ЭКСПЕРТНЫЕ СИСТЕМЫ И АНАЛИЗ ДАННЫХ
(Вычислительные системы)
1991 год
Выпуск 144

УДК 519.95

БЛОК АНАЛИЗА ДАННЫХ В ЭКСПЕРТНОЙ СИСТЕМЕ ЭКСНА

В.Н.Ёлкина, Н.Г.Загоруйко

В в е д е н и е

Материалы данной статьи представляют собой первую редакцию пользовательской документации сопровождения экспертной системы ЭКСНА для IBM-совместимых ПЭВМ [1]. Система ЭКСНА разработана коллективом сотрудников Института математики СО АН СССР и Международной Лаборатории Интеллектуальных систем СИНТЕЛ под руководством и при участии Загоруйко Н.Г., Ёлкиной В.Н. и Бушуева М.В. В разработке системы также участвовали Шемякина Е.Н., Киприянова Т.П., Ефанова Н.В. и Мурзенко Т.А..

Система ЭКСНА представляет собой инструментальный комплекс,предназначенный для построения прикладных экспертных систем, обладающих некоторыми элементами систем второго поколения или "партнерских" систем [2,3]. Одним из таких элементов, повышающих уровень интеллекта системы, является Блок Анализа Данных (БАД). Помимо таких традиционных методов анализа данных, как таксономия, выбор признаков и распознавание образов в БАД имеются программы заполнения пробелов в эмпирических таблицах и программы получения знаний из данных, что особенно важно для экспертных систем.

В данной работе приводится описание состава Блока Анализа Данных и описываются правила работы с программами этого блока.

Основу БАД составляют алгоритмы и программы пакета прикладных программ ОТЭКС, подробно описанные в [4], в связи с чем в данной работе мы ограничимся лишь кратким изложением основных идей алгоритмов и назначения программ БАД, сосредоточив внимание на правилах взаимодействия пользователя с этими программами в процессе решения прикладных задач.

Программы БАД могут использоваться как в составе системы ЭКСНА, так и в виде самостоятельного пакета прикладных программ ОТЭКС. В последнем случае пакет снабжается таким же диалоговым интерфейсом с пользователем, как и система ЭКСНА.

1. Назначение и состав БАД

Блок Анализа Данных включает в свой состав программы, которые позволяют получить ответы на многие вопросы, возникающие при обработке экспериментальных или статистических данных, в том числе на такие:

- Как сгруппировать изучаемые объекты по похожести их свойств? Верна ли Ваша гипотеза об их сходстве и различии в выбранном Вами признаковом пространстве?

- Можно ли сократить исходную систему признаков, характеризующих изучаемые объекты, выбрав наиболее информативную подсистему?

- Есть ли в исходной системе признаков те, что связаны с целевым признаком и могут обеспечить разделение на классы, заданные этим целевым признаком?

- Как отнести новый объект к одному из уже известных таксонов (образов, классов)? Есть ли среди известных классов такой, куда может быть включен этот объект?

- Как найти правило, разделяющее заданные Вами классы? Достаточна ли для этого имеющаяся информация?

- Есть ли ошибки в таблицах?

- Можно ли заполнить пробелы в таблице, восстановить значение отсутствующих данных?

- Не дублируют ли новые данные информацию, уже имеющуюся в системе?

- Не противоречат ли новые данные закономерностям, имеющимся в старых данных?

Программы, используемые в БАД, в составе ППП ОТЭКС много лет применяются для решения задач анализа информации, в том числе медицинской, геологической, экономической и в других областях. Они постоянно развиваются и совершенствуются. В рамках системы ЭКСНА мы старались сделать общение с ними как можно более простым и освободить пользователя от выполнения многих рутинных и утомляющих операций. Результаты получаемых решений снабжаются комментариями и хорошо документированы. В систему встроены описания и информация о режимах работы каждой программы. С помощью клавиши F1 можно получить "подсказку" по любой из них.

