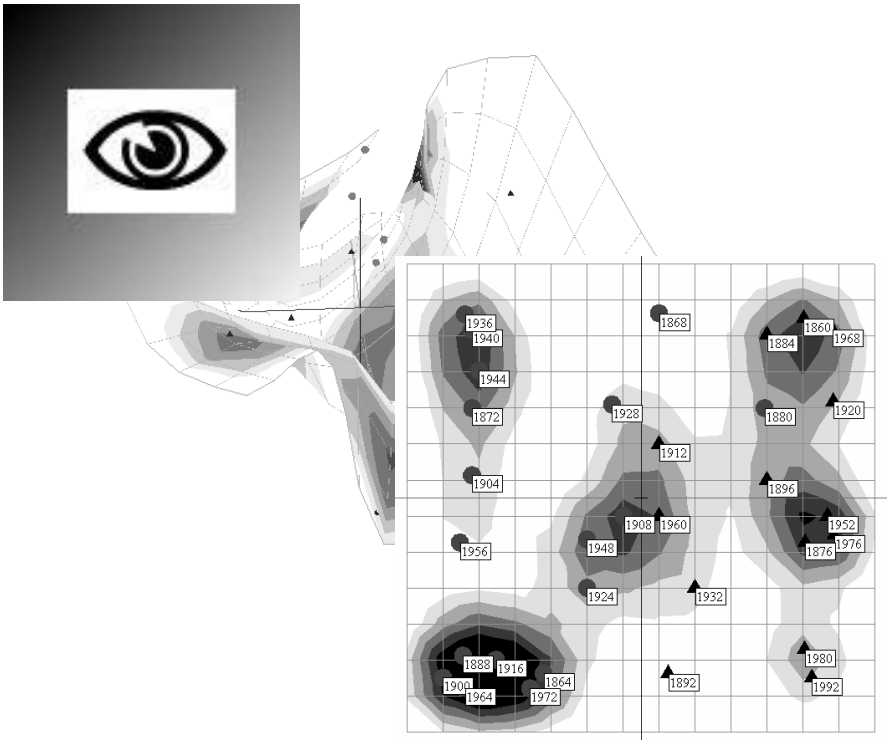
А.Ю.ЗИНОВЬЕВ

ВИЗУАЛИЗАЦИЯ МНОГОМЕРНЫХ ДАННЫХ



Предисловие

Визуализация данных – задача, с которой сталкивается в своей работе любой исследователь. К задаче визуализации данных сводится проблема представления в наглядной форме данных эксперимента или результатов теоретического исследования. Традиционные инструменты в этой области – графики и диаграммы – плохо справляются с задачей визуализации, когда возникает необходимость изобразить более трех взаимосвязанных величин.

С другой стороны, существует мощнейший инструмент изображения информации, привязанной к географической сетке координат. Это очень быстро развивающийся сегодня арсенал ГИС-технологий (ГИС – геоинформационные системы). К сожалению, как только исчезает подложка для изображения информационных слоев – географическая карта – все методы ГИС остаются не у дел.

В книге поставлена и до некоторой степени решена задача построения такой подложки для *произвольного* набора данных. С помощью нее можно визуализировать данные, одновременно нанося на подложку сопровождающую информацию (подписи, аннотации, атрибуты, информационные раскраски). Подложка, называемая *картой данных*, служит заменой географической карте там, где ее просто не существует. Принципиальное отличие в следующем: на географической карте соседние объекты обладают близкими географическими координатами, на карте данных близкие объекты обладают *близкими свойствами*.

Хорошей особенностью предлагаемых методов построения карт данных является то, что данные могут содержать *пробелы* – отсутствующие или недостоверные значения отдельных признаков. Такие точки данных также могут наноситься на карту.

Помимо роли подложки для нанесения информации, карта служит *информационной моделью* данных. Такая модель может решать важную задачу заполнения в данных пробелов. Эта способность может быть использована для *правдоподобного* прогнозирования поведения системы в задаваемых исследователем условиях.

Книга состоит из трех относительно независимых глав.

В *первой главе* изложение сделано максимально доступным. Здесь читатель не найдет ни одной формулы, зато обнаружит большое количество иллюстраций. Цель первой главы – дать читателю наглядное представление об используемых в данной области понятиях и приемах.

Вторая глава содержит математическое изложение алгоритмов и методов визуализации. Здесь уточняются понятия, введенные в первой

главе. Цель второй главы – снабдить читателя всем необходимым для того, чтобы начать самостоятельную творческую деятельность в области визуализации данных.

Третья глава начинается с краткого описания программы визуализации данных ViDa Expert, в создании которой я принимал непосредственное участие. С помощью этой программы было сделано большинство используемых в книге иллюстраций. После этого приведено несколько характерных примеров визуализации, с которыми я имел дело на практике.

В создании этой книги так или иначе принимало участие большое количество людей. Всем им я очень благодарен. Фактическим соавтором является мой наставник по науке и не только – доктор физ.-мат. наук, профессор Александр Николаевич Горбань. Без его живейшего участия вероятность написания этой книги была бы равна нулю. Я выражаю ему глубокую признательность за постановку задач, генерирование идей, за обсуждение текста во время встреч и на семинарах, а также за помощь в издании книги. Огромное спасибо моему коллеге Александру Питенко, с которым мы теснейшим образом сотрудничали во время работы практически по всем проектам. Вместе с Александром мы написали компьютерную программу ViDa Expert, в которой реализованы почти все предложенные в книге идеи. Я благодарен кандидату физ.-мат. наук Евгению Моисеевичу Миркесу – признанному авторитету в нейроинформатике, с чьей помощью книга сделана более читабельной и исправлено большое количество неточностей.

Андрей Зиновьев

СОДЕРЖАНИЕ

Стр.

Глава 1. Море информации и океан данных

1.1. Возникновение и представление данных

1.1.1. Таблицы данных.

Любое практическое исследование на начальном этапе включает в себя стадию *собирания данных*. Если исследователь разрабатывает совершенно необжитую область, то он склонен понимать под данными практически все, что ему удается зафиксировать более-менее содержательного и более-менее объективного в исследуемой системе (разумеется, содержательного и объективного с точки зрения его самой общей методологии).

Тем не менее, даже если первопроходец бережно коллекционирует все, что попадается под руку, он вынужден каким-либо образом систематизировать собранное. Наиболее распространен следующий подход.

Как правило, исследователь выделяет в системе объекты, сходные по природе и по своему усмотрению наделяет всю совокупность таких объектов набором свойств, с помощью которых он намеревается отличать одни объекты от других той же природы. В обозначении такого набора свойств или *признаков* объекта состоит первый шаг к тому, чтобы придать исследованию определенное направление, и после такого выбора исследователь уже способен отвечать на один существенный для поиска вопрос – *одинаковы ли два встреченных явления или они как-то отличаются друг от друга*?

В наши задачи ни в коей мере не входит оспаривать универсальность такого метода – более того, с другими мы работать и не будем. Главное – это то, что в результате возникает средство для представления и хранения собранных данных – *таблица данных*. В дальнейшем мы будем считать понятия данных и таблицы синонимами, считая, что все собранные материалы исследователь хранит в виде таблиц.

✂ Разумеется, таблица данных может содержать данные не об исследуемых отдельных объектах или явлениях, а данные о состоянии одного и того же объекта, но в разных ситуациях или в разные моменты времени. Тогда следует говорить о различии в состоянии одного и того же объекта. ✂

Будем представлять, согласно традициям и соображениям удобства, что каждой строке таблицы соответствует определенный объект или явление изучаемой системы, а в столбцах таблицы размещаются значения или метки признаков. В результате получается таблица типа «*объект-признак*»:

	Признак 1	Признак 2	Признак 3	Признак 4	…	Признак m-1	Признак m
Объект 1					…		
Объект 2					…		
Объект 3							
…	…	…	…	…	…	…	…
Объект *N*					. . .		

✂ Мы намеренно оставляем в стороне вопрос о том, насколько общо такое представление данных. Однако приведем пример, где отношения между понятиями объекта и признака объекта несколько запутываются. Простейшим примером является таблица результатов соревнований, проведенных по олимпийской системе «каждый играет с каждым». Или пример таблиц, встречающихся при учете миграционных потоков населения. В строках такой таблицы стоят страны, а в столбцах – те же страны, но играющие роль центров иммиграции. На пересечении соответствующей строки и столбца – число иммигрантов.

Одним из способов все-таки представить такие таблицы в виде «объект-признак» состоит в том, чтобы считать объектом не отдельную страну, а явление миграции из страны А в страну В. Один из признаков такого явления – количество

иммигрантов и т.п. Таблица в этом случае «вытянется» по вертикали и предстанет в не совсем привычном виде. ✂

Характерные примеры

Приведем несколько примеров и постараемся сделать небольшой обзор способов представить имеющиеся данные в виде таблиц.

✋ *Пример 1. Экономическая таблица.*

В журнале «Эксперт» ежегодно публикуются таблицы экономических показателей для двухсот самых крупных предприятий России. В числе таких показателей указываются валовый годовой доход, выраженный в рублях и долларах по среднегодовому курсу, темп роста предприятия, его балансовая прибыль до и после налогообложения, а также число работающих и некоторые производные характеристики, общепринятые в экономике, типа производительности предприятия. Кроме этого, указана территориальная и отраслевая принадлежность предприятия. Начальный фрагмент таблицы приведен на рис.1.

1999 г.	1998 г.	Компания	Регион	Отрасль	Объем реализации в 1998 г. (млн руб.)**	Темп роста (%)	Объем реализации в 1998 г. (млн долл.)***	Балансовая прибыль за 1998 г. (млн руб.)	Прибыль после налогообложения за 1998 г. (млн руб.)	Количество работающих за 1998 г. (тыс. чел.)	Производительность труда (тыс. руб./чел.)
1	1	РАО «ЕЭС России»*		Электроэнергетика	218802.1	2.0	22349.5	21534.3	16045.6	697.8	313.6
2	2	ОАО «Газпром»*		нефтяная и нефтегазовая промышленность	171295.0	23.4	17496.9	-22147.0	-30119.0	278.4	615.3
3	3	Нефтяная компания «ЛУКойл»*		нефтяная и нефтегазовая промышленность	81660.0	52.2	8341.2	2032.0	573.0	102.0	800.6
4		Башкирская топливная компания*	Башкирия	нефтяная и нефтегазовая промышленность	33081.8	–9.1	3379.1	1228.3	517.2	104.8	315.7
5	4	Сибирско-Дальневосточная нефтяная компания («Сиданко»)*****		нефтяная и нефтегазовая промышленность	31361.8	0.9	3203.5			80.0	392.0

Рис. 1. Таблица экономических показателей крупнейших предприятий России.

В дальнейшем мы познакомимся с этой таблицей поближе, а сейчас следует отметить некоторые характерные особенности данных, представленных в этой таблице:

- большая часть признаков таблицы измеряется числом – значением соответствующего показателя; все количественные признаки могут принимать любые вещественные значения в определенном диапазоне; мы можем сравнить два предприятия по значению того или иного признака (например, отметить, что темп роста ГазПрома больше, чем ЕЭС);
- данные *неполны* – значения некоторых признаков неизвестны или недостоверны по тем или иным причинам;

- один из признаков (валовый объем производства) для разных предприятий принимает значения, отличающиеся на порядки.
- некоторые признаки (например, принадлежность к отрасли) не являются числовыми по смыслу (их значения являются лишь метками);

Пример 2. Медицинская таблица

В Красноярске и не только в нем весьма популярной для апробации различных методов анализа данных является так называемая таблица осложнений инфаркта миокарда. Более подробно о ней будет сказано в третьей главе. Некоторые исследователи, пробовавшие свои силы на этой таблице, утверждают, что она содержит большинство характерных трудностей и подводных камней, которые встречаются при анализе таблиц реальных (не модельных) данных. Число признаков в полном варианте таблицы – 128, поэтому приведем на рис. 2 несколько строк таблицы с теми признаками, что вошли на экран компьютера.

	FIO	AGE	SEX	INF_ANAM	STENOK_AN	FK_STENOK	IBS_POST
1	Говязина Н.Г.	68	0	0	0	0	0
2	Казанцев В.К.	51	1	0	6	2	1
3	Викулов В.Л.	38	1	0	0	0	0
4	Быстров И.П.	55	1	0	0	0	0
5	Бояркина М.С.	69	0	0	0	0	2
6	Васильев В.Г.	57	1	0	0	0	0
7	Шелковников В.С.	51	1	0	0	0	0
8	Прохоров В.М.	63	1	0	0	0	0
9	Потылицин И.А.	71	1	0	0	0	2
10	Козлова Н.И.	45	0	0	1	2	1
11	Коношкин В.П.	50	1	0	0	0	2
12	Ростов Н.А.	53	1	0	0	0	0
13	Востриков В.С.	43	1	0	0	0	0

Рис. 2. Таблица осложнений инфаркта миокарда.

Как и в предыдущем случае отметим характерные черты такого набора данных.

- большая часть признаков – это бинарно закодированные ответы на вопросы, то есть единица соответствует ответу «да», ноль – «нет»; кроме этого, встречаются такие признаки, которые хоть и принимают целочисленные значения из определенного диапазона, но не имеет большого смысла сравнивать двух пациентов по величине таких

признаков, то есть само числовое значение – это тоже всего лишь метка ответа на вопрос (когда ответов «да» и «нет» - недостаточно).

✋ *Пример 3. Данные мониторинга*

Исследователь может по сути иметь дело с одним и тем же объектом, наблюдая его различные состояния. Весьма популярным объектом наблюдения является биржа ценных бумаг. Его состояние может быть охарактеризовано несколькими десятками различных параметров – финансовых индикаторов, которые изменяются ежедневно или даже ежеминутно.

На рис. 3 изображена таблица, где показано несколько состояний фондового рынка США, каждое из которых характеризуется значением и последним изменением шести основных финансовых индикаторов. Более подробная таблица включает значения нескольких сотен индексов.

Date	Time	Dow 30 Industrials		S&P 500 Index		Nasdaq Composite		AMEX Composite		Nyse Composite	
		Value	Change	Value	Change	Value	Change	Value	Change	Value	Change
21.07.2000	6.50PM	10733.56	-110.31	1480.19	-15.38	4094.45	-90.11	927.64	-12.14	655.38	-4.81
22.07.2000	6.50PM	10628.51	-105.05	1470.16	-10.03	4014.14	-80.31	917.58	-10.06	650.35	-5.03
23.07.2000	6.50PM	10514.29	-114.22	1465.01	-5.15	4004.02	-10.12	918.62	+1.04	648.15	-2.20

Рис. 3. Таблица основных фондовых индексов США.

✋ *Пример 4. Частотный анализ текстовой базы данных*

В случае, если предметом исследования является некоторая совокупность текстов (например, все статьи, опубликованные в журнале за десять лет), то содержание (естественно, не смысловое, а формальное) этих текстов можно представить в виде частотной таблицы.

Для составления такой таблицы сначала проводится полный частотный анализ всей текстовой базы и находятся наиболее часто употребляемые слова (как правило, при частотном анализе игнорируют различные варианты написания слов, то есть их окончания и т.д., а также выбрасывают заведомо бессодержательные, но часто употребляемые слова-связки). В результате составляется словарь из некоторого фиксированного набора наиболее часто употребляемых слов во всей совокупности текстов. Этот словарь и играет роль набора признаков, характеризующих каждый отдельный текст из базы. Каждый признак – это

отдельное слово из словаря, его значение для конкретного текста – число, описывающее сколько раз данное слово было встречено в тексте.

На рис.4 приводится простая частотная таблица, где в качестве объектов выбраны разделы этой книги, которые описываются частотами некоторых самых распространенных в тексте книги слов.

В качестве характерной особенности данной таблицы укажем следующее:

- если текстовая база очень неоднородна по содержанию (например, база из всех статей UseNet), то для того, чтобы можно было охватить все темы, частотный словарь должен содержать достаточно много словоформ; необходимое число столбцов в частотной таблице может вырасти до тысячи; при этом сама частотная таблица окажется очень разреженной – будет содержать большое количество нулей.

Текст	данн	точк	карт	модел	сетк	табл	визуал
Миркес Е.М. Нейрокомпьютер. Проект стандарта.	0.006558	0.000822	0.000249	0.000000	0.014876	0.003212	0.000000
Горбань А.Н. Демон Дарвина	0.001367	0.000481	0.000751	0.009319	0.000154	0.000058	0.000019
Визуализация данных. Глава 1	0.025356	0.003356	0.010938	0.007644	0.003356	0.004164	0.001678
Визуализация данных. Глава 2	0.017673	0.004564	0.004467	0.004273	0.011750	0.000777	0.001457
Визуализация данных. Глава 3	0.020914	0.000775	0.018203	0.000387	0.003098	0.009295	0.003486

Рис. 4. Частотный анализ содержания глав этой книги
на фоне двух других книг.

Предложенный способ составления частотных таблиц достаточно широко применяется для автоматизированного составления каталога текстовых баз. Познакомиться с применениями такого подхода можно в Интернете (http://websom.hut.fi) и в работах [7,22,45,77].

✋ *Пример 5. Прогнозирование поведения временного ряда*

Предположим, что результатом некоторых наблюдений является временной ряд – информация о состоянии какого-то явления (например, курса доллара на торгах ММВБ) в разные моменты времени. Можно поставить задачу прогнозирования поведения временного ряда, то есть предсказания значения каких-то величин в будущие моменты времени. В этом направлении существуют два подхода.

В первом предполагается, что значение величины зависит главным образом от некоторых сторонних факторов и задача предсказания в этом случае сводится к выявлению зависимости прогнозируемой величины от

других факторов. Для такого подхода удобно представлять временной ряд в естественном виде, то есть выбирать в качестве признаков время наблюдения, численное значение прогнозируемой величины, значения остальных факторов, предположительно имеющих отношение к делу.

Второй подход предполагает, что значение какой-либо величины можно предсказать, если знать ее поведение в прошлом. В этом случае изучаемый объект – это факт того, что прогнозируемая величина приняла определенное значение вместе с определенной предысторией изменения величины в прошлом.

Рассмотрим в качестве конкретного примера, как можно преобразовать простую таблицу изменения курса доллара для применения последнего из упомянутых подходов. В качестве признаков выберем значение самого курса, а также значения курса за последние *n* дней. Фрагмент такой таблицы приведен на рис. 5.

- таблица имеет характерный вид: значения признаков смещаются в каждой последующей строке на одну позицию вправо.

	N1	N2	N3	N4	N5	N6	N7	N8
1745	11-11-97	5.898	5.89	5.89	5.89	5.89	5.89	5.889
1746	12-11-97	5.899	5.898	5.89	5.89	5.89	5.89	5.89
1747	13-11-97	5.9005	5.899	5.898	5.89	5.89	5.89	5.89
1748	14-11-97	5.9005	5.9005	5.899	5.898	5.89	5.89	5.89
1749	15-11-97	5.9015	5.9005	5.9005	5.899	5.898	5.89	5.89
1750	16-11-97	5.9015	5.9015	5.9005	5.9005	5.899	5.898	5.89
1751	17-11-97	5.9015	5.9015	5.9015	5.9005	5.9005	5.899	5.898
1752	18-11-97	5.903	5.9015	5.9015	5.9015	5.9005	5.9005	5.899
1753	19-11-97	5.905	5.903	5.9015	5.9015	5.9015	5.9005	5.900
1754	20-11-97	5.9065	5.905	5.903	5.9015	5.9015	5.9015	5.900
1755	21-11-97	5.9085	5.9065	5.905	5.903	5.9015	5.9015	5.901
1756	22-11-97	5.9105	5.9085	5.9065	5.905	5.903	5.9015	5.901
1757	23-11-97	5.9105	5.9105	5.9085	5.9065	5.905	5.903	5.901
1758	24-11-97	5.9105	5.9105	5.9105	5.9085	5.9065	5.905	5.903
1759	25-11-97	5.912	5.9105	5.9105	5.9105	5.9085	5.9065	5.905
1760	26-11-97	5.914	5.912	5.9105	5.9105	5.9105	5.9085	5.906
1761	27-11-97	5.916	5.914	5.912	5.9105	5.9105	5.9105	5.908
1762	28-11-97	5.917	5.916	5.914	5.912	5.9105	5.9105	5.910
1763	29-11-97	5.919	5.917	5.916	5.914	5.912	5.9105	5.910

Рис. 5. Таблица изменений курса доллара.

Зоология шкал признаков

В вышеописанных примерах объекты исследования описывались признаками, которые отличались друг от друга допустимыми наборами значений. Опишем различные типы *шкал признаков* согласно общепринятым определениям:

- *непрерывная* шкала – признак в этой шкале может принимать любое вещественное значение; разумеется, некоторые признаки могут принимать, например, только положительные значения, то есть лежать в определенном допустимом *диапазоне*;
- *дискретные* шкалы – применяются в том случае, если признак не является по смыслу задачи вещественным числом; здесь есть два существенно разных варианта:

  ♦ *номинальные* шкалы – применяются, если целое число не является выражением какой-либо меры, а служит просто меткой варианта ответа на вопрос; в случае если допустимыми вариантами ответа являются только «да» и «нет», шкала называется *бинарной* и признак принимает значение 1 или 0 ;

  ♦ *порядковые* или *ординальные* шкалы – применяются, если целое число отражает степень проявления определенного качества (например, степень уверенности в ответе); порядковая шкала может изменяться а) от одной противоположности до другой и тогда допустимые значения располагаются симметрично относительно нуля – точки неопределенности; б) от точки отсутствия качества до точки наивысшего его проявления – и тогда естественно придавать признаку только положительные значения.

  ✂ относительно номинальных признаков можно сделать следующее замечание – в литературе [1,19,34] встречаются рекомендации разбивать любую номинальную шкалу на несколько бинарных, в соответствии с присутствием или отсутствием какого-либо варианта ответа ✂

Исследователь должен прежде всего четко представлять в каких шкалах измеряются те или иные признаки. Тип шкалы может быть важен для ответа на вопрос исследователя: *если два явления отличаются, то насколько сильно*?

Выбор шкалы осуществляется не формальным образом, а только по смыслу задачи. В примере с экономической таблицей число работающих на предприятии может измеряться целым числом (если выбрать в качестве

единицы измерения точное число людей), однако признак по смыслу все равно измеряется в непрерывной шкале (так, если измерять в тысячах, то признак начнет принимать дробные значения). С другой стороны, можно превратить этот признак в ординальный, если разбить все предприятия по категориям относительно числа работников (0 - малое, 1 - среднее, 2 - крупное, 3 - сверхкрупное предприятие и т.п.).

Некоторые предварительные выводы

Итак, предметом нашего исследования будут прежде всего таблицы данных, полученных в результате наблюдения за изучаемой системой объектов или явлением. Еще раз отметим, что само по себе построение таблицы предполагает, что исследованию задано определенное направление, а об области исследования имеется какое-то предварительное представление. Таким образом, уже на этапе собирания данных делается первый шаг к абстрагированию от конкретной действительности, когда из бесконечного числа способов описывать объекты исследования выбирается один, характеризуемый выбором набора признаков, с помощью которого объекты отделяются друг от друга.

За собиранием данных следует их анализ, конечная цель которого – извлечение определенного рода *информации*, или, более общо, *знания* из таблицы.

Как видно из примеров, на практике в таблицах типа «объект-признак» число объектов (строк таблицы) обычно измеряется тысячами, а число признаков (столбцов таблицы) – сотнями. Естественно, что восприятие такого массива данных весьма затруднено, а следовательно, затруднен и анализ. Графики или диаграммы способны наглядно показать отношения лишь между двумя-тремя признаками, оставляя остальные количественные характеристики за пределами внимания исследователя. Таким образом, чем более признаков содержит таблица, тем, с одной стороны полнее описываются объекты исследования, а с другой – тем труднее извлекать из таблицы необходимую информацию.

Изложенные далее методы и идеи будут направлены главным образом на то, чтобы создать у исследователя некоторый целостный наглядный образ данных, с помощью которого он мог бы ориентироваться в бесконечных полях чисел, собранных в больших таблицах.

Более того, в связи с основной темой изложения нам будут неинтересны таблицы с числом столбцов менее четырех, поскольку для таких таблиц вполне эффективно могут применяться традиционные методы представления данных.

1.1.2. Представление данных в абстрактном виде.

Первым шагом на пути к созданию наглядного образа данных является представление объектов (строк таблицы) в виде геометрических образов. При этом следует учесть несколько возможных обстоятельств:

- значения признаков, как правило, известны не абсолютно достоверно, а с некоторой конечной *точностью*; разумеется, это замечание уместно в случае применения непрерывных шкал признаков;
- для значений некоторых признаков могут допускаться определенные отклонения, величины которых устанавливаются здравым смыслом и задачами исследования; в этом случае говорят, что данные измерены с определенным *допуском*;

  ✂ не следует смешивать понятия допуска и точности; точность – это отклонение от некоторого «истинного» значения, определяемое возможностями измерительной аппаратуры, методикой эксперимента и т.д., то есть внешними объективными причинами; допуск – это такое отклонение, которым исследователь намеренно пренебрегает, поскольку оно не играет значимой роли в его задаче, в пределах этого отклонения исследователь считает любые два значения признака совпадающими; естественно, что значение допуска, вообще говоря, не может быть меньше точности ✂

- информация об отдельных объектах может быть известна не полностью, в этом случае говорят о данных, содержащих *пробелы*; при этом, как правило, на недостающие или недостоверные значения признаков можно наложить некоторые априорные ограничения.

Упомянутые обстоятельства приводят к тому, что отдельный объект из исследуемой совокупности данных может быть представлен с помощью одного из следующих геометрических образов в некотором абстрактном пространстве R^m (m – число признаков объекта):

точкой – в случае если объект характеризуется набором дискретных признаков, или считается, что все его признаки известны с абсолютной точностью; координатами точки являются значения соответствующих признаков;

m-мерной *сферой* или *эллипсоидом* – если задается погрешность (или допустимое отклонение) положения объекта в абстрактном пространстве данных относительно гипотетического точного положения;

m-мерным *параллелепипедом* – если погрешность (или допуск) задаются отдельно для каждого из признаков;

отрезком прямой, параллельной одной из координатных осей – если один из признаков неизвестен, а остальные известны точно; длина отрезка отражает априорный допустимый диапазон, в котором может находиться пропущенное значение;

куском *k*-мерной *плоскости*, если пропущены *k* значений из набора признаков объекта;

куском *k*-мерного *слоя* некоторой толщины – если *k* значений признаков в наборе пропущены, а остальные известны с некоторой точностью, при этом толщина слоя отражает значение точности (или допуска).

После сопоставления каждому из объектов геометрического образа в абстрактном многомерном пространстве данных возникает облако из геометрических объектов, которое и отражает структуру исследуемого набора данных. Изучая это облако, мы, тем самым будем изучать сами данные.

Однако, для того, чтобы говорить о геометрических отношениях внутри облака данных, следует решить важный вопрос о выборе подходящей *метрики* для пространства. Не определившись в этом вопросе, исследователь не может ответить на вопрос о том*, насколько сильно по своим свойствам один объект отличается от другого*. От удачного выбора метрики часто зависит насколько геометрическая метафора данных соответствует структуре самих данных.

Во второй главе нами будут рассмотрены конкретные формулы для расчета расстояния между двумя объектами для некоторых наиболее распространенных вариантов выбора метрики. Здесь же ограничимся следующими качественными замечаниями:

- достаточно очевидно требование того, чтобы расстояние между объектами не зависело от того, меряем ли мы его от первого объекта до второго или от второго к первому;
- для правила вычисления расстояний должно выполняться «неравенство треугольника» (рис. 6), смысл которого состоит в том, что расстояние между двумя объектами должно вычисляться в каком-то смысле по кратчайшей линии;

- в случае если признаки объекта принимают дискретные наборы значений, правило вычисления расстояния должно быть соответствующим образом модифицировано для того, чтобы лучше соответствовать специфике таких шкал признаков – так для порядковых признаков часто применяется так называемая городская метрика, а для бинарных – расстояние Хэмминга (рис. 7а и 7б)

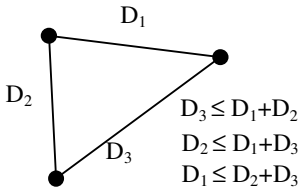


Рис. 6. Неравенства треугольника

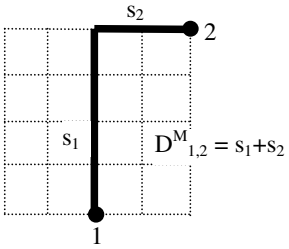


Рис. 7а. Городская метрика

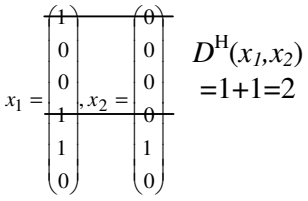


Рис. 7б. Расстояние Хэмминга

1.2. Игры с данными. Обряды и ритуалы.

1.2.1. Почему все так несерьезно?

Почему игры, обряды и ритуалы, а не просто методы анализа? Речь пойдет о достаточно серьезных вещах, и о вполне работающих методах работы с данными. Однако за пределами любой методологии лежит опыт исследователя, который трудно или невозможно формализовать. Прочитав учебник по математической статистике, можно уяснить себе методологические основы анализа данных, но на практике всегда окажется, что кроме того, что «надо делать, потому что это оправданно» есть еще то, что «обычно делают, не особенно задумываясь». Словами психологов – «на каждый полезный совет нужно еще тысяча о том, как его выполнить».

Это и составляет основу «обрядов» обращения с данными. Обряд – это одновременно действия, оправданные с точки зрения строгой методологии и то, что исследователь делает, не особенно задаваясь строгим доказательством необходимости своих действий. Обряд состоит из ряда «ритуалов» (один из ритуалов – обращение к строгим методическим принципам и терминам), множество обрядов образуют формальные и неформальные правила игры с данными.

Ситуация эта не является специфичной для прикладной статистики. В физике, например, практически невозможно формализовать способы

соотнесения реальному объекту его теоретической модели, в чем обычно состоит сложность изучающих физику – общие принципы известны, но как «вставить» в уравнения данное реальное явление – все это остается за рамками. Для того, чтобы набраться неформального опыта учащийся решает большое число стандартных задач, с помощью которых он уясняет как можно воплотить те или иные теоретические принципы.

Приведем пример. Что делает исследователь в первую очередь, если в руки ему попадает некоторый набор точек данных? Ответ практически очевиден – считает среднее арифметическое координат этих точек и дисперсию – разброс около среднего (либо, более общо, ковариационную матрицу). Тем самым вычисляются достаточные статистики нормального распределения. Исследователь, как правило, осознает, что распределение реальных данных может оказаться далеким от нормального и среднее значение точек облака может находиться вовсе вне области скопления данных, но делает это отчасти потому, что «так положено», отчасти чтобы представить себе данные «в первом приближении». Исследователь может оправдать себя тем, что, как и в физике, с точки зрения «удаленного наблюдателя», облако данных выглядит как скопление точек около их среднего значения. Они примерно занимают область, соответствующую по объему шару с радиусом, равным приблизительно квадратному корню из дисперсии (точки, выпадающие из этой области будут практически незаметны для удаленного наблюдателя). В дальнейшем исследователь может отбросить гипотезу о нормальном распределении как нереалистичную, но, тем не менее, исследования данных начинаются с нее. Потому что таково предписание ритуала, имеющего свою историю и основание.

Другой пример: дана таблица чисел 3x3 с одним пропущенным значением: $\begin{vmatrix} 1 & 1 & 1 \\ 1 & ? & 1 \\ 1 & 1 & 1 \end{vmatrix}$. Задача: дать оценку пропущенного значения (задачи такого типа составляют большую часть тестов на определение IQ – коэффициента умственного развития). Первое, почти подсознательное, предположение – оценка равна единице. Разумеется, что это предположение не выдерживает никакой критики с точки зрения более или менее строгой методологии. Для статистического доказательства этой гипотезы данных явно не достаточно. Однако, это значение является правдоподобным. Поэтому, если нет никаких дополнительных соображений, но пропущенное значение необходимо как-то восстановить, то разумно предположить, что оценка равна не нулю, и не тысяче, а именно единице, что тоже составляет часть ритуала. С другой стороны, это значение вовсе не является необходимым.

Итак, применение каждого метода на практике сопровождается определенным ритуалом, включающего действия с данными, необходимость (и достаточность) которых невозможно строго вывести из самого метода. Важную часть ритуала составляют «заклинания» – устойчивые словесные формулировки, которыми сопровождается ритуал.

Заклинания можно разделить на рекламные («этот метод делает что-то, что не делают другие») и технологические («мы сделали это, потому что это вытекает отсюда»). Рекламные заклинания, вроде «нейросети могут все», «метод главных компонент максимально сохраняет структуру расстояния между точками данных» или «самоорганизующиеся карты Кохонена сохраняют топологические особенности набора данных» являются эффективным средством выделить метод среди аналогичных и продвигать его на конференциях, давая обещания его потенциальным пользователям. От удачной формулировки рекламного заклинания может зависеть судьба метода. В рекламных заклинаниях, как правило, нет прямого обмана, но к таким заклинаниям всегда следует относиться осторожно, поскольку выполнение обещания всегда сопровождается определенными оговорками. Лучше быть заранее в курсе условий применения метода, чем выяснять их самостоятельно, на своем «печальном опыте».

Технологические заклинания призваны создать у слушателя или читателя впечатление того, что исследователь на своем пути строго следует логически оправданным методологическим установкам. Хотя, как правило, реальный путь исследования оказывается слишком извилистым, чтобы рассказывать о нем в подробностях. Технологические заклинания полезны, потому что они «спрямляют» этот путь, но и к ним необходимо относиться с осторожностью. Во-первых, существенная часть таких заклинаний делается «задним числом» (раз это сработало – значит это верно), а во-вторых, они могут скрадывать некоторые существенные детали исследования. Характерная ситуация в статистике, когда без технологических заклинаний не обойтись: это обработка «аномальных наблюдений» – объектов из набора данных, которые ни каким образом не укладываются в построенную модель. Такие данные либо уточняются, либо «ремонтируются» (тоже по определенному ритуалу со своими заклинаниями), либо просто уничтожаются (по ритуалу вывода из игры «недостоверных» наблюдений). Неосторожное обращение с технологическими заклинаниями может легко подорвать у пользователя доверие к методу. Наверняка именно оно стало причиной появления мнения о том, что «есть три вида лжи: ложь, наглая ложь и статистика». Если исследователь не желает, чтобы его метод заработал против него

самого, он должен с величайшей аккуратностью формулировать технологические заклинания.

Мы не претендуем на то, что в этой книге читатель найдет ясное и подробное описание всего набора ритуалов, необходимых для применения изложенных методов, и тексты заклинаний, хотя при желании он может извлечь и то, и другое. Цель нескольких последних абзацев – предупредить неискушенного читателя, с чем он неизбежно столкнется на практике, чего необходимо опасаться. Читатель предупрежден – и мы следуем дальше.

1.2.2. О происхождении данных[1].

Можно сказать, что в выбранной системе признаков таблица данных содержит описание объектов или функционирования системы с максимально возможной полнотой. Тем не менее, исследователю требуется не столько полное знание всего массива информации, сколько определенного рода «сухой остаток». Для того, чтобы делать выводы, он должен иметь возможность как то представить себе данные: что в них более, а что менее существенно, какие тенденции в них присутствуют. Тогда он будет способен содержательно о них рассуждать, сравнивать с другими системами объектов (возможно, отыскивая при этом полезные аналогии), делать правдоподобные предсказания о возможных качествах новых объектов, которые могут появиться в системе.

Таблицу данных можно воспринимать как прямое описание фрагмента действительности – «такой, как она есть». Для того, чтобы извлечь пользу из этого описания (например, создать математическую теорию этого фрагмента), исследователь должен использовать метод представления действительности в форме, удобной для оперирования и осмысления. Такой метод, давно закрепившийся в естественных науках, называется *моделированием.* Создание моделей действительности служит обязательным промежуточным звеном в связывании теории с положением вещей «как оно есть». Абстрактная теория не может описывать действительность напрямую, она лишь указывает на определенные отношения между абстрактными объектами в абстрактном мире (например, теория утверждает, что коммутатор двух определенных матриц всегда равен числу). В модели реализуется способ интерпретации теории, то есть способ сопоставления реальным объектам абстрактных. Одна и та же теория может рождать разные модели, относящиеся к совершенно

---

[1] В заголовок раздела вынесено перефразированное название работы Ч. Дарвина «О происхождении видов». Определенная аналогия здесь уместна, тем более что часть материала взята из книги А.Н. Горбаня «Демон Дарвина» [8].

различным фрагментам действительности. Более того, одна и та же теория может рождать разные модели одного и того же объекта.

Таким образом, модель – промежуточный шаг на пути от прямого описания действительности к ее абстрактному описанию. При решении поставленной задачи исследователь неоднократно проходит этот путь как в одном, так и в другом направлении.

Из всего множества функций, которые могут выполнять модели, можно выделить две основных. Во-первых, модель служит удобной заменой объектов действительности, своеобразным «аккумулятором знаний» об объекте. С помощью модели можно имитировать функционирование системы и прогнозировать ее поведение в эмпирически недоступных условиях, то есть так или иначе расширять полученный опыт. Во-вторых, модели могут играть смыслообразующую роль, то есть давать толчок к рождению новых смыслов, понятий, терминов в системе научного знания, которые могут быть использованы для качественно нового описания изучаемого фрагмента действительности.

Какое место занимают *модели данных* во всем многообразии встречающихся в науке моделей? В [8] изложена классификация моделей по 8-ми типам. Она, в свою очередь, основана на предложенной Р.Пайерслом классификации моделей в современной физике. Будем считать, что аналогичное положение вещей наблюдается во всех областях естественнонаучного знания.

Перечислим предложенные восемь типов моделей. К каждому из названий прилагается краткая характеристика, описывающая методологическую позицию исследователя (заметим, что эти характеристики – пример своего рода технологических заклинаний).

1. Гипотеза (такое могло бы быть).
2. Феноменологическая модель (ведем себя, как если бы).
3. Приближение (что-то считаем очень малым или очень большим).
4. Упрощение (опустим для ясности некоторые детали).
5. Эвристическая модель (количественного подтверждения нет, но модель способствует более глубокому проникновению в суть дела).
6. Аналогия (учтем только некоторые особенности).
7. Мысленный эксперимент (главное состоит в опровержении возможности).
8. Демонстрация возможности (главное – показать внутреннюю непротиворечивость возможности).

Теперь покажем какие модели из этого «зоопарка» в основном населяют мир прикладной статистики. Большая часть учебников заполнена моделями первого типа. При этом существуют два основных варианта:

1. *Модели, основанные на гипотезе о статистическом происхождении данных.*

Эта гипотеза подразумевает, что набор данных является выборкой из бесконечной *генеральной совокупности* объектов, чье распределение подчиняется определенному вероятностному закону. Более того, эта выборка должна быть сформирована *независимым случайным* выбором объектов генеральной совокупности. Принятие этой гипотезы (в реальных ситуациях весьма сильной) дает возможность со спокойной совестью применять к данным теоретико-вероятностные подходы, то есть фактически моделью такого набора данных является вся генеральная совокупность (данные дополняются до бесконечного числа объектов) с ее законом распределения.

Реальная практика эксплуатации этой гипотезы, как правило, приводит к тому, что исследователю приходится прибегать к определенного рода лукавству. Дело в том, что для того, чтобы иметь по-настоящему «спокойную совесть», корректность принятия (с хорошей достоверностью) гипотезы требуется строго доказать (по определенному ритуалу, который подробно описан в разделе математической статистики о проверке статистических гипотез). «Проклятием» большинства статистических исследований (в медицине, биологии, экономике, гидрологии и пр.) является то обстоятельство, что строгого доказательства провести не удается (как правило, просто недостаточно данных), и степень достоверности гипотезы может вызывать сомнение. Но поскольку в руках исследователя часто просто не оказывается других инструментов, он вынужден все равно принимать малодостоверную гипотезу. Это приводит его к созданию модели типа 2 (феноменологическая модель), которая отличается от первой по сути лишь разной степенью достоверности.

2. *Модели, основанные на гипотезе о порождении данных динамическим законом*.

Согласно общей установке законы природы делятся на *динамические* и *статистические*. Считается, что первые выполняются со всей необходимостью (содержат детерминированные правила), вторые выполняются лишь «в среднем». Большинство законов в природе не являются ни чисто динамическими, ни чисто статистическими. «Лучшими» в свое классе представителями динамических законов можно назвать законы небесной механики (именно поэтому положения космических объектов в Солнечной системе могут быть вычислены с большой точностью и практически из первых принципов), для

статистических законов в качестве хорошего примера можно указать на закон распределения вероятности появления электрона в квантовомеханической модели атома водорода.

Можно сделать предположение о том, что данные имеют не статистическую природу, а получены как результат детерминированного функционирования определенной системы, но, возможно, с наложением различного рода флуктуаций, которые, в свою очередь, могут быть описаны статистическими законами. Выбор конкретного вида динамического закона осуществляется исходя из априорных соображений, положений других теорий, интуиции исследователя и т.п. Этот закон может не носить на себе никаких следов физического осмысления механизмов системы (например, исследователь решает: данные распределены «по закону параболы» при наличии шума, который имеет нормальное распределение с дисперсией, которую можно оценить из имеющихся данных).

Зачастую, принимая гипотезу о динамическом законе, исследователь не в состоянии даже оценить ее достоверность. Поэтому почти все такие модели оказываются ближе ко второму типу (феноменологическая модель).

Итак, существенная часть моделей прикладной статистике основана на тех или иных гипотезах о природе происхождения данных: динамической и статистической. И та, и другая гипотеза заставляет исследователя догадываться о том, что стоит за данными, что их породило, и каковы свойства этого «нечто». В результате исследователь привязывает свои данные к тем существующим модельным системам, свойства которых ему лучше всего известны. Можно ли без этого обойтись?

Альтернативой производству гипотез является описание данных «как есть». В принципе, исследователь может представлять себе, что данные порождены некоторой системой (иногда эту систему можно даже «пощупать руками»: например, контейнер для хранения промышленных отходов, как в статье [44]), но он исходит из того, что внутреннее устройство этой системы ему неизвестно из-за высокой сложности и многокомпонентности. Ему проще описывать сами данные, отражающие работу системы в тех или иных условиях, чем создавать ее динамическую модель. Такой тип моделей может быть назван *информационными*.

Основным принципом информационного моделирования является принцип «черного ящика» – моделируется не внутреннее, а внешнее функционирование системы. Такие модели по общей классификации могут быть отнесены к 4 типу (упрощение). При этом упрощение необходимо понимать следующим образом: исследователь понимает при создании модели, что реальная система качественно более сложна, чем любая

известная теоретическая модель, и для того, чтобы как-то ухватить это качество сложности, он описывает систему так, как она проявляет себя для внешнего наблюдателя. В каком-то смысле исследователь соглашается с тем, что набор данных о системе и сама система эквивалентны, упрощая при этом, разумеется, реальное положение дел.

В способе построения информационных моделей есть свои преимущества и недостатки. Несомненным преимуществом является принципиальная возможность моделирования (причем не наукоемкого моделирования) сколь угодно сложных систем. Недостатками являются низкая «объяснимость» результатов, выдаваемых моделью, и привязка модели к конкретной системе (часто бывает, что опыт, накопленный на одном объекте, в выбранной системе признаков-свойств будет неадекватен опыту, накопленному на другом, вполне аналогичном объекте).

✂ Упоминания заслуживает пригодность информационных моделей для количественно точных оценок прогнозируемых явлений. Изначально, информационная модель строится именно с целью точного или почти точного описания действительности, то есть задаваемого набора данных. Вместе с тем информационная модель должна обладать предсказательными (обобщающими) способностями для того, чтобы иметь возможность правдоподобно прогнозировать свойства новых, не участвовавших в настройке модели объектов. Отдельным направлением являются методы автоматического извлечения знаний из информационных моделей. Модели, подвергнутые подобным процедурам, могут быть отнесены в разряд эвристических моделей (тип 5). Они могут обладать худшими способностями к количественным предсказаниям за счет увеличения «объяснимости» получаемых результатов. Подробнее об этом можно познакомиться в [48] ✂

В упомянутой работе [44] предлагается следующая типология информационных моделей по их предназначению:

- Моделирование отклика системы на внешнее воздействие
- Классификация внутренних состояний системы
- Прогноз динамического изменения системы
- Оценка полноты описания системы и сравнительная информационная значимость параметров системы
- Оптимизация параметров системы по отношению к заданной целевой функции
- Адаптивное управление системой

Предлагаемые в этой книге способы построения информационных моделей могут применяться для любой из поставленных выше целей – достаточно лишь разработать соответствующий ритуал на базе предлагаемого метода. Однако, главной целью создания двумерных информационных моделей является

- Визуализация многомерной структуры данных

Преследуя эту цель, мы, с одной стороны, ограничиваем метод в точности (хотя нами предлагаются и способы практически неограниченного увеличения точности описания), с другой – даем возможность исследователю создать себе наглядный образ набора данных, с помощью которого он сможет анализировать их структуру, практически не прибегая к сторонним методам.

Итак, нашей основной целью является создание двумерных информационных моделей. Но прежде всего дадим краткий обзор традиционных методов анализа данных, классифицируя эти методы по тем вопросам, на которые они могут дать содержательный ответ.

1.2.3. Вопросы которые мы ставим, глядя на данные…

Как уже упоминалось, основная цель анализа данных – извлечение из них информации. То, в каком виде будет представлена извлеченная информация – зависит главным образом от двух обстоятельств – а) от задач и целей исследования; б) от характера и качества тех процедур извлечения информации, которые исследователь имеет в своем распоряжении.

✂ В отличие от неопределенного, интуитивно постигаемого понятия данных, термин *информация, содержащаяся в данных* определен несколько точнее. Существует два основных подхода к понятию информации.

Первый из них связан с именем К.Шеннона и лежит в основе раздела кибернетики – теории информации. Согласно этому подходу, количество информации содержащееся в одном случайном объекте (событии, величине) относительно другого случайного объекта может быть измерено положительным числом. Пусть ξ – случайная величина, принимающая значения $x_1, x_2 \dots x_n$ с вероятностями $p_1, p_2 \dots p_n$, а η – случайная величина, принимающая значения $y_1, y_2 \dots y_n$ с

вероятностями q_1, q_2 … q_n. Тогда количество информации, содержащееся в ξ относительно η равно $I(\xi,\eta)=\sum_{i,j} p_{ij}\log_2(p_{ij}/p_i q_j)$, p_{ij} – вероятность совмещения событий $\xi = x_i$ и $\eta = y_j$. В случае, если ξ и η – величины независимые, то $I(\xi,\eta) = 0$. Имеет место неравенство $I(\xi,\eta) \le I(\eta,\eta)$, причем равенство достигается только в случае, если η является точной функцией от ξ (например, $\eta = \xi^2$).

С понятием информации тесно связано понятие *энтропии* случайной величины. Так энтропия ξ, по определению равна $H(\xi)=I(\xi,\xi)=\sum_j p_j\log_2(1/p_j)$. Смысл энтропии – среднее число двоичных знаков, необходимое для различения (или записи, кодирования) возможных значений случайной величины.

Второй подход к понятию информации существует в математической статистике, в теории статистических оценок, и предложен Р.Фишером. В данном подходе вводится понятие *достаточных статистик* – набора функций от данных, с помощью которых можно полностью восстановить характер исходного распределения данных в пространстве. Так, для нормального распределения достаточными статистиками являются среднее арифметическое и выборочная дисперсия. В статистике говорят, что знание этих величин дает полную информацию о распределении данных (то есть является их лаконичным описанием). Соответственно, знание не полного набора достаточных статистик дает частичную (неполную) информацию о распределении данных, и количество этой информации может быть выражено некоторой мерой. Напомним, что для статистических подходов существенно принятие *статистической гипотезы* о наличии *генеральной совокупности,* подчиненной определенному *статистическому закону*. ✂

Разумно предположить, что интуитивно под информацией исследователь подразумевает некоторое описание данных, которое по своей длине, по крайней мере, не превосходит простое перечисление тех значений, которое принимают признаки объектов. То есть в качестве конечного результата применения технологий извлечения информации исследователь желает получить по возможности *лаконичное*, *наглядное* и

полезное описание данных. Итак, один из основных тезисов нашего изложения –

✓ Анализ данных –
это наглядное, лаконичное и полезное их описание ✓

Попробуем перечислить основные моменты общепринятых на сегодня технологий анализа данных. Для этого сформулируем ряд вопросов, которые исследователь традиционно задает себе во время применения процедур анализа, и из ответов на которые постепенно складывается упомянутое выше описание данных. Последовательность таких вопросов составляет определенный ритуал, в котором используются исторически устоявшиеся термины и формулировки («заклинания»). Общность языка позволяет разным исследователям сравнивать результаты своего анализа.

Итак, проанализировав набор данных, исследователь должен быть готов ответить на следующие общие вопросы:

- *Стоит ли что-нибудь за данными и как оно устроено?*

В основе подавляющего большинства методов математической статистики лежит гипотеза о том, что набор данных представляет из себя независимую *выборку* из некоей генеральной совокупности – бесконечного набора точек, плотность распределения которых строго подчинена определенному закону. Ответ на заданный вопрос подразумевает указание на вид закономерности, соответствующей генеральной совокупности и правдоподобную оценку ее параметров. Соответствующие технологии составляют основу параметрических методов теории статистических оценок. Знание всех необходимых параметров (достаточных статистик) дает полную информацию о наборе данных в случае принятия основной статистической гипотезы.

Другим видом гипотезы является предположение о динамическом законе порождения данных.

