

СРЕДНЕЕ  
ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАНИЕ

# БОЛЬШИЕ ДАННЫЕ

## Big DATA

А. В. Макшанов  
А. Е. Журавлев  
Л. Н. Тындыкарь



E.LANBOOK.COM

**А. В. МАКШАНОВ,  
А. Е. ЖУРАВЛЕВ,  
Л. Н. ТЫНДЫКАРЬ**

# **БОЛЬШИЕ ДАННЫЕ. BIG DATA**

*Учебник*



**ЛАНЬ**

**САНКТ-ПЕТЕРБУРГ  
МОСКВА  
КРАСНОДАР  
2021**

УДК 004

ББК 32.973-018.2я723

**М 17** Макшанов А. В. Большие данные. Big Data : учебник для СПО / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. — Санкт-Петербург : Лань, 2021. — 188 с. : ил. — Текст : непосредственный.

**ISBN 978-5-8114-6811-9**

В представленном учебнике рассматриваются базовые аспекты профессиональной части дисциплин, непосредственно связанных с технологиями работы с большими данными, например, «Компьютерный анализ», «Большие данные», «Слияние данных» и т. п. профессионального учебного цикла по специальностям среднего профессионального образования «Прикладная математика и информатика», «Информационные системы» и «Организация и технология защиты информации».

Рассмотрены основные аспекты работы с большими данными, методы и технологии «Big Data» и «Data Mining», а также общие приемы интеллектуального анализа данных. В качестве инструментальной среды разработки используется интегрированный пакет MatLab версий 6.5 и выше.

УДК 004

ББК 32.973-018.2я723

**Обложка**  
**П. И. ПОЛЯКОВА**

© Издательство «Лань», 2021  
© Коллектив авторов, 2021  
© Издательство «Лань»,  
художественное оформление, 2021

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1. ПАРАДИГМА МАШИННОГО ОБУЧЕНИЯ В ИНТЕЛЛЕКТУАЛЬНОМ АНАЛИЗЕ ДАННЫХ.....	7
1.1. Машинное обучение.....	7
1.2. Нейробиологическое направление в ИИ.....	7
1.3. Нейросети.....	9
1.4. Эволюционное моделирование как исследовательский метод.....	12
1.5. Генетические алгоритмы.....	14
1.6. Ансамблевые методы: джекнайф и бутстрэп.....	17
1.7. Ансамблевые методы: бэггинг, бустинг, стекинг.....	18
2. ИЗВЛЕЧЕНИЕ ЗНАНИЙ.....	22
2.1. Виды знаний и способы их представления.....	22
2.2. Модели представления знаний.....	22
2.3. Извлечение знаний.....	23
2.4. Некоторые подходы к интеллектуальному анализу данных.....	29
2.5. Формирование знаний методами локальных геометрий.....	34
3. ИММУНОКОМПЬЮТИНГ.....	37
3.1. Вычислительная процедура сингулярного разложения матриц.....	40
3.2. Распознавание в пространстве проекций.....	41
3.3. Формирование индексов риска.....	42
3.4. Алгоритм формирования электронной цифровой подписи.....	44
4. КЛАСТЕРНЫЙ АНАЛИЗ.....	47
4.1. Кластеризация. Выбор метрики.....	47
4.2. Метод $k$ средних и $EM$ -алгоритм.....	47
4.3. Иерархическая кластеризация на основе дендрограммы.....	49
4.4. Оценка качества разделения.....	50
4.5. Кластер-анализ.....	50
4.6. Снижение размерности за счет выделения компонент.....	52
5. ПРОГНОЗНАЯ АНАЛИТИКА.....	53
5.1. Прогнозирование.....	53
5.2. Классификация методов прогнозирования.....	53
5.3. Временные ряды.....	54
5.4. Множественная регрессия.....	56
5.5. Адаптивная модель множественной регрессии.....	58
5.6. Прогнозирование МВР.....	60
5.7. Прогнозирование МВР в пространстве проекций.....	62
5.8. Анализ сингулярных спектров.....	62
5.9. Прецедентный анализ.....	65

6. СЛИЯНИЕ ДАННЫХ.....	67
6.1. Проблемы. Оценивание в условиях неопределенности.....	67
6.2. Комплексирование координатной оценки и оценки пеленга.....	68
6.3. Байесовское слияние.....	71
6.4. Примеры комплексирования данных.....	73
7. МАШИНЫ ОПОРНЫХ ВЕКТОРОВ.....	78
7.1. Постановка задачи.....	78
7.2. Идея метода опорных векторов.....	78
7.3. Разделение полосой на плоскости.....	79
7.4. Случай отсутствия линейной отделимости.....	83
7.5. Развитие метода.....	84
7.6. Регрессионный анализ на базе метода опорных векторов.....	86
8. НЕЙРОМАТЕМАТИКА.....	88
8.1. Пример: персептрон Розенблатта.....	88
8.2. Краткий исторический обзор.....	94
8.3. Архитектура нейронных сетей.....	95
8.4. Области применения нейронных сетей.....	100
9. НЕЙРОННЫЕ СЕТИ.....	102
9.1. Распространение ошибок.....	103
9.2. Многослойные сети. Некоторые архитектуры сетей.....	105
9.3. Функции создания нейронных сетей в ИМС MatLab.....	109
9.4. Примеры создания и использования нейронных сетей.....	110
10. ЭВОЛЮЦИОННОЕ МОДЕЛИРОВАНИЕ И ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ.....	115
10.1. Эволюционное моделирование.....	115
10.2. Модели возникновения МГИС.....	116
10.3. Применение в задачах функциональной оптимизации.....	117
10.4. ЭМ как исследовательский метод в информатике.....	117
10.5. Генетические алгоритмы.....	119
10.6. Естественный отбор в природе.....	120
10.7. Что такое генетический алгоритм.....	123
10.8. Особенности генетических алгоритмов.....	128
11. ВЗАИМОДЕЙСТВИЕ СФЕР МАШИННОГО ОБУЧЕНИЯ.....	133
11.1. Задачи нейросетевой математики.....	133
11.2. Алгоритмы обучения сети.....	133
11.3. Области применения нейронных сетей.....	136
11.4. Взаимодействие различных областей.....	137
11.5. ANFIS: функциональный эквивалент нечеткой модели.....	138
11.6. Нейронные сети и эволюционное моделирование.....	139
11.7. Искусственные нейронные сети и экспертные системы.....	141
11.8. Соображения надежности.....	142

<b>12. КОГНИТИВНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРОБЛЕМНЫХ СИТУАЦИЙ</b> .....	143
12.1. Ситуационный анализ на основе когнитивных карт .....	143
12.2. Обеспечение целенаправленного поведения .....	145
12.3. Методика когнитивного анализа сложных ситуаций.....	146
12.4. Построение когнитивной модели .....	147
12.5. Моделирование.....	149
12.6. Внешняя среда.....	149
12.7. Нестабильность внешней среды .....	150
12.8. Слабоструктурированность внешней среды .....	150
12.9. Общее понятие когнитивного анализа.....	151
12.10. Механизмы реализации частных задач.....	153
12.11. Виды факторов .....	154
12.12. Выявление факторов (элементов системы) .....	156
12.13. Два подхода к выявлению связей между факторами .....	157
12.14. Проблема определения силы воздействия факторов .....	158
12.15. Проверка адекватности модели .....	159
12.16. Применение когнитивных моделей в СППР .....	159
12.17. Компьютерные СППР.....	162
<b>13. НОВЫЕ ПРОБЛЕМЫ БОЛЬШИХ ДАННЫХ И ПРИМЕРЫ</b> .....	165
13.1. Примеры успешных применений аналитики БД .....	165
13.2. Новые проблемы, обусловленные особенностями БД.....	166
13.3. Накопление ошибок .....	167
13.4. Возникновение ложных выборочных корреляций .....	171
13.5. Зависимости между помехой и переменными модели.....	172
13.6. Некоторые возможные решения ключевых проблем.....	173
<b>ЗАКЛЮЧЕНИЕ</b> .....	180
<b>СПИСОК ЛИТЕРАТУРЫ</b> .....	181

## ВВЕДЕНИЕ

В западной традиции науковедения одним из самых влиятельных трактатов признается книга Томаса Куна «Структура научных революций», вышедшая в 1962 г. Краеугольным камнем теории, предложенной Куном, является его концепция *парадигмы*, понимаемой как набор процедур или идей, косвенно инструктирующих ученых в том, чему верить и как работать. Научная революция трактуется Куном как процесс «*смещения парадигмы*».

В том же 1962 г. Институт перспективных исследований НАТО издал четырехтомный трактат Дж. Тьюки «Анализ данных», в котором была представлена научному сообществу парадигма интеллектуального анализа данных (ИАД). Дж. Тьюки в 1970–1980-х гг. имел славу одного из самых влиятельных самого высокооплачиваемого математика в мире с зарплатой около \$300 000 в год (примерно 60 тыс. долларов в месяц по современному курсу). К сожалению, его труды не отличались какими-либо литературными достоинствами, его идеи приобрели широкую популярность только когда стали появляться книги, написанные им в соавторстве Маклауфлином, Мостеллером и др.

Одно из важных направлений в Анализе данных Дж. Тьюки состоит в оптимизации моделей многомерного статистического анализа, основанных на многомерном нормальном законе. Основные принципы анализа данных Дж. Тьюки:

- принцип многократного возвращения к одним и тем же данным;
- принцип множественности возможных моделей;
- принцип варьирования предпосылок с рассмотрением последствий такого варьирования;
- принцип многовариантных нелинейных преобразований данных;
- принцип множественности результатов и выбора на основе неформальных процедур принятия решений;
- принцип полного использования эндогенной информации и максимального учета информации экзогенной.

В одном из своих интервью Томас Кун привел анализ данных Тьюки в качестве новой парадигмы, приходящей на смену традиционной статистике. Анализ данных – вычислительно затратный подход, он смог себя проявить только благодаря взрывному развитию возможностей компьютерной техники.

# 1. ПАРАДИГМА МАШИННОГО ОБУЧЕНИЯ В ИНТЕЛЛЕКТУАЛЬНОМ АНАЛИЗЕ ДАННЫХ

## 1.1. Машинное обучение

*Новой парадигмой* в интеллектуальном анализе данных стал подход, моделирующий процесс получения результата с минимальными претензиями на понимание механизма – *машинное обучение*, в значительной степени основанный на идеях Джона Тьюки.

*Машинное обучение* – это построение на основании известных данных моделей, которые предсказывают неизвестное. Основные задачи машинного обучения:

- *регрессия* (предсказание числовых значений признаков, например, предсказание будущих объемов продаж на основании известных данных о продажах в прошлом);
- *классификация* (предсказание того, к какому из известных классов относится объект, например, предсказание того, вернет ли заемщик кредит, на основании данных о том, как возвращали кредиты заемщики этого типа в прошлом);
- *кластеризация* (разделение большого множества объектов на кластеры – классы, внутри которых объекты похожи между собой, например, сегментирование рынка, разделение всех потребителей на классы так, что внутри классов потребители похожи между собой, а в разных классах – отличаются);
- *поиск аномалий* (поиск редких и необычных объектов, существенно отличающихся от основной массы, например, поиск мошеннических транзакций).

## 1.2. Нейробиологическое направление в ИИ

В длинном списке классов моделей и методов в области искусственного интеллекта особо выделяется нейробиологическое направление.

До недавнего времени главные результаты в области ИИ были достигнуты в рамках символического направления на базе традиционных вычислительных



истем последовательного действия. С математической точки зрения основу таких технологий можно свести к одному из двух формальных подходов, *логическому и лингвистическому*. Логический подход к ИИ основан на автоматизации логического вывода, а лингвистический подход основан на формальных грамматиках, введенных Н. Хомским (ок. 1960 г.) в результате попыток построения математических моделей естественного языка. Вообще говоря, сама логика возникла для выделения и описания глубинных структур, скрытых за поверхностными лингвистическими конструкциями.

В рамках символического направления ИИ создан широкий набор технологий манипулирования знаниями, обозначаемый специальным термином *инженерия знаний (knowledge engineering)*. Использование результатов этого направления в областях, где существуют строгие правила и где современные компьютеры превосходят возможности мозга (вычислительные задачи, перебор вариантов и др.), может быть исключительно эффективным. Кроме того, наблюдается возрастающий интерес к решению проблем распределенного интеллекта, связанного с многоагентными системами.

Биологическое направление ИИ можно определить как попытки моделировать биологические механизмы мышления для их лучшего понимания и реализации в технических устройствах. Наиболее развитыми областями этого направления являются:

- искусственные нейронные сети (ИНС);
- генетические алгоритмы (ГА);
- эволюционное моделирование;
- формальные иммунные системы (ФИС) или иммунокомпьютинг (ИК);
- нечеткая логика.

Программное обеспечение, написанное с применением этих алгоритмов, применяется во всех областях техники и технологии. Многие высокотехнологичные компании в настоящее время занимаются созданием искусственного интеллекта. Практически во всей сложной бытовой технике используется тот или иной подход.

- *Алгоритмы нечеткой логики* применяются в японских стиральных машинах типа автомат. Нечеткая логика в Японии де факто стала стандартом.
- *Нейронные сети* обычно используются там, где необходимо обучить объект перед использованием. Часто это элементы распознавания речи. Компания Apple в своей продукции использует такой подход.
- *Эволюционные алгоритмы* используются для поиска оптимальных вариантов. Часто используются для настройки нейронных сетей.
- *Иммунокомпьютинг* – целенаправленное проектирование и снижение размерности. Рассматривают как часть эволюционного моделирования.
- *Генетические алгоритмы* (Случайный лес, *Reinforcement Learning* и т. п.).

Сегодня область искусственного интеллекта содержит в себе в первую очередь взаимодействие нейронных сетей, эволюционного программирования и нечеткой логики. Включение концепции нечеткой логики в ИС дает возможность гибридной системе иметь дело с человекоподобным процессом рассуждений, закладывать в информационное поле ИС априорный опыт экспертов-экономистов, использовать нечеткое представление информации, извлекать знания из входного потока показателей.

### 1.3. Нейросети

Если говорить простыми словами, то нейросеть – это программа, которая, предположительно, работает аналогично нашему мозгу. В человеческом мозге все сигналы передаются нейронами, а процесс обучения представляет собой повторную активацию уже имеющихся нейронных связей. Чем чаще повторяются обращения к конкретной нейронной связи, тем более плотными становится эта связь, и увеличивается вероятность её вывода при получении той же самой вводной информации. Можно рассматривать нейросети как *взвешенный граф*, где нейроны являются узлами, а процесс обучения представляет собой адаптивную настройку весов рёбер.

В качестве самого нейрона можно представить некую нелинейную функцию, называемую *активационной функцией*, задача которой – представить совокупность поступивших в нее сигналов в виде одного результирующего значения, которое потребуется уже дальше для нейронов следующего слоя, если таковой имеется.

Несмотря на примитивность в сравнении с биологическими системами, ИНС обладают рядом полезных свойств и способны решать очень важные задачи. К таким свойствам в первую очередь относятся:

1. **Обучаемость.** Выбрав одну из архитектур НС и свойства нейронов, а также проведя алгоритм обучения, можно обучить сеть решению задачи, которая ей по силам. Нет гарантий, то это удастся сделать всегда, но во многих случаях обучение бывает успешным.
2. **Способность к обобщению.** После обучения сеть становится нечувствительной к малым изменениям входных сигналов (шуму или вариациям входных образов) и даёт правильный результат на выходе.
3. **Способность к абстрагированию.** Если предъявить сети несколько искажённых вариантов входного образа, то сеть сама может создать на выходе идеальный образ, с которым она никогда не встречалась.

Среди решаемых задач следует выделить:

- распознавание образов (например, зрительных или слуховых);
- реализацию ассоциативной памяти;
- кластеризацию;
- аппроксимацию функций;
- прогнозирование временных рядов;
- управление;
- принятие решений;
- диагностику.

Многие из этих задач сводятся к следующей постановке. Требуется построить такое отображение  $X \leftarrow Y$ , чтобы на каждый возможный сигнал  $X$  формировался правильный выходной сигнал  $Y$ . Отображение задаётся конечным

числом пар (Вход – Известный выход). Число этих пар (обучающих примеров) существенно меньше общего числа возможных сочетаний значений входных и выходных сигналов. Совокупность всех обучающих примеров носит название *обучающей выборки*.

Например, в задачах распознавания образов  $X$  – некоторое представление образа (матрица, вектор признаков),  $Y$  – номер класса, к которому принадлежит входной образ. В задачах управления  $X$  – набор контролируемых параметров управляемого объекта, – код, определяющий управляющее воздействие, соответствующее текущим значениям управляющих параметров. В задачах прогнозирования в качестве входных сигналов используются значения наблюдаемой величины до текущего момента времени, на выходе – следующие во времени значения.

Эти и вообще большая часть прикладных задач может быть сведена к построению некоторой многомерной функции. Каковы при этом возможности нейросетей, которые вычисляют линейные и нелинейные функции одного переменного, а также всевозможные композиции, функции от функций, получаемые при каскадном соединении нейронов? Что можно получить, используя такие операции?

В результате многолетней научной полемики между Колмогоровым и Арнольдом была доказана возможность точного представления непрерывных функций нескольких переменных в виде композиции непрерывных функций одного переменного и сложения. Наиболее полно на вопрос об аппроксимационных возможностях нейронных сетей отвечает обобщенная теорема Стоуна, которая утверждает универсальные аппроксимационные возможности произвольной нелинейности: с помощью линейных операций и каскадного соединения можно на базе произвольного нелинейного элемента получить устройство, вычисляющее любую непрерывную функцию с любой заданной точностью. Таким образом, нейроны в сети могут использовать практически любую нелинейную функцию активации, важен лишь факт её нелинейности.

Искусственные нейросети имитируют поведение мозга, т. е. их тоже можно обучить. Есть два вида обучения нейросети – *контролируемое и неконтролируемое*.

Контролируемое – это, например, спам-фильтр. В этом случае системе даётся исходная информация (письма, приходящие на почту) и входная информация (список слов, по которым нужно отфильтровать письма).

При неконтролируемом обучении система должна «сама» понять структуру входной информации и определить, как с ней дальше работать.

Для того чтобы нейросеть работала, ее нужно обучить на *дэйтасете*. Обучение нейросети состоит из двух основных этапов, которые повторяются много раз:

- попытка модели выдать результат (оно же прямое распространение, или *forward propagation*);
- корректировка работы модели, наложение штрафа (оно же обратное распространение, *backward propagation*).

Если модель уже обучена, для ее работы требуется только первый этап, который будет выполняться на новых данных, для которых мы хотим получить результат.

При обучении эти два этапа повторяются столько раз, сколько потребуется, чтобы модель дала удовлетворительный результат. Как правило, для обучения используют не по одному объекту из дэйтасета для выполнения прямого и обратного прохода, а группу размером в несколько объектов, которая называется *батч*. Размер батча можно настраивать, и за один проход модель учитывает характеристики всех объектов батча для обучения. Один повтор прямого и обратного распространения по всему дэйтасету называется *эпохой*. Зачастую таких итераций требуется не одна сотня. В этом обучение нейросети похоже на обучение людей.

#### 4. Эволюционное моделирование как исследовательский метод

Эволюционное моделирование, ЭМ (*Evolutionary computation*):

1. Использует признаки теории Дарвина для построения интеллектуальных систем (методы группового учёта, генетические алгоритмы). Является частью более обширной области искусственного интеллекта – вычислительного интеллекта.

2. Направление в математическом моделировании, объединяющее компьютерные методы моделирования эволюции, а также близкородственные по источнику заимствования идеи (теоретическая биология), другие направления в эвристическом программировании. Включает в себя, такие разделы, как генетические алгоритмы, эволюционные стратегии, эволюционное программирование, искусственные нейронные сети, нечеткую логику.

Поскольку эволюция, по-видимому, представляет собой основу механизма обработки информации в естественных системах, исследователи стремятся построить теоретические и компьютерные модели, реально объясняющие принципы работы этого механизма. Для исследований этого направления характерно понимание, что модели должны содержать не только рождение и смерть популяций, но и что-то между ними. Чаще всего привлекаются следующие концепции.

**Роевой интеллект** (*Swarm intelligence*) описывает коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации.

Термин был введен Херардо Бени и Ван Цзином в 1989 г. в контексте системы клеточных роботов. Системы роевого интеллекта, как правило, состоят из множества агентов, локально взаимодействующих между собой и с окружающей средой. Сами агенты обычно довольно просты, но все вместе, локально взаимодействуя, создают так называемый роевой интеллект. Примером в природе может служить колония муравьёв, рой пчёл, стая птиц, рыб и т. п.

**Коллективный интеллект** – термин, который появился в середине 1980-х гг. в социологии при изучении процесса коллективного принятия решений. Исследователи из *NJIT* определили коллективный интеллект как способность группы находить решения задач более эффективные, чем лучшее индивидуальное решение в этой группе.

**Социологическое направление** – поскольку человеческое общество представляет собой реальный, к тому же хорошо поддающийся наблюдению и задокументированный (в отличие от человеческого мозга) инструмент обработки

формации, социологические метафоры и реминисценции присутствуют в работах по кибернетике и смежным направлениям с самого их возникновения.

Если роевой интеллект ориентирован на получение сложного поведения в системе из простых элементов, этот подход, наоборот, исследует построение острых и специальных объектов на базе сложных и универсальных: *«государство глупее, чем большинство его членов»*.

Для этого направления характерно стремление дать социологическим понятиям определения из области информатики. Элита определяется как носитель ределенной частной модели реального мира, а базис (т. е. народ) играет роль битра между элитами. Эволюционный процесс заключается в порождении и бели элит. Базис не в состоянии разобраться в сути идей и моделей, представляемых элитами, и не ставит перед собой такой задачи. Однако именно в силу своей невовлеченности он сохраняет способность к ясной эмоциональной оценке, позволяющей ему легко отличать харизматические элиты от загнивающих, пытающихся сохранить свои привилегии, понимая, что их идея или модель не подтвердилась.

## 5. Генетические алгоритмы

Адекватным средством реализации процедур эволюционного моделирования являются *генетические алгоритмы*. Идея генетических алгоритмов «подотрена» у систем живой природы, у систем, эволюция которых развертывается в сложных системах достаточно быстро.

*Генетический алгоритм* – это алгоритм, основанный на имитации генетических процедур развития популяции в соответствии с принципами эволюционной динамики. Часто используется для решения задач оптимизации (в том числе многокритериальной), поиска, управления.

Данные алгоритмы адаптивны, развивают решения, развиваются сами. Особенность этих алгоритмов – их успешное использование при решении сложных проблем (проблем, для которых невозможно построить алгоритм с полиномиальной алгоритмической сложностью).

**Пример.** Работу банка можно моделировать на основе генетических алгоритмов. С их помощью можно выбирать оптимальные банковские проценты (вкладов, кредитов) некоторого банка в условиях конкуренции с тем, чтобы привлечь больше клиентов (средств). Тот банк, который сможет привлечь больше вкладов, клиентов и средств, и выработает более привлекательную **стратегию поведения** (эволюции) – тот и выживет в условиях естественного отбора. Филиалы такого банка (гены) будут лучше приспосабливаться и укрепляться в экономической нише, а, возможно, и увеличиваться с каждым новым поколением. Каждый филиал банка (индивид популяции) может быть оценен мерой его приспособленности. В основе таких мер могут лежать различные критерии, например, аналог экономического потенциала – **рейтинг надежности** банка или соотношение привлеченных и собственных средств банка. Такая оценка эквивалентна оценке того, насколько эффективен организм при конкуренции за ресурсы, т. е. его выживаемости, биологическому потенциалу. При этом банки (филиалы) могут приводить к появлению потомства (новых банков, получаемых в результате слияния или распада), сочетающего те или иные (экономические) характеристики родителей. Например, если один банк имел качественную политику кредитования, а другой – эффективную инвестиционную политику, то новый банк может приобрести и то и другое. Наименее приспособленные банки (филиалы) совсем могут исчезнуть в результате эволюции. Таким образом, отрабатывается генетическая процедура воспроизводства новых банков (нового поколения), более приспособленных и способных к выживанию в процессе эволюции банковской системы. Эта политика со временем пронизывает всю банковскую «популяцию», обеспечивая достижение цели – появления эффективно работающей, надежной и устойчивой банковской системы.

**Примеры.** Для тестирования алгоритмов нахождения глобального экстремума в сложных многоэкстремальных задачах предложены специальные тестовые функции: функция Розенброка и функция Растргина (рис. 1.1–1.3).



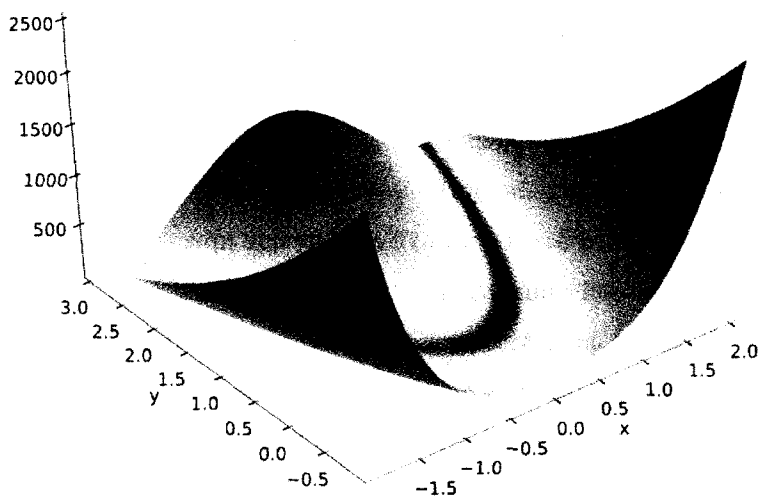


Рис. 1.1. Функция Розенброка

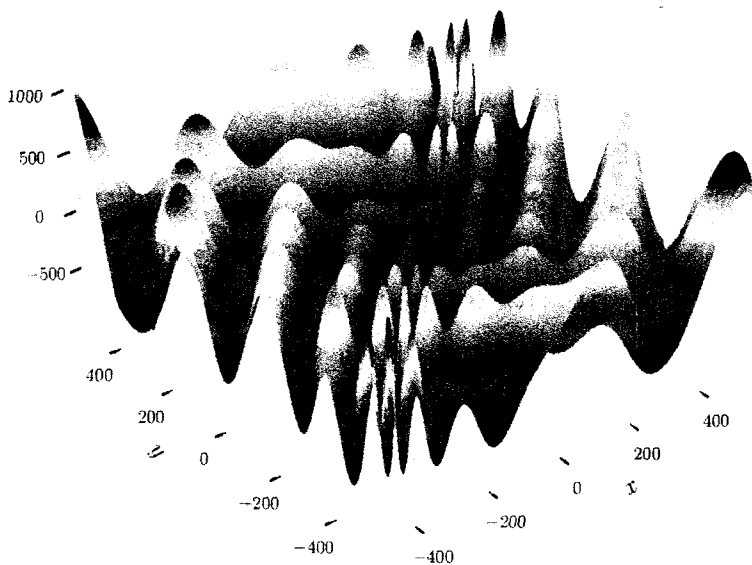


Рис. 1.2. Функция Растригина

Генетические методы не гарантируют нахождение оптимального решения, по крайней мере – за приемлемое время. Главное их преимущество в том, что они позволяют решать сложные задачи, для которых не разработаны устойчивые

чивые и приемлемые методы, особенно на этапе формализации и структурирования системы, в когнитивных системах.

### 1.6. Ансамблевые методы: джекнайф и бутстрэп

Начальный этап развития ансамблевого подхода связан с работами М. Кенуя и Дж. Тьюки и известен как *метод расщепления выборки* или *джекнайф*. Пусть  $\theta(n)$  – оценка, вычисленная по измерениям  $x_1, \dots, x_n$ , а  $\theta(n-1, j)$  – оценка такой же формы, вычисленная по измерениям, из которых устранено значение  $x_j$ . Обозначим  $\bar{\theta}(n-1)$  среднее по всем  $\theta(n-1, j)$ .

Предположим, что оценка  $\theta(m)$  имеет смещение:

$$M[\theta(m)] = \theta + a_1(\theta)/m + a_2(\theta)/m^2 + O(m^{-2}).$$

Тогда

$$\begin{aligned} M[\theta(n)] &= \theta + a_1(\theta)/n + a_2(\theta)/n^2 + O(n^{-2}), \\ M[\bar{\theta}(n-1)] &= \theta + a_1(\theta)/(n-1) + a_2(\theta)/(n-1)^2 + O(m^{-2}) \end{aligned}$$

и линейная комбинация

$$\theta^J(n) = n\theta(n) - (n-1)\bar{\theta}(n-1)$$

имеет математическое ожидание со сниженным смещением

$$M[\theta^J(n)] = \theta + O(n^{-2}).$$

Оценку  $\theta^J(n)$  называют расщепленной оценкой, отвечающей  $\theta(n)$ .

Описанная техника имеет много различных модификаций. Прежде всего, анализируя облако значений  $\theta(n-1, j)$ , можно выделить в нем компактное ядро и выпадающие значения, что позволяет организовывать отбраковку выпадающих измерений. Во-вторых, процедуру можно применять несколько раз, добиваясь все более полного устранения смещения. Наконец, вводя величины

$$\theta^P(j) = n\theta(n) - (n-1)\theta(n-1, j),$$

называемые псевдозначениями, можно оценить дисперсию  $\text{var}[\theta^J(n)]$  или  $\text{var}[\theta(n)]$  как

$$\frac{1}{n(n-1)} \cdot \sum_{j=1}^n [\theta^P(j) - \theta^J(n)]^2.$$

Следующий этап развития подхода носит название *бустреп*. Его разветвления описаны в монографии Брэдли Эфрона. Базовая модель выглядит следующим образом. Рассмотрим в традиционных обозначениях линейную модель измерений  $Y = A\theta + \varepsilon$ . Оценка параметра  $\theta$  по методу наименьших квадратов получается минимизацией квадратичной формы

$$V(\theta) = (Y - A\theta)^T(Y - A\theta)$$

и имеет вид  $\hat{\theta} = (A^T A)^{-1} A^T Y$ . Рассмотрим семейство псевдооценок

$$\hat{\theta}(s) = [A^T (I + \text{diag}(s)) A]^{-1} A^T Y,$$

где  $I$  – единичная матрица, а  $s$  – случайный вектор, составленный из независимых одинаково распределенных случайных величин с дисперсией, достаточно малой, чтобы обеспечить обращение матрицы. Разыгрывая с помощью датчика случайных чисел большое число реализаций псевдооценок и вычисляя для них  $\exp\{-V[\hat{\theta}(s)]/2\}$ , получаем после нормировки выборочную оценку распределения оценки  $\hat{\theta}$ , в частности, ее ковариационную матрицу. Метод хорошо зарекомендовал себя при получении оценок в негауссовых ситуациях, при локальной линеаризации нелинейных процедур оценивания, а также в тех случаях, когда требуется получить несмещенную оценку не самого  $\theta$ , а некоторой нелинейной функции от этого параметра. При этом для снижения смещения оценки используется техника, аналогичная приведенной выше при обсуждении процедуры джекнайфа. В частности, на этом пути легко реализуются процедуры оценивания при наличии ограничений (рестриктивного оценивания). Для этого при формировании оценки выборочного распределения учитывают только те псевдооценки, которые удовлетворяют введенным ограничениям.

### 1.7. Ансамблевые методы: бэггинг, бустинг, стекинг

*Ансамблевые методы* – это парадигма машинного обучения, где несколько моделей (часто называемых *слабыми учениками*) обучаются для решения одной и той же проблемы и объединяются для получения лучших результатов. Основная гипотеза состоит в том, что при правильном сочетании слабых моделей мы можем получить более точные и/или надежные модели.

- В машинном обучении, независимо от того, сталкиваемся ли мы с проблемой классификации или регрессии, выбор модели чрезвычайно важен. Этот выбор может зависеть от многих переменных задачи: количества данных, размерности пространства, гипотезы о распределении и т. п.
- Слабое *смещение (bias)* и *разброс (variance)* модели чаще всего изменяются в противоположных направлениях, так что между ними требуется найти *компромисс*.

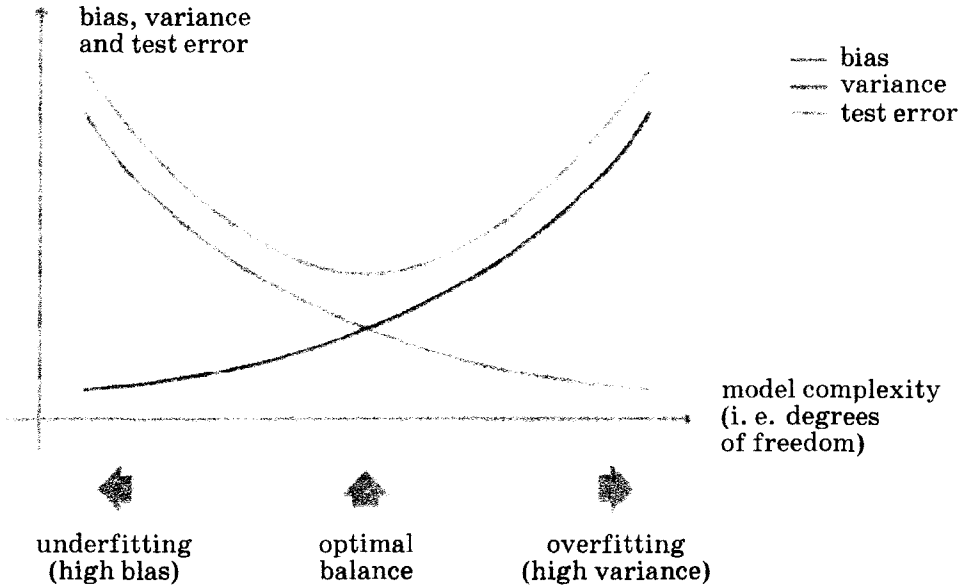


Рис. 1.3. График специальной функции ансамблевых методов

Идея ансамблевых методов состоит в том, чтобы попытаться уменьшить смещение и/или разброс таких слабых учеников, объединяя их вместе, чтобы создать *сильного ученика (или модель ансамбля)*, который достигает лучших результатов.

- **Объединение слабых учеников.** Чтобы реализовать ансамблевый метод, требуется правильно комбинировать эти модели. Есть три основных типа *мета-алгоритмов*, которые направлены на объединение слабых учеников.
- **Бэггинг (bootstrap aggregation).** В этом случае рассматривают однородных слабых учеников, обучают их параллельно и независимо, а затем объединяют, следуя некоторому процессу усреднения. Одним из больших

преимуществом бэггинга является его параллелизм: поскольку различные модели обучаются независимо друг от друга, при необходимости могут использоваться методы интенсивного распараллеливания.

- **Бустинг.** В этом случае рассматривают однородных слабых учеников, обучают их последовательно адаптивным способом (слабый ученик зависит от предыдущих) и объединяют, следуя детерминированной стратегии. Идея состоит в том, чтобы последовательно (итеративно) обучать модели таким образом, чтобы обучение на данном этапе зависело от моделей, обученных на предыдущих этапах.
- **Стекинг.** В этом случае учитывают разнородных слабых учеников, изучают их параллельно и объединяют, обучая метамодель для вывода прогноза, основанного на предсказаниях различных слабых моделей.

Можно сказать, что бэггинг в основном сосредоточен на получении ансамблевой модели с меньшим разбросом, чем ее компоненты, в то время как бустинг и стекинг в основном будут пытаться производить сильные модели с меньшим смещением, чем их компоненты.

Существует несколько возможных способов объединить несколько моделей, обученных параллельно. Для задачи регрессии выходные данные отдельных моделей могут быть буквально усреднены для получения выходных данных модели ансамбля. Для задачи классификации класс, предсказываемый каждой моделью, можно рассматривать как голос, а класс, который получает большинство голосов, является ответом модели ансамбля (это называется **мажоритарным голосованием**). Что касается задачи классификации, мы также можем рассмотреть вероятности каждого класса, предсказываемые всеми моделями, усреднить эти вероятности и сохранить класс с самой высокой средней вероятностью (это называется **мягким голосованием**). Средние значения или голоса могут быть простыми или взвешенными, если будут использоваться любые соответствующие им веса.

*Деревья решений* являются очень популярными базовыми моделями для ансамблевых методов. Сильных учеников, состоящих из нескольких деревьев

решений, можно назвать «лесами». Деревья, составляющие лес, могут быть выбраны либо неглубокими (глубиной в несколько узлов), либо глубокими (глубиной в множество узлов, если не в полную глубину со всеми листьями). Неглубокие деревья имеют меньший разброс, но более высокое смещение, и тогда для них лучшим выбором станут *последовательные методы*. Глубокие деревья, с другой стороны, имеют низкое смещение, но высокий разброс.

*Метод случайного леса* – это метод бэггинга, где глубокие деревья, обученные на бутстрэп выборках, объединяются для получения результата с более низким разбросом. Случайный лес – это, как утверждают, один из самых потрясающих алгоритмов машинного обучения, придуманные Лео Брейманом и Адель Катлер ещё в прошлом веке. Он дошёл до нас в «первозданном виде» (никакие эвристики не смогли его существенно улучшить) и является одним из немногих универсальных алгоритмов. Есть случайные леса для решения задач классификации, регрессии, кластеризации, поиска аномалий, селекции признаков и т. д.

Рассматривают две важные модификации бустинга: *adaboost (адаптивный)* и *градиентный*. Эти два мета-алгоритма отличаются тем, как они создают и объединяют слабых учеников в ходе последовательного процесса. Адаптивный бустинг обновляет веса, приписываемые каждому из объектов обучающего дэйтасета, тогда как градиентный бустинг обновляет значения самих частных результатов.

Существуют варианты исходного алгоритма *adaboost*, такие как **LogitBoost** (классификация) или **L2Boost** (регрессия), которые в основном различаются по своему выбору функции потерь.

## 2. ИЗВЛЕЧЕНИЕ ЗНАНИЙ

### 2.1. Виды знаний и способы их представления

*Знания* – это формализованная и структурированная информация, используемая в процессе решения задачи.

*Фактические знания* – это основные закономерности предметной области – факты, понятия, взаимосвязи, оценки, правила, эвристики (индивидуальный опыт).

*Процедурные знания* – способы оперирования или преобразования фактических знаний.

*Стратегические знания* – основные закономерности принятия решений в данной области.

Знания о конкретных объектах – *экстенциональные*. Знания о связях между атрибутами (признаками) в данной предметной области – *интенциональные*. Понимание структуры предметной области, назначения и взаимосвязи понятий – *глубинные знания* (законы, теоретические основания). Внешние эмпирические ассоциации – *поверхностные знания*.

*Жесткие знания* приводят к четким однозначным рекомендациям. *Мягкие знания* допускают размытые решения и множественные варианты рекомендаций.

Знания, сведенные в логически связанную систему, называют *онтологией*. Так называют хорошо организованную *базу знаний*.

### 2.2. Модели представления знаний

*Продукционные системы*. Знания представляются в виде совокупности специальных информационных единиц, включающих данные (факты), правила получения продукций (выводов) и интерпретатор (правила работы с продукциями). Особенность – отсутствие средств для установления иерархии правил.

*Логические модели* основаны на логике предикатов (функций со значениями И и Л) и исчислении высказываний. Хорошо работают в условиях, когда предметная область полностью известна и формально описана.

**Фреймы** – структуры данных для представления стереотипных ситуаций. Характеристики ситуации – *слоты*, их значения – *заполнители слотов*. **Протофрейм** – оболочка, **экзофрейм** – результат ее заполнения. Слот может содержать указание на процедуру своего заполнения, в т. ч. эвристическую. **Фасет** – диапазон или перечень значений слота.

**Процедуры-демоны** запускаются автоматически при выполнении некоторого условия.

**Процедуры-слуги** активизируются только по специальному запросу.

Предметную область описывают с помощью *иерархической системы фреймов*, объединенной с помощью родовидовых связей.

Системы программирования, основанные на фреймах, называют *объектно-ориентированными*.

**Семантическая сеть** описывает знания в виде сетевых структур. В качестве вершин сети выступают понятия, факты, объекты, события и т. п., в качестве дуг – отношения.

Из других методов широко применяется *представление знаний по примерам*. Имеется расширяемая матрица примеров; для данной ситуации можно формально искать ближайший пример в заданной метрике.

### 2.3. Извлечение знаний

**База знаний** является ключевым элементом систем искусственного интеллекта. Чтобы эти системы достигли высокого уровня совершенства, база знаний должна соответствовать, по крайней мере, некоторой совокупности требований. В настоящее время известно достаточно много удачно накопленных и формализованных *баз знаний (онтологий)* в различных областях применения, пригодных для непосредственного использования в системах искусственного интеллекта. В создании подобных систем, как правило, участвует много людей. Часть из них является программистами, другая часть – носителями знаний о данной предметной области. На ранних этапах развития систем искусственного интеллекта посредников между этими людьми называли *инженерами знаний*. Они должны были извлечь знания из носителей знаний в конкретной предмет-



ной области и уметь воплотить эти знания в формальные онтологии для последующего использования в виде баз знаний системы искусственного интеллекта. Создаваемая в процессе извлечения знаний онтология должна быть:

- обоснованной (построенной на основе знаний высококвалифицированных экспертов);
- полной – способной давать ответ (осуществлять вывод) на все возможные вопросы в предварительно очерченной предметной области;
- непротиворечивой (давать непротиворечивые ответы на любые заданные вопросы).

Таким образом, *инженер знаний* – это специалист, основной задачей которого является проектирование баз знаний и наполнение их знаниями о предметной области. В процессе этой деятельности он выбирает форму представления знаний, удобную для данной предметной области, организует приобретение знаний из различных источников, включая специалистов в данной предметной области.

Приведем список (возможно, неполный) наиболее известных способов получения знаний о предметной области:

- извлечение знаний из книг, инструкций, документов и т. п.;
- использование специальных опросников с последующей их обработкой;
- интервьюирование специалистов, в ходе которого задаваемые вопросы определяются не только общим планом интервью, но и характером получаемых от специалистов ответов;
- получение нужных знаний в режиме «мозгового штурма»: в специально созданной стимулирующей обстановке носители знаний (эксперты) спонтанно генерируют информацию, которая затем подвергнется обработке и анализу;
- получение информации от эксперта, выступающего в роли учителя (лектора);
- формирование базы знаний самим экспертом, совмещающим роли эксперта и инженера знаний;

- использование методов распознавания образов для накопления знаний путем обработки экспериментально получаемой информации;
- использование методов машинного обучения для автоматического заполнения базы знаний.

На рисунке 2.1 приведена классификация методов извлечения знаний.

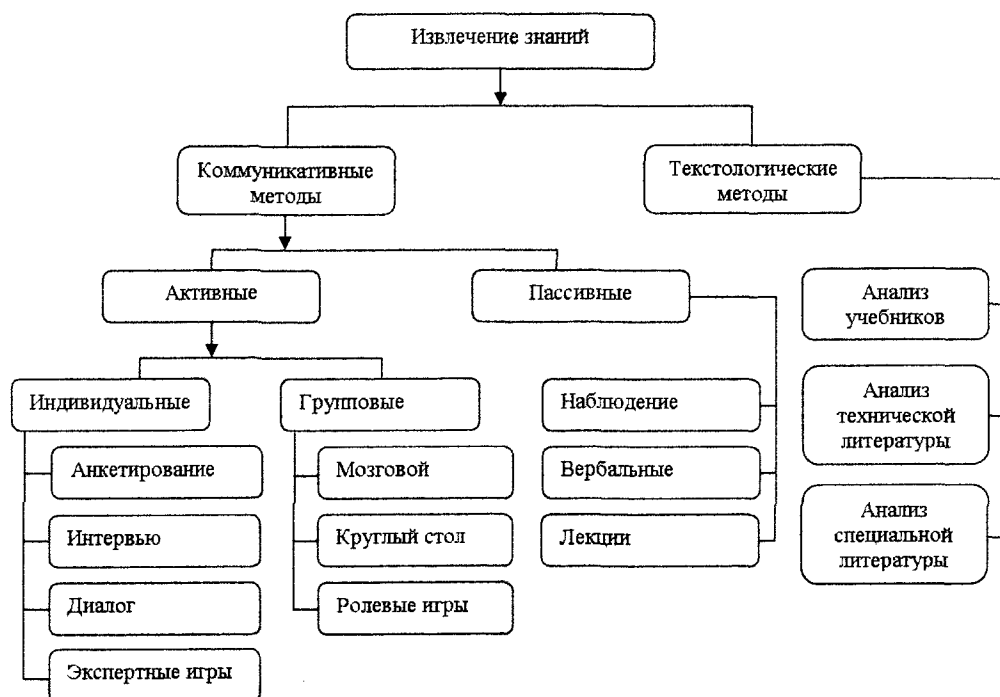


Рис. 2.1. Методы извлечения знаний

В рамках психологического аспекта наибольший интерес вызывает когнитивный (познавательный) слой, который в настоящее время является наименее изученным. Замечено, например, что наиболее легко понимаются, усваиваются и передаются от одного человека к другому сведения, являющиеся ответом на вопросы типа: «зачем?», «что?», «как?», «почему?». Проблематика лингвистического аспекта тесно переплетается с проблематикой когнитивного слоя.

В общем случае следует признать, что *эксперт – не очень надежный источник знаний*, но никем еще не предложено удовлетворительной общей методики, позволяющей приобретать достаточный объем достоверных знаний. Извлечение и представление знаний, т. е. их приобретение пока еще остается

искусством, и лишь отдельные элементы этого процесса имеют инструментальную поддержку.

Признано, что перечень свойств, которыми должен был бы обладать «идеальный» эксперт, может быть отражен в виде рисунка 2.2.

