего свое начало в работе [11].

Стратегии принятия решений

Выявление *стратегий принятия решений* — это самый сложный этап в построении модели предметной области. Частично стратегии принятия решений могут быть получены путем тщательного анализа семантических отношений. Стратегии экспертных рассуждений служат стержнем, на который «нанизываются «все предыдущие компоненты знаний. Как правило, стратегии отражают некоторые теоретические представления, фундаментальные законы или общепринятые классификации, известные в предметной области и используемые экспертом. В описательных науках проблема усложняется наличием различных теоретических представлений и классификаций, а опыт работы с экспертом показал, что эксперт сам не осознает, какой же конкретной парадигмой он пользуется при решении задач.

Основной принцип, которым должен руководствоваться инженер по знаниям при работе с экспертом, формулируется как принцип использования инструментов структурирования. Смысл этого принципа заключается в следующем. Как правило, взаимодействуя с экспертом, инженер по знаниям что-то старается писать. Бессистемные «заметки» иногда выливаются в многостраничные труды, из которых при дальнейшем анализе трудно что-либо понять. Можно, конечно, стенографировать или записывать на магнитофон экспертные рассуждения или объяснения какого-либо понятия. Но все равно это остается неструктурированным текстом. Инженерия знаний предлагает не писать тексты при взаимодействии с экспертом, а сразу же что-то рисовать, облекать экспертные знания и рассуждения в некоторые формы, которые должны быть понятны и эксперту. Такие формы и представляют собой инструменты структурирования. Существуют общепризнанные, стандартные инструменты, которые помогают, с одной стороны, найти общий язык с экспертом, а с другой — с большим успехом систематизировать или формализовать само взаимодействие. Любой инженер по знаниям может выдумать свои собственные инструменты, ниже рекомендуются те, которые хорошо зарекомендовали себя на практике.

Смысл использования инструментов заключается в том, что определенные элементы экспертных знаний на этапе первоначального сбора знаний представляются наглядно и иллюстрированно в такой форме, которая может быть легко модифицирована при последующей работе с экспертом, а впоследствии перекодирована в любую из моделей представления знаний, реализованную тем или иным программным инструментарием.

К стандартным инструментам (формам) структурирования относятся таблицы решений, деревья вывода, блок-схемы или структурные схемы, классификационные деревья, семантические сети, правила типа «если — то» или «условие — действие», диаграммы Венна и др.

Таблицы решений

Таблицы решений (decision table) представляют собой таблицы, в которых указаны действия, предпринимаемые в различных условиях, причем решение — это выбор между альтернативными действиями. Обычно таблица состоит из четырех частей, расположение и назначение которых показано в табл. П.15.

Таблица П.15. Таблица решений

Предварительные условия	Окончательные условия
Возможные действия	Предпринимаемые действия

В разделе «Предварительные условия» перечислены отдельные условия, от которых зависят предпринимаемые действия, а в разделе «Возможные действия» — действия, которые могут быть предприняты. В разделе «Окончательные условия» перечислены уточнения предусловий, при которых предпринимаются те или иные действия. Этот раздел организован в виде столбцов, в каждом из которых дается уточнение каждого предусловия. Затем в столбцах раздела «Предпринимаемые действия» проставляются крестики, указывающие на совершение того или иного действия. Обычно в таблице все возможные комбинации входных значений (предусловий) перечислены так, что применение таблицы всегда дает в точности одно предпринимаемое действие.

Таблица П.16 представляет собой пример таблицы решений при выборе способа, как добираться до работы. Символ «–» в разделе «Окончательные условия» означает «безразлично».

Таблица П.16. Пример таблицы решений

Дождь	НЕТ	НЕТ	ДА	–	–	–	–
Снег	–	НЕТ	НЕТ	ДА	–	–	–
Туман	–	НЕТ	НЕТ	НЕТ	НЕТ	ДА	ДА
Температура	–	>8	<8	–	–	>0	>0
Ехать на велосипеде	+						
Ехать на автомобиле		+	+				
Ехать поездом				+	+		
Остаться дома						+	+

Деревья вывода

Деревья вывода (decision tree) — это такие двоичные деревья, в которых каждый неконечный узел представляет решение. В зависимости от решения, принятого в таком узле, управление передается левому или правому (относительно этого узла) поддереву. Результатом принятия последовательности решений, представленных узлами, начиная с корня, является лист дерева.