Однако исследователь волен подвергнуть сомнению разумность гипотез о существовании генеральной совокупности или динамических законов и вовсе отказаться от них, утверждая, что за данными не стоит ничего, кроме них самих. Соответствующая концепция может быть названа гипотезой об *автоинформативности* набора данных. В этом случае исследователь занимается прямым описанием самого облака данных и применяет для этого соответствующие технологии.

Своеобразным компромиссом между противопоставленными выше подходами является применение непараметрических статистических

методов. В этом случае по прежнему предполагается наличие генеральной совокупности, но считается, что ее особенности не могут быть описаны простыми формулами с разумным количеством параметров. В этом случае с помощью набора данных в каждой точке пространства оценивается плотность распределения точек – создается непараметрическая модель генеральной совокупности. Иначе можно сказать, что плотность распределения точек генеральной совокупности оценивается с помощью общих формул, в которые в качестве параметров входит сразу весь набор данных (а не отдельные статистики, из них образованные). Существует несколько подходов для такого оценивания – со своими достоинствами и недостатками, и о них вкратце будет упоминаться во второй главе.

- *Возможно ли построить на множестве данных сколько-либо разумную (естественную, полезную) систему отношений?*

Здесь мы, прежде всего, подразумеваем применение всего семейства методов, связанных с *классификацией*, *кластерным анализом, таксономией* данных и т.д.

Решение задач классификации составляют основную цель применения методов теории распознавания образов. В задаче *классификации с учителем* отношения на системе объектов заданы изначально, требуется экстраполировать эти отношения на все пространство данных – отнести каждую из точек пространства данных к определенному классу при помощи *классификационного* или *решающего* правила, построенного при помощи набора данных (или его подмножества), в котором каждой точке заранее сопоставлена метка класса. В этом случае те данные, классификация которых заранее известна, называются *обучающим множеством (задачником)*. Как правило, той информацией, которая извлекается из данных в данном подходе является указание на вид решающей функции, или вид решающих поверхностей, отделяющих пространственно один класс от другого.

При *классификации без учителя* разбиение множества на классы осуществляется в результате решения задачи на оптимизацию некоторого критерия (например, критерия близости точек, принадлежащих каждому из классов к их *центроиду* – «типичному представителю» класса). В этом случае возможны два варианта, когда

а) число классов известно заранее, и тогда извлеченной из данных информацией является разбиение множества данных на заданное число классов эквивалентности, положения центроидов в пространстве данных, меры близости точек одного класса к их центроиду, меры удаленности одного класса от другого и т.д.;

б) число классов заранее неизвестно, и тогда к извлеченной информации добавляется количество классов (кластеров данных, точек сгущения), и, в конечном итоге, данные описываются как *иерархия* классов эквивалентности (например, все данные разбиваются на три класса, в каждом из них есть свое разбиение на подклассы и так далее).

- *Какова эффективная размерность множества данных?*

Извлечение информации при ответе на данный вопрос, как правило, ведется в двух направлениях (впрочем, не вполне независимых друг от друга): а) поиск многообразия меньшей размерности, вложенного в пространство данных, вдоль которого данные располагаются достаточно тесно; б) выделение группы *наиболее информативных признаков*, с помощью которых с заданной точностью можно было бы восстановить значения остальных, и определение зависимостей, связывающих признаки.

Естественным критерием при поиске многообразия, *моделирующего* или *аппроксимирующего* данные, является требование минимизации среднего расстояния от точки данных до ближайшей точки многообразия. На практике одного такого требования оказывается недостаточно (если минимизировать только указанную величину, то полученное многообразие может обладать совершенно неприемлемыми свойствами). Например, многообразие может представлять из себя ломанную в произвольном порядке соединяющую все точки данных – тогда среднее расстояние равно нулю.

При анализе «номенклатуры» признаков существуют несколько подходов.

Во-первых, это анализ *значимости* той или иной группы признаков относительно определенного критерия. Критерий формируется исходя из задаваемого «руками» вида зависимости, которая будет связывать одни признаки с другими (например, линейная связь). Задача состоит в том, чтобы указать минимальный поднабор признаков, с помощью которого, используя признаки как независимые переменные, с указанной точностью можно было бы восстановить значения других признаков при заданном виде зависимости. Простейший из способов – полный перебор сочетаний признаков – неприемлем с точки зрения вычислительных затрат уже для таблиц с несколькими десятками столбцов, поэтому для определения такого поднабора существует довольно обширный арсенал эвристических приемов.

Если указан поднабор наиболее значимых признаков, то менее значимые признаки могут быть вообще удалены из рассмотрения, как лишенные информативной нагрузки (однако, если не сказано явно, всегда

нужно мысленно добавлять – «в переделах заданной точности или допуска»).

Другой подход лежит в основе методов *факторного анализа*. Признаки объектов в этом случае представляются в виде комбинаций (линейных или других) меньшего числа других, непосредственно не измеряемых (скрытых, латентных) факторов. Факторы обычно конструируются так, чтобы они оказались взаимно некоррелированными. Число факторов с помощью которых удается свести погрешность описания – остаток, к приемлемому уровню и является эффективной размерностью пространства в этой модели.

В итоге, следуя тому плану, который набросан в отмеченных общих вопросах, исследователь описывает рассматриваемые данные (рассказывает о них) в терминах того или иного подхода. В результате у него складывается некоторое внутренне представление о структуре набора данных и о зависимостях, присущих данным. Причем, и это существенно для нашего изложения, следует отметить, что точные количественные характеристики, извлеченные с помощью разных подходов, играют второстепенную роль по сравнению с возникающим качественным представлением о наборе данных. Не столько конкретные числа, сколько наглядный образ позволит исследователю приступить на следующем этапе к созданию теоретической модели, обобщающей особенности изучаемой системы и обладающей способностью к прогнозу качественных и количественных характеристик для новых объектов, появляющихся в системе.

1.2.4. …и ответы, которые мы можем получить.

Равномерное и нормальное распределения

Стоит уделить внимание двум самым простым, традиционным и тщательнейшим образом изученным моделям данных – равномерному и нормальному распределениям. Трудно переоценить их роль для современного состояния практически любой из наук, имеющей дело со статистическими оценками, где либо одно, либо другое выступают в роли или базовых моделей, или начальных приближений, или предельных случаев.

Для того, чтобы избежать путаницы, сразу обратим внимание на то, что понятие *выборки* и *распределения* существенно отличны. Независимая выборка из генеральной совокупности – это конечное множество точек данных, полученных в результате случайного выбора точек из генеральной совокупности. Распределение – это закон сопоставления каждой

бесконечно малой области пространства вероятности появления в ней точки данных. Выборка дискретна, а распределение, как правило, кусочно-непрерывно.

Можно сказать, что равномерное распределение выражает идею равновозможности исходов в случае непрерывных случайных величин. Тогда вероятность принятия случайной величиной значения в произвольной точке фазового объема везде одинакова и обратно пропорциональна величине объема.

✂ Непрерывное равномерное распределение можно получить из дискретного, например, следующим образом. Возьмем бесконечный набор случайных величин Z_1, Z_2 … Z_n, принимающих значения 0 и 1 с вероятностью ½. Если мы построим непрерывную случайную величину $X = \sum_{i=1}^{\infty} Z_n 2^{-n}$ (фактически формула дает способ представления числа из диапазона [0,1] в двоичной форме), то ее распределение будет равномерным на отрезке [0,1].

Другой пример равномерного распределения – если из произвольных независимых непрерывных случайных величин X_1, X_2 … X_n образовать сумму $s_n = X_1 + X_2 \ldots + X_n$, то в пределе $n \to \infty$ дробная часть s_n будет распределена равномерно на отрезке [0,1).✂

Допустим, исследователь решил, что данные в пространстве распределены равномерно (то есть, например, являются выборкой из генеральной совокупности с равномерным законом распределения). Это соответствует ситуации, когда «все возможно» – то есть в системе может существовать объект или явление с любым из допустимых набором свойств с равной вероятностью.

Каким образом тогда исследователь может описать свои данные – то есть каковы будут его ответы на заданные выше вопросы? Надо сказать, что ответы эти будут максимально (из всех возможных моделей) бедны. То есть, если данные распределены в пространстве равномерно, то в облаке данных нельзя выделить никакой естественной структуры (нет ни кластеров, ни точек сгущения), в облаке нет выделенных направлений, и эффективная размерность множества данных совпадает с размерностью пространства, то есть нет никаких значимых связей между признаками объектов. Таким образом, равномерное распределение является наименее емким с точки зрения содержащейся в нем информации.

✂ С помощью простого расчета можно показать, что равномерное распределение соответствует максимальной энтропии случайной величины.

В статистической физике максимум энтропии пространственного распределения молекул соответствует равновесному (устойчивому) состоянию системы. ✂

Если для модели равномерного распределения характерно «отсутствие» информации в наборе данных, то модель нормального распределения является простейшим случаем, когда в наборе данных содержится информация, сформировавшаяся «стихийно», то есть если исследуемая система объектов находится под воздействием большого числа независимых случайных факторов, каждое из которых изменяет свойства-признаки объектов в определенном направлении, но среди этих факторов нельзя выделить «главного», чье воздействие на систему было бы несопоставимо по масштабу в сравнении с суммой всех остальных воздействий (то есть действие любого фактора может быть полностью скомпенсировано либо другим, либо суммой нескольких факторов).

✂ Термин *нормальное распределение* был введен К.Пирсоном. Однако сам вид и свойства распределения подробно изучались, начиная с Гаусса. На сегодняшний день нормальное распределение является наиболее теоретически изученной и практически применяемой статистической моделью. Она возникает в качестве основной в огромном количестве естественнонаучных приложений.

Нормальное распределение может быть получено как предел дискретного биномиального распределения при большом количестве испытаний. Соответствующий результат оформлен в виде *теоремы Муавра-Лапласа*.

Полное теоретическое исследование исключительной роли нормального распределения было закончено только в тридцатый годах двадцатого столетия. Обоснование такой исключительности дается в *предельных теоремах* теории вероятностей. Утверждение центральной предельной теоремы состоит в том, что для последовательности X_1, X_2 … X_n независимых случайных величин отклонение суммы $s_n = X_1 + X_2 \ldots + X_n$ от своего математического ожидания $M(s_n)=M(X_1)+M(X_2)+\ldots+M(X_n)$ подчиняется нормальному закону распределения в пределе $n \to \infty$. Условия применения центральной предельной теоремы даются в

теореме Ляпунова (основной вывод теоремы – среди последовательности $X_1, X_2 \ldots X_n$ не должно быть величин, имеющих больших значений моментов по сравнению с общей дисперсией суммы s_n).

Интересной особенностью нормального распределения является то, что сумма нормально распределенных величин также распределена нормально. Для равномерного распределения аналогичного утверждения сделать нельзя. Сумма нескольких равномерно распределенных величин не является равномерно распределенной, но распределение такой суммы быстро стремится к нормальному при увеличении числа слагаемых. ✂

Допустим, что облако объектов «похоже» на выборку из генеральной совокупности, подчиненной закону нормального распределения (уточнению понятия «похоже» посвящена литература по проверке статистических гипотез, например [4,29], здесь мы не будем вдаваться в тонкости этой серьезной науки). Попробуем дать описание распределения точек данных в пространстве. Данные представляют собой один кластер, имеют одну точку сгущения (*унимодальная плотность*) в точке среднего арифметического значений всех признаков. Чем ближе к этой точке, тем выше плотность распределения объектов. Более 60% всех объектов находятся в области, представляющей собой эллипсоид, центрированный в точке сгущения с осями, равными собственным значениям так называемой ковариационной матрицы (*эллипсоид рассеяния*, подробнее об этом во второй главе, см. рис. 8).

Обратимся теперь к анализу эффективной размерности. Прежде всего, ответим на следующий вопрос: *что из себя представляет линия, для которой среднее квадрата расстояния от нее до точек данных минимально*?

Сразу следует отметить – если мы не накладываем никаких

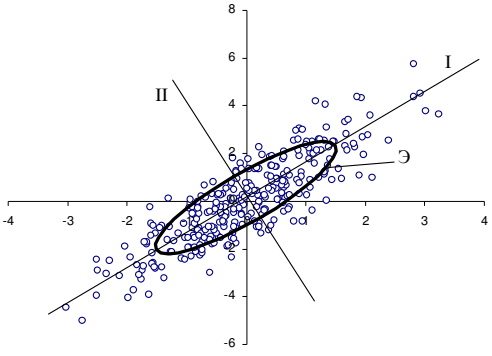


Рис. 8а. Двумерное нормальное Распределение точек.
I, II – главные компоненты,
Э – эллипсоид рассеяния

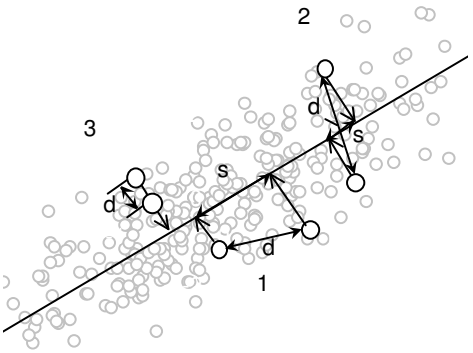


Рис. 8б. Искажения, возникающие при проецировании.
d – реальное расстояние,
s – расстояние между проекциями
1) s ≈ d ; 2) s << d ; 3) s = 0

дополнительных ограничений на регулярность[2] этой линии, то подойдет любая ломаная, соединяющая точки данных в произвольном порядке. Поступим иначе и потребуем, чтобы эта линия была максимально регулярной, пусть это будет прямая.

Назовем такую прямую первой из *главных компонент*. Она проходит через центр облака и ориентируется вдоль наибольшей вытянутости (*дисперсии*) облака данных (см. рис. 8а). Это направление совпадает с направлением наибольшей по длине оси эллипсоида рассеяния.

Значения координат вектора, задающего направление первой из главных компонент, являются количественными мерами значимости признаков. Чем меньше значение соответствующей координаты, тем менее значим и информативен признак. Этот же вектор является фактором для простейшей однофакторной модели набора данных с распределением, «похожим» на нормальное. В этом смысле знание координат этого вектора является самой существенной долей извлеченной из набора данных информации, причем тем более существенной, чем длиннее большая из осей эллипсоида рассеяния по сравнению с остальными. Первая из главных компонент позволяет приближенно восстановить значения всех признаков, если известно значение только одного из них.

Если точность такого моделирования данных оказывается недостаточной, то определяется направление второй из главных компонент. Из векторов, соответствующих каждой точке данных вычтем вектор ортогональной проекции точки на первую главную компоненту. Назовем новый полученный набор векторов *множеством первых остатков*. Построим в этом множестве первую главную компоненту. Ее направление окажется направлением второй главной компоненты для исходного множества. Это будет прямая, проходящая через центр распределения, перпендикулярно к первой из главных компонент, совпадающая с направлением второй из главных полуосей эллипсоида рассеяния.

На полученные два вектора можно натянуть *плоскость первых двух главных компонент*. Среди всех плоскостей эта плоскость обладает свойством *минимума суммы квадратов расстояний от нее до точек данных*. С помощью нее можно а) построить двухфакторную модель данных; б) восстановить значения признаков объекта, если известны значения *двух* признаков; в) простым образом *визуализировать* многомерные данные, спроецировав каждую точку данных ортогонально на плоскость первых двух главных компонент.

---

[2] Если бы мы имели дело с распределением (бесконечным числом точек), то не пришлось бы накладывать никаких ограничений – в соответствующей постановке задачи мы бы и так получили регулярную кривую.

Остановимся несколько подробнее на последнем моменте. Процедура ортогонального проецирования точки на плоскость задает отображение из исходного пространства большой размерности R^m в пространство R^2, то есть сопоставляет каждой точке исходного пространства две координаты на плоскости. Среди всех отображений типа ортогонального проецирования на *плоский экран* такое отображение будет оптимальным по отношению к сохранению структуры расстояний между точками в исходном пространстве. Если же бы мы имели дело с бесконечным числом объектов генеральной совокупности, подчиненной нормальному закону распределения, то такое отображение было бы оптимальным среди любых отображений из R^m в R^2.

Здесь мы впервые касаемся основного вопроса нашего изложения – *зачем же нужно визуализировать данные*? Дело в том, что было бы очень полезно иметь возможность представить многомерное облако данных в виде наглядной двумерной картинки, то есть снизить размерность облака до двух измерений, но таким образом, чтобы на полученном изображении некоторым оптимальным образом были видны основные закономерности, присущие набору данных: его кластерная структура, изначальное разделение данных на классы (если таковое имеется), существование различных зависимостей между признаками и так далее. Вообще, если исследователь будет иметь возможность хоть как-то наглядно представить себе многомерное облако данных, многие задачи анализа (то есть в конечном итоге – описания данных) решаются с помощи непосредственного зрительного восприятия картины множества объектов.

Итак, наиболее приемлемым способом визуализировать набор точек данных, чье распределение «похоже» на выборку из нормальной генеральной совокупности, является ортогональное проецирование на плоскость первых двух главных компонент. Плоскость проектирования является, по сути плоским двумерным «экраном», расположенным в пространстве таким образом, чтобы обеспечить «картинку» данных с наименьшими искажениями. Такая проекция будет оптимальна (среди всех ортогональных проекций на разные двумерные экраны) в трех отношениях:

1) *Минимальна сумма квадратов расстояний от точек данных до проекций* на плоскость первых главных компонент, то есть экран расположен максимально близко по отношению к облаку точек.

2) *Минимальна сумма искажений расстояний между всеми парами точек из облака данных* после проецирования точек на плоскость. Поясним это подробнее. Возьмем любую пару точек в исходном пространстве. Между ними есть какое-то ненулевое расстояние. После проецирования каждой из точек на плоскость главных компонент

расстояние между проекциями будет уже иным (см. рис. 8б) – в некоторых случаях оно может даже оказаться нулевым – для разных точек проекции могут совпасть, если они лежат на одной прямой, перпендикулярной плоскости проецирования. Можно ввести меру искажения расстояния между точками после проецирования (например, относительную погрешность). Утверждается что при использовании плоскости первых двух главных компонент сумма этих искажений достигает минимума (разумеется, если точек достаточно много).

3) *Минимальна сумма искажений расстояний между всеми точками данных и их «центром тяжести», а также сумма искажений углов между векторами, соединяющими точки и «центр тяжести»*. В случае нормального распределения центр тяжести распределения совпадает с точкой сгущения – геометрическим центром распределения и средним арифметическим значений признаков всех объектов.

✋ ***Пример. Проецирование данных на плоскость 1-ой и 2-ой главных компонент.***

На рис. 9 показана проекция четырехмерного облака точек данных, соответствующей реальной таблице данных, собранных в результате измерения ботанических классификационных признаков трех различных видов цветка ириса. Эта база данных традиционно используется в литературе в качестве иллюстраций при испытании различных алгоритмов анализа данных. В таблице представлено по 50 результатов измерений для каждого вида цветка. В качестве координат четырехмерного пространства использовались длина и ширина лепестка цветка, и длина и ширина чашелистика цветка. Из рисунка видно, что, используя метод главных компонент, можно уверенно отделить класс Iris-setosa от двух других классов, в то время как классы Iris-versicolor и Iris-virginica остаются перемешанными.

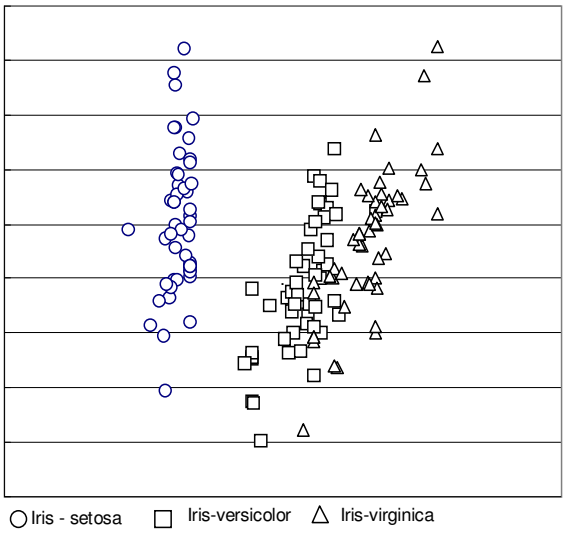


Рис. 9. Проецирование на плоскость первых двух главных компонент.

Возникает естественный вопрос – а как обстоит дело с наборами данных, которые не могут считаться выборками из генеральной совокупности с нормальным распределением? Перечисленные свойства плоскости главных компонент сохраняются для *произвольного* облака точек при условии, что рассматривается визуализация (проецирование) только при помощи двумерных *линейных* многообразий (различным образом ориентированных в пространстве плоскостей-экранов). Разумеется, может найтись такое *криволинейное* двумерное многообразие, с помощью которого будет возможно добиться еще меньших значений перечисленных критериев. Способы построения таких *оптимальных многообразий* и составляют основу нашего изложения.

В своем рассмотрении мы почти полностью ограничиваемся двумерным случаем, как наиболее естественным для визуализации. Понятно, что метод главных компонент позволяет построить трех-, четырех- и более факторные модели, и, вообще, выбрать k главных компонент, использование которых в качестве факторов обеспечивало бы необходимую точность описания данных. Тогда можно сказать, что набор данных является эффективно k-мерным. Разумеется, наиболее интересны случаи, когда k существенно меньше размерности пространства.

Некоторые выводы

Итак, мы качественно рассмотрели основную модель многомерного статистического анализа данных – многомерное нормальное распределение. В качестве оптимальных факторов, описывающих данные,

в этой модели выступают линейные комбинации признаков, задающие в пространстве направления, вдоль которых дисперсия данных максимальна. Простота и изученность такой модели связана как раз с её *линейностью*.

Можно сказать, что нормальное распределение возникает всякий раз, когда создается статистическая модель *линейной системы*. Поскольку точные науки хорошо умеют справляться почти исключительно только с такими простыми системами, то легко понять популярность нормального распределения в физике, химии, биологии и так далее.

Как мы уже упоминали, традиционным первым шагом в статистическом исследовании является оценка математического ожидания и общей дисперсии для набора данных (в многомерном случае – ковариационной матрицы, то есть по сути m дисперсий, где m – размерность пространства). Эти величины являются достаточными статистиками нормального распределения, то есть первым и традиционным шагом является представление набора данных в виде простейшей линейной модели.

Если облако точек является явно более сложным, например, имеет более одной точки сгущения, то его в первом приближении можно моделировать пространственным наложением нескольких нормальных выборок, для каждой из которых определена своя ковариационная матрица и набор главных компонент. Если считать каждую из точек данных вершиной своей нормальной «шапочки» распределения, то в результате такого наложения возникает самый простой из непараметрических способов оценивания плотности распределения генеральной совокупности.

С другой стороны, за исключением модельных тестовых примеров, практически все реальные данные по своей структуре оказываются весьма далеки от нормальных выборок, точно так же как реальные физические системы почти всегда отнюдь не линейны.

В связи с этим особый интерес представляют принципиально нелинейные способы моделирования и визуализации данных, позволяющие построить эффективную технологию анализа реальных таблиц данных.

Для начала в традициях построения физических моделей природы рассмотрим возможные полумеры – малые отклонения от линейного (нормального) случая.

1.2.5. Квазилинейные подходы

Выше мы уже упоминали, что моделирование (и визуализация) данных с помощью линейных факторов является оптимальными лишь в случае близкого к нормальной выборке облака точек данных. Если это не так, то есть нормальное распределение никаким образом не может являться моделью данных, то для их моделирования необходимо, как минимум, прояснить следующие моменты:

1. Выбрать критерий оптимальности, согласно которому будет решаться задача построения моделирующего многообразия, и, собственно, разработать способ построения этого многообразия.
2. Определить способ, с помощью которого точки данных из исходного пространства будут переноситься на моделирующее многообразие. В случае, если оптимальным многообразием является плоскость, такой проблемы нет – наиболее естественным является ортогональное проецирование на плоскость.

> ✂ На самом деле и в этом случае возникает ряд вопросов. В многомерном пространстве для вложенной двумерной плоскости вектор нормали определен неоднозначно. Под ортогональным проектированием обычно понимают проектирование в ближайшую точку плоскости, однако для различных метрик такая точка, вообще говоря, может быть разной. В частности, для обычной евклидовой и взвешенной евклидовой метрики ортогональные проекции не совпадают. ✂

Смесь гауссовых компонент

Во-первых, один из очевидных способов обобщения линейных моделей является представление общего распределения данных в виде взвешенной суммы нормальных распределений, должным образом «сшитых» друг с другом, для каждого из которых оптимальным моделирующим многообразием является линейное многообразие, натянутое на несколько собственных векторов ковариационной матрицы соответствующей нормальной компоненты смеси. Из отдельных кусков этих плоскостей можно так или иначе сшить главную поверхность.

Согласно критерию оптимальности определяется состав набора нормальных компонент, коэффициенты смеси, условия сшивки и т.д.

Очевидно, что чем больше нормальных компонент будет в смеси, тем, с одной стороны, более криволинейной будет полученная в результате сшивки поверхность. В предельном случае каждая точка данных может использоваться как представитель своей нормальной компоненты (этот случай может быть использован для непараметрических оценок плотности распределения). С другой стороны, чем меньше точек определяют каждую нормальную компоненту, тем менее ясен смысл вычисления собственных векторов соответствующей ковариационной матрицы (в упомянутом предельном случае вообще ни о каких собственных векторах речи не идет, соответствующие дисперсии либо полагаются равными заданной величине «окна», либо применяются какие-либо варианты их оценки по положению соседей).

В целом, подход, основанный на построении оптимального многообразия с использованием модели смеси нормальных распределений, является весьма вычислительно трудоемким. Кроме собственно решения проблемы построения моделирующего многообразия необходимо решать задачу разбиения множества данных на компактные подмножества, то есть решать, фактически, задачу кластерного анализа, и, таким образом, становится понятно желание подойти к проблеме с иных позиций.

Сглаженные квазилинейные развертки данных

После того, как построена первая из главных компонент, у нас, после применения операции проектирования каждой из точек на прямую, соответствующую главной компоненте, появляется возможность сопоставить каждой из точек данных определенное число – координату. Достаточно только произвольным образом выбрать на прямой начало отсчета, см. рис. 10). Поскольку после проецирования все точки лежат на одной прямой, то они оказываются упорядоченными. В результате появляется возможность последовательно соединить соседние точки данных отрезками и в результате получить одномерную развертку данных, то есть такое одномерное многообразие, которое проходит через все точки данных согласно выбранному способу упорядочивания данных.

✂ Существуют другие способы упорядочивания точек данных таким образом, чтобы близкие в пространстве точки оказались соседями на полученной развертке. Например, можно упомянуть *метод адаптивной*

развертки, в котором точки данных упорядочиваются согласно следующему алгоритму [19].

1. Найти точку данных x_i, для которой суммарное расстояние до остальных объектов максимально, то есть $d_\Sigma(x_i)= \max$. Это будет первая точка в развертке.
2. Из еще не вошедших в развертку точек найти такую точку x_j, для которой выполняется условие $\left[d_\Sigma^M(x_j)/M\right]^\beta \Big/ d(z,x_j) = \max$, где z – точка, полученная на предыдущем шаге, $d_\Sigma^M(x_j)$ – суммарное расстояние от объекта xj до его M ближайших соседей, M и β – параметры метода. Эта точка будет следующей в развертке.
3. Шаг 2 повторяется до тех пор, пока все точки не окажутся в развертке.

Пример применения метода адаптивной развертки см. на рис. 11. ✂

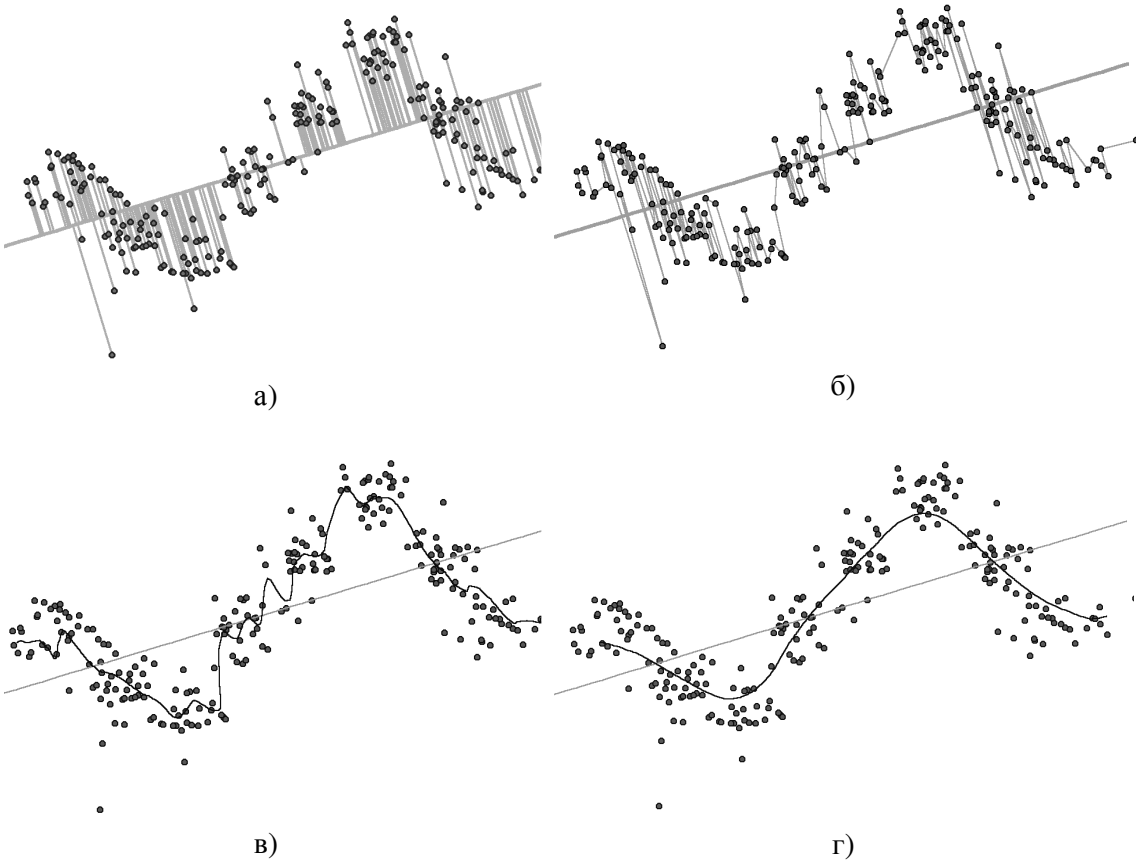


Рис. 10. Сглаженные квазилинейные развертки

а) Данные могут быть упорядочены одномерной координатой с помощью произвольно ориентированной в пространстве прямой и ортогонального проецирования на нее. Оптимальным направлением прямой является направление первой главной компоненты.

б) Результатом такого упорядочивания может стать развертка данных, которая служит своеобразной моделью данных.

в) Развертка может быть сглажена тем или иным методом и в результате получается гладкая модель данных, обладающая обобщающей способностью.

г) Сглаживание может быть сделано более радикальным – в результате из данных извлекается «главная кривая», с помощью которой можно сократить описание данных и построить одномерную модель данных, отражающую главные особенности, присущие данным без учета случайных шумов.

Преимуществами такой развертки является то, что исчезает проблема проецирования данных на полученное многообразие – точки данных и так ему принадлежат. Кроме того, что данные оказываются упорядоченными, из них можно извлечь информацию, например, о взаимном расстоянии между соседними точками, направлении отрезков,

соединяющих соседние точки и которую можно графически отобразить с помощью разного рода диаграмм.

Однако, практическая полезность такого многообразия, как модели данных, находится под вопросом. Во-первых, как уже отмечено, данные могут быть заданы с допуском или погрешностью, и, соответственно, их положение в пространстве определено не точно. Во-вторых, данные вообще могут содержать пробелы – и тогда многообразие оказывается не определено. И, наконец, самым большим недостатком является то, что такая модель данных не отслеживает общей *тенденции* (или *закона*), содержащегося в данных, кроме той, что заключена в способе упорядочивания. Если данные «зашумлены» – а именно это характерно для реальных наборов данных – то полученная ломанная отслеживает все случайные флуктуации – таким образом, длина описания данных в такой модели не сокращается – для задания положения вершин ломанной необходимо то же количество информация, что и для описания всех координат точек данных.

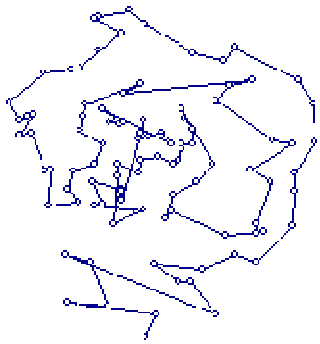


Рис. 11а. Адаптивная развертка двумерного облака данных. β=1, M=10

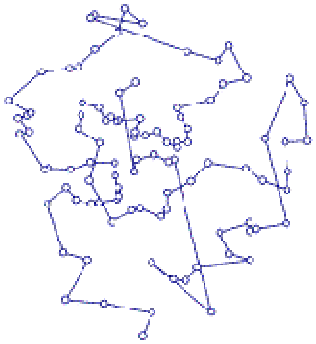


Рис. 11б. Адаптивная развертка двумерного облака данных. β=1, M=99

Вообще, возникшая ситуация достаточно характерна. При построении модели данных исследователю всегда приходится искать компромисс между точностью представления данных и *обобщающей способностью модели*. Увеличивая число задаваемых параметров модели, можно добиться сколь угодно малой погрешности описания тех данных, которые были использованы для построения модели. Однако, цель создания такой модели – *обобщение некоторого «опыта»*, накопленного в

данных; при этом не достигается. Это значит, что для любого объекта, который не был использован при создании модели, погрешность его описания может быть весьма велика. Более подробно мы остановимся на этом моменте в следующем разделе.

Тем не менее, можно улучшить обобщающую способность полученной выше развертки, если сгладить полученную ломанную. Некоторые конкретные варианты сглаживания рассмотрены во второй главе. Здесь нам важнее подчеркнуть, что при использовании сглаживания возникает, как минимум, один параметр, который является регулятором типа «точность-общность». Чем более точной по отношению к описанию исходных данных мы делаем нашу модель – тем большее количество параметров необходимо задавать для описания модели. Предельный случай – полученные выше развертки. С другой стороны, увеличивая общность модели, мы снижаем количество необходимых для описания модели параметров за счет снижения точности описания исходных данных и получаем в пределе линейное подпространство, натянутое на главные компоненты.

Итак, сглаженные линейные развертки, которые могут быть положены в основу построения квазилинейных одномерных моделей данных, возникают в результате конкуренции двух факторов – стремления описать исходные данные как можно точнее и сделать модель более гладкой, сокращая при этом длину описания модели. Квазилинейность подхода заключается в том, что он существенным образом использует построенную линейную модель данных, и надстраивается над ней.

Квазилинейные модели могут быть построены и в двумерном случае, надстраиваясь над плоскостью двух первых главных компонент. Тогда каждая точка после проектирования на плоскость получает две координаты и данные, естественно, уже не будут упорядочены в прямом смысле слова. Тем не менее, взяв за основу двумерные координаты каждой из точек можно построить сглаженную двумерную поверхность, форма которой будет как-то отражать отклонения распределения данных от нормального.

1.2.6. Существенно нелинейные случаи

Роль линейных методов в статистике весьма велика. Существует общий рецепт – если линейный метод работает хорошо и решает поставленные задачи – то его и следует использовать, даже если нет статистически оправданных посылок для его применения. Однако, часто ситуация бывает обратной, и тогда задача исследователя – описывать данные «так, как они есть», без использования дополнительных предположений о характере их распределения.

Исследователь должен, прежде всего, сформулировать критерий оптимальности, которому должна удовлетворять модель данных. Такой критерий должен обладать следующими свойствами:

1. Он должен быть компромиссным по отношению к конкуренции между точностью описания и обобщающей способностью модели. По всей видимости, критерий должен содержать параметры, позволяющие в зависимости от условий задачи увеличивать то или другое свойство.

2. Желательно, чтобы критерий был таким, чтобы для нормального распределения данных для него наилучшей моделью являлись линейные многообразия, натянутые на главные компоненты. Это позволит сравнивать преимущества такого моделирования по сравнению с традиционными методами.

Задачу построения модели данных можно сформулировать как задачу аппроксимации многомерного набора точек данных более или менее гладкими поверхностями, вложенными в многомерное пространство.

> ✂ *Аппроксимацией* в самом общем определении называется метод, заключающийся в замене сложных объектов другими, более простыми. В этом смысле сложное многомерное множество точек данных заменяется более простым и регулярным объектом – многообразием или сеткой, для описания которой требуется меньше информации. ✂

1.2.7. Нейросетевые модели данных

В нашем изложении при построении аппроксимирующих поверхностей мы предпочитаем говорить на «языке геометрии». То есть, по возможности, явно представлять себе размещение аппроксимирующей поверхности в многомерном пространстве. Однако, стоит отметить, что вполне возможно вести изложение и на другом языке – например, нейросетевом, который стал в последние десятилетия очень популярен. В наши задачи не входит изложение методов нейросетевого анализа данных, однако, подчеркнем, что как результат работы нейросети так или иначе может быть представлен геометрически, так и практически любой из алгоритмов, приведенных в этой книге может быть естественным образом «переведен» на нейросетевой язык. Так, например, способ построения карт Кохонена, о которых речь пойдет ниже, традиционно излагается с помощью описания соответствующей нейросетевой архитектуры. Задача снижения размерности данных может быть описана как с помощью

наглядных образов криволинейных поверхностей, вложенных в многомерное пространство, так и с помощью описания такой нейросети, в которой число входов равно размерности пространства, а количество выходов равно размерности моделирующего многообразия.

✂ Более интересно рассмотреть нейросеть «с узким горлом» (см. рис. 12). В ней число выходов равно число входов, но сеть содержит внутренний слой с небольшим числом нейронов. Сеть обучается на воспроизведение входов – то есть ответ нейросети считается правильным, когда значения сигналов на каждом выходе совпадает со значением на соответствующем ему входе. Если удается обучить такую нейросеть, то она способна решать задачу сокращения размерности – и тогда сигнал необходимо снимать с нейронов «горла» сети, и задачу проецирования данных, не вошедших в задачник, в пространство меньшей размерности. ✂

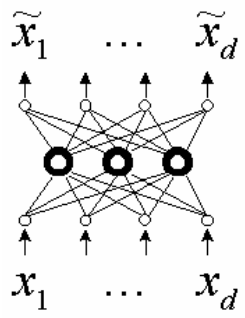


Рис. 12. Архитектура нейронной сети с узким горлом.
На вход такой сети подаются примеры из задачника. Требуемые значения на выходе должны максимально точно восстанавливать значения на входах (сеть обучается решать задачу $x_i = \tilde{x}_i$, i = 1 .. d). Таким образом в центральном слое происходит сжатие информации.

Тем не менее, возможность перевода алгоритмов на нейросетвую основу является очень ценной, поскольку в случае реального применения алгоритмов на практике, нейросетевой подход позволяет получать высокоэффективные параллельные архитектуры при аппаратной реализации алгоритмов в устройствах.

Для нас будут важны некоторые понятия, характерные для описания функционирования нейросетевых моделей. Так, для обучения нейросетей важен поиск компромисса между величиной *ошибки обучения* и *ошибки*

обобщения. Под первой подразумевается средняя погрешность воспроизведения сетью тех данных, которые были использованы для ее обучения (например, процент случаев с правильным распознанием класса для исходного набора данных). Как правило, если обучающая выборка не является противоречивой, с помощью увеличения числа нейронов и синаптических связей можно добиться сколь угодно малой ошибки обучения. Для оценки ошибки обобщения необходимо иметь тестирующую выборку, то есть набор «проверочных заданий», которые не были использованы для обучения. Как правило, если исходных данных достаточно много, то часть из них может быть удалена из процесса обучения и использоваться для оценки обобщающей способности нейросети.

В нейросетевых подходах регулятором типа «точность-общность» является число нейронов и синаптических связей сети, число слоев нейросети или время обучения (для циклических сетей), вид нелинейности характеристической функции. Как правило, исследователь экспериментирует с этими параметрами с целью добиться приемлемых результатов. Успех применения нейросетей и своеобразный «нейробум», с ними связанный, с одной стороны, вызван тем обстоятельством, что нейросетевые архитектуры позволяют получать информационные модели и «хорошими» аппроксимационными свойствами – а именно, в отличие от давно известных интерполяционных формул, нейросети с хорошей точностью описываю данные в местах их скоплений и гладко интерполируют их в местах их разрежения, а, с другой стороны, с высокоэффективными приемами обучения нейросетей (такими, как вычисление градиентов с помощью обратного функционирования и т.д.)

1.2.8. Физикалистские игры с данными. Преобразования пространства данных.

Вообще говоря, смысл расстояния между двумя точками в пространстве признаков весьма условен. За редкими исключениями, расстояние не имеет никакого физического смысла, кроме меры различения объектов. С одной стороны, это создает проблему выбора подходящей метрики, с другой – манипулируя этим выбором, исследователь имеет возможность добиваться определенных «эффектов» над представлением облака данных с целью подчеркнуть те или иные особенности их структуры.

Начнем с того, что предобработка данных может рассматриваться на языке изменения метрики пространства признаков. Линейные преобразования значений признаков (например, нормировка на

среднеквадратичное отклонение, интервал) приводят к различным линейным метрикам. Часто оказывается, что набор значений отдельных признаков полезно подвергнуть нелинейному преобразованию (например, если значения признака отличаются на порядки, то логично применить какой-либо вариант логарифмического преобразования).

Удачно выбранное преобразование пространства признаков вообще может нелинейную задачу распознавания образов свести к линейной (например, в случае распределения данных, расположенных вдоль поверхностей концентрических сфер). Разумеется, крайне трудно «угадать» подходящее преобразование без предварительного анализа структуры данных, но если такой «разведочный» анализ укажет на существование определенных зависимостей в пространстве признаков, то задачи классификации, снижения размерности могут быть существенно упрощены.

Рассмотрим некоторые идеи, возникающие в этом направлении. Все они могут быть реализованы как варианты в процессе предобработки данных перед их визуализацией.

Гравитирующие данные

Если дать волю фантазии, то на преобразование пространства признаков, которое, в конечном итоге, приводит к тому, что меняются расстояния между объектами в пространстве, можно посмотреть с иных позиций, и предположить, что эти расстояния меняются не из-за выбора метрики, а вследствие того, что сами точки данных обладают подвижностью. Разумеется, нет нужды интерпретировать конкретные траектории движения точек данных, интерес представляет только их новое расположение, в котором будет достигаться минимум какого-либо функционала от координат положений точек.

Первое, что приходит в голову – это определенным образом задать закон взаимодействия частиц-точек друг с другом, указав, например, какой-либо центрально-симметричный (хотя и не обязательно) потенциал взаимодействия. Точки данных в таком подходе могут как отталкиваться, так и притягиваться друг к другу, им можно приписывать различную «массу», то есть некоторые точки могут считаться «весомее» других. Взаимодействие между частицами может быть дальнодействующим (по типу гравитации) или иметь определенный радиус (по типу ядерных сил). Частицы могут взаимодействовать выборочно (например, если заданно разбиение множества точек на подмножества, то точка может взаимодействовать сильнее с точками, принадлежащими ее же классу). В любом случае критерий оптимальности конфигурации точек – величина

энергии их взаимодействия, то есть после каждого перемещения частиц суммарная энергия их взаимодействия должна становиться меньше.

Рассмотрим, например, облако точек, гравитирующих в многомерном пространстве признаков. Энергия их взаимодействия равна нулю в пределе бесконечного разнесения точек и становится отрицательной и бесконечной в случае, если все точки собрались вместе. Таким образом, облако точек будет стремиться «коллапсировать». Разумеется, если все точки окажутся собранными вместе, то будет потеряна всякая информация о первоначальной структуре данных, поэтому параллельно с гравитацией необходимо ввести эффект, приводящий к «разбеганию» точек друг от друга. Таким образом, мы получаем своеобразную «Вселенную данных», в которой конкурируют два процесса – с одной стороны, каждая их точек данных стремится притянуться к близлежащим точкам – и это приводит к собиранию данных в «сгустки» - кластеры, а с другой – происходит «расширение Вселенной». Такая конкуренция приводит в результате к тому, что данные автоматически собираются в компактные, отделенные друг от друга в пространстве группировки, но при этом сохраняется информация о первоначальной структуре этих группировок. Скорости конкурирующих процессов должны быть определенным образом подобраны, чтобы, например, средняя плотность данных оставалась постоянной.

Конкретные варианты реализации предложенных идей приведены в разделе 2.1.7.

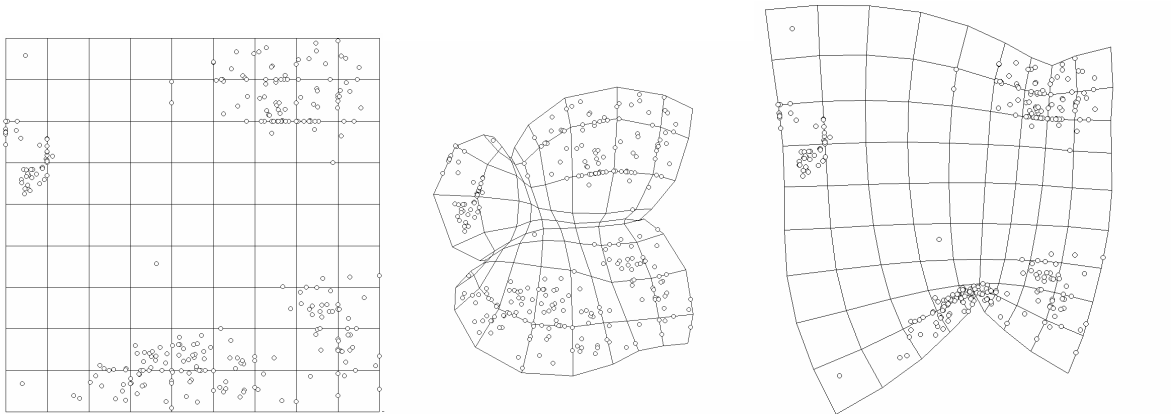
Далее, можно устроить так, чтобы данные взаимодействовали не друг с другом, а с многообразием, определенным образом вложенном в пространство признаков. Таким образом, если мы, например, разместим среди данных двумерную поверхность, то с помощью такой процедуры можно заставить точку данных перемещаться к точке поверхности до тех пор, пока она не окажется в непосредственной близости от многообразия.

И, наконец, может оказаться так, что более интересным будет задать для точек потенциал отталкивания, а параллельно запускать процесс «сжимания» Вселенной данных. Если скорости этих процессов будут подогнаны для сохранения средней плотности, то в результате точки данных более или менее равномерно заполнят доступный им вначале объем пространства.

Нелинейные преобразования пространства признаков.

Изложенная выше идея может быть рассматриваться в несколько ином виде. Так мы можем задаваться целью некоторым регулярным

способом равномерно распределить данные в исходном доступном им объеме пространства так, чтобы соседние данные оставались соседними. Для этой цели необходимо тем или иным образом оценить плотность данных (например, любым из непараметрических способов) и в тех областях, где эта плотность высока «сжать координатную сетку» пространства, а где, наоборот, данные расположены редко, «растянуть» координатные линии (см. рис.13)



а) б) в)

Рис. 13. Преобразования двумерного пространства признаков.
а) исходное распределение точек данных
б) координатная сетка для метрики, в которой данные распределены более равномертно
в) координатная сетка для метрики, в которой подчеркнута кластерная структура данных

В результате, данные нанесенные на новую координатную сетку будут расположены почти равномерно (за исключением, может быть, граничных областей).

Этот же тип преобразования можно применить в обратном направлении, и тогда области с плотным размещением данных окажутся еще более плотными, и наоборот. Таким образом, мы получаем своеобразный регулятор «контрастности» кластерной структуры данных, «поворачивая» этот регулятор в ту или иную сторону, мы можем или подчеркнуть кластерное разбиение данных, либо, наоборот, размывать его. Соответствующие математические выкладки приведены в разделе 2.1.5.

Существуют преобразования пространства признаков, для которых критерием оптимальности является максимум суммы квадратов

коэффициентов линейной корреляции для некоторой группы признаков. Такое преобразование может применяться в двух случаях:

1. При использовании номинальных и порядковых шкал числовые метки шкалы, присваиваемые тем или иным признакам, могут быть выбраны с большой степенью произвола, поскольку смысловую роль играет не величина интервала между значениями меток, а только их порядок следования. Таким образом, любое монотонное преобразование оставляет этот порядок неизменным. Выбор функции преобразования часто осуществляется таким образом, чтобы конечные наборы числовых меток обеспечивали максимальную степень *линейной* зависимости между выбранными признаками. Процедуры такого типа называются *оцифровкой* и предназначены для приведения признаков всех типов к единой непрерывной количественной шкале.

2. Даже для непрерывных количественных шкал признаков можно попробовать задаться определенным семейством преобразований шкалы признака и оценить параметры семейства такие, чтобы признаки после преобразования зависели друг от друга максимально линейно. Это, как уже указывалось, может привести к существенному упрощению задач распознавания образов и классификации.

Локальные статистики

В связи с темой нашего изложения можно упомянуть относительно новый подход к анализу многомерных данных, связанный с построением так называемых *локальных статистик*. В основе этого подхода лежит идея о том, что преобразование пространства признаков можно построить таким образом, чтобы удовлетворить определенному критерию оптимальности не для всего набора данных, а лишь для его единственной точки. Таким образом в методах, связанных с построением локальных статистик рассматривается набор данных «с точки зрения» одного из объектов.