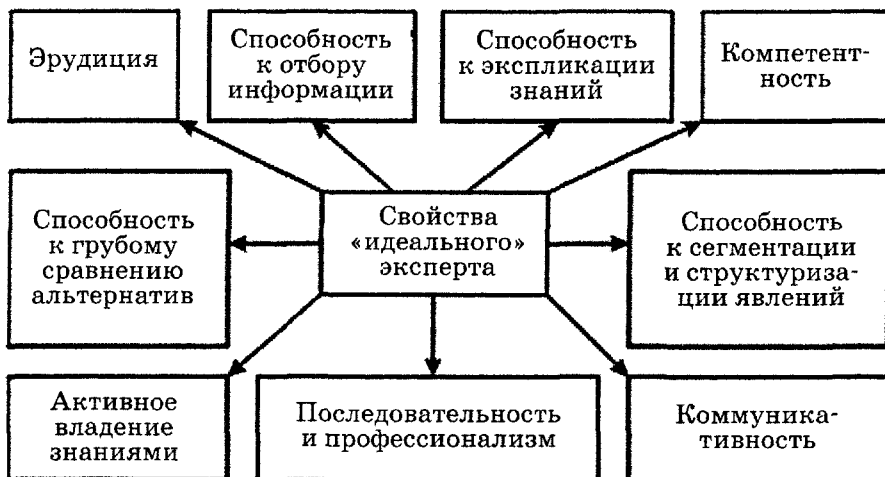


Рис. 2.2. Свойства «идеального» эксперта

Однако в силу особенностей личности человека, его психологических свойств, существует ряд ограничений, которые практически почти непреодолимы. Процесс извлечения знаний из экспертов чрезвычайно трудоемок и составляет, по некоторым оценкам, 75–90% от трудоемкости всего процесса создания системы искусственного интеллекта, плохо прогнозируется, обладает большими временными затратами и психологическими нагрузками для всех его участников. При этом характерным для эксперта является плавное снижение уровня его компетентности. Если процесс решения какой-либо задачи, лежащей «на стыке» разных областей, не укладывается в рамки его профессионального опыта, то недостаточная компетентность проявляется не во внезапном отказе от решения задачи, а в постепенном ухудшении качества решения.

До настоящего времени не существует единой классификации методов извлечения знаний из экспертов, но на практике существует более сотни практических методик для работы с источником знаний.

*Инженер знаний* должен работать с экспертом, наблюдая, как он решает конкретные задачи. Редко оказывается эффективным подход, при котором эксперту напрямую задаются вопросы о его правилах (методах) решения конкретного класса задач в его области компетентности. Знание о том, что считать основным и относящимся к делу и не требующим дальнейшей переоценки – вот что делает специалиста экспертом. Знания эксперта – это не просто неупорядоченный свод фактов, так как большое число образцов-шаблонов решения задач определенного класса служат ему указателями.

В большинстве перечисленных способов роль инженера знаний оказывается чрезвычайно высокой. Типовой процесс приобретения знаний при построении систем искусственного интеллекта схематически может быть представлен в виде рисунка 2.3.

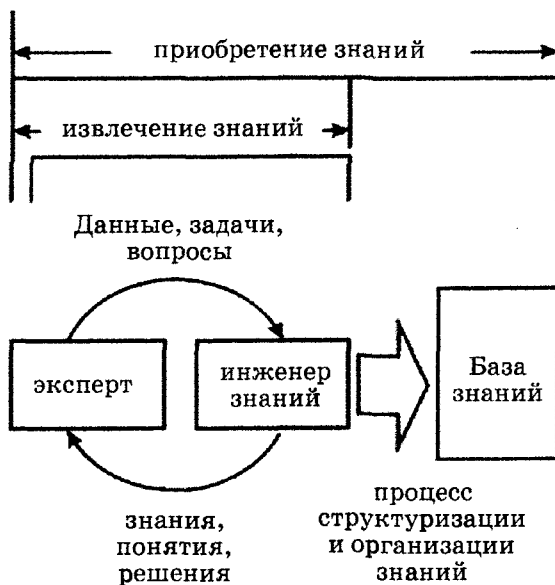


Рис. 2.3. Типовой процесс приобретения знаний

Опыт свидетельствует, что по мере продвижения от научно-исследовательских работ по созданию систем искусственного интеллекта к реальным, наблюдаются *трудности с поиском экспертов*. Эксперт при решении задач, в которых качественные, трудно формализуемые и неопределенные факторы имеют тенденцию доминировать, в принципе располагает возможностью:

- применять свои знания и опыт для оптимального решения задач, делать достоверные выводы и умозаключения, исходя из неточных или ненадежных данных;
- общаться с другими экспертами (не обязательно в данной предметной области) и приобретать новые знания;
- объяснять и обосновывать свои действия;
- заново систематизировать свои знания;
- нарушать правила, так как в распоряжении специалиста находится практически столько же исключений из правил, сколько и самих правил, а специалист разбирается в правилах не только по форме, но и по содержанию;
- определять степень своей компетентности в каждом конкретном случае, так как отчетливо представляет какие задачи выходят из сферы его компетенции и в каких случаях следует обращаться за консультацией к другим источникам.

Методы извлечения знаний, являющиеся коммуникативными, используются не изолированно, а совместно, дополняя друг друга. Существуют различные стратегии их объединения, например в следующей последовательности:

- наблюдения;
- протоколы;
- интервью;
- диалог;
- круглый стол.

*Текстологические методы* продолжают развиваться. Проблемы текстологических методов связаны с трудностями понимания смысла, закладываемого автором и постигаемого аналитиком: смысл, отражаемый в тексте, образуется из совокупности опыта эксперта. Смысл, постигаемый инженером знаний из текста, образуется в результате интерпретации текста за счет привлечения индивидуального научного и общекультурного опыта. Заметим, что возможность

применения того или иного метода в значительной степени зависит от структурированности знаний, которыми владеет эксперт.

Существующие методы извлечения знаний предназначены в основном для решения тех проблем, где набор признаков и перечень возможны и считаются известными.

В целом, среди проблем, затрудняющих эффективную реализацию процесса извлечения знаний, можно выделить следующие:

- отсутствие достаточно всеобъемлющих теоретических методов для описания природы экспертизы;
- трудности вербального выражения знаний и декомпозиции;
- сильный субъективизм существующих способов извлечения знаний;
- влияние личностных качеств эксперта и его ответственности за успех создания базы знаний и многие другие.

В настоящее время большинство специалистов по искусственному интеллекту склоняется к мнению о существовании проблематики трех основных теоретических аспектов извлечения знаний (рис. 2.4): психологического, лингвистического и гносеологического.

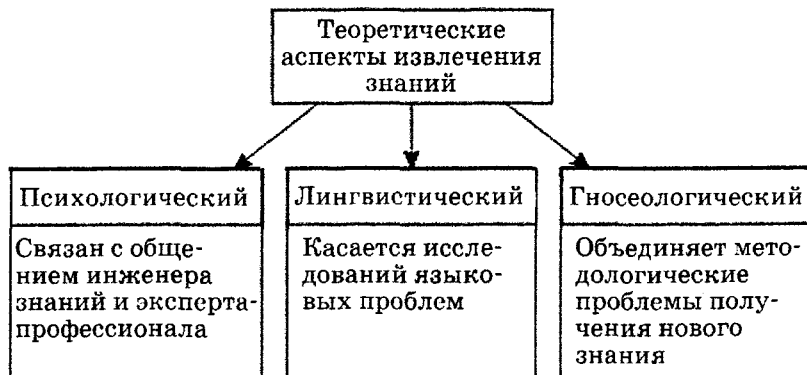


Рис. 2.4. Теоретические аспекты извлечения знаний

## 2.4. Некоторые подходы к интеллектуальному анализу данных

Компьютеры уверенно берут на себя функции, считавшиеся прерогативой интеллектуальной деятельности человека. Причиной интеллектуализации компьютеров стали:

- исследования, *моделирующие процесс получения результата* с минимальными претензиями на понимание механизма;
- перенесение главного акцента компьютерных разработок с вычислительных программ на приложения, осуществляющие *представление и манипулирование знаниями*.

В создании интеллектуальных компьютерных систем выделяют следующие основные направления.

**Интеллектуальные информационно-поисковые системы (ИИПС).** Они отличаются важнейшей способностью формировать адекватные ответы даже на недостаточно четко сформулированные вопросы. Другой их особенностью является способность «переваривать» огромные количества информации, осуществлять ее автоматическое реферирование и проверку на противоречивость и неполноту.

**Экспертные системы (ЭС)** предназначены для решения практических задач в слабо структурированных и трудно формализуемых предметных областях. Эти системы аккумулируют профессиональные знания опытных экспертов.

**Обучающие системы,** которые нередко называют «*тьюторами*», являются разновидностью экспертных систем. Тьюторы применяются для профессионального обучения будущих специалистов, на первый план выходят *знания о методе*.

Развитие этих направлений определяется *тремя парадигмами*.

- *Первая* связана с *архитектурными решениями* на основе параллельных и асинхронных процессов, перемещающихся по структуре взаимосвязанных компьютеров.
- *Вторая* парадигма ИИ – *когнитивная компьютерная графика* – наглядное изображение внутреннего содержания предмета, любого абстрактного научного понятия, гипотезы или теории. Использование мультимедиа открывает «правополушарные» каналы связи между исследователем и интересующей его проблемой.

- *Третья* парадигма состоит в создании *интеллектуальных гибридных систем*, обеспечивающих комфортное взаимодействие с пакетами прикладных программ и делающих доступной для него широкие возможности вычислительной математики.

Всё это обусловлено быстрым увеличением технических возможностей современных компьютеров. Узким местом ИИ остается *проблема получения и манипулирования знаниями*, которые составляют основу любой интеллектуальной системы.

При решении проблемы получения знаний выделяют три стратегии: *приобретение знаний, извлечение знаний и формирование знаний*.

Под *приобретением (acquisition)* знаний понимается способ автоматизированного наполнения базы знаний посредством диалога эксперта и специальной программы. Автоматизированные средства получают готовые фрагменты знаний в соответствии со структурами, заложенными разработчиками системы. Большинство инструментальных средств ориентировано на конкретные экспертные системы со своей предметной областью и моделью представления знаний. Например, система **TEIRESIAS** предназначена для пополнения базы знаний системы **MYCIN** или ее дочерних ветвей, построенных на оболочке **EMYCIN** в области медицинской диагностики с использованием продукционной модели представления знаний.

При попытке использования систем приобретения знаний возникают следующие проблемы:

- неудачный способ приобретения, не совпадающий со структурой данной области;
- неадекватная модель представления знаний;
- отсутствие целостной системы знаний в результате приобретения фрагментов;
- упрощение и уплощение «картины мира» и пр.

*Извлечением (elicitation)* знаний называют процедуру взаимодействия инженера по знаниям с источником знаний (экспертом, специальной литерату-



рой и др. Это длительная и емкая процедура, в которой *инженеру по знаниям*, владеющему методами системного анализа, математической логики нужно *воссоздать модель предметной области, используемой экспертами*. Актуальность включения в процесс инженера по знаниям обусловлена следующими причинами:

- во-первых, значительная часть знаний эксперта является результатом многочисленных наслоений и опыта, эксперт не всегда может самостоятельно анализировать детали в цепи своих умозаключений;
- во-вторых, диалог инженера и эксперта служит наиболее естественной формой «раскручивания» лотков памяти эксперта, в которых хранятся знания, часто носящие неверный характер;
- в-третьих, многочисленные причинно-следственные реальной предметной области образуют сложную систему, скелет которой более доступен для восприятия аналитика, владеющего системной логикой и не обремененного знанием большого количества подробностей.

Термин *«формирование знаний»* связывают с созданием систем автоматического получения знаний, *«машинного обучения» (machine learning)*. На сегодняшний день это наиболее перспективное направление, предполагающее, что система сможет самостоятельно сформировать необходимые знания на основе имеющегося материала. Инженер по знаниям с помощью одного лишь диалога с экспертом конкретной области не способен добыть все нужные для разработки интеллектуальной системы сведения. Требуется еще и множество примеров, на которых удастся обучить машину.

В самом общем виде формирование знаний – это задача обработки данных с целью перехода к базам знаний (БЗ). В базе данных (БД) накапливаются, хранятся эмпирические факты из исследуемой предметной области (данные, примеры экспертных заключений, элементарные высказывания с некоторой оценкой и т. п.), представленные в виде троек <объект, пример, значение признака>. В БЗ заносятся сведения, выражающие закономерности структуры множества эмпирических фактов, связанные с прикладным текстом.

Чаще всего на практике встречаются отношения эквивалентности и порядка. Отношения эквивалентности присущи, в частности, задачам *классификации, диагностики и распознавания образов* (свой-чужой). Отношения порядка свойственны задачам *шкалирования, прогнозирования* и т. п.

Методы формирования знаний имеют много общего с решениями задач классификации, диагностики и распознавания образов. Но одной из их главных черт является требование интерпретации закономерностей, лежащих в основе правил вхождения объектов в классы эквивалентности. В инженерии знаний большое распространение получили логические методы: «эмпирическое предсказание», «индуктивное формирование понятий», «построение квазиаксиоматической теории».

Еще одна важная причина, обусловившая приоритет логических методов, заключается в сложной системной организации изучаемых областей. Они относятся, как правило, к надкибернетическому уровню организации систем, закономерности здесь не могут быть достаточно точно описаны на языке статистических или иных математических моделей.

Центральной проблемой создания таких конструкций остается проблема перебора большого количества вариантов. Применение логических методов часто вынуждено опираться на эвристические соображения, не имеющие строгого обоснования.

Альтернативу логическим символьным методам составляет *геометрический подход*, переводящий задачу формирования знаний на язык геометрических соотношений между эмпирическими фактами, отображаемыми точками в пространстве признаков. В геометрическом подходе главными элементами выступают *объекты*, а основным видом операций является определение расстояния между объектами в многомерном пространстве признаков. Это делает более понятными критерии и принципы построения правил вхождения объектов в определенные классы эквивалентности в виде мер, имеющих интерпретацию расстояний. Использование геометрического подхода при неограниченном расширении множества эмпирических фактов автоматически приводит к мини-

мальным теоретически достижимым ошибкам при принятии решений. Можно получать *наглядные визуальные представления о логических закономерностях* в структуре данных – для этого применяется специальная локальная геометрия.

**Материал для размышления – теорема Джеймса-Стейна.** В 50-х гг. XX века американские математики Джеймс и Стейн показали, что даже для близких критериев качества методы, оптимальные для одного критерия, могут оказаться недопустимыми для другого и наоборот. Это означает необходимость специального исследования гладкости результата в окрестности исходных предположений: рекомендации экспертов для интуитивно близких ситуаций могут не допускать гладкой интерполяции.

Коренная проблема геометрического подхода состоит в поиске ответа на вопрос: *какие признаки и какую меру следует выбрать для определения расстояний между объектами?* В известных методах анализа эта задача формулируется как подбор взвешенной метрики с использованием логики и обучающей информации или как оцифровка переменных, основанная на максимизации статистического критерия.

Традиционные методы анализа многомерных данных, опирающиеся на геометрическую метафору, используют представление об *общем пространстве признаков* для всех объектов и об *одинаковой мере*, применяемой для оценки их сходства или различия.

В задачах формирования знаний, когда мы имеем дело с системами надкибернетического уровня сложности, каждый объект имеет важные уникальные особенности. Они раскрываются путем конструирования для любого объекта *собственного пространства признаков* и нахождения *индивидуальной меры*, определяющих иерархию его сходства с другими объектами в заданном контексте.

## 2.5. Формирование знаний методами локальных геометрий

Конструирование собственного пространства признаков и нахождение индивидуальной меры будем называть *локальным преобразованием пространства* признаков. Пусть  $X = [x_{ij}]$ ,  $i = 1, \dots, N$ ,  $j = 1, \dots, p$  – матрица дан-

ных, где  $x_{ij}$  – значение  $j$ -го признака у  $i$ -го объекта. Тогда задача преобразования описания объекта  $x_i$  формулируется как определение *контекстно-зависимой локальной взвешенной метрики*  $d_i(x_i, x_j)$  того или иного типа, обеспечивающей релевантную контексту иерархию близостей (удаленностей) объектов  $x_j$  ( $j = 1, \dots, p$ ) относительно объекта  $x_i$ .

Индивидуально сконструированные локальные метрики обеспечивают каждому объекту, как представителю своего класса, максимально возможную сферу действия, чего нельзя достигнуть при построении общего пространства признаков и использовании одинаковой метрики для всех объектов.

В свете представлений о контекстно-зависимых локальных метриках очевидно, что один и тот же объект может поворачиваться разными гранями своего многомерного описания сообразно заданному контексту. К любому объекту, запечатленному в памяти как целостная многомерная структура, «привязан» набор различных локальных метрик, каждая из которых оптимизирует иерархию его сходства (различия) с другими объектами соответственно целям определенной задачи отражения отношений между объектами реального или идеального мира.

В задачах отражения отношений эквивалентности (например, проблемах диагностики или распознавания образов) после построения локальной метрики каждый объект может интерпретироваться как самостоятельный линейный классификатор с некоторыми оптимальными свойствами, определяемыми менявшимся критерием. Соответственно вся выборка данных должна рассматриваться с учетом совокупности  $N$  локально оптимальных линейных классификаторов. Для исследования их взаимодействия с целью формулирования конечных выводов пригодны известные подходы к построению решающих правил. Проведение такого исследования возможно в русле все той же геометрической метафоры.

В результате построения локальных метрик отношения между объектами выражаются матрицей удаленностей. Так как локальные метрики у разных объектов могут не совпадать, для элементов этой матрицы могут не выполняться

требования симметричности и неравенства треугольника. Поэтому данная матрица, хотя и отражает соотношения различия между объектами, не может истолковываться как матрица расстояний. Один из возможных приёмов здесь состоит в том, что на основе локальной метрики каждого объекта проводится ранжирование, а потом с получившимися ранжировками работают стандартными методами непараметрической статистики (коэффициенты ранговой корреляции, медианы Кендалла и Спирмена, коэффициенты конкордации).

Имеется еще одна ценная возможность использования визуальных отображений полученных геометрических структур данных. Ее предоставляют средства современной интерактивной графики, которые позволяют обосновывать принятие решения о принадлежности неизвестного объекта какому-либо классу эквивалентности, получая ответы на вопросы типа: «Что общего у данного объекта с другим объектом или группой объектов (например, визуально ближайших или, наоборот, удаленных) с известной классификацией?», «Чем отличается данный объект от другого объекта или группы объектов с известной классификацией?» и т. п. Ответы даются в виде пересечения описания неизвестного объекта с описаниями объектов, которые оптимизированы привязкой контекстно-зависимых локальных метрик. Совокупность таких ответов, индивидуальных для каждого нового случая, обладает полиморфностью, свойственной естественному языку при описании явлений со сложной системной организацией, и обеспечивает объяснение принятых решений посредством аргументации. Здесь нет дерева логического вывода. Ответы воспринимаются параллельно.

Интересными представляются возможности применения описанного подхода для анализа совокупности объектов при отсутствии информации об их группировании в какие-либо классы. В этом случае в целях конструирования локальных метрик искусственно создается альтернативный класс из равномерно распределенных объектов. Тогда сконструированные локальные метрики оптимизируют описание каждого объекта таким образом, что в нем остается только то, что является важным для выражения отличия структуры анализируемой совокупности от случайно организованной структуры данных.

### 3. ИММУНОКОМПЬЮТИНГ

Пусть результаты  $n$  векторных измерений представлены в виде таблицы  $X \langle n \times m \rangle$ ,  $n > m$ ,  $\text{rank}(X) = p$ :  $X$  является матрицей ранга  $p \leq m$ . Задача состоит в том, чтобы аппроксимировать ее матрицей меньшего ранга  $k < p$ .

С точки зрения анализа данных это означает, что мы пытаемся объяснить имеющуюся структуру многомерных данных меньшим числом  $k$  обобщенных признаков,  $k < p < m$ . При вероятностном подходе этой цели служат различные модификации метода главных компонент, математической основой которых является предположение о том, что строки  $X$  – это независимые случайные векторы, подчиняющиеся  $m$ -мерному нормальному закону  $N_m(a, \Sigma)$ . В данном разделе мы рассмотрим эту задачу без какой-либо вероятностной подоплеки.

Таким образом, решается следующая *экстремальная задача*: для данной матрицы  $X = [x_{ij}]$  размерности  $\langle n \times m \rangle$  найти матрицу  $Y$  той же размерности из условия

$$\sum_{i,j} (x_{ij} - y_{ij})^2 \rightarrow \min \quad (1)$$

при ограничении  $\text{rank}(Y) = k < \min(n, m)$ .

**Сингулярное разложение матрицы (SVD, Singular Value Decomposition).** Произвольную вещественную матрицу  $X \langle n \times m \rangle$ ,  $n > m$  можно представить в виде *сингулярного разложения*

$$X = L * S * R^T, \quad (2)$$

где

- $S = \text{diag}(s_1, \dots, s_n)$  – диагональная матрица; ее элементы  $s_1 \geq s_2 \geq \dots \geq s_n \geq 0$  называются сингулярными числами матрицы  $X$ ;
- $L$  – матрица размерности  $\langle n \times n \rangle$ ; ее столбцы  $L_1, \dots, L_n$  – ортогональные векторы единичной длины, их называют левыми сингулярными векторами  $X$ ;  $L * L^T = L^T * L = E$ ;
- $R$  – матрица размерности  $\langle m \times m \rangle$ ; ее столбцы  $R_1, \dots, R_m$  – ортогональные векторы единичной длины, их называют правыми сингулярными векторами  $X$ ;  $R * R^T = R^T * R = E$ .

Ортогональность здесь понимается в евклидовом смысле, для матриц данных они будут коррелированы.

Если ранг матрицы  $X$   $rank(X) = p < m$ , то среди сингулярных чисел только  $p$  будут отличными от нуля.

Разложение (2) можно переписать в виде суммы

$$X = \sum_{i=1}^p s_i L_i R_i^T = s_1 L_1 R_1^T + \dots + s_p L_p R_p^T. \quad (3)$$

**Теорема** (Эккардт-Янг). Решением экстремальной задачи (1) является сумма первых  $k$  слагаемых в (3):

$$X \cong \sum_{i=1}^k s_i L_i R_i^T = s_1 L_1 R_1^T + \dots + s_k L_k R_k^T.$$

В частности, при выборе  $k=1$  наилучшее приближение дают первое (максимальное) сингулярное число и соответствующие ему сингулярные векторы:

$$A \cong s_1 L_1 R_1^T. \quad (4)$$

Если  $X$  – это матрица данных, то она здесь превращается в сумму небольшого числа «полотнищ» той же размерности, но очень простой структуры: каждое из «полотнищ» представляет собой матрицу единичного ранга.

Сингулярное разложение матриц устойчиво к малым возмущениям матриц, т. е. является хорошо обусловленной процедурой. Такие свойства не свойственны спектральному разложению, которое формирует основу для многомерного статистического анализа.

***Имеется пять основных линий развития данного подхода.***

- При решении задач распознавания, классификации и кластеризации используются проекции векторов на пространство, порожденное несколькими сингулярными компонентами (3), что порождает специфическую псевдометрику.
- Аналогично решаются задачи ситуационного анализа: наблюдаемая ситуация  $x_0$  ассоциируется с ближайшей в псевдометрике из штатных ситуаций  $x_1, \dots, x_k$ .
- При решении задач интерполяции случайных полей значение  $f(x_0)$  оценивается как результат линейной интерполяции по  $k$  ближайшим точкам  $x_1, \dots, x_k$ :

$$f = c_1 f(x_1) + \dots + c_k f(x_k),$$

где

$$c_j = 1 / \left( 1 + d_j \sum_{i \neq j}^k \frac{1}{d_i} \right).$$

Специфика подхода состоит в том, что в качестве меры близости  $d_j$  от  $x_0$  до  $x_j$  опять используется метрика в пространстве проекций.

- Если отрезок  $m$ -мерного ряда представлен в виде матрицы  $\langle n \times m \rangle$ , то можно аппроксимировать её суммой элементарных матриц единичного ранга и анализировать по отдельности ряды-слагаемые упрощённой структуры. Это позволяет значительно уменьшить размерность задачи.
- При анализе одномерных рядов рассматривается группа методов, основанная на вложении временного ряда в многомерное пространство с последующим сингулярным разложением полученной ганкелевой матрицы. Кульминацией этих идей стал метод *анализа сингулярного спектра* (*ACC, Singular Spectrum Analysis, SSA*, в отечественной литературе также известный под названием «Гусеница»), который позволяет решать задачи выделения компонент временного ряда различной структуры и решать задачи описания их структуры, прогнозирования, оценки параметров, обнаружения различных типов разладки.

Рассмотренные ниже примеры основаны на реальном 5-мерном временном ряде измерений газодинамических параметров, оформленном в виде матрицы `YY.mat` размерности  $\langle N, m \rangle$ ,  $n = 144000$ ,  $m = 5$ . При выполнении студентами лабораторных работ им можно выдать аналогичные моделированные данные, которые получаются по следующей схеме:

$$n=144000; m=5;$$

$$S = [12.2591 \quad 4.0351 \quad 6.7405 \quad 2.6970 \quad -2.3032;$$

$$4.0351 \quad 20.7811 \quad 2.7705 \quad 10.1093 \quad 8.8594;$$

$$6.7405 \quad 2.7705 \quad 16.5108 \quad 1.8950 \quad -0.7011;$$

$$2.6970 \quad 10.1093 \quad 1.8950 \quad 9.3217 \quad 4.3154;$$

$$-2.3032 \quad 8.8594 \quad -0.7011 \quad 4.3154 \quad 12.2759];$$

$$E = \text{randn}(n, m); \quad Z = \text{sqrtm}(S) * E'; \quad YY = \text{cumsum}(Z, 2);$$



Каждый студент выбирает для исследования отрезок  $Y=YY(K*n+1:(K+1)*n,:)$ , где  $K$  – его вариант (например, порядковый номер в списке группы).

### 3.1. Вычислительная процедура сингулярного разложения матриц

В ИМС MatLab имеется готовая процедура для нахождения сингулярного разложения матрицы  $X$  произвольной размерности  $\langle n \times m \rangle$  ранга  $p$ :

$$[L, S, R] = svd(X). \quad (5)$$

При таком обращении матрица  $L$  имеет размерность  $\langle n \times n \rangle$ ,  $S - \langle n \times m \rangle$ ,  $R -$  размерность  $\langle m \times m \rangle$ , все лишние элементы матрицы  $S$  заполнены нулями. При обращении

$$[L, S, R] = svd(X, 0) \quad (6)$$

(так называемая *экономная форма SVD*) матрица  $L$  имеет размерность  $\langle n \times p \rangle$ ,  $S -$  диагональная матрица  $\langle p \times p \rangle$ ,  $R -$  размерность  $\langle m \times p \rangle$ .

При отсутствии подходящего математического обеспечения SVD можно получить по следующей рекуррентной схеме:

$$L_{(0)} = [1 \dots 1]';$$

$$R^1 = L'_{(k-1)} A; \quad R_{(k)} = \frac{R}{|R|}; \quad |R| = \sqrt{r_1^2 + \dots + r_m^2}, \quad (7)$$

где

$$R^1 = [r_1 \dots r_m], \quad L^1 = [l_1 \dots l_n],$$

$$L = AR_{(k)}, \quad L_{(k)} = \frac{L}{|L|}, \quad |L| = \sqrt{l_1^2 + \dots + l_n^2},$$

$$s_{(k)} = L_{(k)} AR_{(k)}, \quad k = 1, 2, \dots,$$

до выполнения условия сходимости итераций, когда изменение сингулярного числа, полученное на последующей итерации, становится ничтожно малым:

$$|s_{(k)} - s_{(k-1)}| < \varepsilon.$$

Так получается первое сингулярное число  $s$  и соответствующие ему первые столбцы матриц  $L$  и  $R$ . На следующем этапе вводится матрица  $X - sLR^T$ , для нее по той же схеме ищется второе сингулярное число и вторые столбцы матриц  $L$  и  $R$  и т. д.

**Задача 1.1.** Для матрицы данных  $Y \langle n, m \rangle$ ,  $n = 1000$ ,  $m = 5$  получить разложение с помощью стандартной процедуры ИМС MatLab (5) и (6), проверить ожидаемые размерности и вид матриц  $L$ ,  $S$ ,  $R$ , проверить выполнение соотношения

$$X = L * S * R^T.$$

**Задача 1.2.** Разработать программный модуль, реализующий алгоритм (7) и протестировать его на матрице  $Y$ , сравнив с результатами задачи 1.1.

### 3.2. Распознавание в пространстве проекций

Пусть  $X$  – обучающая выборка и для матрицы  $X$  получено её сингулярное разложение. Для произвольного  $m$ -мерного вектора  $Z$ , подлежащего распознаванию, вычисляется его *энергия связи* с  $j$ -м столбцом  $R_j$  матрицы  $R$  (проекция  $Z$  на  $j$ -ю базисную ось):

$$w_j(Z) = \frac{1}{s_j} Z^T R_j.$$

В  $j$ -м столбце  $L_j$  матрицы  $L$  выбирается элемент  $l_i^{(j)}$ , который имеет минимальное расстояние  $d_j$  до проекции  $w_j(Z)$ :

$$d_j = \min_i |w_j - l_i|, \quad i = 1, \dots, n. \quad (8)$$

Так перебираются первые  $k$  сингулярных чисел. Обычно  $k = 3$ , но можно ограничиться и меньшим их набором. Величину

$$d = \min_i \sqrt{(w_1 - l_i^{(1)})^2 + (w_2 - l_i^{(2)})^2 + (w_3 - l_i^{(3)})^2}$$

будем называть *LR-расстоянием* между вектором  $Z$  и обучающей выборкой  $X$ . Если имеется несколько классов объектов, заданных своими обучающими выборками, то  $Z$  относится к тому классу, для которого *LR-расстояние* минимально.

Данный подход обобщается на более сложные задачи анализа многомерных измерений с использованием *LR-расстояния* вместо более традиционного расстояния Махаланобиса и др.

В молекулярной биологии основной мерой взаимодействия между молекулами является энергия связи. В иммунологии распознавание чужеродного белка (антигена)  $Z$  сводится к определению степени его связывания с известным набором антител  $X$ . В соответствии с этой аналогией рассмотренный метод называют иммунокомпьютингом.

**Задача 2.1.** В матрице  $Y$  выделяем первые 200 и последние 200 элементов – это обучающие выборки  $Y_1$  и  $Y_2$ . Построить для них сингулярные разложения.

**Задача 2.2.** Выбираем один из центральных элементов  $Y$  (например  $Y(300,:)$ ) – это вектор  $Z$ . Вычислить  $LR$ -расстояние от  $Z$  до обучающих выборок  $Y_1$  и  $Y_2$ . определить класс, к которому следует отнести  $Z$ .

**Задача 2.3.** Проиграть данный сценарий для 1–2–3 сингулярных компонент и для нескольких различных векторов  $Z$ .

### 3.3. Формирование индексов риска

*Индексом* сложной многомерной системы является общая величина, которая объединяет большое количество особых множителей (факторов) или переменных величин, называемых *индикаторами*. В некоторых случаях такой индекс является единственным путем представления текущего состояния системы и ее динамики, по которым возможно оценить активность системы и предсказать риски и тенденции. Например, риски деловой активности, такие как Dow-Jones или NASDAQ, широко используются в экономике и финансах. Как правило, такие индексы были введены на основе эмпирических соображений, некоторые из них рассчитываются достаточно легко, как среднее арифметическое последовательностей определенных переменных. К примеру, The Standard and Poor Index применяет стоимость к среднему представлению 500 акций на Нью-Йоркской фондовой бирже в течение дня. The Retail Price Index, как другой пример, измеряет среднее увеличение цены обычной сети продуктов питания в Великобритании.

Аналогичные индексы являются такими же важными, как в экономике, так и в других областях, например в медицине.

Имеются объекты, состояние каждого из которых может быть описано набором значений  $m$  индикаторов – вектором  $x = [x_1, \dots, x_m]$ . Для  $n$  таких объектов – эталонов с индикаторами  $X = [x_{ij}]$ ,  $i = 1, \dots, n, j = 1, \dots, m$  эксперты рассчитали индексы риска  $y = [y_1, \dots, y_n]^T$ . Требуется оценить вектор коэффициентов  $C = [c_1, \dots, c_m]^T$  такой, чтобы индекс риска для любого объекта можно было оценить как  $y = xC$ .

Предлагается следующий алгоритм. Рассмотрим сингулярное разложение матрицы  $X$ :

$[L, S, R] = \text{svd}(X, 0)$  (экономная форма  $SVD$ ), и перепишем его в форме

$$X = s_1 L_1 R_1^T + s_2 L_2 R_2^T + \dots + s_m L_m R_m^T.$$

Сформируем матрицу  $X^+$  в виде

$$X^+ = \frac{1}{s_1} R_1 L_1^T + \frac{1}{s_2} R_2 L_2^T + \dots + \frac{1}{s_m} R_m L_m^T.$$

**Предложение 3.1.**  $X^+ X = E$  ( $E$  – единичная матрица).

Это сразу получается за счет свойств ортогональности матриц в  $\text{svd}$ .

**Предложение 3.2.** Матрица  $X^+$  определяет решение уравнения  $y = XC$  в виде  $C = X^+ B$ .

Действительно, умножая соотношение  $C = X^+ B$  слева на  $XX^+$  и учитывая, что  $X^+ X = E$ , получаем  $X(C - X^+ B) = [0]$ .

**Предложение 3.3.** При  $n > m$   $X^+ = (X^T X)^{-1} X^T$ , т. е. решение сводится к оценке по МНК.

Если  $n > m$ , то соотношение  $y = XC$  можно транспонировать и получить, что  $X^+ = X^T (XX^T)^{-1}$ .

Если  $n = m$ , то  $X^+ = X^{-1}$ .

Матрица  $X^+$  называется псевдообратной матрицей (Мура-Пенроуза) для матрицы  $X$ . На основе сингулярного разложения, нетрудно проверить, что матрица  $X^+$  удовлетворяет следующим четырем условиям Мура-Пенроуза:

$$X \cdot X^+ \cdot X = X; \quad X^+ \cdot X \cdot X^+ = X^+; \quad (XX^+)^T = XX^+; \quad (X^+ \cdot X)^T = X^+ \cdot X.$$

Важным свойством представления  $M^+$  через компоненты сингулярного разложения является то, что задача определения оптимальных коэффициентов  $C$  индекса решается без обращения к процедурам обращения матриц, которая при больших размерностях плохо обусловлена и может приводить к накоплению ошибок.

**Задача 3.1.** Взять из матрицы  $Y$  первые 20 строк и проверить на них выполнение условий Мура-Пенроуза.

**Задача 3.2.** Сравнить процедуру на основе  $\text{svd}$  со стандартной процедурой псевдообращения матриц  $\text{pinv}$  в ИМС MatLab.

Проиллюстрировать действие псевдообратной матрицы в системе линейных уравнений  $\langle 3 \times 2 \rangle$  с точки зрения пересечения 3 прямых на плоскости.

### 3.4. Алгоритм формирования электронной цифровой подписи

При помощи средств, предоставляемых Internet, любой пользователь становится малой ячейкой во всемирной «паутине», объединившей мир. Возникает острая необходимость в разработке эффективных аппаратных или программных средств для обеспечения защиты компьютерных от несанкционированного доступа извне. Одним из таких средств безопасности передаваемых данных является аутентификация автора электронного документа и самого документа, т. е. установление подлинности автора и отсутствия изменений в полученном документе. Этой цели служит электронная цифровая подпись, которая в принципе аналогична обычной рукописной подписи и обладает ее основными достоинствами:

- удостоверяет, что подписанный текст исходит от лица, поставившего подпись;
- не дает самому этому лицу возможности отказаться от обязательств, связанных с подписанным текстом;
- гарантирует целостность подписанного текста.

Цифровая электронная подпись представляется в виде относительно небольшого количества дополнительной цифровой информации, передаваемой вместе с подписываемым текстом.

Система электронной цифровой подписи включает в себя две процедуры:

1. **процедуру постановки подписи**, в которой отправитель прежде всего вычисляет хэш-функцию  $h(M)$  подписываемого текста  $M$ . Вычисленное значение хэш-функции  $h(M)$  представляет собой один короткий блок информации, характеризующий весь текст  $M$  в целом. Затем этот блок шифруется секретным ключом отправителя. Получаемое при этом число (или пара чисел) представляет собой электронную цифровую подпись для данного текста  $M$ .
2. **процедуру проверки подписи**, при которой получатель сообщения снова вычисляет хэш-функцию  $h(M)$  принятого по каналу текста  $M$ , после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению хэш-функции.

Как видно, в обеих процедурах системы электронной цифровой подписи участвует алгоритм хеширования исходного текста  $M$ , т. е. по существу сжатия

подписываемого документа  $M$  произвольной длины до фиксированной длины в несколько десятков или сотен бит. Следует отметить, что хэш-функция должна удовлетворять целому ряду условий:

- хэш-функция должна быть чувствительна к всевозможным изменениям в тексте  $M$ , таким как вставки, выбросы, перестановки и т. п.;
- хэш-функция должна обладать свойством необратимости, т. е. задача подбора документа  $M^*$ , который обладал бы требуемым значением хэш-функции, должна быть вычислительно неразрешима;
- вероятность того, что значения хэш-функций двух различных документов (вне зависимости от их длин) совпадут, должна быть ничтожно мала.

Вычислительный алгоритм безопасного хэширования на основе использования сингулярных разложений (иммунокомпьютинга) состоит из следующих шагов:

*Шаг 1.* Представим текстовую информацию исходного текста  $M$  в виде множества отдельных букв, т. е. в виде  $I = \{i_1, i_2, \dots, i_k\}$  и поставим в соответствие каждой букве  $i_k$  определенную цифру  $a_k$ , т. е. вместо буквенного множества  $M$  в результате получим цифровой ряд вида  $A = \{a_1, \dots, a_k\}$ .

*Шаг 2.* Сформируем определенным образом из ряда матрицу  $A$  размерности  $\langle n \times m \rangle$ .

*Шаг 3.* Сгенерируем случайным образом набор цифр  $b = [b_1, \dots, b_k]$ . В качестве примера возьмем формулу генератора случайных чисел вида:  $b_k = (p+k)^2/100$ , т. е. получим цифровой ряд вида  $\{b_1, \dots, b_k\}$ .

*Шаг 4.* Сформируем из этого ряда матрицу  $B$  одинаковой с матрицей  $A$  размерности  $\langle n \times m \rangle$ .

*Шаг 5.* Осуществим сингулярное разложение матрицы  $A$ , т. е. определим множества сингулярных чисел, правых и левых сингулярных векторов:  $[L, S, R] = svd(A)$ .

*Шаг 6.* Результатом хэширования будет величина  $\omega = L_1^T B R_1$ .

После применения указанного алгоритма, следуя процедуре постановки подписи, автор электронного документа передает получателю тройку  $(M, l, p)$ ,

где  $l$  представляет собой электронную цифровую подпись, а  $p$  – открытый ключ, который позволяет однозначно восстановить матрицу  $B$ .

Пусть, например, сообщение  $M$  длиной в 64 знака преобразовано в матрицу  $A$  размерности  $(8 \times 8)$  и имеет вид

$$A = \begin{vmatrix} 149 & 18 & \dots & 158 \\ 82 & 149 & \dots & 18 \\ \dots & \dots & \dots & \dots \\ 53 & 152 & \dots & 82 \end{vmatrix}.$$

Взяв, например,  $p = 5$ , сгенерируем ряд  $b$  и трансформируем его в матрицу  $B$   $(8 \times 8)$ :

$$B = \begin{vmatrix} 0,25 & 0,36 & \dots & 1,44 \\ 1,69 & 1,96 & \dots & 4 \\ \dots & \dots & \dots & \dots \\ 37,21 & 38,44 & \dots & 46,24 \end{vmatrix}.$$

Применяя шаги 1–6, получим:  $\omega = 1656,77$ .

Согласно процедуре проверки подписи получатель электронного документа, в свою очередь, по заранее согласованной с автором и переданной по защищенному каналу формуле генерирования матрицы  $B$  и алгоритму хэширования восстанавливает для пришедшей матрицы  $MM$  значение  $L_1^T B R_1$  и сравнивает его с пришедшим значением  $\omega$ . Если они совпадают, то можно с уверенностью сказать, что электронный документ дошел по телекоммуникационному каналу до адресата в неискаженном злоумышленником виде и электронная цифровая подпись автора подлинная.

Для обеспечения наилучшей защищенности передаваемой информации исходный текст  $M$  и значение хэш-функции  $l$  тоже можно зашифровать при помощи известных, действующих криптографических методов.

**Задача 4.1.** Сформировать ряд из 64 случайных чисел, домножить его на 100 и округлить до целых. Так получается матрица-сообщение  $A$ . Построить для неё сингулярное разложение.

**Задача 4.2.** По данному открытому ключу  $p$  сформировать матрицу  $B$ . Построить хэш-функцию для матрицы  $A$ . Проанализировать, как изменяется хэш-функция при изменении нескольких символов в матрице-сообщении.

## 4. КЛАСТЕРНЫЙ АНАЛИЗ

### 4.1. Кластеризация. Выбор метрики

Общая постановка задачи *кластерного анализа (сегментации, таксономии)* состоит в разделении многомерной выборки на  $k$  компактных классов так, чтобы в некоторой метрике расстояния между элементами каждого класса были малы по сравнению с расстояниями между классами. Если число классов заранее неизвестно, задачу приходится решать в интерактивном режиме.

**Типовые метрики:**

- ‘Euclid’ – евклидово расстояние;
- ‘SEuclid’ – нормализованное евклидово расстояние;
- ‘Mahal’ – расстояние Махаланобиса;
- ‘CityBlock’ или ‘Hamming’ – расстояние по Манхэттену (расстояние Хэмминга).

Еще рассматривают:

- ‘Minkovski’ – расстояние Минковского (зависит от показателя  $p$ , по умолчанию  $p = 2$ );
- ‘cosine’ – единица минус косинус угла между векторами – элементами выборки;
- ‘correlation’ – единица минус выборочный коэффициент корреляции между векторами – элементами выборки;
- ‘spearman’ – единица минус выборочный коэффициент ранговой корреляции Спирмена;
- ‘jaccard’ единица минус коэффициент Жакарда (процент ненулевых различных значений компонент векторов);
- ‘chebychev’ – расстояние Чебышева (максимальная разность между координатами векторов).

### 4.2. Метод $k$ средних и EM-алгоритм

1)  $n$  исходных объектов выбирается  $k$  объектов, которые объявляются центрами классов. Их можно выбирать:



- случайным образом;
- можно выбирать самые далекие друг от друга объекты;
- можно брать пары объектов, дальше всего разнесенные друг от друга по разным координатным осям и т. п.

2) Сканируется весь набор исходных объектов: очередной объект присоединяется к  $j$ -му классу, если его расстояние до центра этого класса минимально.

3) Определяются новые центры сформированных классов и сканирование повторяется заново.

Процесс останавливается либо, когда стабилизируются центры классов, либо после заданного числа итераций. Такой подход называют **алгоритмом Мак-Кина**.

Вместо расстояния до центра можно использовать среднее арифметическое (или медиана) расстояний до всех объектов класса. Такой подход называют **методом динамических сгущений**.

Присоединение очередного объекта к  $j$ -му классу корректирует его центр и ковариационную матрицу. Эта корректировка вычисляется по рекуррентным формулам:

$$\hat{a}(n+1) = \frac{1}{n+1}[n\hat{a}(n) + X_{n+1}] = \hat{a}(n) + \frac{1}{n+1}[X_{n+1} - \hat{a}(n)];$$

$$\hat{\Sigma}(n+1) = \frac{1}{n+1}[v\hat{\Sigma}(n) + X_{n+1}^o \cdot X_{n+1}^{oT}] = \hat{\Sigma}(n) + \frac{1}{n+1}[X_{n+1}^o \cdot X_{n+1}^{oT} - \hat{\Sigma}(n)].$$

Начальные условия имеют вид  $\hat{a}(1) = X_1$ ,  $\hat{\Sigma}(1) = X_1^o \cdot X_1^{oT}$ . Выбирается тот класс, который после этой корректировки оказывается наиболее компактным: контролируется либо максимальное расстояние между элементами этого нового класса, либо максимальное собственное число матрицы  $\hat{\Sigma}$ , либо ее определитель. Этот алгоритм можно интерпретировать как вычисление апостериорных вероятностей принадлежности объекта тому или иному классу и выбор класса с наибольшей апостериорной вероятностью. Для этого подхода разработана эффективная численная реализация – **EM-алгоритм**.

### 4.3. Иерархическая кластеризация на основе дендрограммы

Функция  $Z = \text{linkage}(Y, \text{'method'})$  возвращает иерархическое дерево кластеров. Ее аргументами является вектор  $Y$ , возвращаемый функцией  $\text{pdist}$ . Строка  $\text{'method'}$  задает метод кластеризации:

- $\text{'single'}$  – алгоритм ближайшего соседа;
- $\text{'complete'}$  – алгоритм дальнего соседа;
- $\text{'average'}$  – алгоритм средней связи;
- $\text{'centroid'}$  – центроидный метод – по центрам тяжести групп;
- $\text{'ward'}$  – пошаговый алгоритм.

На первом шаге ближайшие объекты объединяются парами, и каждая найденная пара рассматривается как новый объект. Каждый следующий объект присоединяется или к одному из исходных, или к одной из ранее образованных групп, группы могут объединяться. Матрица  $Z$  имеет  $(m - 1)$  строку и 3 столбца. Первые 2 столбца – номера объединяемых объектов, третий – расстояние между ними.

Функция  $\text{dendrogram}(Z)$  создает графическое отображение полученного дерева кластеров (рис. 4.1).