Приведем пример дерева вывода. Предположим, что всех людей, обращающихся за удостоверением на право вождения автомобиля, можно описать с помощью следующих критериев: пол (м, ж), возраст (молодой, совершеннолетний, средний, пожилой), судимость (к суду не привлекался, привлекался по мелкому делу, совершил тяжкое преступление), водительские навыки (сдал экзамен, не сдал экзамен). Правило присвоения квалификации состоит в следующем: любое лицо, которое не совершило тяжкого преступления и сдало соответствующий экзамен, получает официальный статус водителя. Это же правило можно записать в виде дерева решений, представленного на рис. П.26.

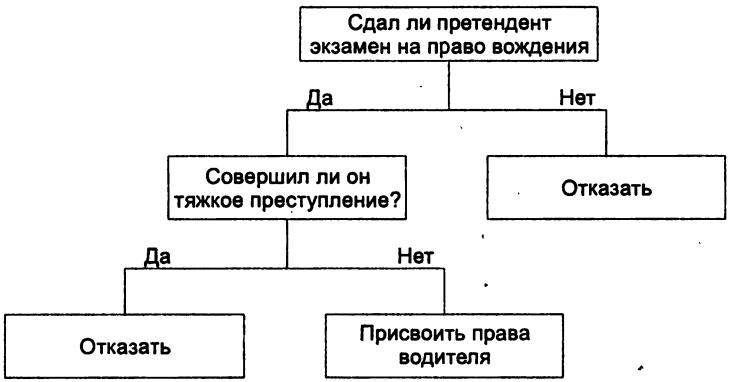


Рис. П.26. Дерево вывода

Блок-схемы

Блок-схемы, или *структурные схемы* (flowchart), — это подробное графическое представление структуры рассуждений, в котором упор сделан на логические взаимосвязи и осуществляемые при рассуждении элементарные операции, а не на используемые в ней информационные структуры. Состоит из множества блоков различной формы, соединенных совокупностью направленных связей. Связь показывает передачу управления, а форма блока характеризует особенности выполняемых действий и принимаемых решений. Для описания действий и логических операций внутри блоков применяется произвольная форма записи, типичными вариантами являются псевдокод и естественный язык.

Классификационные деревья

Классификационные деревья — это обычно деревья, отражающее ту или иную классификацию, общепринятую в данной предметной области. На рис. П.27. приведен пример классификации в области психологии познания.

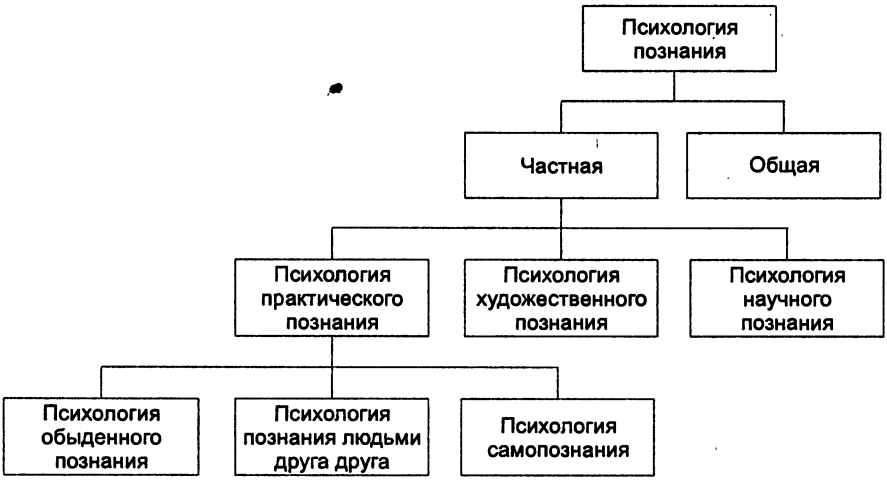


Рис. П.27. Пример классификации в области психологии познания

Семантические сети

Семантические сети, как указывалось выше (урок 1), — это ориентированный граф (вершины, связанные стрелками или дугами), вершины которого — понятия, а дуги — отношения между ними.

Примеры правил типа «если — то» или «условие — действие» также были приведены выше.

Диаграмма Венна

Диаграмма Венна (Venn diagram) — это схематические представление множества, впервые использованное Джоном Венном в XIX веке. Универсальное множество U обычно представляется прямоугольником, а подмножество S этого множества U — внутренней частью окружности (или какой-нибудь другой простой замкнутой кривой), полностью лежащей внутри этого прямоугольника.