Как правило, удобно поместить начало координат в ту точку, где находится этот «базовый» объект. После этого можно, например, анализировать диаграммы распределения расстояний от точек данных до начала координат. Из вида этих диаграмм, например, можно увидеть, принадлежит ли точка данных крайним областям облака данных или находится внутри нее (см. рис. 14)

Если множество данных изначально разбито на классы, то таким образом можно представить себе как точки того или иного класса расположены относительно «базовой» точки (рис. 14)

Приведенные на рис. 14 диаграммы построены, исходя из предположения, что в обоих вариантах выбора базовой точки, расстояние от нее до остальных измеряется с помощью одной и той же евклидовой метрики. Однако, можно построить для каждой из выбранных точек данных свою «локальную» метрику, удовлетворяющую определенному критерию. В разных задачах этот критерий выбирается по-разному.

В задачах построения классифицирующих разделяющих поверхностей можно поставить следующим образом: найти такое преобразование метрики, в котором отношение суммы расстояний от базового объекта до объектов своего же класса к сумме расстояний до объектов других классов минимально.

✂ Пусть x^k координаты выбранного в качестве базового объекта, w_k – множество объектов его класса, $d_k(x^k,x^i)$ – расстояние от x^k до x^i, измеренное в метрике, построенной для «базового объекта» x^k.
Тогда упомянутый критерий выглядит следующим образом: $J = \dfrac{\sum_{x_i \in \omega_k} d_k(x^k, x^i)}{\sum_{x_i \notin \omega_k} d_k(x^k, x^i)}$ ✂

В результате применения такого подхода получается N локальных метрик (N – число объектов). В каждой из них можно построить классифицирующую разделяющую поверхность (или, более общо, свое классифицирующее правило), и получить *коллектив решающих правил* (N классификаторов, каждый из которых производит распознавание образов с

точки зрения одного объекта) и применять к этому коллективу соответствующие методы (например, различные варианты «голосования»).

В случае анализа структуры многомерных данных можно, например, выбрать в качестве локальной метрику Махаланобиса. В результате с точки зрения любой «базовой» точки распределение данных будет выглядеть нормальным и изотропным (то есть данные будут расположены внутри многомерного шара с радиусом, равным дисперсии, которая будет равна единице во всех направлениях).

Следует отметить, что после того, как для каждой точки данных построена некоторая локальная метрика, возникает проблема измерения расстояния между объектами, поскольку расстояние, измеренное в локальной метрике одного объекта не будет совпадать с расстоянием, измеренным в локальной метрике другого объекта (будут нарушаться условия независимости расстояния от направления измерения – от 1-го объекта ко 2-ому или наоборот, а также возможно нарушение неравенства треугольника). Для преодоления этой трудности на основе исходных N локальных метрик конструируется некоторая новая, общая для всех то чек данных метрика, в которой в некоторой степени будут присущи свойства локальных. Различные варианты построения таких обобщенных метрик рассмотрены в разделе 2.1.8.

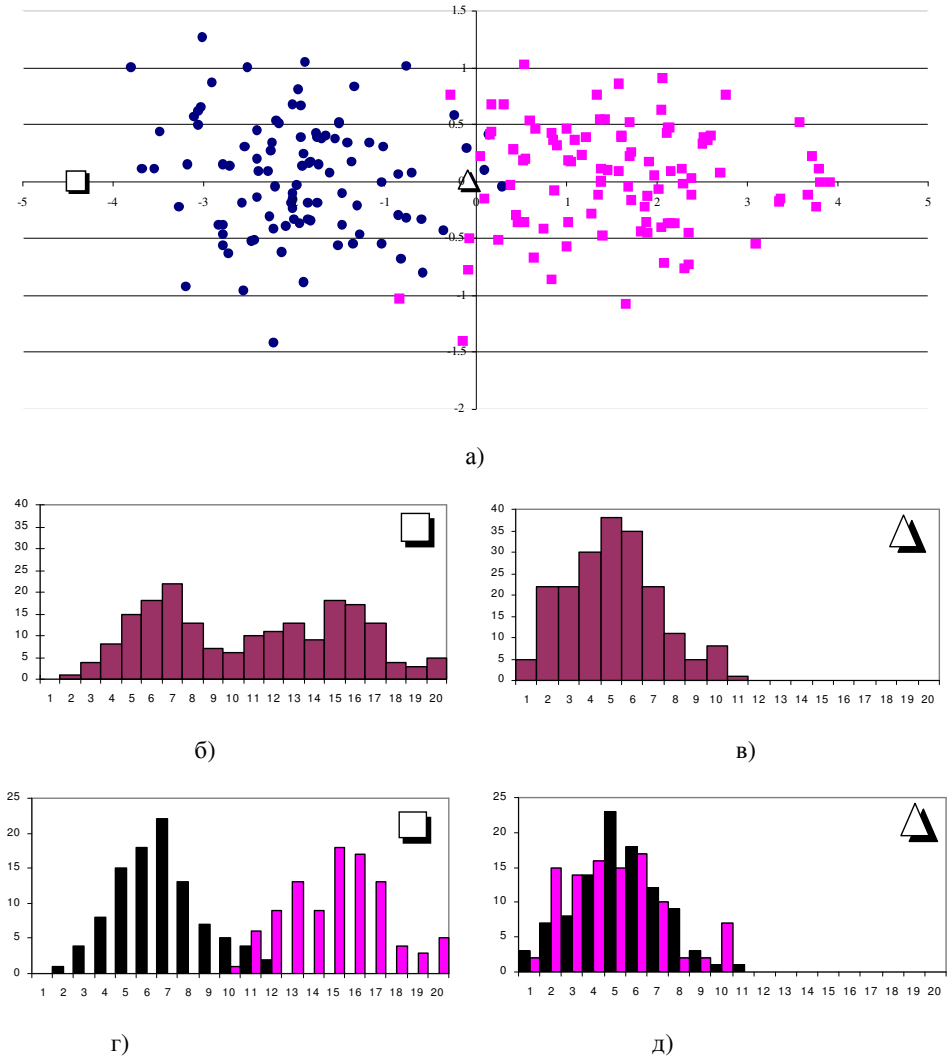


Рис. 14. Построение локальных статистик

а) исходное распределение данных; точки исходно разделены на два почти неперекрывающихся класса; квадратом и треугольником выделены точки на границе и в центре облака соответственно;

б),в) диаграммы распределения расстояний от выделенных объектов до всех остальных (по оси x – расстояния до объектов, по y – их количество); на первой диаграмме можно выделить две вершины диаграммы, соответствующие двум точкам сгущения в облаке данных;

г),д) диаграммы распределения расстояний от выделенных объектов до объектов двух классов; из первой диаграммы видно, что с помощью статистики объекта, помеченного квадратом, можно построить разделяющее классы правило, а для треугольника – нет.

1.3. Данные в виде картинки

1.3.1. Задача визуализации данных

В этом разделе нашей целью является дать обзор тех методов, которые в настоящее время используются для визуального представления сразу всей структуры многомерного набора данных. Для визуализации могут быть использованы 1-, 2- и 3-мерные пространства отображений, но мы в своем рассмотрении практически целиком ограничимся способом визуализации с помощью 2-мерных поверхностей, поскольку именно в таком виде человек воспринимает геометрические структуры наиболее естественно и отношения между объектами выглядят наиболее наглядно.

Под ***визуализацией данных*** мы понимаем такой ***способ представления многомерного распределения данных на двумерной плоскости, при котором, по крайней мере, качественно отражены основные закономерности, присущие исходному распределению – его кластерная структура, топологические особенности, внутренние зависимости между признаками, информация о расположении данных в исходном пространстве*** и т.д. В качестве основных применений методов визуализации можно указать следующие:

а) наглядное представление геометрической метафоры данных;

б) лаконичное описание внутренних закономерностей, заключенных в наборе данных;

в) сжатие информации, заключенной в данных;

г) восстановление пробелов в данных;

д) решение задач прогноза и построения регрессионных зависимостей между признаками.

Мы кратко рассмотрим традиционные методы, решающие поставленную задачу непосредственным и несколько громоздким образом – это *целенаправленное проецирование* данных и *многомерное шкалирование*. Затем обратимся к очень популярному в последнее время способу визуализации с помощью самоорганизующихся карт Кохонена. Глава будет завершена рассмотрением нового подхода, разработанного группой «Нейрокомп» на базе Института Вычислительного Моделирования г.Красноярска и названного **методом упругих карт**.

1.3.2. Методы целенаправленного проецирования в пространства малой размерности

Один из способов поставить задачу представления данных в виде двумерной картинки заключается в следующем: найти такое отображение (способ проецирования) из исходного пространства на двумерную плоскость, которое бы оптимизировало заданный критерий качества – некоторый функционал от координат точек данных до и после процедуры проецирования. Такая постановка задачи лежит в основе совокупности подходов, объединяемых под названием *целенаправленное проецирование* (в зарубежной литературе – projecting pursuit) в пространство малой размерности.

Можно выделить два варианта решения этой задачи:

1. Вид отображения U известен заранее и является, как правило, линейным отображением на плоскость. Оптимизируемый функционал в данном случае называется *проекционным индексом* и обозначается $Q(U,X)$, под X понимается весь набор многомерных данных, Q зависит от параметров отображения. В зависимости от поставленной задачи могут быть использованы следующие проекционные индексы:

а) индекс, минимизирующий расстояние от точек данных до их проекций – и это дает классический метод снижения размерности с помощью главных компонент;
б) индексы, максимизирующие расстояния между кластерами (один из вариантов таких индексов максимизирует энтропию конечного двумерного распределения данных);
в) индексы, максимально разделяющие заранее заданные классы для построения линейного классификатора;
г) индексы, использующиеся для выделения *аномальных наблюдений*, далеко отстоящих от основной массы распределения точек данных;
д) индексы, выделяющие нелинейные структуры в многомерных данных.

Явный вид этих проекционных индексов приведен в литературе по прикладной статистике [2].

2. Вид отображения заранее неизвестен. Тогда оптимизируемый критерий является функцией от набора двумерных координат, приписанной каждой точке данных. Задачей в этом случае является назначить каждой из точек исходного набора данных пару координат таким образом, чтобы минимизировать функционал, описывающий «меру искажения» структуры данных.

Одним из самых популярных является функционал, являющийся аналогом стресса в многомерном шкалировании и описывающий меру искажения взаимных расстояний между точками в исходном и результирующем пространстве отображения.

Остановимся здесь на одном важном для нашего изложения моменте. В разделе, посвященном квазилинейным моделям мы уже сталкивались с ситуацией, когда каждой точке данных можно было бы приписать две координаты (метод главной плоскости). Это позволяет построить в пространстве данных гладкое многообразие, которое обладает свойством обобщать заключенную в данных информацию и служить для лаконичного описания, сжатия информации или для восстановления пробелов в данных. Совершенно аналогично над методом проецирования в пространство меньшей размерности можно надстроить процедуру построения моделирующей двумерной поверхности, вложенной в многомерное пространство признаков.

1.3.3. Многомерное шкалирование

Иногда исходная информация бывает изначально представлена не в виде таблицы типа «объект-признак», а в виде квадратной таблицы удаленностей объектов друг от друга. На пересечении i-ой строки и j-ого столбца в такой таблице стоит оценка расстояний от i-го до j-го объекта. Такой вид представления информации характерен для психологических исследований, когда человеку предлагается оценивать сходство или различие в некоторой системе объектов или понятий.

Таким образом, изначально каждому объекту не сопоставляется никакой координаты в многомерном пространстве и представить такую информацию в виде геометрической метафоры затруднительно. Задача *многомерного шкалирования* заключается в том, чтобы сконструировать распределение данных в пространстве таким образом, чтобы расстояния между объектами в соответствовали исходно заданным в матрице удаленностей. Возникающие координатные оси могут быть интерпретированы как некоторые неявные факторы, значения которых определяют различия объектов между собой. Если попытаться сопоставить каждому объекту пару координат, то в результате мы получим способ визуализации данных.

Различают два основных алгоритма многомерного шкалирования – метрический и неметрический, хотя сами вычислительные процедуры этих алгоритмов практически не отличаются [4,14,43].

1. В основе *метрического многомерного шкалирования* лежит допущение о том, что расстояния в таблице удаленностей соответствуют реальным расстояниям между объектами в конструируемом пространстве признаков.

В линейном методе метрического шкалирования применяется метод главных компонент, но не к исходной матрице расстояний, а к так называемой *дважды центрированной матрице*, в которой среднее значение чисел в любой строке и столбце равно нулю. Дважды центрированная матрица однозначно вычисляется по исходной. После этого существует возможность определить размерность пространства, обеспечивающего *точное* воспроизведение матрицы удаленностей, либо определить эффективную размерность конструируемого пространства признаков, которая обеспечит воспроизведение матрицы удаленностей с заданной точностью.

В нелинейных методах размерность пространства задается изначально и с помощью градиентных методов оптимизируется функционал качества, описывающий меру искажения матрицы удаленностей. Этот функционал, называемый *стрессом*, уже упоминался нами в предыдущем разделе и мы вернемся к нему позже.

2. В *неметрическом многомерном шкалировании* предполагается, что удаленность объектов измерена в ординальной шкале, то есть важны не столько сами численные значения попарных расстояний, сколько их ранговый порядок. Процедуры неметрического шкалирования строят такую геометрическую конфигурацию точек в q-мерном пространстве, чтобы ранговые порядки расстояний совпали, по возможности, с ранговыми порядками исходных расстояний. Для оценки качества выбранных ранговых координат применяется все тот же критерий стресса.

Аналогично традиционному факторному анализу, в многомерном шкалировании существует неоднозначность выбора координат, связанная с тем, что координатную систему в полученном пространстве можно произвольным образом повернуть – расстояния между объектами при этом не изменяются. Как правило, поворот осуществляют таким образом, чтобы либо полученные координатные оси имели максимально наглядную интерпретацию, либо значения определенных признаков оказались максимально скоррелированы.

1.3.4. Вложенные поверхности

Итак, в основе методов целенаправленного проецирования и многомерного шкалирования лежит идея оптимизации некоторого

функционала, который зависит от начального положения точек в пространстве и конечного расположения точек на двумерной плоскости. Выбирая различные виды функционалов, можно строить различные проекции данных, на которых будут подчеркнуты те или иные их особенности. В целом такой подход является достаточно прозрачным и ясным, но при его практическом использовании возникают определенные трудности.

Во-первых, задача оптимизации нелинейной функции является трудной сама по себе. В упомянутых методах используются, как правило, градиентные процедуры, требующие больших вычислительных затрат, которые растут пропорционально квадрату от числа точек данных (нужно вычислять все попарные расстояния в пространстве отображений). Это делает весьма затруднительной практическую реализацию этих алгоритмов для таблиц данных, содержащих большое (порядка нескольких тысяч) число строк.

Во-вторых, оказывается, что выразительная картина многомерного распределения данных, изображенная на двумерной картинке еще не решает всех вопросов, которые может поставить себе исследователь. Заманчива идея наносить на двумерную карту не только сами точки данных, но и разнообразную информацию, сопутствующую данным – например, отображать так или иначе положение точек в исходном пространстве, плотности различных подмножеств, другие непрерывно распределенные величины, заданные в исходном пространстве признаков. Все это подталкивает к мысли использовать как можно полнее тот «фон», на который наносятся данные, а также вид самих точек данных для отображение различной количественной и аттрибутивной информации.

Наконец, после того, как данные нанесены на двумерную плоскость, хотелось бы, чтобы появилась возможность расположить на двумерной плоскости те данные, которые не участвовали в настройке отображения. Это позволило бы, с одной стороны, использовать полученную картину для построения различного рода экспертных систем и решать задачи распознавания образов, с другой – использовать ее для восстановления данных с пробелами.

Таким образом можно подойти к идее использования для визуализации данных и извлечения информации некоторого вспомогательного объекта, который в дальнейшем мы будем называть *картой*. Этот объект представляет из себя ограниченное двумерное нелинейное многообразие, вложенное в многомерное пространство данных таким образом, чтобы служить моделью данных, то есть форма и расположение такого многообразия должна отражать основные особенности распределения множества точек данных.

Простой пример карты данных – плоскость первых двух главных компонент. Как мы уже упоминали, среди всех двумерных плоскостей, вложенных в пространство она служит оптимальным экраном, на котором можно отобразить основные закономерности, присущие данным. В качестве другой, еще более простой (но не оптимальной) карты можно использовать любую координатную плоскость любых двух выбранных координат.

> ✂ Среди различных проекций на пары координатных осей наиболее информативными будут те, где в качестве координат выбираются наиболее значимые признаки, например, те, которые имеют наибольший вес в векторе, задающем направление первой главной компоненты. ✂

Обобщением способа представлять данные с помощью метода главных компонент будет случай, когда карта может иметь любую нелинейную форму, не используя в процессе построения карты никаких гипотез о распределении данных. На пути создания такой нелинейной модели данных необходимо ответить на следующие вопросы:

1. Как описывать расположение карты в пространстве?

Для того, чтобы описывать в многомерном пространстве вложенное двумерное многообразие, используют обычно вектор-функцию $\boldsymbol{r} = \boldsymbol{r}(u,v)$ от двух координат u, v, которые называются *внутренними* координатами или параметрами. Линии, вдоль которых одна из внутренних координат принимает постоянное значение, задают на поверхности внутреннюю координатную сетку. Таким образом, любая точка на поверхности задается, с одной стороны, только двумя внутренними координатами (именно поэтому размерность многообразия, задаваемого формулой $\boldsymbol{r} = \boldsymbol{r}(u,v)$ равна по построению двум), а с другой стороны, будучи точкой в m-мерном пространстве имеет m значений координат в исходном пространстве.

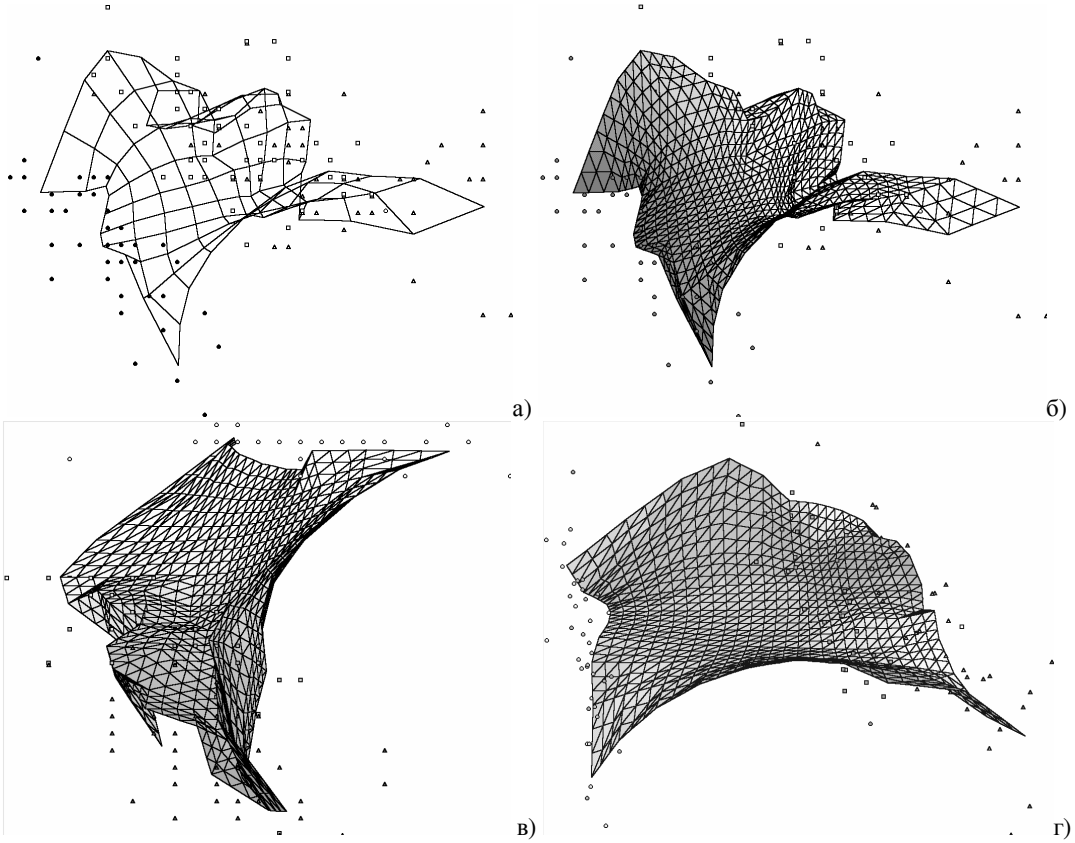


Рис. 15. Вид построенной карты с точки зрения различных двумерных плоскостей-экранов.

На проекциях показаны точки данных, соответствующие упоминавшейся базе данных по цветкам ириса.

а) вид построенной сетки на координатной плоскости «длина лепестка – ширина лепестка»;

б) вид карты, для которой применена процедура кусочно-линейной интерполяции между узлами;

в) вид карты в координатах «ширина лепестка – ширина чашелистика»;

Для вычислительных процедур гораздо удобнее производить операции не с самим многообразием, а с его точечной аппроксимацией, задаваемой с помощью сетки узлов (удобно, если в этих узлах значения внутренних координат принимают целые значения). Для описания положения прямоугольной сетки узлов в пространстве достаточно $m \cdot p \cdot q$ чисел, где m – размерность пространства, а p и q – число узлов прямоугольной сетки по вертикали и горизонтали. Если число узлов сетки гораздо меньше числа точек данных, то используя такую сетку в качестве модели данных, можно получить сжатие информации, заключенной в

данных, с точностью, зависящей от способа построения карты и особенностями структуры данных.

Изначально карта может быть задана с помощью плоской равномерной сетки узлов, как-то размещенных в пространстве признаков. Под действием тех или иных вычислительных процедур карта может искривляться, прилегая к данным и отражая особенности их структуры.

После того, как получена точечная аппроксимация многообразия, для того, чтобы восстановить карту нужно воспользоваться подходящей процедурой интерполяции между узлами. Самым простым вариантом интерполяции является кусочно-линейная интерполяция. Для ее построения изначально на сетке реализуется тот или иной вариант триангуляции, в результате чего карта состоит из отдельных треугольных кусков плоскостей.

На рис.15 показано, как может выглядеть построенная карта с точки зрения различных плоских двумерных экранов, расположенных в пространстве – разных координатных плоскостей и плоскости главных компонент.

2. Каким образом сопоставить каждой точке данных точку на карте?

После того, как многообразие построено, для визуализации данных необходимо указать правило, с помощью которого данные из исходного пространства признаков переносятся на карту. Предполагается, что длина вектора переноса не будет слишком велика, поскольку карта аппроксимирует данные и достаточно плотно в среднем к ним прилегает.

Простейшим способом переноса или *проецирования* является сопоставление каждой точке данных ближайшего узла сетки. Такой способ даже не требует доопределения сетки до многообразия и разбивает все множество данных на $p \times q$ подмножеств – *таксонов*, внутри каждого из них ближайшим является один и тот же узел карты. В некоторых задачах такой способ проектирования является приемлемым, однако, он не дает непрерывного отображения пространства данных на двумерное многообразие – при переходе от узла к узлу функция отображения имеет разрыв – и поэтому такое проектирование называется *кусочно-постоянным*.

Другой идеей, которая может быть применена при проецировании, является сопоставление точке данных *ближайшей точки* на карте (а не ближайшего узла!). В случае гладкого многообразия нахождение такой точки может быть связано с определенными вычислительными трудностями, в случае же упомянутой линейной интерполяции между узлами достаточно просто ближайшую точку многообразия определить

достаточно просто. Соответствующий алгоритм приведен в разделе 2.5.6, а здесь укажем, что ближайшей точкой карты может оказаться точка внутри треугольного куска плоскости, образующего *грань* карты или точка внутри отрезка, соединяющего два соседних узла, образующего *ребро* карты, а также ближайшей точкой может оказаться узел (*вершина*) карты. Соответственно, в случае кусочно-линейного многообразия пространство вокруг карты разбивается на области, для которых ближайшей является узел, ребро или грань построенной карты. Такой способ проецирования оказывается кусочно-линейным.

Для проецирования многомерных данных на двумерную кусочно-линейную поверхность могут применяться другие способы кусочно-линейного проецирования, например, центральное проецирование. Центр проекции определяется с помощью координат ближайшего узла сетки и прилегающих к нему соседей, то есть для каждого таксона данных будет найден свой центр проекции.

Возможно построение кусочно-гладкого проектора с помощью построения в каждом узле карты некоторого двумерная гладкая поверхность, аппроксимирующая в окрестности узла карту. Наиболее прост случай построения поверхности второго порядка, однако, в этом случае не удается сшить построенные куски поверхностей и в результате построенное многообразие будет выглядеть как «папье-маше» – на каждый из узлов приклеивается поверхность второго порядка с вершиной или седловой точкой в узле.

На рис.16 сравниваются два способа проецирования – в ближайший узел и в ближайшую точку карты. На нем же показана карта в своих внутренних координатах (развернутая на плоскости) и расположение точек, полученных в результате этих способов проецирования.

Итак, на построенной карте можно разместить точки данных. Кроме этого, карта сама по себе обладает рядом свойств: прилегая к данным, карта в одних областях пространства сжимается, в других растягивается, и это задает на карте двумерную метрику, разные точки карты имеют разные координаты в пространстве, карта по-разному прилегает к данным в разных областях и т.д. Таким образом, карта сама по себе служит носителем разнообразной информации.

Теперь мы обратимся к конкретным способам построения моделирующих карт.

1.4. Самоорганизующиеся карты Кохонена и их приложения

Тойво Кохонен предложил нейросетевую архитектуру для автоматической кластеризации (классификации без учителя), в которой учитывается информация о взаимном расположении нейронов, которые образуют решетку. Сигнал в такую нейросеть поступает сразу на все нейроны, а веса соответствующих синапсов интерпретируются как координаты положения узла и выходной сигнал формируется по принципу «победитель забирает все» - то есть ненулевой выходной сигнал имеет нейрон, ближайший (в смысле весов синапсов) к подаваемому на вход объекту.. В процессе обучения веса синапсов настраиваются таким образом, чтобы узлы решетки «располагались» в местах локальных сгущений данных, то есть описывали кластерную структуру облака данных, с другой стороны, связи между нейронами соответствуют отношениям соседства между соответствующими кластерами в пространстве признаков.

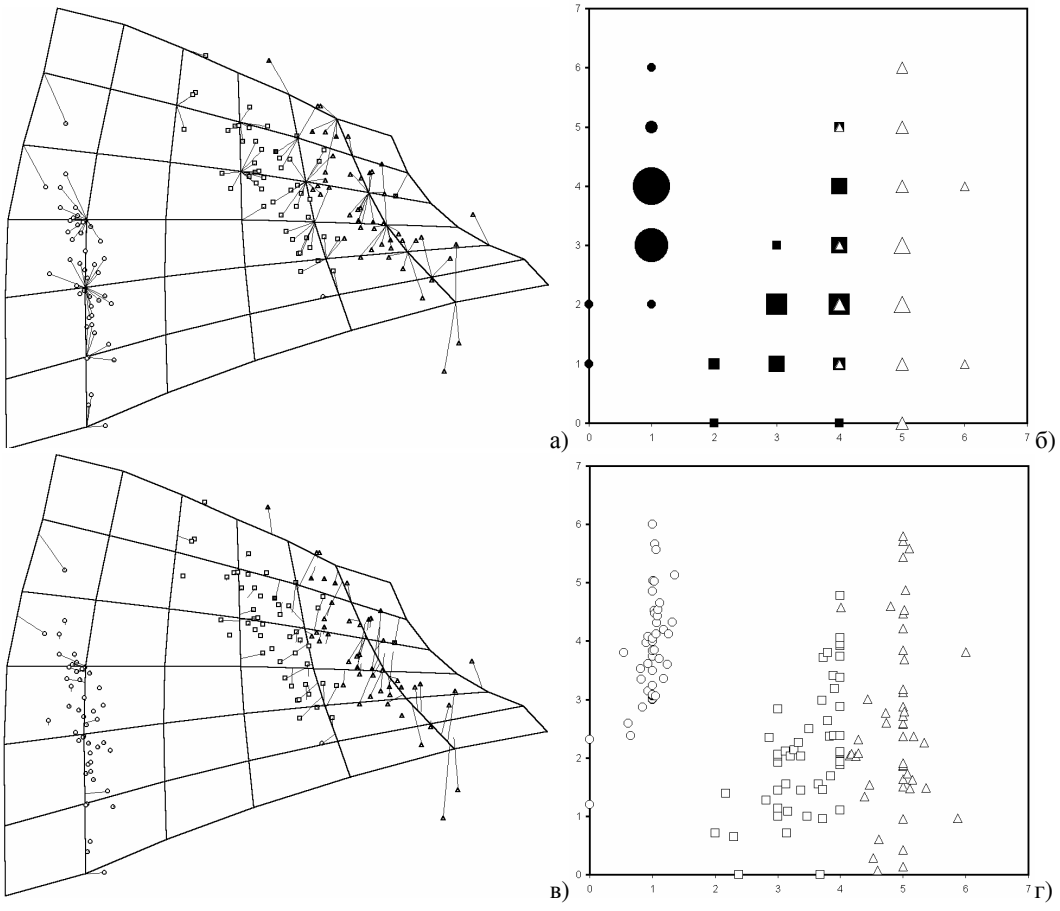


Рис. 16. Сравнение способов проецирования данных в ближайший узел и ближайшую точку карты (база данных по цветкам ириса).
а),б) проектирование в ближайший узел, на развертке условно изображено количество точек того или иного класса, попавших в узел; видно, что в четырех узлах после проектирования оказываются точки разных классов;
в),г) проектирование в ближайшую точку, на развертке видно, что разделение на классы можно сделать более четким (а также выделить «истинных» ренегатов класса).

Несмотря на то, что самоорганизующиеся карты Кохонена (СОК или SOM – Self-Organizing Maps или SOFM – Self-Organizing Feature Maps) изначально были описаны на нейросетевом языке, нам будет удобно рассматривать такие карты как двумерные сетки узлов, размещенных в многомерном пространстве, не прибегая к нейросетевой терминологии. Тем не менее, следует держать в уме то, что, если когда-нибудь алгоритм SOM будет воплощаться на аппаратном уровне, то для реализации высокоэффективных параллельных схем вычислений нужно будет вспомнить о изначальной нейросетевой архитектуре.

Итак, изначально SOM представляет из себя сетку из узлов, соединенный между собой связями. Кохонен рассматривал два варианта соединения узлов – в прямоугольную и гексагональную сетку (см. рис. 17) – отличие состоит в том, что в прямоугольной сетке каждый узел соединен с 4-мя соседними, а в гексагональной – с 6-ю ближайщими узлами. Таким образом, для двух таких сеток процесс построения SOM отличается лишь в том месте, где перебираются ближайшие к данному узлу соседи.

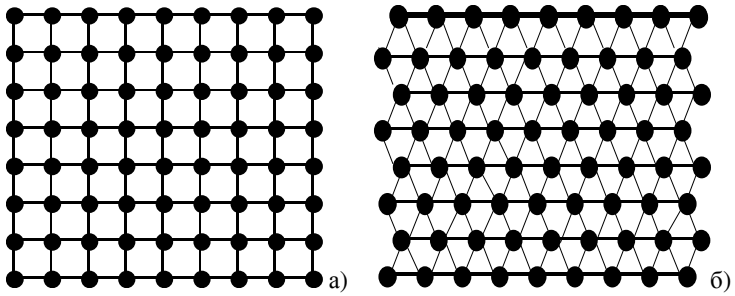


Рис. 17. Два варианта расположения узлов сетки SOM.
а) прямоугольная сетка – каждый узел (кроме краевых) имеет 4 ближайших соседа
б) гексагональная сетка – каждый узел (кроме краевых) имеет 6 ближайших соседей

Начальное положение сетки выбирается произвольным образом. А авторском пакете SOM_PAK предлагаются варианты случайного начального расположения узлов в пространстве и вариант расположения узлов в плоскости.

После этого узлы начинают перемещаться в пространстве согласно следующему алгоритму:

1. Случайным образом выбирается точка данных x.

2. Определяется ближайший к x узел карты r_{ij} (BMU – Best Matching Unit).

3. Этот узел перемещается на заданный шаг по направлению к x. Однако, он перемещается не один, а увлекает за собой определенное количество ближайших узлов из некоторой окрестности на карте. Поясним сказанное: если радиус окрестности равен 1, то вместе с ближайшим узлом r_{ij} по направлению к x двигаются 4 его соседа по карте в случае прямоугольной сетки и 6 соседей в случае гексагональной сетки.

В настройке карты различают два этапа – этап грубой (*ordering*) и этап тонкой (*fine-tuning*) настройки. На первом этапе выбираются большие значения окрестностей и движение узлов носит коллективный характер – в результате карта «расправляется» и грубым образом отражает структуру данных; на этапе тонкой настройки радиус окрестности равен 1–2 и настраиваются уже индивидуальные положения узлов.

О характере движения следует заметить следующее: обычно он настраивается так, чтобы во всем из всех двигающихся узлов наиболее сильно смещается центральный – ближайший к точке данных – узел, а остальные испытывают тем меньшие смещения, чем дальше они от центра узла (см. рис. 18)

Это соответствует так называемым затухающим функциям соседства (neighborhood function). Если это не так и все соседи из окружения испытывают равные смещения, то такая функция соседства называется пузырьковой (bubble) и для нее характерно большее число актов перемещения для обучения и менее гладкая сетка.

Кроме этого, величина смещения равномерно затухает со временем, то есть она велика в начале каждого из этапов обучения и близка к нулю в конце.

4. Алгоритм повторяется определенное число тактов. На первом этапе число тактов выбирается порядка тысяч, на втором – десятка тысяч (понятно, что число шагов может сильно изменяться в зависимости от задачи).

Отметим, что предложенный алгоритм не использует явно никакого критерия оптимизации. Хотя ясно, что, по крайней мере, будет уменьшаться среднее расстояние от каждой точки данных до ближайшего узла карты. Средний квадрат такого расстояния в пакете SOM_PAK служит критерием качества построенной карты. В этом пакете, как правило, строится несколько десятков карт, из которых выбирается лучшая согласно упомянутому критерию.

В результате действия алгоритма строится карта, то есть двумерная сетка узлов, размещенных в многомерном пространстве. Для того, чтобы изобразить их положение используются различные средства. Одно из них – такое раскрашивание карты, когда цвет отражает расстояние между узлами

(см. рис.19) Узлы могут при этом маркироваться, если в исходном наборе данных имеются характерные точки данных с присвоенными метками – тогда эта метка ставится на узел, ближайший к данной точке.

Кроме того, сетка узлов может быть раскрашена согласно значению того или иного признака, причем это может быть признак, который не использовался при обучении в качестве координаты пространства (см. рис. 20)

Популярным способом изображения самих данных являются диаграммы Хинтона, когда на каждом узле сетки изображается квадрат, размер которого пропорционален числу точек данных, ближайших к данному узлу, а оттенок соответствует значению соответствующего отображаемого признака (см. рис. 21).

Если рассматривать самоорганизующуюся карту Кохонена как точечную аппроксимацию некоторого многообразия, то следует отметить следующее. Если считать, что узлы карты соответствуют равномерной сетке значений внутренних координат многообразия, то внутренняя метрика такого многообразия характеризуется сильной неравномерностью, – в местах скопления данных оказывается много узлов, и там многообразие получается «скомканным», а в местах, где данные отсутствуют узлов крайне мало, и там многообразие растянуто (см. рис. 26). Это означает, что после того, как многообразие будет «расправлено» на двумерной плоскости и изображено в равномерной сетке внутренних координат, точки данных, перенесенные на карту, окажутся расположены на ней более или менее равномерно. Второе замечание касается самого способа переноса точек из пространства на карту. Поскольку каждой точке данных сопоставляется ближайший узел карты, то, как уже упоминалось, такой вид проектирования является кусочно-постоянным. Это приводит к тому, что средний квадрат расстояния от точки данных до ее проекции на карте сильно зависит от количества узлов в сетке.

Примеры применения SOM.

✋ 1. Картографирование коллекций текстов.

Принципы представления большого количества текстов в виде частотных таблиц были рассмотрены в разделе 1.1. Визуализация таких таблиц с помощью SOM используется в Интернете для автоматического упорядочивания больших коллекций текстов.

Проект WebSOM визуализирует текстовую информацию из нескольких миллионов статей групп новостей UseNet. Полученная в результате карта показана на рис.22а и позволяет ориентироваться в море информации, собранной в этих статьях по тематикам. В отличие от простого каталога, такой способ представления рубрик обладает наглядностью и определенной ассоциативной непрерывностью – смежные темы занимают на карте смежные позиции.

Группой Терехова была предпринята аналогичная попытка визуализации тематического содержания статей и тезисов, опубликованных за несколько лет на российских конференциях по нейроинформатике. В результате был создан удобный интерфейс (см. рис.22б), позволяющий пользователю по заданному ключу или автору получить визуальную картинку (диаграмму Хинтона) распределения интересующей его информации в базе статей и быстро сориентироваться в большом количестве публикаций.

✋ 2. Раскраска географической карты по экономической «схожести»

В Интернете (http://www.cis.hut.fi/nnrc/worldmap.html) есть пример использования SOM для построения некоего комплексного экономического показателя, по которому раскрашена обычная географическая карта. В результате станы со сходной экономической ситуацией изображены на карте цветами, близкими по спектру. (см. рис. 23)

✋ 3. Анализ основных экономических показателей крупнейших российских предприятий.

Шумским С.А., Кочкиным А.Н. проанализирована упомянутая в разделе 1.1 таблица основных экономических показателей для 200 крупнейших предприятий России. Для визуализации были использованы самоорганизующиеся карты Кохонена и диаграммы Хинтона.

✋ 4. Анализ фондового рынка

На рис.24 приведен пример анализа состояния фондового рынка с помощью самоорганизующихся карт, позволяющий одновременно оценить состояние рынка продаж по нескольким показателям (автором использовались Индекс релятивной силы (RSI), Степень изменчивости цены (Price ROC), Индикатор Вильямса (William's, %R), Индекс ценовых диапазонов (CCI), Осциллятор "Рэинбоу" (Rainbow Oscillator) и

Стандартная ошибка (Standard Error)). Авторы примера (http://www.com2com.ru/dav/index.htm) считают, что такой подход является эффективным средством помощи при принятии решений на фондовом рынке («продавать/покупать»).

В литературе [51,57,59,61,63,69,77] приведено большое количество публикаций, в которых сообщается, что SOM применяется для автоматической классификации изображений, сжатия изображений, анализа временных рядов, задач ассоциативного поиска и т.д.

Со времени создания алгоритм SOM был подвергнут тщательному теоретическому исследованию. В литературе [60,62,64,71] описано большое количество модификаций первоначального алгоритма, краткий обзор этих идей сделан в разделе 2.4. В основном, модификации приводят к тому, что задаваемая «руками» в первоначальном варианте алгоритма величина радиуса соседства определяется и настраивается автоматически с помощью тех или иных эвристических приемов, в результате чего самоорганизующиеся карты приобретают те или иные дополнительные свойства (регулярности, точности и т.д.).

1.5. Упругие карты

Многочисленные примеры использования идеологии SOM показывают, что визуализация данных при помощи вложенных в многомерное пространство двумерных сеток достаточно эффективна как средство анализа структуры многомерного облака данных.

Еще раз подчеркнем «технологические» особенности применения этой идеологии:

1) В настройке сети не используется оптимизация какого-либо функционала. Единственное, чего можно ожидать –это уменьшение среднего расстояния от точки данных до ближайшего к ней узла сетки. С одной стороны, тенденция к такому уменьшению заложена в самом алгоритме построения карты, с другой стороны утверждать достигает ли построенная сетка в действительности хотя бы локального минимума среднеквадратичной ошибки наверняка нельзя.

2) Проектирование данных осуществляется в ближайший узел карты, таким образом, этот узел является представителем своего локального мини-кластера данных.

3) Узлы построенной карты распределены в пространстве неравномерно. Если в облаке данных есть сгущения, то в них окажется больше узлов, чем в свободных от точек областях пространства. Таким образом, самоорганизующаяся карта осуществляет сокращение описания множества точек данных с помощью замены локальных сгущений в облаке данных на несколько «узлов-представителей», количество которых пропорционально размерам сгущения (*линейное векторное квантование* - LVQ) и одновременно связывает эти узлы в двумерную сетку, что позволяет расположить их на плоскости для визуального анализа отношений и расстояний между сгущениями.

Нами предлагается иная технология построения вложенных многообразий, которые мы будем называть *упругими картами*. Задачу построения вложенного многообразия, в отличие от SOM, мы поставим как оптимизационную, что соответствует общей методологической установке в прикладной статистике (см., например,[3,4]). То есть построенная карта будет решением задачи на оптимизацию заданного функционала от взаимного расположения карты и данных.

В принципе, в качестве критерия оптимальности мог бы быть использован любой из упомянутых проекционных индексов из методов целенаправленного проецирования (например, критерий стресса), однако, как мы уже указывали, во-первых, такая постановка задачи по существу нелинейна и ее применение требует использования градиентных процедур оптимизации, что само по себе составляет проблему для большого количества точек; во-вторых, как мы уже упоминали, карта как таковая строится уже после решения задачи оптимизации, и ее расположение в пространстве оказывается сильно неоднозначным, так как одни и те же результаты проецирования могут быть получены с использованием различных карт.

Попробуем сформулировать такой критерий оптимальности, в который бы входили не начальные и конечные положения точек (до и после проецирования на карту), а положения самих узлов карты в пространстве относительно данных. Тогда существенно снизится размерность задачи оптимизации – с размерности mN в случае задач целенаправленного проецирования до mpq (p, q – число узлов прямоугольной сетки по горизонтали и вертикали, m – размерность пространства, N – число точек). Кроме того, вид функционала можно попробовать сделать квадратичным по положению координат узлов карты для того, чтобы в результате решать систему линейных уравнений для нахождения минимума критерия.

Ясно, что в критерий должно входить среднее расстояние от точки данных до ближайшего узла карты. Мы должны минимизировать его для того, чтобы карта моделировала данные. С одной стороны, такой критерий в случае нормального распределения заставит все узлы карты разместиться в плоскости первых двух главных компонент, и, таким образом, построенная карта может служить обобщением метода главных компонент . С другой стороны, если в качестве меры длины выбрать обычное евклидово расстояние, то эта часть критерия оптимальности будет квадратична по координатам положения узлов, что весьма желательно.

Однако минимизировать указанный критерий можно бесконечным числом способов, строя в том числе и такие карты, узлы которых будут совершенно неупорядочены. В самоорганизующихся картах упорядочивание достигается за счет того, что между узлами существуют связи и каждый узел, двигаясь в пространстве, подтягивает за собой соседей. На основании этого соображения можно добавить в критерий требование упругости карты. Что это означает?

Вообразим себе упругую двумерную поверхность – например, кусок упругой пластинки. Такая пластина при различных деформациях стремиться восстановить свою первоначальную форму. Однако, деформировать пластину можно двумя способами – растягивая ее «вдоль» и изгибая «поперек» (см. рис.25) – и в одном случае она стремиться сохранить свою длину, в другом – свою плоскую форму. Назовем возникающие при деформациях силы в пластике упругостью по отношению к растяжению и упругостью по отношению к изгибу.

Теперь, если мы потребуем, чтобы наша сетка обладала обоими этими свойствами, то в минимизируемый критерий необходимо добавить меру суммарного растяжения сетки и меру суммарного изгиба. Такие меры в самом простом варианте описания упругих сил также оказываются квадратичными по отношению к координатам положения узлов сетки в пространстве.

Складывая вместе все три упомянутые меры (средний квадрат расстояния до узла и две меры упругости) с определенными весами, мы получаем общий критерий, благодаря которому сетка, с одной стороны, будет притягиваться к точкам данных, с другой – стремиться минимизировать свое растяжение и принять максимально гладкую форму (стать более регулярной). Конкретный вид рассмотренного критерия и алгоритм его минимизации подробно рассмотрены в разделе 2.5.

У построенной сетки остаются неопределенными два параметра – веса суммы – они могут быть интерпретированы как коэффициенты упругости сетки по отношению к растяжению и изгибу. Их приходится задавать «руками», но можно построить и такую процедуру настройки

этих параметров, чтобы сетка приняла желаемый вид. С одной стороны, чем более упруга сетка, тем более гладкую модель данных она собой представляет, но и тем хуже она описывает малые отклонения от предполагаемого закона; с другой – чем менее упруга сетка, тем она точнее описывает данные, но при этом воспроизводятся и все случайные шумы, которые обычно присущи реальным данным, при этом ухудшается способность модели к обобщению информации.

Самой простой идеей настройки коэффициентов упругости является последовательное их уменьшение от больших значений к малым до тех пор, пока не будет достигнута необходимая точность. При больших значениях коэффициентов упругости узлы карты практически находятся в одной плоскости, и это будет плоскость главных компонент, на которой, кстати, их можно изначально и разместить. Далее карта будет приобретать криволинейную форму, аппроксимируя распределение данных.

Что будет, если упругость карты станет равна нулю? Рассмотрим те точки, которые окружают определенный узел карты и находятся ближе к нему, чем к любым другим узлам. Как и прежде, назовем множество таких точек *таксоном* данного узла. Если таксон не содержит ни одной точки, то узел останется на месте, если таксон состоит из единственной точки, то узел переместится в нее, если таксон состоит из нескольких точек, то узел разместится в точке среднего значения всех координат точек таксона (см. рис.25б) В этом смысле карта с нулевой упругостью схожа с картой Кохонена – положения ее узлов совпадают с центрами локальных сгущений. Однако, упругая карта в силу способа построения отличается от самоорганизующейся карты – так число узлов, размещенных в областях сгущения данных не будет пропорционально мощности сгущений. Дело в том, что некоторым узлам всегда будет «энергетически выгодно» расположиться в пространстве между сгущениями – и, таким образом, сетка окажется более-менее равномерной, а не так сильно деформированной, как в случае SOM.

Так как узлы сетки притягиваются не только к точкам данных, но и к друг другу, то некоторые данные могут располагаться в пространстве между узлами, на сравнительно большом удалении от них. Отсюда следует, что процедура проецирования данных в ближайший узел сетки может давать большие значения ошибки, по сравнению с Кохоненовскими картами. Таким образом, для упругих карт особенно актуально становится использование кусочно-линейных методов проектирования, – например, проектирование в ближайшую точку (не узел!!) карты. Но, как уже было сказано, для этого необходимо интерполировать многообразие в промежутках между узлами, например, это можно сделать кусочно-

линейным способом, сделав «граненую» карту (см. рис.26), для которой найти ближайшую точку достаточно просто.

1.6. Картографирование данных

После того, как карта построена, и точки данных перенесены из пространства признаков на поверхность карты, можно пользоваться ее изначальной двумерностью, расправив ее складки и развернув на плоскости. Теперь каждая точка данных имеет две координаты во внутренней системе координат на карте.

Плоскую карту можно изобразить двумя способами (см. рис.26). В первом можно попытаться максимально воспроизвести те расстояния между узлами, которые были в исходном пространстве, получив при этом двумерную *криволинейную координатную сетку*. Понятно, что сделать это совсем без искажений не удастся, поскольку исходная сетка вложена в многомерное пространство (расправить сетку без искажений нельзя именно на плоскости, известно, что в евклидово трехмерное пространство можно вложить двумерную поверхность с произвольной метрикой). Во втором можно изображать проекции данных в исходных внутренних координатах. Тогда «развернутая» карта просто имеет вид прямоугольника.

Таким образом, мы получим своеобразную подложку, которая сформируется при помощи данных под влиянием двух конкурирующих тенденций – стремлении узлов притянуться к данным, и стремлении к минимальному растяжению и изгибу сетки. На этой подложке мы можем изображать различную информацию. С помощью линий уровня и различных раскрасок можно изображать значения тех или иных интересующих исследователя величин и объектов. Каждый из способов раскраски предстанет в конечном виде как *информационный слой*, являющийся аналогом слоя в ставших традиционными ГИС-технологиях (ГИС – *геоинформационные системы* – средства компьютерного представления информации, привязанной к географической карте). Разница состоит лишь в том, что вместо обычной географической карты используются подложка – двумерное многообразие, вложенное в многомерное пространство данных. Подобно тому, как на географической карте рядом оказываются объекты с близкими значениями географических координат, так и на построенной подложке рядом располагаются объекты с близкими значениями признаков в исходном пространстве. Это позволяет «один к одному» применять весь богатейший арсенал средств ГИС. В результате для применения ГИС-технологий открывается *возможность картографирования данных произвольной природы*, представленной в виде

таблиц. Окончательный результат применения ГИС-идеологии – атлас тематических раскрасок, дающих представление о внутренней структуре данных.

Какие раскраски – информационные слои – могут быть включены в этот атлас? Или, другими словами, что можно изобразить на построенной карте?