Освоив работу с БАД, Вы станете обращаться к нему постоянно, отмечая высокую обоснованность получаемых решений и простоту их получения.

Иногда могут быть и огорчения - не всегда подтверждается первоначальная гипотеза - исходных данных, необходимых для получения ожидаемого результата, может оказаться недостаточно.Но это будет стимулом для поиска дополнительной информации, возникновения новых идей, выдвижения новых гипотез, а БАД системы ЭКСНА обеспечит Вам их быструю проверку.

В состав БАД входят следующие группы программ:

** Таксономия (автоматическая классификация, группировка): FOREL, FOREL2, KRAB, KRAB2, SKAT, SKAT2, JOINT,FOREL5, FOR5R2.

** Выбор информативных признаков, распознавание: NTPP, NTPP5, PROVINF, ACQUIS, RASP (RASPP), TRF, TRF2.

** Заполнение пробелов, редактирование таблицы, прогноз: ZET-комплекс.

** Поиск аналогов, отбор объектов по заданным условиям: SIM, DIFER, SELECT.

** Анализ временных рядов: SETTIP, DEFINE, ZET.

Сюда же входят вспомогательные программы:

** Нормировка данных: NORM .

** Транспонирование таблиц данных: TRANM.

** Выдача информации о содержимом таксонов: TAKS.

** Построение кратчайшего незамкнутого пути (КНП): GRAPH.

** Ранжирование, формирование целевого вектора: RANG,ITOG, ORDER.

Для подготовки файлов к работе пользователю предоставляются следующие дополнительные возможности:

** Просмотр правил описания таблицы и ввода данных.

** Просмотр списка таблиц с данными (с расширением DAT).

** Просмотр списка таблиц с описанием данных (с расширением PET).

** Перевод таблицы данных с произвольным именем в "теку - щую" таблицу (WORK.DAT, WORK.PET, WORK.INF).

** Сохранение "текущей" таблицы под другим именем.

** Создание "текущих" файлов из исходных данных по заданным номерам.

** Работа с редактором текстов.

** Временный выход в Norton Commander (возврат в систему ЭКСНА - F10).

** Выход в меню блока анализа данных.

2. Краткая характеристика програм БАД

В данном разделе будут даны только самые общие сведения о программах, входящих в состав БАД. Более подробно их назначе -

ние, особенности и возможные интерпретации получаемых результатов рассматриваются в соответствующих разделах.

Поскольку БАД постоянно развивается и пополняется, то в каждой его конкретной версии возможны некоторые отклонения от сведений, приведенных в данном руководстве. Особенности конкретной версии обязательно отражаются в поставляемых с системой справочных сведениях, для получения которых достаточно нажать клавишу F1 во время работы с программным комплексом.

Вначале ответим на самые первые вопросы, которые обычно возникают у начинающего пользователя: о характере обрабатываемых данных, способе их организации и о решаемых задачах.

Какие данные могут быть обработаны программами БАД?

БАД системы ЭКСНА предназначен для работы с данными,которые могут быть заданы в виде таблицы "объект-свойство". Например, объекты - сельхозпредприятия, районы, области, а их свойства - характеристики угодий, количество животных, обеспеченность ресурсами, показатели работы. Если объекты - сорта семян пшеницы, то их признаками могут быть урожайность, количество белка, клейковины, сроки созревания и т.д.

Количество объектов и свойств в таблице ограничивается только объемом оперативной памяти. Большинство программ, если это не оговорено дополнительно, работают с таблицами, содержащими не более 15 тысяч чисел. Таблица может содержать ошибки и пробелы, свойства могут быть количественными, порядковыми,номинальными, бинарными.

С какими файлами работают программы?

Все программы построены так, что они работают с "текущими" файлами - файлами со стандартными именами:

WORK.DAT - таблицы данных;

WORK.PET - описания таблицы;

WORK.INF - наименования объектов и признаков.