Приведенные выше примеры инструментов структурирования, фактически, являются некоторым языком взаимодействия инженера по знаниям с экспертом. Процесс представления знаний — это процесс перевода знаний с языка структурирования на другой язык (или почти другой), который имеет вид моделей представления знаний, реализованных тем или иным программным продуктом. Этот процесс осуществляет инженер по знаниям совместно с разработчиком соответствующего инструментария.

В завершение изложения основ инженерии знаний приведем маленький, игрушечный пример, иллюстрирующий использование инструментов структурирования. Предположим, что происходит процесс взаимодействия инженера по знаниям с экспертом, который очень хорошо разбирается во фруктах.

Метод извлечения знаний — интервью. Ниже представлен пример такого интервью, которое проводит инженер по знаниям (ИнЗн) с экспертом (Экс).

ИнЗн: Расскажите мне, пожалуйста, как я могу распознавать фрукты.

Экс: Вы хотите, чтобы я рассказал вам о фруктах?

ИнЗн: Да, вот у меня есть несколько штук в сумке.

Экс (вынимая из сумки грейпфрут): Хорошо, вот это грейпфрут. Он круглый, имеет едкий запах, желтого цвета, на вкус — кислый, кожа — шероховатая, внутри у него зерна, для того чтобы его съесть, надо очистить.

ИнЗн: Таким образом, говоря о фруктах, Вы рассматриваете форму, запах, цвет, вкус, вид кожи, наличие зерен и способ употребления. Это так?

Экс: Да.

ИнЗн: Расскажите мне, пожалуйста, о других фруктах. Можно ли их рассматривать по тем же критериям?

Экс: Пожалуй, правильно, может быть, будет что-то еще, хотя я думаю, это полный набор.

ИнЗн: Тогда давайте нарисуем таблицу решений (табл. П.17).

Таблица П.17. Таблица решений для распознавания фруктов

	Атрибуты				
Форма	Круглый	Круглый	Круглый	Продолговатый	Продолговатый
Запах	Едкий	Едкий	Душистый	Душистый	Душистый
Цвет	Желтый	Оранжевый	Желтый	Желтый	Коричневый
Вкус	Кислый	Сладкий	Сладкий	Сладкий	Сладкий
Кожа	Шероховатая	Шероховатая	Гладкая	Гладкая	Гладкая
Зерна	Есть	Есть	Есть	Нет	Есть
Очищать	Надо	Надо	Не надо	Надо	Не надо
	Выходы				
Грейпфрут	+				
Апельсин		+			
Яблоко			+		
Банан				+	
Груша					+

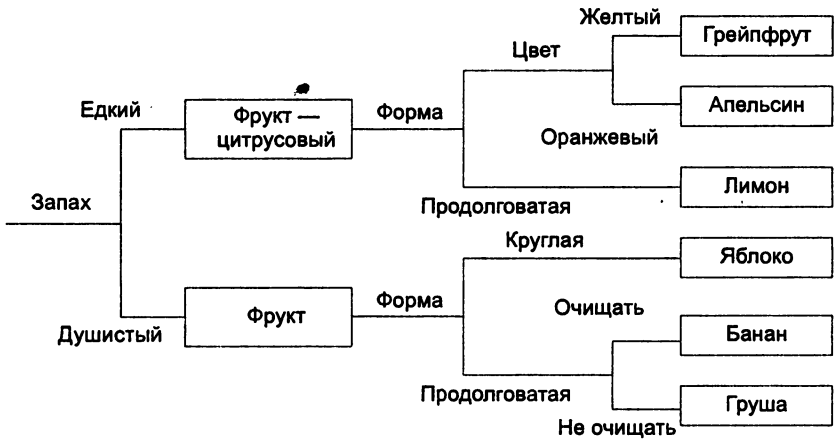


Рис. П.28. Дерево вывода для распознавания фруктов

Из этой таблицы очень легко нарисовать дерево вывода, изображенное на рис. П.28.

Анализируя таблицу решений и дерево вывода, можно записать полученные знания в виде правил «если — то».

Правила «если—то» для распознавания фруктов. Р1: ЕСЛИ (запах=едкий и форма=круглая и цвет=желтый)

ТО (фрукт=грейпфрут).