Во-первых, можно изобразить *сами данные*. Данные можно изображать точками, однако, эффективно иметь возможность отображать в местах проекций точек данных разнообразную связанную с ними информацию. Для «увеличения размерности» точек данных могут быть использованы следующие приемы:

а) использование цвета, размера и формы для изображения точек данных; это дает возможность отражать три дополнительных измерения, связанных с точками: цвет и размер позволяют изображать количественные (непрерывные) шкалы, форма – номинальные шкалы признаков;

б) использование сложных изображений – например, точку можно изображать круговой диаграммой, на которой цветами изображено соотношение между значениями координат признаков, а размер отражает абсолютные величины координат (см. рис. 27)

Во-вторых, на карте с помощью линий уровня можно изображать значения любого функционала, заданного в пространстве признаков. Естественно, эти значения будут вычисляться в точках размещения самой карты. В качестве таких функционалов можно использовать следующие величины:

а) значения координаты какого-либо признака – это самый простой тип раскраски, который позволяет выделить на карте области с определенным значением того или иного признака и размещение в них точек данных; понятно, что таких раскрасок будет столько, сколько признаков было в исходном пространстве, то есть число раскрасок по признакам равно размерности пространства данных, и для случая высокой размерности исследователь может просто запутаться в большом количестве картинок. Для того, чтобы предоставить исследователю наиболее содержательные и информативные раскраски, необходимо иметь возможность предварительного отбора признаков по их значимости;

б) простые функции от пары признаков – например, их разность или отношение; смысл таких раскрасок может заключаться в том, чтобы сравнить раскраски по значениям нескольких признаков; например, если раскраски по значениям координат двух признаков схожи, то это указывает на их сильную скореллированность; и наоборот, сравнение

раскрасок позволяет выделить те области пространства и точки данных, для которых корреляционная зависимость нарушается;

в) сложные функционалы от координат признаков – например, в каждой точке карты каким-либо непараметрическим способом можно оценить многомерную плотность распределения данных. При этом можно изображать раскраску, отражающую распределение как всего облака данных, так и какого-либо его подмножества, или более содержательную раскраску по относительной плотности подмножества, равной отношению значения плотности подмножества к общей плотности всего множества;

> ✂ Смысл вычисления относительной плотности подмножества заключается в следующем. Если в какой-либо области пространства содержится мало точек какого-то класса, это еще не означает, что для этой области вообще вероятность появления точек этого класса мала, – возможно, что в обучающем множестве также находилось мало данных из этой области. Говоря о «малой» или «большой» плотности подмножества в какой-либо точке, всегда необходимо указывать, с чем эта плотность сравнивается. ✂

С другой стороны, плотность данных и их подмножеств можно рассчитывать уже в двумерном пространстве самой карты – такую плотность можно назвать двумерной.

г) на карте в виде непрерывных раскрасок можно изображать свойства самой карты; например, цветом можно изобразить области ее наибольшего растяжения и сжатия в исходном пространстве; в частности, можно изобразить значения метрических коэффициентов на поверхности карты (понятно, что в случае кусочно-линейной карты эти значения постоянны в пределах одной плоской грани карты);

д) на карте можно изображать разнообразные поля раскрасок, иллюстрирующие взаимные свойства карты и данных, отражающие способность построенной карты служить моделью данных; одна из таких раскрасок – расстояние от точки карты в исходном пространстве признаков до ближайшей точки данных; так на карте можно увидеть 1) насколько сильно узлы карты удалены друг от друга и от скоплений данных и 2) те области пространства, где карта неплотно прилегает к данным и, следовательно, не может служить в этих областях хорошей моделью множества точек.

Другой вариант раскраски поможет исследователю оценить, насколько хорошо сохраняются отношения соседства после проецирования точек данных на карту. Дело в том, что любое проектирование в

пространство меньшей размерности чревато *изменением топологических особенностей* исходного множества – далекие точки в многомерном пространстве могут оказаться близкими на двумерной карте, и, наоборот, точки, близкие в исходном пространстве могут оказаться разнесены на карте на далекие расстояния (см. рис. 28). Такие несоответствия можно назвать *искажениями структуры данных*.

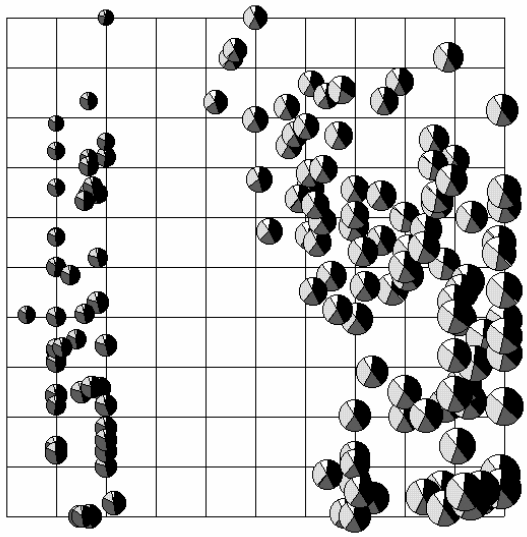


Рис. 27. Использование вида точек-проекций данных для увеличения «размерности» точек данных.

На рисунке изображены проекции точек данных для визуализации базы данных «Ирис». Каждая точка данных представлена в виде круговой диаграммы. Черный цвет изображает значение признака «длина лепестка», темно-серый – «ширина лепестка», светло-серый – «длина чашелистика», белый – «ширина чашелистика».

Один из способов наглядно изобразить, каким образом искажается структура данных после проецирования на двумерную поверхность заключается в следующем. В каждой точке карты вычисляется множество K1 ближайших к данной точке соседних точек в исходном пространстве и множество K2 ближайших соседей в двумерном пространстве карты после проецирования. Мощность пересечения K1 и K2 (число совпадающих точек) может служить мерой сохранения отношений соседства между точками данных после их проецирования. Чем меньше это число, тем хуже передаются топологические особенности исследуемого множества объектов. С помощью таких раскрасок исследователь может выделить те

области пространства, в которых карта в упомянутом смысле плохо отражает структуру данных. Это означает, что в таких областях моделирование и визуализация данных с помощью двумерных поверхностей не имеет большого смысла.

✂ Некоторыми авторами специально исследовалась способность карт Кохонена к отражению топологической структуры исходного множества данных. Например, в [64] предлагается для оценивания качества построения самоорганизующейся карты использовать наряду с MSE (Mean Square Error – среднеквадратичная ошибка) еще и так называемую *топографическую ошибку*. Вычисляется она следующим образом: Для каждой точки данных находится ближайший узел на карте r_1 и второй по близости – узел r_2. Если на карте эти два узла являются смежными, то такая точка пропускается, если нет – то считается, что проекция такой точки *неустойчива* – при небольшом изменении карты или положения точки перемещение ее проекции на карте может произойти скачком. Относительное число таких неустойчивых точек и называется топографической ошибкой. Для каждой из неустойчивых точек можно оценить «меру неустойчивости» – выяснив, насколько далеко отстоят друг от друга узлы r_1 и r_2. Построив гистограмму распределения неустойчивых точек по «мере неустойчивости», получаем изображение так называемой *функции топографической ошибки*. ✂

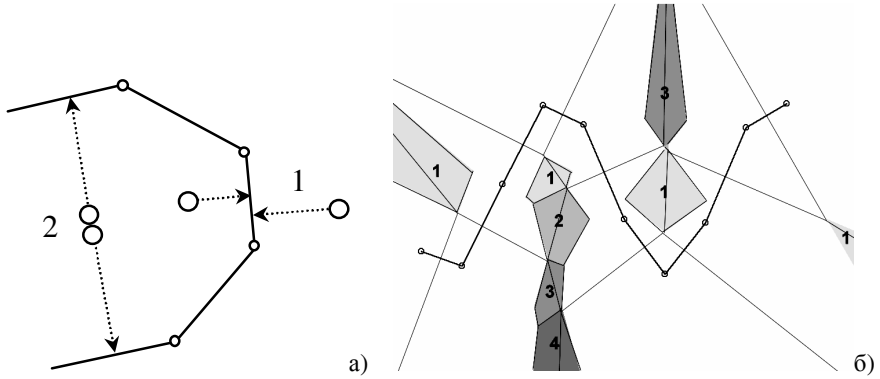


Рис. 28. Искажения структуры данных при проецировании на карту.

а) 1 - далекие точки в пространстве могут оказаться на карте рядом (плохое разрешение); 2 – близкие точки могут оказаться разнесенными на карте на большое расстояние (искажение топологии данных);

б) искажение топологии при проецировании: толстая линия – одномерная карта на плоскости, если точка данных находится в закрашенной области, то она окажется неустойчивой – ближайший (BMU) и второй по близости (BMU2) узел карты не являются смежными, цифрой обозначена степень неустойчивости – насколько далеки оказались друг от друга узлы BMU и BMU2; на картину областей неустойчивости тонкими линиями нанесены границы ячеек Вороного.

1.7. Мультикартирование и восстановление данных

Мы уже подчеркивали, что при создании модели данных необходимо найти компромисс между двумя свойствами модели – свойством воспроизводить исходные данные с высокой точностью и свойством иметь обобщающую способность, то есть отражать не случайные, а существенные характеристики набора данных, что позволяет использовать модель для тех данных, которые не участвовали в настройке модели. На языке упругих карт это означает, что, с одной стороны, карта должна быть достаточно «мягкой», чтобы близко прилегать к точкам данных и аппроксимировать их с достаточной точностью, с другой, карта должна быть упругой, чтобы быть гладкой и не аппроксимировать случайный шум (но при этом снижается точность аппроксимации). Если карта используется не только для визуализации данных, но и для построения регрессионных зависимостей одних признаков от других, или для прогнозирования (предсказания значений признаков), или для восстановления пробелов в данных, о чем речь пойдет ниже, то весьма перспективной может оказаться применение такого приема: по данным составляются *несколько* карт, каждая из которых имеет достаточно гладкую поверхность, но первая из них картографирует и аппроксимирует сами данные, вторая – отклонения от первой модели, то есть множество векторов, которые начинаются в точках данных и заканчиваются в точках соответствующих проекций данных на первую карту. Эти отклонения имеют две составляющие – случайный шум и неточности первой модели. Для построения третьей карты используются отклонения от проекций первых отклонений и так далее.

Рассмотрим чуть подробнее, как будет выглядеть карта первых отклонений. Если большинство точек все-таки более-менее адекватно описываются первой картой, то большая часть данных будет тяготеть к нулю новых координат (в пространстве отклонений). Если все отклонения могут быть отнесены на счет случайного шума, то закон распределения отклонений будет близок к нормальному (это, например, часто предполагается в традиционном факторном анализе).

В результате применения процедуры *мультикартирования* пользователь получает набор карт – *видов* данных, которые можно обозначить следующим образом: «вид данных», «вид первых остатков», «вид вторых остатков» и так далее. На рис.29 показан пример такого мультикартирования.

Какой смысл имеет построение нескольких карт? Во-первых, пользователь имеет возможность выделить на карте «аномальные» объекты, которые плохо описываются с помощью модели и

проанализировать из расположение в пространстве остатков с помощью информационных раскрасок. Во-вторых, использование нескольких карт позволяет существенно увеличить точность описания данных, не делая при этом карту данных слишком «неровной». Таким образом, мы, с одной стороны, получаем хорошую точность описания данных, с другой – хорошую обобщающую способность построенной модели, поскольку моделирующие многообразия могут быть сделаны достаточно гладкими.

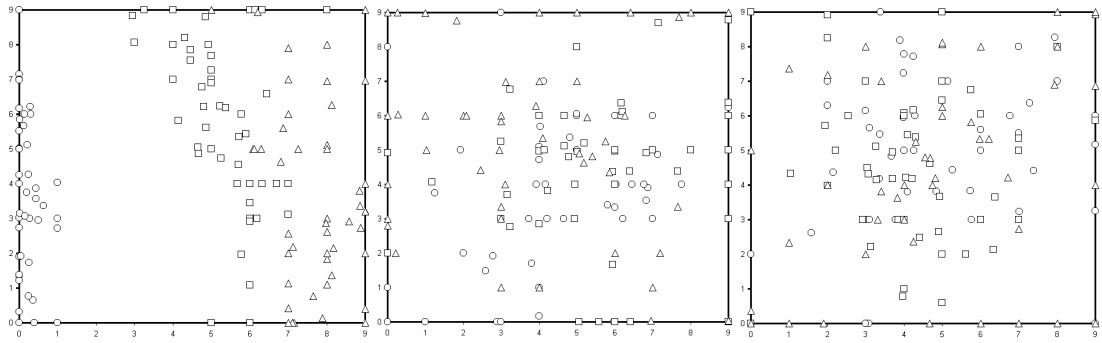


Рис. 29. Пример мультикартирования данных (база Iris)

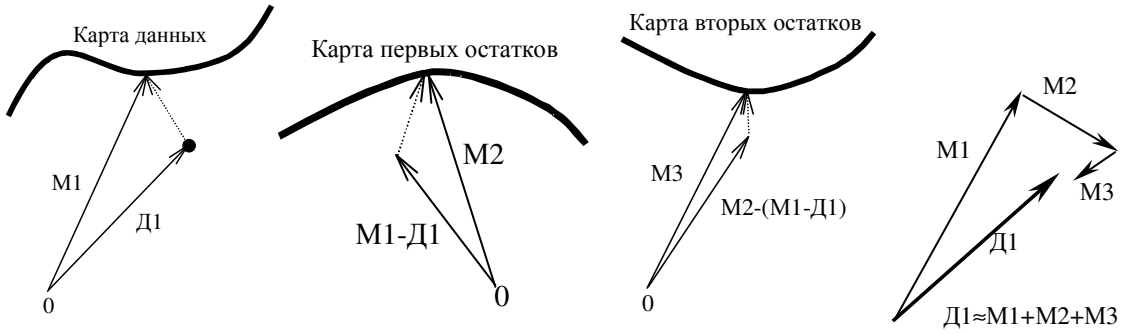


Рис. 30. Восстановление данных с помощью нескольких карт

Иными словами, мы по отдельности создаем гладкую модель самих данных, затем гладкую модель помех и так далее. При этом, в отличие от стандартного факторного анализа, не делается никаких априорных представлений о природе и структуре отклонений от модели – они описываются «как есть».

Повышение обобщающей способности модели открывает возможность правдоподобного восстановления проекций данных. На рис.30 показано каким образом вектор данных Д1, содержащий

пропущенные значения признаков может быть восстановлен последовательностью моделирующих векторов М1, М2, М3 …, которая строится по последовательности карт. Здесь существенную роль играет то обстоятельство, что мы можем проецировать вектор Д1 на карту, несмотря на то, что он содержит пропущенные значения признаков.

1.8. Особенности и ограничения подхода

1.8.1. Экстраполяция и интерполяция карты

Построенная карта, в отличие от плоскости первых главных компонент, представляет собой *ограниченное* многообразие. Его ограниченность связана с самим способом построения (нам было бы трудно оперировать с бесконечной сеткой). Поскольку карта стремится расположиться как можно плотнее к данным, в результате она представляет собой кусок криволинейной поверхности, расположенный *внутри* облака точек данных.

Из-за этой особенности карты неизбежно возникают краевые эффекты. На рис.31 видно, что после проецирования много точек оказываются расположенными на границе карты из-за того, что ближайшими точками карты для точек, лежащих на периферии облака данных, будут граничные узлы и ребра. Это существенно искажает вид таких периферийных структур после проецирования на карту – создается ложное впечатление, что данные группируются в периферийных областях.

В связи с этим замечанием к технологии построения карты желательно было бы добавить различные способы экстраполяции карты на окрестность таким образом, чтобы свести к минимуму нежелательные краевые эффекты (карта должна частично выходить за пределы группирования данных).

Самый простой способ экстраполяции – линейный, когда карта продолжается за свои пределы вдоль направлений, которые задаются ребрами, прилегающими к краям карты.

Более интересными могут оказаться нелинейные способы экстраполяции. Например, для экстраполяции можно использовать двумерную формулу Карлемана. Однако, для нее существенным является вопрос о граничных условиях – о форме поведения экстраполируемой поверхности вдали от карты. Два простых варианта – в первом в качестве асимптотического условия используется

плоскость первых главных компонент, во втором – довольно своеобразный способ соединения границ карты в одной точке – например, в какой-либо удаленной точке или в точке среднего значения координат точек данных. В последнем случае карта «замыкается», то есть все граничные точки карты мысленно склеиваются, что в некоторых ситуациях может иметь физический смысл.

На рис.31 показаны разные способы экстраполяции карты на окрестность и получаемые в результате проекции точек данных.

Необходимость в интерполяции карты (добавлении дополнительных узлов в аппроксимирующую сетку) возникает в случае, когда карта представляет из себя кусочно-линейное многообразие. Дело в том, что для сильно «угловатой» карты размеры областей пространства, для которых ближайшей точкой карты является узел, велики и многие точки проектируются в результате в этот узел, в результате чего ухудшается разрешающая способность карты – точки из упомянутых областей сливаются в одну проекцию. Разрешающую способность можно повысить, сделав карту более гладкой, введя в промежутки дополнительные узлы.

Также можно выделить два способа интерполяции – линейный, в результате которого треугольники, образующие грани карты разбиваются на несколько маленьких треугольников и нелинейные, в результате применения которых новые узлы располагаются на нелинейной поверхности, полученной в результате применения различных интерполяционных формул. И в этом случае хорошие результаты показывает применение двумерной формулы Карлемана. Интерполяция с использованием формул Карлемана, помимо прочего, является в некотором роде оптимальной.

На рис.31 показаны варианты сглаживания карты и результирующие виды спроецированных данных.

1.8.2. Качество визуализации и сложные распределения данных

Понятно, что представление данных с помощью вложенных двумерных карт тем адекватнее отражает реальные структуры, содержащиеся в данных, чем ближе эффективная размерность облака данных к двум. В этом смысле хуже всего картографируются данные, распределение которых близко к равномерному.

Не так просто дать количественные оценки качества картографирования данных. Одна из оценок очевидна – это среднее расстояние от точки данных до ближайшего к ней узла (MSE). Уточненным вариантом этой оценки является среднее расстояние от

точки данных до ее проекции (назовем ее MSPE), что более естественно в случае некусочно-постоянных способов проецирования.

Вторая оценка характеризует устойчивость проецирования по отношению к малым изменениям положения точки данных в пространстве признаков. Выше мы уже вводили величину, характеризующую эту устойчивость – это топографическая ошибка (TE).

Третью оценку можно получить, выясняя различия в двух списках – M ближайших соседей для каждой точки в пространстве признаков и M ближайших соседей в двумерном пространстве карты. Функция зависимости среднего числа различий в двух таких списках от M дает представление о сохранении отношений соседства между точками после проецирования на карту. Обозначим эту оценку через NRE (Neighborhood Relation Error).

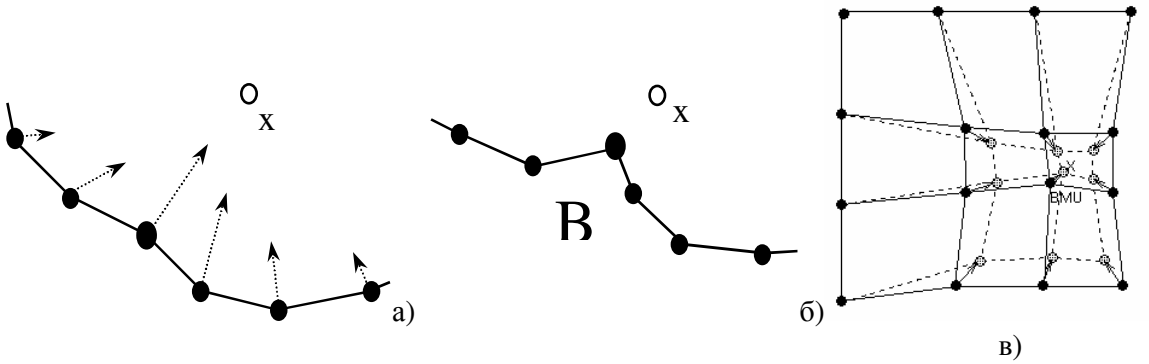


Рис. 18. Иллюстрация работы алгоритма SOM
а),б) результат действия одной итерации в случае одномерной сетки узлов;
в) результат действия одной итерации в случае двумерной сетки узлов

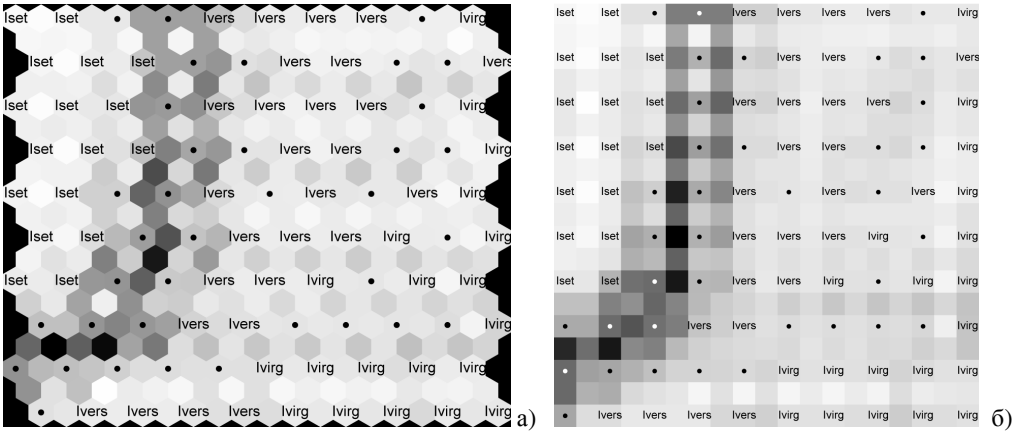
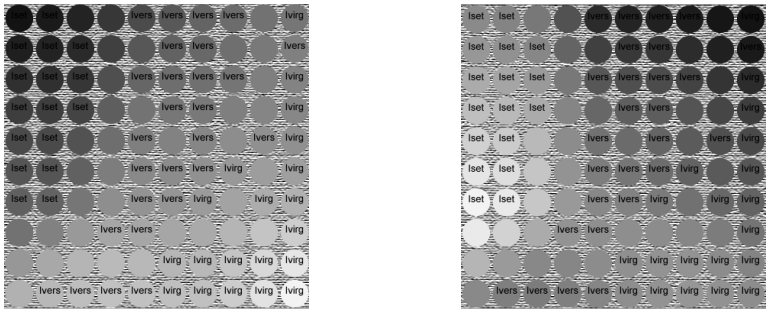


Рис. 19. Визуализация данных в виде U-матрицы.
Точками и метками обозначены узлы, метки показывают точки какого класса оказались в узле. Оттенок ячейки, расположенной между двумя узлами, отражает расстояние между узлами в исходном пространстве. Более темный оттенок соответствует большему расстоянию. Оттенок самого узла вычисляется с помощью усреднения. Из раскраски можно сделать вывод о том, что класс Iris-setosa хорошо пространственно отделен от двух других классов.
а) случай гексагональной сетки; б) случай прямоугольной сетки.



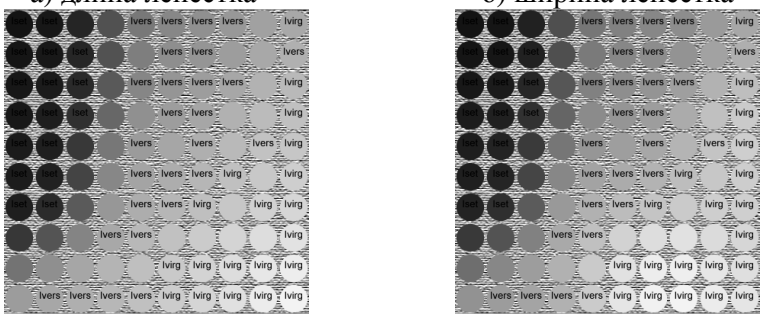
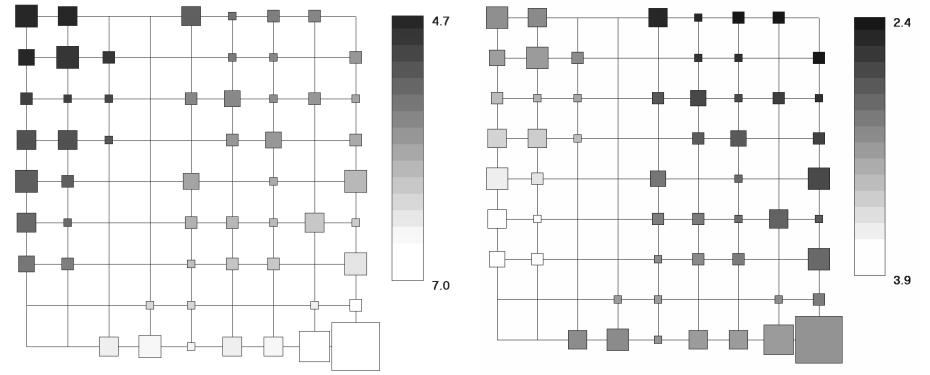


Рис. 20. Представление значений признаков с помощью раскрасок. Более темный оттенок соответствует меньшим значениям признака.



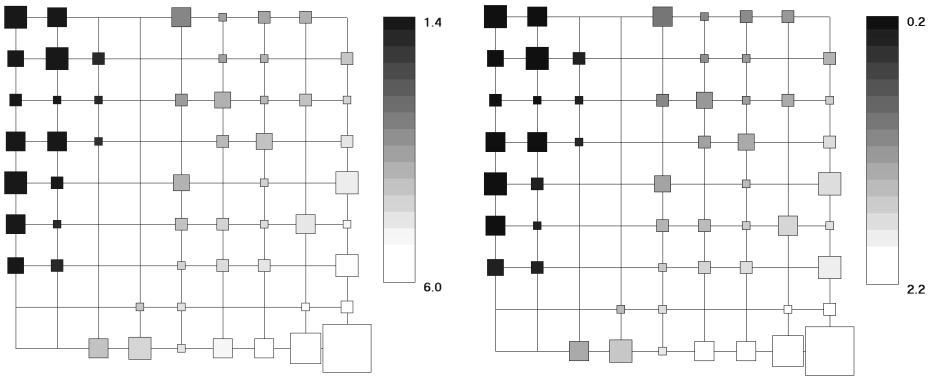


Рис. 21. Визуализация данных с помощью диаграмм Хинтона.

Чем бо

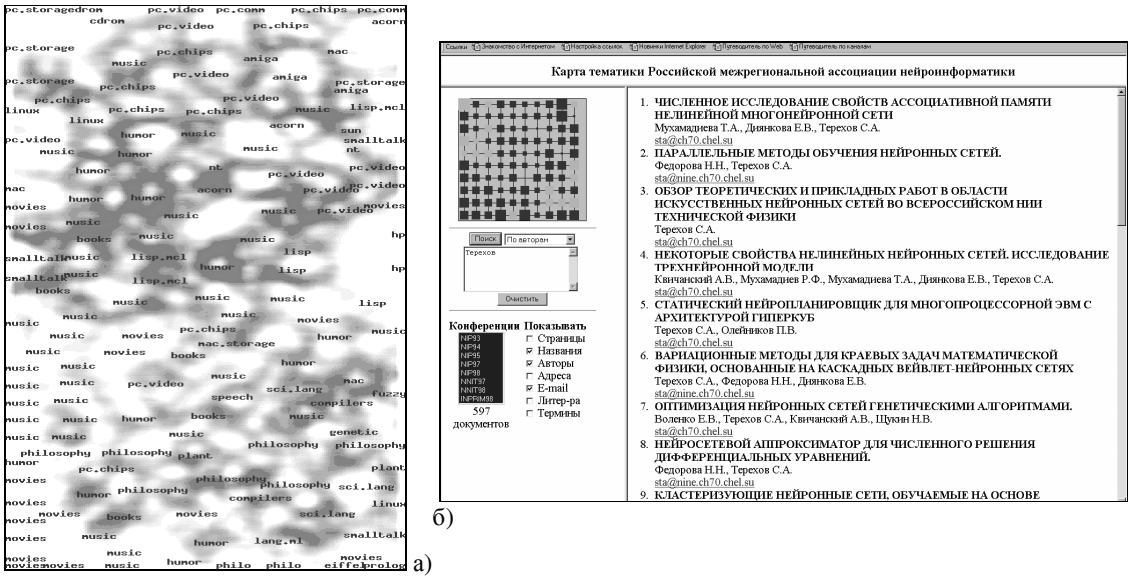
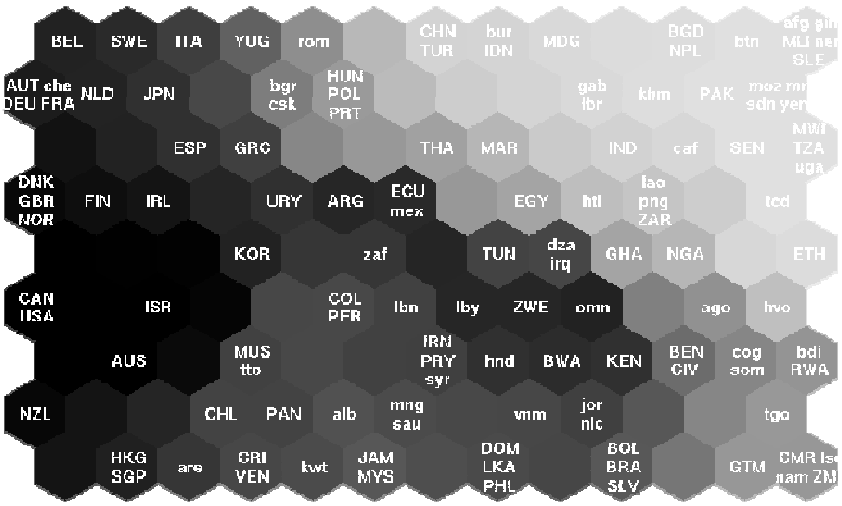


Рис. 22. Примеры применения SOM для картографирования больших текстовых коллекций.

а) Проект WebSOM – картографирование содержания более полутора миллионов статей UseNet. На рисунке показана карта «верхнего уровня» – своеобразный двумерный каталог статей, в котором статьи, близкие по тематике расположены по возможности в близких областях. Метки обозначают наиболее часто употребимое слово в данной области тематической карты. Цвет отражает насыщенность той или иной темы статьями. Щелкнув по соответствующей области, можно получить карту более низкого уровня, где информация представлена более подробно для выбранной темы. Таким образом, можно добраться до карты самого низкого уровня, где можно уже будет просматривать содержание отдельных статей. Такое построение карт называется «иерархическим».

б) Проект ConfWeb – визуализация в виде диаграммы Хинтона содержания нескольких сотен статей и тезисов докладов, представленных на российских конференциях по нейроинформатике за несколько лет. Изначально статьи представлены в виде одноцветной диаграммы Хинтона. В режиме поиска «По автору» подсвечиваются те квадраты, которые отображают те кластеры данных, в которых фигурирует имя автора, давая своеобразную «картину интересов» автора. В режиме поиска «По ассоциациям» подсвечивается тот квадрат, в кластере которого содержится введенное слово. Щелкнув по любому квадрату, в правом фрейме можно получить заголовки и авторов статей, вошедших в кластер.



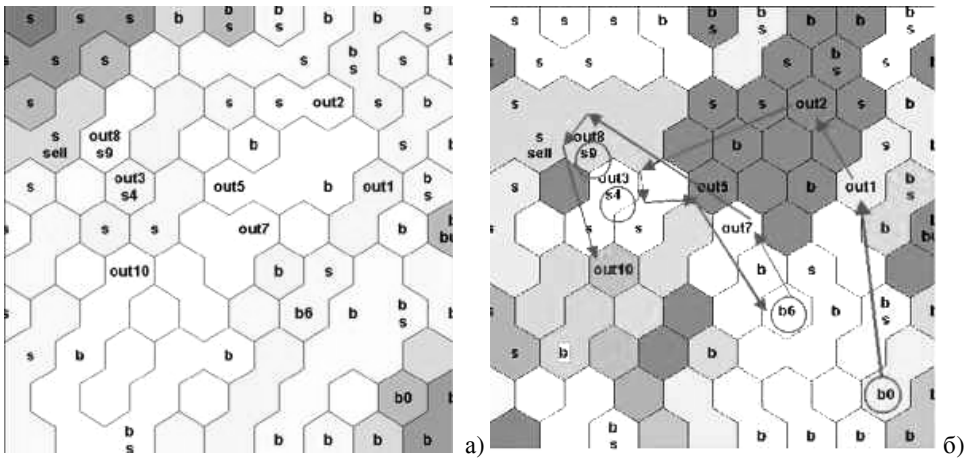
а)

б)

Рис. 23. Раскраска географической карты с помощью SOM.

а) Карта Кохонена, построенная по нескольким десяткам экономических и социальных показателей разных стран (по странам СНГ у создателей карты данных не было). После построения гексагональная сетка не была окрашена, на нее были лишь нанесены метки стран, причем страны, оказавшиеся рядом на карте обладают сходными показателями. Затем на карту Кохонена был наложен двумерный цветной спектр (на рисунке нет возможности показать цветную раскраску), в результате чего каждый узел получил свой цвет, причем соседние узлы получили близкие по спектру цвета.

б) Цвета узлов были использованы для раскраски мировой карты по «похожести» показателей.

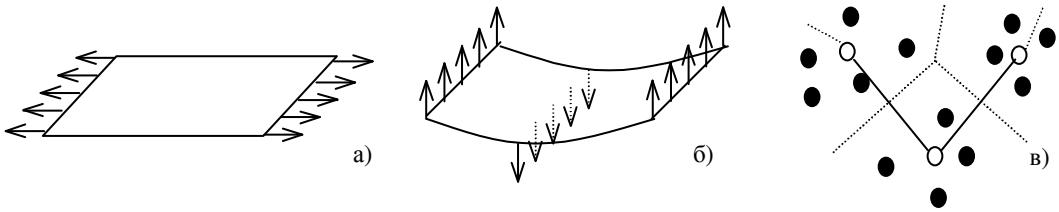


а) б)

Рис. 24. Применение технологии SOM для анализа состояния фондового рынка.

а) Карта Кохонена, построенная по ежесуточным значениям основных показателей фондового рынка за некоторый промежуток времени. Светлым цветом изображены узлы, расположенные ближе к центру многомерного облака данных, темным – «периферийные» узлы. Значки «b», «s», «out» обозначают состояния рынка, в которых предпочтительнее покупка (buy), продажа (sell) и бездействие соответственно. В узлах, где одновременно находятся значки «b» и «s» возможна и покупка, и продажа. Таким образом, на карту нанесен некоторый «опыт» покупок и продаж.

б) Раскраска несколько иная. Светлым обозначены узлы, расположенные ближе к сгущениям данных. Стрелками изображены состояния рынка за 11 последовательных дней. Кружками отмечены дни, когда инвестором совершались сделки. Таким образом, инвестор может мгновенно сопоставлять свою деятельность с накопленным опытом рынка, принимая более оперативные решения.



а) б) в)

Рис. 25.

а) силы, растягивающие упругую пластину; б) силы, изгибающие упругую пластину; в) одномерный случай абсолютно мягкой карты – каждый узел карты располагается в центре «таксона», пунктиром изображены границы зон, в переделах которых точки являются ближайшими к данному узлу – так называемых *ячеек Вороного*.

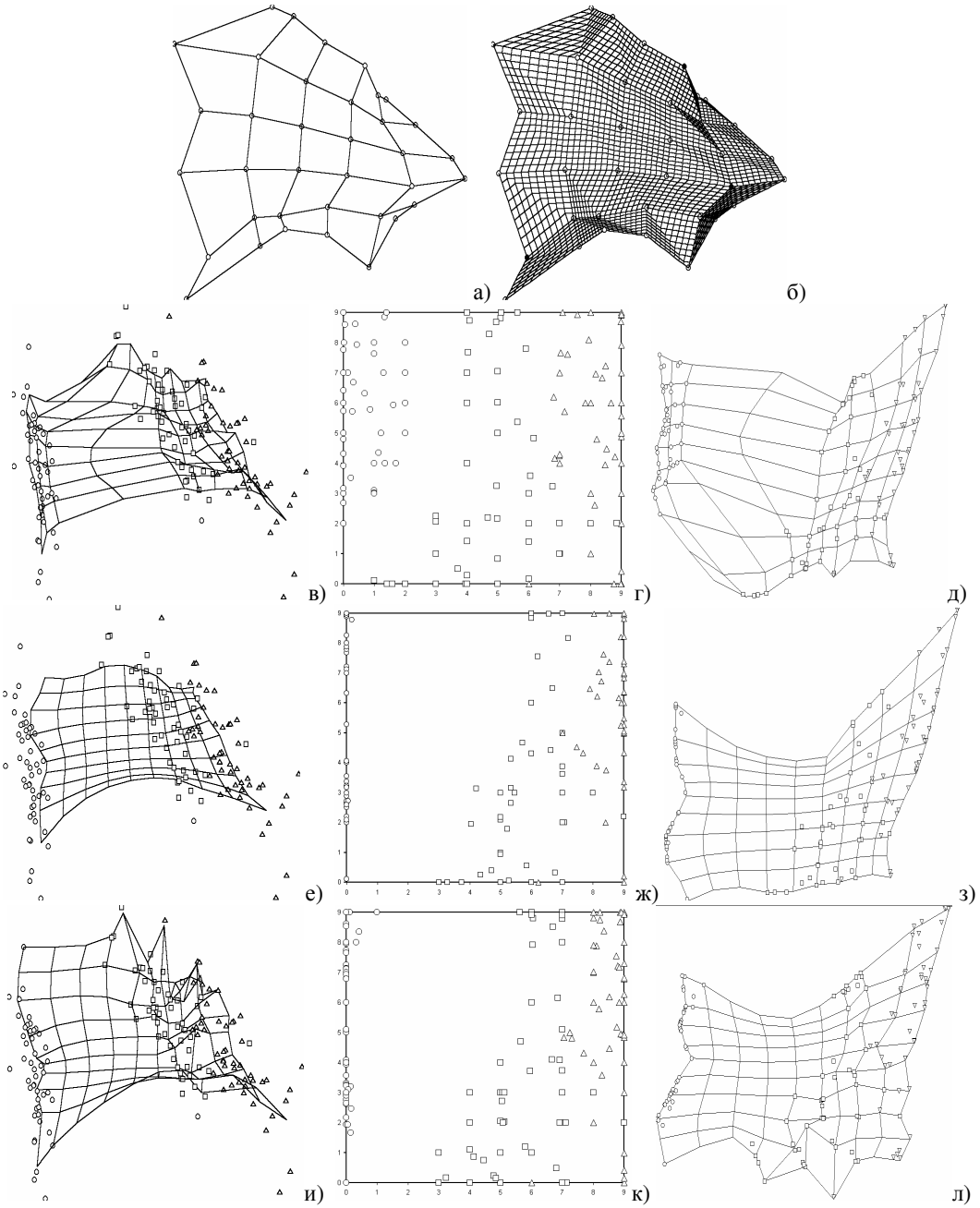


Рис. 26. Иллюстрации к работе метода упругих карт.

а),б) кусочно-линейный способ доопределения карты до многообразия («граненая» карта);

в),г) карта, построенная в результате работы алгоритма SOM и результирующие проекции данных; видно, что сетка сильно растянута в середине, где данных нет, в результате проекции располагаются на карте более равномерно, чем в исходном пространстве (кластерная структура «смазывается»);

е),ж) упругая карта с относительно большими значениями коэффициентов упругости, видно, что сетка более равномерная, в результате проекции расположены более адекватно; на проекциях виден недостаток упругой карты – много проекций оказались на краю карты (об этом подробнее в разделе 1.8.1);

и),к) упругая карта с относительно малыми значениями коэффициентов упругости;

д),з),л) соответствующие криволинейные развертки карты (максимально сохраняются расстояния между узлами в исходном пространстве).

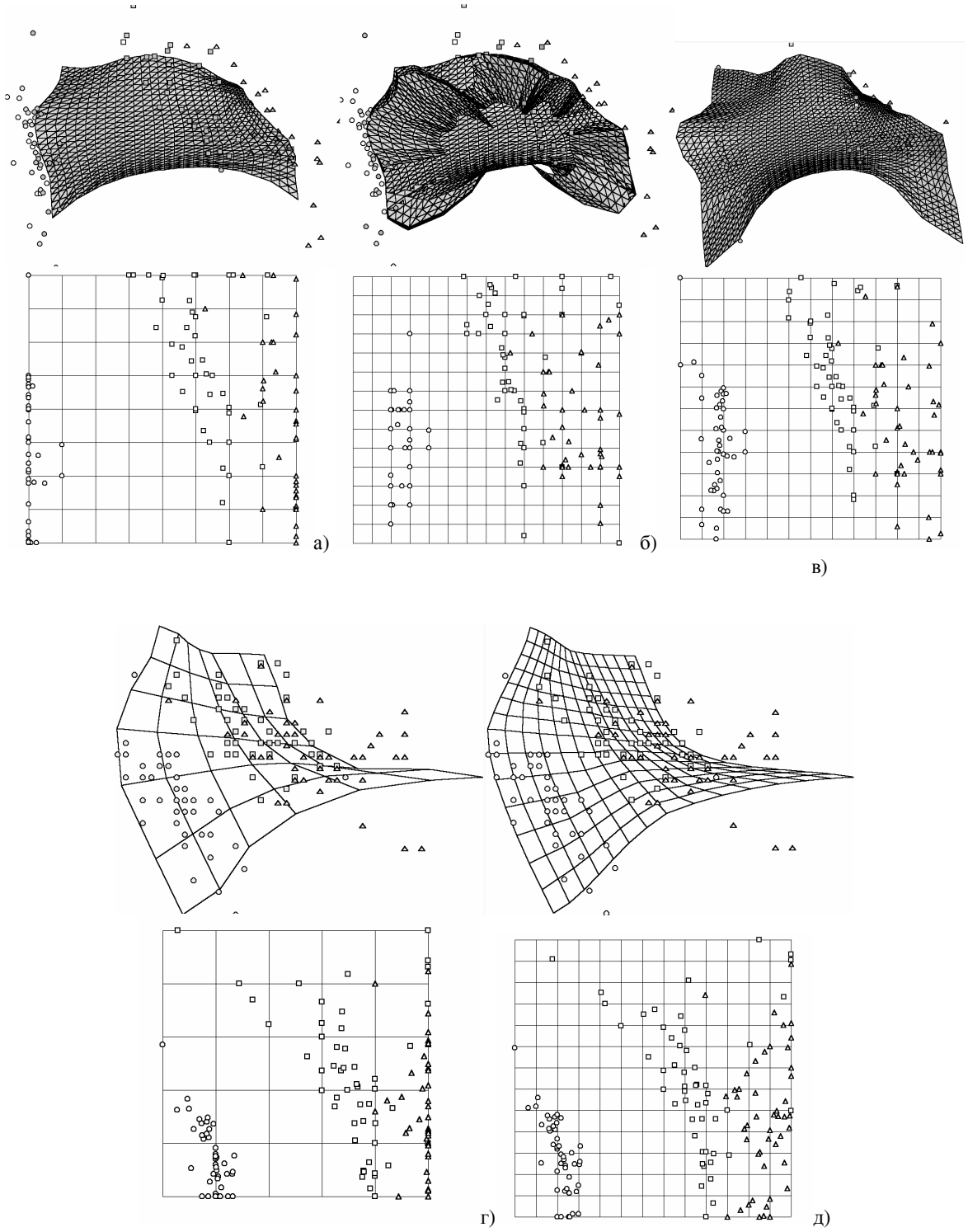


Рис. 31. Экстраполяция карты на окрестность и интерполяция карты

а) исходно построенная карта располагается «внутри» облака данных, граничные точки проецируются на крайние ребра;
б) экстраполяция по двумерной формуле Карлемана с граничным условием – узлы сходятся к центру облака данных;
в) линейная экстраполяция;
г), д) пример нелинейной интерполяции г) – исходная карта, д) – после применения процедуры интеполяции, карта в результате имеет лучшее разрешение.

Глава 2. Плетение и закидывание сетей и неводов

2.1. Предобработка данных

Среди m признаков (которые иначе могут называться *переменными*) могут быть признаки, измеряемые в количественной, номинальной или порядковой шкалах. В самом простом случае все признаки измеряются в одной и той же шкале, но в реальных ситуациях, как правило, используются несколько типов шкал измерения признаков.

Перед применением к данным различных алгоритмов анализа их структуры всегда возникает необходимость применения различных способов предобработки, которая в задаче визуализации данных заключается в *оцифровке*, *нормировке* и *выборе метрики*.

2.1.1. Обозначения

Для того, чтобы последующее изложение было более ясным, введем систему обозначений, которой будем придерживаться на протяжении всей последующей главы.

X_i, Y_i – совокупность координат i-ой точки данных (радиус-вектор);

$X_i \cdot Y_i$ или (X_i, Y_i)– скалярное произведение вектора X_i и Y_j;

$(X_i)^2$ – «квадрат вектора» – сумма квадратов его координат (число);

x_{ij} – значение j-ой координаты i-ой точки объекта (число);

ξ_i, η_i – обозначения i-ой координаты пространства данных (как меняющейся величины);

m – размерность пространства данных;

$|X|$, N – число объектов;

δ_{ij} – «дельта-символ» Кронеккера: $\begin{cases} \delta_{ij} = 1, & i = j \\ \delta_{ij} = 0, & i \neq j \end{cases}$.

2.1.2. Оцифровка дискретных шкал

Под оцифровкой, как правило, подразумевается приведение всех типов признаков к одной количественной шкале.

В самом простом случае дихотомической шкалы, то есть когда признак может принимать значения «да» или «нет», нет большой разницы какие числа будут приписаны положительному или отрицательному ответу. Самые распространенные варианты: ответу «да» приписывают число 1, ответу «нет» – либо –1, либо 0.

В случае порядковых шкал, как правило, порядок следования градаций признака отражает степень усиления или ослабления того или иного качества. Числовые метки

признака в этом случае присваиваются таким образом, чтобы расстояния между двумя отметками интуитивно соответствовали разнице между соответствующими градациями (например, если признак имеет шкалу «плохо-никак-хорошо», то логично приписать градациям метки –1;0;1, а вот в случае шкалы «малый-средний-крупный-сверхкрупный» более уместным может оказаться использовать логарифмические метки, т.е. 0.1;1;10;100). Не играет большого значения выбор «начала отсчета» шкалы признака, но психологически удобнее измерять в числовой шкале с отрицательными и положительными числовыми метками качество, меняющееся от противоположности к противоположности (как «плохо-хорошо»), а в шкалах с «абсолютным нулем» измерять постепенное нарастание какого-либо качества (как «отсутствие-частичное присутствие-полное присутствие»).

Большей свободой и математическим осмыслением обладает процедура оцифровки номинальных шкал. В это случае, как правило, не играет роли порядок следования и расстояния между градациями признаков.

✂ Хотя возможны и исключения – для примера возьмем признак «Отрасль промышленности». Значения признака «цветная металлургия» и «черная металлургия» психологически воспринимаются ближе, чем «нефтяная промышленность» и «пищевая промышленность», хотя признак, обозначающий принадлежность предприятия к определенной отрасли нельзя назвать порядковым. Для упорядочивания значений шкалы признака можно пользоваться методами многомерного шкалирования и в конечном счете возможно, что признаку будут отвечать не одна, а несколько числовых меток, если построенное «пространство восприятия» окажется эффективно двумерным. ✂

Свобода в выборе числовых меток для номинальных шкал дает возможность искусственно упростить структуру набора данных, например, добиться того, чтобы шкалы признаков были максимально скоррелированы друг с другом. Популярным методом является максимизация функционала Q, где

$$Q^2 = \sum_{i<j}^{m} r_{ij}^2 ,$$

где r_{ij} – коэффициент линейной корреляции между i-ым и j-ым признаком, m – число признаков, среди которых есть как номинальные, так и количественные. Допустим, что номинальным признаков уже были каким-то образом присвоены числовые метки. Тогда

$$Q^2 = Q_1^2 + Q_{1,2}^2 + Q_2^2 ,$$

где в Q_1 входят коэффициенты корреляции между номинальными признаками, подлежащими оцифровке, в $Q_{1,2}$ – коэффициенты корреляции между номинальными и количественными признаками, а Q_2 – часть функционала, состоящая из коэффициентов корреляции между количественными признаками. Последняя не зависит от оцифровки и оптимизации на самом деле подлежит функционал $\tilde{Q}$, где

$$\tilde{Q}^2 = Q_1^2 + Q_{1,2}^2$$

Пусть $X^{(1)}$ – набор номинальных признаков, подлежащих оцифровке, $X^{(2)}$ – набор количественных признаков, c_k^i – числовая метка, присвоенная k-ой категории i-го номинального признака. $N(i,j)$ – матрица сопряженности между i-ым и j-ым номинальными признаками, в качестве оценки которой можно взять числа $n_{kl}(i,j)$ одновременного появления для i-го признака категории k, а для j-го признака – категории l, $p_{i,k}$ – частота появления k-ой градации признака i, который имеет l_i градаций, $P_i=\mathrm{diag}(p_{i,1};p_{i,2}..,p_{i,li})$; наконец, $\bar{c}_k^{ij}$ – среднее значение количественного признака j ($j\in X^{(2)}$) на тех объектах, у которых i-ый номинальный признак имеет категорию k.

Тогда градиент функционала

$$\frac{\partial \tilde{Q}}{\partial c_k^i} = \sum_{j=1}^{l_i} a_{kj}^i c^i_j \text{, где}$$

$$a_{kj}^i = \sum_{\substack{m\in X^{(1)} \\ m\neq i}} \sum_{i_1=1}^{l_m}\sum_{i_2=1}^{l_m} n_{ki_1}(i,m) c_{i_1}^m c_{i_2}^m n_{i_2 j}(m,i) + \sum_{m\in X^{(2)}} \bar{c}_k^{im}\bar{c}_j^{im} p_j p_k, \qquad i\in X^{(1)},$$

$k=1..l_i$.

При этом предполагается, что все данные (включая номинальные признаки) предварительно нормированы на единичную дисперсию и центрированы. Для номинальных признаков это означает, что

$\sum_{i=1}^{N} c_k^j(i) = 0$, $\frac{1}{N}\sum_{i=1}^{N}(c_k^j(i))^2 = 1$ для всех j, k; N – число объектов; $c_k^j(i)$ – значение числовой метки для i-го объекта. Выполнения этих условий всегда можно добиться линейным преобразованием.

Зная градиент функционала $\tilde{Q}$ можно воспользоваться любым из методов градиентной оптимизации. Для количественных шкал значения признаков в результате не меняются, а значения меток номинальных признаков назначаются категориям таким образом, чтобы они были максимально скореллированы друг с другом и с количественными признаками, что приводит к снижению эффективной размерности

пространства данных (часть признаков становится статистически линейно зависима от других).