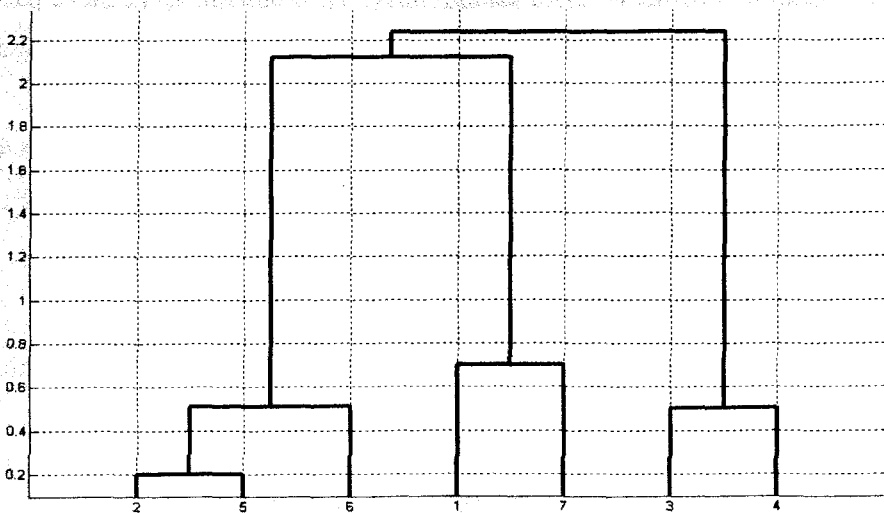


Рис. 4.1. Дендрограмма – графический вывод

Функция `corphenet (Z, Y)` возвращает аналог коэффициента корреляции, характеризующий качество разбиения: чем ближе к 1, тем лучше.

Функция `inconsistent (Z)` возвращает коэффициенты несовместимости для каждого уровня дерева и тоже характеризует качество разбиения.

Функция `T = cluster (Z, cutoff)` или `T = cluster (Z, cutoff, depth)` объединяет все перечисленные функции. Здесь

- $Z$  – матрица, возвращаемая функцией `linkage`;
- если параметр `cutoff` имеет значение от 0 до 1, то он задает порог для коэффициентов несовместимости, при достижении которого кластер считается сформированным;
- если параметр `cutoff` – целое число  $>1$ , то он задает число кластеров;
- `depth` – промежуточный параметр, указывающий, сколько уровней просматривается при нахождении коэффициентов несовместимости;
- $T$  – столбец длины  $n$  с номерами кластеров, к которым приписан каждый объект.

Функция `T = clusterdata (X, cutoff)` объединяет все перечисленные функции.

#### 4.4. Оценка качества разделения

Качество разбиения обычно характеризуют отношением среднего расстояния между центрами классов к среднему расстоянию элементов внутри каждого класса от его центра.

В версиях ИМС MatLab, начиная с 7.x, этой цели служит специальная функция `silhouette (X0, T, 'distance', 'sqEuclidean')`. Результат вычислений с использованием этой функции приведен на рисунке 4.2.

#### 4.5. Кластер-анализ

**Задача 4.1.** Сформировать на плоскости 3 облака точек с различными центрами и ковариационными матрицами.

**Задача 4.2.** Провести кластерный анализ средствами библиотеки ИМС MatLab.

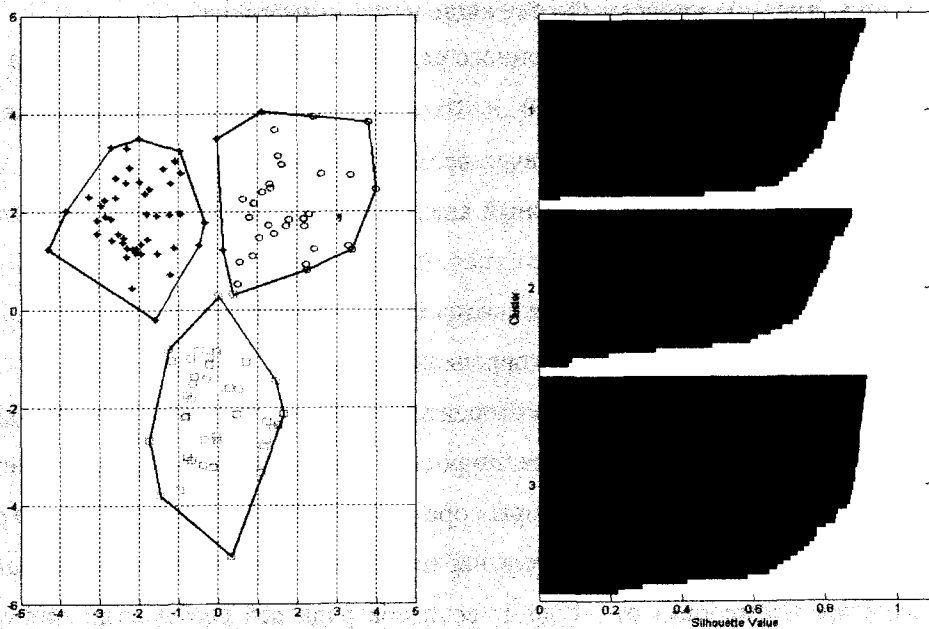


Рис. 4.2. Результат вычислений с использованием функции *silhouette*

**Задача 4.3.** Произвести визуализацию результатов (рис. 4.3), используя средства библиотеки ИМС MatLab и функцию *convhull* (или *delonay*)

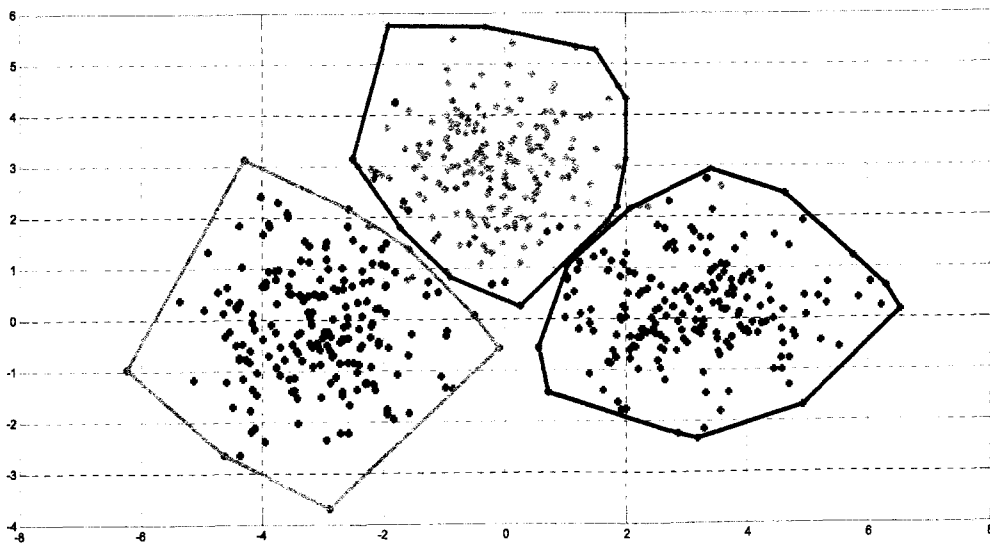


Рис. 4.3. Результат кластеризации (метод  $k$  средних *kmeans*)

#### 4.6. Снижение размерности за счет выделения компонент

Разделим свой отрезок эталонного ряда  $Y$  на 2 части и сформируем из них одну матрицу размерности  $\langle 500 \times 10 \rangle$ . Вычислим попарные корреляции 10 компонент и используем их в качестве мер близости компонент.

**Задача 1.** Провести кластерный анализ, разбив 10 компонент на 2 или 3 однородных группы.

**Задача 2.** Провести такой же анализ 10 компонент:

а) на основе коэффициента корреляции;

б) на основе коэффициентов относительных приростов (Ю. Лукашин).

Коэффициент относительных приростов  $r$  для 2 рядов вычисляет разность между числом интервалов, на которых оба ряда имеют приращения одного знака и числом интервалов, на которых наблюдаются приращения разных знаков. Проводится нормировка от  $-1$  до  $1$ : если оба ряда всё время возрастают, то  $r = 1$ , если они всё время убывают,  $r = -1$ .

Результаты кластеризации 10 компонент по коэффициентам относительных приростов:

<i>Cophenet</i>	0.4343
Кластер 1	[1 5 8 10]
Кластер 2	[2 3 6]
Кластер 3	[4 7]

Результаты кластеризации 10 компонент по коэффициентам корреляции:

<i>Cophenet</i>	0.3728
Кластер 1	[3 9]
Кластер 2	[2 5 6 10]
Кластер 3	[1 4 7 8]

В принципе это означает, что обработку, например, анализ главных компонент, можно проводить внутри каждого кластера, а потом сравнивать между собой полученные обобщенные результаты.

## 5. ПРОГНОЗНАЯ АНАЛИТИКА

### 5.1. Прогнозирование

*Прогноз* – это научно обоснованное суждение о возможных состояниях объекта в будущем.

Сегодня разрабатываются методы прогнозирования, использующие положения теории хаоса и фракталов. В отличие от «мягких» алгоритмов, они пока мало проработаны как с теоретической точки зрения, так и в плане практической реализации. Отдельные моменты иногда применяются при анализе финансовых рынков: трейдеры, как правило, первыми испытывают все новые методы прогнозирования. В результате могут быть получены методы довольно точного прогнозирования резких и внезапных изменений: экономических кризисов, скачкообразной динамики спроса, банкротств и т. д.

### 5.2. Классификация методов прогнозирования

Чтобы получить общее представление о методах прогнозирования, необходимо для начала классифицировать эти методы. Их принято разделять на количественные и качественные. Далее они различаются

по горизонту прогноза:

- краткосрочные (как правило, в пределах года или нескольких месяцев);
- среднесрочные (несколько лет);
- долгосрочные (более пяти лет);

по типу прогнозирования:

- эвристические (использующие субъективные данные, оценки и мнения);
- поисковые (в свою очередь делятся на экстраполяционные, проецирующие прошлые тенденции в будущее, и альтернативные, учитывающие возможности скачкообразной динамики явлений и различные варианты их развития);
- нормативные (оценка тенденций проводится исходя из заранее установленных целей и задач);

по степени вероятности событий:

- варианты (подразумевают вероятностный характер будущего и предлагают несколько сценариев развития событий);
- инвариантные (предполагается единственный сценарий);

по способу представления результатов:

- точечные (прогнозируется точное значение показателя);
- интервальные (прогнозируется диапазон наиболее вероятных значений);

по степени однородности: простые и комплексные (сочетают в себе несколько взаимосвязанных простых методов);

по характеру базовой информации:

- фактографические (основываются на имеющейся информации о динамике развития явления или объекта, бывают статистическими и опережающими);
- экспертные (индивидуальные и коллективные, в зависимости от числа экспертов);
- комбинированные (использующие разнородную информацию).

### 5.3. Временные ряды

*Временной ряд* представляет собой набор данных, описывающих объект в последовательные равноотстоящие моменты времени. Если исходные данные относятся к различным моментам времени, традиционный подход состоит в аппроксимации данных кубическим сплайном и использовании интерполированных отсчетов на равномерной сетке.

Для оценивания качества прогноза один из основателей прогностики Г. Тейл предложил использовать коэффициент расхождения (или коэффициент несоответствия), представляющий собой отношение среднеквадратической ошибки прогноза и среднеквадратической оценки рассеяния исходного ряда. Методы прогнозирования по своему информационному основанию делятся на три класса.

– *Фактографические методы* базируются на имеющемся информационном материале об объекте прогнозирования и его прошлом развитии.

– *Экспертные методы* базируются на информации, обеспечиваемой систематизированными процедурами выявления и обобщения мнений специалистов-экспертов.

– К *комбинированным* относятся методы со смешанной информационной основой, использующие как фактографическую, так и экспертную информацию.

В действительности, любой прогноз использует экспертную информацию, хотя бы в части предположений о неизменности условий протекания изучаемого процесса на каком-то временном отрезке в будущем.

Наиболее распространенными и разработанными при фактографическом прогнозировании являются методы *экстраполяции тенденций*, в основе которых лежит предположение о том, что рассматриваемый процесс изменения переменной  $x(t)$  представляет собой сочетание нескольких составляющих, регулярных и случайных:

$$x(t) = \sum_{i=1}^r f_i(t) + \xi(t). \quad (1)$$

Считается, что регулярные составляющие  $f_i(t)$  представляют собой достаточно гладкие функции от аргумента  $t$  (в большинстве случаев – времени), которые сохраняют свой вид на промежутке упреждения процесса. Они отвечают интуитивному представлению о какой-то очищенной от помех сущности исследуемого процесса. Сумма регулярных составляющих образует тренд исследуемого процесса. Экстраполяционные методы прогнозирования делают основной упор на выявление наилучшего в том или ином смысле описания тренда и получение прогнозных значений путем его экстраполяции.

Регулярную часть ряда оценивают в виде разложения по некоторому ортогональному базису. Этот базис обычно стараются задать на основе априорных предположений о природе изучаемого процесса, но наибольшую ценность имело бы использование базиса, непосредственно порождаемого самим исходным временным рядом. В настоящее время наибольший интерес вызывают методы, основанные на разложениях в первом сингулярном базисе (*иммунокомпьютинг*), который, как утверждается, позволяет сформировать слагаемые в



регулярной части (1) способом, в наибольшей степени соответствующим внутренней структуре имеющегося ряда.

Наиболее распространенным вариантом представления (1) является разложение Юла. Оно включает медленную регулярную составляющую (собственно тренд), периодические компоненты с известными, физически обусловленными периодами – сезонные составляющие, периодические компоненты с периодами, определяемыми непосредственно из данных, и чисто случайную составляющую, идентифицируемую как реализация некоторого стационарного случайного процесса.

Хаотические временные ряды характеризуются наличием временного тренда, детерминированного или случайного, глобального или локальных. Обнаружение трендов и их правильный учет в структуре модели представляют собой важную задачу в анализе временных рядов.

#### 5.4. Множественная регрессия

Пусть теперь имеется случайный вектор  $Z = [x_1, \dots, x_k, y_1, \dots, y_r]^T$ , который подчиняется  $(k + r)$ -мерному нормальному закону  $N_{k+r}(a, \Sigma)$  с известными вектором средних  $a$  и ковариационной матрицей  $\Sigma$ . Рассмотрим случай, когда первые  $k$  компонент  $x_1, \dots, x_k$  вектора  $Z$  наблюдаются в эксперименте, а оставшиеся компоненты  $y_1, \dots, y_r$  являются ненаблюдаемыми. Требуется получить оценки ненаблюдаемых компонент.

**Пример 1.** В лабораторных условиях текущее состояние смазочных материалов контролируется по 16 параметрам и имеется достаточная база данных для оценки их средних и их ковариационной матрицы. В полевых условиях экспресс-анализ позволяет проконтролировать только 7 параметров. На основе знания значений этих 7 параметров требуется оценить оставшиеся 9 показателей.

**Пример 2.** Для технологической установки имеется база данных, содержащая значения выходных параметров (продукта) при различных комбинациях входных и управляющих параметров. Предположим, что эта база достаточна для оценивания средних и ковариационной матрицы всего вектора контролируемых показателей.

а) Заданы текущие значения входных (сырья) и управляющих параметров. Требуется получить прогноз вектора выходных параметров установки (продукта).

б) Заданы требуемые значения выходных параметров и используемые в настоящий момент значения управляющих параметров. Требуется сформулировать требования к входным параметрам (сырью).

в) Заданы текущие значения входных параметров (свойства сырья). Требуется найти комбинацию управляющих параметров, оптимизирующих целевую функцию, которая отражает свойства выходных показателей (например, в единицах стоимости).

Согласно поставленным условиям, средние и ковариационная матрица вектора  $Z$  имеют блочную структуру:

$$a = \begin{bmatrix} a_x \\ a_y \end{bmatrix}; \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix}.$$

Будем сразу рассматривать центрированные величины:

$$\hat{X} = X - a_x = [x_1 - a_{x1}, \dots, x_k - a_{xk}]^T; \quad \hat{Y} = [y_1 - a_{y1}, \dots, y_r - a_{yr}]^T.$$

Требуется определить матрицу  $C$  размерности  $r \times k$ , доставляющую минимум функции потерь  $G(C)$ , которая представляет собой сумму дисперсий погрешностей прогноза:

$$\begin{aligned} G(C) &= \text{tr}\{M[(C\hat{X} - \hat{Y})^T(C\hat{X} - \hat{Y})]\} = M\{\text{tr}[(C\hat{X} - \hat{Y})(C\hat{X} - \hat{Y})^T]\} = \\ &= \text{tr}\{CM(\hat{X}\hat{X}^T)C^T - M(\hat{Y}\hat{X}^T)C^T - CM(\hat{X}\hat{Y}^T) + M(\hat{Y}\hat{Y}^T)\} = \\ &= \text{tr}\{C\Sigma_{xx}C^T - \Sigma_{xy}^T C^T - C\Sigma_{xy} + \Sigma_{yy}\} \rightarrow \min. \end{aligned} \quad (2)$$

В проведенных преобразованиях использован тот факт, что если оба матричных произведения  $AB$  и  $BA$  имеют смысл, то  $\text{tr}(AB) = \text{tr}(BA)$ .

Функция матричного аргумента  $G(C)$  является выпуклой и имеет единственный экстремум – минимум. Вычислим ее производную по матрице  $C$  и приравняем ее нулю, используя очевидное соотношение  $\frac{\partial}{\partial A} \text{tr}(A) = I$ . При дифференцировании нужно все время иметь в виду, что производная матричной функции по матрице  $C$  размерности  $r \times k$  должна иметь тот же размер  $r \times k$ .

$$\frac{\partial G}{\partial C} = 2C\Sigma_{xx} - 2\Sigma_{xy}^T = 0 \Rightarrow \hat{C} = \Sigma_{xy}^T \cdot \Sigma_{xx}^{-1}.$$

Возвращаясь к исходным величинам  $X$ ,  $Y$ , получаем *уравнение множественной линейной регрессии  $Y$  на  $X$* :

$$\hat{Y} = a_y + \Sigma_{xy}^T \Sigma_{xx}^{-1} (X - a_x). \quad (3)$$

Ковариационную матрицу ошибок прогноза в компактной форме представить не удастся, но можно вычислить ее след (упрощение достигается за счет того, что  $tr(AB) = tr(BA)$ ):

$$\begin{aligned} G(\hat{C}) &= tr\{M[(C\hat{X} - \hat{Y})^T(C\hat{X} - \hat{Y})]\} = \\ &= tr\{\hat{C}\Sigma_{xx}\hat{C}^T - \Sigma_{xy}^T \hat{C}^T - \hat{C}\Sigma_{xy} + \Sigma_{yy}\} = tr\{\Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xx} \Sigma_{xx}^{-1} \Sigma_{xy} - 2\Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy} + \Sigma_{yy}\} = \\ &= tr\{\Sigma_{yy} - \Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy}\} = tr\{\Sigma_{yy}[I - \Sigma_{yy}^{-1} \Sigma_{xy}^T \Sigma_{xx}^{-1} \Sigma_{xy}]\} \end{aligned} \quad (4)$$

На формуле (4) основаны различные определения множественного коэффициента корреляции.

Аналогичная техника позволяет решить, например, такую задачу. Имеется одна ненаблюдаемая компонента. Требуется построить линейную комбинацию наблюдаемых компонент, в максимальной степени коррелированную с данной ненаблюдаемой.

**Пример 3.** Требуется спрогнозировать биржевой курс акций данной компании по курсам ряда других компаний с некоторым отставанием во времени.

## 5.5. Адаптивная модель множественной регрессии

Классический регрессионный анализ опирается на гипотезу о возможности аппроксимации изучаемого процесса линейным уравнением с постоянными коэффициентами. Эти коэффициенты отражают степень связи различных переменных с изучаемой величиной. В реальной жизни сила взаимодействия переменных не остается неизменной, как не остается неизменной и внешняя среда, в которой развивается исследуемый процесс. Таким образом, множественная регрессия с постоянными коэффициентами имеет ограниченное применение и желательно найти способ корректировки, обновления ее коэффициентов. Это открывает возможность исследовать направление и характер эволюции взаимосвязей.

Рассмотрим способ адаптации коэффициентов множественной регрессии. Предположим, что исследуется связь ряда  $y$  с рядом  $x(t)$  и что оценку значения  $y(t + \tau)$  можно получить как взвешенную сумму вида

$$\hat{y}(t + \tau) = \sum_{i=0}^p a_i(t)x(t-i), \quad \tau \geq 0.$$

Это уравнение множественной регрессии. В случае, когда  $\tau = 0$ , будем решать задачу чистого анализа эволюции коэффициентов связи  $a_i(t)$ . При  $\tau > 0$  – задачу анализа эволюции коэффициентов множественной регрессии и прогнозирования на  $\tau$  шагов вперед на основе текущей информации. Сравнивая оценки  $\hat{y}(t + \tau)$  с фактической точкой ряда  $y(t + \tau)$ , можем вычислить ошибку:

$$\hat{e}(t + \tau) = y(t + \tau) - \hat{y}(t + \tau) = y(t + \tau) - \sum_{i=0}^p a_i(t)x(t-i),$$

и на основе полученного результата произвести корректировку коэффициентов  $a_i(t)$ . Для адаптации коэффициентов  $a_i(t)$  воспользуемся методом наискорейшего спуска, т. е. обновление весов будем осуществлять по следующему правилу:

$$A_n = A_c - h \text{grad}(\hat{e}^2(t + \tau)),$$

где  $A_c$  – вектор старых коэффициентов;  $A_n$  – вектор новых коэффициентов;  $h > 0$ .

Используя выражение для  $\hat{e}(t + \tau)$ , находим элементы градиента:

$$\frac{\partial \hat{e}^2(t + \tau)}{\partial a_i(t)} = 2\hat{e}^2(t + \tau) \frac{\partial \hat{e}(t + \tau)}{\partial a_i(t)} = -2\hat{e}(t + \tau) \cdot x(t-i).$$

Таким образом, корректировка коэффициентов должна осуществляться по правилу:

$$A_n = A_c + 2h X(t), \quad X(t) = [x(t), x(t-1), \dots, x(t-p)].$$

Неизвестным в этом выражении остается лишь значение коэффициента  $h$ , определяющего скорость движения в направлении, обратном градиенту. Обычно принимают

$$h = \frac{\alpha}{\sum_{i=0}^p x^2(t-i)}, \quad 0 < \alpha < 1.$$

$\alpha$  называют параметром адаптации и считают его постоянным для данной модели. Оптимальное значение  $\alpha$  можно определить методом проб, т. е. в процессе «обучения» модели.

Следует сказать несколько слов о проблеме мультиколлинеарности. Мультиколлинеарность, т. е. корреляция между независимыми переменными уравнения, имеет место тогда, когда существуют линейные соотношения между экзогенными переменными. Сильная мультиколлинеарность часто возникает при введении в уравнение лаговых переменных. В нашем случае мультиколлинеарность проявляется в ухудшении процесса адаптации. Это приводит к тому, что оценки параметров могут значительно исказить представление о реальной структуре объекта в текущий момент времени. Перед построением модели адаптивной множественной регрессии рекомендуем строить обычную множественную регрессию методом наименьших квадратов. Это помогает на начальном этапе моделирования определить структуру уравнения множественной регрессии, отобрать переменные.

## 5.6. Прогнозирование МВР

Рассмотрим подробнее процесс прогнозирования многомерного временного ряда (МВР) на основе алгоритма множественной регрессии. Пусть имеется случайный вектор  $Z = [x_1, \dots, x_k, y_1, \dots, y_r]^T$ , который подчиняется  $(k + r)$ -мерному нормальному закону  $N_{k+r}(a, \Sigma)$  с известными вектором средних  $a$  и ковариационной матрицей  $\Sigma$ . Рассмотрим случай, когда первые  $k$  компонент  $x_1, \dots, x_k$  вектора  $Z$  наблюдаются в эксперименте, а оставшиеся компоненты  $y_1, \dots, y_r$  являются ненаблюдаемыми. Требуется получить оценки ненаблюдаемых компонент.

Согласно поставленным условиям, средние и ковариационная матрица вектора  $Z$  имеют блочную структуру:

$$a = \begin{bmatrix} a_x \\ a_y \end{bmatrix}; \quad \Sigma = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_{yy} \end{bmatrix}. \quad (5)$$

Требуется определить матрицу  $C$  размерности  $r \times k$ , доставляющую минимум функции потерь  $G(C)$ , которая представляет собой сумму дисперсий погрешностей прогноза:

$$G(C) = \text{tr} \{ M[(CX - \hat{Y})^T(CX - \hat{Y})] \} \rightarrow \min.$$

Вычисляя производную  $G(C)$  по матрице  $C$  и приравнивая её нулю, получаем *уравнение множественной линейной регрессии  $Y$  на  $X$* :

$$\hat{Y} = a_y + \Sigma_{xy}^T \Sigma_{xx}^{-1} (X - a_x).$$

Пусть теперь  $X$  – отрезок  $m$ -мерного временного ряда длиной  $n$ . Требуется получить на основе процедуры множественной регрессии прогноз компонент ряда на  $r$  шагов вперед.

**Задача 6.1.** Выделим в своём отрезке эталонного ряда  $Y$  отрезок шириной  $n = 200 + r$ ,  $r$  – глубина прогнозирования. Получается матрица  $X$  размерности  $\langle (200 + r) \times 5 \rangle$ . Выберем для прогнозирования, например, первую компоненту  $Y$ . Присоединим к матрице  $X$  вектор  $y = Y(1 + r : n + r, 1)$ . Получим блочную матрицу  $Z = [X(1 + r : n + r, :) \ y]$  размерности  $\langle n \times 6 \rangle$ . Оценим её ковариационную матрицу и разделим её на блоки в соответствии с (5). Среднее  $[a_x \ a_y]$  находится как  $mean(Z)$ , но  $a_y$  выгоднее считать по более короткому промежутку, например  $a_y = mean(y(n - 20 : n + r))$ . Требуется разработать программный модуль (процедуру-сценарий) для получения прогноза на заданное число шагов  $r$ .

**Задача 6.2.** Вывести на экран графики прогнозов при разных  $r$  и графики реального поведения процесса на участке  $(n + r + 1 : n + 2 * r)$ .

**Задача 6.3.** Собрать в один модуль прогнозы для всех 5 компонент и представить их в графическом виде (рис. 5.1).

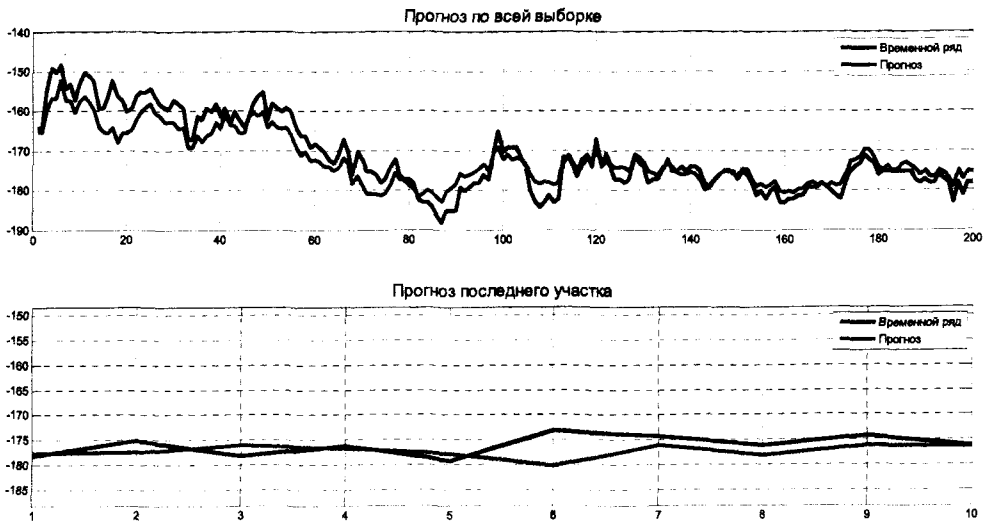


Рис. 5.1. Прогноз по всей выборке и по последнему участку

## 5.7. Прогнозирование МВР в пространстве проекций

**Задача 7.1.** Построить для матрицы  $X$  из работы сингулярное разложение и определить, какое число  $k$  сингулярных проекций содержат более 90% информации. Разработать процедуру-функцию для формирования этих  $k$  проекций.

**Задача 7.2.** Ввести эту процедуру в программный модуль, получить прогнозы в пространстве проекций и вернуться от них в пространство исходных переменных.

## 5.8. Анализ сингулярных спектров

На сингулярных разложениях основан оригинальный алгоритм сглаживания и прогнозирования – метод «Гусеница». При решении задачи выявления структуры ряда и, в частности, выделения в нем периодических составляющих, данный подход известен как *анализ сингулярных спектров (SSA, Singular Spectra Analysis)*.

В теории одномерных временных рядов ряду  $x_1, \dots, x_N$  ставится в соответствие *матрица пошагового движения*

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_M \\ x_2 & x_3 & x_4 & \dots & x_{M+1} \\ x_3 & x_4 & x_5 & \dots & x_{M+2} \\ \dots & \dots & \dots & \dots & \dots \\ x_k & x_{k+1} & x_{k+2} & \dots & x_{M+k+1} \end{bmatrix}, \quad M+k+1=N.$$

В ее сингулярном разложении  $[L, S, R] = \text{svd}(X)$  столбцы матрицы  $R$  – базисные векторы, столбцы  $L$  – сингулярные компоненты  $L_1, \dots, L_k$  ряда  $x_1, \dots, x_N$ , диагональные элементы матрицы  $S$  соответствуют вкладам этих компонент в исходный временной ряд. Таким образом, сингулярное разложение матрицы  $X$  порождает набор линейных фильтров, настроенных на составляющие исходного ряда, а базисные векторы выступают в роли весовых функций этих фильтров.

Для их получения строят матрицы  $Y_j = L_j R_j^T$  и усредняют значения элементов на их побочных диагоналях. Признаком наличия периодической составляющей является *появление пары равных сингулярных чисел*. Найденную периоди-

ческую составляющую либо можно вывести на экран как сумму соответствующих усредненных значений, либо – на фазовую плоскость этих двух значений.

Процесс выделения главных сингулярных компонент носит, в общем случае, интерактивный характер, поскольку, например, важные периодические составляющие могут быть связаны с малыми значениями сингулярных чисел и теряться среди компонент, относимых к случайной части.

Серьёзный недостаток метода состоит в том, что он полностью *игнорирует фазу периодических составляющих*, что затрудняет «склежку» для целей, например, прогнозирования.

**Задача 8.1.** Смоделировать временной ряд, содержащий 2 периодические компоненты на фоне белого шума. Частота дискретизации 100 Гц.

**Задача 8.2.** Построить для этого ряда матрицу пошагового движения и идентифицировать сингулярные компоненты, соответствующие периодическим составляющим.

**Задача 8.3.** Вывести эти составляющие на экран (рис. 5.2–5.5):

- а) на фоне отрезка ряда, выровняв их по дисперсии;
- б) на фазовой плоскости.

**Задача 8.4.** Провести спектральный анализ выделенных составляющих.

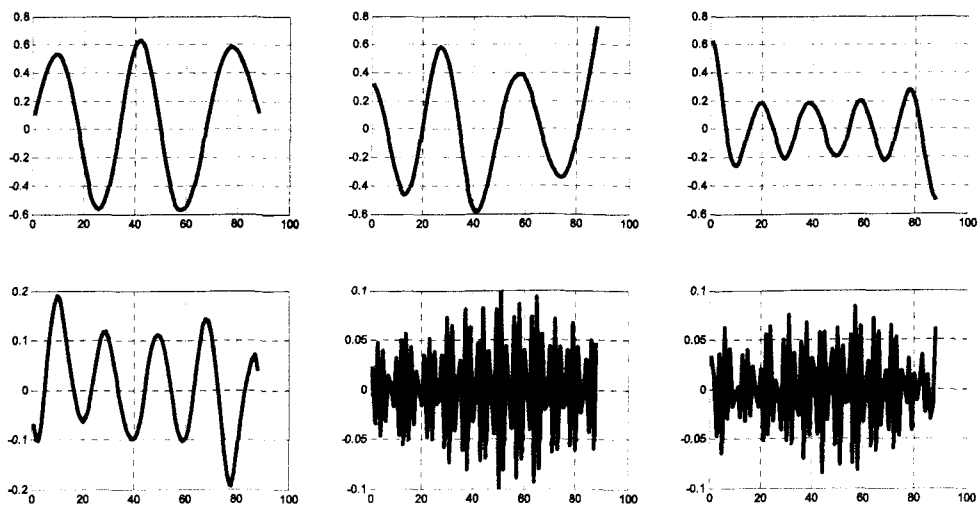


Рис. 5.2. Первые 6 сингулярных компонент (после усреднения по диагоналям)



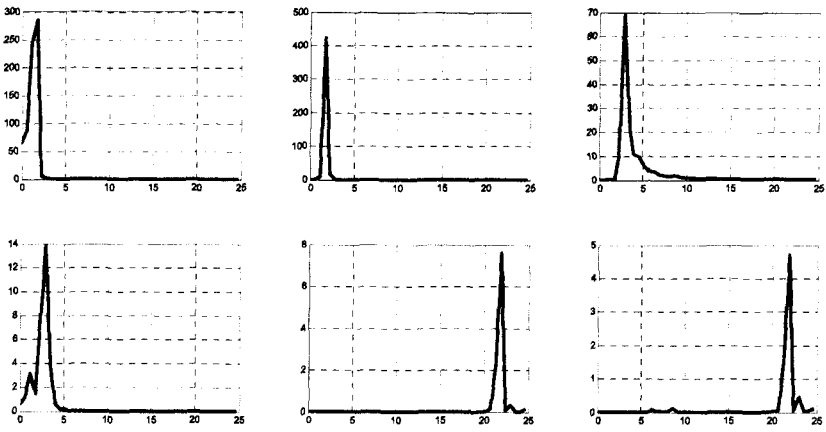


Рис. 5.3. СПИМ

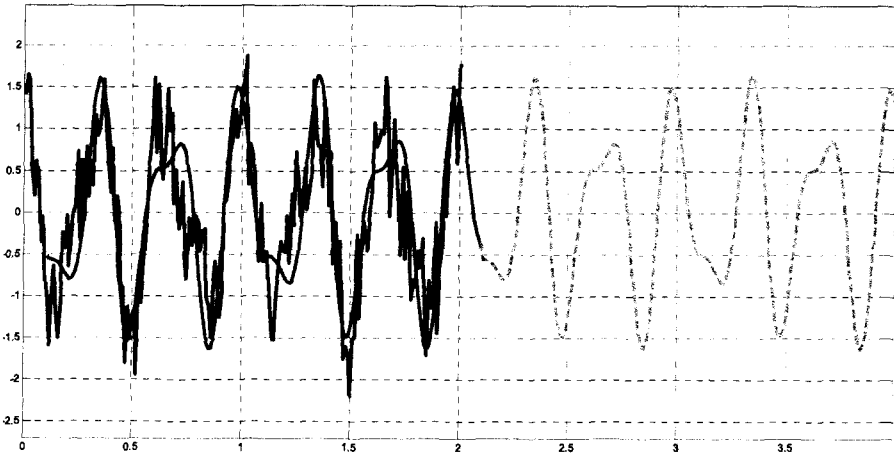


Рис. 5.4. Временной ряд, выделенные гармоники, прогноз

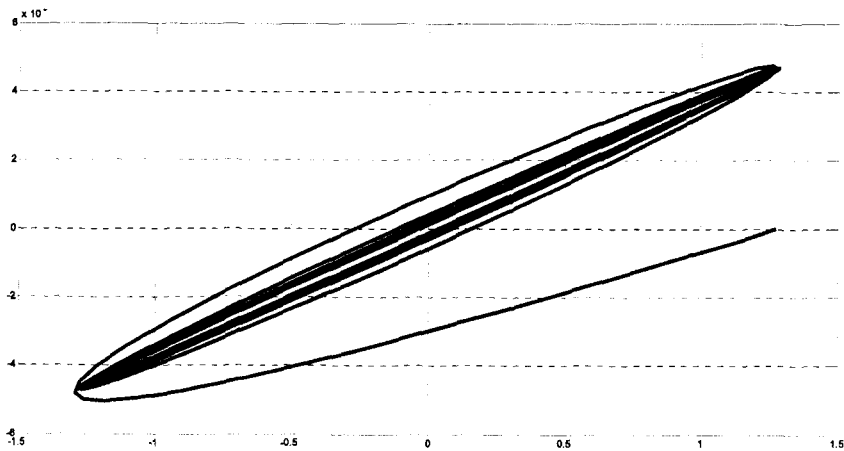


Рис. 5.5. Выделенная гармоника на фазовой плоскости

## 5.9. Прецедентный анализ

Если требуется сделать прогноз вперед с какого-то участка  $X$  временного ряда, можно поискать в базе данных похожий отрезок, посмотреть его продолжение и использовать это продолжение в качестве прогноза. Главная проблема – как выбрать меру схожести, адекватную природе рассматриваемого ряда.

**Задача 9.1.** Разбить эталонный ряд  $Y <1000 \times 5>$  на отрезки (окна) по 200 измерений со сдвигом 20 шагов. Получается 40 окон, рассматриваемых в качестве возможных прецедентов, окно  $X$ , с которого делается прогноз, и последнее окно измерений для оценки качества получившегося прогноза.

**Задача 9.2.** В качестве меры близости предлагается использовать стандартные статистики многомерного дисперсионного анализа (MANOVA) для ряда первых конечных разностей:

- статистику Уилкса  $W(k) = \log \left( \frac{|\hat{S}(k)|}{|\hat{\Sigma}|} \right)$ ;
- статистику Хотеллинга  $H(k) = \text{trace}(\hat{S}(k) \cdot \hat{\Sigma}^{-1})$ ;
- расстояние Махаланобиса  $M(k) = \text{trace}(\hat{\Sigma}^{-1} \cdot (\hat{x} - \hat{y}(k))^T \cdot (\hat{x} - \hat{y}(k)))$ ;
- дивергенцию Кульбака-Ляйблера  $J(k) = \{W(k) + [H(k) - m] + M(k)\}$ .

Требуется для отрезка  $X$  выбрать наиболее похожий на него отрезок и использовать его продолжение для прогноза вперед с отрезка  $X$  (рис. 5.6, 5.7).

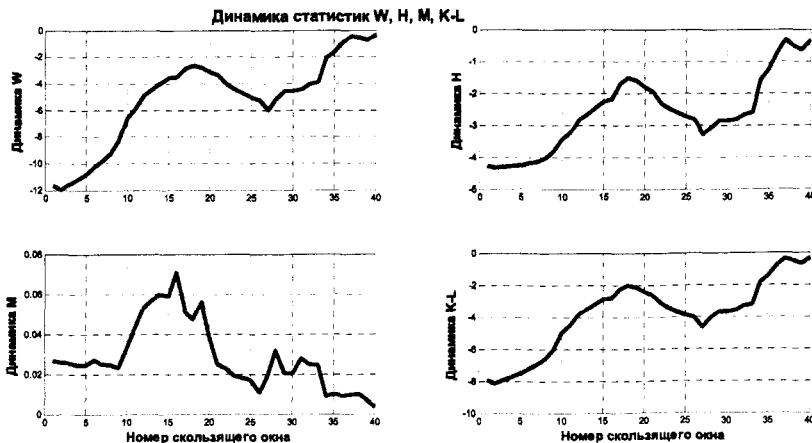


Рис. 5.6. Наиболее близким относительно выбранной меры близости можно считать окно № 2

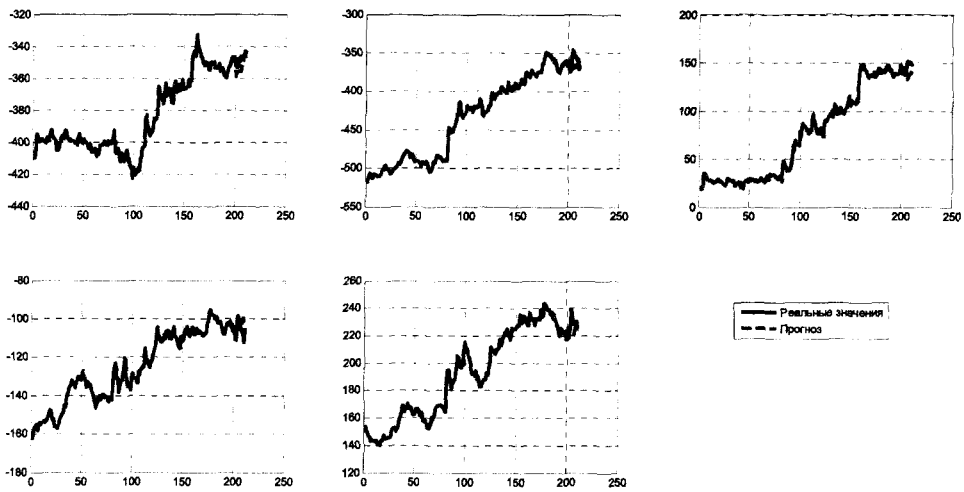


Рис. 5.7. Прогноз на 10 шагов по найденному окну-аналогу № 2

## 6. СЛИЯНИЕ ДАННЫХ

*Слияние данных* от распределенной системы датчиков – это технологии, позволяющие комплексировать информацию от различных источников для получения обобщенной картины. Такие системы в настоящее время получили широкое распространение в таких областях, робототехника, обработка визуальной информации и т. д. Например, *беспроводные сенсорные сети* содержат большое числа узлов, что приводит к проблемам при передаче данных. Если данные обрабатываются на месте и передается только результат, то число посылок уменьшается, столкновение интересов избегается и энергетические затраты используются более эффективно.

Имеется много споров по поводу терминологии построения формальных теорий слияния. Наиболее разумной представляется модель Кокара и Дасаратхи, где система слияния рассматривается с точки зрения программного обеспечения как поток данных от входа к выходу вместе с процедурой обработки: преобразование данных, обработка, передача и адаптивные комбинации этих процессов.

### 6.1. Проблемы. Оценивание в условиях неопределенности

Наиболее важная проблема в слиянии данных – разработка адекватной модели неопределенности, ассоциированной как динамикой, так и с измерениями. Важнейший элемент слияния – архитектура системы. Основные проблемы связаны с характером данных, несовершенством и разнообразием технологий датчиков, а также с окружающей средой. Главный механизм – устранение за счет избыточности всевозможных несовершенств, среди них:

- выбросы и ложные данные;
- коррелированные данные;
- конфликтующие данные;
- различные системы кодирования и системы отсчета;
- качественно-различные данные;
- неточно атрибутированные данные: требуется правильно ассоциировать измерения с объектами;

- различия в характере обработки: централизованный или децентрализованный;
- временные ограничения. Темп выдачи измерений у разных датчиков может различаться, данные могут запаздывать. Эта задача решается за счет прогнозирования или интерполяции на заданную сетку по времени. Для таких процедур имеется специальный термин – *диахронный анализ*.

Теория вероятностей всегда использовалась для описания всех типов неопределенных данных, поскольку ей просто не было альтернатив. Она обеспечивает наиболее развитый и унифицированный подход к описанию и управлению неопределенностью. Вероятностные модели позволяют комплексировать информацию, описывать различные архитектуры их слияния, управлять сенсорами и сенсорной информацией. Тем не менее, их использование связано с некоторыми ограничениями.

Сейчас наряду с теорией информации используют:

- теорию размытых множеств;
- теорию возможностей;
- теорию грубых множеств;
- теорию очевидности Демпстера – Шаффера;
- теорию случайных множеств;
- гибридные подходы – это теория размытых грубых множеств (FRST), размытая теория Демпстера – Шеффера (Fuzzy DSET) и другие, отражающие различные аспекты неопределенных данных.

Оценивание – важнейшая задача в проблеме слияния данных. Оценка – это решающее правило, использующее в качестве аргумента последовательности измерений (наблюдений) от групп сенсоров и вырабатывающее на этой основе искомое значение параметра из пространства состояний.

## 6.2. Комплексирование координатной оценки и оценки пеленга

**Задача 1.** Имеется оценка положения объекта  $X = X_0 + E$ ,  $E \in N_2(0, \Sigma)$ .

В координатной записи

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \end{bmatrix}, \quad \text{cov} \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \Sigma = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}.$$

Требуется получить исследовать оценку пеленга на объект:

$$\Pi = \text{arctg}(x/y) = \Pi_0 + e_\Pi, \quad \Pi_0 = \text{arctg}(x_0/y_0), \quad M(e_\Pi) = 0.$$

**Решение.** Не подвергая критике гауссовость и несмещенность оценки, ограничимся оцениванием ее дисперсии  $\sigma_\Pi^2$ . Найдем полный дифференциал функции  $\Pi(x, y)$ .

$$d\Pi = \frac{\partial\Pi}{\partial x} dx + \frac{\partial\Pi}{\partial y} dy = \frac{1}{x^2 + y^2} (y - x).$$

В приращениях

$$\Delta\Pi = \frac{\partial\Pi}{\partial x} \Delta x + \frac{\partial\Pi}{\partial y} \Delta y = \frac{1}{x^2 + y^2} (y\Delta x - x\Delta y).$$

Отсюда

$$M(\Delta\Pi)^2 = \frac{1}{(x^2 + y^2)^2} [y^2 M(\Delta y)^2 - 2xy M(\Delta x \Delta y) + x^2 M(\Delta x)^2],$$

так что

$$\sigma_\Pi^2 = \frac{1}{(x^2 + y^2)^2} [y^2 \sigma_y^2 - 2xy \rho \sigma_x \sigma_y + x^2 \sigma_x^2] = \frac{1}{(x^2 + y^2)^2} X^T \begin{bmatrix} \sigma_x^2 & -\rho\sigma_x\sigma_y \\ -\rho\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix} X. \quad (1)$$

**Задача 2.** В условиях задачи 1 кроме координатной информации имеется независимая оценка пеленга на объект слежения:

$$\Pi_1 = \text{arctg}(x/y) = \Pi_0 + e_\Pi, \quad \Pi_0 = \text{arctg}(x_0/y_0), \quad M(e_\Pi) = 0, \quad D(e_\Pi) = \sigma_1^2.$$

Требуется получить комплексную оценку пеленга  $\Pi$  с учетом двух доступных источников информации.