Иначе то же самое можно записать и так: Р2: ЕСЛИ (запах=едкий)

ТО (тип фрукта=цитрусовый); Р3: ЕСЛИ (тип фрукта=цитрусовый и форма=круглая и цвет=желтый) ТО (фрукт=грейпфрут).

Следует отметить, что правила типа «если — то» использованы только после таблицы решений и дерева вывода. Эта особенность применения инструментов распространяется не только на данный пример, но и справедлива в общем. В большинстве случаев с правил типа «если — то» лучше стараться не начинать собирать знания.

итература

1. Бурлачук Л. Ф., Морозов С. М. Словарь-справочник по психологической диагностике. — Киев: Наукова думка. — 1989.
2. Вассерман Л. И., Дюк В. А., Иовлев Б. В., Червинская К. Р. Психологическая диагностика и новые информационные технологии. — СПб.: СЛП, 1997.
3. Гаврилова Т. А., Червинская К. Р. Извлечение и структурирование знаний для экспертных систем. — М.: Радио и связь, 1992.
4. Гельфанд И. И., Розенфельд Б. И., Шифрин М. А. Структурная организация данных в задачах медицинской диагностики и прогнозирования//Вопросы

кибернетики. Задачи медицинской диагностики и прогнозирования с точки зрения врача. — М.: АН СССР, 1988. — С. 5–64.

5. Ноэль Э. Массовые опросы: Пер. с нем. — М.: Прогресс, 1978.
6. Пажитнов Л. А. Логическая структура компьютерной игры//Микропроцессорные средства и системы. — 1987. — № 3. — С. 11–13.
7. Поспелов Д. А. Ситуационное управление: теория и практика. — М.: Наука, 1986.
8. Франселла Ф., Баннистер Д. Новый метод исследования личности. — М.: Прогресс, 1987.
9. Червинская К. Р. Методы концептуального анализа знаний//Методы и системы принятия решений. Системы поддержки процессов проектирования на основе знаний. — Рига: Рижск. техн. ун-т, 1991. — С. 116–122.
10. Osgood Ch. E., Susi G. E., Tannenbaum P. N. The Measurement of Meaning. — Urbana: I 11, press, 1957.
11. Torgerson W. S. Multidimensional Scaling. Theory and Method//Psychometrika. 1952. Vol. 17. № 4.

Толковый словарь основных терминов интеллектуального анализа данных

analytical model (аналитическая модель)

Структура и процесс анализа базы данных. Например, моделью классификации набора данных является дерево решений.

См. также: classification, decision tree, exploratory data analysis, predictive model.

anomalous data (аномальные данные)

Данные, возникшие в результате ошибок или представляющие необычные события. Аномальными данными не следует пренебрегать, поскольку они могут нести важную информацию.

См. также: data cleansing, data clearing and standardization.

artificial neural networks (искусственные нейронные сети)

См. neural networks.

CART, classification and regression trees (деревья классификации и регрессии)

Один из методов построения деревьев решений, используемых в целях классификации данных. Предлагает набор правил, применяемых к новой (неклассифицированной) совокупности данных для предсказания того, какие последовательности имеют заданный исход. Наборы данных сегментируются с помощью разбиения на две части. Требует менее тщательной предварительной подготовки данных, чем метод CHAID.

См. также: CHAID, classification, decision tree.

CHAID, chi square automatic interaction detection (автоматическое выявление зависимости по критерию хи-квадрат)

Метод построения деревьев решений, используемый в целях классификации данных. Предлагает набор правил, которые можно применять к новой совокупности данных. Сегментирует данные на несколько частей с помощью критерия хи-квадрат. Требует специальной предварительной подготовки данных.

См. также: CART, classification, decision tree.

classification (классификация)

Процесс разбиения набора данных на непересекающиеся группы, обеспечивающий максимальную «близость» элементов одной группы и максимальное различие («удаление») групп. Принципиальной особенностью методов классификации (в узком смысле слова) является наличие априорной информации о структуре и/или статистических свойствах анализируемых данных, что позволяет, например, при отнесении очередного объекта к конкретному классу руководствоваться статистическими решающими правилами. Нередко для настройки алгоритма используется обучающая выборка (так называемая классификация с обучением), разбиение которой на классы предполагается известным.