2.1.3. Нормировка данных

После того, как все признаки оказываются описанными в количественной шкале, их обычно *центрируют* и *нормируют*.

Первым шагом в статистической обработке данных, как правило, является нахождение точки среднего значения всех признаков – геометрического центра многомерного облака точек данных. Обычно удобно сдвинуть все точки данных на один и тот же вектор таким образом, чтобы центр облака оказался в начале координат.

Далее следует нормировка – то есть деление всех значений признаков на определенное число таким образом, чтобы значения признаков попадали в сопоставимые по величине интервалы. В качестве такого числа обычно выбирается один из *характерных масштабов*.

В многомерном облаке данных существует несколько масштабов. Во-первых, это квадратный корень из общей дисперсии облака данных, называемый среднеквадратичным отклонением:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2}\ ,\ \bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i\ .$$

Напомним, что здесь и далее большими буквами X_i обозначаются вектора данных, а маленькими – x_{ij}: j-ая координата i-го вектора.

В случае, если выборка может считаться полученной из нормального распределения, то в шаре с центром в $\bar{x}$ радиусом σ находится около двух третей от числа точек данных.

Существует масштаб, характеризующий максимальный разброс в облаке данных

$$R = \max_{i=1..N} \left\| X_i - \bar{X} \right\|.$$

Нормировка всех признаков на R приводит к тому, что все облако данных оказывается заключено в шар единичного радиуса.

Если в качестве масштаба выбраны σ или R, то соответствующие формулы предобработки (нормировки на «единичную дисперсию» и «на единичный шар») имеют вид:

$$\tilde{X}_i = \frac{X_i - \bar{X}}{\sigma},\ \tilde{X}_i = \frac{X_i - \bar{X}}{R},$$

где $\tilde{X}_i, X_i$ – новые и старые значения векторов признаков.

С помощью масштаба σ, как правило, определяются понятия кластера и сгущения в облаке данных. Приведем эти определения (см. [4]).

Кластер – группа точек G такая, что средний квадрат внутригруппового расстояния до центра группы меньше среднего расстояния до общего центра в исходном наборе объектов, т.е. $\bar{d}_G^2 < \sigma^2$, где $\bar{d}_G^2 = \frac{1}{N}\sum_{X_i \in G}(X_i - \bar{X}_G)^2, \bar{X}_G = \frac{1}{N}\sum_{x_i \in G} X_i$.

Сгущение – группа точек G такая, что максимальный квадрат расстояния точек из G до центра группы меньше σ^2, т.е. $\bar{d}_{G,\max}^2 < \sigma^2$, где $, \bar{X}_G = \frac{1}{N}\sum_{x_i \in G} X_i$.

✂ К сожалению, такие определения не всегда соответствуют интуитивным представлениям о кластерах и сгущениях – например, когда сгущение представляет из себя сильно вытянутое облако точек. ✂

Кроме того, если диапазоны значений для разных признаков очень сильно отличаются друг от друга, то разумно для каждого из признаков применять собственный масштаб. Для каждого из признаков можно ввести свое среднеквадратичное отклонение и разброс:

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2}\ ,\ \bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}\ ,\ R_j = \max_{i=1..N}\left\|x_{ij} - \bar{x}_j\right\|,$$

где x_{ij} – значение j-го признака на i-ом объекте. Как результат, получаем формулы для нормировки на «единичную дисперсию для каждого из признаков» и «на единичный куб»:

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{\sigma_j},\ \tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{R_j}.$$

Эти нормировки не являются «изотропными», то есть они сжимают облако данных в некоторых направлениях сильнее, в некоторых – меньше, что в некоторых случаях является желательным, а в некоторых – нарушает структуру данных (взаимных расстояний). Такая нормировка фактически эквивалентна выбору взвешенной евклидовой метрики, о которой речь пойдет ниже.

Наконец, следует упомянуть, что с каждым из количественных признаков могут быть связаны еще два масштаба – это точность и допуск (см раздел 1.1), с помощью которых также можно «обезразмерить» значения этих признаков.

2.1.4. Выбор метрики для пространства данных

Любая нормировка данных приводит к тому, что изменяются взаимные расстояния между точками данных. Это можно истолковать как выбор метрики иной по сравнению с обычной евклидовой. Выбор метрики является важным моментом в любой методике анализа структуры данных.

Сначала введем понятия *матрицы связи* признаков и *матрицы расстояний* между объектами.

Матрица связи – квадратная симметрическая матрица размерами $m \times m$ типа «признак-признак», где на пересечении i-ой строки j-ого столбца стоит мера «взаимосвязанности» i-го и j-го признака. Самой популярной мерой связи количественных признаков является коэффициент корреляции Пирсона, который вычисляется по формуле

$$r_{kj} = \frac{s_{kj}}{\sqrt{s_{kk} s_{jj}}}, \text{ где } \; s_{kj} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{ik} - \bar{x}_k)(x_{ij} - \bar{x}_j).$$

В результате матрицей связи становится *корреляционная матрица*

$$R = \begin{bmatrix} r_{11} & \dots & r_{1m} \\ \dots & \dots & \dots \\ r_{m1} & \dots & r_{mm} \end{bmatrix}.$$

✂ В случае применения порядковых и номинальных шкал могут применяться другие меры связи такие как коэффициент ранговой корреляции для вычисления связи между порядковыми признаками или бисериальный коэффициент корреляции, применяемый для измерения связи между порядковым и количественным признаком. Подробный анализ этих коэффициентов приведен в [28] ✂

Матрица расстояний – квадратная матрица размерами $N \times N$ типа «объект-объект», где на пересечении i-ой строки j-ого столбца стоит мера удаленности между i-ым и j-ым объектом:

$$D = \begin{bmatrix} d_{11} & \dots & d_{1N} \\ \dots & d_{ij} & \dots \\ d_{N1} & \dots & d_{NN} \end{bmatrix}.$$

Для того, чтобы величины d_{ij} имели смысл расстояний между объектами в многомерном пространстве, необходимо, чтобы для всех $i, j = 1…N$ выполнялись требования:

1. Максимальное сходство объекта с самим собой: $d_{ii} = \min\limits_{j=1..N} d_{ij}$.
2. Требование симметрии: $d_{ij} = d_{ji}$.
3. Выполнение неравенства треугольника: $d_{ij} \le d_{ik} + d_{kj}$.

Если введенная мера удаленности между объектами такова, что выполняется это условия, то будем называть D матрицей расстояний.

✂ Данные могут быть исходно заданы в виде матрицы связи или удаленностей. Тогда возникает задача по заданной матрице восстановить в каком-либо смысле исходное множество точек данных таким образом, чтобы для него матрица связанности или удаленностей имели заданный вид (точно или приближенно). ✂

Правило вычисления расстояния между объектами может сильно видоизменяться в зависимости от специфики задачи. Если такое правило задано, то говорят, что в пространстве признаков введена метрика. Рассмотрим различные правила измерения расстояний:

I. *Квадратичные метрики*:

Для класса квадратичных метрик квадрат расстояния между объектами является квадратичной формой от разностей значений их координат:

$$d_{ij} = \sqrt{(X_i - X_j)^T G (X_i - X_j)},$$

где G – симметричная положительно определенная матрица.

В качестве матрицы G размерами $m \times m$ можно выбрать

а) единичную матрицу $G = E$, в результате чего получаем обычное евклидово расстояние

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2};$$

б) диагональную матрицу $G = \mathrm{diag}(g_1, g_2 \ldots g_m)$, в результате получим взвешенную евлидову метрику

$$d_{ij} = \sqrt{\sum_{k=1}^{m} g_k (x_{ik} - x_{jk})^2};$$

в) матрицу, обратную *ковариационной матрице* $G = S^{-1}$:

$$S = \begin{bmatrix} s_{11} & \dots & s_{1N} \\ \dots & s_{ij} & \dots \\ s_{N1} & \dots & s_{NN} \end{bmatrix}, \text{ где } s_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j),$$

что дает *махаланобисову метрику.*

С ковариационной матрицей связано понятие *эллипсоида рассеяния* облака точек. Осями эллипсоида рассеяния являются направления собственных векторов S (поскольку S-симметричная матрица, то собственные вектора образуют полную ортогональную систему векторов), длины осей выбираются равными значениям соответствующих собственных чисел.

Особенностью махаланобисовой метрики является то, что в ней эллипсоид рассеяния точек данных является шаром с единичным радиусом.

Основным преимуществом использования квадратичных метрик является тот факт, что производная от квадрата расстояния, измеренного в такой метрике является линейной функцией от координат объектов, что может быть использовано при решении различных задач (например, задач оптимизации).

II. *Специальные виды метрики*:

а) *городская метрика* или расстояние Минковского:

$$d_{ij} = \sum_{k=1}^{m} I_k(X_i, X_j),$$

применяется для измерения расстояний между объектами, описываемыми в порядковой шкале; $I_k(X_i,X_j)$ – разница в номерах градаций по k-ому признаку у сравниваемых объектов с векторами X_i и X_j;

б) Расстояние Хэмминга

$$d_{ij} = \sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|$$

чаще всего применяется для измерения расстояния между объектами, описываемыми в дихотомической шкале. Тогда расстояние Хэмминга – число несовпадающих значений признаков в рассматриваемом i-ом и j-ом объектах.

в) расстояние Колмогорова

$$d_{ij} = \left(\sum_{k=1}^{m} \left| x_{ik} - x_{jk} \right|^{p} \right)^{1/p}$$

является обобщением евклидовой метрики. Так, при $p = 1$ получаем метрику Хэмминга, при $p = 2$ – евклидову метрику, при $p = \infty$ – метрику «по максимуму модуля»:

$$d_{ij} = \max_{\{k=1..m\}} \left| x_{ik} - x_{jk} \right|$$

Все эти метрики допускают тривиальное обобщение, если производить суммирование с весами и тогда получаем взвешенную городскую, взвешенную Хэммингову и взвешенную метрику Колмогорова. Веса признаков подбираются или с помощью простых эвристических методов, или настраиваются с помощью специальных процедур (см., например, [4]).

III. *Риманова метрика*

Риманова метрика является обобщением квадратичной метрики в случае точечного пространства. В ней задается квадратичное расстояние между бесконечно близкими точками

$$ds = \sqrt{\sum_{i,j} g_{ij} d\xi_i d\xi_j} \ ,$$

где $d\xi_i, d\xi_j$ – бесконечно малые приращения координат, g_{ij} – метрический тензор. Значения g_{ij} зависят от точки пространства, в которой измеряется расстояние между бесконечно близкими точками. Тогда расстояние между объектами измеряется с помощью интеграла по траектории и, вообще говоря, зависит от выбора траектории, соединяющей два объекта

$$d_{ij}(L) = \oint_{L} ds \ .$$

Если мы предполагаем, что расстояния измеряются по кратчайшему пути, то среди всех траекторий L выбирается та, при использовании которой расстояние оказывается наименьшим

$$d_{ij} = \inf_{L} d_{ij}(L) \ .$$

Такая траектория называется *геодезической* и играет роль, аналогичную прямой в евклидовом пространстве.

Приведем несколько простых видов римановых метрик

а) $g_{ij} = diag(f_1(\xi_1), f_2(\xi_2), ..., f_m(\xi_m))$, где $f_1(\xi)$, $f_2(\xi)$, $f_3(\xi)$ – монотонные функции одного аргумента. По существу такая метрика может

быть преобразована в евклидову нелинейным преобразованием координат $\xi_i^{'} = f_i^{-1}(\xi_i)$;

б) конформно-плоская метрика

$g_{ij} = \gamma(\xi_1, \xi_2 ... \xi_m)\delta_{ij}$,

где δ_{ij} – символ Кронеккера.

В общем случае эта метрика не может быть превращена в евклидову сразу во всем пространстве никаким нелинейным преобразованием координат. Ее смысл состоит в том, что масштаб, с помощью которого измеряются расстояния, меняется от точки к точке пространства.

2.1.5. Настройка метрики

При использовании взвешенных метрик остаются неопределенными веса признаков. Иногда условия задачи позволяют выделить те признаки, которые являются «более значимыми» при измерении расстояний и назначить для этих признаков значения весов. Если никаких дополнительных соображений нет, то для настройки весов могут быть использованы некоторые специальные приемы. Приведем примеры:

1) Адаптивные квадратичные метрики:

Рассмотрим метрику

$d_{ij} = \sqrt{(X_i - X_j)^T G (X_i - X_j)}$,

d_{ij} – расстояние между i-ым и j-ым объектом.

Положим, что det G = 1. Такой выбор не делает результаты менее общими, но его использование позволяет не рассматривать решения, отличающиеся друг от друга только преобразованием гомотетии (равномерным растяжением по всем осям). Допустим, что на наборе объектов уже существует определенная система отношений – объекты разбиты на k непересекающихся классов. Введем матрицу внутриклассового разброса W:

$$W = \frac{1}{N} \sum_{i=1}^{k} \sum_{X_l, X_m \in K_i} (X_l - X_m)(X_l - X_m)^T ,$$

где T – обозначение операции транспонирования, K_i – обозначение множества объектов, принадлежащих i-ому классу.

Если выбрать $G = \alpha W^{-1}$, α – числовой множитель, то минимальной (среди всех квадратичных метрик) оказывается величина внутриклассового разброса

$$w = \sum_{i=1}^{k} \sum_{X_l, X_m \in K_i} d^2(X_l, X_m),$$

где $d^2(X_l, X_m)$ – квадрат расстояния между l-ым и m-ым объектом,

что приводит к тому, что классы оказываются максимально компактными [4].

Если разбиение на классы не заданно изначально, то возможна такая настройка метрики, при которой данные будут разбиты на k кластеров «наиболее контрастно». Итерационный алгоритм состоит из двух фаз:

Фаза 1. При фиксированной метрике $G^{(t)}$ производится разбиение множества данных на k кластеров тем или иным способом (см, например, [1,30]). Число кластеров задается в начале работы и далее не меняется.

Фаза 2. По полученной классификации строится матрица внутриклассового разброса W и вводится метрика $G^{(t+1)} = (W^{(t+1)})^{-1}$.

Шаги алгоритма повторяются до тех пор, пока относительные изменения значений элементов G не станут меньше заданного числа ε.

При использовании взвешенной евклидовой метрики вычисление весов может быть упрощено. В качестве их значений можно выбрать $g_k = \alpha w_{kk}^{-1}$, где w_{kk} – k-ый диагональный элемент матрицы W, α – нормирующий множитель (например, при $\alpha = \prod_k w_{kk}$ получаем равенство $\det G^{(t)} = 1$).

2) Использование частично-обучающих выборок

Частично обучающая выборка (ЧОВ) – множество пар объектов, относительно которых известно, что они принадлежат одному классу [4]. Естественно считать такие объекты близкими.

Введем аналог матрицы внутриклассового разброса для ЧОВ:

$$W = \frac{1}{n_{чов}} \sum_{i=1}^{n_{чов}} (X_{1i} - X_{2i})(X_{1i} - X_{2i})^T \text{ , } (X_{1i}, X_{2i} - i\text{-ая пара из ЧОВ}).$$

Если W невырождена, то оптимальную метрику получим, выбирая $G = \alpha W^{-1}$, где α – нормирующий множитель.

Задачу можно упростить, используя взвешенную евклидову метрику и выбирая

$$G = \alpha \cdot diag(w_{11}^{-1}, w_{22}^{-1}, ..., w_{mm}^{-1}),\ \alpha = \prod_i w_{ii}\ .$$

3) Максимизация корреляций.

Рассмотрим риманову метрику. Можно попытаться подобрать вид функций $f_i(x)$ таким образом, чтобы признаки оказались максимально скоррелированы. В качестве критерия, по которому ищется преобразование, можно использовать величину

$Q^2 = \sum_{i<j}^{m} r_{ij}^2$, r_{ij} – коэффициент корреляции между i-ым и j-ым признаком.

Функции могут быть выбраны из некоторого семейства монотонных преобразований, например, преобразования Бокса-Кокса [4]:

$$\begin{cases} f_i(x) = (x^{\alpha_i} - 1)/\alpha_i, & \alpha_i \neq 0 \\ f_i(x) = \ln x, & \alpha_i = 0 \end{cases}$$

или более общее двухпараметрическое семейство

$$\begin{cases} f_i(x) = (x^{\alpha_i} - \beta_i)/\alpha_i, & \alpha_i \neq 0 \\ f_i(x) = \ln(x - \beta_i), & \alpha_i = 0 \end{cases}.$$

Тогда $Q = Q(\alpha_1, \beta_1, \alpha_2, \beta_2, ..., \alpha_m, \beta_m)$ и критерий Q может быть максимизирован по параметрам α_k, β_k, k=1…m.

4) Контрастирование структуры данных

Рассмотрим такой вариант римановой метрики, в котором плотность данных оказалась бы почти равномерной или, наоборот, структура сгущений оказалась бы более контрастной.

Элемент объема риманова пространства вычисляется по формуле $dV = \sqrt{|g|}dx^1 dx^2 \ldots dx^n$, где $|g|$ – определитель матрицы метрического тензора.

Оценим нормированную на одну точку плотность распределения данных в исходном пространстве с помощью какой-либо непараметрической оценки. Например, пусть

$$\rho(x) = \frac{1}{|X|\sigma^n} \sum_{i=1}^{N} \prod_{j=1}^{n} K\left(\frac{x_j - x_j^i}{\sigma} \right),$$

где $K(x)$ – некоторая функция, удовлетворяющая условию $\int_{-\infty}^{\infty} K(x)dx = 1$,

например $K(x) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$, σ - радиус «области влияния», который является свободным параметром или может быть оценен методами непараметрической статистики.

Допустим, в новом пространстве плотность распределения будет постоянна: $\rho = \rho_0$. Например, можно выбрать $\rho_0 = 1/V$, где V – исходный объем фазового пространства данных (который, очевидно, конечен).

Положим, что метрика пространства имеет конформно-плоский вид:

$$ds^2 = \gamma(x) \sum_{ij} \delta_{ij} dx^i dx^j ,$$

где $\gamma(x)$ - конформный множитель, зависящий от точки $\boldsymbol{x}$. Для того, чтобы плотность данных в новом пространстве стала постоянной (может быть за исключением граничных областей), необходимо выбрать

$$\gamma(x) = \left(\frac{\rho_0}{\rho(x)} \right)^{2/n} .$$

С другой стороны, выбирая

$$\gamma(x) = \left(\frac{\rho(x)}{\rho_0} \right)^{\alpha}, \alpha > 0$$

получаем метрику, в которой сгущения данных выглядят тем более контрастно, чем больше параметр α. Малые расстояния между точками данных в такой метрике становятся еще меньше, большие – больше.

2.1.6. Вычисление расстояний для данных с пробелами

Отдельные значения координат точек данных могут оказаться либо недостоверными, либо вообще неизвестными, и тогда объект в многомерном пространстве надо представлять не точкой, а прямой или гиперплоскостью, параллельной координатным осям. Как измерять расстояния между объектами в таком случае?

Будем вычислять такие расстояния как расстояния между соответствующими геометрическими образами (между точкой и прямой, точкой и плоскостью, прямой и прямой и т.д.) В результате получим очень простое правило вычисления расстояний между объектами с пропущенными значениями признаков – *расстояния вычисляются в том подпространстве, в котором значения координат у объектов известны полностью*. Иными словами, при подсчете сумм в формулах вычисления расстояний те слагаемые, которые не могут быть вычислены из-за того, что отдельные значения признаков неизвестны, просто пропускаются.

Покажем, что это так. Допустим, объект имеет следующие значения признаков

$$X = (\xi_1, \xi_2, .., \xi_{k-1}, @, \xi_{k+1}, .., \xi_m).$$

Значком @ мы обозначили неизвестное значение признака ξ_k. Пусть $X(k)$ обозначает, что у объекта X неизвестно значение k-ого признака, а $X^0(k)$ обозначает следующий набор признаков:

$$X^0(k) = (\xi_1, \xi_2, .., \xi_{k-1}, 0, \xi_{k+1}, .., \xi_m),$$

то есть k-ое значение признака заменено нулем.

Тогда геометрический образ, который можно сопоставить объекту $X(k)$ – прямая

$X = X^0(k) + e_k t$, где e_k – единичный орт k-ой координатной оси.

Пусть значения признаков объекта Y известны полностью. Тогда найдем кратчайшее расстояние между X и $Y = (\eta_1, \eta_2 ... \eta_m)$:

$$\frac{d}{dt}\left((X-Y)^2\right) = 0 \;\Rightarrow\; (X-Y)e_k = 0 \;\Rightarrow\; t = Ye_k = \eta_k.$$

Это означает, что при вычислении расстояния неизвестное значение k-го признака у X необходимо заменить значением k-го признака Y. Но тогда

$$(X-Y)^2 = (X^0(k) - Y)^2 - \eta_k = (X^0(k) - Y^0(k))^2,$$

то есть при вычислении расстояний можно просто приравнять к нулю значение k-го признака объектов X и Y. Тогда, например, в случае евклидова расстояния получаем формулу для вычисления расстояния

$$d(X,Y) = \sqrt{\sum_{\substack{i=1 \\ \xi_i \neq @}}^{m} (\xi_i - \eta_i)^2}.$$

Рассмотрим, что будет, если у объекта Y неизвестно значение l-ого признака. Тогда

$$Y = Y^0(l) + e_l s,$$

$$\begin{cases} \dfrac{d}{dt}\left((X-Y)^2\right)=0 \\ \dfrac{d}{ds}\left((X-Y)^2\right)=0 \end{cases} \Rightarrow \begin{cases} (X-Y)e_k = 0 \\ (X-Y)e_l = 0 \end{cases} \Rightarrow \begin{cases} -Y^0(l)e_k + t - \delta_{kl}s = 0 \\ X^0(k)e_l + \delta_{kl}t - s = 0 \end{cases}.$$

Если $k \neq l$, то $t = \eta_k$, $s = \xi_l$ и приходим к той же ситуации, что и в случае с точкой и прямой. Если $k = l$, то $t = s$ и $(X-Y)^2 = \left(X^0(k) - Y^0(k)\right)^2$ и вновь мы просто пропускаем неизвестное значение признака.

Совершенно аналогично обстоит дело в общем случае, когда и X, и Y содержат произвольное число пропущенных признаков.

Теперь вернемся к случаю, когда у Y известны все значения признаков, а для X неизвестно значение k-го признака, но известно, что оно лежит в диапазоне $[a_k,b_k]$ (объект представляется отрезком прямой). Тогда правило вычисления расстояния между X и Y окажется следующим: если $\eta_k \in [a_k, b_k]$, то как и прежде пропускаем значение признака, иначе, если $\eta_k < a_k$, то полагаем $\xi_k = a_k$, а если $\eta_k > a_k$, то $\xi_k = b_k$ и считаем расстояние.

2.1.6. Гравитирующие данные

Процедуры предобработки данных могут заключаться не только в нормировке данных и выборе метрики, но и в целенаправленном изменении расстояний между точками для подчеркивания определенных особенностей структуры облака точек.

Рассмотрим предложенный в главе I вариант преобразования данных как облака гравитирующих точек. Припишем каждой точке X_i «массу» m_i. «Типичные представители» классов, например, могут иметь большую массу по сравнению с другими. В самом простом случае все массы равны.

Будем использовать евклидову метрику для измерения расстояний между точками, тогда

$$\left\|X_i - X_j\right\| = \sqrt{(X_i - X_j)^2}$$

Введем потенциал взаимодействия между точками X_i и X_j. Рассмотрим простой случай центральных сил и предположим, что потенциал не зависит от номеров точек i и j:

$$\varphi(X_i, X_j) = \varphi\left(\left\|X_i - X_j\right\|\right),$$

φ (r) – потенциальная функция, зависящая только от расстояния между точками. Тогда энергия взаимодействия пары точек

$$U_{ij} = m_i m_j \varphi\left(\left\| X_i - X_j \right\|\right)$$

и суммарная энергия

$$U = \frac{1}{2}\sum_{i \neq j} m_i m_j \varphi\left(\left\| X_i - X_j \right\|\right).$$

Тогда

$$\frac{\partial U}{\partial X_k} = \frac{1}{2}\sum_{i \neq j} m_i m_j \frac{\varphi'\left(\left\| X_i - X_j \right\|\right)}{\left\| X_i - X_j \right\|}\left(\delta_{ik}(X_i - X_j) - \delta_{jk}(X_i - X_j)\right) =$$

$$= m_k \sum_{i \neq k} m_i \frac{\varphi'\left(\left\| X_i - X_j \right\|\right)}{\left\| X_i - X_j \right\|}(X_k - X_i)$$

Будем считать, что легкая частица движется в вязкой среде, так что инерция движения гасится средой. Тогда уравнение движения оказывается первого порядка:

$$\frac{\partial X_i}{\partial t} = -\frac{\partial U}{\partial X_k}.$$

Проще всего решать это уравнение по схеме Эйлера, что дает следующую итерационную формулу:

$$X_i^{(t+1)} = X_i^{(t)} - \frac{\partial U}{\partial X_i}\Delta t.$$

Шаг Δt должен быть достаточно мал, чтобы обеспечить адекватность решения.

Теперь вспомним, что наша Вселенная данных должна раздуваться, чтобы обеспечить постоянную плотность данных. Этого эффекта можно достигнуть простой перенормировкой данных на первоначальный объем. Для этого нужно вычислить новый фазовый объем данных $V^{(t+1)}$ – объем прямоугольного параллелепипеда со сторонами, равными диапазонам значений признаков. Тогда получаем окончательный итерационный алгоритм:

Шаг 1. Движение частиц

$$X_i^{(t+1)} = X_i^{(t)} - \frac{\partial U}{\partial X_i}\Delta t$$

Шаг 2. Перенормировка.

На этом шаге можно предложить два варианта:

1) изотропный (расширение происходит одинаково во всех направлениях):

$\left(X_i^{(t+1)}\right)' = \left(\frac{V^{(0)}}{V^{(t+1)}}\right)^{1/m} \cdot X_i^{(t+1)}$, где $V^{(0)}$ – начальный объем данных.

В этом варианте возможно «схлапывание» Вселенной данных вдоль какого-то направления (параллелепипед может начать неограниченно вытягиваться).

2) неизотропный (расширение происходит так, чтобы сохранять форму исходного фазового объема)

$$\left(x_{ik}^{(t+1)}\right)' = \frac{\Delta_k^{(0)}}{\Delta_k^{(t+1)}} x_{ik}^{(t+1)},$$

где $\Delta_k^{(t)}$ – диапазон изменения значений k-го признака на шаге t.

Теперь рассмотрим некоторые варианты потенциальных функции $\varphi(r)$:

1) ньютоновский потенциал без сингулярности

$$\varphi(r) = \frac{\alpha}{r + \varepsilon^2}$$

Если $\alpha < 0$, то данные притягиваются и их кластерная структура становится более контрастной, при $\alpha > 0$ – отталкиваются и распределение становится более равномерным (но отношения соседства при этом нарушаются не слишком сильно) . Регуляризующая постоянная ε^2 нужна для того, чтобы точки данных не испытывали слишком сильных перемещений (за времена порядка Δt) вблизи $r = 0$.

2) потенциал ядерных сил

$$\varphi(r) = \frac{\alpha \exp(-r / r_0)}{r + \varepsilon^2}.$$

Смысл постоянной r_0 – эффективный радиус взаимодействия: на расстояниях больше r_0 точки данных практически «не замечают» друг друга.

3) использование других степенных зависимостей

$$\varphi(r) = \frac{\alpha}{r^{\beta} + \varepsilon^2}.$$

Так, для многомерного ньютоновского потенциала $\beta = \frac{m-1}{2}$.

2.1.8. Локальные статистики

Рассмотрим вкратце идею локальных статистик, которая тоже может быть использована при обработке данных перед применением процедур визуализации.

Выберем *k*-ый объект, который будет играть роль базового. Перейдем к новой векторной переменной $r' = r - r_k$, или, что тоже самое – отцентрируем данные так, чтобы *k*-ый объект оказался в начале координат:

$X'_p = X_p - X_k$, $p = 1 \ldots N$.

Теперь введем квадратичную метрику

$d_k^2(X'_p, X'_j) = (X'_p - X'_j) G_k (X'_p - X'_j)^T$,

G_k – положительно определенная симметричная матрица. Ее коэффициенты $g_{ij}^{(k)}$ могут быть настроены из различных соображений. Приведем два самых простых:

1) Оптимизация разделения классов. Коэффициенты $g_{ij}^{(k)}$ настраиваются из условия минимума функционала

$$J = \frac{\sum_{X'_i \in w_k} d_k(0, X'_i)}{\sum_{X'_i \notin w_k} d_k(0, X'_i)} \to \min,$$ где w_k – класс, к которому изначально принадлежал объект X_k.

Таким образом, объекты того же класса, что и X_k, оказываются в результате ближе к началу координат (где находится X_k), чем все остальные объекты.

2) Нормализация распределения

Матрица $G_k = W_k^{-1}$, где

$$W_k = \frac{1}{N} \sum_{i=1}^{m} X'_i (X'_i)^T .$$

Если матрица W_k невырождена, то G_k существует и в результате распределение векторов X'_i «с точки зрения» X'_k будет выглядеть похожим на нормальное.

Для решения упомянутых задач можно упростить G_k, выбрав ее диагональной (метрика окажется взвешенной евклидовой):

$d_k^2(X'_p, X'_j) = \sum_i w_{ki} (x'_{pi} - x'_{ji})^2$, $w_{kp} \geq 0$.

Тогда, например, для нормализации распределения можно выбрать $w_{ki} = (W_k)_{ii}^{-1}$, где $(W_k)_{ii}$ – *i*-ый диагональный элемент W_k.

Построенное пространство признаков можно анализировать (и визуализировать) любыми методами анализа многомерных данных. В результате мы получим описание набора данных «с точки зрения» объекта

X_k. Строя N таких пространств (и получая N описаний) и сравнивая их между собой, можно сделать полезные выводы об структуре данных. Разумеется, для эффективного сравнения N результатов обработки необходимо каким-то образом автоматизировать этот процесс, выдавая уже «сухой остаток» результатов сравнения.

Другой способ состоит в конструировании нового пространства, которое определенным образом «обобщает» все построенные пространства. Допустим, построены N метрик $d_k(X_i', X_j')$, $k = 1\ldots N$. Тогда можно записать такую матрицу удаленностей:

$$D = \begin{Bmatrix} d_1(X_1, X_1) & \ldots & d_1(X_1, X_N) \\ \ldots & & \ldots \\ d_k(X_k, X_1) & \ldots & d_k(X_k, X_N) \\ \ldots & & \ldots \\ d_N(X_N, X_1) & \ldots & d_N(X_N, X_N) \end{Bmatrix},$$

где на пересечении i-ой строки и j-го столбца стоит расстояние между i-ым и j-ым объектом в метрике, построенной для i-го объекта. Поскольку для различных объектов будут построены различные метрики, то для элементов матрицы D могут не выполняться условия симметричности ($d_i(X_i, X_j) \neq d_j(X_j, X_i)$) и неравенства треугольника ($d_i(X_i, X_j)$ может быть больше $d_i(X_i, X_k) + d_k(X_k, X_j)$). Поэтому такая матрица не может напрямую служить матрицей расстояний.

Попробуем устранить указанные нарушения, вводя новый класс так называемых $d^{(s)}$-метрик:

$$d^{(s)}(X_i, X_j) = a \cdot s[\varphi(d_{ik}), \varphi(d_{jk})] + b,\ k = 1\ldots N,$$

где d_{ik}, d_{jk} – элементы i-ой и j-ой строк матрицы D; $\varphi(x)$ – монотонная функция, например, $\varphi(x) = x$ или $\varphi(x) = \mathrm{rank}(x)$ – преобразование к порядковой шкале; s[…,…] – мера подобия двух последовательностей; a, b – константы, значения которых подбираются с целью масштабирования и выполнения метрической аксиомы неравенства треугольника. Тогда расстояние между объектами в $d^{(s)}$ метрике имеет ясный смысл – это различие двух последовательностей чисел – всех расстояний до объекта X_i в метрике d_i и всех расстояний до объекта X_j в метрике d_j. Другими словами – это мера сходства двух представлений данных – с точки зрения объекта X_i и X_j.

В качестве конкретных вариантов $d^{(s)}$-метрик могут быть использованы:

$d^{(d)}$-метрика:

$$d^{(d)}(X_i, X_j) = \sum_{k=1}^{N} \left[\varphi(d_{ik}) - \varphi(d_{jk})\right]^2$$

основана на простой евклидовой формуле для сравнения последовательностей. Она автоматически обеспечивает выполнение условия симметричности и неравенства треугольника, однако нивелирует некоторые важные особенности рядов $\varphi(d_{ik})$ и $\varphi(d_{jk})$.

$d^{(s)}$-метрика, основанная на стандартных мерах связи:

$$d^{(s)}(X_i, X_j) = \frac{1 - s_{ij}}{2},$$

где в качестве s_{ij}, можно выбрать коэффициент корреляции Пирсона (что обеспечивает условие симметрии, но возможны незначительные нарушения неравенства треугольника) или коэффициент связи τ-Кендалла между ранговыми признаками (обеспечивает выполнение всех условий).

После того, как исследователь остановится на том или ином варианте $d^{(s)}$ метрики, матрица D преобразуется таким образом, чтобы она могла служить матрицей расстояний для некоторого распределения точек данных. После этого можно применять те методы анализа данных, в которых исходной информацией служит матрица расстояний (например, методы метрического шкалирования).

2.2. Линейный анализ данных

Теперь перейдем к краткому описанию традиционных линейных методов анализа, которые так или иначе можно использовать для визуализации структуры данных.

Сначала выпишем формулы, которыми описывается случайная величина (вектор), подчиненная многомерному нормальному закону распределения. Плотность вероятности в этом случае равна

$$\rho(X) = C \exp\left(-\frac{1}{2}(X - MX)^T \Sigma^{-1} (X - MX)\right),\ C = \frac{1}{(2\pi)^{m/2}\sqrt{\det \Sigma}}.$$

Здесь MX – математическое ожидание X, Σ – ковариационная матрица

$$\Sigma = M((X - MX)(X - MX)^T).$$

Величина MX и ковариационная матрица оцениваются с помощью данных выборки:

$$MX \approx \frac{1}{N}\sum_i X_i \,,\ \Sigma \approx S = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - MX)(X_i - MX)^T \,.$$

2.2.1. Метод главных компонент

Как уже упоминалось, цель анализа данных – извлечение содержащихся в них информации. Задача снижения размерности набора данных – описание каждой точки данных с помощью величин, число которых меньше размерности пространства и которые являются функциями исходных координат

$$\eta_k = F_k(\xi_1, \xi_2, ..., \xi_m),\ k = 1 \ldots m',\ m' < m.$$

Функции F_k задают отображение F из исходного пространства R^m в пространство $R^{m'}$. Это отображение должно выбираться таким образом, чтобы на наборе данных X максимизировать определенный критерий, как-то отражающий количество сохраняемой при этом преобразовании информации. Выбирая отображение F из определенного класса отображений и критерий сохранения информации J, можно получать различные методы сокращения размерности пространства признаков.

В методе главных компонент F – некоторое линейное ортогональное нормированное отображение, т.е.

$$F_k(\xi_1, \xi_2, ..., \xi_m) = c_{1k}(\xi_1 - \mu_1) + ... + c_{mk}(\xi_m - \mu_m), \quad \text{где} \quad \mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij} -$$

средние по набору данных значения признаков, а на коэффициенты c_{ij} накладываются условия

$$\sum_{k=1}^{m} c_{ik}^2 = 1,\ \sum_{k=1}^{m} c_{ik} c_{jk} = 0\,,\ i, j = 1 \ldots m,\ i \neq j.$$

Вид критерия J :

$$J = \frac{D\eta_1 + ... + D\eta_{m'}}{D\xi_1 + ... + D\xi_m},$$

где D – вычисление дисперсии случайной величины.

Согласно этому критерию, количество сохраненной информации равно доле «объясненной» с помощью новых признаков $\eta_1 \ldots \eta_m$ дисперсии исходных признаков.

Первой главной компонентой называют такую нормированно-центрированную линейную комбинацию исходных признаков, которая

среди всех прочих нормированно-центрированных линейных комбинаций обладает на данном наборе данных наибольшей дисперсией.

Решим задачу нахождения первой главной компоненты. Для этого необходимо решить задачу

$$D(l_1 X) \to \max_{l_1},$$

где l_1 – вектор-строка размерности m, при условии нормировки $l_1 l_1^T = 1$. Вектор l_1 можно представлять как единичный вектор пространства данных, тогда $(l_1, X_i) l_1$ – точка проекции вектора X_i на вектор l_1.

Положим, что система векторов данных является центрированной, т.е. $E(X) \equiv \overline{X} = 0$ Тогда

$$D(l_1 X) = E(l_1 X)^2 = E(l_1 X X^T l_1^T) = l_1 E(X X^T) l_1^T = l_1 S l_1^T,$$

где S – ковариационная матрица набора данных X.

Введем функцию Лангранжа $\varphi(l_1, \lambda) = l_1 S l_1^T - \lambda(l_1 l_1^T - 1)$, тогда

$$\frac{\partial \varphi}{\partial l_1^T} = 2 S l_1^T - 2 \lambda l_1^T = 0,$$

и

$$(S - \lambda I) l_1^T = 0,$$

то есть l_1^T – собственный вектор ковариационной матрицы. Но $D(l_1 X) = l_1 S l_1^T = \lambda$, значит для того, чтобы $D(l_1 X)$ достигало максимума, нужно выбрать максимальное собственное значение. Нормированный собственный вектор, отвечающий этому значению и задаст направление первой главной компоненты в пространстве.

k-ой главной компонентой ($k = 2 \ldots m$) называется такая нормированно-центрированная линейная комбинация исходных признаков, которая не коррелирована с k-1 предыдущими главными компонентами и среди всех прочих нормированно-центрированных линейных комбинаций, не коррелированных с предыдущими k-1 главными компонентами обладает на данном наборе данных наибольшей дисперсией.

Можно показать, что k-ая главная компонента задается собственным вектором ковариационной матрицы данных, который соответствует k-ому по величине собственному значению.

Заметим, что решение задачи нахождения главных компонент не является инвариантным относительно смены масштабов у разных признаков. Поэтому перед применением метода данные нормируются так, чтобы все признаки были измерены в сопоставимых масштабах.

На *m'* главных компонент можно натянуть подпространство размерности *m'*. Легко понять, что сумма квадратов расстояний от точек данных до этого подпространства равна умноженной на *N* (число точек) остаточной дисперсии, «не объясненной» с помощью *m'* главных компонент, то есть $N(D\xi_{m'+1}+...+D\xi_m)=N(\lambda_{m'+1}+...+\lambda_m)$, где $\lambda_{m'+1},\ldots,\lambda_m$ – наименьшие по величине собственные значения. Отсюда становится понятным важное экстремальное свойство указанного подпространства:

Свойство 1. Сумма квадратов расстояний от исходных точек-наблюдений $X_1,\ldots,X_N$ до пространства, натянутого на m' главных компонент наименьшая среди всех других подпространств размерности m', полученных с помощью произвольной линейно-независимой системы из m' векторов.

Укажем еще два экстремальных свойства подпространства главных компонент.

Зададим правило перехода к меньшему числу переменных с помощью линейного преобразования:

$z_{ij}=\sum_{k=1}^{m}c_{jk}x_{ik}$, $j = 1\ldots m'$, $i = 1\ldots N$, или

$Z = CX$,

здесь x_{ik} – *k*-ая координата вектора данных X_i, z_{ij} – *j*-ая координата *i*-ой точки данных в некотором подпространстве меньшей размерности $R^{m'}$. Можно рассматривать эти формулы как проекцию точек данных из исходного пространства в $R^{m'}$.

Рассмотрим величины

$$M=\sum_{i=1}^{N}\sum_{j=1}^{N}(X_i-X_j)^2,$$

$$M(C)=\sum_{i=1}^{N}\sum_{j=1}^{N}(Z_i-Z_j)^2.$$

Их смысл – сумма квадратов расстояний между всевозможными парами объектов в исходном пространстве и в $R^{m'}$. Введем в качестве меры искажения суммы квадратов попарных расстояний между точками данных величину *M*-*M*(*C*). Можно показать [4], что

$$M-M(L)=\min_{C}\{M-M(C)\}=N^2(\lambda_{m'+1}+...+\lambda_m),$$

где *L* – матрица, задающая проекцию точек данных в подпространство, натянутое на *m'* главных компонент. Отсюда сдедует

Свойство 2. Среди всех подпространств размерности m' , полученных из исходного пространства данных с помощью произвольного линейного преобразования исходных координат, в подпространстве,

натянутом на первые m' главных компонент наименее искажается сумма квадратов расстояний между всевозможными парами рассматриваемых точек.

Наконец, введем меру искажения расстояний до начала координат и углов между прямыми, соединяющими всевозможные пары точек с началом координат. Обозначим ее $\|H - H(C)\|$, где

$$H = \{h_{ij}\}, \quad h_{ij} = (X_i, X_j),$$

$$H(C) = \{h_{ij}(C)\}, \quad h_{ij}(C) = (Z_i, Z_j),$$

а под $\|A\|$ – евклидова норма матрицы *A*. Оказывается, что

$$\|H - H(L)\| = \min_C \|H - H(C)\| = N^2(\lambda_{m'+1}^2 + ... + \lambda_m^2).$$

То есть справедливо

Свойство 3. Среди всех подпространств размерности m' , полученных из исходного пространства данных с помощью произвольного линейного преобразования исходных координат, в подпространстве, натянутом на первые m' главных компонент наименее искажаются расстояния от точек до начала координат, а также углы между прямыми, соединяющими всевозможные пары точек с началом координат.

Наконец, отметим, что указанное вначале требование «центрированности» данных не является принципиальным. Если в данных нет пробелов, то геометрический центр облака точек определен однозначно. Отличия в формулировках свойств 1-3 будет лишь в том, что вместо линейного подпространства, натянутого на главные компоненты надо рассматривать линейное многообразие, построенное на первых главных компонентах и проходящее через точку геометрического центра.

2.2.2. Итерационный алгоритм нахождения главных компонент

Используя экстремальное Свойство 1 подпространств, натянутых на главные компоненты, можно предложить итерационный алгоритм нахождения первой главной компоненты (ср. с [41,53]). Будем искать прямую в пространстве данных, заданную параметрическим уравнением

$$\mathbf{y} = \mathbf{a}t + \mathbf{b},$$

такую, что сумма квадратов расстояний от точек данных до этой прямой минимальна. Эта сумма, равная

$$Q = \sum_{i=1}^{N} (X_i - \mathbf{a}t_i - \mathbf{b})^2$$

является критерием, который можно минимизировать с помощью следующей простой процедуры:

Зададимся произвольными векторами **a** и **b**. Далее итерация алгоритма состоит из двух шагов:

Шаг 1. При заданных векторах **a** и **b** определяется набор $\{t_i\}$, $i = 1 \ldots N$:

$$\frac{\partial Q}{\partial t_i} = -2(X_i - \mathbf{a}t_i - \mathbf{b})\mathbf{a} = -2(X_i - \mathbf{b})\mathbf{a} - 2\mathbf{a}^2 t_i = 0,$$

$$t_i = \frac{(X_i - \mathbf{b})\mathbf{a}}{\mathbf{a}^2}.$$

Шаг 2. При заданном наборе $\{t_i\}$ определяются новые координаты векторов **a** и **b**:

$$\begin{cases} \dfrac{\partial Q}{\partial \mathbf{a}} = -2\sum\limits_{i=1}^{N}(X_i - \mathbf{a}t_i - \mathbf{b})t_i = 0 \\ \dfrac{\partial Q}{\partial \mathbf{b}} = -2\sum\limits_{i=1}^{N}(X_i - \mathbf{a}t_i - \mathbf{b}) = 0 \end{cases},$$

$$\begin{cases} \mathbf{a}\sum\limits_{i=1}^{N} t_i^2 + \mathbf{b}\sum\limits_{i=1}^{N} t_i = \sum\limits_{i=1}^{N} X_i t_i \\ \mathbf{a}\sum\limits_{i=1}^{N} t_i + \mathbf{b}N = \sum\limits_{i=1}^{N} X_i \end{cases},$$

что дает m систем линейных уравнений 2x2 для определения всех компонент векторов **a** и **b.**

Проверка на останов. Алгоритм останавливается, когда $\dfrac{\Delta Q}{Q} < \varepsilon$, где ΔQ – изменение величины Q за итерацию, а ε – малая величина.

Преимущество такого способа нахождения первой главной компоненты состоит в том, что он легко обобщается на случай, когда некоторые данные содержат неполные значения. Рецепт прост – если в соответствующей сумме встречается неизвестное значение, то такое слагаемое пропускается. Тогда, если неполных данных нет, то **b** дает вектор среднего значения всех координат: $\mathbf{b} = \dfrac{1}{N}\sum\limits_{i=1}^{N} X_i$, иначе – некоторый «эффективный» вектор среднего. Вектор **a** в случае полных данных задает направление первой главной компоненты, в случае неполных – «эффективную» первую главную компоненту.

Для того, чтобы найти вторую главную компоненту, поступают следующим образом:

1. Рассчитывается множество векторов первых остатков X': $X'_i = X_i - \mathbf{a}t_i - \mathbf{b}$. Это множество лежит в пространстве, ортогональном первой главной компоненте, размерностью на единицу меньше размерности исходного пространства данных.

2. Для нового множества векторов рассчитывается первая главная компонента. Она и будет второй главной компонентой исходного набора данных.

Для нахождения третьей главной компоненты ищется множество вторых остатков и для него определяется первая главная компонента, и т.д.

2.2.3. Модели линейного факторного анализа

Напомним, что метод главных компонент может быть сформулирован как задача оптимизации функционала *J* качества «сохранения» информации при заданном отображении *F* из исходного пространства в пространство меньшей размерности. В методе главных компонент в качестве функционала *J* выступает доля «объясненной» с помощью новых координат дисперсии.

В модели факторного анализа каждому вектору данных X_i сопоставляется набор из *m'* значений факторов $y_i^1,\ldots, y_i^{m'}$:

$$x_{ij} - \mu_j = \sum_{k=1}^{m'} q_{jk} y_i^k + u^j, j = 1\ldots m', \text{ или}$$

$$X_i - \overline{X} = QY_i + U,$$

где μ_j – среднее значение *j*-го признака, x_{ij} – значение *j*-го признака для *i*-го объекта, q_{ij} – «нагрузки» факторов, u^j – остаточная случайная компонента. При этом выполняются условия:

$$Ey^k = 0,\ Eu^k = 0,\ Dy^k = 1,$$

и $y_i^1,\ldots, y_i^{m'}$, $u^1,\ldots,u^{m'}$ попарно некоррелированы.

В качестве *F* выбирается линейное преобразование координат такое, чтобы выполнялись эти условия, и достигал максимума функционал

$$J(F) = 1 - \left\| R_X - R_{\hat{X}} \right\|^2,$$

где R_X – корреляционная матрица исходных признаков, $R_{\hat{X}}$ – корреляционная матрица «проекций» в пространство факторов $\hat{X}_i = QY_i$, $\|A\|$ – евклидова норма матрицы *A*.

Поясним выкладки. Матрица нагрузок *Q* размерами *m'×m* осуществляет линейное отображение из пространства факторов (размерности *m'*) в исходное пространство (размерности *m*). В результате

получается множество данных $\hat{X}_i = QY_i$, которое совпадает со множеством исходных данных с точностью до случайной компоненты *U*. Это множество лежит в некотором подпространстве размерности *m'*, натянутом на *m'* столбцов матрицы *Q*. Матрица *Q* и исходный набор факторов $y_i^1,\ldots,y_i^{m'}$, $i = 1\ldots N$ выбираются таким образом, чтобы корреляционная матрица набора данных $\hat{X}$ максимально точно воспроизводила корреляционную матрицу исходного набора данных. Таким образом, критерием количества «сохраненной» информации является здесь объяснение не дисперсии признаков, а их взаимной скоррелированности.

Что касается «шумовой» компоненты *U*, то обычно полагают, что она не зависит от распределения данных и подчинена *m*-мерному нормальному распределению с нулевым средним значением. Тогда ковариационная матрица распределения *U* имеет диагональный вид:

$V = E(UU^T)$, $V = diag(v_{11}...v_{mm})$, $v_{ii} = Du^i$.

Если исходное распределение данных предполагается центрированным, то его ковариационная матрица

$S = QQ^T + V$

Решением задачи факторного анализа называют пару матриц (*Q*,*V*), удовлетворяющую этому условию. Очевидно, что если одно такое решение существует, то одновременно решением является (*QC*,*V*), где *C* – произвольное ортогональное преобразование (поворот векторов-столбцов матрицы *Q*). По этой и другим причинам решение задачи факторного анализа является неоднозначным, поэтому для ее решения необходимо выбрать какие-либо дополнительные предположения о свойствах матрицы *Q*. Независимо от этих предположений итерационный метод решения задачи выглядит следующим образом.

Вначале задается нулевое приближение матрицы $V=V^{(0)}$.

Шаг 1. Получаем нулевое приближение матрицы $\Psi = QQ^T$, т.е. $\Psi^{(0)} = S - V^{(0)}$.

Шаг 2. С помощью $\Psi^{(0)}$ определяем нулевое приближение матрицы *Q*.

Алгоритм продолжается до получения необходимой точности.

Дополнительные предположения о структуре матрицы *Q* используются для реализации Шага 2 алгоритма. Рассмотрим два условия:

1) Q^TQ – диагональная матрица, причем диагональные элементы различны и упорядочены в порядке убывания.