**Решение.** Согласно результатам решения задачи 1, имеются две независимых несмещенных оценки пеленга  $\Pi_0$ :  $\Pi_1$  с дисперсией  $\sigma_1^2$  и  $\Pi_2$  с дисперсией  $\sigma_2^2$ , даваемой формулой (1). Согласно принципу наименьших квадратов, требуется минимизировать квадратичную форму

$$G(\Pi) = \frac{1}{\sigma_1^2} (\Pi_1 - \Pi)^2 + \frac{1}{\sigma_2^2} (\Pi_2 - \Pi)^2.$$

Вычисляя производную  $\frac{\partial G}{\partial \Pi}$  и приравнявая ее нулю, получаем оценку по

МНК:

$$\Pi = \frac{\frac{\Pi_1}{\sigma_1^2} + \frac{\Pi_2}{\sigma_2^2}}{\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}} = \frac{\Pi_1 \sigma_2^2 + \Pi_2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2}, \quad M(\Pi) = 0, \quad D(\Pi) = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}.$$

**Задача 3.** В условиях задачи 2 требуется получить комплексную оценку вектора  $X_0$  с учетом дополнительной информации о пеленге  $\Pi$  на объект слежения.

**Решение.**  $\Pi = \arctg(x/y) = \Pi_0 + e_\Pi$ ;

$$\operatorname{tg}(\Pi) = \operatorname{tg}(\arctg(x/y) + e_\Pi) = \frac{x_0/y_0 + \operatorname{tg}(e_\Pi)}{1 + x_0/y_0 \cdot \operatorname{tg}(e_\Pi)} \cong \frac{x_0 + y_0 e_\Pi}{y_0 + x_0 e_\Pi};$$

$$\begin{aligned} \Delta \operatorname{tg}(\Pi) &= \operatorname{tg}(\Pi) - \operatorname{tg}(\Pi_0) = \frac{x_0 + y_0 e_\Pi}{y_0 + x_0 e_\Pi} - \frac{x_0}{y_0} = \frac{y_0(x_0 + y_0 e_\Pi) - x_0(y_0 + x_0 e_\Pi)}{y_0(y_0 + x_0 e_\Pi)} = \\ &= \frac{y_0^2 - x_0^2}{y_0^2} \cdot \frac{1}{1 + \frac{x_0}{y_0} e_\Pi} e_\Pi \cong \frac{y_0^2 - x_0^2}{y_0^2} e_\Pi. \end{aligned}$$

$$D(\Delta \operatorname{tg}(\Pi)) \cong \left( \frac{y_0^2 - x_0^2}{y_0^2} \right)^2 \sigma_\Pi^2. \quad (2)$$

Принятые аппроксимации разумны, если  $y_0 > x_0$ ; если это не так, расчеты выгодно производить для величины

$$\frac{\pi}{2} - \Pi = \arctg\left(\frac{y_0}{x_0}\right),$$

меняя во всех формулах местами  $x_0$  и  $y_0$ .

Обозначим  $\operatorname{tg}(\Pi) = x/y = r$ ,  $\operatorname{tg}(\Pi_0) = x_0/y_0 = r_0$ . Тогда

$$r = \frac{x}{y} = r_0 + e_r, \quad M(e_r) = 0,$$

дисперсия  $e_r$  дается формулой (2). Теперь для нахождения уточненной оценки координат  $(x_0, y_0)$  получается система линейных уравнений

$$\begin{cases} x = x_0 + e_x \\ y = y_0 + e_y \\ 0 = x_0 - r y_0 + y e_r \end{cases},$$

где вектор погрешностей  $e = [e_x, e_y, e_r]^T$  имеет среднее ноль и ковариационную матрицу

$$Q = \begin{bmatrix} \Sigma & 0 \\ 0 & y^2 \sigma_r^2 \end{bmatrix}.$$

Оценка по МНК в этом случае имеет вид

$$\hat{X} = (A^T Q^{-1} A)^{-1} A^T Z, \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -r \end{bmatrix}, \quad Z = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix}.$$

Очевидно, что в большинстве случаев дисперсия  $r$  настолько велика, что значимого улучшения имеющейся оценки координат на больших дальностях ожидать не приходится.

### 6.3. Байесовское слияние

Вероятностные методы используют описание неопределенности в виде вероятностного распределения. В их основе лежит Байесовский подход, который позволяет на основе априорного распределения и набора результатов измерений получать апостериорное распределение.

Метод Байеса основан на трактовке искомого вектора  $u$  как случайного вектора с известной априорной плотностью  $p(u)$ . Тогда апостериорная плотность  $p(u|f)$  вычисляется по формуле Байеса:

$$p(u|f) = \frac{p(u)p(f|u)}{\int p(u)p(f|u)du}, \quad (3)$$

где  $p(f|u)$  – плотность вероятностей случайного вектора  $f$  в модели, совпадающая с точностью до математического ожидания с плотностью распределения вектора погрешностей  $\varepsilon$ . Если законы неизвестны, приходится вводить их гипотетические версии. В этом случае точность полученных оценок во многом зависит от правильности подбора этих законов распределения. В качестве байесовской оценки обычно выбирают среднее, медиану или моду апостериорного распределения  $p(u|f)$ . Байесовскую оценку в аналитической форме удается получить только для достаточно редких специальных постановок.

Байесовский подход создает прекрасную базу для слияния данных из различных источников, поскольку из одной группы данных можно получить распределение искомых параметров и трактовать его как априорное для другой.

**Задача 4.** Пусть результаты измерений  $y_i, i = 1, \dots, n$  связаны с известными параметрами  $x_i$  уравнениями



$$y_i = ax_i + \xi_i, \quad \xi_i - \text{н.о.р.} \in N(0, \sigma_\xi^2), \quad (4)$$

где  $a$  – неслучайный параметр, подлежащий оцениванию.

Функция правдоподобия для модели (4) имеет вид

$$L(a) = \prod_{i=1}^n \frac{1}{\sigma_\xi \sqrt{2\pi}} \exp \left[ -\frac{(y_i - ax_i)^2}{2\sigma_\xi^2} \right] = \frac{1}{\sigma_\xi^n (2\pi)^{n/2}} \exp \left[ -\frac{1}{2\sigma_\xi^2} \sum_{i=1}^n (y_i - ax_i)^2 \right].$$

Задача ее максимизации  $L(a) \rightarrow \max$  эквивалентна задаче минимизации квадратичной формы  $G(a)$  в показателе экспоненты

$$G(a) = \sum_{i=1}^n (y_i - ax_i)^2 \rightarrow \min$$

(метод наименьших квадратов, МНК). Вычисляя ее производную и решая уравнение  $G'(a) = 0$ , получаем оценку по МНК:

$$\hat{a} = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} \in N \left( a, \frac{\sigma_\xi^2}{\sum_{i=1}^n x_i^2} \right).$$

**Задача 5.** Предположим теперь, что параметр  $a$  сам является реализацией некоторой случайной величины с известной плотностью  $\phi(x)$ , независимой от  $\{\xi_i\}$ , так что в уравнениях (4) требуется учесть дополнительную априорную информацию  $a \in \phi(x)$ . В этом случае функция правдоподобия объединенной модели имеет вид  $L(a)\phi(a)$  и задача состоит в численной максимизации этой обобщенной функции:

$$a = \arg \max [L(a)\phi(a)].$$

Особенно интересным является случай, когда  $a \in N(a_0, \sigma_a^2)$ , так что

$$\phi(a) = \frac{1}{\sigma_a \sqrt{2\pi}} \exp \left[ -\frac{(a - a_0)^2}{2\sigma_a^2} \right].$$

В такой постановке требуется решить задачу

$$L(a)\phi(a) = \frac{1}{\sigma_\xi^n (2\pi)^{n/2}} \exp \left[ -\frac{1}{2\sigma_\xi^2} \sum_{i=1}^n (y_i - ax_i)^2 \right] \frac{1}{\sigma_a \sqrt{2\pi}} \exp \left[ -\frac{(a - a_0)^2}{2\sigma_a^2} \right] \rightarrow \max,$$

что, очевидно, эквивалентно задаче

$$G(a) = \frac{1}{\sigma_\xi^2} \sum_{i=1}^n (y_i - ax_i)^2 + \frac{(a - a_0)^2}{\sigma_a^2} \rightarrow \min.$$

Решая уравнение  $G'(a) = 0$ , получаем оценку параметра  $a$  в виде

$$\hat{a} = \frac{\frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^n x_i y_i + \frac{a_0}{\sigma_a^2}}{\frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^n x_i^2 + \frac{1}{\sigma_a^2}} = \frac{\sum_{i=1}^n x_i y_i + a_0 \frac{\sigma_\varepsilon^2}{\sigma_a^2}}{\sum_{i=1}^n x_i^2 + \frac{\sigma_\varepsilon^2}{\sigma_a^2}}.$$

Если априорная информация верна и  $Ma = a_0$ , такая оценка оказывается несмещенной с дисперсией

$$D\hat{a} = \sigma_\varepsilon^2 \frac{\sum_{i=1}^n x_i^2}{\left( \sum_{i=1}^n x_i^2 + \frac{\sigma_\varepsilon^2}{\sigma_a^2} \right)^2}.$$

**Задача 6.** Теперь предположим, что в условиях задачи 4 кроме измерений, описываемых уравнением (1), имеется еще одна группа измерений  $u_j$ ,  $j = 1, \dots, m$  того же неслучайного параметра  $a$ , описываемая уравнениями

$$v_j = au_j + \varepsilon_j, \quad \varepsilon_j - \text{н.о.р.} \in N(0, \sigma_\varepsilon^2).$$

В этом случае функция правдоподобия комбинированной модели  $L(a) = L_1(a)L_2(a)$  имеет вид

$$L(a) = \frac{1}{\sigma_\varepsilon^n (2\pi)^{n/2}} \exp \left[ -\frac{1}{2\sigma_\varepsilon^2} \sum_{i=1}^n (y_i - ax_i)^2 \right] \frac{1}{\sigma_\varepsilon^m (2\pi)^{m/2}} \exp \left[ -\frac{1}{2\sigma_\varepsilon^2} \sum_{j=1}^m (v_j - au_j)^2 \right]$$

и ее максимизация эквивалентна задаче

$$G(a) = \frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^n (y_i - ax_i)^2 + \frac{1}{\sigma_\varepsilon^2} \sum_{j=1}^m (v_j - au_j)^2 \rightarrow \min.$$

Решая уравнение  $G'(a) = 0$ , получаем оценку параметра  $a$  в виде

$$\hat{a} = \frac{\frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^n x_i y_i + \frac{1}{\sigma_a^2} \sum_{j=1}^m u_j v_j}{\frac{1}{\sigma_\varepsilon^2} \sum_{i=1}^n x_i^2 + \frac{1}{\sigma_a^2} \sum_{j=1}^m u_j^2} \in N \left( a, \frac{\sigma_\varepsilon^2}{\sum_{i=1}^n x_i^2 + \frac{\sigma_\varepsilon^2}{\sigma_a^2} \sum_{j=1}^m u_j^2} \right).$$

#### 6.4. Примеры комплексирования данных

Далее рассмотрим основные примеры комплексирования данных на основе решетчатого фильтра и техники Монте-Карло.

**Задача 7.** Имеются пеленги на объект слежения от двух наблюдателей с базой  $B$  и известными СКО. Требуется построить для оцененных координат

нат объекта доверительный эллипс надежности 0.95 в декартовых координатах.

Расчет доверительного эллипса в задаче о двух пленгах

```
clear; clc; %Два пленга
B=5; %База
a=pi/3; b=3*pi/4; %пленги
s_a=a*0.03; s_b=b*0.03;
[X0,Y0]=dec(a,b,B);
%Линии пленгов
x=0:0.1:5;
y1=x*tan(a);
y2=(x-B)*tan(b);
plot(x,[y1;y2], 'LineWidth', 2); grid;
axis([0 5 0 4]);
%Частные производные
del=0.001;
[X1,Y1]=dec(a+del,b,B);
[X2,Y2]=dec(a,b+del,B);
dX_a=(X1-X0)/del; dX_b=(X2-X0)/del;
dY_a=(Y1-Y0)/del; dY_b=(Y2-Y0)/del;
%Дисперсии и ковариации
DX=(dX_a)^2*(s_a)^2+(dX_b)^2*(s_b)^2;
DY=(dY_a)^2*(s_a)^2+(dY_b)^2*(s_b)^2;
K=dX_a*dY_a*(s_a)^2+dX_b*dY_b*(s_b)^2;
%Эллипс рассеяния
SIG=[DX K; K DY]; S=inv(SIG);
c=-2*log(0.05);
Z=ell_1(S,c,X0,Y0);
hold on; plot(Z(1,:),Z(2:,:), 'LineWidth', 2);
hold on; fill(Z(1,:),Z(2:,:), [0 1 0]);
hold on; plot(X0,Y0, 's', 'LineWidth', 4);
```

function [x,y]=dec(a,b,B); %2 пленга a,b и база B - точка пересечения декартовых координатах

```
x=B*sin(b)*cos(a)/sin(b-a);
y=B*sin(b)*sin(a)/sin(b-a);
```

function X=ell\_1(A,c,x0,y0);

%Построение эллипса

```
[P,Q]=eig(A);
a=sqrt(c/Q(1,1)); b=sqrt(c/Q(2,2));
t=linspace(0,2*pi,100);
x=a*cos(t); y=b*sin(t);
```

```

X=[x;y]; X=P*X;
X(1,:)=x0+X(1,:);
X(2,:)=y0+X(2,:);

```

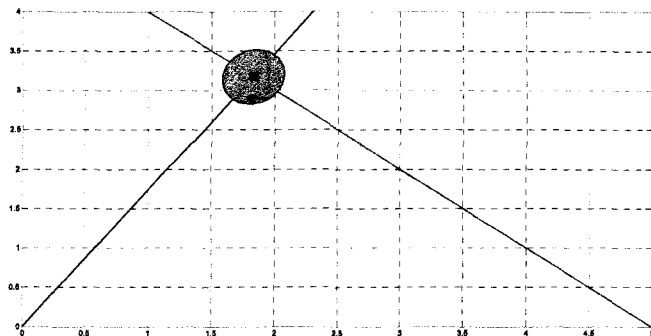


Рис. 6.1. Расчетный вид доверительного эллипса надежности 0.9 (задача 1)

**Задача 8.** В условиях задачи 7 построить доверительный эллипс, разыгрывая возможные положения объекта слежения (виртуальные сценарии) на решетке в окрестности его расчетного положения.

Расчет доверительного эллипса в задаче о двух пеленгах (решетчатый фильтр)

```

clear; clc; %2 наблюдателя - решетчатый фильтр
B=5; %база (км)
a0=pi/3; b0=3*pi/4; %пеленги (рад)
s_a=a0*0.03; s_b=b0*0.03; %СКО погрешностей (рад)
[X0,Y0]=dec(a0,b0,B);
%Линии пеленгов
x0=0:0.1:5;
y1=x0*tan(a0); y2=(x0-B)*tan(b0);
plot(x0,[y1;y2],'LineWidth',2); grid; axis([0 5 0 4]);

%Решетка в окрестности (a0,b0)
n=61; del=0.0005; %шаг решетки (рад)
SS=zeros(2,2);
for i=1:n;
    a=a0+del*(i-(n+1)/2);
    for j=1:n;
        b=b0+del*(j-(n+1)/2);
        [x,y]=dec(a,b,B);
        r=((a-a0)/s_a)^2+((b-b0)/s_b)^2;
        Z=[x-X0 y-Y0];
    end
end

```

```

SS=SS+Z'*Z*exp(-r/4);
R(i,j)=r;
end;
end;
SS=SS/sum(sum(R));
%Эллипс рассеяния
c=-2*log(0.05);
Z=ell_1(inv(SS),c,X0,Y0);
hold on; plot(Z(1,:),Z(2:,:), 'LineWidth',2);
hold on; fill(Z(1,:),Z(2:,:), [0 1 0]);
hold on; plot(X0,Y0,'s', 'LineWidth',4);

```

```

function [x,y]=dec(a,b,B); %2 пеленга a,b и база B - точка пересечения
%в декартовых координатах
x=B*sin(b)*cos(a)/sin(b-a);
y=B*sin(b)*sin(a)/sin(b-a);

```

**Задача 9.** В условиях задачи 7 построить доверительный эллипс, разрывая возможные положения объекта слежения в окрестности его расчетного положения (виртуальные сценарии) с помощью датчика случайных чисел (техника Монте-Карло).

Расчет доверительного эллипса в задаче о двух пеленгах (техника Монте-Карло)

```

clear; clc; %2 наблюдателя - Монте-Карловский фильтр
BB=5; %база (км)
a0=pi/3; b0=3*pi/4; %пеленги (рад)
s_a=a0*0.03; s_b=b0*0.03; %СКО погрешностей (рад)
[X0,Y0]=dec(a0,b0,BB);
%Линии пеленгов
x0=0:0.1:5;
y1=x0*tan(a0); y2=(x0-BB)*tan(b0);
plot(x0,[y1;y2], 'LineWidth',2); grid; axis([0 5 0 4]);

%Случайные точки в окрестности (X0,Y0)
N=5000; %число имитаций
SS=zeros(2,2); h=0.001;
for k=1:N;
a=a0+(h*rand-h/2); b=b0+(h*rand-h/2);
[x,y]=dec(a,b,BB);
r=((a-a0)/s_a)^2+((b-b0)/s_b)^2;

```

```

Z=[x-X0 y-Y0];
SS=SS+Z'*Z*exp(-r/2);
R(k)=r;
%X(k)=x; Y(k)=y;
%A(k)=a; B(k)=b;
end;
SS=SS/sum(R); %масштабирующий множитель
%Эллипс рассеяния
c=-2*log(0.05);
Z=ell_1(inv(SS),c,X0,Y0);
hold on; plot(Z(1,:),Z(2:),'LineWidth',2);
hold on; fill(Z(1,:),Z(2:),[0 1 0]);
hold on; plot(X0,Y0,'s','LineWidth',4);
%figure;
%plot(A,B,'r. '); grid; hold on; plot(a0,b0,'s','LineWidth',3);
%plot(X,Y,'r. '); grid; hold on; plot(X0,Y0,'s','LineWidth',3);

```

Любопытно, что сценарии нужно разыгрывать в пространстве пеленгов, и в нем строить эмпирическое распределение. Если сценарии разыгрывать в системе  $(xOy)$ , то в выражение для эллипса войдет *масштабирующий множитель*, связанный с якобианом при нелинейном преобразовании.

## 7. МАШИНЫ ОПОРНЫХ ВЕКТОРОВ

### 7.1. Постановка задачи

В парадигму анализа данных естественно вписывается ряд новых методов, использующих дуализм «человек – машина», в том числе – метод опорных векторов или метод обобщенного портрета, разработанный в 1970-е гг. под руководством В. Н. Вапника. Метод основан на построении оптимальной разделяющей гиперплоскости, требование оптимальности означает, что обучающие объекты должны быть максимально удалены от разделяющей поверхности. В 1990-е гг. метод получил широкую мировую известность и после серии обобщений стал называться *машиной опорных векторов (support vector mashine, SVM)*. Метод изначально относится к бинарным классификаторам, хотя существуют способы заставить его работать и для задач мультиклассификации.

### 7.2. Идея метода опорных векторов

Идею метода удобно проиллюстрировать на следующем простом примере: даны точки на плоскости, разбитые на два класса (рис. 7.1). Проведем линию, разделяющую эти два класса. Далее, все новые точки (не из обучающей выборки) автоматически классифицируются следующим образом:

- точка выше прямой попадает в класс **A**;
- точка ниже прямой – в класс **B**.

Такую прямую назовем *разделяющей прямой*. В пространствах высоких размерностей место прямых занимают гиперплоскости – пространства, размерность которых на единицу меньше, чем размерность исходного пространства.

С точки зрения точности классификации лучше всего выбрать прямую, расстояние от которой до каждого класса максимально. Другими словами, выберем ту прямую, которая разделяет классы наилучшим образом (рис. 7.2). Такая прямая, а в общем случае – гиперплоскость, называется оптимальной разделяющей гиперплоскостью.

Векторы, лежащие ближе всех к разделяющей гиперплоскости, называются *опорными векторами (support vectors)*.

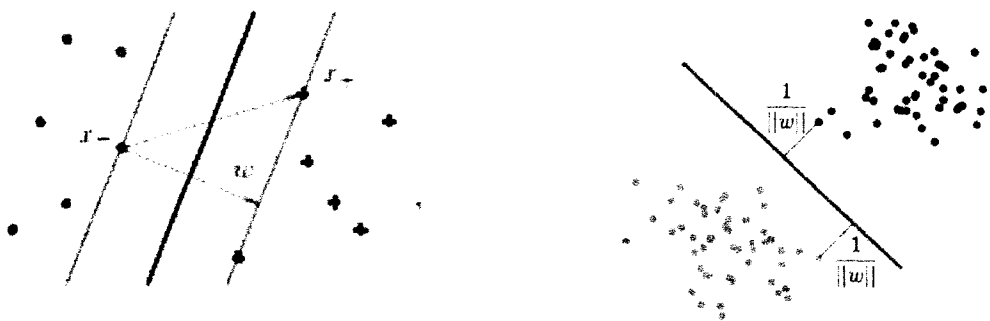


Рис. 7.1. Визуализация метода опорных векторов

Метод опорных векторов сводит обучение классификатора к задаче квадратичной оптимизации. Метод также известен как *метод классификатора с максимальным зазором*.

### 7.3. Разделение полосой на плоскости

Пусть на плоскости имеются два класса объектов, представленных двумерными обучающими выборками размерности  $m$  и  $n$  соответственно. Они образуют объединенное облако точек:

$$A = \{(x_i, y_i), i = 1, \dots, m + n\},$$

в котором каждая точка снабжена идентификатором  $h_i$ : для объектов первого класса  $h_i = 1$ , для второго  $-h_i = -1$ . Предположим, что эти классы являются *линейно-отделимыми*, т. е. существует такая прямая, что классы лежат от нее по разные стороны (рис. 7.2). *Ставится задача*: подобрать параметры прямой так, чтобы точки разных классов лежали от неё по разные стороны и чтобы сумма их расстояний до этой прямой была максимальной (рис. 7.2).

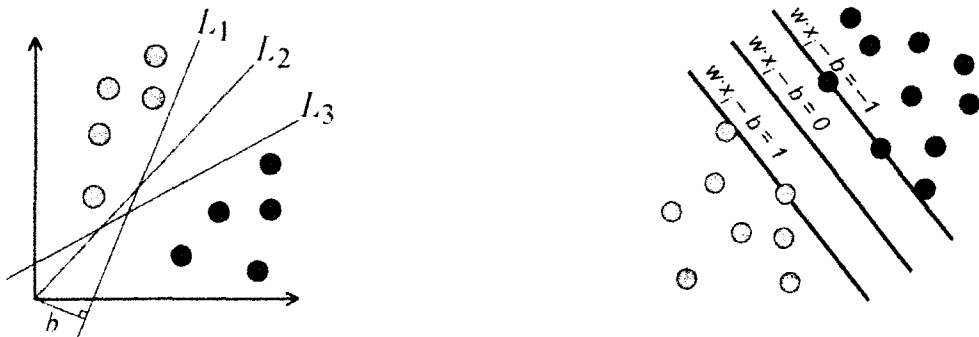


Рис. 7.2. Оптимальная разделяющая гиперплоскость



Будем искать такую прямую в виде  $l: ax + y - c = 0$ . Это уравнение описывает все прямые, кроме горизонтальных, для которых  $y = const$ . Расстояние произвольной точки  $(x_i, y_i)$  до прямой

$$\rho((x_i, y_i), l) = \pm \frac{ax_i + y_i - c}{\sqrt{1^2 + a^2}}, \quad (1)$$

при этом для точек одного класса (по одну сторону от прямой  $l$ ) в правой части (1) будет стоять знак «+», а для другого — знак «-». Предположим, что для точек первого класса это будет «+» и для них же идентификатор  $h_i = 1$ . Тогда для обоих классов выражения

$$h_i(ax_i + y_i - c)$$

будут принимать только положительные значения.

Пусть имеются две параллельные прямые с общим коэффициентом  $a$ , разделяющие два класса. Расстояние между ними («зазор», «margin»)

$$\rho(l_1, l_2) = \rho(0, l_1) - \rho(0, l_2) = \frac{|c_1 - c_2|}{\sqrt{1^2 + a^2}}.$$

Поставим следующую экстремальную задачу: найти коэффициент  $a$  из условия  $1 + a^2 \rightarrow \min$  при ограничениях  $h_i(ax_i + y_i - c) \geq 0, i = 1, \dots, m+n$ . Это значит, что для двух параллельных прямых, которые являются разделяющими для данных классов, ищется их общий угловой коэффициент  $a$ , обеспечивающий максимальное расстояние, «зазор» между ними. Коэффициенты  $c_1, c_2$  определяются в самом конце при уже найденном  $a$ . Выпишем для данной задачи с ограничениями функцию Лагранжа:

$$L(a, c, \lambda) = \frac{1}{2}a^2 - \sum_{i=1}^{m+n} \lambda_i [h_i(ax_i + y_i - c) - 1] \rightarrow \min.$$

Имеем:

$$\frac{\partial L}{\partial a} = a - \sum_{i=1}^{m+n} \lambda_i h_i x_i = 0 \Rightarrow a = \sum_{i=1}^{m+n} \lambda_i h_i x_i;$$

$$\frac{\partial L}{\partial c} = \sum_{i=1}^{m+n} \lambda_i h_i = 0 \Rightarrow \sum_{i=1}^{m+n} \lambda_i h_i = 0.$$

Подставим найденные соотношения в функцию Лагранжа:

$$L = \frac{1}{2}a^2 - a \sum_{i=1}^{m+n} \lambda_i h_i x_i + c \sum_{i=1}^{m+n} \lambda_i h_i + \sum_{i=1}^{m+n} \lambda_i = -\frac{1}{2}a^2 + \sum_{i=1}^{m+n} \lambda_i = -\frac{1}{2} \sum_{i,j=1}^{m+n} \lambda_i h_i x_i \lambda_j h_j x_j + \sum_{i=1}^{m+n} \lambda_i \rightarrow \min.$$

Так находится угловой коэффициент оптимальной разделяющей прямой. Те крайние точки  $(x_i, y_i)$ , для которых прямые  $y_i = ax_i + c$  еще остаются разделяющими, называются *опорными*.

Разделяющую прямую обычно проводят через среднее найденных опорных точек. Можно также найти ближайшие друг к другу точки двух облаков и провести прямую через середину между ними.

На плоскости задача резко упрощается, поскольку её можно решать не для всех точек  $(x_i, y_i)$ , а только для точек из выпуклых линейных оболочек рассматриваемых множеств. При решении задачи на плоскости можно было бы обойтись более простыми методами, однако в пространствах более высокой размерности метод опорных векторов оказывается чрезвычайно полезным и имеет массу важнейших приложений.

**Задача 1.** Сформировать две выборки объёмом  $n = 100$  из двумерного нормального закона с центрами  $(1, 3.5)$  и  $(-1, -3.5)$ ; построить для них выпуклые линейные оболочки (процедура *convhull*).

**Задача 2.** Построить для этих облаков точек разделяющую прямую и границы разделяющей полосы (рис. 7.4), найти величину «зазора» (*margin*). При расчетах принимать во внимание только точки из выпуклых линейных оболочек двух данных облаков.

**Задача 3.** Представить все результаты в графической форме (рис. 7.3).

Разделение полосой на плоскости

```
clear; clc;
m=2; n1=100; n2=100;
%формируем 2 матрицы pxm и собираем из них одну размерности <(2n)xm>:
X1=randn(n1,m); X2=randn(n2,m);
a0=[1 3.5];
X1(:,1)=X1(:,1)-a0(1); X1(:,2)=X1(:,2)-a0(2);
X2(:,1)=X2(:,1)+a0(1); X2(:,2)=X2(:,2)+a0(2);
save dat_new X1 X2;
%load dat_new X1 X2;
[n1,m]=size(X1); [n2,m]=size(X2);
plot(X1(:,1),X1(:,2),'b*','LineWidth',3); grid;
```

```

hold on; plot(X2(:,1),X2(:,2),'r*','LineWidth',3);
title('Разделение полосой','FontSize',14);
%====Составляем матрицу попарных расстояний====
for i=1:n1;
    for j=1:n2;
        R(i,j)=norm(X1(i,:)-X2(j,:));
    end;
end;
[R1,I]=min(R); [R2,J]=min(R1);
X01=X1(I(J),:); X02=X2(J,:); X0=(X01+X02)/2;

K1=convhull(X1(:,1),X1(:,2));
hold on; plot(X1(K1,1),X1(K1,2),'LineWidth',3);
K2=convhull(X2(:,1),X2(:,2));
hold on; plot(X2(K2,1),X2(K2,2),'r','LineWidth',3);
hold on; plot(X01(1,1),X01(1,2),'gs','LineWidth',4);
hold on; plot(X02(1,1),X02(1,2),'gs','LineWidth',4);
hold on; plot(X0(1,1),X0(1,2),'gs','LineWidth',4);

XX1=X1(K1,:); XX2=X2(K2,:);
k1=length(XX1); k2=length(XX2);
x0=X0(1); y0=X0(2);
for j=1:200;
    a=-1+0.01*j; A(j,1)=a;
    marge1(j,1)=(min(XX1(:,2)-y0-a*(XX1(:,1)-x0)))/sqrt(1+a^2);
    marge2(j,1)=(min(XX2(:,2)-y0-a*(XX2(:,1)-x0)))/sqrt(1+a^2);
    marge(j,1)=marge2(j)-marge1(j);
end;
[M,I]=max(marge); a0=A(I);
x=-4:0.1:4; y=y0+a0*(x-x0);
del=marge2(I);
hold on; plot(x,y,'g--','LineWidth',3);
hold on; plot(x,y+del,'g--','LineWidth',3);
hold on; plot(x,y-del,'g--','LineWidth',3);

figure;
plot(A,marge,'LineWidth',3); grid;
xlabel('Угловой коэффициент','FontSize',14);
ylabel('Зазор (marge)','FontSize',14);

```

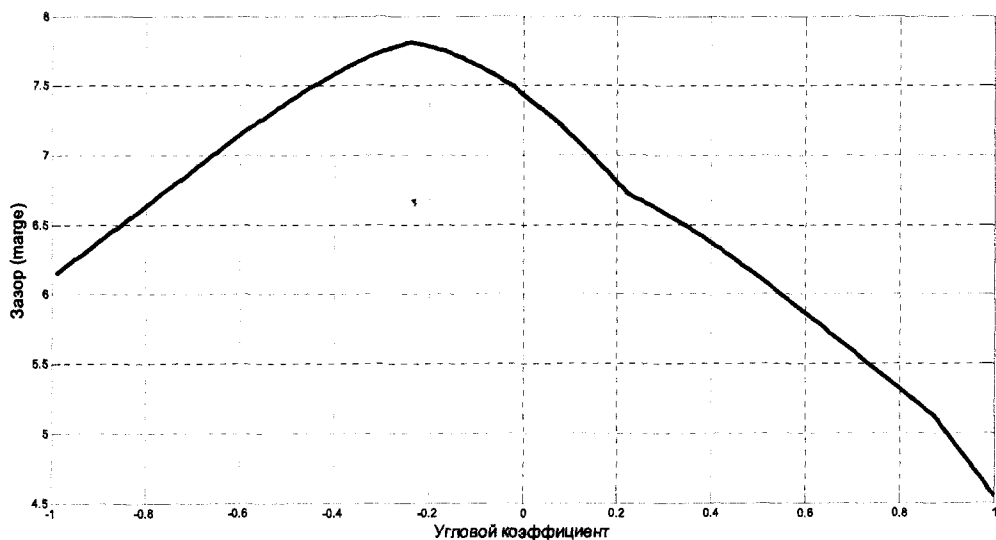


Рис. 7.3. Зависимость зазора (ширины полосы) от углового коэффициента

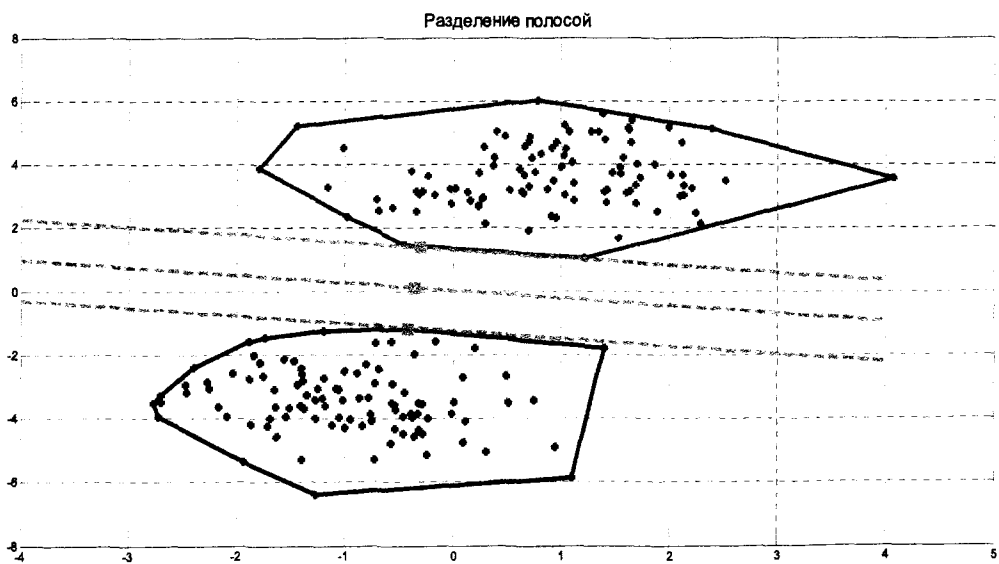


Рис. 7.4. Разделение полосой

#### 7.4. Случай отсутствия линейной отделимости

На практике случаи, когда данные можно разделить гиперплоскостью, или, как еще говорят, *линейно*, довольно редки. Пример линейной неразделимости можно видеть на рисунке 7.5.

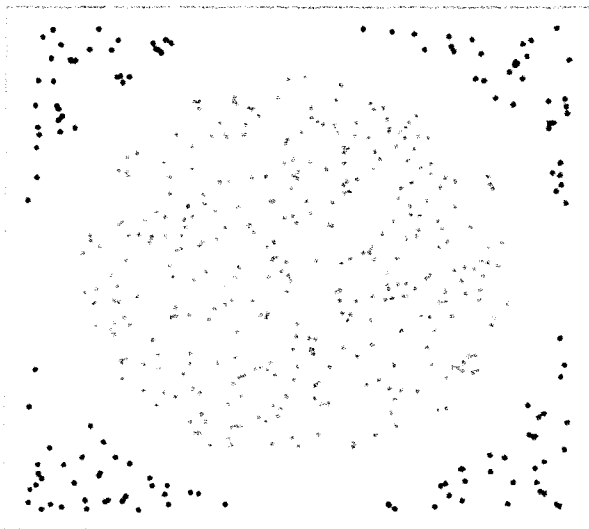


Рис. 7.5. Случай отсутствия линейной отделимости

Легко видеть, что такие облака точек разделяются при переходе к полярным координатам с началом отсчета в центре «зелёного» облака.

Один подход в случае отсутствия линейной отделимости классов состоит во введении штрафов за попадание точек из обучающих выборок внутрь разделяющей полосы. При этом вводятся ограничения вида

$$y_i - kx_i \leq \varepsilon; \quad kx_i - y_i \leq \varepsilon; \quad v_j - ku_j \leq \varepsilon; \quad ku_j - v_j \leq \varepsilon, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Замечательное свойство такой постановки состоит в том, что она по-прежнему сводится к задаче квадратичного программирования, но теперь требуется многократное повторение решения при разных значениях константы, отвечающей за штрафы.

### 7.5. Развитие метода

Более радикальное предложение состоит в том, чтобы расширить пространство признаков или произвести его нелинейное преобразование так, чтобы задача стала линейно-отделимой.

Основная идея метода – перевод исходных векторов в пространство более высокой размерности и поиск разделяющей гиперплоскости с максимальным зазором в этом пространстве. Все элементы обучающей выборки вкладываются в пространство  $X$  более высокой размерности с помощью специального отоб-

ражения  $\phi: R^n \rightarrow X$ . При этом отображение  $\phi$  выбирается так, чтобы в новом пространстве  $X$  выборка была *линейно* разделима.

Выражение  $k(x^1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$  называется **ядром классификатора**. С математической точки зрения ядром может служить любая положительно определенная симметричная функция двух переменных. Положительная определенность необходима для того, чтобы соответствующая функция Лагранжа в задаче оптимизации была ограничена снизу, т. е. задача оптимизации была бы корректно определена.

Точность классификатора зависит в большой степени от выбора ядра. Чаще всего на практике встречаются следующие ядра:

- полиномиальное:  $k(x_1, x_2) = (\langle x_1, x_2 \rangle + \text{const})^d$ ;
- радиальная базисная функция:  $k(x, x') = e^{-\gamma \|x - x'\|^2}$ ,  $\gamma > 0$ ;
- гауссова радиальная базисная функция:  $k(x, x') = e^{-\frac{\|x - x'\|^2}{2a^2}}$ ;
- сигмоид:  $k(x, x') = \tanh(k \langle x, x' \rangle + c)$ ,  $k > 0$ ,  $c < 0$ .

#### **Преимущества SVM:**

- Обучение SVM сводится к задаче квадратичного программирования.
- Положение оптимальной разделяющей гиперплоскости зависит лишь от небольшой доли обучающих объектов – опорных векторов.
- При введении функции ядра метод обобщается на случай нелинейных разделяющих поверхностей (переход к спрямляющему пространству).
- Можно не строить спрямляющее пространство в общем виде, а просто подобрать подходящее ядро.
- Максимизация зазора между классами улучшает обобщающую способность.

#### **Недостатки SVM:**

- Неустойчивость к шуму в исходных данных: объекты-выбросы оказываются опорными.
- Нет общих методов подбора ядер под конкретную задачу.

- Выбор параметра регуляризации, который управляет компромиссом между шириной разделяющей полосы и суммарной ошибкой, требует многократного решения задачи.

В нейроматематике SVM рассматривают как один из классов универсальных сетей прямого распространения, который может трансформироваться в другие классы сетей при соответствующем выборе функции ядра.

Такая классификация имеет довольно широкое применение: от распознавания образов или создания спам-фильтров до вычисления распределения горячих алюминиевых частиц в ракетных выхлопах.

Среди других классификаторов стоит отметить также *метод релевантных векторов (Relevance Vector Machine, RVM)*. В отличие от SVM данный метод дает вероятности, с которыми объект принадлежит данному классу. То есть если SVM говорит « $x$  принадлежит классу А», то RVM скажет, что « $x$  принадлежит классу А с вероятностью  $p$  и классу В с вероятностью  $1 - p$ ».

## 7.6. Регрессионный анализ на базе метода опорных векторов

Метод опорных векторов (*support vector machine – SVM*) был разработан в 1995 г. в американской корпорации AT&T Bell Laboratories под руководством В. Н. Вапника. Изначально он позиционировался как алгоритм решения классификационной задачи, однако позднее его стали использовать и для выполнения регрессионного анализа. При этом аббревиатуру SVM заменяют на SVR (*support vector regression*). При этом оценивание параметров регрессионной модели тоже сводится к решению задачи квадратичного программирования, имеющей единственное решение.

При использовании SVR нелинейная регрессия в исходном пространстве  $F$  трансформируется в задачу построения линейной регрессии в расширенном пространстве  $H$  более высокой размерности. В отличие от формулировки SVM, где гиперплоскость должна отделять одну группу векторов от другой, в SVR гиперплоскость строится таким образом, чтобы как можно больше точек попало на нее в качестве опорных или хотя бы оказалось в некоторой адаптивно настраиваемой полосе.

Задачу поиска оптимальной гиперплоскости *SVR* – анализ сводит к задаче минимизации квадратичного функционала при ограничениях, включающих положительную константу *C*, регулирующую штраф за ошибки.

Векторы выборки входят в задачу регрессии только через скалярные произведения, что позволяет перейти к ядерной версии *SVM*. Наиболее распространенными ядерными функциями являются следующие: линейная, гауссова радиально-базисная, полиномиальная, сигмоидальная, экспоненциальная радиально-базисная.

В тех случаях, когда необходимо использовать регрессионную связь между управляемыми обобщенными координатами, значительные перспективы имеют применение *SVR*-анализа, обладающего следующими преимуществами:

- параметры регрессионной модели оцениваются при решении задачи квадратичного программирования, имеющей единственное решение;
- можно задавать и адаптивно подстраивать доверительную зону вокруг основной поверхности регрессии, в которой ошибка соотношения между управляемыми координатами считается допустимой.

Эталонная целевая функция, которая формируется в результате *SVR*-анализа, может быть легко изменена путем переобучения алгоритма *SVR* и не требует подбора специальных типов зависимостей, поскольку необходимый линеаризующий функционал уже заложен в ядерной функции (в случае нелинейной регрессии наилучшие результаты дает ядерная функция Gaussian RBF).

Методы машин опорных векторов (*SVM*) и *SVR* получили очень широкое развитие и применение в последние несколько лет в самых различных областях статистического анализа данных. Включая в себя мощные теоретические и практически развитые принципы теории нейронных сетей, дающие возможность эффективного обучения, машины опорных векторов являются очень хорошим инструментом извлечения знаний из статистических данных.



## 8. НЕЙРОМАТЕМАТИКА

### 8.1. Пример: перцептрон Розенблатта

Для каждого из 274 призывников Вооруженных сил имеется 7-мерный вектор  $x = (x_1, \dots, x_7)$  признаков, включающих обобщенные начальные данные о медицинском, физическом и психологическом состоянии, интеллекте, отношении к службе и т. п. Восьмой признак – это оценка, выставленная психологами в конце срока службы: «2» – «3» – «4». Требуется предсказать результат по первым 7 признакам – результатам входного контроля.

Проекция данных на плоскость 2 главных факторов представлена на рисунке 8.1, результаты линейного дискриминантного анализа – на рисунке 8.2 (файлы *percep.m* и *percep\_1.m*).

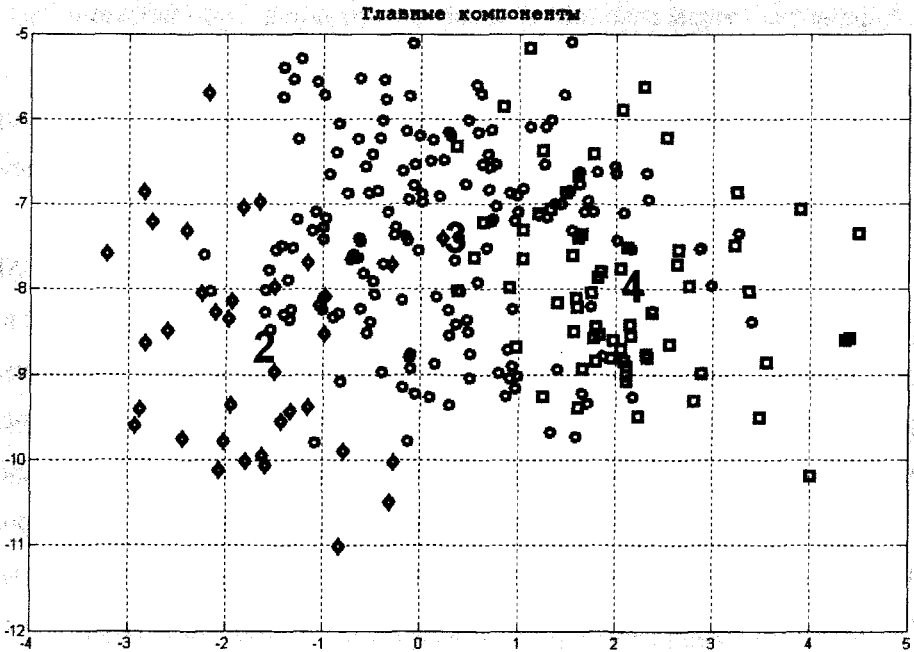


Рис. 8.1. Представление данных на плоскости 2 главных факторов

Процесс классификации можно усовершенствовать:

1. Предполагая, что данные 3 класса представляют собой реализации 3 случайных гауссовых векторов, оценим их средние и ковариационные матрицы, после чего построим дискриминационные информанты для каждо-

го класса. На плоскости рисунка 8.2 разделяющие прямые заменятся отрезками кривых 2-го порядка (гипербол, эллипсов, парабол).

2. Можно провести робастный анализ: оценить средние и ковариационные матрицы, найти расстояния Махаланобиса каждого измерения от центра его класса, затем отбросить, например, 3–5% самых удаленных измерений и оценить параметры классов заново. Эту процедуру повторяют несколько раз.
3. Можно попытаться построить оценки многомерных плотностей классов (не предполагая их гауссовыми) и строить дискриминационные информанты на основе этих оценок. Этот путь хорош при наличии тысяч измерений. При сравнительно небольшом их числе результат будет очень сильно зависеть от использованного механизма сглаживания.