См. также: clustering, decision tree.

clustering (кластеризация)

Процесс разбиения данных, как правило, на непересекающиеся классы. Существенное отличие от методов классификации состоит в отсутствии какой-либо предварительной информации о свойствах исследуемых данных. К настоящему времени предложено множество алгоритмов кластеризации, результаты применения которых к одним и тем же данным могут различаться. Основной недостаток методов кластеризации заключается в невозможности строгого обоснования статистической достоверности полученного разбиения.

См. также: classification, decision tree, predictive model.

data cleansing (очистка данных)

Процесс, обеспечивающий согласованность и корректную запись всех значений в наборе данных.

См. также: data clearing and standardization.

data clearing and standardization (очистка и стандартизация данных)

Один из методов, применяемых при построении информационного хранилища и последующего добавления к нему данных. Под очисткой и стандартизацией понимаются такие операции, как идентификация и объединение повторных

(дублирующихся) записей, стандартизация сокращений и коррекция полей, имеющих разную длину.

См. также: data cleansing, data warehouse.

data mart (информационная «витрина»)

Подмножество хранилища данных, выделенное для отдельного подразделения компании или для выполнения конкретной функции. Таким образом, это просто «облегченный» вариант хранилища данных уровня подразделения.

См. также: data warehouse.

data mining (интеллектуальный анализ данных)

Процесс обнаружения значимых корреляций, зависимостей и тенденций в результате анализа содержимого информационных хранилищ с применением методов распознавания и выявления ассоциаций (аналогичных последовательностей, кластеров) данных. Часто для поиска зависимостей в накопленных данных (информационном хранилище) используется параллельная обработка. Применяемые интеллектуальные методы выделения и извлечения информации позволяют исследовать намного более широкий спектр возможностей, чем самая сложная совокупность запросов. В результате удается выявить шаблоны (систематизированные структуры) данных и вывести из них правила для принятия решений и прогнозирования их последствий (например, путем предсказания значений непрерывных переменных). Интеллектуальные средства интерпретации и представления данных способны также ускорять анализ за счет выделения наиболее важных переменных. Они используют четыре основных инструмента: нейронные сети, деревья решений, индуктивные правила и визуализацию данных. Иногда применяется комбинация этих методов.

См. также: classification, clustering, data visualization, decision tree, linear regression, neural networks, rule induction.

data modelling software (программное обеспечение моделирования данных)

Позволяет разработчикам определять, с какими данными они имеют дело, что эти данные означают, как соотносятся с другими данными и кто их использует. Подобное ПО обычно применяется в системах разработки приложений, а иногда и при создании хранилищ данных.

См. также: analytical model, data warehouse.

data navigation, database navigation (перемещение в БД)

Процесс просмотра различных измерений, «срезов» и уровней детализации в многомерной базе данных.

См. также: MDDBMS, OLAP.

data visualization (визуализация данных)

Визуальная интерпретация сложных взаимосвязей в многомерных данных.

См. также: data mining, MDDBMS.

data warehouse (информационное хранилище, хранилище данных)

Система хранения данных большого объема (до нескольких терабайт) и извлечения информации, реализуемая на основе баз данных различных типов (от плоских файлов и реляционных БД до патентованных решений). Нередко для организации хранилищ данных применяются многомерные СУБД (MDDBMS), хотя многомерный анализ может выполняться и в рамках реляционного механизма. Процесс построения информационного хранилища сводится к объединению информации из многих рабочих БД в единый информационный массив большого объема, обеспечивающий эффективную структуру для анализа, интерпретации и представления данных (data mining). При организации хранилища информация проходит процесс проверки (согласования формата данных) и очистки (удаления ошибок и сведения к минимуму числа пропущенных значений). Для уточнения стратегии формирования хранилища и оценки возможного влияния на результат ошибок, имеющихся в исходных данных, как правило, создается предварительная мини-модель. При добавлении данных к хранилищу обычно контролируется их качество. Источником данных часто служат OLTP-системы. Для построения эффективного информационного хранилища применяется ПО семи категорий: инструментальные средства моделирования данных, хранилище метаданных, центральная БД, программы транспортировки данных, инструменты их извлечения, очистки и сортировки, связующее ПО (обеспечивающее совместимость данных разных типов) и пользовательские приложения доступа к данным. Их дополняют средства управления, тиражирования и синхронизации БД, продукты разработки приложений и другое ПО.