Тогда

$A = Q^T Q = diag(\lambda_1 ... \lambda_{m'})$, $\lambda_1 > \lambda_2 > .. > \lambda_m$,

$\Psi Q = QQ^T Q = AQ$,

то есть *m'* столбцов $q_1 \ldots q_{m'}$ матрицы Q удовлетворяют уравнениям $\Psi q_i - \lambda_i q_i = 0$ на собственные значения матрицы Ψ. Собственные вектора матрицы Ψ, отвечающие первым *m'* по величине собственным значениям и составят очередное приближение матрицы нагрузок Q.

2) $Q^T V Q$ – диагональная матрица, причем диагональные элементы различны и упорядочены в порядке убывания.

Тогда

$A = Q^T V Q = diag(\lambda_1 ... \lambda_{m'})$, $\lambda_1 > \lambda_2 > .. > \lambda_m$,

$\Psi Q = V^{-1} QQ^T VQ = AV^{-1} Q$,

то есть *m'* столбцов $q_1 \ldots q_{m'}$ матрицы Q удовлетворяют уравнениям $\Psi q_i - \lambda_i v_{ii}^{-1} q_i = 0$ на обобщенные собственные значения матрицы Ψ. Обобщенные собственные вектора матрицы Ψ, отвечающие первым *m'* по величине обобщенным собственным значениям составляют очередное приближение матрицы нагрузок Q.

Подведем некоторые итоги. По данному набору данных, используя метод главных компонент или методы факторного анализа, можно построить линейные модели данных. Фактически эти методы строят специальное линейное многообразие меньшей размерности, на которое проецируются исходные данные. Это подпространство оказывается в некотором смысле оптимальным среди всех других линейных многообразий той же размерности. В случае метода главных компонент оптимальность заключается в том, что проекции данных максимально воспроизводят дисперсию исходных данных. В случае методов факторного анализа значения признаков проекций максимально похожи на исходные значения признаков в смысле взаимной корреляции. Следует заметить, что в случае, когда остаточные дисперсии (суммарные расстояния до построенного подпространства) невелики, оба метода дают сходные результаты (это становится особенно понятно, если рассмотреть условие 1 на структуру матрицы Q).

2.3. Моделирование данных с помощью нелинейных многообразий

Подпространство, натянутое на *m'* главных компонент обладает свойством «минимума остаточной дисперсии» – средний квадрат расстояния от точек данных до этого подпространства минимален среди всех других линейных подпространств размерности *m'*.

Кажется перспективной идея построения «главных поверхностей» – нелинейных многообразий, обладающих тем же оптимальным свойством. В литературе можно встретить работы [58,66,67], в которых строятся такие поверхности с использованием, например, метода максимального правдоподобия.

Если вид параметрической зависимости координат точек многообразия неизвестен (нет априорных соображений о структуре данных), то задача построения неограниченного нелинейного многообразия, то есть вычисление координат каждой из его точек, является весьма трудоемкой с точки зрения вычислений.

Мы с самого начала будем следовать пути, на котором моделирующее многообразие будет предполагаться ограниченным и задается в конечном числе своих точек. Другими словами, мы будем строить *точечную аппроксимацию многообразия*.

Допустим, что мы имеем размещенную в пространстве данных *сетку узлов*, на которую и будет натягиваться искомое многообразие. Для этого вводится определенная процедура *интерполяции* между узлами.

Сразу заметим, что задания одних только положений узлов в пространстве данных недостаточно для восстановления многообразия. Мы должны, кроме того, обладать информацией о том, какие узлы являются на нем *соседними*. То есть на конечном множестве узлов должны быть введены *отношения соседства*. Для излагаемого ниже алгоритма SOM (Self-Organizing Maps – самоорганизующиеся карты Кохонена) необходимо знать не только какие узлы являются соседними, но также и какие являются *вторыми* соседями, *третьими* соседями и т.д.

Рассмотрим, например двумерное многообразие. Будем считать, что узлы образуют прямоугольную или гексагональную сетку (см. рис.17). Тогда отношения соседства определяются естественным образом – можно считать, что каждый узел (кроме крайних) на прямоугольной сетке имеет четыре первых, восемь вторых и т.д. соседей, на гексагональной – шесть первых, двенадцать вторых и т.д. соседей. Если многообразие является трехмерным, а сетка прямоугольной, то каждый узел имеет, соответственно, шесть первых соседей, и т.д.

Рассмотрим два алгоритма построения сетки – алгоритм *самоорганизующихся карт Кохонена* (SOM) и его модификации, а также алгоритм построения *упругих сеток*.

2.4. Алгоритм SOM и его модификации

Самоорганизующиеся карты Кохонена (SOM) – это модифицированный алгоритм *линейного векторного квантования данных*, то есть представления N точек данных с помощью меньшего числа точек-образцов (*samples*). Каждый из образов представляет и заменяет собой локальное сгущение данных. В результате такой замены данные представляются с определенной ошибкой аппроксимации – среднеквадратичного расстояния от точки данных до ближайшего к ней образца:

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - y_{BMU}(X_i))^2},$$

где $y_{BMU}(X_i)$ – ближайший к точке данных X_i образец.

«Изюминкой» метода SOM оказалось то, что получаемая при его применении система узлов (образцов) оказывается определенным образом упорядочена. Узлы могут быть представлены в виде прямоугольной или гексагональной сетки. Соседние на этой сетке узлы в результате действия алгоритма SOM оказываются соседними в пространстве данных, что дает после размещения точек данных по ближайшим узлам в случае двумерной или трехмерной сетки возможность визуализировать данные.

Опишем исходный вариант алгоритма SOM.

1. Сетка узлов инициализируется – размещается в пространстве данных. Простейшим вариантом является случайное расположение узлов, другой вариант – размещение сетки в пространстве, натянутом на главные компоненты. Существуют и другие, более или менее эффективные схемы инициализации (см., например, [70]).
2. Выбирается случайным образом или по порядку точка данных X_i.
3. Среди всех узлов сетки выбирается ближайший к точке X_i. Обозначим его радиус вектор через y_{BMU} (BMU – Best Matching Unit).
4. *Все* узлы сетки двигаются по направлению к X_i по правилу:

$$(y_j)' = y_j + h\big(r(y_j, y_{BMU}), t\big)(X_i - y_j), j = 1\ldots p,$$

где p – количество узлов, $h(x,t)$ – так называемая *функция соседства* (neighborhood function), $r(y_1,y_2)$ – расстояние между узлами y_1 и y_2, но не в пространстве данных, а согласно введенным на сетке отношениям соседства (то есть, если y_1 и y_2 являются ближайшими соседями, то $r(y_1,y_2)=1$, если вторыми, то $r(y_1,y_2)=2$, и т.д.), t – номер итерации.

5. Шаги 2-4 алгоритма повторяются до тех пор, пока либо не будет достигнута определенная точность, или так, чтобы каждая точка данных несколько раз поучаствовала в процессе адаптации узлов (то есть kN итераций, где k – число порядка нескольких единиц).

Функция соседства $h(x,t)$ выбирается таким образом, чтобы достигать максимума при $x=0$ и монотонно спадать с ростом x. Наиболее популярными являются гауссов вид функции соседства и так называемая bubble-function.

Гауссов вид функции:

$$h(x,t)=\alpha(t)\exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2(t)}\right),$$

Bubble-function:

$$h(x,t)=\begin{cases}\alpha(t), x\le\sigma(t)\\ 0, \quad x>\sigma(t)\end{cases}.$$

Здесь $\alpha(t)$ – *темп* обучения, $\sigma(t)$ – *радиус захвата соседей* (neighborhood width). В изначальном варианте SOM эти функции не зависят от условий обучения и просто монотонно уменьшаются от некоторого начального значения до нуля, например, по линейному закону:

$\alpha(t)=\alpha_0(1-t/n_{iter})$,

$\sigma(t)=\sigma_0(1-t/n_{iter})$,

где n_{iter} – число итераций, α_0 – число порядка десятых или сотых, σ_0 – число порядка нескольких единиц.

Обсудим, что происходит в результате каждой итерации алгоритма. Больше всего испытывает смещение узел y_{BMU}. Он сдвигается в направлении выбранной точки данных на величину $\alpha(t)$. Остальные узлы испытывают тем меньшие смещения, чем они дальше от y_{BMU} в указанном выше «сеточном» смысле. Заметные смещения испытывают узлы, которые являются для y_{BMU} соседями порядка $\sigma(t)$. Так как функции $\alpha(t)$ и $\sigma(t)$ убывают со временем, то и темп обучения и число узлов, участвующих в коллективном движении, уменьшается. В результате сетка модифицируется все меньше и меньше и в конечном счете «застывает».

Обычно настройку сетки производят в два этапа:

Этап 1. Ordering. На этом этапе обычно выбирается $\alpha_0 \approx 0.1$, $n_{iter} \approx N$, σ_0 – выбирается так, чтобы в движении участвовало более половины узлов.

Этап 2. Fine-tuning. На этом этапе обычно выбирается $\alpha_0 \approx 0.01$, $n_{iter} \approx 10N$, σ_0 – выбирается так, чтобы в движении участвовало 2-3 узла.

Алгоритм SOM обеспечивает случайное движение узлов таким образом, что среднее расстояние от точки данных до ближайшего к ней узла постоянно уменьшается. При этом узлы упорядочиваются согласно введенным на системе узлов отношениям соседства. Полученная сетка узлов в результате становится более или менее гладкой, причем тем более гладкой, чем большие значения $\sigma(t)$ участвовали в настройке.

Со времени своего создания для алгоритма SOM было предложено множество модификаций, преследующих те или иные цели. Модификации алгоритма осуществлялись двумя основными техническими приемами:

а) изменение направления движения узлов – когда узел помимо того, что двигается по направлению к точке данных, смещается еще и в другом, выбираемом из тех или иных соображений, направлении;

б) настройка радиуса захвата соседей – когда величина $\sigma(t)$ становится разной для разных условий, в которых находится выбранный узел y_{BMU}.

Целью модификаций являлось ускорение работы алгоритма, улучшение точности аппроксимации при заданном числе узлов, динамическое (по ходу настройки) изменение числа узлов, улучшение гладкости или регулярности сетки.

Остановимся лишь на нескольких характерных модификациях.

Алгоритм Batch SOM [65]

Идея этой модификации – осуществлять движение узлов не поодиночке, а разом, за один такт. Последовательность действий – следующая:

1. Сетка узлов инициализируется.

2. Все множество данных разбивается на подмножества K_i, $i = 1 \ldots p$, p – число узлов. Для точек из подмножества K_i ближайшим узлом сетки является узел y_i. Назовем такое подмножество *таксоном* узла y_i. В результате все точки данных распределяются «по ближайшим узлам».

3. Вычисляются центры таксонов $\omega_i = \frac{1}{n_i} \sum_{X_i \in K_i} X_i$.

4. Положение каждого узла модифицируется по правилу

$y_i = \omega_i + \varepsilon\alpha_i$,

где α_i – среднее центров таксонов узлов-первых соседей, ε – некоторый параметр порядка десятых единицы. Таким образом соседние узлы «тянут» настраиваемый узел к себе. В результате сетка становится более регулярной.

5. Шаги 2-4 повторяются определенное количество раз.

Алгоритм регуляризации SOM [60]

Идея этой модификации – сделать сетку более гладкой, локально спрямить слишком большие ее изгибы. Для этого вводится понятие «идеальное положение узла» – для одномерной сетки это точка $\tilde{y}_j^1$ ортогональной проекции узла на прямую, соединяющую два соседних узла (см. рис.32а)

Действие алгоритма вполне аналогично стандартному SOM, за исключением того, что правило настройки узла в случае прямоугольной сетки теперь имеет вид

$$(y_j)' = y_j + h(r,t)(X_i - y_j) + \tilde{h}(t)(\tilde{y}_j^1 - y_j) + \tilde{h}(t)(\tilde{y}_j^2 - y_j),$$

где $\tilde{y}_j^1, \tilde{y}_j^2$ – ортогональные проекции на прямые, соединяющие верхнего и нижнего, левого и правого соседей соответственно, $\tilde{h}(t)$ – функция, монотонно убывающая с номером итерации. Таким образом, кроме того, что узел смещается в направлении точки данных, он также испытывает смещение в сторону точек $\tilde{y}_j^1$ и $\tilde{y}_j^2$, что приводит к частичному спрямлению линии, соединяющей три соседних узла и к более гладкой сетке.

Алгоритм Density Tracking SOM [62]

Идея этой модификации (похожей на Batch SOM) – сделать так, чтобы в областях скопления данных оказалось больше узлов, чем в более «разреженных» областях:

1. Сетка узлов инициализируется.

2. Как и в алгоритме Batch SOM, все множество данных разбивается на подмножества K_i, $i = 1 \ldots p$, p – число узлов. Для точек из подмножества K_i ближайшим узлом сетки является узел y_i.

3. Вычисляются центры таксонов $\omega_i = \frac{1}{n_i} \sum_{X_i \in K_i} X_i$;

4. Рассчитывается количество точек данных в каждом таксоне.

5. Положение каждого узла модифицируется по правилу

$y_i = \omega_i + \varepsilon\omega_j$,

где ω_j – центр одного из соседних таксонов, в котором содержится максимальное (среди всех соседей) количество точек, ε – некоторый параметр порядка десятых единицы. Узел смещается в сторону более «весомого» соседа, в окрестности которого содержится большее количество точек данных.

6. Шаги 2-5 повторяются определенное количество раз.

Алгоритм Adaptive SOM (AdSOM)

В работе [64] авторы указывают на общую проблему, возникающую при моделировании многомерного множества точек с помощью многообразий меньшей размерности. Для того, чтобы воспроизвести особенности многомерного множества, многообразие стремится «свернуться», образовать большое количество складок. При этом часть изгибов многообразия обусловлена самой структурой точек данных, часть – тем обстоятельством, что размерность многообразия не соответствует размерности множества (см. рис.32б).

В случае SOM это означает, что для некоторых точек данных ближайший узел сетки и второй по близости не являются соседями на сетке. Такие точки называются *неустойчивыми*. Отношение числа неустойчивых точек к общему количеству точек называется *топографической ошибкой* картирования.

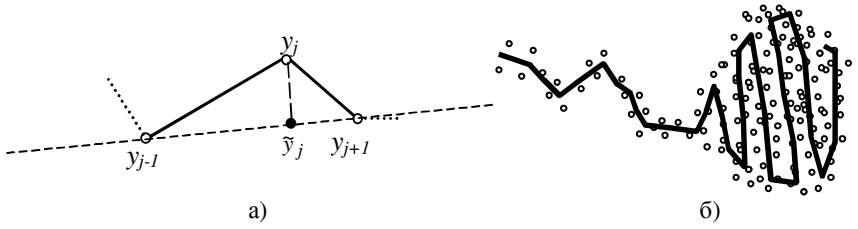


Рис. 32. а) Иллюстрация к алгоритму регуляризации SOM. Точка $\tilde{y}_j$ – проекция на прямую,
соединяющую два соседних узла.
б) Иллюстрация к алгоритму Adaptive SOM. Часть изгибов SOM многообразия обусловлена самой структурой точек данных, часть – тем обстоятельством, кривая стремится аппроксимировать множество большей размерности. Большая часть точек в шаровом скоплении справа окажутся *неустойчивыми*.

Модификация AdSOM сводит величину топографической ошибки практически к нулю за счет локальной настройки радиуса захвата соседей. В результате карта оказывается менее «изогнутой», что приводит к несколько худшей точности аппроксимации (такая жертва неизбежна – это и составляет суть дилеммы «регулярность-точность»).

Настройка радиуса захвата соседей производится следующим образом:

1. В алгоритме AdSOM для каждого из узлов y_k назначается свой собственный радиус завата соседей σ_k. Далее, как и в стандартном SOM выбирается точка данных X_i. Для нее определяются *два* ближайших узла y_i и y_j. Если эти узлы являются первыми соседями, то точка – устойчива, и для нее производится обычная процедура движения узлов. Если нет, то производится настройка всех величин σ_k следующим образом:

$$\sigma_k = \begin{cases} r(y_i, y_j), & \max\{r(y_k, y_j), r(y_i, y_k)\} \le r(y_i, y_j) \\ r(y_i, y_j) - s, & s < r(y_i, y_j), \quad s = \min\{r(y_k, y_j), r(y_i, y_k)\} \\ 1, & \text{иначе} \end{cases}$$

где $r(y_i,y_j)$ – упомянутое выше «сеточное» расстояние. Иными словами, радиус захвата соседей равен сеточному расстоянию между узлами y_i и y_j для всех узлов, находящихся на сетке между ними и линейно спадает до 1 вне области их «влияния».

2. Время от времени (каждые n_{rec} итераций) значения всех σ_k пересчитываются:

$$(\sigma_k)' = (\sigma_k)^{\beta},$$

где $\beta < 1$ – еще один параметр метода.

Иерархические алгоритмы SOM

В некоторых работах [52,63,71] сетка предполагается «растущей», то есть количество узлов время от времени меняется.

Самый простой алгоритм – количество узлов удваивается после окончания процесса настройки, если в результате не достигнута необходимая точность. Новые узлы помещаются в промежутки между уже настроенными узлами и сетка донастраивается.

Более гибкий метод – вычисление «растяжений» ребер сетки вдоль отдельных строк и столбцов сетки. В том ряду (вертикальном или горизонтальном), где суммарное растяжение оказалось наибольшим, ребра

делятся – посередине вставляются новые узлы. Сетка вновь донастраивается. Исследователь наблюдает за графиком изменения ошибки аппроксимации, и останавливает процесс деления ребер когда график ошибки выйдет на «плато» – свою пологую часть. Количество узлов сетки, которое отвечает началу пологой части графика ошибки считается «правильным» (для данного уровня точности).

Сделаем некоторые выводы. Почти во всех модификациях SOM вводятся параметры, с помощью которых сетка делается более регулярной – более гладкой или более равномерной, или лучше соответствующей локальной структуре данных. Вместе с тем ни в одной модификации нет указания на вид меры оптимальности построенной сетки. В следующем разделе мы явно введем такую меру, в результате оптимизации которой получится *алгоритм построения упругих сеток.* В алгоритме появятся два параметра – один будет явно «регулировать» гладкость построенной сетки, другой – ее равномерность.

Сначала мы рассмотрим случай двумерной прямоугольной сетки. Затем приведем алгоритм построения произвольной упругой сетки и рассмотрим различные способы ее настройки.

2.5. Алгоритм построения упругих сеток

2.5.1. Прямоугольная сетка

Рассмотрим двумерную прямоугольную сетку узлов, в которой p узлов по горизонтали, q узлов по вертикали. Перенумеруем узлы этой сетки с помощью двух индексов – y^{ij}, $i = 1\ldots p$, $j = 1\ldots q$. Сетка должна обладать следующими свойствами:

1) *Свойство близости к точкам данных*. Сетка должна быть в каком-то смысле аналогична плоскости первых двух главных компонент – оптимальной в смысле минимума среднего квадрата расстояния от точек данных до ближайшего узла при определенных ограничениях на свойства сетки.

2) *Свойство упругости по отношению к растяжению*. Это свойство до некоторой степени обеспечит *равномерность* сетки.

3) *Свойство упругости по отношению к изгибу*. Это свойство до некоторой степени обеспечит *гладкость* результирующего многообразия.

Как и в алгоритме Batch SOM разобьем все множество данных X на $p \times q$ подмножеств K_{ij} ($i = 1\ldots p$, $j = 1\ldots q$) (таксонов), в пределах каждого из которых точки подмножества оказываются ближе к узлу сетки y^{ij}, чем к

какому-нибудь другому узлу. Обозначим это обстоятельство следующим образом

$$K_{ij} = \left\{ x \in X \left\| y^{ij} - x \right\|^2 \to \min_{i,j} \right\}$$

В качестве меры близости сетки к точкам данных выберем величину среднего (на одну точку данных) квадрата расстояния от точки до ближайшего узла сетки.

Каждый узел сетки (кроме граничных) имеет четырех соседей, с которыми он соединяется «ребром». Чем больше средняя (на один узел) длина ребра, тем сильнее сетка «растянута», поэтому мы должны по возможности минимизировать эту величину. Таким образом, в минимизируемый функционал должны войти разности между положениями соседних узлов. Степень изогнутости определим с помощью точечной оценки величины второй производной (с помощью так называемых *вторых разностей*). В результате получим следующий функционал «качества» построенной сетки:

$$D = \frac{D_1}{|X|} + \lambda \frac{D_2}{pq} + \mu \frac{D_3}{pq} \to \min$$

где $|X|$ - число точек в X; λ, μ - коэффициенты упругости, отвечающие за растяжение и изогнутость стеки соответственно; D_1, D_2, D_3 – слагаемые, отвечающие за свойства сетки, именно:

$D_1 = \sum_{ij} \sum_{X_k \in K_{ij}} \left\| X_k - y^{ij} \right\|^2$ – является мерой близости расположения узлов сетки к данным. Здесь K_{ij} –подмножества точек из X, для которых узел сетки y^{ij} является ближайшим (*таксоны*);

$D_2 = \sum_{i=1}^{p} \sum_{j=1}^{q-1} \left\| y^{ij} - y^{i,j+1} \right\|^2 + \sum_{i=1}^{p-1} \sum_{j=1}^{q} \left\| y^{ij} - y^{i+1,j} \right\|^2$ – мера растянутости сетки;

$D_3 = \sum_{i=1}^{p} \sum_{j=2}^{q-1} \left\| 2y^{ij} - y^{i,j-1} - y^{i,j+1} \right\|^2 + \sum_{i=2}^{p-1} \sum_{j=1}^{q} \left\| 2y^{ij} - y^{i-1,j} - y^{i+1,j} \right\|^2$ – мера изогнутости (кривизны) сетки.

Отметим, что границы суммирования выбраны так, чтобы в функционале D_2 ребро не входило в сумму дважды.

Пусть метрика является евклидовой. В этом случае функционал D является квадратичным по положениям узлов y^{ij}, это значит, что при

заданном разбиении множества точек данных на таксоны для его минимизации потребуется решить систему линейных уравнений размерами $pq \times pq$. Следовательно, эффективным методом минимизации функционала D окажется такой алгоритм:

Шаг 0. Узлы сетки так или иначе располагаются в пространстве данных.

Шаг 1. При заданных положениях узлов сетки производится разбиение множества данных на таксоны – подмножества K_{ij}.

Шаг 2. При заданном разбиении множества точек данных на таксоны производится минимизация функционала D.

Шаги 1 и 2 повторяются до тех пор пока функционал D не перестанет изменяться (в пределах заданной точности). Процесс сходится, поскольку на каждом этапе минимизации величина D, очевидно, будет уменьшаться, вместе с тем она ограничена снизу нулем (величина D неотрицательна). Более того, он сходится за конечное число шагов, поскольку число вариантов разбиения точек данных на таксоны конечно (хотя и может быть весьма велико).

Выпишем явно коэффициенты матрицы системы линейных уравнений, которую необходимо решать на каждой итерации алгоритма минимизации. Приводимые ниже выкладки весьма просты, хотя и громоздки.

Непосредственное дифференцирование дает:

$$\frac{1}{2}\frac{\partial D}{\partial y^{kl}} = a_{kl}^{(-2)} y^{k-2,l} + a_{kl}^{(-1)} y^{k-1,l} + a_{kl} y^{k,l} + a_{kl}^{(+1)} y^{k+1,l} + a_{kl}^{(+2)} y^{k+2,l} +$$

$$+ b_{kl}^{(-2)} y^{k,l-2} + b_{kl}^{(-1)} y^{k,l-1} + b_{kl}^{(+1)} y^{k,l+1} + b_{kl}^{(+2)} y^{k,l+2} - \sum_{x \in K_{kl}} x$$

где

$$a_{kl}^{(-2)} = \frac{\mu}{pq}(1-\delta_{l,2})(1-\delta_{l,1}),$$

$$a_{kl}^{(-1)} = \frac{\lambda}{pq}(\delta_{l,1}-1) + \frac{2\mu}{pq}(\delta_{l,2}-1)(1-\delta_{l,1}),$$

$$a_{kl} = \frac{n_{kl}}{|X|} + \frac{\lambda}{pq}\left[4 - \delta_{k,1} - \delta_{k,p} - \delta_{l,1} - \delta_{l,q}\right] +$$

$$+ \frac{\mu}{pq}\begin{bmatrix}(1-\delta_{k,2})(1-\delta_{k,1}) + (1-\delta_{k,p-1})(1-\delta_{k,p}) - 2(1-\delta_{k,p})(1-\delta_{k,1}) + \\ + (1-\delta_{l,2})(1-\delta_{l,1}) + (1-\delta_{l,q-1})(1-\delta_{l,q}) - 2(1-\delta_{l,q})(1-\delta_{l,1})\end{bmatrix}$$

$$a_{kl}^{(+1)} = \frac{\lambda}{pq}(\delta_{l,q} - 1) + \frac{2\mu}{pq}(\delta_{l,q-1} - 1)(1-\delta_{l,q}),$$

$$a_{kl}^{(+2)} = \frac{\mu}{pq}(1-\delta_{l,q-1})(1-\delta_{l,q}),$$

$$b_{kl}^{(-2)} = \frac{\mu}{pq}(1-\delta_{k,2})(1-\delta_{k,1}),$$

$$b_{kl}^{(-1)} = \frac{\lambda}{pq}(\delta_{k,1} - 1) + \frac{2\mu}{pq}(\delta_{k,2} - 1)(1-\delta_{k,1}),$$

$$b_{kl}^{(+1)} = \frac{\lambda}{pq}(\delta_{k,p} - 1) + \frac{2\mu}{pq}(\delta_{k,p-1} - 1)(1-\delta_{k,p}),$$

$$b_{kl}^{(+2)} = \frac{\mu}{pq}(1-\delta_{k,p-1})(1-\delta_{k,p}),$$

где n_{kl} – число элементов в таксоне K_{kl}, δ_{ij} – символ Кронекера, множители вида $(1-\delta_{ij})$ введены для того, чтобы учесть «краевые эффекты». Если индексы k,l при y^{kl} не соответствуют никакому узлу сетки, то этот множитель автоматически обратит это слагаемое в ноль.

Уравнения $\frac{\partial D}{\partial y^{kl}} = 0$, $k=1\ldots p$, $l=1\ldots q$ дают m систем линейных уравнений (по одной на каждую из m компонент векторов y^{kl}).

«Вытянем» набор y^{kl} в один столбец. В результате вектор неизвестных будет

$$x = (y_{11},..,y_{1q},y_{2,1},..,y_{2,q},.....y_{(p-1),1},...y_{(p-1),q},y_{p,1}...,y_{p,q})$$

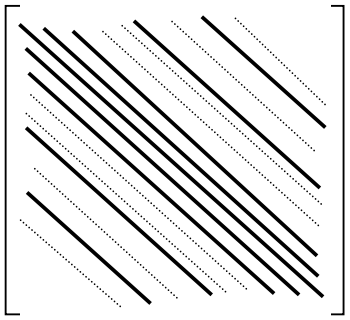
Система имеет вид $Ax=b$, где s-ая компонента вектора свободных членов равна

$b_s = \sum_{x \in K_{ij}} x$, $i = \left[\frac{s-1}{q}\right] + 1$, $j = s - \left[\frac{s-1}{q}\right] q$, $[\ldots]$ – операция взятия целой части числа.

$$A_{st} = \begin{cases} a_{kl}, i = k, j = l \\ a_{kl}^{(-1)}, i = k-1, j = l \\ a_{kl}^{(+1)}, i = k+1, j = l \\ a_{kl}^{(-2)}, i = k-2, j = l \\ a_{kl}^{(+2)}, i = k+2, j = l \\ b_{kl}^{(-2)}, i = k, j = l-2 \\ b_{kl}^{(-1)}, i = k, j = l-1 \\ b_{kl}^{(+1)}, i = k, j = l+1 \\ b_{kl}^{(+2)}, i = k, j = l+2 \\ 0, else \end{cases}, \text{ где } \begin{cases} i = \left[\frac{s-1}{q}\right] + 1 \\ j = s - \left[\frac{s-1}{q}\right] q \\ k = \left[\frac{t-1}{q}\right] + 1 \\ l = t - \left[\frac{t-1}{q}\right] q \end{cases}$$

Таким образом, матрица системы имеет вид

A =



то есть в ней есть пять «центральных» и четыре «побочные» диагонали. На протяжении всей работы алгоритма значения элементов матрицы остаются неизменными, изменяются лишь компоненты вектора *b* (которые зависят от способа разбиения множества точек данных на таксоны).

2.5.2. Непрямоугольные сетки

Упругая сетка вовсе не обязательно должна быть прямоугольной. В некоторых случаях, например, может оказаться разумным применение гексагональной сетки (см. рис.17). Для гексагональной сетки каждая точка имеет не два, а три «правых нижних соседа» и три «ребра жесткости». Вид функционалов D_2, D_3 в этом случае принимает вид:

$$D_2 = \sum_{i=1}^{p}\sum_{j=1}^{q-1}\left\|y^{ij} - y^{i,j+1}\right\|^2 + \sum_{i=1}^{p-1}\sum_{j=1}^{q}\left\|y^{ij} - y^{i+1,j}\right\|^2 +$$

$$+ \sum_{i=2}^{p}\sum_{\substack{j=1 \\ j-нечет.}}^{q-1}\left\|y^{ij} - y^{i-1,j+1}\right\|^2 + \sum_{i=1}^{p-1}\sum_{\substack{j=1 \\ j-четн.}}^{q}\left\|y^{ij} - y^{i+1,j+1}\right\|^2 ;$$

$$D_3 = \sum_{i=2}^{p-1}\sum_{j=1}^{q}\left\|2y^{ij} - y^{i-1,j} - y^{i+1,j}\right\|^2 +$$

$$\sum_{i=2}^{p}\sum_{\substack{j=2 \\ j-нечет.}}^{q-1}\left(\left\|2y^{ij} - y^{i-1,j-1} - y^{i,j+1}\right\|^2 + \left\|2y^{ij} - y^{i,j-1} - y^{i+1,j+1}\right\|^2\right) +$$

$$+ \sum_{i=2}^{p}\sum_{\substack{j=2 \\ j-четн.}}^{q-1}\left(\left\|2y^{ij} - y^{i,j-1} - y^{i+1,j+1}\right\|^2 + \left\|2y^{ij} - y^{i,j+1} - y^{i+1,j-1}\right\|^2\right)$$

Вычисление производных в этом случае приводит к громоздким выражениям, поэтому рассмотрим обобщение метода на случай произвольной двумерной сетки.

Теперь будем описывать сетку, явно указывая на способ соединения узлов и правило образования «ребер жесткости».

Положим, что сетка состоит из p узлов, каждому соответствует радиус-вектор y^i, $i = 1\ldots p$. Некоторые узлы соединяются между собой ребрами упругости E^i – их количество s штук. Три узла могут образовать ребро жесткости R^j – пусть их будет r штук. Каждое из s ребер упругости может иметь вес w_i, а каждое из ребер жесткости – вес v_j. Тогда общий вид функционала имеет вид

$$D = \frac{1}{|X|}D_1 + \frac{1}{p}\left(\lambda D_2 + \mu D_3\right),$$

$$D_1 = \sum_{i=1}^{p}\sum_{X_i \in K_i}\left(y^i - X_i\right)^2,$$

$$D_2 = \sum_{i=1}^{s} w_i \left(E^i(1) - E^i(2)\right)^2 ,$$

$$D_3 = \sum_{i=1}^{r} v_i \left(R^i(3) + R^i(2) - 2R^i(1)\right)^2 ,$$

где через $E^i(1)$ обозначен радиус вектор начала i-го ребра упругости, а через $E^i(2)$ – конец, далее $R^i(2)$, $R^i(3)$ – крайние точки ребра жесткости, $R^i(1)$ – центральный узел i-го ребра жесткости.

Введем обозначение

$$\Delta(x, y) = \begin{cases} 1, x = y \\ 0, x \neq y \end{cases}$$

Тогда дифференцирование дает

$$\frac{1}{2}\frac{\partial D_1}{\partial y^j} = n_j y_j - \sum_{X_i \in K_j} X_i ,$$

$$\frac{1}{2}\frac{\partial D_2}{\partial y^j} = \sum_{i=1}^{s} w_i \left(E^i(1) - E^i(2)\right)\left[\Delta(E^i(1), y^j) - \Delta(E^i(2), y^j)\right],$$

$$\frac{1}{2}\frac{\partial D_3}{\partial y^j} = \sum_{i=1}^{r} v_i \left(R^i(3) + R^i(2) - 2R^i(1)\right)\begin{bmatrix} \Delta(R^i(3), y^j) + \Delta(R^i(2), y^j) - \\ -2\Delta(R^i(1), y^j) \end{bmatrix}$$

Обозначим

$$\Delta E^{ij} \equiv \Delta(E^i(1), y^j) - \Delta(E^i(2), y^j) ,$$

$$\Delta R^{ij} \equiv \Delta(R^i(3), y^j) + \Delta(R^i(2), y^j) - 2\Delta(R^i(1), y^j) ,$$

тогда

$$\frac{1}{2}\frac{\partial D_2}{\partial y^j} = \sum_{i=1}^{s} w_i \left(E^i(1) - E^i(2)\right)\Delta E^{ij} =$$

$$-\sum_{i=1}^{s} w_i \sum_{k=1}^{p} y_k \Delta(E^i(1), y^k)\Delta E^{ij} - \sum_{i=1}^{s} w_i \sum_{k=1}^{p} y_k \Delta(E^i(2), y^k)\Delta E^{ij} =$$

$$= \sum_{k=1}^{p} y_k \sum_{i=1}^{s} w_i \Delta E^{ij} \Delta E^{ik} = \sum_{k=1}^{p} y_k e_{jk};$$

$$\frac{1}{2}\frac{\partial D_3}{\partial y^j} = \sum_{i=1}^{r} v_i \left(R^i(3) + R^i(2) - 2R^i(1)\right)\Delta R^{ij} =$$

$$= \sum_{i=1}^{r} v_i \sum_{k=1}^{p} y^k \left(\Delta(R^i(3), y^k) + \Delta(R^i(2), y^k) - 2\Delta(R^i(1), y^k)\right)\Delta R^{ij} =$$

$$= \sum_{k=1}^{p} y^k \sum_{i=1}^{r} v_i \Delta R^{ij} \Delta R^{ik} = \sum_{k=1}^{p} y^k r_{jk},$$

где $e_{jk} = \sum_{i=1}^{s} w_i \Delta E^{ij} \Delta E^{ik}$, $r_{jk} = \sum_{i=1}^{r} v_i \Delta R^{ij} \Delta R^{ik}$. В результате получаем систему уравнений

$$\frac{1}{2}\frac{\partial D}{\partial y^j} = \sum_{k=1}^{p} y^k \left(\frac{n_j \delta_{jk}}{|X|} + \frac{\lambda}{p} e_{jk} + \frac{\mu}{p} r_{jk} \right) - \frac{1}{|X|} \sum_{X_i \in K_j} X_i = 0, j = 1 \ldots p .$$

Отсюда получаем систему линейных уравнений для нахождения одного из компонентов векторов положений узлов $\{y^i\}$:

$$\sum_{k=1}^{p} a_{jk} y^k = \frac{1}{|X|} \sum_{X_i \in K_j} X_i ,$$

$$a_{jk} = \frac{n_j \delta_{jk}}{|X|} + \frac{\lambda}{p} e_{jk} + \frac{\mu}{p} r_{jk}, \; j = 1 \ldots p \qquad (*)$$

Поясним еще раз – вид уравнений одинаков для каждого из компонентов векторов y^i. Более того, элементы матрицы a_{ij} одинаковы для всех компонентов – необходимо лишь суммировать соответствующие компоненты векторов X_i в правой части уравнения.

Кроме того, если значения коэффициентов λ, μ не меняются, и способ соединения узлов в сетке неизменен, то в выражении для элементов матрицы a_{ij} меняется лишь первое слагаемое, связанное с разбиением множества точек данных на таксоны.

2.5.3. Применение сложных сеток

Отметим, что с помощью предложенного выше алгоритма можно строить одномерные, двумерные, вообще n-мерные сетки, необходимо лишь правильно разбить сетку на узлы, ребра упругости и ребра жесткости. Для целей визуализации более подходят двумерные упругие сетки, поскольку их можно «развернуть» на плоскости.

Предложенный способ построения непрямоугольных сеток позволяет строить замкнутые сетки (например, сферические или тороидальные). Для этих сеток существует некоторая сложность, связанная с отображением или «разворачиванием» их на плоскость. Как результат, такие сетки дают развертки, в которых границы «склеены» друг с другом.

Более полезным может оказаться применение способов настройки сетки, которые основаны на регулировании локальной гладкости и равномерности. В алгоритме упругих карт параметры метода λ, μ регулируют равномерность и гладкость сетки в целом, а с помощью весов

отдельных ребер ω_i, ν_i можно регулировать локальные свойства сетки, «подгоняя» ее под локальные «детали»распределения данных.

Приведем некоторые соображения, которые могут быть здесь использованы. Все алгоритмы излагаются для случая прямоугольной сетки.

1) Адаптивный рост сетки

Идеи, лежащие в основе иерархических алгоритмов SOM могут применяться и для настройки упругой сетки.

Для того, чтобы рассчитать суммарные «натяжения» в вертикальном или горизонтальном ряду ребер упругости, нужно рассчитать величины

$$\varepsilon_k = \sum_{E^i \in G_k} w_i \left(E^i(2) - E^i(1) \right)^2 ,$$

где G_k – подмножество ребер упругости, которые образуют вертикальный или горизонтальный ряд. Тот ряд, для которого величина натяжения оказалась наибольшей, делится новыми узлами пополам, веса w_i при этом присваиваются новым образованным ребрам те же, что были у исходного делящегося ребра (см. рис.32).

2) Адаптивная настройка структуры сетки

После настройки сетки можно увеличивать точность аппроксимации следующим образом (рассмотрим случай прямоугольной сетки):

1. Рассчитывается число точек в каждом таксоне;
2. Для каждого из квадратов сетки вычисляется суммарное количество точек в таксонах его вершин;
3. Тот квадрат, для которого число точек оказалось наибольшим, делится на 4 части (см. рис.33); при этом веса вновь образовавшихся ребер упругости увеличиваются вдвое (поскольку длина ребра уменьшается вдвое, то вдвое должна увеличиться и его жесткость).

3) Адаптивная настройка весов

Существует способ перестроить сетку таким образом, чтобы ее структура оставалась прежней, а плотность узлов оказалась приблизительно пропорциональна плотности данных в окрестности этого узла. Этого можно достигнуть, делая сетку менее равномерной.

1. Рассчитывается число точек в каждом таксоне;

2. Для каждого из ребер упругости сетки вычисляется суммарное количество точек в таксонах его вершин;

3. Далее все веса ребер упругости пересчитываются, например, следующим образом:

$(w_i)' = \alpha w_i$, $\alpha = n_i / n_{aver}$,

где n_i – число точек в таксоне i-го узла, n_{aver} – среднее число точек в таксоне.

В результате те ребра, в окрестности которых плотность данных ниже среднего, становятся «мягче» и удлиняются, где выше – становятся «жестче» и укорачиваются. В конце концов данные должны быть распределены по таксонам более или менее равномерно.

4) Уменьшение топографической ошибки

Можно поставить целью уменьшение топографической ошибки и настраивать гладкость сетки таким образом, чтобы исчезли неустойчивые точки (см. выше описание алгоритма AdSOM):

1. Рассчитывается число точек в каждом таксоне, при этом подсчитывается величина «неустойчивости» узла – относительная доля неустойчивых точек в таксоне;

2. Выбирается узел y_1 с самой большой величиной неустойчивости;

3. По очереди выбираются неустойчивые точки таксона узла, выбранного на предыдущем шаге, для каждой выполняется процедура пересчета весов ребер жесткости R^i следующим образом:

3.1. Для каждой из выбранных точек рассчитывается второй по близости узел сетки –y_2; в силу определения неустойчивых точек он не является соседним;

3.2. От узла y_1 до y_2 «прокладывается» кратчайший маршрут из ребер жесткости – то есть ищутся ребра жесткости, с помощью которых можно соединить два узла, и для каждого из вошедших в маршрут ребер вес увеличивается:

$(v_i)' = v_i + \alpha n$,

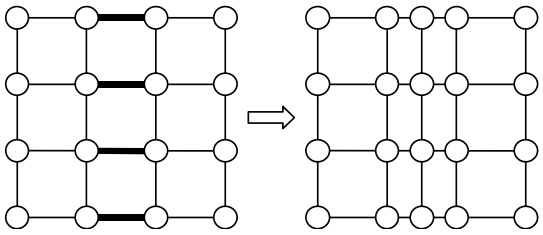
где α – параметр, n – длина маршрута (количество ребер жесткости, вошедших в маршрут);

4. Для всех ребер жесткости, центральные узлы которых обладают *нулевой* величиной неустойчивости вес уменьшается:

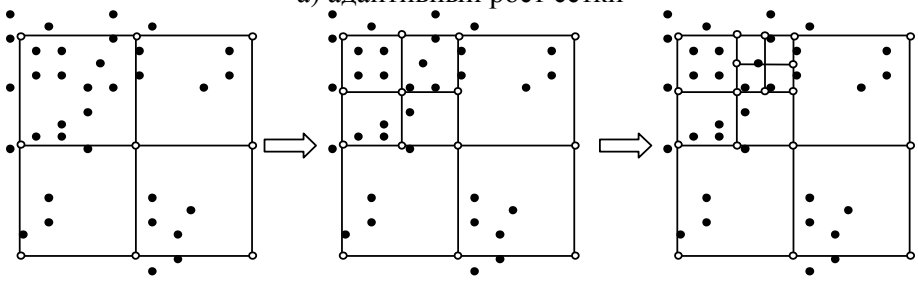
$(v_i)' = v_i - \alpha$;

5. Сетка донастраивается;

6. Процесс повторяется до тех пор, пока либо величина топографической ошибки не достигнет допустимого уровня, либо число итераций не сатнет больше некоторого заданного числа.



а) адаптивный рост сетки



б) Адаптивная настройка структуры сетки

Рис. 33. Иллюстрации к работе алгоритмов адаптивного изменения структуры сетки

а) жирным выделен тот ряд ребер, которые оказались наиболее растянуты в пространстве данных;

б) квадрат, в окрестности вершин которого плотность узлов наибольшая, делится на четыре части, после донастройки сетки процесс продолжается.

2.5.4. Настройка сетки “online”

После того, как сетка настроена на определенном «базовом» наборе данных, может потребоваться «дообучить» сетку на новых данных, которые поступают динамически. Рассмотрим, как можно донастроить сетку на одном примере, который не входил в исходное обучающее множество.

В формуле (*) раздела 2.5.2 можно заметить, что от самих данных в матрице системы линейных уравнений зависит лишь первое слагаемое – $n_i / |X| \delta_{ij}$, где n_i – число точек данных в таксоне узла y_i. В столбце свободных членов стоят величины $\sum_{X_i \in K_i} X_i$ – сумма векторов данных, принадлежащих таксону K_i.

Это значит, что для однозначного построения сетки можно указать не весь набор данных, а только его частотный «словарь» – набор векторов-«образцов» по числу узлов в сетке и количество точек данных в каждом

таксоне. С другой стороны, любое расположение сетки в пространстве данных задает определенный словарь.

Рассмотрим два варианта донастройки сетки:

1. *Вся информация о данных недоступна, доступен лишь словарь данных*, тогда вновь поступившая точка данных *x'* слегка корректирует словарь – увеличивает частоту n_k того «образца», который оказался к ней ближайшим (обозначим его x_k) и сдвигает сам этот образец на вектор

$$\Delta x_k = \frac{x' - x_k}{n_k + 1}.$$

Тогда изменяются и матрица системы, и вектор свободных членов. Вид новой системы $(A + \Delta A)(y + \Delta y) = (b + \Delta b)$, тогда для нахождения поправки Δy имеем $(A + \Delta A)\Delta y = \Delta b - \Delta A y$.

Поправка ΔA матрицы системы имеет единственный ненулевой элемент $1/N$ на пересечении k-го столбца и k-ой строки (N – число точек данных), поправка Δb столбца свободных членов – вектор с единственным ненулевым k-ым элементом, равным $\frac{\Delta x_k (n_k + 1) + x_k}{N}$. Таким образом, для нахождения новых положений узлов нужно найти малую поправку Δy, которая является решением системы

$$\sum_{i=1}^{p} (a_{ij} + \frac{1}{N}\delta_{ij}\delta_{jk})\Delta y_i = \frac{\Delta x_k (n_k + 1) + x_k - y_k}{N}\delta_{jk}, j = 1..p,$$

где p – число узлов.

Можно решить эту систему для каждой из компонент векторов и получить новые положения всех узлов $(y_i)' = y_i + \Delta y_i$. Однако дешевле с точки зрения вычислений найти приближенное решение и считать, что смещается только один, ближайший к *x'* узел, а остальные остаются на месте, то есть $\Delta y_i = \Delta y_i \delta_{ik}$. Тогда получаем схему пересчета сетки для вновь поступившей точки данных *x'*:

а) Находим ближайший узел y_k.

б) Пересчитываем словарь $(x_k)' = x_k + \Delta x_k = \frac{x' + x_k n_k}{n_k + 1}$, $(n_k)' = n_k + 1$.

в) Рассчитываем новое положение узла $(y_k)' = y_k + \Delta y_k = \frac{y_k a_{kk} N + x'}{a_{kk} N + 1}$.

г) Пересчитываем коэффициенты матрицы $(a_{kk})' = a_{kk} + 1/N$.

д) Увеличиваем N: $N = N + 1$.

2. *Вся информация о данных доступна*. Тогда схема пересчета положений узлов сетки остается такой же как и в предыдущем случае, но в расчет можно включить шаг, учитывающий перерасчет словаря с учетом изменившегося положения узла (так как положение узла изменилось, то окружающие его данные могут «перескочить» с одного на другой таксон). Добавляем шаг

е) Учет «перескоков». Перебираем все точки в таксоне K_k и сравниваем расстояние до ближайшего узла с расстоянием до узла-ближайшего соседа (в пространстве данных). Если узел-сосед оказался теперь для точки ближайшим, то она «перескакивает» в его таксон. Также перебираем точки в таксоне узла-соседа. Если теперь для точки оказался ближайшим узел y_k, то она «перескакивает» в таксон K_k.

2.5.5. Доопределение сетки до многообразия

Используя сетку узлов, можно построить на ее основе непрерывное многообразие. Для этого подходит любой метод, в котором восстанавливается многообразие по конечному числу заданных точек.

Формально задача ставится следующим образом. Требуется восстановить вектор функцию $r = r(u,v)$ по значениям в конечном числе ее точек $\{y_i = r_i(u_i,v_i),\ i = 1\ldots p\}$. Пара чисел u_i, v_i – приписываемые заранее каждому узлу *внутренние координаты* на двумерной карте.

Рассмотрим самый простой случай построения кусочно-линейного многообразия. Для этого двумерная сетка предварительно *триангулируется* – для всего множества узлов указывается правило объединения их в *треугольники*.

Зададим внутренние координаты точки карты u, v в области определения вектор функции $r(u,v)$. Тогда, используя заданное правило триангуляции, можно определить тот треугольник, которому принадлежит выбранная точка (он будет ближайшим). Допустим, что этот треугольник образован узлами с номерами i_1, i_2, i_3. Можно определить относительные координаты α, β точки относительно этих узлов. Например, определим их так:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_{i1} \\ v_{i1} \end{pmatrix} + \alpha \begin{pmatrix} u_{i2} - u_{i1} \\ v_{i2} - v_{i1} \end{pmatrix} + \beta \begin{pmatrix} u_{i3} - u_{i1} \\ v_{i3} - v_{i1} \end{pmatrix}.$$

Тогда имеем систему линейных уравнений для определения α, β:

$$\alpha \begin{pmatrix} u_{i2} - u_{i1} \\ v_{i2} - v_{i1} \end{pmatrix} + \beta \begin{pmatrix} u_{i3} - u_{i1} \\ v_{i3} - v_{i1} \end{pmatrix} = \begin{pmatrix} u - u_{i1} \\ v - v_{i1} \end{pmatrix},$$

решив которую, найдем искомое значение вектор-функции:

$$r(u,v) = y_{i1} + \alpha(y_{i2} - y_{i1}) + \beta(y_{i3} - y_{i1}).$$

Построение такого кусочно-линейного многообразия наименее трудоемко с точки зрения вычислений. Однако, можно использовать и более изощренные техники – например, многомерную формулу Карлемана [5].

В результате мы получаем гладкую поверхность, проходящую через все узлы построенной сетки.

2.5.6. Проецирование данных на построенную карту

Как мы уже указывали, алгоритм построения карты не предполагает ее двумерности. С помощью одного и того же алгоритма можно строить сетки различной размерности – отличие будет лишь в способе соединения узлов в ребра жесткости и упругости. Вместе с тем, сетки разной размерности имеют различную «специализацию»:

а) *одномерные сетки* – наиболее подходящие кандидаты для решения задач *нелинейного факторного анализа*;

б) *двумерные сетки* удобнее всего использовать для *визуализации данных*;

в) *трехмерные сетки* задают некоторое эффективное трехмерное пространство, пригодное для решения задач визуализации данных, когда двух измерений недостаточно. Например, такая ситуация складывается при визуализации *вложений временных рядов* (см. раздел 2.6).

г) *m*-мерные сетки могут использоваться для эффективного *квантования данных*, то есть сжатия информации.

Наиболее простой способ переноса точки данных из пространства на построенную карту – *кусочно-постоянное проецирование*, когда каждой точке данных сопоставляется *ближайший узел карты*. Для такого способа проецирования даже не обязательно доопределять сетку узлов до многообразия.