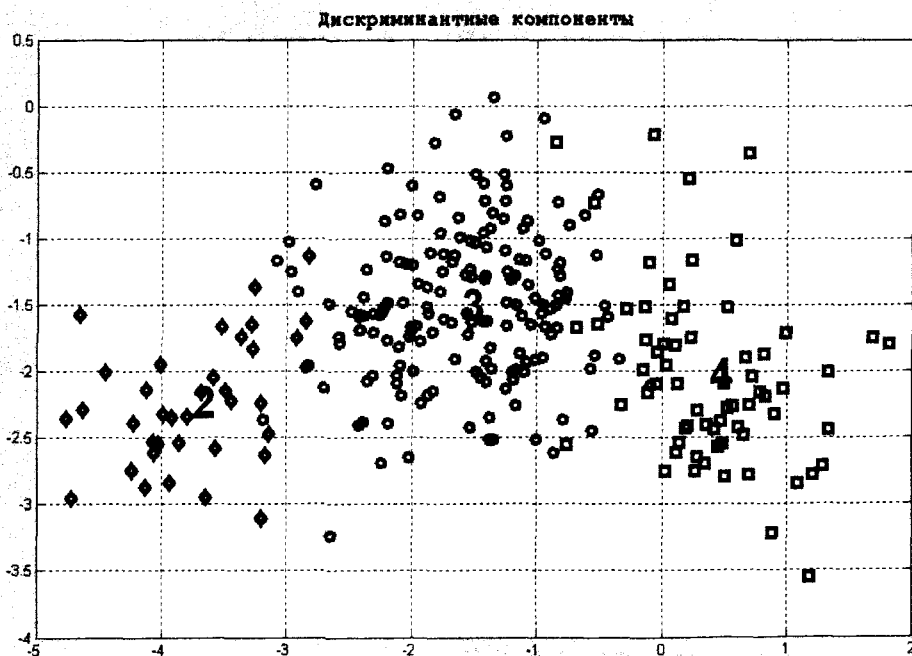


Рис. 8.2. Представление данных на плоскости 2 главных дискриминантных факторов

4. Можно, наконец, перейти в пространство более высокой размерности, например, рассмотреть наряду с  $x_1, \dots, x_7$  их попарные произведения. Ло-

гика здесь состоит в том, что нелинейная процедура разделения в исходном пространстве превращается в линейную в пространстве большей размерности.

Рассмотрим самую простую процедуру подгонки под известный результат, хорошо известную всем студентам. Пусть  $X$  – матрица данных  $\langle 274 \times 7 \rangle$ ,  $Z$  – вектор  $\langle 274 \times 1 \rangle$  – 8-й признак, оценки. Подберем линейную комбинацию  $\theta = (\vartheta_1, \dots, \vartheta_7)^T$  как решение по МНК системы линейных уравнений

$$z_i = \sum_{j=1}^7 x_{ij} \vartheta_j + \xi_i.$$

Обычно бывает выгодно добавить в  $X$  нулевой столбец из единиц, а в  $\theta$  – нулевой компоненту – *смещение*, тогда модель имеет вид

$$z_i = \vartheta_0 + \sum_{j=1}^7 x_{ij} \vartheta_j + \xi_i.$$

### Программа подгонки данных по МНК – файл *percep\_2.m*

```
clear; clc; %Перцептрон
load dat_linn.prn; %Данные Линника по тестированию призывников
X=dat_linn(:,2:8); %результаты тестирования (274x7)
Y=X*inv(diag(std(X))); %нормализация
%Y=X;
x=dat_linn(:,1); %оценки
I2=find(x==2); Y2=Y(I2,:); %двойки
I3=find(x==3); Y3=Y(I3,:); %тройки
I4=find(x==4); Y4=Y(I4,:); %четверки
Y0=[Y ones(size(x))];
teta=inv(Y0'*Y0)*Y0'*x; %веса

arg2=2*ones(size(I2)); y2=teta(8)+Y2*teta(1:7);
arg3=3*ones(size(I3)); y3=teta(8)+Y3*teta(1:7);
arg4=4*ones(size(I4)); y4=teta(8)+Y4*teta(1:7);
plot(arg2,y2,'k*','LineWidth',3); grid;
hold on; plot(arg3,y3,'bo','LineWidth',3);
hold on; plot(arg4,y4,'rs','LineWidth',3);
title('Перцептрон','FontName','Courier New Cyr',...
'FontSize',14,'FontWeight','Bold');
axis([1 5 1 5]);
```

Результат вычислений представлен на рисунке 8.3 и его нельзя признать вполне удовлетворительным – слишком велики зоны перекрытия между соседними классами.

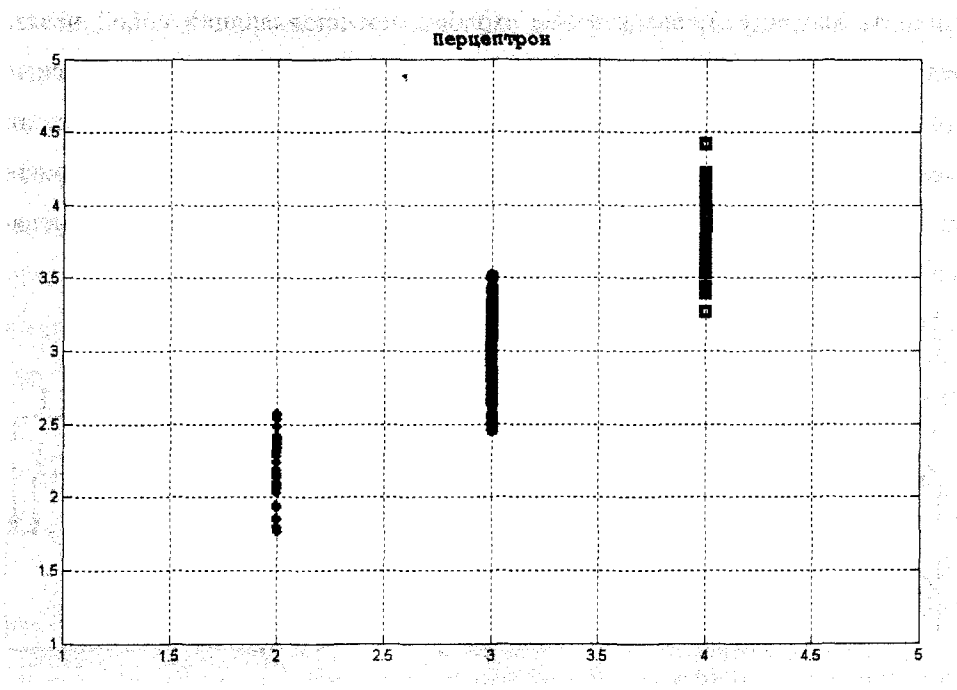


Рис. 8.3. Результат подгонки данных по МНК

После того как *веса*  $\vartheta_1, \dots, \vartheta_7$  и *смещение*  $\vartheta_0$  определены, для любого вектора  $x \langle 7 \times 1 \rangle$  можно получить *выходной сигнал сумматора*

$$s = \vartheta_0 + \sum_{j=1}^7 x_j \vartheta_j.$$

Этот сигнал подвергается нелинейному преобразованию:  $y = f(s)$ . Функция  $f$  называется *функцией активации* или *передаточной функцией*, сигналы нелинейного выхода

$$y_i = f\left(\vartheta_0 + \sum_{j=1}^7 x_{ij} \vartheta_j\right)$$

выходными сигналами.

Данная конструкция называется *элементарным перцептроном* или *нейроном* (рис. 8.4) и рассматривается как упрощенная модель биологического

нейрона (рис. 8.5). Наиболее распространенные функции активации приведены в таблице 1.

Следующий шаг (такие шаги называются *эпохами*) состоит в том, чтобы расширить матрицу  $X$ , введя в нее столбец, представляющий собой невязки между целевыми значениями «2», «3», «4» и нелинейными выходными сигналами первого нейрона  $y_i$ , и еще один столбец из единиц. Возникает нелинейная модель с  $7 + 1 + 1 + 1 = 10$  – мерным вектором  $\theta$ , который приходится оценивать рекуррентно: любое изменение в первых 8 компонентах  $\theta$  изменяет 9-й столбец матрицы плана.

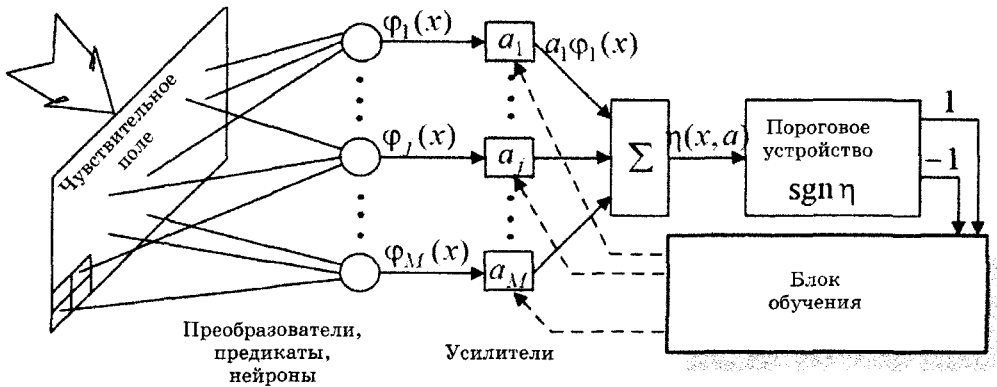


Рис. 8.4. Одношаговый перцептрон (искусственный нейрон)

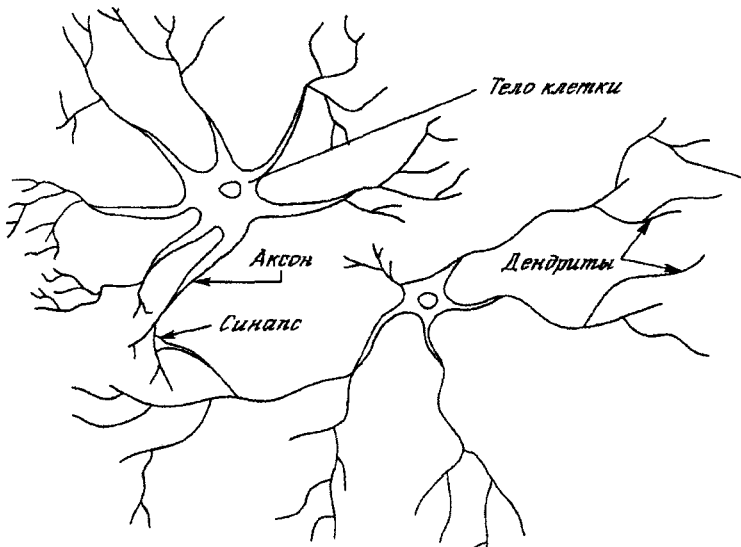
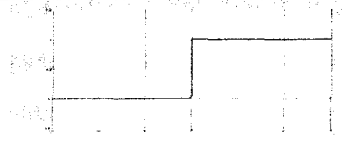
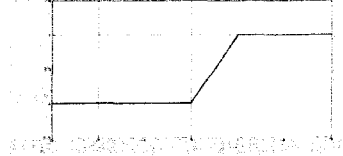
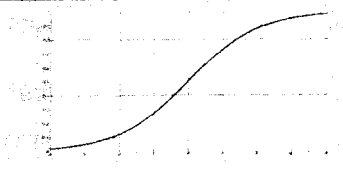
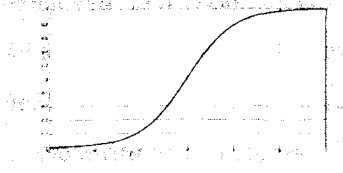


Рис. 8.5. Биологический нейрон

После этого невязки нелинейного выхода второго нейрона присоединяются к матрице плана и начинается третья эпоха – оценивание 12-мерного вектора  $\theta$  и т. д. Первая итерация второй эпохи построена в файле *percep\_3.m*.

Таким образом, алгоритм в каждую эпоху действует итеративно, в каждой эпохе на вход сети подаются все обучающие наблюдения, выходные значения сети сравниваются с целевыми значениями и вычисляется ошибка, которая преобразуется функцией  $f$ . Доказано, что такой алгоритм является сжимающим, т. е. в каждой эпохе после нескольких итераций вектор  $\theta$  стабилизируется. Это, в частности, значит, что на рисунке 8.3 вертикальные столбики постепенно сжимаются около заданных значений «2», «3», «4» и зоны перекрытия между ними уменьшаются. Примеры активационных функций.

Таблица 1

Название	Формула	Область значений	График
Пороговая	$f(s) = \begin{cases} 0, & s < \theta \\ 1, & s \geq \theta \end{cases}$	(0,1)	
Полулинейная с насыщением	$f(s) = \begin{cases} 0, & s \leq 0 \\ s, & 0 < s < 1 \\ 1, & s \geq 1 \end{cases}$	(0, 1)	
Сигмоид (логистическая)	$f(s) = \frac{1}{1 + \exp(-s)}$	(0, 1)	
Сигмоид (гиперболический тангенс)	$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$	(-1, 1)	

Очевидно, когда число оцениваемых в очередную эпоху параметров приблизится к объему выборки ( $N = 274$ ), система будет полностью распознавать

все 3 класса, т. е. полностью настроится на заданные обучающие выборки. Если имеются независимые контрольные выборки, то сначала результаты распознавания для них будут улучшаться, а затем начнут ухудшаться. Этот эффект называется *переобучением*. Таким образом, в процессе обучения *важно вовремя остановиться*.

Геометрически процесс обучения в рассматриваемом примере можно представить как построение и последовательное усложнение некоей 7-мерной гиперповерхности. Алгоритм сходится к одному из ее локальных минимумов. При этом в первой части эпох поверхность описывает и уточняет общую структуру данных, а затем заикливается на мелочах, связанных с особенностями обучающей выборки.

Исторически оптимизация основывалась на алгоритме Левенберга-Марквардта – нелинейном рекуррентном МНК, в котором на каждом шаге имеется возможность уточнять оценку несколькими способами:

- введением нового измерения;
- удлинением вектора параметров;
- рекурсией для более точного отображения нелинейных особенностей модели.

С развитием данного подхода в нем стали использоваться самые разные современные алгоритмы оптимизации и различные сложные схемы соединения нейронов в многослойные сети.

## 8.2. Краткий исторический обзор

Термин «нейронные сети» сформировался в конце 40-х гг. XX века в среде исследователей, изучавших принципы организации и функционирования биологических нейронных сетей. Основные результаты, полученные в этой области, связаны с работой МакКаллока и Питтса.

Второй пик интереса возник в 1960-х гг. благодаря теореме сходимости перцептрона Розенблатта. Работа Минского и Пейперта, указавшая ограниченные возможности простейшего перцептрона, несколько охладила пыл энтузиастов.

стов, но обеспечила время для необходимой консолидации и развития лежащей в основе теории.

С начала 1980-х гг. искусственные нейронные сети (ИНС) вновь привлекли интерес исследователей, что связано с энергетическим подходом Хопфилда и алгоритмом обратного распространения для обучения многослойного перцептрона (многослойные сети прямого распространения), впервые предложенного Вербосом и независимо разработанного рядом других авторов. Открытие методов обучения многослойных сетей повлияло на возрождение интереса и исследовательских усилий в большей степени, чем какой-либо иной фактор.

В последние годы наблюдается повышенный интерес к нейронным сетям, которые нашли применение в самых различных областях человеческой деятельности. Интеллектуальные системы на основе ИНС позволяют с успехом решать проблемы распознавания образов, выполнения прогнозов, оптимизации, ассоциативной памяти и управления. Безусловно, известны и иные, более традиционные подходы к решению этих проблем, однако они не обладают необходимой гибкостью за пределами ограниченных условий. ИНС дают многообещающие альтернативные решения, и многие приложения выигрывают от их использования.

### 8.3. Архитектура нейронных сетей

Нейронной сетью будем называть структуру, состоящую из связанных между собой нейронов. Как правило, передаточные (активационные) функции всех нейронов в сети фиксированы, а веса меняются в процессе настройки.

Нейронные сети могут иметь различные архитектуры. По архитектуре связей ИНС могут быть сгруппированы в два класса (рис. 8.6): сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

На рисунке 8.6 представлены типовые сети каждого класса. Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети. Рекуррентные сети являются динамическими, так



как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети. В этом случае вход сети надо рассматривать как последовательность векторов, подаваемых на сеть в определенные моменты времени.

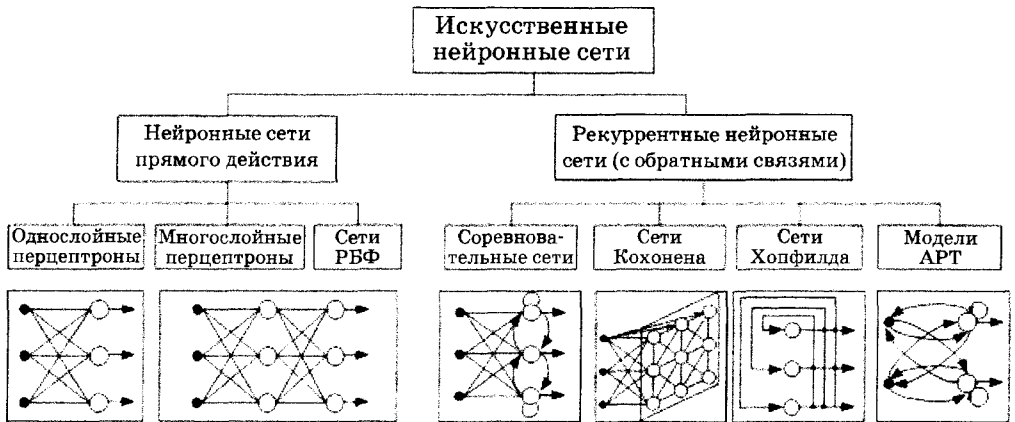


Рис. 8.6. Систематизация архитектур нейронных сетей

**Способность к обучению** является фундаментальным свойством мозга. В контексте ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения задания. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Это достигается с помощью *процедур обучения*. Путем анализа имеющихся в распоряжении входных и выходных данных веса и смещения сети автоматически настраиваются так, чтобы минимизировать разность между желаемым сигналом и полученным на выходе в результате моделирования.

Ошибка обучения определяется путем прогона через сеть всех примеров и определяется как разница выходных и целевых значений. Эти разницы позволяют сформировать функцию оценок. В качестве такой функции чаще всего берется сумма квадратов ошибок  $E$ . Если после прохождения нескольких циклов  $E \leq E_{\text{доп}}$ , обучение считается законченным, в противном случае циклы повторяются (рис. 8.7).

Для конструирования процесса обучения, во-первых, необходимо знать доступную для сети информацию. Эта информация определяет парадигму обу-

чения. Во-вторых, необходимо задать правила обучения, т. е. определить, как модифицировать весовые параметры сети в процессе настройки. Алгоритм обучения – это набор формул, который позволяет по вектору ошибки вычислить требуемые поправки для весов сети.

Существуют три парадигмы обучения: «с учителем», «без учителя» (самообучение) и смешанная.

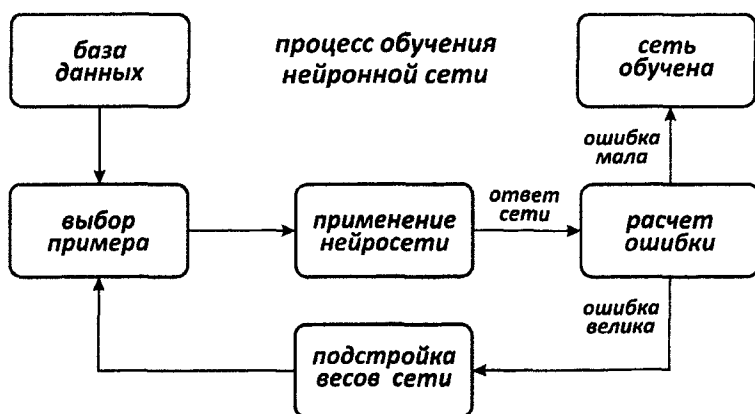


Рис. 8.7. Схема процесса обучения нейронной сети

В *многослойных сетях* нейроны объединяются в слои. *Слой* – это совокупность нейронов с единым входным сигналом. Внешние входные сигналы подаются на входы нейронов первого слоя, а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев.

Стандартная  $L$ -слойная сеть прямого распространения состоит из слоя входных узлов,  $(L - 2)$  скрытых слоев и выходного слоя, соединенных последовательно в прямом направлении и не содержащих связей между элементами внутри слоя и обратных связей между слоями. На рисунке 8.8 приведена структура трехслойной сети.

Наиболее популярный класс многослойных сетей прямого распространения образуют многослойные перцептроны, в которых каждый вычислительный элемент использует пороговую или сигмоидальную (логистическую) функцию активации.

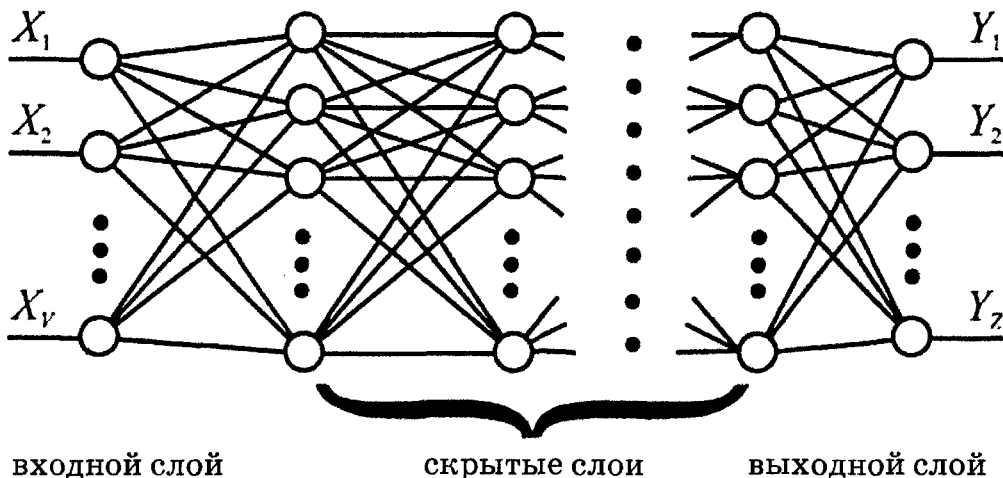


Рис. 8.8. Типовая структура многослойной сети прямого распространения

Теоретически для моделирования любой задачи достаточно многослойного персептрона с двумя промежуточными слоями (в точной формулировке этот результат известен как теорема Колмогорова), но иногда бывает выгодно увеличить число слоев, уменьшив общее число нейронов.

Имеются эвристические формулы, позволяющие оценить необходимое число нейронов в сети. Например, для однородной многослойной сети с сигмоидальными передаточными функциями необходимое число синаптических весов  $L_w$  удовлетворяет неравенству

$$\frac{mN}{1 + \log_2 N} \leq L_w \leq m \left( \frac{N}{m} + 1 \right) (n + m + 1) + m,$$

где  $n$  – размерность входного сигнала,  $m$  – размерность выходного сигнала,  $N$  – число элементов обучающей выборки. В этом случае число нейронов в двухслойной сети

$$L = \frac{L_w}{n + m}.$$

Известны и другие подобные формулы, например

$$2(L + n + m) \leq N \leq 10(L + n + m); \quad \frac{N}{10} - n - m \leq L \leq \frac{N}{2 - n - m}.$$

В *PNN-сетях* выходному значению приписывается вероятностный смысл. При обучении такой сети время тратится практически только на то, что-

бы подавать ей на вход обучающие наблюдения, и сеть работает настолько быстро, насколько это вообще возможно. Существенным недостатком таких сетей является их объем. *PNN*-сеть фактически вмещает в себя все обучающие данные, поэтому она требует много памяти и может медленно работать.

*PNN*-сети особенно полезны при пробных экспериментах (например, когда нужно решить, какие из входных переменных использовать), так как благодаря короткому времени обучения можно быстро проделать большое количество пробных тестов.

Отличие *сетей GRNN* состоит в том, что они используются в задачах обобщенной регрессии, анализа временных рядов и аппроксимации функций, а *PNN* – в задачах классификации. *GRNN*-сеть обучается почти мгновенно, но может получиться большой и медленной.

Самоорганизующиеся *карты Кохонена* обладают благоприятным свойством сохранения топологии, когда близкие входные примеры возбуждают близкие выходные элементы. Это свойство воспроизводит важный аспект карт признаков в коре головного мозга высокоорганизованных животных. Поддерживая такое топологическое свойство, карта Кохонена близким кластерам входных векторов ставит в соответствие близко расположенные нейроны.

Хопфилд использовал функцию энергии как инструмент для построения рекуррентных сетей и для понимания их динамики. *Сеть Хопфилда* эволюционирует в направлении уменьшения своей энергии. Главное свойство энергетической функции состоит в том, что в процессе эволюции состояний сети согласно уравнению, она уменьшается и достигает локального минимума (аттрактора), в котором она сохраняет постоянную энергию. Это позволяет решать комбинаторные задачи оптимизации, если они могут быть сформулированы как задачи минимизации энергии. Сети Хопфилда обладают ассоциативными возможностями. Они относятся к классу рекуррентных нейронных сетей, обладающих тем свойством, что за конечное число тактов времени они из произвольного начального состояния приходят в состояние устойчивого равновесия, называемое аттрактором.

#### 8.4. Области применения нейронных сетей

Типичной для нейросетевого подхода можно считать задачу распознавания букв в рукописном тексте. Пусть дано растровое черно-белое изображение буквы размером  $30 \times 30$  пикселей. Оно преобразуется во входной вектор из  $30 \times 30 = 900$  двоичных символов. Строится нейросеть с 900 входами и 33 выходами, которые помечены буквами. В результате обучения достигается такое состояние, что если на входе сети, например буква «А», то максимальное значение выходного сигнала наблюдается на выходе «А».

Многие задачи, для решения которых используются нейронные сети, могут рассматриваться как частные случаи следующих основных проблем:

- построение функции по конечному набору значений;
- оптимизация;
- построение отношений на множестве объектов;
- распределенный поиск информации и ассоциативная память;
- фильтрация;
- сжатие информации;
- идентификация динамических систем и управление ими;
- нейросетевая реализация классических задач и алгоритмов вычислительной математики: решение систем линейных уравнений, решение задач математической физики сеточными методами и др.

К задачам, успешно решаемым нейронными сетями на данном этапе их развития, относятся:

- распознавание зрительных, слуховых образов; огромная область применения: от распознавания текста и целей на экране радара до систем голосового управления;
- ассоциативный поиск информации и создание ассоциативных моделей; синтез речи; формирование естественного языка;
- формирование моделей и различных нелинейных и трудно описываемых математических систем, прогнозирование развития этих систем во времени;

- применение на производстве; прогнозирование развития циклонов и других природных процессов, прогнозирование изменений курсов валют и других финансовых процессов;
- системы управления и регулирования с предсказанием; управление роботами, другими сложными устройствами;
- разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключая логический вывод; особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере.

Самоорганизующиеся карты Кохонена могут быть использованы для проектирования многомерных данных, кластеризации входных векторов, аппроксимации плотности. Эта сеть успешно применяется для распознавания речи, обработки изображений, в робототехнике и в задачах управления.

## 9. НЕЙРОННЫЕ СЕТИ

*Нейронные сети (НС)* – это класс моделей, основанных на биологической аналогии с мозгом человека и предназначенных, после прохождения этапа обучения, для решения разнообразных задач анализа данных. Искусственная нейронная сеть может рассматриваться как *направленный граф со взвешенными связями, в котором искусственные нейроны являются вершинами.*

При применении НС прежде всего встает вопрос о конкретной архитектуре сети. Размер и структура сети должны соответствовать сущности исследуемого явления. *Для определения архитектуры сети могут применяться методы искусственного интеллекта.*

На этапе обучения нейроны сети итеративно обрабатывают входные данные и корректируют свои веса так, чтобы сеть наилучшим образом прогнозировала данные, на которых осуществляется обучение. Геометрически процесс обучения можно представить как построение и последовательное усложнение некоей гиперповерхности (рис. 9.1). Алгоритм сходится к одному из ее локальных минимумов. При этом в первой части эпох поверхность описывает и уточняет общую структуру данных, а затем заикливается на мелочах, связанных с особенностями обучающей выборки.

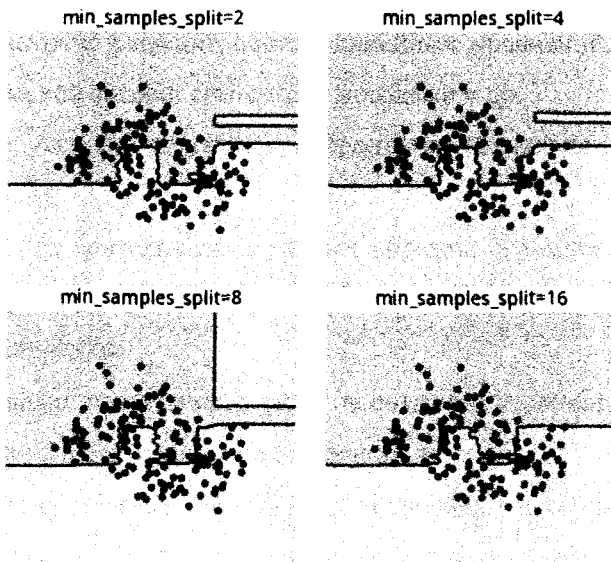


Рис. 9.1. Усложнение разделяющей гиперповерхности

Обученная НС отражает закономерности, присутствующие в данных, и оказывается функциональным эквивалентом некоторой модели зависимостей, но эти зависимости не могут быть представлены в явном виде. *НС представляют собой типичный пример нетеоретического подхода («черный ящик»)*. Их полезность оценивается по точности прогнозов и их прикладной ценности. Тем не менее, методы НС могут применяться и для построения *объясняющей модели* явления, поскольку могут выделять *значимые переменные* и наиболее важные входные переменные.

*Преимущество НС* состоит в том, что они могут аппроксимировать любую непрерывную функцию без гипотез о виде модели и степени важности переменных.

*Недостаток НС* состоит в том, что результат зависит от начальных установок сети и его невозможно интерпретировать в традиционных аналитических терминах.

Для конструирования процесса обучения необходимо знать доступную для сети информацию. Эта информация определяет *парадигму обучения*. Алгоритм обучения – это набор формул, который позволяет по вектору ошибки вычислить требуемые поправки для весов сети. Существуют три парадигмы обучения: «с учителем», «без учителя» (самообучение) и смешанная.

### 9.1. Распространение ошибок

По архитектуре связей ИНС могут быть сгруппированы в два класса (рис. 8.6): *сети прямого распространения*, в которых графы не имеют петель, и рекуррентные сети, или *сети с обратными связями*.

**Как работает прямое распространение?** Нейросеть состоит из *слоев*, которые включают в себя настраиваемое количество *нейронов*. Если слои – полностью связанные, то в них каждый нейрон связан со всеми нейронами предыдущего слоя, и к каждой из связей нейрон хранит вес, который определяет, насколько он будет учитывать сигнал с нейрона, с которым он связан. Обучение нейросети заключается в обучении весов, – изменении их значений так, чтобы на одни сигналы нейроны научились «реагировать» больше, а на другие – мень-



ше. Помимо полносвязных, бывают и варианты слоев, но суть их работы от этого не меняется. Связи есть ничто иное, как скрытые зависимости, которые нейросеть должна научиться замечать.

В самом начале обучения нейросеть работает неверно, так как веса в самом начале задаются случайно. Как же настраивать значения весов так, чтобы они помогали решать нашу задачу? На помощь приходит *функция потерь*, которая показывает, как нужно штрафовать модель, и механизм *обратного распространения ошибки*.

**Как работает обратное распространение ошибки?** Для каждой задачи выбирается специальная функция потерь, задача которой – получить на вход результат прямого распространения нейронной сети, и показать, насколько ответ был близок к цели. Для типичных задач (например, классификация или регрессия) можно использовать типичные функции потерь для этих целей, но для более трудных задач (например, построение скелета человека по фото) требуется проявить фантазию и высокий профессионализм, чтобы составить функцию потерь так, чтобы она позволяла решить поставленную задачу. Функцию потерь можно представить как сложную функцию от результирующего, финального слоя, который на входе содержит веса и значения, которые поступают вместе с ними с предыдущих слоев, а также известные данные.

Мы можем определить направление наискорейшего убывания функции, посчитав антиградиент функции потерь (в нашем случае – частную производную по всем весам) в полученной точке. Таким образом, мы получим вектор частных производных функции потерь по весам, которые требуется домножить на шаг градиента (также известный как *learning rate*), и, при сложении весов к нейронам предыдущего слоя (начиная с последнего слоя) с этими значениями поэлементно, получим изменение весов так, чтобы при этом понижалось значение функции потерь, а значит – происходило итеративное приближение к желаемому результату. С остальными слоями производится такая же процедура, от слоя к слою, до самого первого.

Так за каждую итерацию модель начинает всё больше учитывать свойства данных, и в зависимости от этого менять веса. О чем стоит позаботиться специалисту – это о формировании дэйтасета подходящим образом, о подборе функции потерь и о выборе архитектуры нейросети.

## 9.2. Многослойные сети. Некоторые архитектуры сетей

В *многослойных сетях* нейроны объединяются в слои. *Слой* – это совокупность нейронов с единым входным сигналом. Внешние входные сигналы подаются на входы нейронов первого слоя, а выходами сети являются выходные сигналы последнего слоя. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев.

Стандартная  $L$ -слойная сеть прямого распространения состоит из слоя входных узлов,  $(L - 2)$  скрытых слоев и выходного слоя, соединенных последовательно в прямом направлении и не содержащих связей между элементами внутри слоя и обратных связей между слоями. Ранее, на рисунке 8.8 приведена структура трехслойной сети.

**Многослойный перцептрон.** Наиболее популярный класс многослойных сетей прямого распространения образуют многослойные перцептроны, в которых каждый элемент использует пороговую или сигмоидальную (логистическую) функцию активации (рис. 9.2).

Непрерывность первой производной позволяет обучать сеть градиентными методами. Функция симметрична относительно точки  $(s = 0, y = 0.5)$ , что существенно в работе сети. Многослойный перцептрон может формировать сколь угодно сложные границы принятия решения и реализовывать произвольные булевы функции. Разработка алгоритма обратного распространения для определения весов в многослойном перцептроне сделала эти сети наиболее популярными у исследователей и пользователей нейронных сетей.

Теоретически для моделирования любой задачи достаточно многослойного перцептрона с двумя промежуточными слоями (этот результат известен как **теорема Колмогорова**), но иногда бывает выгодно увеличить число слоев, уменьшив общее число нейронов.

Логистическая функция активации

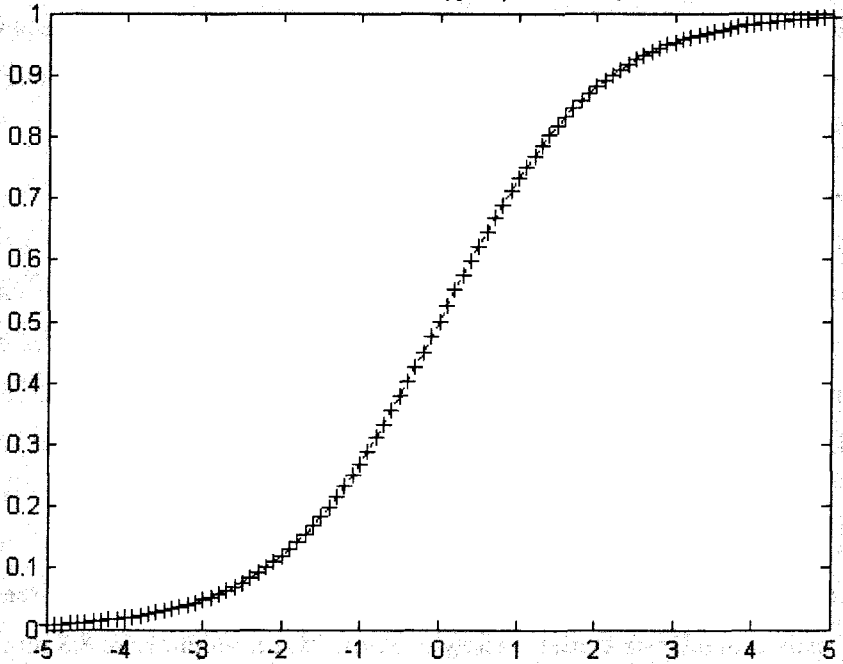


Рис. 9.2. Логистическая функция активации  $f(s) = \frac{1}{1+\exp(-s)}$

Имеются эвристические формулы, позволяющие оценить необходимое число нейронов в сети. Например, для однородной многослойной сети с сигмоидальными передаточными функциями необходимое число синаптических весов  $L_w$  удовлетворяет неравенству

$$\frac{mN}{1+\log_2 N} \leq L_w \leq m \left( \frac{N}{m} + 1 \right) (n+m+1) + m,$$

где  $n$  – размерность входного сигнала,  $m$  – размерность выходного сигнала,  $N$  – число элементов обучающей выборки. В этом случае число нейронов в двух-слойной сети

$$L = \frac{L_w}{n+m}.$$

Известны и другие подобные формулы, например

$$2(L+n+m) \leq N \leq 10(L+n+m); \quad \frac{N}{10} - n - m \leq L \leq \frac{N}{2-n-m}.$$

**RBF-сети.** Сети, использующие радиальные базисные функции (*Radial Basis Function Network* – сеть с радиальными базисными элементами, или *RBF-сети*), являются частным случаем двухслойной сети без обратных связей, которая содержит скрытый слой радиально симметричных скрытых нейронов. Каждый элемент скрытого слоя использует в качестве активационной функции  $f(s)$  радиальную базисную функцию типа гауссовой (рис. 9.3).

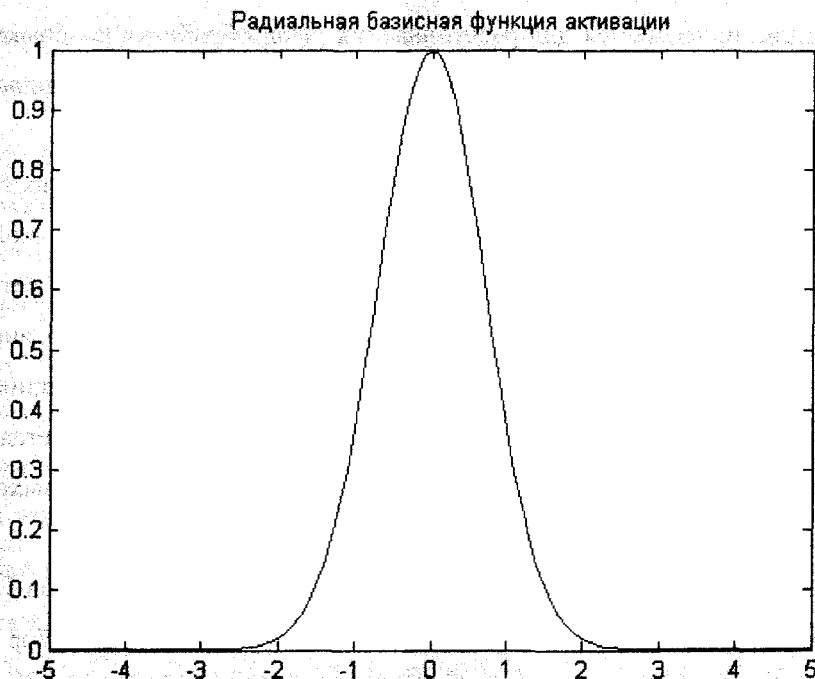


Рис. 9.3. Радиальная базисная функция активации

Известны 2 специальных типа радиальных базисных сетей: сети GRNN (Generalized Regression Neural Networks) и сети PNN (Probabilistic Neural Networks).

PNN-сети используются в задачах классификации. В них выходному значению приписывается вероятностный смысл. PNN-сеть фактически вмещает в себя все обучающие данные, поэтому она требует много памяти и может медленно работать.

GRNN-сети используются в задачах обобщенной регрессии, анализа временных рядов и аппроксимации функций. GRNN-сеть обучается почти мгновенно, но может получиться большой и медленной.

Самоорганизующиеся карты Хохонена близким кластерам входных векторов ставит в соответствие близко расположенные нейроны – сохраняют топологию данных. Они могут быть использованы для проектирования многомерных данных, кластеризации входных векторов, аппроксимации плотности. Эта сеть успешно применяется для распознавания речи, обработки изображений, в робототехнике и в задачах управления. Такая сеть является специальным случаем сети, обучающейся методом соревнования.

Хопфилд использовал функцию энергии как инструмент для построения рекуррентных сетей и для понимания их динамики. Сеть Хопфилда эволюционирует в направлении уменьшения своей энергии. Спроектировать сеть Хопфилда – это значит создать рекуррентную сеть с множеством точек равновесия, таких, что при задании начальных условий сеть в конечном счете приходит в состояние покоя в одной из этих точек. Свойство рекурсии проявляется в том, что выход сети подается обратно на вход. Можно надеяться, что выход сети установится в одной из точек равновесия.

Сеть ART – попытка приблизить механизм запоминания образов в ИНС к биологическому. Результатом работы ART является устойчивый набор запомненных образов и возможность выборки «похожего» вектора по произвольному предъявленному на входе вектору.

Сеть ART-1, предложенная Карпентером и Гроссбергом в 1986 году, представляет собой векторный классификатор и обучается без учителя, лишь на основании предъявляемых входных векторов. ART-1 работает только с двоичными векторами, состоящими из нулей и единиц. В настоящее время известно много разновидностей этой модели. Например, ART-2 запоминает и классифицирует непрерывные входные векторы, FART использует нечеткую логику.

Важнейшей особенностью нейросетей является возможность организации распределенного поиска информации и наличие ассоциативной памяти.

Ассоциативная память, или память, адресуемая по содержанию, доступна по указанию заданного содержания. Содержимое памяти может быть вызвано даже по частичному входу или искаженному содержанию. Ассоциативная память чрезвычайно желательна при создании мультимедийных информационных баз данных.

Нейросети сегодня широко используются для решения классической проблемы производства (раскопок) знаний из накопленных данных. Обучаемые нейронные сети могут производить из данных скрытые знания: создается навык предсказания, классификации, распознавания образов и т. п.

### 9.3. Функции создания нейронных сетей в ИМС MatLab

- **network** – функция создания нейронной сети пользователя (шаблона сети);
- **newp** – функция создания перцептрона;
- **newlin** – функция создания слоя линейных нейронов;
- **newlind** – функция проектирования линейной НС;
- **newc** – функция создания слоя Кохонена;
- **newcf** – функция создания каскадной НС;
- **newelm** – функция создания сети Элмана;
- **newff** – функция создания многослойной НС;
- **newfftd** – функция создания многослойной НС с задержками по вход;
- **newhop** – функция создания сети Хопфилда;
- **newgrnn** – функция создания обобщенно-регрессионной сети;
- **newpnn** – функция создания вероятностной НС;
- **newrb** – функция создания сети с радиальными базисными элементами;
- **newrbe** – функция создания сети с радиальными базисными элементами с нулевой ошибкой на обучающей выборке;
- **newlvq** – функция создания сети встречного распространения;
- **newsom** – функция создания карты Кохонена.