Результат также помещается в стандартный файл (TABL.DAT). Файл WORK.INF не является обязательным. Наименования объектов и признаков обычно заменяются условными номерами. Если есть необходимость выдавать пользователю "словесно документированный" результат, то наименования объектов и признаков могут быть выданы в их истинном виде. В этом случае в директории DIALOG файл WORK.INF должен быть задан.

Для работы с алгоритмами выбора информативных признаков и распознавания необходимо выполнение ряда дополнительных требований (смотрите справки к данным режимам в меню БАД). Итак,для работы в подсистеме прикладных программ необходимо наличие (в директории DIALOG) файла данных WORK.DAT, файла описания этих данных WORK.PET и файла с названиями объектов и признаков WORK.INF. Их желательно подготовить до входа в конкретный пункт БАД.

Данные, хранящиеся в ваших директориях, вы можете перевести в "текущие", не выходя из системы. Для подготовки "теку - щих" файлов с расширениями DAT, PET и INF в системе имеются специальные вспомогательные средства в блоке "Подготовка фай - лов к работе". Обратившись в пункт этого блока "Перевод таблицы данных с произвольным именем в ТЕКУЩУЮ", вы должны будете указать путь для вызова ваших файлов. (Не забудьте, что эти файлы должны иметь одинаковое имя!). Если ваши файлы располо - жены в директории DIALOG, то достаточно указать только их имена.

Какие задачи можно решать с помощью БАД?

БАД позволяет решать следующие задачи эмпирического предсказания.

ТАКСОНОМИЯ (автоматическая классификация) .Данные, введенные в таблицу, намного легче понять, если их удается описать более кратким способом, чем перечислением объектов со всеми

их свойствами. В многомерном признаковом пространстве объекты с почти одинаковыми значениями свойств отобразятся в близкие точки, а объекты с сильно отличающимися свойствами будут представлены далекими друг от друга точками. Сгустки точек целесообразно выделить в отдельные структурные части множества - таксоны (классы, группы, образы). Программы таксономии позволяют получать таксоны сферической или произвольной формы.

РАСПОЗНАВАНИЕ. Решение о принадлежности распознаваемых объектов в алгоритмах, использующих таксономические решающие правила, принимается на основании вычисляемого в программах таксономического критерия. Объект относится к тому таксону, в структуру которого он лучше вписывается, где максимально значение критерия качества таксономии.

При принятии решения о принадлежности распознаваемых объектов по алгоритмам, использующим логические решающие правила, объект относится к тому образу, для которого будет выполнено соответствующее логическое условие.

ВЫБОР ИНФОРМАТИВНЫХ ПРИЗНАКОВ. Для выбора заданного числа наиболее информативных признаков по алгоритмам направленного таксономического поиска признаков производится таксономия всех исходных признаков. Далее, из каждого таксона выбирается по одному "типичному" (ближайшему к среднему) представителю, в результате чего система признаков сокращается. Затем тестируются различные подсистемы из типичных признаков. Лучшей подсистемой считается та, при которой получается наилучшее распознавание обучающей выборки.

В алгоритмах, основанных на логических решающих правилах, решение будет получено в виде перечня признаков, включенных в дерево логических условий.

ЗАПОЛНЕНИЕ ПРОБЕЛОВ. Алгоритм ZET предназначен для прогнозирования значений пропущенных элементов (заполнения пробелов) в таблицах "объект-свойство" и для редактирования (проверки)

всей таблицы и ее части. В реальных таблицах данных имеется избыточность, т.е. многие признаки (столбцы) связаны друг с другом определенной зависимостью, есть в таблице и объекты (строки), похожие друг на друга по значениям своих характеристик. В алгоритме выявляются такие связи и похожести, и на их основе выполняется предсказание значения пропущенного элемента с высокой точностью.