Более интересны кусочно-непостоянные способы проецирования. Например, если по сетке узлов строится кусочно-линейное многообразие, то можно предложить естественный способ *проецирования в ближайшую точку карты*.

Начнем с одномерных сеток. Введем понятие *расстояния от точки до отрезка*. Будем определять его следующим образом.

1. Выполним ортогональное проецирование на прямую, содержащую отрезок. Если проекция принадлежит отрезку, то искомое расстояние – это расстояние до проекции.

2. Иначе искомое расстояние – это расстояние до ближайшего конца отрезка.

Расстояние до треугольника найдем так:

1. Выполним ортогональное проецирование на плоскость, содержащую треугольник. Если точка проекции принадлежит треугольнику, то искомое расстояние – это расстояние до проекции.

2. Иначе искомое расстояние – это расстояние до ближайшей стороны треугольника (каждая из которых представляет собой отрезок).

Расстояние до тетраэдра найдем так:

1. Выполним ортогональное проецирование в трехмерное линейное многообразие, содержащее тетраэдр. Если точка проекции принадлежит тетраэдру, то искомое расстояние – это расстояние до проекции.

2. Иначе искомое расстояние – это расстояние до ближайшей стороны тетраэдра (каждая из которых представляет собой треугольник).

Продолжая аналогично, можно найти расстояние до любого k-мерного симплекса.

Одномерная кусочно-линейная карта состоит из отрезков. Поэтому ближайшая точка такой карты – это ближайшая точка ближайшего отрезка ломаной. Соответственно, ближайшая точка двумерной карты – это ближайшая точка ближайшего треугольника, и т.д.

Проектор в ближайшую точку построенного многообразия понятен и идея его неизменна для карт разной размерности. Однако, для некоторых целей он может оказаться слишком грубым (существуют целые области пространства, из которых точки проецируются в один узел). В случае одномерных сеток возможно применение алгоритма центрального проецирования [41]. В этом алгоритме центр проецирования выбирается в пересечении двух перпендикуляров к ребрам, которые прилегают к узлу, ближайшему к точке данных.

2.6. Моделирование вложений временных рядов

Геометрическую метафору облака точек в многомерном пространстве можно сопоставить не только данным, изначально

представленным в виде таблицы «объект-признак», но и временному ряду. По ряду значений z_t, t=1,2…, M меняющейся величины мы можем сделать его *d-мерное вложение*, где каждая точка соответствует куску ряда из d последовательных элементов: $X_k = (z_k, z_{k+1}, \ldots, z_{k+d-1})$. Таким образом, временному ряду сопоставляется таблица, в которой первая строка – это первые d значений ряда, вторая – d значений ряда, начиная со второго, в третьей – d-окно сдвигается еще на одну позицию и т.д. (будем называть d-окном «рамку» ширины d, в которой мы рассматриваем кусок ряда). Такой способ представления временного ряда называется *вложением по Таккенсу* [20].

Соответствующая последовательность точек в d-мерном пространстве опишет определенную траекторию. Интересно, что эта траектория может целиком лежать в некотором k-мерном подпространстве ($k < d$), или не слишком сильно выходить за его пределы (находиться в нем в пределах заданной точности).

Будем называть *пространством вложения* d-мерное пространство, в которое вложена траектория. Каждая точка этого пространства соответствует вырезанной окном последовательности значений ряда длиной d. Назовем такую последовательность *паттерном*. Совокупность паттернов – это все возможные варианты поведения временного ряда на отрезках времени в d элементов.

На рис.34 приведено 4 примера наборов паттернов для простых функциональных временных рядов, и соответствующие им траектории в многомерном пространстве. Из рисунков видно, что при любой длине нарезки d (однако большей вводимой ниже *размерности ряда по Таккенсу*) и шаге дискретизации Δt траектория оказывается вложенной в подпространство меньшей размерности. Так, постоянный временной ряд оказался нуль-мерным (в пространстве вложения ему соответствует единственная точка $X_0 = (C,C\ldots C)$, $C = const$), линейный ряд – одномерный (в пространстве вложения он представлен прямой, не проходящей через начало координат и параллельной диагонали $x_1 = x_2 = \ldots = x_d$), функция синуса порождает «двумерный» временной ряд, а синус со второй гармоникой оказался четырехмерным.

Легко понять почему это так. Рассмотрим линейный ряд. Если нам известна первая точка паттерна (или, вообще, одна любая точка), то этот паттерн однозначно идентифицируется и восстанавливается. Иными словами, для того, чтобы определить поведение линейного ряда на ближайшие d шагов, достаточно знать одно-единственное начальное значение паттерна. Для синуса это не так. Здесь в наборе паттернов есть нисходящие, восходящие и «экстремальные» паттерны (которые также вначале либо возрастают, либо убывают), поэтому для однозначного

восстановления паттерна необходимы две его точки. В последнем примере есть два вида восходящих и два вида нисходящих паттернов, это и приводит к тому, что траектория целиком лежит в четырехмерном линейном подпространстве (заметим, что часть траектории оказалась, тем не менее, плоской).

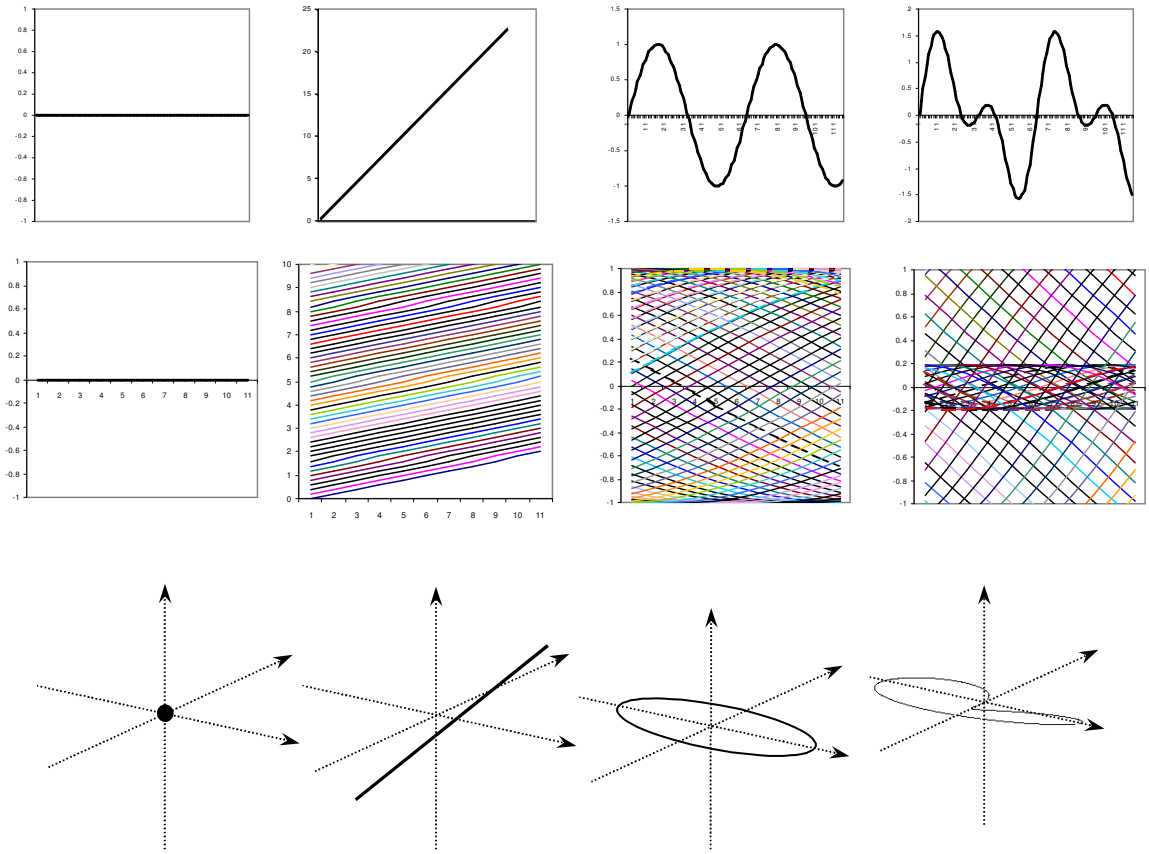


Рис.34. Иллюстрации к вложению временных рядов по Таккенсу

1 ряд графиков. Простые функциональные ряды, построенные на основе функций:

1. $f(t)=0$ 2. $f(t)=t$. 3. $f(t)=\sin(t)$. 4. $f(t)=\sin(t)+0.8\sin(2t)$.

2 ряд графиков. Семейство паттернов, построенных по соответствующим рядам.

1. Паттерн единственен; 2. Семейство патернов «одномерно»;
3. Семейство паттернов «двумерно»; 4. Семейство паттернов «одномерно».

3 ряд графиков. Траектории вложений рядов в 10-мерное пространство в пространстве, натянутом на 3 первые главные компоненты

1. Траектория – точка; 2. Траектория – прямая;
3. Траектория – эллипс; 4. Траектория – «четырехмерная» фигура Лиссажу.

Добавление белого шума к функциональному ряду синуса приводит к тому, что он остается двумерным лишь приблизительно. Добавление линейного тренда может привести к тому, что ряд станет одномерным (если функция в результате станет монотонной), или увеличит размерность ряда. Вообще, любая монотонная функция порождает одномерный временной ряд.

Задачу прогнозирования временного ряда можно поставить следующим образом: по m точкам ($m < d$) идентифицировать паттерн ряда и дать прогноз на d-m значений вперед. Если нужен более глубокий прогноз, то для выбора следующего паттерна можно воспользоваться m последними точками предыдущего. Таким образом, ряд будет «конструироваться» из заданного набора паттернов. Оправданность такого подхода основывается на предположении о том, что найденное семейство паттернов поведения является характерным для данного временного ряда. С другой стороны, «внезапное» появление принципиально новых паттернов свидетельствует о разладках в поведении ряда, смене режима, что тоже может оказаться полезной информацией.

Сколько же точек нужно взять для того, чтобы при выполнении указанного предположения однозначно восстановить паттерн? Ответ на этот вопрос зависит от того, какова эффективная размерность семейства паттернов, т.е. сколько паттернов в среднем проходит через ε-окрестность выбранной точки в окне. Это число и определит размерность временного ряда по Таккенсу.

Один из способов найти эффективную размерность линейного пространства, в котором находится траектория ряда в пространстве вложений – использование метода главных компонент. Действительно, первая главная компонента в пространстве вложений дает оптимальное в смысле среднеквадратичного расстояния между паттернами, приближенное описание заданного временного ряда с помощью «одномерного» ряда. Другими словами, набор паттернов реального ряда заменяется на однопараметрическое семейство *моделирующих паттернов*. В d-окне паттерны этого семейства не пересекаются. В случае использования плоскости первых двух главных компонент получаем двухпараметрическое моделирующее семейство и т.д.

Рассмотрим подробнее случай «одномерного» моделирования. Допустим, уравнение первой главной компоненты в пространстве вложений имеет вид

$$X = X_0 + Y_1 t\,,\ \|Y_1\|^2 = 1,\ t - \text{параметр}.$$

Здесь X_0 – точка пространства вложений, соответствующая геометрическому центру траектории ряда. В d-окне ей соответствует

«усредненный» паттерн $\hat{X}_0$. Вектору главной компоненты Y_1 в d-окне соответствует «несмещенный» паттерн-образец $\hat{Y}_1$. В результате реальный паттерн $\hat{X}_k$ заменяется на модельный

$$X'_k = X_0 + Y_1(X_k - X_0, Y_1),$$

скобками обозначено стандартное скалярное произведение.

Совершенно аналогично происходит моделирование с помощью двумерного ряда по формуле

$$X'_k = X_0 + Y_1(X_k - X_0, Y_1) + Y_2(X_k - X_0, Y_2),$$

где Y_2 – вектор второй главной компоненты в пространстве вложений.

Будем теперь рассматривать среднюю точку паттерна как «текущую» точку ряда. Тогда $\frac{d-1}{2}-1$ первых значений в окне (предположим, что d – нечетное) соответствует «прошлому» точки, а $\frac{d-1}{2}-1$ последних значений «будущему». Метод сглаживания временных рядов с помощью «скользящего среднего» заключается в замене текущего значения ряда на среднее по всем точкам из d-окна. С точки зрения изложенного выше это соответствует проецированию в пространстве вложений на единичный вектор с совпадающими компонентами $Y_{cp} = \left(\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}}, ..., \frac{1}{\sqrt{d}}\right)$, то есть соответствует «одномерному моделированию» ряда. У вектора Y_1 компоненты не равны и это задает другой способ фильтрации ряда. Впрочем, если временной ряд достаточно гладкий и шаг дискретизации мал, то $Y_1 \approx Y_{cp}$.

2.7. Мультикартирование

Может оказаться так, что точность моделирования данных с помощью двумерного многообразия, вложенного в пространство большей размерности окажется недостаточно хорошей. Точность моделирования можно повышать с помощью таких приемов:

а) увеличение числа узлов;

б) уменьшение упругой энергии карты (карта начинает более плотно прилегать к данным).

И в том, и в другом случае карта становится менее гладкой, что приводит к ухудшению ее обобщающих способностей (см. раздел 1.2).

Идея мультикартирования или *итерационного моделирования* состоит в том, чтобы описывать данные последовательностью карт для разных наборов данных.

Первая карта описывает само облако точек в исходном пространстве данных.

Вторая карта описывает ошибки описания данных первой картой. Набор данных, на котором она строится – вектора первых остатков, полученные вычитанием из радиус-векторов исходных данных радиус-векторов их проекций на карте.

Третья карта описывает вторые остатки и т.д.

Моделирование данных с помощью набора карт можно назвать нелинейным факторным анализом данных. Карту данных можно назвать *нелинейным фактором* (одномерным, двумерным, трехмерным и т.д.).

Главная отличительная особенность подхода от традиционного факторного анализа – его нелинейность, то есть линейная модель данных $X_i = Qy_i + U$ заменяется на нелинейную:

$$X_i = F(u_i, v_i) + \Delta_i,$$

где u_i, v_i – внутренние координаты точки X_i на карте (рассматриваем двумерный случай), F – вектор-функция. Остатки Δ_i , в свою очередь, описываются следующей нелинейной моделью: $\Delta_i = F_1(u_i^1, v_i^1) + \Delta_i^{(1)}$ и так далее.

Сколько карт понадобится, чтобы описать данные с нулевой ошибкой? Рассмотрим случай полных данных и пусть в качестве факторов выступают направления главных компонент. Гарантируется, что ошибка описания станет нулевой, если использовать m линейных факторов, где m – размерность пространства. Это очевидно, поскольку тогда моделирование сводится к замене координатной системы. Вектора первых остатков лежат в линейном многообразии размерности m-1, ортогональном первому фактору, вектора вторых остатков образуют облако эффективной размерности m-2 и т.д.

Ситуация меняется, если используются нелинейные факторы. В этом случае вектора первых, вторых, третьих остатков уже не лежат в линейных многообразиях меньшей размерности – они, вообще говоря, по прежнему образуют m-мерное образование. В этом смысле никакое количество факторов не гарантирует нулевой ошибки моделирования, можно лишь гарантировать монотонное убывание дисперсии облака остатков с ростом числа факторов. Нелинейные факторы имеет смысл использовать только если они обеспечивают меньшую остаточную дисперсию, чем линейные для числа факторов, меньшего m. Соответственно, не имеет смысла использовать больше m нелинейных факторов.

2.8. Информационное моделирование с помощью упругих карт

Допустим, исследователь построил по данным карту самих данных, карту остатков, карту вторых остатков и т.д. – всего *s* карт. Назовем последовательность таких карт *информационной моделью данных*.

Подчеркнем отличие построенной модели от традиционных нейросетевых информационных моделей, описанных, например, в [44], где образом модели является «черный ящик» с *p* входами и *m-p* выходами (см. рис. 35а). Заметим, что на вход нейросети не могут подаваться «пробелы» – они должны быть уже определенным образом заполнены.

При моделировании данных с помощью многообразий на вход модели подается вектор пространства $\boldsymbol{x}$, а на выходе снимается «смоделированный» вектор того же пространства $\tilde{x}$ (например, точка проекции на карте). Схема напоминает нейросетевую архитектуру «узкое горло», однако существенное отличие заключается в том, что среди компонент $x^1,x^2,x^3\ldots,x^m$ вектора $\boldsymbol{x}$ могут быть «пустые» значения, а на выходе эти значения окажутся заполненными (восстановленными).

Таким образом, исследователь может в рамках одной и той же модели произвольно разделить множество признаков на «входные» и «выходные». Подавая на вход вектор, в котором заполнены входные компоненты вектора, в выходные оставлены «пустыми», на выходе можно снять восстановленные значения выходных признаков.

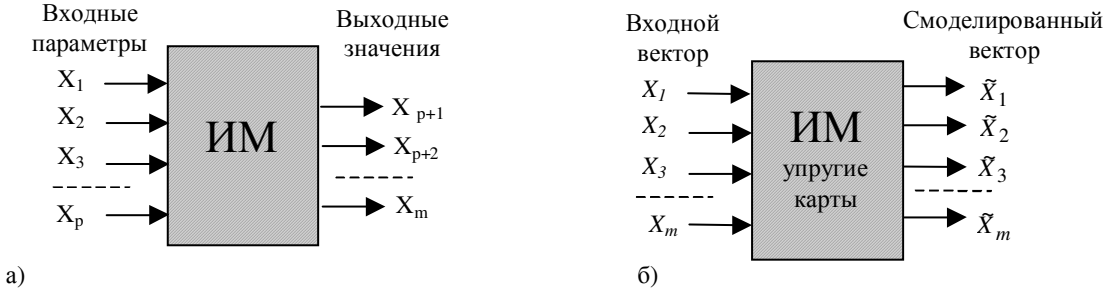


Рис. 35. Два типа информационных моделей.
а) «стандартная» нейросетевая схема моделирования;
б) моделирование данных с помощью многообразий.

Это открывает возможности для выявления взаимосвязей признаков, решения задач прогнозирования и построения регрессионных зависимостей между признаками.

С помощью информационной модели набор данных описывается с определенной *ошибкой обучения*:

$$Err_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \tilde{X}_i)^2},$$

где X_i – вектор значений признаков i-го объекта в исходном наборе данных, $\tilde{X}_i$ – смоделированный вектор, N – число точек в наборе данных. В случае, когда исследователь ограничивается одной картой, величина Err_t совпадает введенной ранее величиной MSPE. Естественно ожидать, что с увеличением числа карт, ошибка обучения будет уменьшаться. Удобно пользоваться безразмерной величиной

$$err_t = \frac{Err_t}{\sigma},$$

где σ – корень из среднего квадрата расстояния данных до их среднего значения (среднеквадратичное отклонение).

Число карт выбирается таким образом, чтобы выполнялось условие $err_t < \varepsilon$, где ε – *допустимая относительная ошибка* обучения. Величина $(1-\varepsilon)\cdot 100\%$ называется *точностью моделирования*.

Увеличивая число карт, можно добиться сколь угодно малой ошибки описания обучающей выборки. Это обстоятельство можно использовать для задачи сжатия информации. Однако качество модели в большей степени характеризуется величиной *ошибки обобщения*:

$$Err_g = \sqrt{\frac{1}{N_g}\sum_{i=1}^{N_g}(X_i^{(g)} - \tilde{X}_i^{(g)})^2},$$

где $X^{(g)}{}_i$ – вектор значений признаков i-го объекта *тестирующей выборки*, $\tilde{X}_i^{(g)}$ – соответствующий смоделированный вектор, N_g – число точек в тестирующей выборке. Тестирующая выборка обычно формируется с помощью случайного удаления из процесса настройки модели определенного процента примеров, которые в дальнейшем и формируют N_g тестирующих примеров. Понятно, что чем больше этот процент, тем точнее оценка величины ошибки обобщения, меньше примеров остается для настройки самой модели. Введем безразмерную величину ошибки обобщения:

$$err_g = \frac{Err_g}{\sigma}.$$

Если значение (1-err_g)·100% достигает 80-90%, то моделирование можно считать успешным, поскольку такова точность прогноза хорошего эксперта.
Рассмотрим возможные задачи, которые можно решать с помощью информационных моделей, построенных с помощью метода упругих карт:

1. Визуализация данных.

Основной особенностью и преимуществом построения *двумерных* информационных моделей является возможность наглядного представления данных и ошибок описания данных моделью. Любая из *s* построенных карт позволяет визуально анализировать распределение самих данных или погрешностей описания.

2. Группирование объектов.

С помощью карты можно производить разбиение объектов на группы. Это можно делать визуально, оценивая компактность и форму имеющихся в наборе сгущений данных. Если деление на группы нельзя произвести четко или не имеет смысла, то можно использовать следующий прием (ср. с примером из). На карту накладывается двумерный непрерывный цветной спектр. В результате каждая точка получает определенный цвет, причем точки, соседние на карте, получают близкие цвета. Полученные цвета можно использовать, например, если точки данных можно расположить на географической карте. Тогда сходные по значению признаков точки на карте будут иметь близкие цвета.

3. Оценка значимости признаков.

В наборе данных могут оказаться дублирующие друг друга (сильно скоррелированные) признаки или шумящие признаки, не несущие в себе никакой существенной информации для целей моделирования. Введем понятие значимости *i*-ого признака на *j*-ом объекте

$$\chi(i, X_j) = \left| err_t - err_t^{(i)} \right|,$$

где $err_t^{(i)}$ - ошибка обучения полученная при замене *i*-го признака у *j*-го объекта на некоторое предопределенное значение (в качестве такого значения может использоваться среднее значение признака, «пустое» значение или ноль). Назовем такую замену «фиксацией» признака.

Если в результате «фиксации» ошибка обучения изменилась не слишком сильно, то такой признак для данного объекта можно считать малозначимым.

Значимость *i*-го признака определим как

$$\chi_i = \frac{1}{N}\sum_{j=1}^{N}\chi(i, X_j)$$

Введем также понятие значимости набора из *k* признаков на *j*-ом примере

$$\chi(i_1 i_2 ... i_k, X_j) = \left| err_t - err_t^{(i_1 i_2 ... i_k)} \right|,$$

где $err_t^{(i_1 i_2 ... i_k)}$ - величина ошибки обучения, полученная при замене у j-го объекта признаков с номерами $i_1, i_2 \ldots i_k$ на предопределенное значение. Тогда значимость набора из *k* признаков определим как

$$\chi(i_1 i_2 ... i_k) = \frac{1}{N}\sum_{j=1}^{N}\chi(i_1 i_2 ... i_k, X_j).$$

Значимость признака может быть мала в двух случаях. Во-первых, признак действительно может оказаться малоинформативным. Во-вторых, он может быть зависим от остальных признаков. Малоинформативный признак не имеет большого смысла использовать в задачах информационного моделирования, тогда как с зависимыми признаками ситуация более сложна. Пусть несколько признаков образуют группу, в которой значения каждого могут быть восстановлены с определенной точностью при использовании значений других признаков из группы. При анализе признаков этой группы окажется, что значимость каждого отдельного признака мала. Однако, если один или несколько признаков в группе будут «зафиксированы», то значимости остальных могут резко возрасти.

Введем понятие значимости признака *i* на *j*-ом объекте при условии что признаки $i_1, i_2 \ldots i_k$ фиксированы $\chi_f(i, X_j | i_1 i_2 ... i_k)$ и соответствующую условную значимость *i*-го признака $\chi_f(i | i_1 i_2 ... i_k) = \frac{1}{N}\sum_{j=1}^{N}\chi_f(i, X_j | i_1 i_2 ... i_k)$.

4. Отбор признаков

В рамках построенной информационной модели можно решать задачу выделения среди всех признаков группы наиболее информативных независимых признаков. Решение этой задачи имеет прикладное значение, но может потребовать большого количества вычислений. В этом направлении существуют следующие подходы [4]:

а) *полный перебор сочетаний* применим лишь в случае небольших размерностей, так как требует C_m^k оценок значимости;

б) *методы последовательного формирования группы*, когда из группы последовательно удаляются или добавляются один или несколько признаков по определенному критерию;

в) *стохастические методы*, когда группа формируется случайно и в зависимости от оценки значимости вероятность последующего выбора признаков, входящих в группу увеличивается или уменьшается;

г) *методы целенаправленного поиска*, позволяющие отбросить заведомо неприемлемые сочетания признаков.

Предложим алгоритм последовательного уменьшения группы признаков, основанный на использовании понятия условной значимости χ_f.

1. Из всех признаков фиксируется тот, чья значимость $\chi(\underline{i})$ минимальна. Пусть это будет признак i_1.

2. Рассчитываются значения $\chi_f(i|i_1)$.

3. Фиксируется признак i_2, для которого величина $\chi_f(i_2|i_1)$ наименьшая.

4. Рассчитываются значения $\chi_f(i|i_1 i_2)$ и т.д.

5. Процесс повторяется до тех пор, пока в группе не останется заданное число n признаков.

Другой класс задач связан с разбиением всех признаков на группы $S_1 \ldots S_l$ в пределах каждой из которой признаки оказываются коррелированными, а признаки, принадлежащие разным группам относительно нескоррелированы. Для линейных моделей соответствующий метод носит название *метода экстремальных группировок* [4].

Для выявления групп взаимосвязанных признаков может оказаться полезной визуализация *транспонированной задачи*, когда таблица данных транспонируется и признаки начинают играть роль номеров объектов в новом пространстве, а номера бывших объектов – названия новых признаков. Каждая точка при такой визуализации соответствует определенному признаку, близкие точки соответствуют скоррелированным признакам. При таком представлении признаков исследователь может произвести разбиение на группы с помощью визуального анализа распределения точек на карте.

5. Восстановление пропущенных значений в данных.

Любой точке пространства признаков в информационной модели сопоставляется точка моделирующего многообразия в исходном

пространстве, пространстве первых остатков, вторых остатков и т.д. В конечном итоге при наличии s карт точке X исходного пространства сопоставляется точка

$$\tilde{X} = P_{M0}(X) + P_{M_1}(X_1) + P_{M_2}(X_2) + ... + P_{M_{s-1}}(X_{s-1}),$$

где $P_{Mi}(X)$ – оператор проецирования в пространстве i-ых остатков, M_0 – исходная карта данных, M_1 – карта остатков, M_2 – карта вторых остатков и т.д., X_i – остатки: $X_1 = X - P_{M0}(X)$, $X_2 = X_1 - P_{M1}(X_1)$ и т.д.

Важно то, что если вектор X содержит пропущенные значения признаков, то в результирующем векторе $\tilde{X}$ все равно будут известны все компоненты. Значения соответствующих компонент $\tilde{X}$ можно использовать для восстановления пробелов в X или прогноза намеренно пропущенных значений.

6. Прогнозирование значений отдельных признаков.

Тот факт, что информационная модель позволяет правдоподобным образом восстановить отсутствующие компоненты вектора пространства, дает возможность использовать ее для прогнозирования значений отдельных признаков по информации (возможно неполной), содержащейся в других признаках.

Разделим всю совокупность признаков на *прогнозируемые* и *информационные*. Задавая значения части информационных признаков в векторе X, а остальные считая неизвестными, значения прогнозируемых признаков можно взять из вектора $\tilde{X}$. Чем большее число информационных признаков будет известно, тем более правдоподобным будет прогноз.

Рассмотрим в качестве примера таблицу данных экологических измерений. Допустим, что значения вредных выбросов различных химических веществ измерялись в разных точках, каждая из которых имеет две координаты на географической карте. Каждому измерению соответствуют окружающая температура воздуха, давление, время суток, направление ветра и другие характеристики. Таким образом, эти характеристики, а также географические координаты могут играть роль информационных признаков, а измеренные значения выбросов – роль прогнозируемых величин. Можно получить информационную раскраску географической карты, восстанавливая значения выбросов только по географическим координатам, считая значения других информационных признаков неизвестными. Эти значения будут восстанавливаться некоторым правдоподобным образом, причем для каждой точки они будут разными, соответствуя тем условиям, в которых были проведены реальные

измерения. Исследователь может «задать» температуру, и карта выбросов несколько видоизменится. Далее можно уточнять давление, время суток и другие значения информационных признаков, получая при этом более правдоподобные прогнозы и соответствующие им раскраски карты. Задавая временной ряд изменения значений информационных признаков исследователь получит возможность моделировать динамическую картину изменения значений прогнозируемых признаков. Таким образом, исследователь получает инструмент для моделирования составляющих экологической обстановки в зависимости от окружающих условий, причем тем более реалистичного моделирования, чем больший объем информации об экологических измерениях имеется в его распоряжении.

Работу такой прогнозирующей системы можно сравнить с экспертом, который по неполной информации, имеющейся в его распоряжении, «вспоминает» наиболее похожие случаи из своего опыта и корректирует свои выводы сообразно новым условиям. При этом чем больше информации попадает в руки такого эксперта, тем точнее и правдоподобнее оказывается его прогноз.

С помощью информационной модели исследователь может также решать обратную задачу прогнозирования – так, например, моделировать условия, при которых значение выбросов может превышать определенный порог. Об особенностях прямой и обратной задачи информационного моделирования можно подробнее прочитать в [44].

7. Построение регрессионных зависимостей

Задача построения регрессионных зависимостей одних признаков от других во многом подобна задаче прогнозирования значений. Различие состоит лишь в том, что значения всех информационных признаков считаются известными. Вся совокупность признаков делится на входные и выходные. Считается, что значения выходных признаков можно вычислить с заданной точностью ε_0 по значениям входных признаков. Отдельной задачей является выяснение вопроса о том, насколько правомерно предположение о существование зависимости между входными и выходными признаками. Известен метод (ритуал) анализа таблицы данных, позволяющий оценить оправданность такой гипотезы [35].

Разобъем таблицу на две части (см. рис.36) – слева сгруппируем входные признаки (их набор можно сформировать упомянутыми методами отбора наиболее значимых признаков), а справа – выходные. Зададимся определенной точностью ε. Теперь создадим две информационные модели: одна будет описывать всю таблицу данных с заданной точностью и содержать s_1 карт, вторая будет построена только по набору входных

признаков и описывать с точностью ε левую часть таблицы и содержать s_2 карт. Если число карт в первой и во второй модели будет приблизительно равным, т.е. $s_1 \approx s_2$, то можно считать, что в выходных признаках не содержится дополнительной информации, кроме той, что содержится во входных.

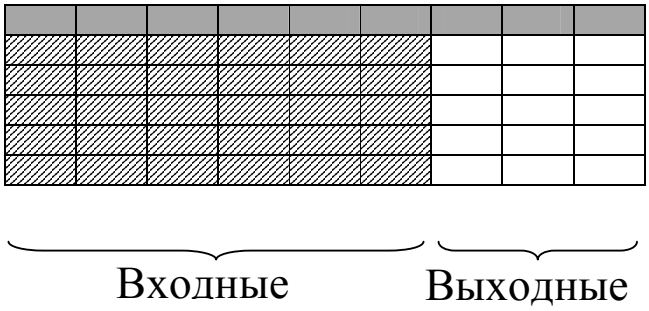


Рис.36. Деление таблицы на входные и выходные признаки. Можно сравнить число факторов, необходимых для описания всей таблицы, и только ее части из входных признаков. Если количество факторов оказалось одинаково – зависимость действительно существует.

Глава 3. Навигация по картам

3.1. Описание программы ViDa Expert 1.0

3.1.1. Внутренняя структура объектов.

Программа ViDa Expert имеет внутреннюю иерархию объектов. Некоторые из них соответствуют тем объектам, с которыми оперирует исследователь на практике, другие объекты являются контейнерами, содержащими и упорядочивающими объекты исследования. Знакомство с внутренней структурой объектов необходимо пользователю для осмысленного использования программы. На рис. 36 изображены сами объекты системы и отношения между ними.

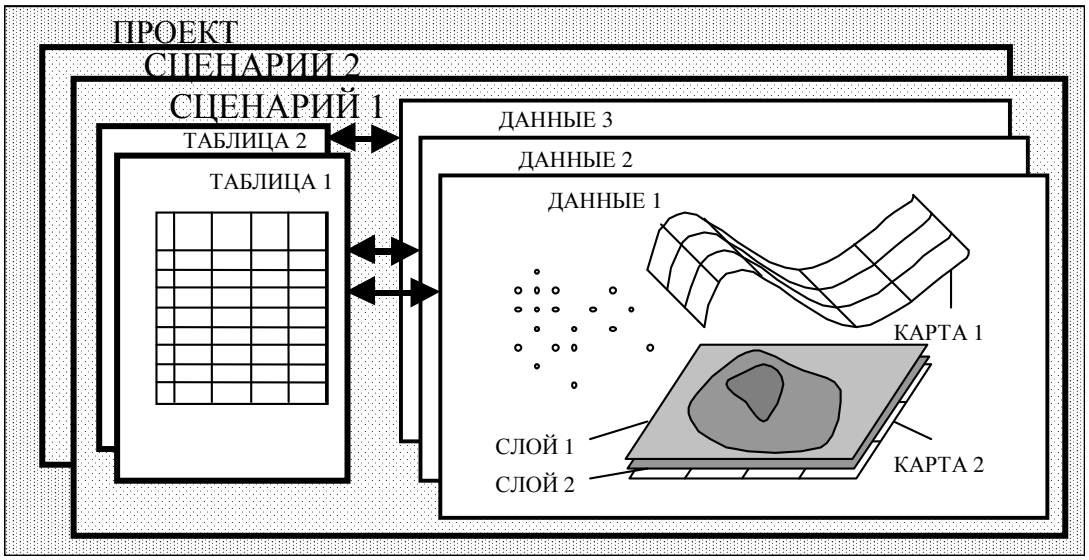


Рис.37. Внутренняя структура объектов программы ViDa Expert

Объектом-контейнером верхнего уровня является ПРОЕКТ, который содержит в себе несколько СЦЕНАРИЕВ. СЦЕНАРИЙ – это совокупность определенным образом настроенных наборов данных и карт. Каждый сценарий как объект-контейнер содержит в себе набор объектов ТАБЛИЦА и набор объектов ДАННЫЕ.

Объект типа ТАБЛИЦА предназначен для хранения исходной табличной информации. В программе ViDa Expert реализовано три способа заполнить таблицу данных – через стандартные файлы баз данных (Paradox и DBase), через текстовые файлы баз данных, и с помощью так

называемого vet-файла (внутренний формат табличных данных системы ViDa).

Загрузив таблицу данных, пользователь на основе объекта ТАБЛИЦА, выбирая необходимые числовые поля и указывая способ их нормировки, создает объект ДАННЫЕ, который содержит числовой массив всех значений выбранных признаков. В дальнейшем объект ДАННЫЕ сохраняет связь с объектом ТАБЛИЦА, на основе которого он был создан. Используя один и тот же объект ТАБЛИЦА, можно создавать различные объекты типа ДАННЫЕ, выбирая разные наборы строк, признаков и способы их нормировки.

На основе объекта ДАННЫЕ пользователь создает различным образом настроенные объекты типа КАРТА, которые в дальнейшем хранятся в нем как в объекте-контейнере. Объекты типа КАРТА содержат всю необходимую информацию о положении узлов сетки в пространстве и о способе ее доопределения до многообразия.

Для визуализации данных на основе объекта КАРТА создается набор объектов типа СЛОЙ. Объект типа слой содержит всю необходимую информацию для отрисовки на экране информационного слоя.

В программе ViDa Expert 1.0 встроено 4 вида слоев, это: Слой точек данных, Слой сетки, Слой раскрасок, Слой объектов.

Каждый СЛОЙ имеет характеристику «Вид». В программе ViDa Expert 1.0 реализовано 4 варианта Видов: это

1) вид на координатные плоскости;
2) вид на плоскость главных компонент;
3) вид во внутренних координатах карты (простая развертка карты);
4) вид во внутренних координатах карты (нелинейная развертка карты).

В 3-ем варианте карта предстает в виде равномерной сетки узлов, точки данных размещены в соответствии с их проекциями на карту. В 4-ом – виде карта изображается в виде криволинейной сетки. Отдельный диалог позволяет настраивать криволинейную развертку несколькими способами: а) так, чтобы расстояние между узлами сетки на плоскости как можно точнее соответствовали расстоянию между узлами в исходном пространстве; б) так, чтобы криволинейная развертка соответствовала определенной раскраске; например, чтобы координатная сетка была более «густой» в светлых областях раскраски и более «разреженной» в темных областях.

Карта может быть различным образом раскрашена. За вариант раскраски отвечает свойство «Тип раскраски» Слоя раскрасок. В программе ViDa Expert 1.0 реализовано 9 вариантов Раскрасок:

1) раскраска по значению выбранного признака;
2) раскраска по двумерной плотности;
3) раскраска по двумерной плотности выделенного подмножества;
4) раскраска по относительной двумерной плотности выделенного подмножества;
5) раскраска по многомерной плотности;
6) раскраска по многомерной плотности выделенного подмножества;
7) раскраска по относительной многомерной плотности выделенного подмножества;
9) раскраска по расстоянию от точки карты до ближайшей точки данных;

3.1.2. Различные варианты работы с программой ViDa Expert

Основные идеологические принципы работы в программе ViDa Expert – это

а) принцип «красной кнопки»;
б) принцип «торчащих хвостиков смысла».

Что это означает?

Принцип «красной кнопки» состоит в том, чтобы пользователь нажимал минимальное количество кнопок (клавиш) для получения «рядового результата», то есть результата, не учитывающего конкретные особенности задачи, но служащего началом для дальнейшей работы. Согласно этому принципу пользователь должен иметь возможность получить результат, не владея всеми тонкостями методов настройки. Иными словами, все уже должно быть максимально настроено для получения результата, который бы как-то удовлетворил пользователя, не желающего или не имеющего времени вникать в детали работы программы. Желательно еще, чтобы этот результат был в некотором отношении «неплохим» (пусть не оптимальным).

Принцип «хвостиков смысла» заключается в том, чтобы, тем не менее, пользователь видел, что за «красной кнопкой» стоит целый массив разнообразных настроек, «рычажков», с которыми он может экспериментировать и видоизменять результат. Пользователь должен

ощущать, что он совершает не бездумные действия и при желании может добиться оптимального результата.

3.1.3. Некоторые типовые задачи

Опишем типовые задачи, которые пользователь может решать с помощью программы ViDa Expert.

Создание задачника и линейный анализ данных

1. Открыть новый проект (пункт *Новый* в меню *Проект*).
2. Добавить новый сценарий (пункт *Добавить* в меню *Сценарий*)
3. Добавить новую таблицу (пункт *Добавить таблицу* в меню *Сценарий*)
4. Отметить в таблице поля-признаки для анализа и выбрать способ нормировки.
   4.1. При желании пользователь может с помощью диалога «Выбор объектов из таблицы» выбрать отдельные строки-объекты из таблицы и указать цвета раскраски, с помощью которых будет задаваться изначальное деление точек на классы.
5. Добавить новый набор данных (пункт *Добавить задачник* в меню *Данные*)
6. С помощью диалога «Линейная статистика» провести простейший анализ данных.
7. При необходимости сохранить набор данных (пункт *Сохранить данные* в меню *Данные*). В файле *ved* сохраняются выбранные поля и варианты нормировки. В дальнейшем файл *ved* может быть открыт в самом начале работы программы (Пункт *Загрузить данные* в меню *Проект*). Таблица автоматически сохраняется в одноименном файле с расширением *vet*.

Автоматическое создание карты и простая визуализация

1. Загрузить данные или создать задачник.
2. Создать карту с помощью кнопки “*See It!*”.
3. Далее данные и карту можно рассматривать в разных пространствах:
   3.1. Трехмерные линейные подпространства, натянутые на отдельные координатные оси (Вид *На плоскость координат*).
   3.2. Трехмерные линейные подпространства, натянутые на главные компоненты (Вид *На главные компоненты*).

3.3. Двумерное пространство карты (если карта двумерная, вид *Во внутренних координатах*).

3.4. Трехмерное пространство карты (если карта трехмерная, вид *Во внутренних координатах*).

4. Карту можно раскрашивать, выбирая раскраски в окне «Раскраска».
5. Точки во всех видах автоматически снабжаются всплывающей аннотацией. Текст аннотации задается полем таблицы, задаваемом в списке «Поле-метка» в панели *Объекты*.

Кластерный анализ с визуальным контролем

1. Загрузить данные или создать задачник.
2. Создать карту с помощью кнопки "*See It!*".
3. В диалоге "Анализ Данных" провести кластерный анализ, результаты которого синхронно отображаются графически и в таблице. Результат классификации можно запомнить в таблице с помощью кнопки «*Номера в таблицу*».

Аннотирование точек данных

1. Загрузить данные или создать задачник.
2. Создать карту с помощью кнопки "*See It!*".
3. В диалоге "Аннотирование данных" произвести аннотирование данных.

3.2. Применение методов визуализации данных к картографированию экономических таблиц

В качестве примера применения технологии визуализации данных нами была предпринята попытка применить методы визуализации произвольных данных к картографированию таблицы крупнейших российских предприятий, взятой из журнала «Эксперт-200» [50]. Файлы исходных данных были получены с официального сайта журнала http://www.expert.com.

Исходная таблица содержала информацию об экономическом положении двухста крупнейших российских предприятий, ранжированную в поряке убывания валового объема производства продукции. Изначально таблица содержала следующие поля-признаки (часть из них является независимыми признаками, часть рассчитывается по явным формулам):

1) Название предприятия;

2) Регион местонахождения предприятия;
3) Отрасль, к которой относится предприятие;
4) Валовый объем производства продукции в 1998 году;
5) Валовый объем производства продукции в 1997 году;
6) Темпы роста предприятия
7) Валовый объем производства в 1998 году, выраженный в долларовом эквиваленте;
8) Балансовая прибыль предприятия;
9) Прибыль предприятия после налогообложения;
10) Прибыльность предприятия;
11) Число работающих на предприятии;
12) Производительность труда.

Шумским С.А. [49] уже была предпринята попытка визуализации таблицы предприятий, взятой из журнала «Эксперт» за 1997 год. В этой работе были использованы традиционные самоорганизующиеся карты Кохонена и диаграммы Хинтона. Там же было предложено использовать в качестве координат пространства данных отношения некоторых независимых признаков из таблицы. Было предложено четыре таких координаты.

Нами было решено расширить пространство исходных данных еще одним измерением, в результате чего был получен следующий набор независимых признаков:

N	Обозначение признака	Значение
1	LG_VO1998	Логарифм валового объема производства продукции в 1998 году
2	TEMP	Валовый объем производства продукции в 1998 году / Валовый объем производства продукции в 1997 году
3	PROFIT_BAL	Балансовая прибыль предприятия / Валовый объем производства продукции в 1998 году
4	PROFIT_NAL	Прибыль предприятия после налогообложения / Валовый объем производства продукции в 1998 году
5	PRODUCTIV	Прибыль предприятия после налогообложения / Число работающих на предприятии

В результате была составлена таблица из двухсот записей с пятью полями. Часть записей содержала неполную информацию (по отдельным признакам информация отсутствовала).

Данные были предварительно нормированы по формуле $\tilde{x}_i = th\left(\frac{x_i - M}{\sqrt{D}}\right)$,

где $\tilde{x}_i, x_i$, *M, D* – новое, старое значения признака, среднее значение и дисперсия признака соответственно.

Карта, с помощью которой осуществлялась визуализация множества данных, была построена по алгоритму построения упругих карт. Первоначальная сетка содержала 10 узлов по вертикали и 10 по горизонтали. Для нахождения локального минимума функционала применялся метод отжига. Параметры μ и λ медленно (так чтобы при каждом изменении карта успевала перейти в близлежащий локальный минимум) менялись от значений $\mu = 5$, $\lambda = 5$ до $\mu = 0.1$, $\lambda = 0.1$.

После построения упругой карты данные из пространства признаков были спроецированы на карту с помощью процедуры нахождения ближайшей точки карты в случае кусочно-линейной интерполяции между узлами.

В качестве иллюстрации анализа экономических данных ниже приведены раскраски полученной карты по координатным полям, а также слой рассчитанной плотности данных в точках карты. На раскрасках большими точками с номерами выделена группа предприятий, принадлежащих нефтегазовой промышленности. Такое выделение позволяет проанализировать место той или иной отрасли промышленности среди других предприятий.

1) Раскраска по признакам

На рисунке 38а изображено значение признака LG_VO1998 в точках карты. При этом более светлым участкам соответствуют более высокие показатели признака. Самый яркий цвет соответствует первым 10% предприятий с самым большим валовым объемом производства. Для примера кружками с цифрами выделены предприятия нефтегазовой промышленности. Цифрам соответствуют следующие названия предприятий:

1 – ОАО «Газпром»; 2 – Нефтяная компания «ЛУКойл»; 3 – Башкирская топливная компания; 4 – Нефтяная компания «Сургутнефтегаз»; 5 – Тюменская нефтяная компания; 6 – «Татнефть»; 7 – Нефтяная компания «Славнефть»; 8 – Нефтяная компания «Роснефть»;

9 – Оренбургская нефтяная компания «Онако»; 10 - Центральная топливная компания; 11 – Нефтяная компания «КомиТЭК».

Рисунок 38б изображает раскраску по показателю TEMP. Как видно из рисунка 38б, область крупнейших предприятий не пересекается с областью наиболее высоких темпов роста. В правом нижнем углу карты, например, располагаются предприятия пищевой промышленности, цветной металлургии и другие быстро развивающиеся отрасли.

На рисунках 38в, 38г, 38д показаны раскраски по признакам PROFIT_BAL, PROFIT_NAL, PRODUCTIV. Эти раскраски схожи, что указывает на корреляцию последних трех признаков. Вместе с этим различия в раскраске позволяют выделить предприятия, которые выпадают из корреляционной зависимости.

2) Раскраска по плотности данных

На рисунках 38е),38ж),38з) показана раскраска карты по плотности данных, оцененной с помощью какой-либо непараметрической оценки. Существует два способа оценить плотность данных. Во-первых, можно рассматривать двумерное распределение точек на карте. Во-вторых, можно рассчитать плотность точек в исходном n-мерном пространстве, и изображать на карте значения этой плотности в точках расположения карты. На рисунках изображено применение первого способа. Более темным участкам соответствуют более высокие значения плотности.

Рисунок 38е) изображает двумерное распределение общей плотности данных. На рисунке 38ж) – распределение плотности предприятий нефтегазовой промышленности. Рисунок 38з) отражает удобную для оценок относительную плотность предприятий нефтегазовой промышленности (то есть отношение первых двух плотностей).

На рисунке 38и) отражено расстояние от каждой из точек карты до ближайшей точки данных. Более темным участкам соответствуют большие расстояния. Видно, что в целом данные достаточно плотно прилегают к карте, за исключением участка в левом верхнем углу (впрочем, точки данных там отсутствуют и темный цвет указывает на то, что точки в левом верхнем углу карты расположены в многомерном пространстве достаточно далеко от основного массива данных).

Беглый взгляд на рисунки позволяет сделать, например, такие выводы. Предприятия нефтегазовой промышленности являются лидерами по объему валового производства, но темпы роста этой области промышленности невелики по сравнению, например, с пищевой

промышленностью. Предприятия нефтегазовой промышленности распадаются на две группы, которые существенно отличаются по прибыльности производства. В целом, набор таких рисунков могут служить удобным средством анализа для специалистов в макроэкономике.

3.3. Нейроинформатика – наука или фантастика?

Попробуем ответить на этот вопрос с помощью приема картографирования текстовых коллекций на основе представления текстов в виде *частотных словарей*.

По 50-ти текстам, представляющих собой научные статьи, доклады конференций и книги был составлен словарь из 800 наиболее употребляемых слов. Аналогичный словарь объемом 700 слов был составлен для коллекции фантастических произведений различных авторов и жанров. Поскольку в русском языке одно и то же слово может быть представлено в нескольких формах, то для идентификации корня брались лишь первые значащие буквы слова. В словари не включались слова-связки и незначащие слова (местоимения, общеупотребительные слова и др.).

Далее оба словаря были объединены. Поскольку словари оказались частично перекрывающимися, в результирующем словаре оказалось 1375 слов. Слова были пронумерованы в алфавитном порядке.

Для того, чтобы представить текст в виде многомерного вектора ему сопоставлялся набор частот $w_i = {}^{n_i}\!/_{N}$, i = 1…1375, где n_i – число встретившихся форм i-ого слова из словаря, N – общее число слов в тексте. В результате была получена таблица из 1376 столбцов (первый из них содержал название текста, остальные – частоты), в которую были занесены частотные словари 113 текстов, в которые вошли фантастические произведения С.Лема, Р.Желязны, С.Кинга, А.Азимова, А.Кларка, Р.Брэдбери, К.Булычева и др., книга А.Н.Горбаня «Демон Дарвина», книга Е.М.Миркеса «Нейрокомпьютер. Проект стандарта», некоторые научные статьи красноярской группы исследователей «Нейрокомп», тезисы докладов, представленных на конференции «Нейроинформатика и ее приложения - 2000», проводившиейся в г.Красноярске в октябре 2000 года, главы этой книги и некоторые другие тексты.

На рис.39а) приведен вид полученного многомерного облака точек на плоскость первых двух главных компонент. Как видно, частотные словари фантастических и научных текстов хорошо разделяются вдоль первой главной компоненты. Рассмотрим те признаки, которые оказались наиболее значимыми для такого разделения. Эти признаки-словоформы

вошли в вектор первой главной компоненты с наибольшими по абсолютной величине весами. Первый десяток таких словоформ показан в табл.1. Наоборот, те признаки, которые имеют близкие к нулю веса, оказались незначимыми для разделения.