## Примеры создания и использования нейронных сетей

Пример 1. Нейронная сеть для аппроксимации функции. Файл nn\_1.m, рис.1

```
clear; clc; %Регрессионные нейросети типа GRNN и RBE
x=[-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1];
y=[1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1];
%Сеть GRNN
a=newgrnn(x,y,0.01); %создание НС с погрешностью 0.01
x1=-1:0.05:1; y1=sim(a,x1); %опрос НС
%Сеть RBE
b=newrbe(x,y); %создание НС
y2=sim(b,x1); %опрос НС

t=-1:0.01:1; T=t.^2;
subplot(1,2,1);
plot(t,T,'LineWidth',2); grid;
hold on; plot(x,y,'rs','LineWidth',3);
hold on; plot(x1,y1,'r','LineWidth',3);
title('Сеть GRNN','FontSize',14,'FontWeight','Bold');
axis([-1 1 -0.1 1]);
subplot(1,2,2);
plot(t,T,'LineWidth',2); grid;
hold on; plot(x,y,'rs','LineWidth',3);
hold on; plot(x1,y2,'g','LineWidth',3);
title('Сеть RBE','FontSize',14,'FontWeight','Bold');
axis([-1 1 -0.1 1]);
```

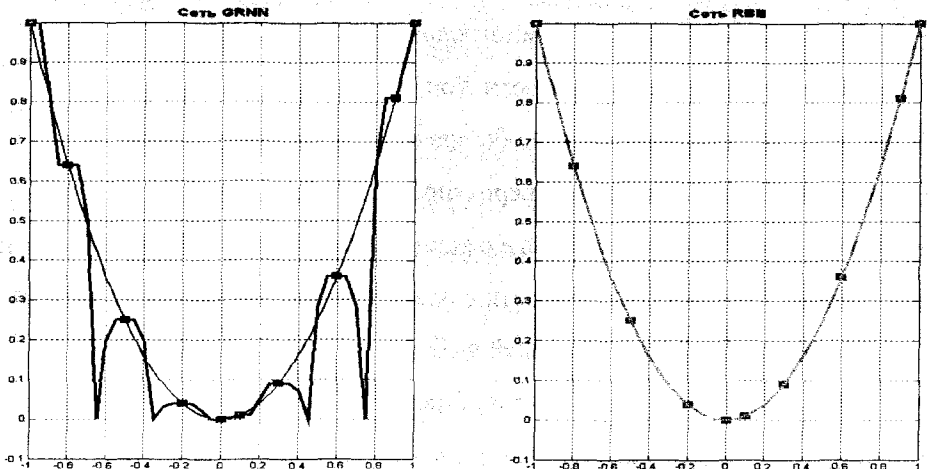


Рис. 9.4. Аппроксимация параболы по 10 точкам

*Пример 2. Восстановление линейной зависимости по зашумленным данным.*

файл nn\_2.m, рис.2

```
clear; clc; %Восстановление линейной зависимости
x=0.1:0.3:2.2;
y0=2*x-1;
y=y0+0.5*randn(size(x));
s=newlind(x,y); %создание НС
x1=0:0.2:2;
y1=sim(s,x1);
plot(x,y0,'LineWidth',2); grid;
hold on; plot(x,y,'rs','LineWidth',2);
hold on; plot(x1,y1,'go','LineWidth',3);
hold on; plot(x1,y1,'g','LineWidth',3);
title('Восстановление линейной зависимости, линейная НС',
'FontSize',14,'FontWeight','Bold')
```

*Пример 3. Прогнозирование значений процесса. Файл nn\_3.m, рис.3.*

```
clear; clc; %Прогнозирование значений процесса по 5 точкам
t=0:0.1:5;
x=sin(t*pi); %Процесс
n=length(x)-1;
%Dанные для обучения
P=zeros(5,n);
P(1,2:n)=x(1,1:n-1);
P(2,3:n)=x(1,1:n-2);
P(3,4:n)=x(1,1:n-3);
P(4,5:n)=x(1,1:n-4);
P(5,6:n)=x(1,1:n-5);
s=newlind(P,x(1:n)); %создание НС
y=sim(s,P);
%Прогноз в точке k по предыдущим 5 точкам
k=50;
z(1,1)=x(k-1); z(2,1)=x(k-2); z(3,1)=x(k-3); z(4,1)=x(k-4); z(5,1)=x(k-5);
y1=sim(s,z);
plot(t,x,'LineWidth',2); grid;
hold on; plot(t(1:n),y,'rs','LineWidth',2);
hold on; plot(t(k),y1,'gs','LineWidth',2);
title('Прогнозирование значений процесса по 5 точкам',...
'FontSize',14,'FontWeight','Bold');
```



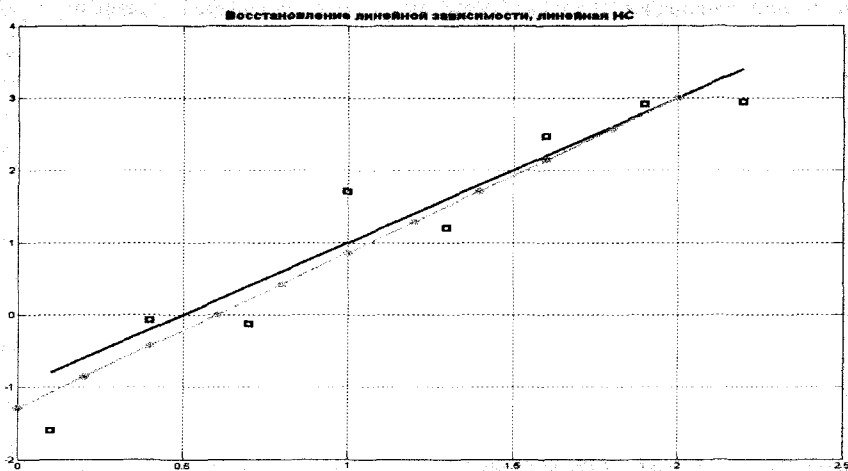


Рис. 9.5. Восстановление линейной зависимости по зашумленным данным

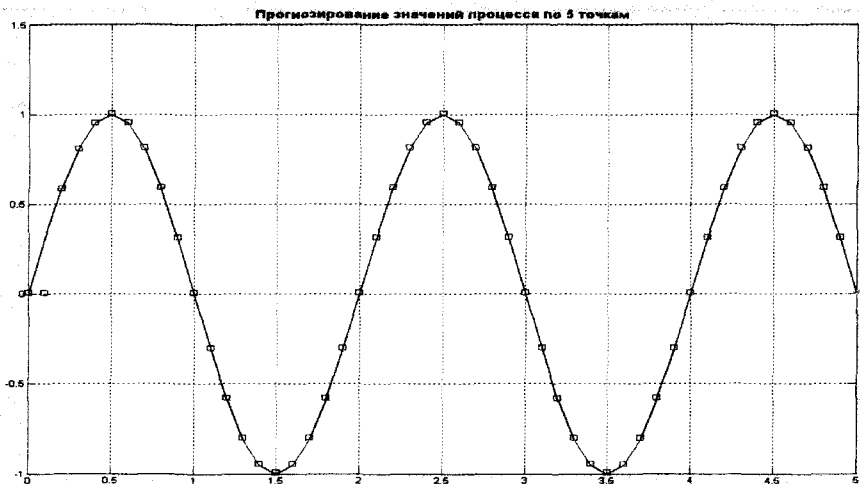


Рис. 9.6. Прогнозирование значений процесса

Пример 4. Классификация с помощью перцептрона. Файл pp\_5.m, рис.4,5

```
clear; clc; %Классификация с помощью перцептрона
load dat_linn_2; Z=dat_linn_2;
T0=Z(:,1)';
I=find(T0==0); n=length(I); %n=171
J=find(T0==1); m=length(J); %m=67
P1=Z(1:m,2:3)'; P2=Z(n+1:n+m,2:3)';
P=[P1 P2]; %обучающая выборка - матрица 2x(m+m)
Q=Z(m+1:n,2:3)'; %контрольная выборка, для нее индекс = 0 (2x104)
T=[zeros(1,m) ones(1,m)]; %индексы для обучения
%Графическое представление классов
```

```

%plotpv(P,T);
plot(P1(1,:),P1(2,:),'sb','LineWidth',3); grid;
hold on; plot(P2(1,:),P2(2,:),'or','LineWidth',3);
%hold on; plot(Q(1,:),Q(2,:),'*g','LineWidth',3);
%=====Создание перцептрона=====
%Границы изменения входов:
gr=[min(Z(1,:)) max(Z(1,:)); min(Z(1,:)) max(Z(1,:))];
My_net=newp(gr,1); %1 нейрон
My_net=init(My_net); %инициализация персептрона
%Адаптивная настройка
for k=1:5;
    [My_net,Y,E]=adapt(My_net,P,T);
end;
a=sim(My_net,Q);
hold on; plotpc(My_net.IW{1},My_net.b{1});
b0=sim(My_net,Z(1:m,2:3)');
b1=sim(My_net,Z(n+1:n+m,2:3)');
bq=sim(My_net,Q);
figure;
plot(b0,'LineWidth',2);
hold on; plot(-b1+2,'r','LineWidth',2);
hold on; plot(bq+2,'g','LineWidth',2);

```

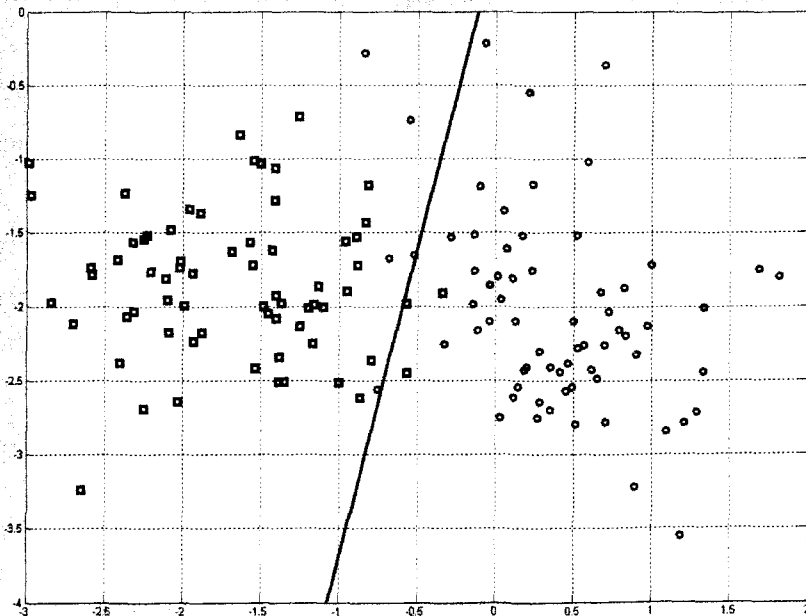


Рис. 9.7. Классификация с помощью однослойного перцептрона (5 итераций)

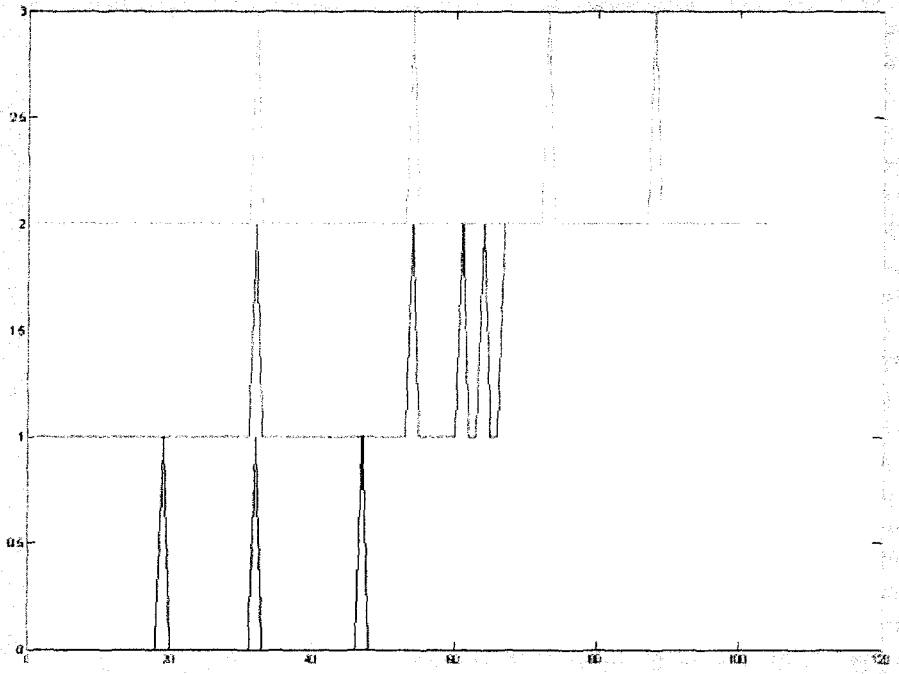


Рис. 9.8. Ошибки при классификации (обучающие выборки и контрольная выборка)

## 10. ЭВОЛЮЦИОННОЕ МОДЕЛИРОВАНИЕ И ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ

### 10.1. Эволюционное моделирование

Эволюционное моделирование, ЭМ (Evolutionary computation):

1) использует признаки теории Дарвина для построения интеллектуальных систем (методы группового учёта, генетические алгоритмы). Является частью более обширной области искусственного интеллекта – вычислительного интеллекта.

2) направление в математическом моделировании, объединяющее компьютерные методы моделирования эволюции, а также близкородственные по источнику заимствования идеи (теоретическая биология), другие направления в эвристическом программировании. Включает в себя как разделы генетические алгоритмы, эволюционные стратегии, эволюционное программирование, искусственные нейронные сети, нечеткую логику.

Эволюционное моделирование представляет собой одно из быстро развивающихся направлений математического моделирования, объединяющее компьютерные методы моделирования эволюционных процессов в естественных и искусственных системах, такие как генетические алгоритмы, эволюционные стратегии, эволюционное программирование и другие эвристические методы.

Главная трудность построения вычислительных систем, основанных на принципах эволюции живой природы и применении этих систем в прикладных задачах, состоит в том, что природные системы достаточно хаотичны, а действия исследователей носят направленный характер. Компьютер используется как инструмент для решения определенных задач, которые пользователь формулирует, акцентируя внимание на максимально быстром решении при минимальных затратах.

Природные системы не имеют подобных целей или ограничений, во всяком случае, они не очевидны. Однако биологические системы обладают свойствами:

- воспроизводства;
- адаптации;

- самоисправления;
- устойчивости;
- гибкости

и многими другими, которые лишь фрагментарно присутствуют в искусственных системах.

## 10.2. Модели возникновения МГИС

В начале 1970-х гг. лауреат Нобелевской премии М. Эйген совершил впечатляющую попытку построения моделей возникновения в ранней биосфере Земли молекулярно-генетических систем обработки информации или молекулярно-генетических информационных систем (МГИС). Наиболее известная из них – модель «квазивидов», описывающая простую эволюцию полинуклеотидных информационных последовательностей. Вслед за Эйгеном в 1980 г. новосибирскими учеными В. Ратнером и В. Шаминам была предложена модель сайзеров.

В модели квазивидов рассматривается поэтапная эволюция популяции информационных последовательностей (векторов), компоненты которых приобретают небольшое число дискретных значений. Приспособленность «особей» в моделях задается как функции векторов. На каждом этапе происходит отбор особей в популяции следующего поколения с вероятностями, пропорциональными их приспособленности, а также мутации особей – случайные равновероятные замены компонентов векторов. Модель сайзера в простейшем случае рассматривает систему из трех типов макромолекул: полинуклеотидной матрицы и ферментов трансляции и репликации, кодированных этой матрицей.

Полинуклеотидная матрица – это как бы запоминающее устройство, в котором хранится информация о функциональных единицах сайзера – ферментах. Фермент трансляции обеспечивает «изготовление» произвольного фермента по записанной в матрице информации. Фермент репликации обеспечивает копирование полинуклеотидной матрицы.

Сайзеры достаточны для самовоспроизведения. Включая в схему сайзера дополнительные ферменты, кодируемые полинуклеотидной матрицей, можно

обеспечить сайзер любыми свойствами, например свойством регулирования синтеза определенных ферментов и адаптации к изменениям внешней среды.

### **10.3. Применение в задачах функциональной оптимизации**

ЭМ часто используются для организации стохастического поиска, особенно в случае многомодальных задач, когда детерминированные методы оптимизации или более простые стохастические методы не позволяют исследовать поведение целевой функции вне областей локальных оптимумов. Методы ЭМ не гарантируют обнаружения глобального оптимума за полиномиальное время.

Практический интерес к ним объясняется тем, что эти методы, как показывает практика, позволяют найти лучшие (или «достаточно хорошие») решения очень трудных задач поиска за меньшее время, чем другие, обычно применяемые в этих случаях, методы. Типичное ограничение на их применение заключается в необходимости многократного вычисления целевой функции (под словом «многократно» обычно подразумеваются числа от сотен до миллионов).

Тем не менее, методы ЭМ оказались достаточно эффективными для решения ряда реальных задач инженерного проектирования, планирования, маршрутизации и размещения, управления портфелями ценных бумаг, поиска оптимальных энергетических состояний химических и молекулярных структур, а также во многих других областях, допускающих подходящий набор представлений, операторов, объемов и структур популяций и т. д.

### **10.4. ЭМ как исследовательский метод в информатике**

Поскольку эволюция, по-видимому, представляет собой основу механизма обработки информации в естественных системах, исследователи стремятся построить теоретические и компьютерные модели, реально объясняющие принципы работы этого механизма. Для исследований этого направления характерно понимание, что модели должны содержать не только рождение и смерть популяций, но и что-то между ними. Чаще всего привлекаются следующие концепции.

**Роевой интеллект** (*Swarm intelligence*) описывает коллективное поведение децентрализованной самоорганизующейся системы. Рассматривается в теории искусственного интеллекта как метод оптимизации.

Термин был введен Херардо Бени и Ван Цзином в 1989 г., в контексте системы клеточных роботов. Системы роевого интеллекта, как правило, состоят из множества агентов, локально взаимодействующих между собой и с окружающей средой. Сами агенты обычно довольно просты, но все вместе, локально взаимодействуя, создают так называемый роевой интеллект. Примером в природе может служить колония муравьев, рой пчёл, стая птиц, рыб и т. д.

**Коллективный интеллект** – термин, который появился в середине 1980-х гг. в социологии при изучении процесса коллективного принятия решений. Исследователи из *NJIT* определили коллективный интеллект как способность группы находить решения задач более эффективные, чем лучшее индивидуальное решение в этой группе.

**Социологическое направление** – поскольку человеческое общество представляет собой реальный, к тому же хорошо поддающийся наблюдению и задокументированный (в отличие от человеческого мозга) инструмент обработки информации, социологические метафоры и реминисценции присутствуют в работах по кибернетике и смежным направлениям с самого их возникновения.

Если роевой интеллект ориентирован на получение сложного поведения в системе из простых элементов, этот подход, наоборот, исследует построение простых и специальных объектов на базе сложных и универсальных: *«государство глупее, чем большинство его членов»*.

Для этого направления характерно стремление дать социологическим понятиям определения из области информатики. Элита определяется как носитель определенной частной модели реального мира, а базис (т. е. народ) играет роль арбитра между элитами. Эволюционный процесс заключается в порождении и гибели элит. Базис не в состоянии разобраться в сути идей и моделей, представляемых элитами, и не ставит перед собой такой задачи. Однако именно в силу своей невовлеченности он сохраняет способность к ясной эмоциональной

оценке, позволяющей ему легко отличать харизматические элиты от загнивающих, пытающихся сохранить свои привилегии, понимая, что их идея или модель не подтвердилась.

### 10.5. Генетические алгоритмы

Адекватным средством реализации процедур эволюционного моделирования являются *генетические алгоритмы*. Идея генетических алгоритмов «подсмотрена» у систем живой природы, у систем, эволюция которых разворачивается в сложных системах достаточно быстро.

*Генетический алгоритм* – это алгоритм, основанный на имитации генетических процедур развития популяции в соответствии с принципами эволюционной динамики. Часто используется для решения задач оптимизации (в т. ч. многокритериальной), поиска, управления.

Данные алгоритмы адаптивны, развивают решения, развиваются сами. Особенность этих алгоритмов – их успешное использование при решении сложных проблем (проблем, для которых невозможно построить *алгоритм* с полиномиальной алгоритмической сложностью).

*Пример.* Работу банка можно моделировать на основе генетических алгоритмов. С их помощью можно выбирать оптимальные банковские проценты (вкладов, кредитов) некоторого банка в условиях конкуренции с тем, чтобы привлечь больше клиентов (средств). Тот банк, который сможет привлечь больше вкладов, клиентов и средств, и выработает более привлекательную *стратегию поведения* (эволюции) – тот и выживет в условиях естественного отбора. Филиалы такого банка (гены) будут лучше приспособливаться и укрепляться в экономической нише, а возможно, и увеличиваться с каждым новым поколением. Каждый филиал банка (индивид популяции) может быть оценен мерой его приспособленности. В основе таких мер могут лежать различные критерии, например аналог экономического потенциала – *рейтинг надежности* банка или соотношение привлеченных и собственных средств банка. Такая оценка эквивалентна оценке того, насколько эффективен организм при конкуренции за ресурсы, т. е. его выживаемости, биологическому потенциалу. При этом банки (филиалы)



могут приводить к появлению потомства (новых банков, получаемых в результате слияния или распада), сочетающего те или иные (экономические) характеристики родителей. Например, если один банк имел качественную политику кредитования, а другой – эффективную инвестиционную политику, то новый банк может приобрести и то, и другое. Наименее приспособленные банки (филиалы) совсем могут исчезнуть в результате эволюции. Таким образом, отрабатывается генетическая процедура воспроизводства новых банков (нового поколения), более приспособленных и способных к выживанию в процессе эволюции банковской системы. Эта политика со временем пронизывает всю банковскую «популяцию», обеспечивая достижение цели – появления эффективно работающей, надежной и устойчивой банковской системы.

#### **10.6. Естественный отбор в природе**

Согласно эволюционной теории, каждый биологический вид целенаправленно развивается и изменяется для того, чтобы наилучшим образом приспособиться к окружающей среде. Эволюция в этом смысле представляет процесс оптимизации всех живых организмов. Природа решает эту задачу оптимизации путем естественного отбора. Его суть состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом благодаря передаче генетической информации (генетическому наследованию) потомки наследуют от родителей основные их качества. Таким образом, потомки сильных особей также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены нескольких десятков или сотен поколений средняя приспособленность особей данного вида заметно возрастает.

Генетические алгоритмы (ГА) и другие адаптивные методы поиска в последнее время часто используются для решения задач функциональной оптимизации. Они основаны на генетических процессах биологических организмов: биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора по принципу «выживает наиболее при-

способленный», открытому Чарльзом Дарвином. Подражая этому процессу, генетические алгоритмы способны «развивать» решения реальных задач, если те соответствующим образом закодированы. Например, генетические алгоритмы могут использоваться, чтобы проектировать строительные конструкции с максимальным отношением прочность/вес, или определять наименее расточительное размещение для нарезки форм из ткани. Они могут использоваться для интерактивного управления производственным процессом или балансировании загрузки на многопроцессорном компьютере.

Основные принципы генетических алгоритмов были сформулированы Голландом (Holland, 1975) и хорошо описаны во многих работах. В отличие от эволюции, происходящей в природе, генетические алгоритмы только моделируют те процессы в популяциях, которые являются существенными для развития. Точный ответ на вопрос: «Какие биологические процессы существенны для развития и какие нет?» все еще открыт для исследователей.

В природе особи в популяции конкурируют друг с другом за различные ресурсы, такие например, как пища или вода. Кроме того, члены популяции одного вида часто конкурируют за привлечение брачного партнера. Те особи, которые наиболее приспособлены к окружающим условиям, будут иметь относительно больше шансов для воспроизводства потомков. Слабо приспособленные особи либо совсем не произведут потомства, либо их потомство будет очень немногочисленным. Это означает, что гены от высокоадаптированных или приспособленных особей будут распространяться в увеличивающемся количестве на каждом последующем поколении. Комбинация хороших характеристик от различных родителей иногда может приводить к появлению «суперприспособленного» потомка, чья приспособленность больше, чем любого из его родителей. Таким образом, вид развивается, лучше и лучше приспособляясь к среде обитания.

Генетические алгоритмы используют прямую аналогию с таким механизмом. Они работают с совокупностью особей – популяцией, каждая из которых представляет возможное решение данной проблемы. Каждая особь оценивается мерой ее «приспособленности» согласно тому, насколько «хорошо» соответ-

ствующее ей решение задачи (в природе это эквивалентно оценке того, насколько эффективен организм при конкуренции за ресурсы). Наиболее приспособленные особи получают возможность воспроизводить потомство с помощью перекрестного скрещивания с другими особями популяции. Это приводит к появлению новых особей, которые сочетают в себе некоторые характеристики, наследуемые ими от родителей. Наименее приспособленные особи с меньшей вероятностью смогут воспроизвести потомков, так что те свойства, которыми они обладали, будут постепенно исчезать из популяции в процессе эволюции.

Так воспроизводится новая популяция допустимых решений, выбирая лучших представителей предыдущего поколения, скрещивая их и получая множество новых особей. Это новое поколение содержит более высокое соотношение характеристик, которыми обладают хорошие члены предыдущего поколения. Таким образом, из поколения в поколение хорошие характеристики распространяются по всей популяции. Скрещивание наиболее приспособленных особей приводит к тому, что исследуются наиболее перспективные участки пространства поиска. В конечном итоге, популяция будет эволюционировать к оптимальному решению задачи.

Дадим краткую справку о том, как устроены механизмы генетического наследования. В каждой клетке любого животного содержится вся генетическая информация данной особи. Эта информация записана в виде набора молекул ДНК, каждая из которых представляет собой цепочку, состоящую из молекул *нуклеотидов* четырех типов, обозначаемых *A*, *T*, *C* и *G*. Информацию несет порядок следования нуклеотидов в ДНК. Таким образом, генетический код особи – это длинная строка, где используются всего 4 символа. В животной клетке каждая молекула ДНК окружена оболочкой, такое образование называется *хромосомой*.

Каждое врожденное качество особи (цвет глаз, наследственные болезни, тип волос и т. д.) кодируется определенной частью хромосомы, которая называется *геном* этого свойства. Например, ген цвета глаз содержит информацию, кодирующую определенный цвет глаз. Различные значения гена называются его *аллелями*.

При размножении особей происходит слияние двух родительских половых клеток, и их ДНК взаимодействуют, образуя ДНК потомка. Основной способ взаимодействия – *кроссовер* (*crossover*) или *скрещивание*. При этом ДНК предков делятся на две части, а затем обмениваются своими половинками.

При наследовании возможны *мутации*, в результате которых могут измениться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Если эти новые свойства полезны, они, скорее всего, сохранятся в данном виде. При этом произойдет скачкообразное повышение приспособленности вида.

### 10.7. Что такое генетический алгоритм

Цель в оптимизации с помощью генетических алгоритмов (ГА) состоит в том, чтобы найти лучшее возможное решение задачи по одному или нескольким критериям. Чтобы реализовать генетический алгоритм, нужно сначала выбрать подходящую структуру для представления этих решений. В постановке задачи поиска экземпляр этой структуры данных представляет точку в пространстве поиска всех возможных решений.

Структура данных генетического алгоритма состоит из одной или большего количества хромосом (обычно из одной). Как правило, хромосома – это битовая строка, так что термин строка часто заменяет понятие «хромосома». Каждая хромосома (строка) представляет собой конкатенацию (объединение) ряда подкомпонентов, называемых генами. Гены располагаются в различных позициях или локусах хромосомы и принимают значения, называемые аллелями. В представлениях с бинарными строками ген – бит, локус – его позиция в строке и аллель – его значение (0 или 1). Биологический термин «генотип» относится к полной генетической модели особи и соответствует структуре в ГА. Термин «фенотип» относится к внешним наблюдаемым признакам и соответствует вектору в пространстве параметров.

Пусть, например, речь идет о минимизации функции двух переменных  $f(x_1, x_2)$ .

Обычно методика кодирования реальных переменных  $x_1$  и  $x_2$  состоит в их преобразовании в двоичные целочисленные строки достаточной длины – достаточной для того, чтобы обеспечить желаемую точность. Предположим, что 10-разрядное кодирование достаточно и для  $x_1$ , и  $x_2$ . Установить соответствие между генотипом и фенотипом закодированных особей можно, разделив соответствующее двоичное целое число на  $2^n - 1$ . Например, 0000000000 соответствует 0/1023 или 0, тогда как 1111111111 соответствует 1023/1023 или 1. Оптимизируемая структура данных – 20-битная строка, представляющая конкатенацию кодировок  $x_1$  и  $x_2$ . Переменная  $x_1$  размещается в крайних левых 10 разрядах, тогда как  $x_2$  размещается в правой части генотипа особи (20-битовой строке). Генотип – точка в 20-мерном хеммининговом пространстве, исследуемом ГА. Фенотип – точка в двумерном пространстве параметров.

Чтобы оптимизировать структуру, используя ГА, нужно задать некоторую меру качества для каждой структуры в пространстве поиска. Для этой цели используется так называемая функция приспособленности или выживаемости. Часто в качестве функции приспособленности выступает сама целевая функция (например, наш двумерный пример).

Простой ГА случайным образом генерирует начальную популяцию структур. Работа ГА представляет собой итерационный процесс, который продолжается до тех пор, пока не пройдет заданное число поколений (или используется иной критерий остановки). На каждом поколении ГА реализуется одноточечный кроссовер и мутация.

Сначала пропорциональный отбор назначает каждой  $i$ -й особи вероятность  $P_s(i)$ , равную отношению ее приспособленности к суммарной приспособленности популяции:

$$P_s(i) = \frac{f(i)}{\sum_{j=1}^n f(j)},$$

где  $n$  – число особей в популяции,  $f(i)$  – значение функции приспособленности  $i$ -й особи.

Затем происходит отбор (с замещением) всех  $n$  особей для дальнейшей генетической обработки, согласно величинам  $P_s(i)$ . Простейший пропорциональный отбор – рулетка (roulette-wheel selection, Goldberg, 1989) – отбирает особей с помощью  $n$  «запусков» рулетки. Колесо рулетки содержит по одному сектору для каждого члена популяции. Размер  $i$ -го сектора пропорционален соответствующей величине  $P_s(i)$ . При таком отборе члены популяции с более высокой приспособленностью (большей вероятностью) будут чаще выбираться, чем особи с низкой приспособленностью.

После отбора  $n$  выбранных особей подвергаются кроссоверу (иногда называемому рекомбинацией) с заданной вероятностью  $P_c$ . Здесь  $n$  строк случайным образом разбиваются на  $n/2$  пары. Для каждой пары с вероятностью  $P_c$  может применяться кроссовер. Соответственно с вероятностью  $(1-P_c)$  кроссовер не происходит, и неизменные особи переходят на стадию мутации. Если кроссовер происходит, полученные потомки заменяют собой родителей и далее переходят к мутации.

Одноточечный кроссовер работает следующим образом. Сначала случайным образом выбирается точка разрыва – участок между соседними битами в строке. Обе родительские структуры разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных родителей «склеиваются» и получаются два генотипа потомков.

Например, предположим, один родитель состоит из 10 нулей, а другой – из 10 единиц. Пусть из 9 возможных точек разрыва выбрана точка 3. Родители и их потомки показаны ниже.

### Кроссовер

Родитель 1	0000000000	000-0000000	⇒	111-0000000	1110000000	Потомок 1
Родитель 2	1111111111	111-1111111	⇒	000-1111111	0001111111	Потомок 2

После того как закончится стадия кроссовера, выполняются операторы мутации. В каждой строке, которая подвергается мутации, каждый бит с некоторой вероятностью  $P_m$  изменяется на противоположный. Популяция, полученная после мутации, записывается поверх старой, и этим цикл одного поко-

ления завершается. Последующие поколения обрабатываются таким же образом: отбор, кроссовер и мутация.

В настоящее время исследователи ГА предлагают много других операторов отбора, кроссовера и мутации. Вот лишь наиболее распространенные из них. Прежде всего, турнирный отбор (Brindle, 1981; Goldberg и Deb, 1991). Турнирный отбор реализует  $n$  турниров, чтобы выбрать  $n$  особей. Каждый турнир построен на выборке  $k$  элементов из популяции и выбора лучшей особи среди них. Наиболее распространен турнирный отбор с  $k = 2$ .

Элитные методы отбора (De Jong, 1975) гарантируют, что при отборе обязательно будут выживать лучшие члены популяции совокупности. Наиболее распространена процедура обязательного сохранения только одной лучшей особи, если она не прошла, как другие, через процесс отбора, кроссовера и мутации. Элитизм может быть внедрен практически в любой стандартный метод отбора.

Двухточечный кроссовер (Cavicchio, 1970; Goldberg, 1989) и равномерный кроссовер (Syswerda, 1989) – вполне достойные альтернативы одноточечному. В двухточечном кроссовере выбираются две точки разрыва, и родительские хромосомы обмениваются сегментом, который находится между двумя этими точками. В равномерном кроссовере каждый бит первого родителя наследуется первым потомком с заданной вероятностью, в противном случае этот бит передается второму потомку и наоборот.

Итак, *генетический алгоритм* – это последовательность управляющих действий и операций, моделирующая эволюционные процессы на основе аналогов механизмов генетического наследования и естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

*Хромосома* – вектор (последовательность) из нулей и единиц, каждая позиция (бит) которого называется геном.

*Особь (индивидуум)* = генетический код – набор хромосом = вариант решения задачи.

*Кроссовер* – операция, при которой две хромосомы обмениваются своими частями.

*Мутация* – случайное изменение одной или нескольких позиций в хромосоме.

Генетические алгоритмы представляют собой скорее подход, чем единые алгоритмы. Они требуют содержательного наполнения для решения каждой конкретной задачи.

На рисунке 10.1 показан один из вариантов структуры генетического алгоритма. Вначале генерируется случайная популяция – несколько особей со случайным набором хромосом (числовых векторов). Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания особей, мутации и смены поколений (отбора).

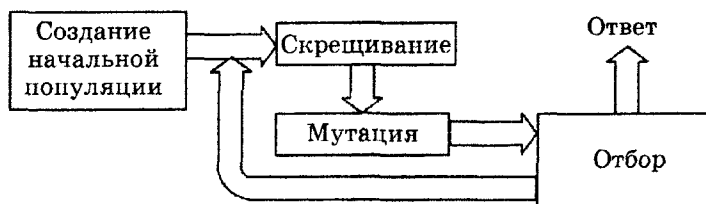


Рис. 10.1. Вариант структуры генетического алгоритма

В течение жизненного цикла популяции в результате нескольких случайных скрещиваний и мутаций к ней добавляется какое-то количество новых вариантов. Далее происходит отбор, в результате которого из старой популяции формируется новая, после чего старая популяция погибает. После отбора к новой популяции опять применяются операции кроссовера и мутации, затем опять происходит отбор и т. д.

Отбор в генетическом алгоритме, как отмечено выше, тесно связан с принципами естественного отбора следующим образом:

- приспособленность особи соответствует значению целевой функции на заданном варианте;
- выживание наиболее приспособленных особей соответствует тому, что популяция следующего поколения вариантов формируется с учетом целевой функции: чем приспособленнее особь, тем больше вероятность ее участия в кроссовере, т. е. в размножении.



Таким образом, модель отбора определяет, как следует строить популяцию следующего поколения. Как правило, вероятность участия особи в скрещивании берется пропорциональной ее приспособленности. Часто используется так называемая *стратегия элитизма*, при которой несколько лучших особей переходят в следующее поколение без изменений, не участвуя в кроссовере и отборе. В любом случае каждое следующее поколение будет в среднем лучше предыдущего. Когда приспособленность особей перестает заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучший из найденных вариантов.

По скорости определения оптимума целевой функции генетические алгоритмы на несколько порядков превосходят случайный поиск. Причина этого заключается в том, что большинство систем имеют довольно независимые подсистемы. Вследствие чего при обмене генетическим материалом от каждого из родителей берутся гены, соответствующие наиболее удачному варианту определенной подсистемы (неудачные варианты постепенно погибают). Генетический алгоритм позволяет накапливать удачные решения для таких систем в целом.

Генетические алгоритмы менее применимы для систем, которые сложно разбить на подсистемы. Кроме того, они могут давать сбой из-за неудачного порядка расположения генов (например, если рядом расположены параметры, относящиеся к различным подсистемам); при этом преимущества обмена генетическим материалом сводятся к нулю. Это замечание несколько сглаживается в системах с диплоидным (двойным) генетическим набором.

### **10.8. Особенности генетических алгоритмов**

Генетические алгоритмы – не единственный способ решения задач оптимизации. Кроме него существуют два основных подхода для решения таких задач, переборный и локально-градиентный, каждый из которых имеет свои достоинства и недостатки.

Сравним стандартные подходы с генетическими алгоритмами на примере задачи коммивояжера (TSP – Traveling Salesman Problem), суть которой состоит

в нахождении кратчайшего замкнутого пути обхода городов, заданных своими координатами.

Уже для 30 городов поиск оптимального пути представляет собой сложную задачу, побудившую развитие новых методов (в том числе нейронных сетей и генетических алгоритмов).<sup>1</sup>

Каждый вариант решения (для 30 городов) – это числовая строка, где на  $j$ -м месте стоит номер  $j$ -го по порядку обхода города. Таким образом, в этой задаче 30 параметров, причем не все комбинации значений допустимы.

Переборный метод наиболее прост для программирования. Для поиска оптимального решения (максимума целевой функции) требуется последовательно вычислить значения целевой функции во всех возможных точках, запоминая максимальное из них. Недостатком метода является большая вычислительная сложность: требуется просчитать длины более  $10^{30}$  вариантов путей, что совершенно нереально. Однако, если перебор всех вариантов за разумное время возможен, то найденное решение является оптимальным.

Второй подход основан на методе градиентного спуска. Вначале выбираются некоторые случайные значения параметров, а затем эти значения постепенно изменяют, добиваясь наибольшей скорости роста целевой функции. При достижении локального максимума такой метод останавливается, поэтому для поиска глобального оптимума требуются дополнительные меры.

Градиентные методы работают быстро, но не гарантируют оптимальности найденного решения. Они идеальны для применения в так называемых унимодальных задачах, где целевая функция имеет единственный локальный максимум (он же – глобальный). Однако задача коммивояжера таковой не является.

Практические задачи, как правило, имеют несколько точек экстремума (многомодальны) и многомерны, т. е. содержат много параметров. Для них не существует универсальных методов, позволяющих достаточно быстро найти абсолютно точные решения. Комбинируя переборный и градиентный методы, можно получить приближенные решения, точность которых будет возрастать с увеличением времени расчета.

Генетический алгоритм представляет собой именно такой комбинированный метод. Механизмы скрещивания и мутации в каком-то смысле реализуют переборную часть метода, а отбор лучших решений – градиентный спуск. На рисунке 10.2 показано, что такое сочетание обеспечивает устойчиво хорошую эффективность генетического поиска для любых типов оптимизационных задач.

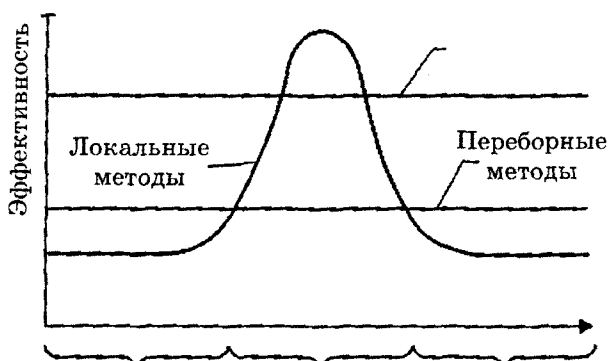


Рис. 10.2. Эффективность генетических алгоритмов

Таким образом, если на некотором множестве задана сложная функция от нескольких переменных, то генетический алгоритм за разумное время находит значение функции, достаточно близкое к оптимальному. Задавая время расчета, можно получить одно из лучших решений, которые реально получить за это время. Следующая таблица характеризует основные различия в решении задачи оптимизации функции с помощью стандартных и генетических алгоритмов.

Два основных различия между генетическими и стандартными алгоритмами оптимизации:

Стандартный алгоритм	Генетический алгоритм
Генерирует единственную точку на каждой итерации. Последовательность таких точек приближается к оптимальному	Генерирует набор (популяцию) точек на каждой итерации. Наборы точек приближаются к оптимальному решению
Определяет следующую точку последовательности детерминистским путем	Определяет следующий набор, используя вероятностный подход

Приведем теперь основную терминологию, используемую при описании генетических алгоритмов и работе с ними (и которая используется в пакете Genetic Algorithm and Direct Search Toolbox системы MatLab).

**Функция приспособленности (fitness function)** – функция, которую требуется оптимизировать. В стандартных оптимизационных алгоритмах обычно используется термин «целевая функция», при этом оптимизационная задача обычно формулируется как задача минимизации данной функции, т. е. как задача

$$\min_x f(x).$$

**Особь (individual)** – любая точка, для которой может быть определена функция приспособленности. Значение данной функции (fitness value) для особи называется **оценкой (score)**. Например, если оптимизируемая функция имеет вид

$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - x_2)^2,$$

то вектор (2, 3, 1), число элементов в котором равно числу переменных оптимизируемой функции, – это особь. Оценка особи – значение функции приспособленности  $f(2, 3, 1) = 195$ .

**Популяции и поколения (Populations and Generations)**. Популяция – это массив особей. Каждая последующая итерация, вырабатывая на каждой последующей итерации ГА, называется поколением.

В пакете Genetic Algorithm and Direct Search Toolbox популяции отображаются в виде матриц, число строк в которых равно числу особей популяции, а число столбцов – числу переменных оптимизируемой функции.

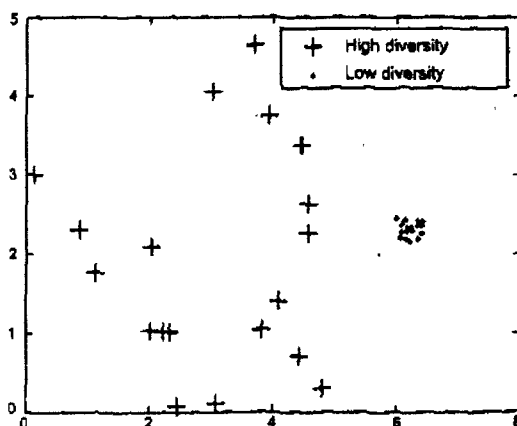


Рис. 10.3. Иллюстрация к понятию разнообразия

**Разнообразие (Diversity)** – понятие, характеризующее среднее расстояние между особями (рис. 10.3). Популяция имеет большое разнообразие, если это

расстояние велико; в противном случае разнообразие мало. Данное понятие играет важную роль для генетических алгоритмов, поскольку отражает размер зоны поиска точки экстремума в пространстве аргументов оптимизируемой функции.

**Наилучшее значение функции приспособленности (*The best fitness value*)** для популяции – это (в задачах минимизации функций) наименьшее значение этой функции для особей данной популяции.

**Родители и потомки (*Parents and Children*)**. Для генерации нового поколения генетический алгоритм выбирает ряд особей текущей популяции, называемых родителями, и использует их для создания особей нового поколения, называемых потомками. Обычно в такой процедуре используются родители с наилучшими (наименьшими) значениями функции приспособленности.

## 11. ВЗАИМОДЕЙСТВИЕ СФЕР МАШИННОГО ОБУЧЕНИЯ

### 11.1. Задачи нейросетевой математики

Хайкин (1994) определил НС как процессор с массивным распараллеливанием операций, обладающий естественной способностью сохранять экспериментальные данные и делать их доступными для дальнейшего использования.

Он похож на мозг в двух направлениях:

- сеть приобретает знания в процессе обучения;
- для хранения информации используются величины интенсивности межнейронных соединений – синаптические веса.

К задачам, успешно решаемым нейронными сетями, относятся:

- распознавание зрительных, слуховых образов: от распознавания текста и целей на экране радара до систем голосового управления;
- ассоциативный поиск информации и создание ассоциативных моделей;
- синтез речи; формирование естественного языка;
- формирование моделей и различных нелинейных и трудно описываемых математических систем, прогнозирование развития этих систем во времени;
- применение на производстве;
- прогнозирование развития циклонов и других природных процессов, прогнозирование изменений курсов валют и других финансовых процессов;
- системы управления и регулирования с предсказанием; управление роботами, другими сложными устройствами;
- разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключаящие логический вывод, особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере.

### 11.2. Алгоритмы обучения сети

Существуют три парадигмы обучения: «с учителем», «без учителя» (самообучение) и смешанная.

В первом случае настройка производится по обучающей выборке, которая состоит из пар (<вход>, <желаемый выход>) – обучающих примеров. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам.

Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае сетью обнаруживается внутренняя структура данных или корреляции между примерами в системе данных, что позволяет распределить их по категориям.

При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

Теория обучения рассматривает несколько аспектов, связанных с обучением по примерам: *емкость*, *сложность примеров* и *вычислительная сложность*. Под *емкостью* понимается, сколько примеров может запомнить сеть и какие функции и границы принятия решений могут быть на ней сформированы. *Сложность* определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать явление *переобучения* сети. Оно заключается в том, что ошибки обучения на примерах обучающей выборки оказываются очень малыми. Когда же сети представляются новые данные, то погрешность существенно возрастает. Это означает, что сеть обучилась минимизировать ошибку на некотором ограниченном обучающем множестве, но не научилась приспосабливаться к новым данным, т. е. решать задачу.

Известны *4 основных типа правил обучения*: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования.

*Правило коррекции по ошибке.* При обучении с учителем для каждого входного примера задан желаемый выход. Реальный выход сети может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала погрешности для модификации весов, обеспечивающей

постепенное уменьшение ошибки. Известны различные модификации этого алгоритма обучения.

**Обучение Больцмана.** Представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах.

**Правило Хебба.** Самым старым обучающим правилом является постулат обучения Хебба. Хебб опирался на следующие нейрофизиологические наблюдения: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом.

**Обучение методом соревнования.** В отличие от обучения Хебба, в котором множество выходных нейронов могут возбуждаться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно, как правило «победитель берет все». Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования позволяет кластеризовать входные данные: подобные примеры группируются сетью в соответствии с корреляциями и представляются одним элементом.

В таблице представлены основные алгоритмы обучения и связанные с ними архитектуры сетей. В последней колонке перечислены задачи, для которых может быть применен каждый алгоритм. Каждый алгоритм обучения ориентирован на сеть определенной архитектуры и предназначен для ограниченного класса задач.



## Основные алгоритмы обучения:

Пара-дигма	Обучающее правило	Архитектура	Алгоритм обучения	Задачи	
С учителем	Коррекция ошибки	Однослойный и многослойный, перцептрон	Алгоритмы обучения перцептрона. Обратное распространение. Adaline и Madaline	Классификация образов. Аппроксимация функций. Предсказание, управление	
	Больцман	Рекуррентная	Алгоритм обучения. Больцмана	Классификация образов	
	Хебб	Многослойная прямого распространения	Линейный дискриминантный анализ	Анализ данных. Классификация образов	
	Соревнование		Соревнование	Векторное квантование	Категоризация внутри класса. Сжатие данных
			Сеть ART	ARTMap	Классификация образов
Без учителя	Коррекция ошибки	Многослойная прямого распространения	Проекция Саммона	Категоризация внутри класса. Анализ данных	
	Хебб	Прямого распространения или соревнования	Анализ главных компонентов	Анализ данных. Сжатие данных	
		Сеть Хопфилда	Обучение ассоциативной памяти	Ассоциативная память	
	Соревнование		Соревнование	Векторное квантование	Категоризация. Сжатие данных
			SOM Кохонена	SOM Кохонена	Категоризация. Анализ данных
			Сети ART	ART1, ART2	Категоризация
Смешанная	Коррекция ошибки и соревнование	Сеть RBF	Алгоритм обучения RBF	Классификация образов. Аппроксимация функций. Предсказание, управление	

Кроме представленных, известны некоторые другие алгоритмы: *Adaline* и *Madaline*, линейный дискриминантный анализ, проекции Саммона, анализ главных компонент.

### 11.3. Области применения нейронных сетей

Типичной для нейросетевого подхода можно считать задачу распознавания букв в рукописном тексте. Пусть дано растровое черно-белое изображение буквы размером  $30 \times 30$  пикселей. Оно преобразуется во входной вектор из  $30 \times 30 = 900$  двоичных символов. Строится нейросеть с 900 входами и 33 выходами, которые помечены буквами. В результате обучения достигается такое

состояние, что если на входе сети, например буква «А», то максимальное значение выходного сигнала наблюдается на выходе «А».

Многие задачи, для решения которых используются нейронные сети, могут рассматриваться как частные случаи следующих основных проблем:

- построение функции по конечному набору значений;
- оптимизация;
- построение отношений на множестве объектов;
- распределенный поиск информации и ассоциативная память;
- фильтрация;
- сжатие информации;
- идентификация динамических систем и управление ими;
- нейросетевая реализация классических задач и алгоритмов вычислительной математики: решение систем линейных уравнений, решение задач математической физики сеточными методами и др.