См. также: data cleansing, data clearing and standardization, data mart, data mining, data modeling software, MDDBMS, metadata, OLTP.

decision tree (дерево решений)

Древовидная структура, представляющая совокупность решений. Деревья решений разбивают данные на группы на основе значений переменных, в результате чего возникает иерархия операторов «ЕСЛИ — ТО», которые классифицируют данные. Позволяют решить многие задачи быстрее, чем нейронные сети.

См. также: analytical model, CART, CHAID, data mining, rule induction, neural networks.

dimension (измерение)

В реляционных БД измерение представляет каждое поле. В многомерной БД измерение — это набор аналогичных значений.

См. также: MDDBMS.

exploratory data analysis (разведочный анализ данных)

Применение графических и описательных статистических методов для изучения и исследования структуры набора данных.

См. также: analytical model, classification, clustering, prospective data analysis.

genetic algorithms (генетические алгоритмы)

Алгоритмы, использующие методы «цифрового дарвинизма». Процедуры оптимизации, имитирующие при проектировании модели данных такие процессы, как генетическая рекомбинация, мутация и отбор, аналогичные тем, что обусловливают естественную эволюцию.

См. также: analytical model, exploratory data analysis.

linear regression (линейная регрессия)

Статистический метод, применяемый в целях отыскания наилучшего линейного приближения для зависимости целевой переменной от независимых переменных по имеющимся значениям.

См. также: logistic regression, predictive model.

logistic regression (логистическая регрессия)

Линейная регрессия, предсказывающая доли категоризированной целевой переменной в общей совокупности.

См. также: linear regression, predictive model.

MDA, multidimensional analysis (многомерный анализ данных)

Анализ содержимого БД при помощи формирования сложных запросов. Для получения результатов используются методы оперативной аналитической обработки (OLAP) и агрегация значений. Инструменты MDA работают с информацией БД, но в основном используют данные, специально организованные в виде многомерных гиперкубов, что позволяет представить информацию в любом разрезе. Они лучше всего подходят для интерактивного исследования данных и предлагают графический интерфейс, упрощающий работу пользователей.

См. также: dimension, MDDBMS, OLAP, query-and-reporting tools.

MDDBMS, multidimensional database management system (многомерная СУБД)

Многомерные базы данных, по существу, транслируют информационное содержимое в *N*-мерный «куб» (гиперкуб), где каждая ось соответствует измерению. При этом выполняется предварительная обработка многомерных представлений данных, которые хранятся как части *N*-мерного куба. Когда приложение вызывает одно из многомерных представлений данных, оно считывается непосредственно, а не путем просмотра обширных двухмерных таблиц, как это обычно происходит при

обработке SQL-запросов к реляционной СУБД. При работе с многомерными БД, как правило, применяется собственный инструментарий составления отчетов, формирования запросов и анализа. Между тем хранение множества представлений может значительно увеличить объем многомерной БД. В этом случае лучше обратиться к методам ROLAP.

См. также: data warehouse, dimension, MDA, OLAP, ROLAP.

metadata (метаданные, «данные о данных»)

Объединяют отдельные блоки информационного хранилища, превращая его в одно работоспособное целое. Это информация об исходных данных: что они означают, где хранятся и как их отыскать. Метаданные описывают также содержимое информационного хранилища: источник данных, трансляцию, агрегацию, табличные перекодировки и другие преобразования, выполненные в процессе его создания. Метаданные позволяют получить детальные сведения о конкретной группе данных или выявить ошибки в процессе организации хранилища. Они используются моделирующими программами (для выявления взаимосвязи между данными), программами извлечения и транспортировки (для поиска правильной информации и помещения ее в нужное место хранилища), инструментами очистки, стандартизации и объединения данных. Кроме того, благодаря семантическому уровню, построенному на основе метаданных, инструменты доступа к информации позволяют скрыть от конечного пользователя сложность ее структуры. В настоящее время предпринимаются усилия по созданию стандарта обмена метаданными, спецификация которого уже разработана.

См. также: data warehouse.

neural networks (нейронные сети)

Один из методов анализа данных. Реализует модель, представляющую собой совокупность связанных друг с другом узлов, для каждого из которых определены видимые входы, выходы и скрытая обработка. Нейронная сеть может обучаться на специальном наборе данных. Для такого набора совокупность входных значений порождает известное множество выходных, что позволяет итерационным путем определить обработку. Таким образом, для построения скрытого уровня логики нейронная сеть применяет правила, выводимые ею из зависимостей данных. Затем скрытый уровень обрабатывает входные значения, классифицируя их на основе «опыта» модели. Результирующая нелинейная модель не имеет четкой интерпретации и применяется независимо от того, на основе каких логических выводов получаются результаты. Термин возник из аналогии со структурой нервной системы организмов.