Для более подробного анализа текстов по набору данных была

построена карта. Данные спроецированные на карту, показаны на

рисунке 39б). Видно, что тексты в той и другой группе распадаются на

подгруппы. Анализ названий текстов, входящих в подгруппы позволяет

произвести классификацию текстов следующим образом:

1. Фантастика гуманитарной направленности;
2. Фантастика технической направленности;
3. Биологические и медицинские приложения нейросетей;
4,5. Технические приемы по созданию нейросетей;
6. Моделирование данных (в эту подгруппу входит и эта книга – объекты «Глава 1», «Глава 2», «Глава 3»).

Эта классификация не охватывает таких выделяющихся текстов, какими

являются книги А.Н. Горбаня «Демон Дарвина», С.Лема «Сумма

технологий» и некоторых других.

На примере картографирования коллекций текстов рассмотрим возможности автоматического *аннотирования* точек данных. На рис.39б) показан самый простой способ аннотирования данных – на точки повешены названия тех признаков, значения которых оказались для данного объекта максимальными. В нашем случае это означает, что соответствующие словоформы оказались в тексте наиболее часто встречающимися.

Другой способ аннотирования состоит в выделении для данного объекта тех признаков, значения которых наименее вероятны по ансамблю объектов. Поясним, что это означает на нашем примере. Для каждого признака-слова может быть построена гистограмма распределения значений по всем объектам. В некоторых интервалах гистограммы

окажется большое количество объектов, в некоторых – малое. Выберем объект (или точку пространства признаков) для аннотирования. Каждому из конкретных значений признаков у выбранного объекта можно сопоставить вероятность его появления (взяв ее из построенной по всем объектам гистограммы). Если вероятность окажется высока, то это значение признака является в некотором роде «типичным» для данной системы объектов. Если вероятность мала – значит такие значения признак принимает лишь на небольшом количестве объектов, то есть выбранный объект по данному признаку является «нетипичным».

Табл.1. Слова-признаки, самые значимые и самые малозначимые для разделения текстов на научные (нейроинформатика) и фантастические

Слова, обладающие отрицательными весами		Слова, обладающие положительными весами		Слова с близкими к нулю весами	
Слово	Вес	слово	вес	слово	вес
верн	-0.09101	использ	0.076008	идеал	0.00129
чувств	-0.08864	основ	0.073729	прост	0.000917
добр	-0.08147	задач	0.068003	резк	0.000833
земл	-0.07893	данн	0.065691	парадокс	0.000689
люд	-0.07666	нейро	0.064884	высказ	0.000601
странн	-0.07657	определ	0.063517	кислот	0.000266
холод	-0.0736	функц	0.057877	ремонт	0.00017
трудн	-0.07148	анали	0.057309	констр	0.000062
страх	-0.06715	модел	0.056592	окрестност	-0.00004
осторожн	-0.0666	параметр	0.055749	вопрос	-0.00019
сомнен	-0.0665	сет	0.053742	груб	-0.00043
мысл	-0.06584	резуль	0.053426	скобк	-0.00052
чуж	-0.06563	обуч	0.052626	сформулир	-0.00079
зло	-0.06536	алгоритм	0.052238	мишен	-0.00083
ужас	-0.0644	решен	0.051107	вражд	-0.00102
тревог	-0.06423	выбор	0.050932	свидет	-0.00112

В случае текстовых коллекций маловероятный признак – это то слово, которое отличает данный текст от остальных. Так, например, слово «данные» может стабильно часто употребляться во всех текстах по нейроинформатике. Поэтому, информация о том, что в тексте часто употребляются слова «анализ», «данные» никак не выделяет его среди остальных текстов. Информация же о том, что слово «лимфоцит» оказалось маловероятным в данном тексте для данного ансамбля текстов, определенным образом его характеризует. Наоборот, если маловероятным оказалось слово «данные», то и это означает, что текст «выпадает» из общей направленности собрания текстов по нейроинформатике.

В случае если коллекция текстов исходно разнородна, как в нашем примере – тогда тоже имеет смысл выделять маловероятные признаки, но вероятности лучше рассчитывать по объектам того класса, в который входит объект.

На рис.40а) представлен способ аннотирования текстов по маловероятным словам.

Источник иной информации о исследуемой системе текстов – визуализация транспонированной задачи, описанная в разделе 2.8 (см. рис.40б). В этом случае роль объектов играют слова из частотного словаря, а признаков – тексты. Если на плоскости первых главных компонент два слова оказались рядом, то это позволяет предположить, что на данной совокупности текстов они коррелируют, то есть примерно одинаково часто встречаются в одних текстах и одинаково редко – в других. Если слова оказались сильно разнесены (на противоположных «полюсах» карты) это указывает на обратную корреляцию – если одно слово встречается часто в каком-либо тексте, то другое в этом тексте встречается, скорее всего, редко.

Аннотирование по маловероятным признакам применимо и здесь. В данном случае маловероятный признак – текст – выделяется для аннотируемого слова. Это означает, что среди выбранных текстов маловероятно встретить заданное слово именно в указанном тексте.

Может показаться, что поставленный в заголовке вопрос не имеет большого «научного» и практического значения. Однако, на примере картографирования текстовых коллекций демонстрируются методы, применимые для задач более серьезных наук. Пример использования частотных словарей – *анализ генетических последовательностей*. В этом случае текст – последовательность «букв» генетического алфавита A, C, G, T. Последовательные буквы можно объединять в слова разной длины. Таким образом, генетический текст можно представлять в виде частотного словаря. Генетическому коду каждой отдельной особи сопоставляется свой частотный словарь. Сравнение и визуализация генетических частотных словарей внутри выборки особей одного вида, а также сравнение генетических частотных словарей особей разных видов может служить источником ценной информации для специалистов в этой области.

Приведем пример. По 1800 реальным генетическим последовательностям бактерий, принадлежащих семействам Proteobacteria, Firmicutes, Acidobacterium, Aerobic bacillus, Cyanobacteria была составлена таблица частот встречающихся в тексте слов длины 1, 2 и 3 (синглеты, дуплеты и триплеты). На рис.41а) показан вариант картографирования таблицы. На карте точками разной формы выделены три отдельных рода бактерий (Proteobacteria a-sd, Proteobacteria b-sd, Firmicutes Actinomycetes).

Видно, что биологическая классификация в многомерном пространстве частотных словарей задает достаточно компактные и отделенные друг от друга группировки объектов.

3.4. Визуализируем выборы

Аннотирование полезно не только для частотных словарей. На рис.42а) показана карта выборов американских президентов, построенная на основе известной таблицы [12,56,68]. В таблице содержатся ответы на 12 вопросов, расшифровка обозначений которых приведена на рисунке. На карте явно выделяются выборы 1880 года, попадающие в область, где преобладают объекты противоположного класса. В своем классе (победа правящей партии) для выборов 1880 года оказались крайне маловероятны значения признаков CONC (*Была серьезная конкуренция при выдвижении от правящей партии?*) и PREZ (*Кандидат от правящей партии был президентом в год выборов?*). Действительно, среди всех побед правящей партии признак CONC был равен единице только для выборов 1880 года. С другой стороны, этот признак оказывается весьма значимым для решения вопроса о победе на выборах.

Чтобы подтвердить последнее утверждение, рассмотрим транспонированную задачу (рис. 42б). В группу визуализируемых признаков включено поле «Ответ», в котором содержится результат выборов. Видно, что признак, наиболее связанный с результатом выборов – CONC. С другой стороны, признаком, наиболее удаленным от «Ответ» является признак PREZ, что может указывать на обратную корреляцию. Признаки DEPR, O_HERO, MIST, THIRD, CHANGES, WAVE образуют группу взаимосвязанных признаков.

3.5. Осложнения инфаркта-миокарда[3]

Инфаркт миокарда – распространенное и грозное заболевание. Бурное распространение этого заболевания за последние полвека сделало его одной из наиболее острых проблем современной медицины.

Заболеваемость инфарктом миокарда остается высокой во всех странах. Особенно это касается городского населения высокоразвитых стран, испытывающего стремительный ритм современной жизни и подвергающегося хроническому воздействию стрессовых факторов, нерегулярного и не всегда сбалансированного питания. В США ежегодно около 1,5 миллионов человек заболевают инфарктом миокарда.

---

[3] Описание ситуации с осложнениями инфаркта миокарда взято из [40,54,55].

Несмотря на то, что внедрение современных лечебно-профилактических мероприятий несколько снизило смертность от инфарктов, она продолжает оставаться довольно высокой. Около 15-20% больных острым инфарктом миокарда погибают на догоспитальном этапе, еще 15% в больнице, т.е. общая летальность при остром инфаркте миокарда 30-35%.

Течение заболевания у пациентов с инфарктом миокарда различно. Инфаркт миокарда может протекать без осложнений или с осложнениями не ухудшающими долгосрочный прогноз. В тоже время около половины пациентов в острый и подострый периоды имеют осложнения, приводящие к ухудшению течения заболевания и даже летальному исходу. Предвидеть развитие этих осложнений может не всегда даже опытный специалист.

Для решения задачи прогнозирования осложнений инфаркта миокарда с целью своевременного проведения необходимых профилактических мероприятий сотрудниками кафедры внутренних болезней № 1 Красноярской государственной медицинской академии была собрана информация о течении заболевания у 1700 больных инфарктом миокарда, проходивших лечение в 1989-1995 годах в Кардиологическом центре городской больницы № 20 г.Красноярска. Информация получена из историй болезни пациентов и сконцентрирована в 128 полях электронной таблицы. База данных содержит сведения о данных анамнеза каждого больного, клинике настоящего инфаркта миокарда, электрокардиографических, лабораторных показателях, лекарственной терапии и особенностях течения заболевания в первые дни инфаркта миокарда.

В результате получилась большая таблица, анализ информации в которой имеет, с одной стороны, практическое значение, с другой – данные в ней имеют весьма сложную структуру. По утверждениям некоторых специалистов в области нейроинформатики: «Красноярская таблица по осложнениям инфаркта миокарда содержит почти все известные сложности, с которыми может столкнуться исследователь при анализе реальных данных. Любой метод анализа, претендующий на практическое применение, должен быть апробирован и на этой таблице.».

На рис.42 приведен пример визуального представления таблицы по осложнениям инфаркта миокарда. На рис. ??? а) приведена оценка плотности распределения объектов. Поскольку в таблице содержится большое количество признаков, необходимо провести предварительный анализ признаков на значимость для того, чтобы предоставить пользователю наиболее значимые признаки. Был проведен самый простой анализ признаков на значимость с помощью первой главной компоненты, и на рис.42б)-42д) показаны раскраски по нескольким признакам, которые

имеют наибольшие по абсолютной величине веса в векторе главных компонент. На рис.42е) большими треугольниками обозначены летальные исходы заболевания.

Подробный анализ таблицы по осложнениям инфаркта миокарда требует отдельного и весьма обширного исследования. Информационные раскраски и применение методов визуализации могут играть в этом исследовании вспомогательную роль иллюстративного материала к другим методам, а также имеют самостоятельную ценность.

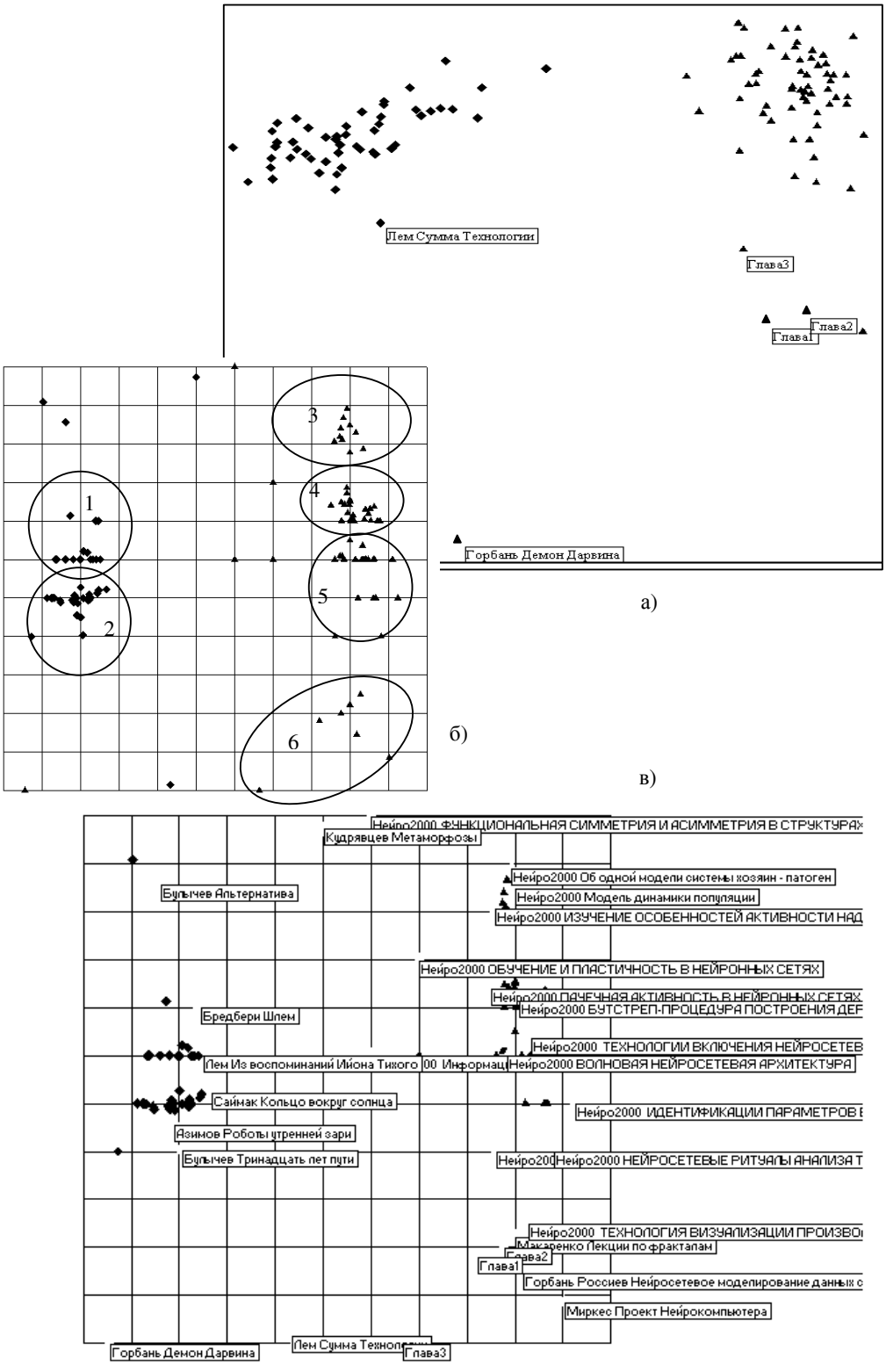


Рис. 39. Визуализация собрания из 116 текстов. Ромбами обозначены фантастические тексты, треугольниками – «научные».

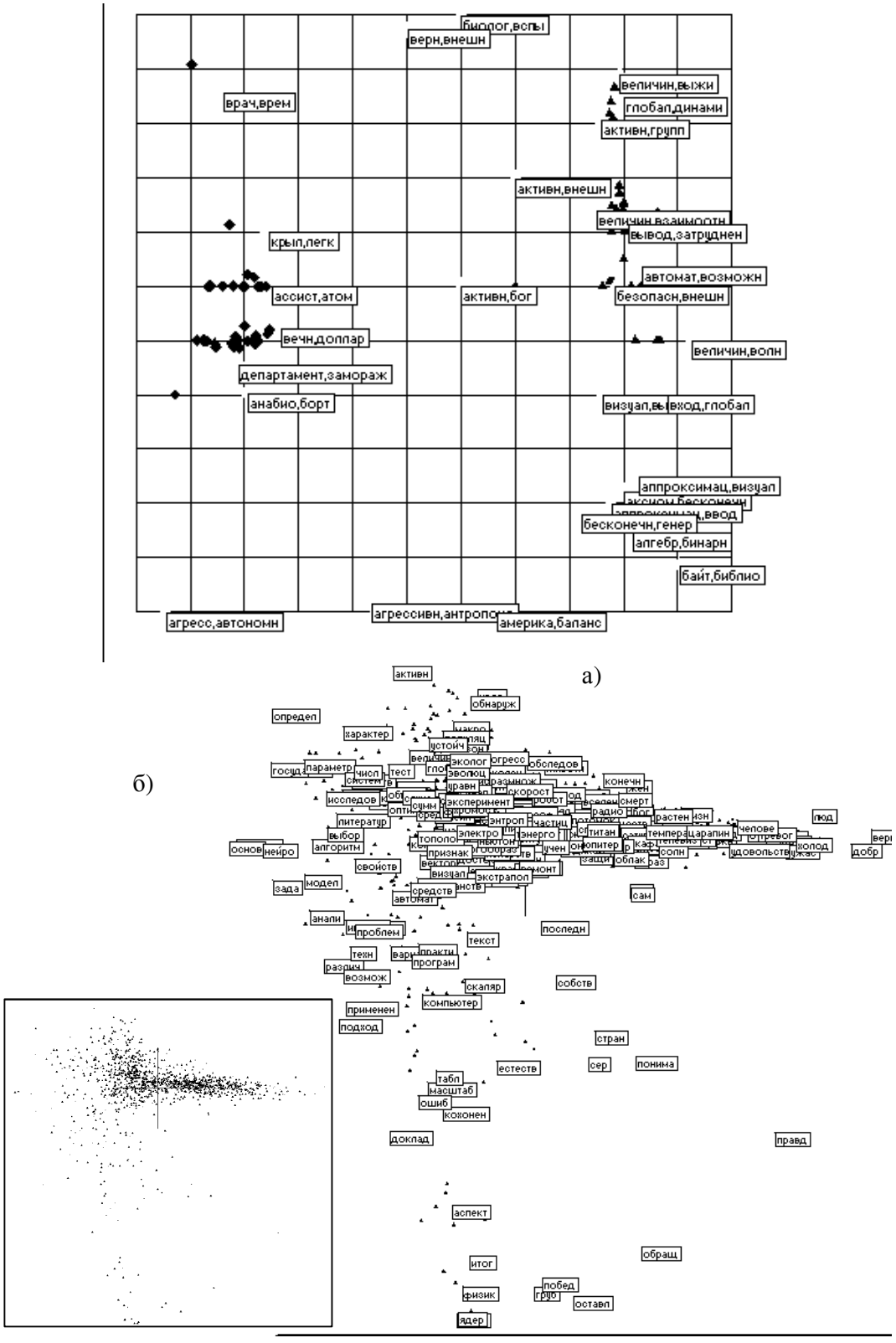
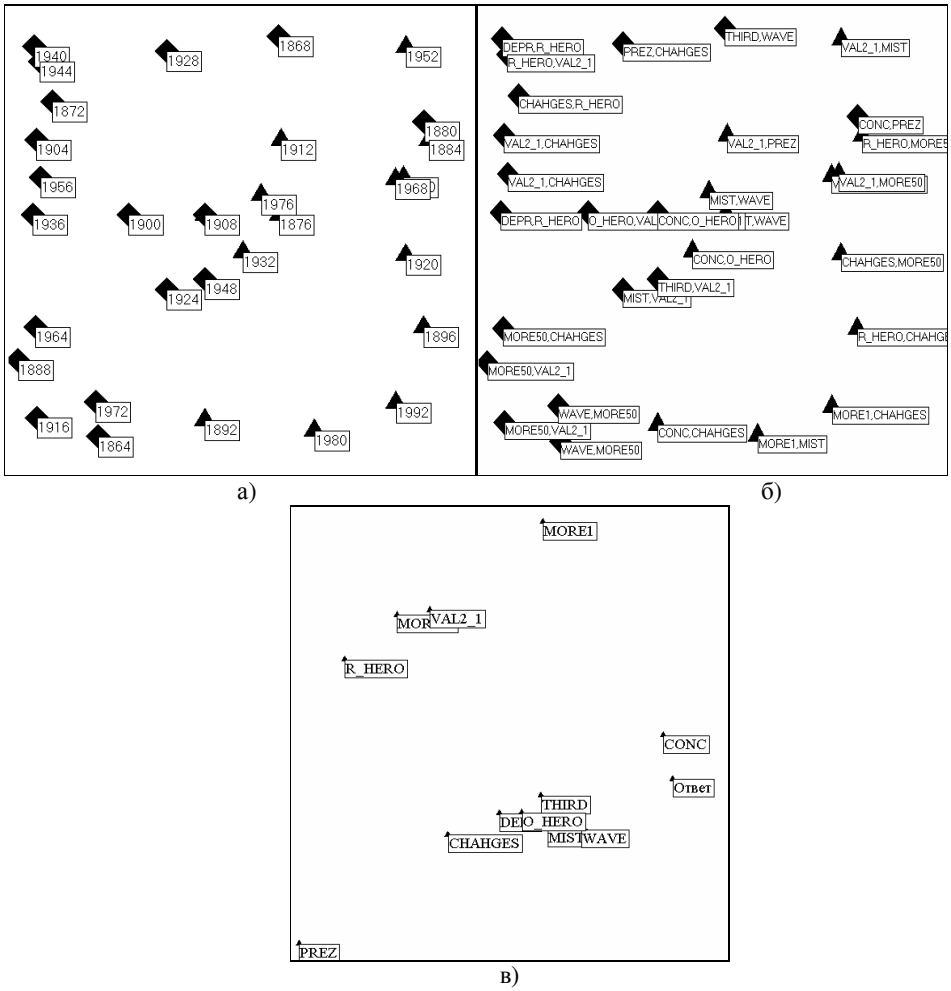


Рис. 40. а) аннотирование текстов по маловероятным словам;
б) визуализация транспонированной задачи.



а) б)

в)

Рис. 41. Карта выборов американских президентов. Ромбами отмечены выборы, на которых победу одержала правящая партия, треугольниками – победы оппозиции.

а) аннотация по году выборов;

б) аннотация по двум самым маловероятным признакам в своем классе;

в) картографирование транспонированной задачи (близкие признаки – коррелированы);

Расшифровка названий признаков:

MORE1	Правящая партия была у власти более одного года?
MORE50	Правящая партия получила больше 50% на прошлых выборах?
THIRD	**В год выборов была активна третья партия?**
CONC	Была серьезная конкуренция при выдвижении от правящей партии?
PREZ	Кандидат от правящей партии был президентом в год выборов?)
DEPR	Был ли год выборов временем спада или депрессии?
VAL2_1	Был ли рост среднего национального валового продукта на душу населения >2,1%
CHANGES	Произвел ли правящий президент существенные изменения в политике?
WAVE	Во время правления были существенные социальные волнения?
MIST	Администрация правящей партии виновна в серьезной ошибке или скандале?
R_HERO	Кандидат правящей партии – национальный герой?
O_HERO	Кандидат оппозиционной партии – национальный герой?

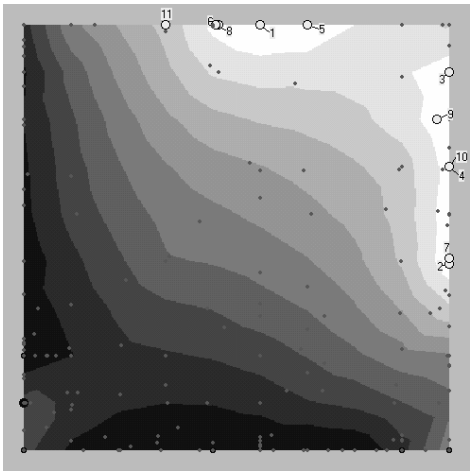


Рис. 38а

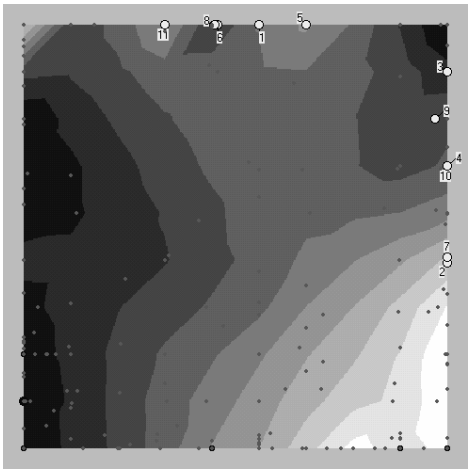


Рис. 38б

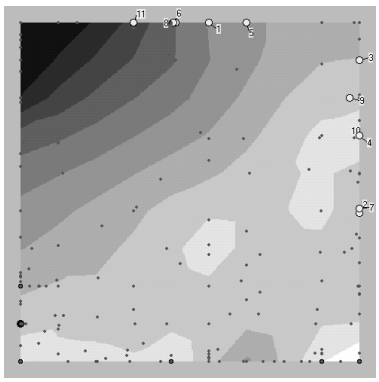


Рис. 38в

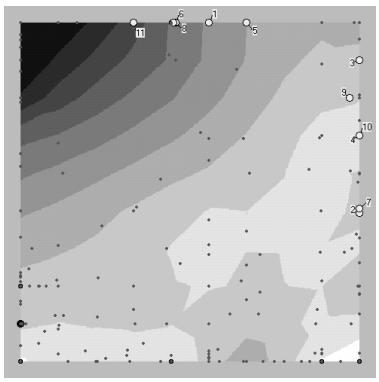


Рис. 38г

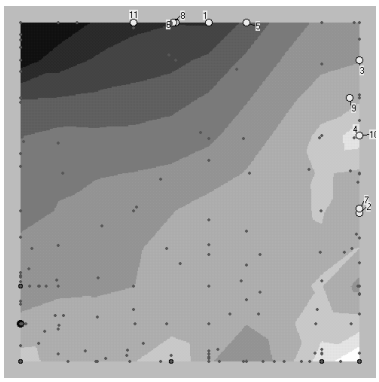


Рис. 38д

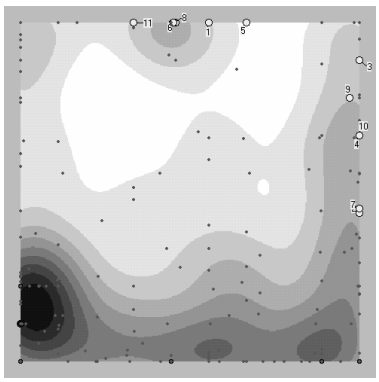


Рис. 38е

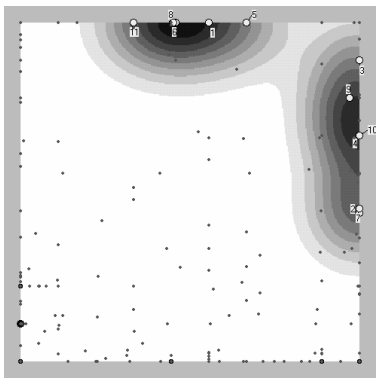


Рис. 38ж

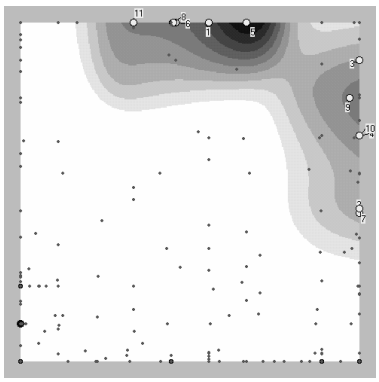


Рис. 38з

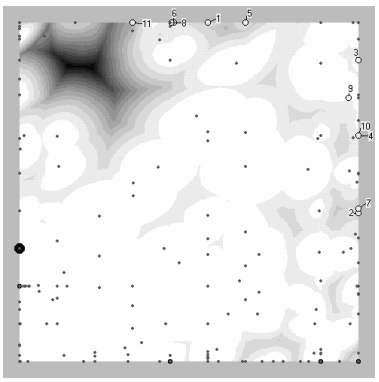


Рис. 38и

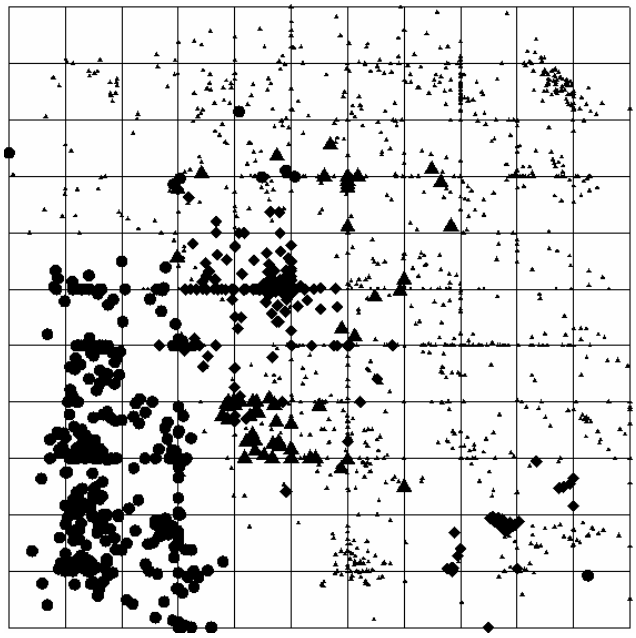


Рис.41а. Картографирование базы частотных словарей генетических последовательностей. Большими точками трех разных форм отмечены три отдельных рода бактерий. Из рисунка видно, что в пространстве частотных словарей биологическая классификация задает более или менее компактные группировки объектов.

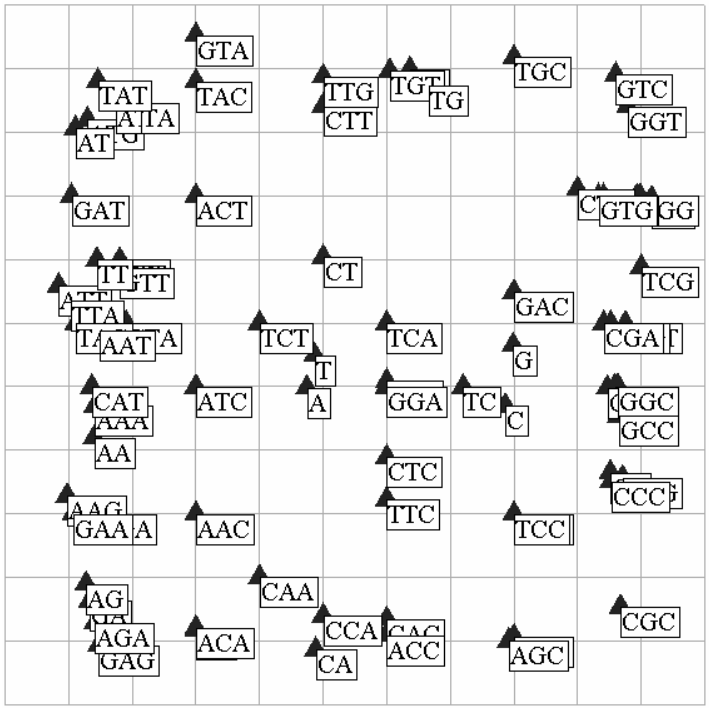
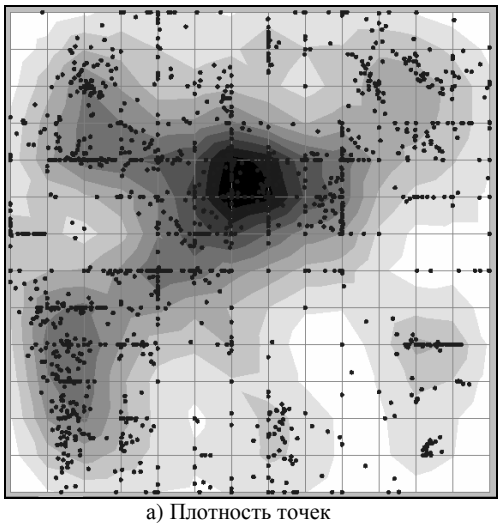
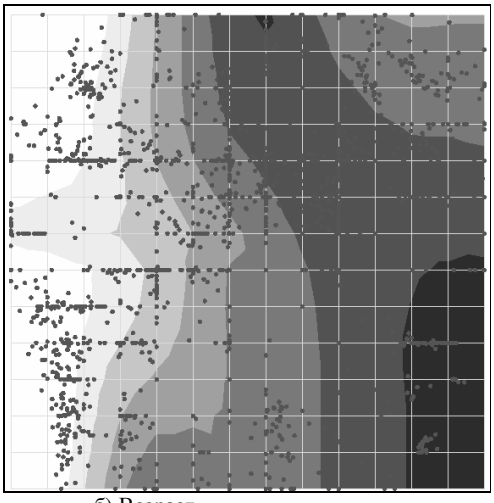


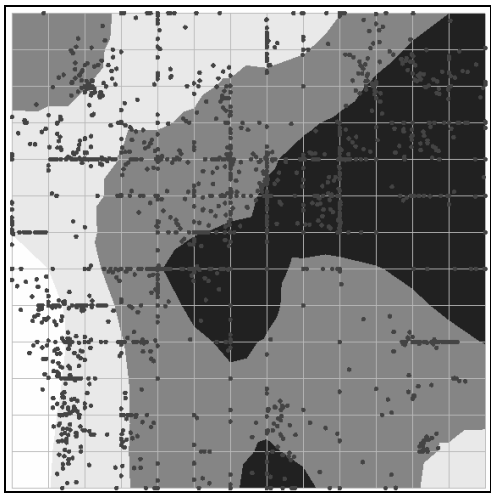
Рис.41б. Визуализация транспонированной задачи для таблицы генетических частотных словарей. Близкие признаки отвечают взаимосвязанным словам в генетическом тексте.



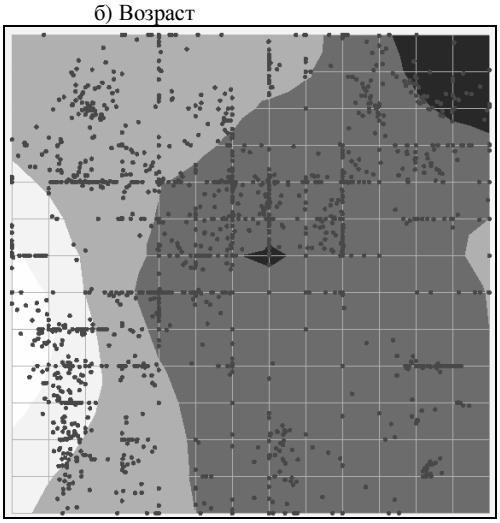
а) Плотность точек



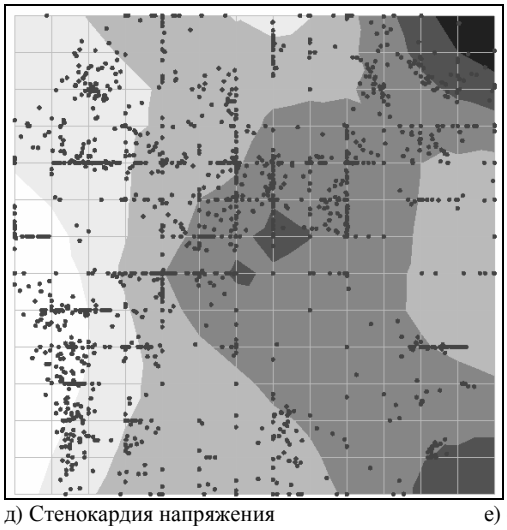
б) Возраст



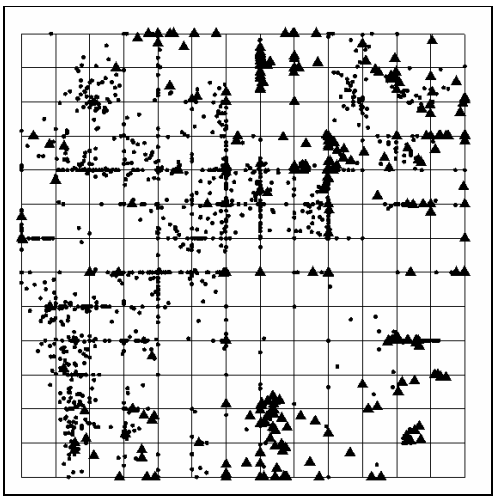
в) Количество инфарктов в анамнезе



г) Функциональный класс стенокардии в последний год



д) Стенокардия напряжения в анамнезе



е) случаи летального исхода (большие треугольники)

Рис. 42. Картографирование базы данных осложнений инфаркта миокарда

ЛИТЕРАТУРА

1. Айвазян С.А., Бежаева З.И., Староверов О.В. Классификация многомерных наблюдений. – М.: Статистика, 1974. – 240 с.
2. Айвазян С.А., Бухштабер В.М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности.- М.: Финансы и статистика, 1989.-607 с.
3. Айвазян С.А., Енюков И.С., Мешалкин Л. Д. Прикладная статистика. Основы моделирования и первичная обработка данных. - М.: Финансы и статистика, 1983.-471 с.
4. Айвазян С.А., Енюков И.С., Мешалкин Л. Д. Прикладная статистика. Статистическое оценивание зависимостей.- М.: Финансы и статистика, 1985.- 484 с.
5. Айзенберг Л.А. Формулы Карлемана в комплексном анализе. Первые приложения. - Новосибирск: Наука, 1990.
6. Андерсон Т. Введение в многомерный статистический анализ.- М.: Физматгиз, 1963.-500 с.
7. Гареев А.Ф. Применение вероятностной нейронной сети для автоматического рубрицирования текстов // Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1999. Часть 3. С.71-79.
8. Горбань А.Н., Хлебопрос Р.Г. Демон Дарвина. М.: Наука (Физ-Мат-Лит), 1988. **http://ddarwin.narod.ru**
9. Горбань А.Н. Обучение нейронных сетей. М.: изд. СССР-США СП "ПараGraph", 1990. 160 с.
10. Горбань А.Н., Зиновьев А.Ю., Питенко А.А. Визуализация данных методом упругих карт // Информационные технологии, изд-во "Машиностроение". - М. - 2000. № 6, - С.26-35.
11. Горбань А.Н., Макаров С.В., Россиев А.А. Нейронный конвейер для восстановления пробелов в таблицах и построения регрессии по малым выборкам с неполными данными // Математика. Компьютер. Образование. Вып. 5. Часть II. Сборник научных трудов / Под ред. Г.Ю. Ризниченко. М.: Изд-во Прогресс-Традиция, 1998. С. 27-32.
12. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. Новосибирск: Наука (Сиб. отделение), 1996. 276 с.
13. Горбань А.Н., Россиев А.А. Итерационный метод главных кривых для данных с пробелами // Проблемы нейрокибернетики: Труды 12 Международной конференции по нейрокибернетике. Ростов-на-Дону: Издательство СКНЦ ВШ, 1999. С. 198-201.
14. Дейвисон М. Многомерное шкалирование: Методы наглядного представления данных.- М.: Финансы и статистика, 1988.
15. Демиденко Е.З. Линейная и нелинейная регрессия. – М.: Финансы и статистика, 1973. – 302 с.
16. Диянкова С.А., Терехов С.А., Мухамадиева Т.А., Квичанский А.В. Java-апплет SOMA для визуализации многомерной информации на

нейросетевых картах Кохонена // Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1999. Часть 3. С.79-83.
17. Дорофеюк А.А. Алгоритмы автоматической классификации: Обзор // Автоматика и телемеханика. – 1971. – № 12. – С. 78-113.
18. Дуда Р., Харт П. Распознавание образов и анализ сцен.- М.: Мир, 1976.- 511 с.
19. Дюк В.А. Компьютерная психодиагностика. – СПб., издательство «Братство», 1994.-364 с.
20. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его приложения в экономике и бизнесе. М.: Изд-во МИФИ, 1998.
21. Елисеева И. И., Рукавишников В. О. Группировка, корреляция, распознавание образов (Статистические методы классификации и измерения связи).- М.: Статистика, 1977.-144 с.
22. Жигирев Н.Н., Корж В.В., Оныкий Б.Н. Использование ассиметрии частотных свойств информационных признаков для построения автоматизированных систем классификации текстовых документов // Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1999. Часть 3. С.83-91.
23. Зиновьев А.Ю., Питенко А.А. Визуализация данных методом упругих карт // Радіоелектроніка. Інформатика. Управління, Запорожье. 2000, № 1, С.76-85.
24. Зиновьев А.Ю., Питенко А.А. Визуализация произвольных данных методом упругих карт // Материалы конференции молодых ученых Красноярского научного центра СО РАН, апрель 2000г. - Красноярск: КНЦ СО РАН. - 2000. - С.18-20.
25. Зиновьев А.Ю., Питенко А.А. Картографирование произвольных данных. // "Студент и научно-технический прогресс": Информационные технологии. Материалы XXXVIII международной научной студенческой конференции.- Новосибирск: НГУ.- 2000. - С.38.
26. Зиновьев А.Ю., Питенко А.А. Система визуализации произвольных данных. // 2-я Всероссийская научно-техническая конференция "Нейроинформатика-2000". Ч.1. М.: МИФИ.- 2000. С.75-80.
27. Зиновьев А.Ю., Питенко А.А., Россиев А.А. Проектирование многомерных данных на двумерную сетку. // 2-я Всероссийская научно-техническая конференция “Нейроинформатика-2000”. Ч.1. М.: МИФИ.– 2000. С.80-88.
28. Кендалл М. Методы ранговой корреляции.-М.: Статистика, 1974.
29. Кендалл М., Стюарт А. Статистические выводы и связи.- М.: Наука, 1973.-900 с.
30. Классификация и кластер // под. ред. Дж. Вэн Райэин.-М.: Мир, 1980.- 390 с.
31. Колмогоров А. Н. Три подхода к определению понятия "количество информацин"//Проблемы передачи информации/под ред. Яглома П. С., 1965, т. 1, вып. 1.

32. Кузнецов А. С. Методы поиска оптимальных групп признаков при статистическом распознавании образов. - Л. : ВИКИ им. А. Ф. Можайского, 1982.- с. 14-23.
33. Лбов Г. С. Выбор эффективной системы зависимых признаков // Труды Сиб. отд. АН СССР: Вычислительные системы. - Новосибирск, 1965, вып. 19.- с. 87-101.
34. Лбов Г.С. Методы обработки разнотипных экспериментальных данных.-Новосибирск: Наука, 1981.-157 с.
35. Миркес Е.М. Нейрокомпьютер. Проект стандарта. - Новосибирск: Наука,1998 .-188 с.
36. Миркин Б. Г. Анализ качественных признаков и структур. - М.: Статистика, 1980.-319 с.
37. Нейроинформатика // А.Н.Горбань, В.Л.Дунин-Барковский, А.Н.Кирдин, Е.М.Миркес, А.Ю.Новоходько, Д.А.Россиев, С.А.Терехов, М.Ю.Сенашова, В.Г.Царегородцев. - Новосибирск: Наука, Сибирская издательская фирма РАН, 1998.-296 с.
38. Питенко А.А. Картографирование всех и всяческих данных. // ИНТЕРКАРТО-5 : Доклады международной конференции, часть 1. – Якутск: ЯГУ, 1999. С.71–78
39. Питенко А.А. Нейросети для геоинформационных систем // Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1998. Часть 3. С.65-69.
40. Россиев Д.А., Головенкин С.Е., Шульман В.А., Матюшин Г.В. Прогнозирование осложнений инфаркта миокарда нейронными сетями. // Нейроинформатика и ее приложения. Материалы III Всероссийского рабочего семинара. 6-8 октября 1995 г. Красноярск.- 1995.- С.128-166.
41. Россиев А.А. Моделирование данных при помощи кривых для восстановления пробелов в таблицах // Методы нейроинформатики / Под ред. А.Н.Горбаня. Красноярск: Изд-во КГТУ, 1998. С. 6-22.
42. Справочник по прикладной статистике. В 2-х т. Т. 2 // под ред. Ллойда Э., Ледермана У., Айвазяна С.А., Тюрина Ю.Н.- М.: Финансы и статистика, 1990.-526 с.
43. Терехина А.Ю. Анализ данных методами многомерного шкалирования.- М.: Наука, 1986.-168 с.
44. Терехов С.А. Нейросетевые информационные модели сложных инженерных систем. Нейроинформатика. С. 101-136. Новосибирск. Наука. 1998.
45. Терехов С.А., Квичанский А.В., Воленко Е.В., Щукин Н.В. Нейросетевая навигация в архивах трудов научно-технических конференций // Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1998. Часть 3. С.122-127.
46. Тьюки Дж. Анализ результатов наблюдений. Разведочный анализ.- М.: Мир, 1981.-693 с.
47. Харман Г. Современный факторный анализ.-М.: Статистика, 1972.- 486 с.

48. Царегородцев В.Г. Производство полуэмпирических знаний из таблиц данных с помощью обучаемых искусственных нейронных сетей // Методы нейроинформатики: - Красноярск. Издательство КГТУ, 1998.
49. Шумский С.А., Кочкин А.Н. Самоорганизующиеся карты финансовых индикаторов 200 крупнейших российских предприятий. Материалы Всероссийской научной конференции «Нейроинформатика-99». Москва, 1999. Часть 3. С.122-127.
50. "Эксперт-200": ежегодный рейтинг крупнейших компаний России // Журнал "Эксперт". 1999. №36.
51. A.Rauber. LabelSOM: On the labeling of self-organizing maps // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1999.
52. B.Back, K.Sere, H.Vanharanta. Analyzing Financial Performance with Self-Organized Maps. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.
53. Gorban A.N., Rossiev A.A. Wunch II D.C. Neural Network Modelling of Data with Gaps: Method of Principal Curves, Carleman's Formula and Other// Радіоелектроніка. Інформатика. Управління, Запорожье. 2000, № 1, С. 47-55.
54. Gorban A.N., Rossiev D.A., Butakova E.V., Gilev S.E., Golovenkin S.E., Dogadin S.A., Dorrer M.A., Kochenov D.A., Kopytov A.G., Maslennikova E.V., Matyushin G.V., Mirkes Ye.M., Nazarov B.V., Nozdrachev K.G., Savchenko A.A., Smirnova S.V., Shulman V.A., Zenkin V.I. Medical, psychological and physiological applications of MultiNeuron neural simulator // The Second International Symposium on Neuroinformatics and Neurocomputers, Rostov-on-Don, Russia, September 20-23, 1995.- Rostov-on-Don, 1995.- P.7-14.
55. Gorban A.N., Rossiev D.A., Gilev S.E., Dorrer M.G., Kochenov D.A., Mirkes Ye.M., Golovenkin S.E., Dogadin S.A., Nozdrachev K.G., Matyushin G.V., Shulman V.A., Savchenko A.A. Medical and physiological applications of MultiNeuron neural simulator // Proc. WCNN 95. (World Congress on Neural Networks 95). - Washington, DC, July 1995.
56. Gorban A.N., Waxman C. Neural Networks for Political Forecast. Proceedings of the WCNN'95 (World Congress on NeuralNetworks'95, Washington DC, July 1995), PP.176- 178.
57. H.Tokutaka, K.Yoshihara, K.Fujimura, K.Iwanoto, T.Watanabe, S.Kisdia. Applications of Self-Organized Map (SOM) to the Composition Determination of Chemical Products. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.
58. Hastie T., Stuetzle W. Principal curves. Journal of the American Statistical Association. 1988, Jun. V. 84, No. 406. PP.502-516.
59. J.Chang, J.Jerry Lin, T.Chuieh. Color Image Vector Quantization Using Binary Tree Structured Self-Organizing Feature Maps // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.

60. J.Goppert. Regularized SOM-Training: A Solution to the Topology-Approximation Dilemma? // Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 38-44
61. J.Laaksonen, M.Koskela, E.Oja. PicSOM: Self-organizing maps for content-based image retrieval. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1999.
62. J.M.Rozmus. The Density-Tracking Self-Organized Map. // Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 44-50
63. K.Kiviluoto, P.Bergius. Two-Level Self-Organizing-Map's for Analysis of Financial Statements. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.
64. K.Kivimoto. Topology Preservation in SOM // Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 294-300
65. Kohonen T. Self-Organizing Maps. Springer: Berlin – Heidelberg, 1997.
66. Kramer M.A. Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal. 1991. V.37, No. 2. PP. 233-243.
67. LeBlank M., Tibshorany N. Adaptive principal surfaces. Journal of the American Statistical Association. 1994, Mar. V. 89, No. 425. PP. 53-64.
68. Lichtman A.J., Keilis-Borok V.I., Pattern Recognition as Applied to Presidential Elections in U.S.A., 1860-1980; Role of Integral Social, Economic and Political Traits, Contribution N 3760. 1981, Division of Geological and Planetary Sciences, California Institute of Technology.
69. M.-C.Su, I-Ch.Liu. Facial image morphing by self-organizing feature maps. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1999.
70. M.-C.Su, T.-K.Liu, H.-T.Chang. An efficient initialization scheme for the self-organizing feature map algorithm. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1999.
71. M.Chang, H.Yu, J.Heh. Evolutionary Self-Organizing Map // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.
72. Q.Liu, S.Ray, S.Levinson, T.Huang, J.Huang. Temporal sequence learning and recognition with dynamic SOM // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1999.
73. Rossiev D.A., Golovenkin S.E., Shulman V.A., Matyushin G.V. Forecasting of myocardial infarction complications with the help of neural networks // Proc. WCNN 95. (World Congress on Neural Networks 95). - Washington, DC, July 1995.
74. Rossiev D.A., Golovenkin S.E., Shulman V.A., Matyushin G.V.The employment of neural network to model implantation of pasemaker in patients with arrhythmias and heart blocks. // Modelling, Measurement & Control. -1995.-V.48.- N.2.-pp.39-46
75. S.Garavaglia. A Heuristic Self-Organizing Map Trained Using the Tanimoto Coefficient. // Proc. of International Joint Conference on Neural NetWorks. Washington, DC, 1998.

76. Shaidurov V.V. Multigrid Method for Finite Elements // Mathematics and Its Applications. Kluwer Academic Publishers, 1995.
77. T.Honkela, S.K.Lagus, T.Kohonen. Exploration of Full-Text Databases with Self-Organizing Map // Proc. of International Conference on Neural NetWorks. Washington, DC, 1996. Vol.1. PP. 56-62