#### **11.4. Взаимодействие различных областей**

Рассмотрим особенности взаимодействия областей нейронных сетей, эволюционного программирования и нечеткой логики. Объединение возможностей нейронных сетей и нечеткой логики является наиболее перспективным подходом к организации систем интеллектуального анализа экономических данных. Системы нечеткой логики компенсируют две основные «непрозрачности» НС в представлении знаний и объяснений результатов работы интеллектуальной системы, т. е. НЛ наилучшим образом дополняет нейронные сети. Нечеткая логика позволяет формализовать качественную информацию, полученную от экспертов-экономистов для конкретной сферы применения, и представить совокупность полученных знаний в виде системы нечетких правил логического вывода, позволяющих анализировать заключения, полученные в процессе работы гибридной интеллектуальной системы. Нейронные сети дают возможность отобразить алгоритмы нечеткого логического вывода в структуре НС, вводя в информационное поле нейронной сети информацию, полученную от экспертов-

экономистов. Сформированная подобным образом база знаний автоматически корректируется в процессе обучения нейро-нечеткой сети, исходя из реальных значений анализируемых экономических показателей, а результаты коррекции могут быть подвергнуты последующему анализу. Важной особенностью нейро-нечетких сетей является способность автоматически генерировать систему нечетких правил, извлекая скрытые закономерности изданных обучающей выборки. Под названием адаптивной нейро-нечеткой системы вывода – ANFIS (Adaptive Neuro-Fuzzy Inference System) известна специализированная пейросетевая структура, характеризующаяся хорошей сходимостью и ориентированная на извлечение знаний в виде системы нечетких правил из данных обучающей выборки.

#### **11.5. ANFIS: функциональный эквивалент нечеткой модели**

Таким образом, проведенный анализ показывает, что знания квалифицированных экономистов для конкретной предметной области, представленные в форме нечетких правил логического вывода, могут быть прозрачным способом отражены в структуре нейро-нечеткой сети. Обучение нечеткой НС позволяет не только настроить веса связей (т. е. откорректировать достоверность нечетких правил логического вывода), но и устранить противоречивость системы нечетких правил в целом. В случае отсутствия исходной информации по данной предметной области, но при достаточном объеме обучающей выборки нейро-нечеткая сеть автоматически преобразует скрытые в анализируемых показателях закономерности в базу знаний в виде системы правил нечеткого логического вывода. Решение задач управления и принятия решений корпоративного уровня сопровождается оптимизацией сайта хозяйствующего субъекта. Причем оптимизация информационной структуры корпоративного сайта в соответствии с интересами его посетителей базируется на придании сайту адаптивных свойств, что требует привлечения современных интеллектуальных средств.

Анализ перспективных интеллектуальных средств подтвердил, что при решении задач управления и принятия решений, для которых характерно наличие неполной и недостаточно достоверной информации, хорошо зарекомендо-

вали себя системы *интеллектуального анализа данных*. Нейронные сети, системы нечеткой логики являются обязательным инструментом интеллектуального поиска и извлечения знаний, так как обладают способностью выявления значимых признаков и скрытых закономерностей в анализируемых экономических показателях.

### 11.6. Нейронные сети и эволюционное моделирование

Обучение нейронных сетей – сложная задача по ряду причин. Нередко процесс поиска адекватной нейросетевой модели заканчивается с нулевым результатом и очень большую роль играет опыт разработчика нейросетевых моделей. Для получения нейросетевой модели, решающей задачу с заданным показателем качества, обычно необходимо пройти следующие шаги:

- необходимо подготовить данные;
- определиться с типом сети;
- определить входы и выходы;
- решить задачу о первоначальной структуре сети – слои и нейроны в них;
- обучить сеть, т. е. подобрать коэффициенты связей между нейронами;
- проверить обученную сеть на валидационной выборке;
- в итоге проверить в реальной работе.

При этом все шаги тесно связаны между собой и некачественная проработка по одному из них ведет, в конечном счете, к длительному обучению сети или вообще к получению неправильно работающей нейросети.

Существует большое количество методов и алгоритмов предварительной подготовки данных, расчета структуры сети и модифицированных методов обучения, но все они в значительной мере опираются на опыт разработчика. Одним из наиболее универсальных способов автоматического получения нейросетевых моделей является использование генетических алгоритмов. Согласно *COGANN (Combinations of Genetic Algorithms and Neural Networks)*, объединение нейронных сетей и генетических алгоритмов может быть как вспомогательным, так и равноправным. Во вспомогательном подходе один ме-

тод идет вслед за другим, а в равноправном оба метода используются синхронно. В качестве вспомогательной парадигмы выделяют следующие виды объединения:

- генетические и нейросетевые алгоритмы применяются одновременно для одной задачи (например, для задач классификации);
- анализ нейронных сетей с помощью генетических алгоритмов;
- подбор параметров нейронных сетей с помощью генетических алгоритмов;
- подбор правил обучения нейронных сетей с помощью генетических алгоритмов;
- формирование исходной популяции для генетических алгоритмов с помощью нейронных сетей.

В случае равноправного объединения выделяют следующие виды совместного использования генетических алгоритмов и нейронных сетей:

- генетические алгоритмы для обучения нейронных сетей (эволюционное обучение нейронной сети);
- выбор топологии нейронной сети с помощью генетического алгоритма (эволюционный подбор топологии сети);
- нейронные сети для решения оптимизационных задач с подбором весов через генетический алгоритм;
- реализация генетического алгоритма с помощью нейронной сети.

Следует отметить, что иногда в связке «генетический алгоритм + нейронная сеть» применяются очень сложные архитектуры нейронных сетей, в частности *ART-1* и *ART-2*. Управляемыми параметрами в генетических алгоритмах являются: длина хромосомы; наполнение хромосомы (локусы и аллели); параметры оператора кроссовера; параметры оператора мутации; параметры оператора инверсии; параметры выбора лучших особей; критерий остановки генерации особей и популяции; параметры генерации начальной и последующих популяций и т. д.

## 11.7. Искусственные нейронные сети и экспертные системы

В последние годы над искусственными нейронными сетями доминировали логические и символично-операционные дисциплины. Например, широко пропагандировались *экспертные системы*, у которых имеется много заметных успехов, так же, как и неудач. Кое-кто говорит, что искусственные нейронные сети заменят собой современный искусственный интеллект, но многое свидетельствует о том, что они будут существовать, объединяясь в системах, где каждый подход используется для решения тех задач, с которыми он лучше справляется.

Эта точка зрения подкрепляется тем, как люди функционируют в нашем мире. Распознавание образов отвечает за активность, требующую быстрой реакции. Так как действия совершаются быстро и бессознательно, то этот способ функционирования важен для выживания во враждебном окружении. Вообразите только, что было бы, если бы наши предки вынуждены были обдумывать свою реакцию на прыгнувшего хищника?

Когда наша система распознавания образов не в состоянии дать адекватную интерпретацию, вопрос передается в высшие отделы мозга. Они могут запросить добавочную информацию и займут больше времени, но качество полученных в результате решений может быть выше.

Можно представить себе искусственную систему, подражающую такому разделению труда. Искусственная нейронная сеть реагировала бы в большинстве случаев подходящим образом на внешнюю среду. Так как такие сети способны указывать доверительный уровень каждого решения, то сеть «знает, что она не знает» и передает данный случай для разрешения экспертной системе. Решения, принимаемые на этом более высоком уровне, были бы конкретными и логичными, но они могут нуждаться в сборе дополнительных фактов для получения окончательного заключения. Комбинация двух систем была бы более мощной, чем каждая из систем в отдельности, следуя при этом высокоэффективной модели, даваемой биологической эволюцией.

## 11.8. Соображения надежности

Прежде чем искусственные нейронные сети можно будет использовать там, где поставлены на карту человеческая жизнь или ценное имущество, должны быть решены вопросы, относящиеся к их надежности.

Подобно людям, структуру мозга которых они копируют, искусственные нейронные сети сохраняют в определенной мере непредсказуемость. Единственный способ точно знать выход состоит в испытании всех возможных входных сигналов. В большой сети такая полная проверка практически неосуществима и должны использоваться статистические методы для оценки функционирования. В некоторых случаях это недопустимо. Например, что является допустимым уровнем ошибок для сети, управляющей системой космической обороны? Большинство людей скажет, любая ошибка недопустима, так как ведет к огромному числу жертв и разрушений. Это отношение не меняется от того обстоятельства, что человек в подобной ситуации также может допускать ошибки.

Проблема возникает из-за допущения полной безошибочности компьютеров. Так как искусственные нейронные сети иногда будут совершать ошибки даже при правильном функционировании, то, как ощущается многими, это ведет к ненадежности – качеству, которое мы считаем недопустимым для наших машин.

Сходная трудность заключается в неспособности традиционных искусственных нейронных сетей «объяснить», как они решают задачу. Внутреннее представление, получающееся в результате обучения, часто настолько сложно, что его невозможно проанализировать, за исключением самых простых случаев. Это напоминает нашу неспособность объяснить, как мы узнаем человека, несмотря на различие в расстоянии, угле, освещении и на прошедшие годы. Экспертная система может проследить процесс своих рассуждений в обратном порядке, так что человек может проверить ее на разумность. Сообщалось о встраивании этой способности в искусственные нейронные сети, что может существенно повлиять на приемлемость этих систем.

## 12. КОГНИТИВНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ ПРОБЛЕМНЫХ СИТУАЦИЙ

### 12.1. Ситуационный анализ на основе когнитивных карт

Сложности анализа процессов и принятия решений при моделировании тактических ситуаций обусловлены рядом особенностей:

- **многоаспектностью** происходящих процессов и их **взаимосвязанностью**, что делает невозможным вычленение и детальное исследование отдельных явлений – все происходящие в них явления должны рассматриваться в совокупности;
- **отсутствием достаточной количественной информации** о динамике процессов, что вынуждает переходить к их качественному анализу;
- **изменчивостью** характера процессов во времени и т. д.

Такие системы называют **слабоструктурированными**. Под текущей ситуацией понимается состояние слабоструктурированной системы в рассматриваемый момент времени. Число факторов в ситуации может измеряться десятками, при этом все они вплетены в паутину меняющихся во времени причин и следствий. Увидеть и осознать логику развития событий на таком многофакторном поле крайне трудно. На многие вопросы здесь можно успешно ответить, только на основе использования специализированных компьютерных средств **познавательного (когнитивного) моделирования ситуаций**.

Методология когнитивного моделирования, предназначенная для анализа и принятия решений в плохо определенных ситуациях, была предложена американским исследователем Р. Аксельродом [Axelrod R. The Structure of Decision: Cognitive Maps of Political Elites. Princeton. University Press, 1976]. Работы по развитию когнитивного подхода и его применению для анализа и управления слабоструктурированными системами проводятся в настоящее время в Институте проблем управления РАН. Согласно опубликованным сборникам статей, результаты этих работ успешно применяются для решения целого ряда прикладных задач. В частности, по заказу Администрации Президента РФ, Правительства РФ и Правительства города Москвы в ИПУ РАН был осуществ-



лён ряд социально-экономических исследований с применением когнитивной технологии. Выработанные рекомендации с успехом применяются соответствующими министерствами и ведомствами. С 2001 г. под эгидой ИПУ РАН регулярно проводятся международные конференции «Когнитивный анализ и управление развитием ситуаций (CASC)».

Когнитивное моделирование способствует лучшему пониманию проблемной ситуации, выявлению противоречий и качественному анализу системы. Цель моделирования состоит в формировании и уточнении гипотезы о функционировании исследуемого объекта, рассматриваемого как сложная система, состоящая из отдельных, но все же связанных между собой элементов и подсистем. Для того чтобы понять и проанализировать поведение сложной системы, строят структурную схему причинно-следственных связей элементов системы. Анализ этих связей необходим для реализации различных управлений процессами в системе.

Исходным понятием в когнитивном моделировании сложных ситуаций является понятие **когнитивной карты ситуации**. Когнитивная карта ситуации представляет собой ориентированный взвешенный граф, в котором:

- вершины взаимно однозначно соответствуют базисным факторам ситуации, в терминах которых описываются процессы в ситуации. Множество первоначально отобранных базисных факторов может быть верифицировано с помощью технологии Data Mining, позволяющей отбросить избыточные факторы, слабо связанные с ядром базисных факторов;
- определяются непосредственные взаимосвязи между факторами путем рассмотрения причинно-следственных цепочек «если..., то...», описывающих распространение влияний одного фактора на другие факторы. Считается, что факторы, входящие в посылку «если...» цепочки, влияют на факторы следствия «то...» этой цепочки, причем это влияние может быть либо усиливающим (положительным), либо тормозящим (отрицательным), либо переменного знака в зависимости от возможных дополнительных условий.

Когнитивная карта отображает лишь факт наличия влияний факторов друг на друга. В ней не отражается ни детальный характер этих влияний, ни динамика изменения влияний в зависимости от изменения ситуации, ни временные изменения самих факторов. Учет всех этих обстоятельств требует перехода на следующий уровень структуризации информации, отображенной в когнитивной карте, т. е. к **когнитивной модели**. На этом уровне каждая связь между факторами когнитивной карты раскрывается до соответствующего **уравнения**, которое может содержать как количественные (измеряемые) переменные, так и качественные (не измеряемые) переменные. При этом количественные переменные входят естественным образом в виде их численных значений. Каждой же качественной переменной ставится в соответствие совокупность лингвистических переменных, отображающих различные состояния этой качественной переменной, а каждой лингвистической переменной соответствует определенный числовой эквивалент в шкале  $[-1,1]$ . По мере накопления знаний о процессах, происходящих в исследуемой ситуации, становится возможным более детально раскрывать характер связей между факторами. Здесь существенную помощь может оказать использование процедур Data Mining. Формально когнитивная модель ситуации может быть, как и когнитивная карта, представлена графом, однако каждая дуга в этом графе представляет уже некую функциональную зависимость между соответствующими базисными факторами, т. е. когнитивная модель ситуации представляется **функциональным графом**.

## 12.2. Обеспечение целенаправленного поведения

При анализе конкретной ситуации пользователь обычно знает или предполагает, какие изменения базисных факторов являются для него желательными. Факторы, представляющие наибольший интерес для пользователя, назовем **целевыми**. Это – выходные факторы когнитивной модели. Задача выработки решений по управлению процессами в ситуации состоит в том, чтобы обеспечить желательные изменения целевых факторов, это – **цель управления**. Цель считается **корректно заданной**, если желательные изменения одних целевых факторов не приводят к нежелательным изменениям других целевых факторов.

В исходном множестве базисных факторов выделяется совокупность так называемых **управляющих факторов** (входных факторов когнитивной модели), через которые подаются управляющие воздействия в модель. Управляющее воздействие считается **согласованным с целью**, если оно не вызывает нежелательных изменений ни в одном из целевых факторов. При корректно заданной цели управления и при наличии управляющих воздействий, согласованных с этой целью, решение задачи управления не вызывает особых трудностей (даже при нелинейной когнитивной модели со знакопостоянными влияниями факторов друг на друга). В общем же случае нахождение условий для обеспечения целенаправленного поведения в ситуации является весьма непростой задачей, требующей специального рассмотрения.

### 12.3. Методика когнитивного анализа сложных ситуаций

Этапы когнитивного анализа сложной ситуации (погружение в проблему, идентификация проблемы):

1. Формулировка задачи и цели исследования.
2. Изучение процесса с позиций поставленной цели.
3. Сбор, систематизация, анализ существующей статистической и качественной информации по проблеме.
4. Выделение основных характеристических признаков изучаемого процесса и взаимосвязей, определение действия основных объективных развития исследуемой ситуации – это позволит выделить объективные зависимости, тенденции в процессах.
5. Определение присущих исследуемой ситуации требований, условий и ограничений.
6. Выделение основных субъектов, связанных с ситуацией, определение их субъективных интересов в развитии данной ситуации – это позволит определить возможные изменения в объективном развитии ситуации, выделить факторы, на которые реально могут влиять субъекты ситуации.
7. Определение путей, механизмов действия, реализации интересов основных субъектов – это позволит в дальнейшем определить стратегии пове-

дения и предотвращения нежелательных последствий развития ситуации.

#### 12.4. Построение когнитивной модели

Рассмотрим процесс построения когнитивной (графовой) модели проблемной ситуации. Итак:

1. Выделение факторов, характеризующих проблемную ситуацию:
  - 1.1. Выделение **базисных (основных) факторов**, описывающих суть проблемы. Выделение в совокупности базисных факторов целевых факторов.
  - 1.2. Определение факторов, **влияющих на целевые факторы**. Эти факторы в модели будут являться потенциально возможными рычагами воздействия на ситуацию.
  - 1.3. Определение факторов-индикаторов, отражающих и объясняющих развитие процессов в проблемной ситуации и их влияние на различные сферы.
2. Группировка факторов по блокам. Объединяются в один блок факторы, характеризующие данную сферу проблемы и определяющие процессы в этой сфере. Здесь возможны варианты в зависимости от специфики проблемы, целей анализа, количества субъектов ситуации и т. д.:
  - 2.1. Выделение в блоке группы интегральных показателей (факторов), по изменению которых можно судить об общих тенденциях в данной сфере.
  - 2.2. Выделение в блоке показателей (факторов), характеризующих тенденции и процессы в данной сфере более детально.
3. Определение связей между факторами:
  - 3.1. Определение связей и взаимосвязей между блоками факторов. Это позволит определить основные направления влияния факторов разных блоков друг на друга.
  - 3.2. Определение **непосредственных связей факторов внутри блока**:
    - 3.2.1. Определение направления влияний и взаимовлияний между факторами.

3.2.2. Определение позитивности влияния (положительное, отрицательное,  $+/-$ ).

3.2.3. Определение силы влияния и взаимовлияния факторов (слабо, сильно).

3.3. Определение связей между факторами различных блоков.

4. Проверка адекватности модели, т. е. сопоставление полученных результатов с характеристиками системы, которые при тех же исходных условиях были в прошлом. Если результаты сравнения – неудовлетворительны, то модель корректируется.

Существуют две главные проблемы построения когнитивной модели. На этапе построения ориентированного графа трудности вызывает выявление факторов (элементов системы) и их ранжирование (выделение базисных и второстепенных). На этапе построения функционального графа трудности связаны с выявлением степени взаимовлияния факторов (определением весов дуг графа).

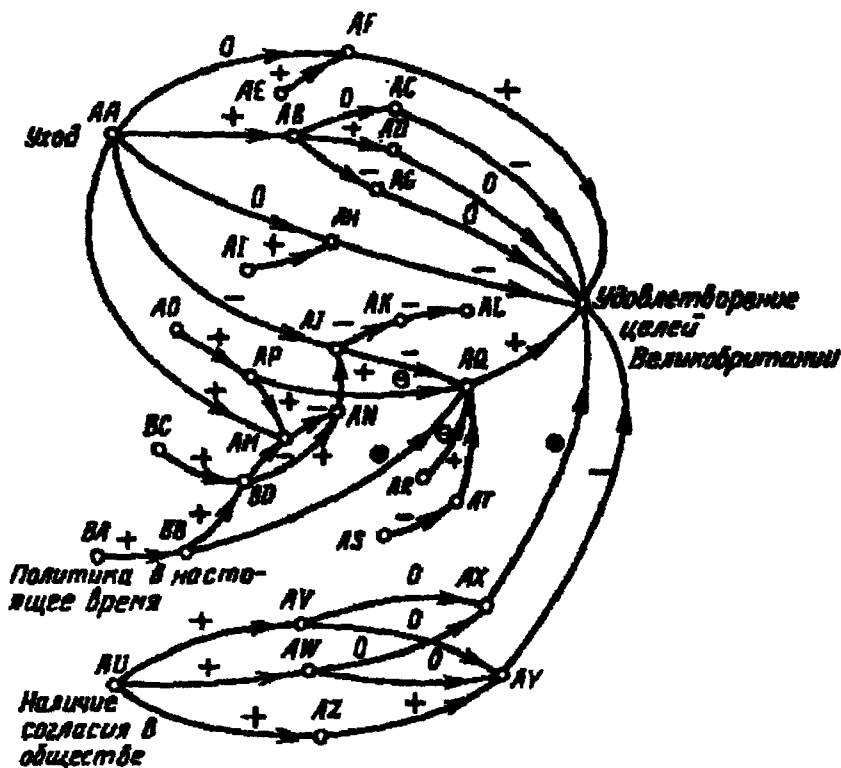


Рис. 12.1. Когнитивная карта эксперта по британской политике в Персии 1918 г.

## 12.5. Моделирование

Моделирование – это средство выявления закономерностей предупреждения и предотвращения негативных тенденций, получения теоретических и практических знаний о проблеме и формулирования на этой основе практических выводов. Моделирование – циклический процесс. Знания об исследуемой проблеме расширяются и уточняются, а исходная модель постоянно совершенствуется. Моделирование основано на сценарном подходе.

## 12.6. Внешняя среда

Для эффективного управления, прогнозирования и планирования необходим анализ внешней среды, в которой функционируют объекты управления. Внешняя среда обычно определяется как совокупность экономических, социальных и политических факторов и субъектов, оказывающих непосредственное или косвенное воздействие на возможность и способность субъекта достигать поставленных целей развития.

Для ориентации во внешней среде и для её анализа необходимо чётко представлять её свойства. Выделяют следующие основные характеристики внешней среды:

- **Сложность** – здесь подразумевается число и разнообразие факторов, на которые субъект должен реагировать.
- **Взаимосвязь факторов**, т. е. сила, с которой изменение одного фактора воздействует на изменение других факторов.
- **Подвижность** – скорость, с которой происходят изменения во внешней среде.

Таким образом, внешняя среда рассматривается как система или совокупность систем. В рамках этого подхода принято представлять любые объекты в виде структурированной системы, выделять элементы системы, взаимосвязи между ними и динамику развития элементов, взаимосвязей и всей системы в целом. Специфика внешней среды объектов управления заключается в том, что эта среда подвержена воздействию человеческого фактора. Иначе говоря, она включает в себя субъекты, наделённые автономной волей, интересами и субъ-

ективными представлениями. Это означает, что эта среда далеко не всегда подчиняется линейным законам, однозначно описывающим связь причин и следствий. Отсюда вытекают два базовых параметра внешней среды, в которой действует человеческий фактор, – нестабильность и слабоструктурированность.

### **12.7. Нестабильность внешней среды**

Нестабильность внешней среды часто отождествляется исследователями с непредсказуемостью. Эта непредсказуемость порождается многофакторностью, изменчивостью факторов, темпов и направления развития среды. Чем выше нестабильность внешней среды, тем сложнее выработать адекватные стратегические решения. Поэтому существует объективная потребность в оценке степени нестабильности среды, а также в выработке подходов к её анализу.

По мнению И. Акоффа, выбор стратегии управления и анализа ситуации зависит от уровня нестабильности внешней среды. При умеренной нестабильности применяется обычное управление на основе экстраполяции знаний о прошлом среды. При среднем уровне нестабильности управление осуществляется на основе прогноза изменений в среде. При высоком уровне нестабильности используется управление на основе гибких экспертных решений.

### **12.8. Слабоструктурированность внешней среды**

Среда, в которой вынуждены работать субъекты управления, характеризуется не только как нестабильная, но и как слабоструктурированная. Две эти характеристики прочно взаимосвязаны, но различны. Впрочем, иногда эти термины употребляются как синонимы. Однако следует заметить, что термин «нестабильность» предполагает невозможность или трудность предсказать развитие системы, а слабоструктурированность – невозможность её формализовать. В конечном итоге, характеристики «нестабильность» и «слабоструктурированность» отражают разные аспекты одного и того же явления, поскольку мы традиционно воспринимаем систему, которую не можем формализовать и таким образом достаточно точно предсказать её развитие (т. е. слабоструктурированную систему), как нестабильную, склонную к хаосу.

Итак, в отличие от технических систем, тактические и прочие аналогичные системы характеризуются отсутствием детального количественного описания происходящих в них процессов – информация здесь имеет качественный характер. Поэтому для слабоструктурированных систем невозможно создание традиционных формальных количественных моделей. Для систем подобного типа характерны неопределенность, описание на качественном уровне, неоднозначность оценки последствий тех или иных решений.

Таким образом, анализ нестабильной внешней среды (слабоструктурированных систем) сопряжён со многими трудностями. При их решении нужна интуиция эксперта, его опыт, ассоциативность мышления, догадки.

С подобным анализом позволяют справиться компьютерные средства познавательного (когнитивного) моделирования ситуаций. Познавательное моделирование призвано помочь эксперту отразить на более глубоком уровне и упорядочить свои знания, а также формализовать свои представления о ситуации в той мере, в какой это возможно.

## **12.9. Общее понятие когнитивного анализа**

Когнитивный анализ иногда именуется исследователями «когнитивной структуризацией».

Когнитивный анализ рассматривается как один из наиболее мощных инструментов исследования нестабильной и слабоструктурированной среды. Он способствует лучшему пониманию существующих в среде проблем, выявлению противоречий и качественному анализу протекающих процессов. Суть когнитивного (познавательного) моделирования – ключевого момента когнитивного анализа – состоит в том, чтобы сложнейшие проблемы и тенденции развития системы отразить в упрощенном виде в модели, исследовать возможные сценарии возникновения кризисных ситуаций, найти пути и условия их разрешения в модельной ситуации. Использование когнитивных моделей качественно повышает обоснованность принятия управленческих решений в сложной и быстроизменяющейся обстановке, избавляет эксперта от «интуитивного блуждания», экономит время на осмысление и интерпретацию происходящих



в системе событий. В основе технологии когнитивного анализа и моделирования (рис. 12.2) лежит когнитивная (познавательльно-целевая) структуризация знаний об объекте и внешней для него среды.

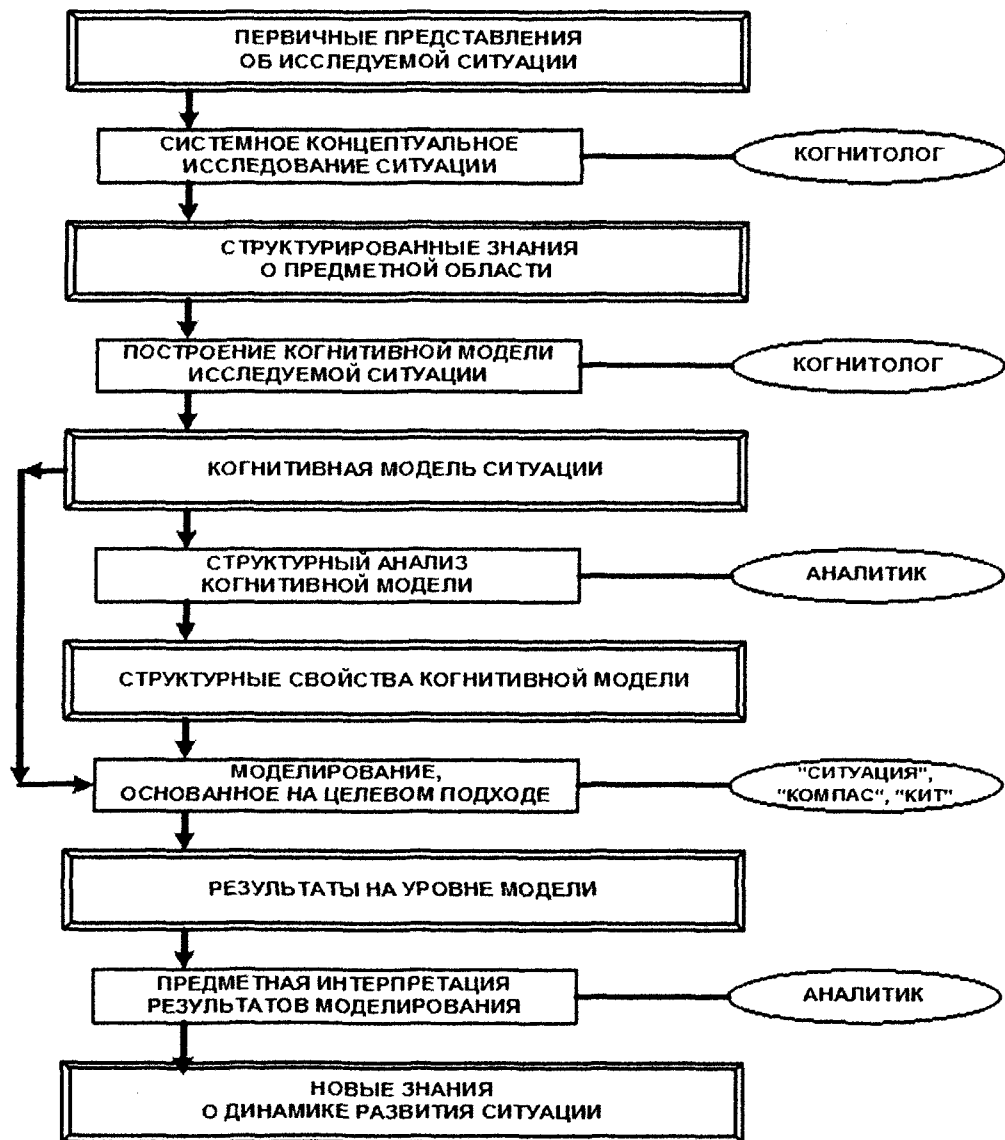


Рис. 12.2. Технология когнитивного анализа и моделирования

Для объяснения принципов использования информационных познавательных (когнитивных) технологий для совершенствования управления используют метафору корабля в бушующем океане – так называемую модель «фрегат-

океан». В таких моделях информационная модель объекта управления («фрегат») взаимодействует с моделью внешней среды («океан»). Цель такого моделирования – дать рекомендации «фрегату» как пересечь «океан» с наименьшими «усилиями». Интерес представляют способы достижения цели с учетом попутных «ветров» и «течений». Сущность когнитивного подхода заключается в том, чтобы помочь эксперту отразить ситуацию и разработать наиболее эффективную стратегию управления, основываясь не столько на своей интуиции, сколько на упорядоченном и верифицированном (насколько это возможно) знании о сложной системе.

### 12.10. Механизмы реализации частных задач

В когнитивную модель входят когнитивная карта (ориентированный граф) и веса дуг графа (оценка взаимовлияния или влияния факторов). При определении весов дуг ориентированный граф превращается в функциональный.

В рамках когнитивного подхода довольно часто термины «когнитивная карта» и «ориентированный граф» употребляются как равнозначные, хотя понятие ориентированного графа шире, а термин «когнитивная карта» указывает лишь на одно из его применений.

Когнитивная карта состоит из факторов (элементов системы) и связей между ними. Для того чтобы понять и проанализировать поведение сложной системы, строят структурную схему причинно-следственных связей элементов системы (факторов ситуации). Два элемента системы  $A$  и  $B$  изображаются на схеме в виде отдельных точек (вершин), соединённых ориентированной дугой, если элемент  $A$  связан с элементом  $B$  причинно-следственной связью:  $A \Rightarrow B$ , где  $A$  – причина,  $B$  – следствие.

Факторы могут влиять друг на друга, причем такое влияние может быть положительным, когда увеличение (уменьшение) одного фактора приводит к увеличению (уменьшению) другого фактора, и отрицательным, когда увеличение (уменьшение) одного фактора приводит к уменьшению (увеличению) другого фактора. Влияние может иметь и переменный знак в зависимости от возможных дополнительных условий. Подобные схемы представления причинно-

следственных связей широко используются для анализа сложных систем, например, в экономике и социологии.

Когнитивная карта отображает лишь факт наличия влияний факторов друг на друга. В ней не отражается ни детальный характер этих влияний, ни динамика изменения влияний в зависимости от изменения ситуации, ни временные изменения самих факторов. Учет всех этих обстоятельств требует перехода на следующий уровень структуризации информации, т. е. к когнитивной модели.

На этом уровне каждая связь между факторами когнитивной карты раскрывается соответствующими зависимостями, каждая из которых может содержать как количественные (измеряемые) переменные, так и качественные (не измеряемые) переменные. При этом количественные переменные представляются естественным образом в виде их численных значений. Каждой же качественной переменной ставится в соответствие совокупность лингвистических переменных, отображающих различные состояния этой качественной, а каждой лингвистической переменной соответствует определенный числовой эквивалент в шкале  $[0,1]$ . По мере накопления знаний о процессах, происходящих в исследуемой ситуации, становится возможным более детально раскрывать характер связей между факторами.

Формально когнитивная модель ситуации может, как и когнитивная карта, быть представлена графом, однако каждая дуга в этом графе представляет уже некую функциональную зависимость между соответствующими факторами, т. е. когнитивная модель ситуации представляется функциональным графом.

### **12.11. Виды факторов**

*Для структуризации ситуации (системы) подразделяют факторы (элементы)* на различные группы, каждая из которых обладает определённой спецификой, функциональной ролью в моделировании. В зависимости от специфики анализируемой ситуации типология факторов (элементов) может быть различна. Во-первых, среди всех обнаруженных факторов выделяются базовые (воздействующие на ситуацию существенным образом, описывающие суть

проблемы) и «избыточные» (малозначащие) факторы, «слабо связанные» с «ядром» базисных факторов.

При анализе конкретной ситуации эксперт обычно знает или предполагает, какие изменения базисных факторов являются для него желательными. Факторы, представляющие наибольший интерес для эксперта, называются целевыми. В исходном множестве базисных факторов выделяется совокупность так называемых управляющих факторов – «входных» факторов когнитивной модели, через которые подаются управляющие воздействия в модель. Управляющее воздействие считается согласованным с целью, если оно не вызывает нежелательных изменений ни в каком из целевых факторов. Управляющие факторы в модели будут являться потенциально возможными рычагами воздействия на ситуацию. Влияние управляющих факторов суммируется в понятии «вектор управляющих воздействий» – совокупность факторов, на каждый из которых подается управляющий импульс заданной величины.

Факторы ситуации (или элементы системы) могут также подразделяться на внутренние (принадлежащие самому объекту управления и находящиеся под более или менее полным контролем руководства) и внешние (отражающие воздействие на ситуацию или систему внешних сил, которые могут не контролироваться или лишь косвенно контролироваться субъектом управления).

Внешние факторы обычно разделяются на предсказуемые, возникновение и поведение которых можно предвидеть на основе анализа имеющейся информации, и на непредсказуемые, о поведении которых эксперт узнает лишь после их возникновения. Иногда исследователи выделяют так называемые факторы-индикаторы, отражающие и объясняющие развитие процессов в проблемной ситуации (системе, среде). Для подобных целей используется также понятие интегральных показателей (факторов), по изменению которых можно судить об общих тенденциях в данной сфере.

Факторы характеризуются также тенденцией изменения своих значений. Различают тенденции роста и снижения. В случае отсутствия изменения фактора говорят об отсутствии тенденции или о нулевой тенденции. Наконец, следу-

ет отметить, что возможно выявление причинных факторов и факторов-следствий, кратковременных и долгосрочных *факторов*.

## **12.12. Выявление факторов (элементов системы)**

Можно констатировать, что исследователями не разработан чёткий алгоритм выявления элементов исследуемых систем. Предполагается, что изучаемые факторы ситуации уже известны эксперту, проводящему когнитивный анализ.

Обычно при рассмотрении крупных (например, макроэкономических) систем применяется так называемый PEST-анализ (Policy – политика, Economy – экономика, Society – общество, Technology – технология), предполагающий выделение 4-х основных групп факторов, посредством которых анализируется политический, экономический, социокультурный и технологический аспекты среды. Подобный подход хорошо известен во всех социально-экономических науках.

PEST-анализ – это инструмент исторически сложившегося четырехэлементного стратегического анализа внешней среды. При этом для каждого конкретного сложного объекта существует свой особый набор ключевых факторов, которые непосредственно и наиболее существенным образом влияет на объект. Анализ каждого из выделенных аспектов проводится системно, так как в жизни все эти аспекты между собой тесно взаимосвязаны.

Кроме того, предполагается, что эксперт может судить о номенклатуре факторов, сообразуясь со своими субъективными представлениями. Так, «Фундаментальный» анализ финансовых ситуаций, близкий по некоторым параметрам к когнитивному анализу, базируется на наборе базисных факторов (финансово-экономических показателей) как макроэкономических, так и более низкого порядка, как долгосрочных, так и краткосрочных. Эти факторы, в соответствии с «фундаментальным» подходом, определяются на основе здравого смысла.

Таким образом, единственный вывод, который можно сделать относительно процесса выявления факторов, заключается в том, что аналитик, преследуя эту цель, должен руководствоваться уже готовыми знаниями наук, занима-

ющихся конкретным изучением разнообразных систем, а также своим опытом и интуицией.

### 12.13. Два подхода к выявлению связей между факторами

Для отображения характера взаимодействия факторов используются позитивный и нормативный подходы. Позитивный подход основывается на учете объективного характера взаимодействия факторов и позволяет провести дуги, приписать им знаки (+ / -) и точные веса, т. е. отразить характер этого взаимодействия. Этот подход применим в том случае, если взаимосвязь факторов может быть подвергнута формализации и выражена математическими формулами, устанавливающими точные количественные взаимосвязи.

Однако далеко не все реальные системы и их подсистемы описываются теми или иными математическими формулами. Можно сказать, что формализованы лишь некоторые частные случаи взаимодействия факторов. Более того, чем сложнее система, тем менее вероятность её исчерпывающего описания посредством традиционных математических моделей. Это связано прежде всего с фундаментальными свойствами нестабильных, слабоструктурированных систем, описанными выше. Поэтому позитивный подход дополняется нормативным.

Нормативный подход основывается на субъективном, оценочном восприятии взаимодействия факторов, и его использование также позволяет приписать дугам веса, т. е. отразить силу (интенсивность) взаимодействия факторов. Выяснение влияний факторов друг на друга и оценки этих влияний опираются на «прикидки» эксперта и выражаются в количественном виде с помощью шкалы  $[-1, 1]$  или лингвистическими переменными типа «сильно», «слабо», «умеренно». Иначе говоря, при нормативном подходе перед экспертом стоит задача интуитивно определить силу взаимовлияния факторов, основываясь на своих знаниях о качественной взаимосвязи.

Кроме того, как уже упоминалось, в ряде случаев трудность представляет даже определение отрицательного или положительного характера влияния факторов, а не только силы влияния. При осуществлении этой задачи, очевидно, возможно использование двух означенных выше подходов.

## 12.14. Проблема определения силы воздействия факторов

Итак, важнейшая проблема когнитивного моделирования – это назначение весов дуг графа, т. е. количественная оценка взаимовлияния или влияния факторов. Дело в том, что когнитивный подход применяется при исследовании нестабильной, слабоструктурированной среды с такими характеристиками, как изменчивость, трудноформализуемость, многофактность и т. д. Такова специфика всех систем, в которые включены люди. Поэтому неработоспособность традиционных математических моделей во многих случаях – это не методологический порок когнитивного анализа, а фундаментальное свойство предмета исследования.

Таким образом, важнейшей особенностью большинства изучаемых в теории управления ситуаций является наличие в них мыслящих участников, причем каждый из которых по-своему представляет ситуацию и принимает те или иные решения, исходя из «своего» представления. Как отметил Дж. Сорос в своей книге «Алхимия финансов», «когда в ситуации действуют мыслящие участники, последовательность событий не ведет напрямую от одного набора факторов к другому; вместо этого она перекрестным образом... соединяет факторы с их восприятиями, а восприятия с факторами». Это приводит к тому, что «процессы в ситуации ведут не к равновесию, а к никогда не заканчивающемуся процессу изменений». Отсюда следует, что достоверное предсказание поведения процессов в ситуации невозможно без учета оценки этой ситуации ее участниками и их собственных предположений о возможных действиях. Эту особенность некоторых систем Дж. Сорос назвал рефлексивностью.

Формализованные количественные зависимости факторов описываются разными формулами (закономерностями), зависящими от предмета исследования, т. е. от самих факторов. Однако, как уже упоминалось, построение традиционной математической модели не всегда возможно. Проблема универсальной формализации взаимовлияния факторов до сих пор не решена и вряд ли когда-либо будет решена, поэтому необходимо смириться с тем, что далеко не всегда возможно описание связей факторов математическими формулами, т. е. далеко

не всегда возможна точная количественная оценка зависимостей. В когнитивном моделировании оценка весов дуг часто основывается на субъективном мнении эксперта. Основная задача при этом – компенсировать субъективность и искажение оценок посредством разного рода процедур верификации. При этом обычно недостаточно одной проверки оценок эксперта на непротиворечивость. Главная цель процедуры обработки субъективных мнений эксперта – помочь ему отрефлексировать, более чётко осознать и систематизировать свои знания, оценить их непротиворечивость и адекватность реальности. В процессе извлечения знаний эксперта происходит взаимодействие эксперта – источника знаний – с когнитологом (инженером по знаниям) или с компьютерной программой, что позволяет проследить за ходом рассуждения специалистов при принятии решений и выявить структуру их представлений о предмете исследования.

### **12.15. Проверка адекватности модели**

Предложено несколько формальных процедур проверки адекватности выстроенной модели. Однако, поскольку модель строится не только на формализованных отношениях факторов, математические методы проверки ее правильности не всегда дают точную картину. Поэтому исследователи предложили своего рода «исторический метод» проверки адекватности модели. Иначе говоря, разработанная модель какой-либо ситуации применяется к подобным ситуациям, существовавшим в прошлом, и динамика которых хорошо известна. В том случае, если модель оказывается работоспособной, т. е. выдаёт прогнозы, совпадающие с реальным ходом событий, она признаётся правильной. Ни один из методов верификации модели в отдельности не является исчерпывающим, поэтому целесообразно применение комплекса процедур проверки правильности.

### **12.16. Применение когнитивных моделей в СППР**

Главное назначение когнитивной модели – помочь эксперту в процессе познания и соответственно выработки правильного решения. Поэтому когнитивный подход наиболее распространён в системах поддержки принятия решений (СППР). Когнитивная модель визуализирует и упорядочивает информацию



об обстановке, замысле, целях и действиях. При этом визуализация выполняет важную когнитивную функцию, иллюстрируя не только результаты действий субъекта управления, но и подсказывая ему способы анализа и генерирования вариантов решений.

Когнитивная модель служит не только для систематизации и «прояснения» знаний эксперта, но и для выявления наиболее выгодных «точек приложения» управляющих воздействий субъекта управления. Иначе говоря, когнитивная модель объясняет, на какой фактор или взаимосвязь факторов необходимо воздействовать, с какой силой и в каком направлении, чтобы получить желаемое изменение целевых факторов с наименьшими затратами.

Управляющие воздействия могут быть кратковременными (импульсными) или продолжительными (непрерывными), действующими вплоть до достижения цели. Возможно и совместное использование импульсных и непрерывных управляющих воздействий.

При достижении заданной цели сразу же встает задача удержания ситуации в достигнутом благоприятном состоянии до тех пор, пока не появится новая цель. В принципе, задача удержания ситуации в требуемом состоянии не отличается от задачи достижения цели. Комплекс взаимосвязанных управляющих воздействий и их логичная временная последовательность составляют целостную стратегию управления (модель управления).

Применение разных моделей управления может привести к разным результатам. Здесь важно уметь предсказать, к каким последствиям приведёт, в конечном итоге, та или иная управленческая стратегия. Для разработки такого рода прогнозов используется сценарный подход (сценарное моделирование) в рамках когнитивного анализа. Иногда сценарное моделирование называют «динамическим имитационным моделированием».

Сценарный подход представляет собой своего рода «разыгрывание» разных вариантов развития событий в зависимости от избранной модели управления и поведения непредсказуемых факторов. Для каждого сценария выстраивается триада «исходные предпосылки – наше воздействие на ситуацию – полученный результат». Когнитивная модель в этом случае позволяет учесть весь

комплекс эффектов управляющих воздействий для разных факторов, динамику факторов и их взаимосвязей при разных условиях.

Таким образом выявляются все возможные варианты развития системы и вырабатываются предложения по поводу оптимальной стратегии управления для реализации желаемого сценария из возможных. Исследователи довольно часто включают сценарное моделирование в число этапов когнитивного анализа или же рассматривают сценарное моделирование как дополнение к когнитивному анализу. Если суммировать и обобщить мнения исследователей относительно стадий сценарного моделирования, то в самом общем виде этапы сценарного анализа можно представить следующим образом.

1. Выработка цели управления (желаемого изменения целевых факторов).
2. Разработка сценариев развития ситуации при применении разных стратегий управления.
3. Определение достижимости поставленной цели (реализуемости сценариев, ведущих к ней).
4. Проверка оптимальности уже намеченной стратегии управления (если таковая имеется).
5. Выбор оптимальной стратегии, соответствующей наилучшему, с точки зрения поставленной цели, сценарию.

Конкретизация оптимальной управленческой модели – разработка конкретно-практических рекомендаций руководителям. Эта конкретизация включает в себя выявление управляющих факторов (посредством которых можно влиять на развитие событий), определение силы и направленности управляющих воздействий на управляющие факторы, предсказание вероятных кризисных ситуаций вследствие влияния непредсказуемых внешних факторов и т. п. Следует заметить, что этапы сценарного моделирования могут меняться в зависимости от объекта исследования и управления.

На начальном этапе моделирования может быть достаточно качественной информации, не имеющей точного числового значения и отражающей суть ситуации. При переходе к моделированию конкретных сценариев все более значимым становится использование количественной информации, представляю-

щей собой числовые оценки значений каких-либо показателей. В дальнейшем для проведения необходимых вычислений используется в основном количественная информация.

Самым первым сценарием, который не требует никаких действий исследователя по его формированию, является саморазвитие ситуации (в данном случае вектор управляющих воздействий «пуст»). Саморазвитие ситуации является отправной точкой для дальнейшего формирования сценариев. Если исследователя устраивают результаты, полученные при саморазвитии (другими словами, если в ходе саморазвития достигаются поставленные цели), то дальнейшее сценарное исследование сводится к изучению влияния на ситуацию тех или иных изменений внешней среды.