См. также: decision tree, predictive model, rule induction.

OLAP, on-line analytical processing (оперативная аналитическая обработка данных)

В отличие от обычного инструментария на основе запросов средства OLAP используют не табличную, а многомерную модель данных. Данные рассматриваются не как отдельные события, а как совокупный результат событий за некий период

времени. OLAP играет важную роль в системах принятия решений и инструментальных средствах многомерного анализа данных (MDA). Методы OLAP хорошо работают в рекурсивных вычислениях с массивами данных объемом до 20 Гбайт, содержащими до 15 переменных (измерений). Продукты OLAP дополняют другие инструменты интеллектуального анализа данных и многомерных СУБД. Они дают возможность пользователям просматривать и анализировать информацию, манипулировать ею и перемещаться по массивам данных.

См. также: MDA, MDDBMS, query-and-reporting tools, time series analysis.

OLTP system (система оперативной обработки транзакций)

Подобные системы применяются как инструмент анализа данных и обычно ориентированы на узкоспециализированную задачу. В отличие от хранилищ данных накапливаемая в OLTP-системах информация часто модифицируется. OLTP-таблицы детализированы, а для данных характерна высокая степень упорядоченности. OLTP-системы нередко являются источниками информации для хранилищ данных.

См. также: data warehouse, ROLAP.

predictive model (модель с предсказанием)

Модель, позволяющая предсказывать значения указанных переменных в наборе данных.

См. также: analytical model, prospective data analysis.

prospective data analysis (анализ тенденций)

Анализ данных, предсказывающий будущие тенденции и события на основе накопленных данных.

См. также: predictive model.

query-and-reporting tools (инструментарий формирования запросов и вывода отчетов)

Такие инструментальные средства часто дополняются графическим интерфейсом и работают со структурированными БД. Между тем диапазон подобных программ довольно широк — от электронных таблиц на ПК до приложений крупных СУБД в архитектуре клиент/сервер, а некоторые из них дополняются функциями многомерного анализа данных. Они используются обычно для интерактивного исследования данных (особенно реляционных), например, в целях проверки гипотез. В большинстве подобных инструментов применяется язык SQL. Если данные распределены по многим таблицам или нескольким слабо индексированным базам данных, время выполнения запроса чрезмерно увеличивается, а неправильно построенный запрос способен вызвать перегрузку системы. Для доступа к данным в информационных хранилищах иногда применяются специализированные средства, не требующие от пользователей знания структуры данных или языка SQL.

См. также: data warehouse, MDA, metadata, ROLAP.

ROLAP, relational on-line analytical processing (оперативная аналитическая обработка реляционных данных)

В отличие от многомерных СУБД инструментальные средства ROLAP не используют представления данных в виде гиперкуба, а сочетают гибкие и мощные инструменты обработки запросов, которые передаются РСУБД, с оптимизацией существующих реляционных БД. Полученная с помощью подобных методов «многоуровневая» архитектура в среде клиент/сервер предлагает клиенту ROLAP многомерное представление данных. При этом на сервере ROLAP поддерживаются механизм вычислений, метаданные, обеспечивается блокировка и защита. Выбор между MDDBMS и ROLAP зависит от частоты изменения фундаментальной модели данных. В MDDBMS любая ее модификация требует перекомпиляции гиперкуба, но дает значительный выигрыш в производительности.

См. также: MDDBMS, metadata, OLAP, query-and-reporting tools.

rule induction (индукция правил)

Выделение полезных правил типа «ЕСЛИ — ТО» (IF — THEN), исходя из статистической значимости данных. Такой метод анализа данных предусматривает создание неиерархического набора условий, которые могут перекрываться. Часто индукция правил выполняется путем генерации частичных деревьев решений, а для того чтобы выбрать, какое из них применить к входным данным, используются статистические методы.

См. также: data mining, decision tree, neural networks.

time series analysis (анализ временных рядов)

Анализ последовательности значений, полученных через заданные интервалы времени.

См. также: data mining, dimension.

Алфавитный указатель