Существуют два основных класса сценариев: сценарии, моделирующие внешние воздействия и сценарии, моделирующие целенаправленное (управляемое) развитие ситуации.

Необходимо заметить, что в исследованиях примеры использования когнитивного сценарного моделирования обычно приводятся в весьма общем виде, поскольку, во-первых, подобного рода информация является эксклюзивной и представляет определённую коммерческую ценность, и, во-вторых, каждая конкретная ситуация (система, среда, объект управления) требует индивидуального подхода. Существующая теоретическая база когнитивного анализа, хотя и требует уточнений и развития, позволяет разным субъектам управления заняться разработкой собственных когнитивных моделей, поскольку, как упоминалось, предполагается, что для каждой области, для каждой проблемы составляются специфические модели.

### **12.17. Компьютерные СППР**

Когнитивный анализ и моделирование являются принципиально новыми элементами в структуре систем поддержки принятия решений. Технологии когнитивного моделирования позволяют:

- исследовать проблемы с нечеткими факторами и взаимосвязями;
- учитывать изменения внешней среды;

– использовать объективно сложившиеся тенденции развития ситуации в своих интересах.

Такие технологии завоевывают все большее доверие у структур, занимающихся стратегическим и оперативным планированием на всех уровнях и во всех сферах управления.

Проведение когнитивного анализа нестабильных, слабоструктурированных ситуаций и сред является крайне сложной задачей, для решения которой привлекаются информационные системы. По существу, эти системы предназначены для повышения эффективности механизма принятия решений, поскольку главной прикладной задачей когнитивного анализа является оптимизация управления. Системы поддержки принятия решений, как правило, являются диалоговыми. Они предназначены для обработки данных и реализации моделей, помогающих решать отдельные, в основном слабоструктурированные или неструктурированные задачи. Эти системы могут обеспечивать работников информацией, необходимой для принятия индивидуальных и групповых решений. Такие системы обеспечивают непосредственный доступ к информации, отражающей текущие ситуации, все факторы и связи, необходимые для принятия решений.

В методиках качественного анализа и, в частности, для построения когнитивных карт используются компьютерные программы, базирующиеся на гипертекстовой технологии.

Hyper RESEARCH, ATLAS/ti, Metamorph, KANT, NUDIST, Meta Design, Гипердок. Разработаны системы, позволяющие строить когнитивные карты непосредственно на основе анализа текста интервью, статьи, – MEGA, Sem Net.

С точки зрения науки управления сегодня особенно важно использование мягкого резонансного управления сложными социально-экономическими системами, искусство которого состоит в способах самоуправления и самоконтроля систем. Слабые, так называемые резонансные явления, чрезвычайно эффективны для «раскрутки» или самоуправления, так как они соответствуют внутренним тенденциям развития сложных систем. Основная проблема заклю-

частся в том, как малым резонансным воздействием подтолкнуть систему на один из собственных и благоприятных для системы путей развития, как обеспечить самоуправление и самоподдерживаемое развитие (самораскрутку).

Для задач, связанных с организационными системами, проблема неопределенности в описании и моделировании функций участников является не методологической, а внутренне присущей самому предмету исследований. Возможны различные постановки задачи об управлении ситуацией в зависимости от полноты доступной участникам информации о ситуации и об остальных участниках, в частности для поиска резонансного и синергетического эффектов, когда улучшение ситуации при одновременном воздействии на нее нескольких участников больше «объединения» положительных эффектов от каждого из участников по отдельности.

С точки зрения науки управления сегодня особенно важно использование мягкого резонансного управления сложными социально-экономическими системами, искусство которого состоит в способах самоуправления и самоконтроля систем. Слабые, так называемые резонансные явления, чрезвычайно эффективны для «раскрутки» или самоуправления, так как они соответствуют внутренним тенденциям развития сложных систем. Основная проблема заключается в том, как малым резонансным воздействием подтолкнуть систему на один из собственных и благоприятных для системы путей развития, как обеспечить самоуправление и самоподдерживаемое развитие (самораскрутку).

## **13. НОВЫЕ ПРОБЛЕМЫ БОЛЬШИХ ДАННЫХ И ПРИМЕРЫ**

### **13.1. Примеры успешных применений аналитики БД**

Рассмотрим некоторые примеры успешных применений средств анализа больших данных для решения практически интересных и достаточно крупномасштабных задач. В основном они построены на базе компьютерных инфраструктур IBM, платформы *Apache Hadoop*, а также перечисленных выше программных средств анализа больших данных разработки IBM.

*Управление воздействием среды на реки – анализ потоковой информации* (Beacon Institute, Clarkson University). Цель проекта – это анализ комплексного *взаимодействия* человеческого сообщества и среды его обитания. Система анализирует потоковые данные физического, химического и биологического характера, собранные в районе рек штата Нью-Йорк, США. Информация собирается сенсорами, роботами и средствами мобильного мониторинга с использованием компьютерных технологий. Массив собираемых данных предназначен для пространственного мониторинга вариаций таких данных среды, как температура, давление, минерализация и мутность воды, растворенный в ней кислород и другие химические характеристики воды. Эти данные поступают в реальном времени, так что поток данных имеет достаточно большую интенсивность. Обработка данных ведется средством *IBM@InfoSphere@Streams*. Он обеспечивает сбор и анализ данных от тысяч источников, визуализацию движения химических составляющих. Выполняется мониторинг качества воды, вырабатываются рекомендации по защите рыбы на путях ее миграции. В целом система помогает ученым лучше понять взаимодействия рек и окружающей их природной среды.

*Онлайн-анализ потоков данных для оценки дорожного трафика* (KTN Institute и Royal Institute of Technology, Швеция). Система предназначена для онлайн-анализа дорожного трафика на основе данных, собираемых с большого количества автомобилей, радарных сенсоров, установленных вдоль дорог, данных о скоплениях машин у станций оплаты, о погоде, о дорожных работах и инцидентах и т. п. Система анализа дорожного трафика построена на базе про-

граммного инструмента *IBM®InfoSphere®Streams*. Информация, получаемая в результате обработки этих данных, используется для оценки времени, которое потребуется тому или иному водителю для перемещения из текущего положения в заданное место. Она используется для того, чтобы предлагать водителям разные маршруты и улучшить состояние дорожного движения в центре города.

В числе других успешных приложений стоит упомянуть такие разработки, как:

- система *TerraEchos@IBM*, предназначенная для разведки и сенсорного наблюдения, а также для защиты критических инфраструктур, для обеспечения безопасности по периметру и в районе границ охраняемого объекта;
- система *Adelos S4*, которая предназначена для анализа структурированного аудиопотока, состоящего их данных, получаемых от акустических сенсоров (она используется в интересах ВМС США).

Однако все эти и большинство других программных средств анализа больших данных, существующих в настоящее время, имеют пока еще достаточно ограниченные возможности и, по сути, реализуют функции технологии OLAP-анализа данных, т. е. технологии, разработанной еще в 1990-х гг. Следует заметить, что возможности существующих средств интеллектуального анализа «*данных умеренного масштаба*» (по сравнению с масштабом больших данных), несравненно более мощные и по разнообразию решаемых задач, и по глубине анализа данных.

Рассмотрим, с чем связаны основные проблемы интеллектуального анализа больших данных и почему в настоящее время практически отсутствуют эффективные программные решения этой проблемы.

### **13.2. Новые проблемы, обусловленные особенностями БД**

Большинство традиционных методов анализа данных, которые базируются на выявлении и анализе связей между атрибутами данных в рамках статистических моделей, напрямую не может быть использовано для работы с большими данными. Это обусловлено рядом специфических свойств БД, среди ко-

торых самым существенным свойством является большая размерность пространства признаков, которая может исчисляться десятками тысяч и более. Интеллектуальный анализ больших данных становится в особенности трудной задачей в тех случаях, когда размерность пространства признаков больше числа примеров в обучающей выборке (как говорят в таких случаях, *размерность данных больше их объема*).

Ключевой задачей интеллектуального анализа данных является поиск зависимостей и других типов связей между атрибутами некоторого процесса или явления. Среди атрибутов данных особую роль играют так называемые *целевые переменные*, которые либо задают значения показателей качества, которые нужно оптимизировать, либо являются именами классов в задачах классификации, и тогда другие атрибуты данных выступают в роли признаков. В общем случае основная задача анализа больших данных состоит в построении модели одной или нескольких целевых переменных в виде функции от других переменных – атрибутов данных.

### 13.3. Накопление ошибок

Хорошо известно, что значения атрибутов данных, которые обычно получаются в результате измерений, всегда содержат ошибки. Обычно это ошибки сенсоров. В том случае, когда число таких атрибутов очень велико, в процессе обработки данных ошибки промежуточных и финальных вычислений, как правило, нарастают и постепенно начинают доминировать над полезным сигналом. Этот эффект принято называть *накоплением ошибок, noise accumulation*. Особенно катастрофический характер такие ошибки начинают играть в случаях, когда алгоритмы обработки включают в себя, например, обращение матриц ковариаций, поиск собственных значений и др. При этом воздействие ошибок может быть столь специфичным, что известные приемы обеспечения вычислительной устойчивости, например, методы регуляризации, не помогают.

Поясним на частном примере, почему это происходит. Обычно среди атрибутов всегда имеется много «слабых», или, как иногда говорят, «малоинформативных» атрибутов. Но заранее неизвестно, какие именно атрибуты малоин-



формативные, а какие информативные, тем более что информативность того или иного атрибута зависит от решаемой задачи. Поэтому **поиск информативных атрибутов** обычно является центральной и наиболее трудной компонентой задачи интеллектуального анализа данных. Негативная роль накопления ошибок наглядно демонстрируется на примере задачи классификации при большом исходном числе атрибутов (признаков), с использованием которых может строиться модель классификации. Рассмотрим этот пример.

Пусть имеются два класса данных, представленных выборками с нормальными распределениями  $X_1, \dots, X_n \in N_d(\mu_1, I_d)$  и  $Y_1, \dots, Y_n \in N_d(\mu_2, I_d)$ , где  $I_d$  – единичная ковариационная матрица размером  $d$ ,  $n = 100$  (объем выборки) и  $d = 1000$  (размерность пространства признаков). Значения компонент вектора математических ожиданий признаков в первом классе полагаются равными нулю для всех элементов вектора. Во втором классе первые 10 элементов вектора математических ожиданий признаков полагаются равными 3, а остальные полагаются равными 0. При небольшом числе признаков эта задача не представляет трудностей и решается достаточно просто, например, с помощью линейных классификаторов.

Пусть при решении описанной задачи классификации во внимание принимаются  $m$  наилучших признаков из всего их пространства размерности 1000, а алгоритм классификации строится с использованием критерия максимума значения меры близости. Пусть выбрана мера близости, значение которой для заданного входного вектора признаков по отношению к каждому классу вычисляется как сумма его проекций на первую и вторую главные компоненты соответствующего класса. Эти компоненты вычисляются по данным выборок обоих классов. Исследуется влияние накопления ошибок в зависимости от значения  $m$  – числа учитываемых признаков (размерности подпространства признаков). Эксперименты выполнены для  $m = 2$ ,  $m = 40$ ,  $m = 200$  и  $m = 1000$ . В каждом таком эксперименте строятся проекции векторов примеров обучающих множеств (общее число точек в каждом множестве данных равно 100) на двумерное пространство первых двух главных компонент (рис. 13.1).

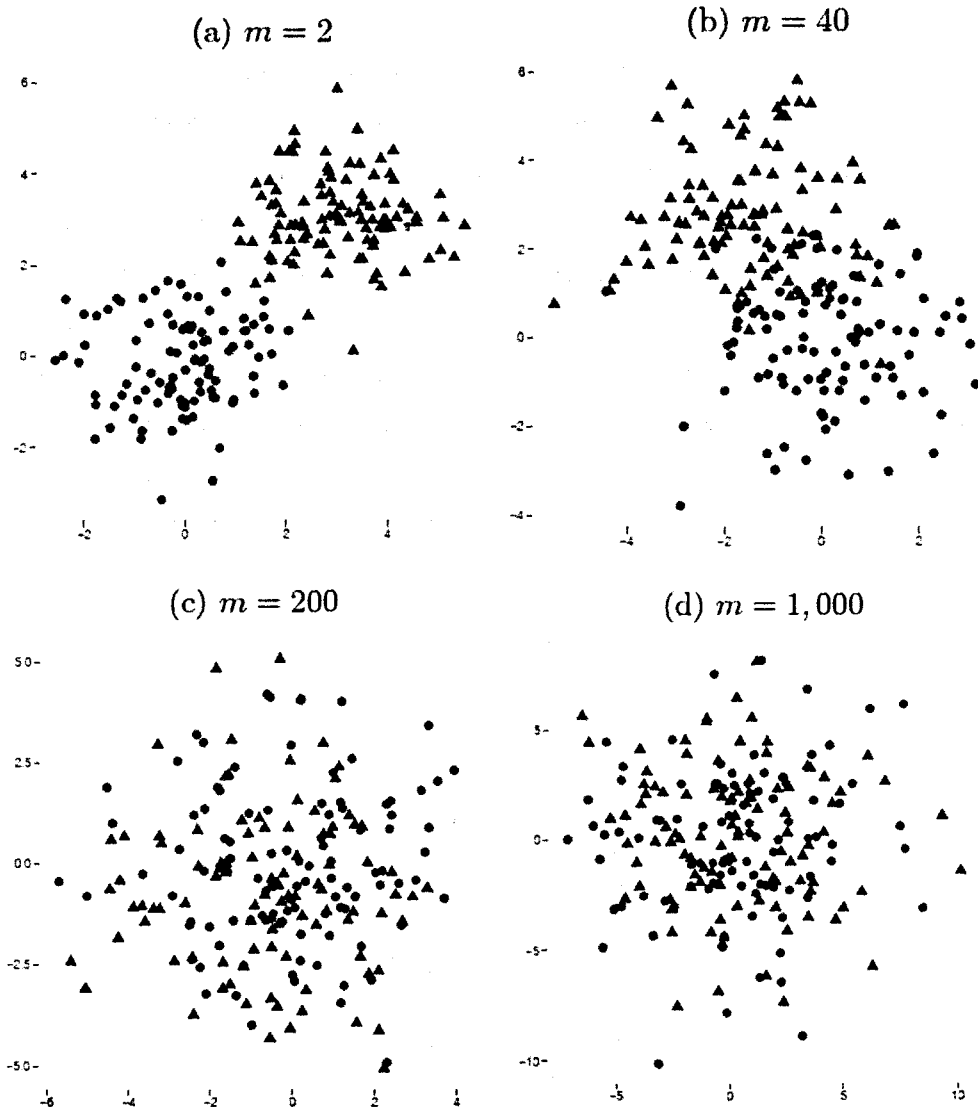


Рис. 13.1. Графики разброса проекций наблюдаемых данных

Видно, что при  $m = 2$  первые две главные компоненты образуют пространство атрибутов, в котором представители разных классов хорошо разделяются линейной границей. Несколько хуже, но они все же разделяются и при использовании 40 наилучших атрибутов. Но уже здесь начинает сказываться влияние эффекта накопления ошибок. При  $m = 200$  и  $m = 1000$  накопленная ошибка (ошибка измерений признаков и ошибка вычислений) уже значительно превосходит полезный сигнал и классы оказываются неразделимыми в про-

странстве первых двух главных компонент. Подобное явление является характерным и для других задач обработки больших данных. Таким образом, накопление ошибок может оказать негативное влияние на результаты обработки больших данных. В этой связи подчеркивается, во-первых, важность правильного выбора *минимальной размерности* пространства для построения моделей целевых переменных для принятия решений и, во-вторых, важность выбора *наилучших переменных* (атрибутов данных). Кроме того, при поиске статистических зависимостей между переменными важно избегать методов, которые требуют обращения матриц или использования других вычислительно-неустойчивых алгоритмов. Но основная проблема здесь состоит именно в том, чтобы выбрать минимальное число наилучших переменных. Эта задача всегда имеет экспоненциальную сложность и для больших данных ее решение осложняется еще и другими свойственными им проблемами.

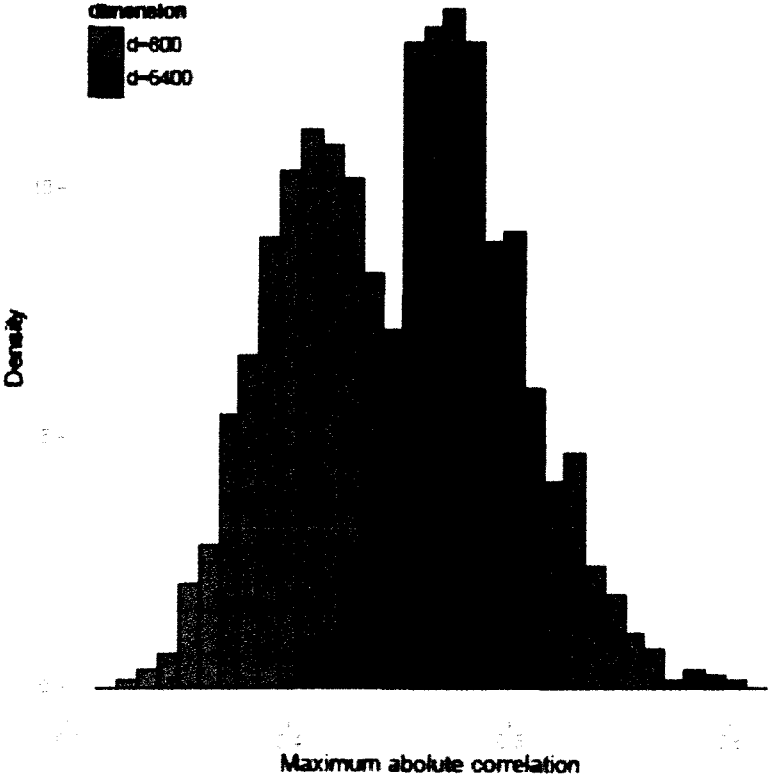


Рис. 13.2. Иллюстрация ложных корреляций

### 13.4. Возникновение ложных выборочных корреляций

В качестве иллюстрации эффекта возникновения ложных выборочных корреляций (рис. 13.2) приводятся результаты экспериментов по оценке значения выборочного коэффициента корреляции на множестве независимых переменных – компонент вектора  $X=[x_1, \dots, x_d]^T$ , распределенных по нормальному закону  $N_d(0, I_d)$ ,  $I_d$  – единичная матрица. И хотя эти переменные *теоретически независимы*, при больших значениях размерности вектора  $X$  их выборочные корреляции могут оказаться значительными. Например, на рисунке 13.2 показано распределение (гистограмма) максимального значения выборочного коэффициента корреляции  $r$  переменной  $x_1$  со всеми другими теоретически независимыми переменными:

$$r = \max_j \{corr(x_1, x_j)\}. \quad (1)$$

Это распределение строилось экспериментально для  $n = 60$  при размерностях  $d$  вектора  $X$ , равных 800 и 6400, при этом объем эксперимента равнялся 1000 реализаций для каждого случая размерности. Результат этих экспериментов представлен на рисунке 13.2. Анализ приведенных результатов показывает, что ложные корреляции действительно возникают, причем их число, а также среднее значение коэффициента корреляции возрастают при возрастании размерности пространства независимых случайных переменных.

В результате появления таких ложных корреляций может быть сделан неправильный выбор переменных, задающих модель некоторой целевой переменной на основе измерений атрибутов данных. Самый простой пример – это традиционная задача оценки целевой переменной на основе линейной модели вида

$$Y = X^T A + \beta.$$

Хорошо известная проблема «переобучения» (*overfitting*) в задачах классификации имеет такое же происхождение: она возникает при большой размерности пространства признаков в правиле классификации.

Обычно для того, чтобы понизить размерность задачи линейного оценивания (например, чтобы избежать накопления ошибок и повысить эффективность вычислений и, чтобы сделать более ясной интерпретацию связей, задава-

емых моделью) размерность  $d$  вектора переменных  $X$  искомой модели выбирается небольшой. Но выбор таких переменных является задачей высокой сложности. При этом ложные корреляции между переменными, которые возникают при больших размерностях исходного пространства атрибутов  $X$ , могут привести к тому, что в число таких атрибутов будут включены переменные, не связанные с целевой переменной  $y$ .

Авторы многих работ подчеркивают важность причинного анализа как основного метода минимизации числа переменных, который позволяет избежать опасности построения неверной модели из-за ложных корреляций.

Но кроме получения неверных моделей ложные корреляции могут приводить также и к неверным статистическим выводам. В частности, в присутствии большого числа переменных, которые ошибочно включены в модель вследствие эффекта ложных корреляций, оценки дисперсий коэффициентов модели оказываются слишком заниженными, что приводит к *неверным выводам о статистической значимости* выбранных переменных модели, которые делаются на основании статистических тестов.

### 13.5. Зависимости между помехой и переменными модели

Еще один специфический эффект, возникающий как следствие большой размерности данных, связан с наличием *статистической зависимости между помехой и переменными модели (incidental endogeneity)*. Если сослаться на модель  $Y = X^T A + \beta$ , то речь идет о корреляции между помехой  $\beta$  и компонентами вектора атрибутов  $X$ . Большинство моделей статистического оценивания параметров существенно использует предположение о независимости помехи и переменных. Обратим внимание на то, что в отличие от ложных корреляций здесь речь идет о реальном существовании зависимости между помехой и переменными модели. При этом, чем больше размерность пространства переменных, тем больше шансов появления такой зависимости. В больших данных этот эффект возникает по двум причинам:

1. Переменных оказывается очень много, а это ведет к высокой вероятности появления зависимости между помехой и атрибутами данных, при том, что исследователи часто стремятся получить как можно больше атрибутов.
2. Данные могут быть получены из разных источников, они могут быть получены в разное время и измерены с разной точностью и смещением, что при их объединении также может привести к появлению зависимостей между переменными и ошибкой.

В настоящее время рассматриваемая проблема пока изучена слабо, однако имеются работы, где предлагаются альтернативные методы статистической обработки больших данных, которые работают при более слабых допущениях. В частности, такие методы предложены для решения задач линейной регрессии.

Существуют и другие проблемы, специфические для больших данных, которые делают задачи анализа больших данных очень тяжелыми, что существенно сужает возможности этого анализа. Для успешного решения таких задач требуется пересмотр подходов и методов решения задач статистического анализа. Они требуют использования новых, более адаптивных и робастных процедур обработки, чем общепринятые. Эти процедуры должны быть направлены на то, чтобы сбалансировать вычислительную эффективность, устойчивость и точность вычислений. Хорошо известно, что для успешности интеллектуального анализа данных любого объема и размерности, а тем более – больших данных, нужно:

- принимать любые меры для снижения их размерности;
- обоснованно выбирать переменные, которые далее используются в качестве переменных модели решаемой проблемы.

Некоторые варианты реализации этой парадигмы кратко описываются далее.

### **13.6. Некоторые возможные решения ключевых проблем**

*Модель данных и роль онтологий.* Обычно одни те же большие данные используются для решения целого ряда прикладных задач, так или иначе связанных с задачами поддержки принятия решений. И эти задачи могут сильно

различаться между собой. Для каждого приложения потребуется строить свою модель данных, опираясь на одни и те же данные. Здесь имеются в виду модели, которые строятся для разных целевых переменных. К ним могут предъявляться различные требования, и для решения своих прикладных задач они могут использовать различные атрибуты одних и тех же данных. Заметим, что обычно семантика приложения описывается его онтологией, которая играет для приложения роль метамодели данных. Поэтому при решении конкретной прикладной задачи на основе анализа больших данных нужно говорить о конкретной модели данных, а не вообще о модели данных. В этом смысле любая задача обработки больших данных сводится к построению специфической *модели данных* для конкретного множества *целевых переменных*.

Далее, искомая модель данных должна иметь ясную предметную семантику, что достигается использованием онтологии в качестве метамодели данных. Она, с одной стороны, специализирует использование данных, а с другой – дополняет данные экспертными знаниями. В соответствии с современными представлениями онтология должна быть непременной компонентой модели данных.

Напомним, что обычно онтология представляется иерархией классов понятий (категорий) предметной области, каждому из которых поставлено в соответствие некоторое множество атрибутов. Кроме того, на множестве понятий онтологии задаются и другие типы отношений. Не останавливаясь на особенностях онтологической модели больших данных и на особенностях технологии ее создания в случае больших данных, отметим, что по большей части эти особенности влекут большую трудоемкость, но не какие-либо новые проблемы. Поэтому здесь важным является привлечение методов автоматизации построения онтологий, которые позволили бы снизить нагрузку на экспертов. Однако это самостоятельная проблема, которая заслуживает отдельного рассмотрения.

*Снижение размерности.* Типовые процедуры технологии работы с большими данными должны строиться так, чтобы реализовать *базовую парадигму их обработки*: в технологии анализа данных должны приниматься все

меры для того, чтобы *снизить размерность данных*, вовлекаемых в обработку. В общем случае методы, реализующие идею снижения размерности, можно разделить на две группы:

- методы, которые позволяют вовлекать в обработку лишь часть данных, сохраняя при этом, по возможности, представительность подвыборки данных и статистическую значимость вычисляемых оценок; эти методы обычно реализуют случайный выбор подвыборки;
- методы, направленные на снижение размерности пространства представления данных.

В общем случае используемые атрибуты могут быть либо подмножеством значимых исходных атрибутов модели, либо функциями от всех или только части этих атрибутов. Заметим, что программный инструмент *Apache Hadoop* частично реализует второй вариант снижения размерности.

Методы первого типа обычно опираются на случайный выбор множества примеров с соблюдением некоторых требований. Они понятны и не требуют особого комментария. Другое дело – методы второго типа. Как уже ранее неоднократно отмечалось, выбор подмножества переменных для задания модели целевых переменных является ключевой проблемой анализа больших данных. Именно ее успешное решение должно обеспечить преодоление проблем, о которых шла речь в предыдущем разделе.

Ниже дается краткое описание методов, которые разработаны в СПИИРАН. Они построены таким образом, чтобы при их реализации не возникало необходимости использовать вычислительно неустойчивых алгоритмов поиска, алгоритмов декорреляции корреляционных матриц, перемножения большого числа малых величин или деления на малые величины и т. п. Кроме того, в рассматриваемых далее алгоритмах основной акцент делается на работу с целочисленными переменными, что всегда повышает устойчивость алгоритмов, поскольку при работе с целочисленными переменными не возникает ошибок округления и ослабляется влияние эффекта накопления ошибок. Кроме того, акцент делается на поиске при-



чинных зависимостей между переменными, что снижает возможность появления ложных корреляций.

Далее рассматриваются три базовых алгоритма, последовательное применение которых позволяет эффективно построить модель прогноза значения целевой переменной.

**Агрегирование атрибутов.** Цель алгоритма – снижение размерности пространства переменных, задающих большие данные, за счет агрегирования областей значений атрибутов данных. Этот алгоритм выполняет преобразование данных, представленных в терминах значений атрибутов, в множество утверждений о свойствах подмножеств их значений. Поясним это на примере. Пусть обнаружено подмножество значений атрибутов, обладающих некоторым свойством, которое формально задается предикатом. Оставляя пока в стороне семантику этого предиката (конкретное свойство, которым обладает задаваемый им агрегат данных), отметим, что он принимает значение «истина» в области, заданной выражением в скобках. Такой предикат может использоваться в качестве посылки некоторого правила, которое задает общее свойство элементов из области истинности этого предиката.

Такая процедура построения агрегатов конструктивна, так как оценки условных вероятностей могут быть просто вычислены по имеющемуся множеству примеров. Здесь для простоты понимания предполагается, что область значений атрибута представляет собой дискретное множество, хотя аналогичный подход реализуется и для числовых (непрерывных) атрибутов. В общем случае могут быть использованы различные показатели качества (меры информативности) для построения подмножеств. Обратим внимание на то, что в модели используется одна из наиболее естественных мер – оценка апостериорной вероятности класса, которая, в частности, используется в модели *наивного байесовского классификатора*.

Построение таких предикатов и вычисление оценок по обучающим данным названо в рассматриваемом алгоритме *агрегированием данных*.

В описанной процедуре в качестве меры информативности агрегирования принята наиболее естественная мера, а именно вероятность правильного распознавания с помощью наивного байесовского классификатора по отношению к тому или иному классу решений. Очевидно, что в общем случае может быть выбран *любой другой классификатор*, который строится на каждом отдельном атрибуте, однако использует ту же самую *вероятностную меру информативности* одинарного признака. Такой подход обладает двумя принципиально важными свойствами:

- в нем используется содержательно понятная и естественная мера информативности;
- результатом агрегирования является преобразование всех атрибутов данных к предикатной форме, т. е. к единой шкале измерения, что существенно упрощает дальнейшую обработку и анализ больших данных;
- оценки мер информативности, построенные в процессе поиска агрегатов на множестве значений отдельных атрибутов, могут использоваться для пороговой фильтрации агрегатов. В результате этой фильтрации малоинформативные агрегаты будут удалены из дальнейшего рассмотрения. Регулируя значение порога, можно регулировать число агрегатов, которые далее продолжают претендовать на роль переменных модели целевой переменной. Конкретный алгоритм фильтрации может зависеть от особенностей данных и приложения.

Обратим внимание на то, что в результате шага агрегирования атрибутов для каждого класса множества решений будет построено свое индивидуальное *множество агрегатов – признаков представления данных*. Индивидуализация модели описания данных, относящихся к каждому классу решений, является одной из важных новых черт описываемого подхода по сравнению с известными моделями обработки данных. Важно также, что построенные агрегаты есть утверждения о свойствах данных, выраженные в терминах понятий, отношений и атрибутов понятий онтологии, построенные с учетом конкретного контекста, в котором такие свойства проявляются в экземплярах объектов данных.

**Причинный анализ агрегатов.** Выражения, построенные на этапе поиска и фильтрации агрегатов, представляют собой ассоциативные правила классификации. Теоретически доказано, что среди ассоциативных правил наибольший интерес представляют такие правила, в которых связь носит причинный характер.

Однако в общем случае выделение причинных связей на множестве ассоциативных связей является непростой задачей. Традиционные методы их обнаружения базируются на использовании модели причинных Байесовских сетей доверия. В них формальное определение причинной связи дается в терминах Марковского покрытия узла причинной байесовской сети, под которым понимается множество его «родителей», его «потомков» и других «родителей потомков» узла. Однако эта задача имеет экспоненциальную сложность. Поэтому построить причинную байесовскую сеть для случая, когда число найденных правил исчисляется тысячами (а на практике их может быть на несколько порядков больше), не представляется возможным. Поэтому еще в конце 1990-х гг. появились работы, которые пытались построить альтернативные модели поиска причинных связей. В них ставилась цель найти *числовые меры* оценки «силы» причинной связи, не опираясь на сложные структуры типа причинных сетей. Эти модели называют *причинно-ассоциативными*. Из публикаций известно, что модель причинной Байесовской сети можно использовать для случая, когда число атрибутов (узлов сети) не более двух десятков.

**Минимизация модели целевой переменной.** Специалисты в области интеллектуальной обработки больших данных обычно обращают мало внимания на то, что существенные переменные могут быть сильно зависимыми. Они могут быть просто сильно коррелированными, одни причины могут быть следствием других причин и др. Эта проблема известна в области коллективного распознавания и слияния данных как проблема разнообразия классификаторов. Уменьшение числа таких сильно коррелированных атрибутов в модели целевых переменных дает еще один очень существенный источник снижения размерности модели целевой переменной.

В известной литературе рассматриваются различные эвристические подходы к решению этой проблемы. Разработан математически корректный алгоритм снижения размерности модели целевой переменной за счет построения иерархии кластеров на множестве причин и использования в модели переменных только по одному представителю каждого кластера. Этот алгоритм, как и описанные выше алгоритмы агрегирования и причинной фильтрации, к настоящему времени проверены на практике и подтвердили работоспособность.

## ЗАКЛЮЧЕНИЕ

Бурное развитие нейронных сетей, эволюционного моделирования и нечеткой логики приводит к большему их взаимодействию. И если раньше они развивались параллельно друг другу, то теперь уже идут бок о бок. На первый план, как апогей такого взаимодействия, выходит создание искусственного интеллекта. И хотя сама идея его создания появилась задолго до появления нейронных сетей, эволюционного моделирования и нечеткой логики, на данном этапе стало понятно, что симбиоз этих методов становится основой для искусственного интеллекта. Нейронные сети, эволюционное моделирование и нечеткая логика бурно развиваются и находят всё новые и новые ниши для себя. При этом идет создание искусственного интеллекта. С каждым годом идей и программ с применением алгоритмов нейронных сетей, эволюционного моделирования и нечеткой логики становится всё больше и больше.

Результаты анализа показывают, что состояние аналитики БД в настоящее время таково, что приходится гораздо больше говорить о трудных проблемах, чем об эффективных решениях. Задачи в области обработки больших данных, которые сейчас оказывается возможным решать существующими средствами, пока далеко не отвечают тем ожиданиям, которые стимулировали развитие этой области. В лекции перечислены некоторые причины такого состояния исследований и разработок, а также приведены некоторые оригинальные результаты, которые на основании уже имеющегося практического их применения могут обеспечить прогресс в интеллектуальной обработке больших данных.

## СПИСОК ЛИТЕРАТУРЫ

1. *Акофф, Р.* Планирование будущего корпорации. – М., 1985.
2. *Алексеев, П. В.* *Философия* / П. В. Алексеев, А. В. Панин. – М.: Проспект, 1996.
3. *Берталанфи, Л.* Общая теория систем: критический обзор // Исследования по общей теории систем. – М., 1969. – С. 23–82.
4. *Блауберг, И. В.* Системный подход и системный анализ / И. В. Блауберг, Э. М. Мирский, В. Н. Садовский // Системные исследования. – М., 1982. – С. 47–64.
5. *Богданов, А. А.* Тектология. – М., 1989.
6. *Вагнер, Р.* Социология: к вопросу о единстве научной дисциплины // Социологический журнал. – М., 1996. – № 3, 4. – С. 60–83.
7. *Вертгеймер, М.* Продуктивное мышление. – М., 1987.
8. *Винер, Н.* Кибернетика, или управление и связь в животном и машине. – М., 1983.
9. *Гиг, Дж. ван.* Прикладная общая теория систем. – М., 1981.
10. *Гидденс, Э.* Элементы теории структуриации // Современная социальная теория: Бурдье, Гидденс, Хабермас. – Новосибирск, 1995. – С. 40–72.
11. *Громов, И. А.* Западная теоретическая социология / И. А. Громов, А. Ю. Мацкевич, В. А. Семенов. – СПб., 1996.
12. *Даяилов-Данильян, В. И.* Моделирование: системно-методологический аспект / В. И. Даяилов-Данильян, А. А. Рыбкин // Системные исследования. 1982. – М., 1982. – С.182–209.
13. *Клини, С. Н.* Введение в метаматематику. – М., 1957.
14. *Кимелев, Ю. А.* Теория общества Энтони Гидденса / Ю. А. Кимелев, Н. Л. Полякова ; под ред. Н. Л. Поляковой // Современные социологические теории общества. – М.: ИНИОН, 1996. – С. 33–57.
15. *Клир, Дж.* Системология. Автоматизация решения системных задач. – М., 1990.

16. *Кравченко, С. А.* Социология: парадигмы и темы / С. А. Кравченко, М. О. Мнацаканян, Н. Е. Покровский. – М., 1997.
17. *Луман, Н.* Глоссарий // Социологический журнал. – 1995. – № 3. – С. 125–127.
18. *Луман, Н.* Понятие общества / Н. Луман ; под ред. А. О. Бороноева // Проблемы теоретической социологии. – СПб., 1994. – С. 25–42.
19. *Манн, М.* Общества как организованные сети власти / М. Манн ; под ред. Н. Л. Поляковой // Современные социологические теории общества. – М. : ИНИОН, 1996. – С. 24–32.
20. *Маркс, К.* Сочинения / К. Маркс, Ф. Энгельс. – 2-е изд. – Т. 23.
21. *Матурана, У.* Биология познания // Язык и интеллект. – М., 1996. – С. 95–142.
22. *Моисеев, Н. Н.* Социализм и информатика. – М., 1988.
23. *Монсон, П.* Современная западная социология: теория, традиции, перспективы. – СПб., 1991.
24. *Морозов, Е. И.* Методология и методы анализа социальных систем. – М. : Изд-во МГУ, 1995.
25. *Наппельбаум, Э. Л.* Системный анализ как программа научных исследований – структура и ключевые понятия // Системные исследования. 1979. – М., 1980. – С. 55–77.
26. *Пригожин, И.* Порядок из хаоса / И. Пригожин, И. М. Стенгерс. – М., 1986.
27. *Перегудов, Ф. И.* Введение в системный анализ / Ф. И. Перегудов, Ф. П. Тарасенко. – М., 1989.
28. *Разумовский, О. С.* Бихевиоральные системы. – Новосибирск : Наука, 1993.
29. *Рапопорт, А.* Мир – созревшая идея. – Дармштадт : Дармштадтер Блаттер, 1993.
30. *Сорокин, П. А.* Система социологии. – Пг., 1920. – Т. 1.

31. *Садовский, В. Н.* Системный подход и общая теория систем: статус, основные проблемы и перспективы развития // *Системные исследования*. 1987. – М., 1987. – С. 29–54.
32. *Хабермас, Ю.* Современная западная теоретическая социология. – М. : ИНИОН, 1992. – Вып. 1.,
33. *Социология. Основы общей теории / под ред. Г. В. Осипова, Л. К. Москвичева.* – М., 1996.
34. *Тернер, Дж.* Структура социологической теории. – М., 1985.
35. *Уемов, А. И.* Системный подход и общая теория систем. – М., 1972.
36. *Урсул, А. Д.* Отражение и информация. – М., 1973.
37. *Форрестер, Дж.* Мировая динамика. – М., 1978.
38. *Хьюбнер, К.* Критика научного разума. – М., 1994.
39. *Штомпка, П.* Социология социальных изменений. – М., 1996.
40. *Ackoff, R. L.* Reflection on systems and their models / R. L. Ackoff, S. Gharajedaghi // *Systems Research*. 1996. – Vol. 13. – № 1. – P. 13–23.
41. *Burns, T. R.* The shaping of social organization / T. R. Burns, H. Flam. – L. : SAGE, 1987.
42. *Flood, R. L.* Dealing with complexity. An Introduction to the Theory and Application of Systems Science / R. L. Flood, E. R. Carson. – N. Y. : Plenum, 1993.
43. *Gigch, S. P. van.* Systems Design, Modeling and Metamodeling. – N.Y. : Plenum, 1991.
44. *Luhmann, N.* Essays on self-reference. – N. Y. : Columbia Univ. Press, 1990.
45. *Maturana, H. R.* Autopoiesis and Cognition: The Realization of Living / H. R. Maturana, F. G. Varela. – Dordrecht : Reidel, 1980.
46. *Mingers, J. A* Comparison of Maturana's Autopoietic Social Theory and Giddens Theory of Structuration // *Systems Research*. 1996. – Vol. 13. – № 4. – P. 469–482.
47. *Mingers, J.* Self-producing systems. Implications and Applications of Autopoiesis. – N. Y. : Plenum. 1995.



48. *Mingers, J.* The cognitive theories of Maturana and Varela // *System Practice*. 1991. – Vol. 4. – № 4. – P. 31–338.
49. *Sewell, W.* A theory of structure: Duality, agency and transformation // *American journal of sociology*. 1992. – Vol. 98. – № 1. – P. 1–30.
50. *Sorokin, P. A.* *Social theory today*. – N. Y. : Harper & Row, 1966.
51. *Spenser Brown, S. G.* *The Laws of Form*. – L. : Alien & Unwin, 1971.
52. *Городецкий, В. И.* Состояние и перспективы интеллектуального анализа больших данных // 7-я Российская мультikonференция по проблемам управления, 7–9 октября 2014 г. – СПб. : Концерн «ЦНИИ „Электроприбор“». – Т. 2. – С. 61–73.
53. *Плотинский, Ю. М.* Модели социальных процессов : учебное пособие для высших учебных заведений. – 2-е изд., перераб. и доп. – М. : Логос, 2001. – 296 с.
54. *Соколова, С. П.* Оценивание состояний сложных систем на основе иммунокомпьютинга / С. П. Соколова, И. В. Усикова, Н. В. Зуева. – СПб.: ГУАП, 2010. – 65 с.

*Андрей Владимирович МАКШАНОВ,  
Антон Евгеньевич ЖУРАВЛЕВ,  
Любовь Николаевна ТЫНДЫКАРЬ*

**БОЛЬШИЕ ДАННЫЕ.  
BIG DATA  
Учебник**

Зав. редакцией  
литературы по информационным технологиям  
и системам связи *О. Е. Гайнутдинова*  
Ответственный редактор *Т. С. Спирина*  
Корректор *В. А. Иутин*  
Выпускающий *О. В. Шилкова*

ЛР № 065466 от 21.10.97  
Гигиенический сертификат 78.01.10.953.П.1028  
от 14.04.2016 г., выдан ЦГСЭН в СПб

Издательство «ЛАНЬ»  
lan@lanbook.ru; www.lanbook.com  
196105, Санкт-Петербург, пр. Юрия Гагарина, д. 1, лит. А  
Тел./факс: (812) 336-25-09, 412-92-72  
Бесплатный звонок по России: 8-800-700-40-71

Подписано в печать 12.01.21.  
Бумага офсетная. Гарнитура Школьная. Формат 70×100<sup>1/16</sup>.  
Печать офсетная. Усл. п. л. 15,28. Тираж 30 экз.

Заказ № 084-21.

Отпечатано в полном соответствии с качеством  
предоставленного оригинал-макета в АО «Т8 Издательские Технологии».  
109316, г. Москва, Волгоградский пр., д. 42, к. 5.

# ГДЕ КУПИТЬ

## ДЛЯ ОРГАНИЗАЦИЙ:

Для того, чтобы заказать необходимые Вам книги,  
достаточно обратиться в любую из торговых компаний  
Издательского Дома «ЛАНЬ»:

**по России и зарубежью**

**«ЛАНЬ-ТРЕЙД»**

РФ, 196105, Санкт-Петербург, пр. Ю. Гагарина, 1

тел.: (812) 412-85-78, 412-14-45, 412-85-82

тел./факс: (812) 412-54-93

e-mail: trade@lanbook.ru

ICQ: 446-869-967

**www.lanbook.com**

пункт меню «Где купить»

раздел «Прайс-листы, каталоги»

**в Москве и в Московской области**

**«ЛАНЬ-ПРЕСС»**

109387, Москва, ул. Летняя, д. 6

тел.: (499) 722-72-30, (495) 647-40-77

e-mail: lanpress@lanbook.ru

**в Краснодаре и в Краснодарском крае**

**«ЛАНЬ-ЮГ»**

350901, Краснодар, ул. Жлобы, д. 1/1

тел.: (861) 274-10-35

e-mail: lankrd98@mail.ru

## ДЛЯ РОЗНИЧНЫХ ПОКУПАТЕЛЕЙ:

интернет-магазин

Издательство «Лань»: <http://www.lanbook.com>

магазин электронных книг

**Global F5**

<http://globalf5.com/>

**Издательство**  
**«ЛАНЬ»**  ЛАНЬ®

**ЕСТЕСТВЕННОНАУЧНАЯ  
ЛИТЕРАТУРА  
ДЛЯ ВЫСШЕЙ ШКОЛЫ**

Мы издаем новые  
и ставшие классическими учебники  
и учебные пособия по общим  
и общепрофессиональным  
направлениям подготовки.

Большая часть литературы  
издательства «ЛАНЬ»  
рекомендована Министерством образования  
и науки РФ и используется вузами  
в качестве обязательной.

Мы активно сотрудничаем  
с представителями высшей школы,  
научно-методическими советами  
Министерства образования и науки РФ,  
УМО по различным направлениям  
и специальностям по вопросам грифования,  
рецензирования учебной литературы  
и формирования перспективных планов издательства.

**Наши адреса и телефоны:**

РФ, 196105, Санкт-Петербург, пр. Юрия Гагарина, 1  
(812) 336-25-09, 412-92-72  
[www.lanbook.com](http://www.lanbook.com)

**Издательство**  
**«ЛАНЬ» ЛАНЬ®**



Мы будем благодарны Вам  
за пожелания по издаваемой нами литературе,  
а также за предложения по изданию книг  
новых авторов или переизданию  
уже существующих трудов.

Мы заинтересованы в сотрудничестве  
с высшими учебными заведениями  
и открыты для Ваших предложений  
по улучшению нашего взаимодействия.

Теперь Вы можете звонить нам бесплатно  
из любых городов России по телефону

**8-800-700-40-71**

Дополнительную информацию  
и ответы на вопросы Вы также можете получить,  
обратившись по электронной почте:

**it@lanbook.ru**



# ЛАНЬ

ЭЛЕКТРОННО-  
БИБЛИОТЕЧНАЯ  
СИСТЕМА

E.LANBOOK.COM

## УЧЕБНАЯ ЛИТЕРАТУРА

от ведущих издательств —  
более 50 000 наименований

## БЫСТРЫЙ ПОИСК

на движке Elasticsearch

## МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ

доступ к контенту  
офлайн

## ВИДЕОИНСТРУКЦИИ

легко и понятно



## НАУЧНАЯ ПЕРИОДИКА

бесплатно 400 000  
научных статей

## ЛИЧНЫЙ КАБИНЕТ ЧИТАТЕЛЯ

простая и удобная  
регистрация, широкий  
функционал работы с текстом

## МОДУЛЬ РПД

удобный сервис  
по формированию  
списков литературы



[E.LANBOOK.COM](http://E.LANBOOK.COM)



[VK.COM/LALA.LANBOOK](https://vk.com/lala.lanbook)



[FACEBOOK.COM/LALA.LANBOOK](https://facebook.com/lala.lanbook)



[YOUTUBE.COM/EBSLAN](https://youtube.com/ebslan)

ISSN 978-5-8114-6611-9



9 785811 468119



# ЛАНЬ

ИЗДАТЕЛЬСТВО

OZON



1665831892