ПРОГНОЗИРОВАНИЕ. Программы ZET-комплекса можно использовать и для продолжения (прогнозирования новых значений) временного ряда. Серия программ, включенных в БАД, дает возможность прогнозировать значения как однопараметрических, так и многопараметрических рядов с помощью закономерностей, выявленных на предшествующей информации. Наряду с прогнозом новых значений временного ряда может быть также выполнен "ретроспективный прогноз": начиная с заданного момента времени, в таблице 'закрываются" известные значения, для них выполняется прогноз с одновременным сравнением вычисленных программой значений с исходными. Этот вариант особенно удобен на стадии подбора параметров программы для наилучшего выполнения основного прогноза.

В БАД имеются вспомогательные сервисные программы и программы работы с таблицами, а также программы, предоставляющие возможность выбирать аналоги для указанного объекта, отбирать из всей таблицы объекты по заданным условиям,сравнивать объект с остальными объектами в таблице,выполнять ранжирование и др.

Где можно прочитать описание БАД?

Более подробно познакомиться с алгоритмами большинства программ, используемых в БАД системы ЭКСНА, можно прочитать в следующей литературе.

1. Загоруйко Н.Г. Экспертные системы и распознавание образов //Анализ данных в экспертных системах. - Новосибирск 1986. - Вып. 117: Вычислительные системы. - С. 3-10.

2. Ёлкина В.Н., Загоруйко Н.Г. Применение ZET-метода в экспертных системах //Анализ разнотипных данных. - Новосибирск, 1983. - Вып. 99: Вычислительные системы. - С. 73-87.

3. Загоруйко Н.Г., Бушуев М.В. Меры расстояний в пространстве знаний //Анализ данных в экспертных системах. - Новосибирск, 1986. - Вып. 117: Вычислительные системы.-С. 24-35.

4. Загоруйко Н.Г., Ёлкина В.Н., Емельянов С.В., Лбов С.Г. Пакет программ ОТЭКС для анализа данных. - М.: Финансы и статистика. 1986.

5. Ёлкина В.Н., Загоруйко Н.Г., Новоселов Ю.А. Математические методы агроинформатики. - Новосибирск: Наука СО,1987.

3. Начало работы с БАД

Для запуска системы войдите в директорию DIALOG и запустите на выполнение файл DIALOG.BAT. На экране появится текст приветствия и сообщение:

```
Введите ваше имя
```

В ответ на него нужно набрать Ваше пользовательское имя, предварительно введенное в систему программой NEWUZER. В поставляемой версии системы введено имя "-" (символ пробела),для входа в систему нажмите клавишу ↵ . Если имя введено неправильно, то система проинформирует о том, что такого имени в ней нет и сеанс работы будет закончен. При благополучном исходе система выдает на экран Главное меню:

```
********* Начало *********

Работа с Базой Знаний
Работа с Блоком Анализа Данных
Запуск произвольной команды операционной системы
Временный выход в Norton Commander (возврат в
        систему ЭКСНА - F10).
```

Пользователь должен выбрать нужный пункт этого меню. Активизированная строка меню выделяется другим цветом. Назначение используемых при этом функциональных клавиш таково:

Клавиши	Выполняемая функция
cursor UP cursor DN	Перемещение по строкам меню вверх и вниз при выборе нужного режима
↵ (Return)	1. Переход в отмеченный режим в блоках меню 2. Фиксирование введенного значения параметра в обрабатывающих модулях
F1	Получение пояснения к отмеченной строке
F10	1. Окончание работы с системой ЭКСНА 2. Возврат в систему ЭКСНА после временной работы в Norton Commander
F2	Возврат в блок "начало"
ESC	Переход в предыдущий блок
F5	Запуск обрабатывающего модуля на выполнение

Для задания параметров можно использовать клавиши: Back - space, стрелки вправо и влево, алфавитно-цифровые клавиши. Для окончания задания параметра нужно нажать клавишу ↵ (Return).

Что делать в случае сбоя системы?

В случае сбоя системы следует запустить на выполнение файл SBOJ.BAT, который скопирует резервную базу по диалогу в рабочую.

Как завершить работу с системой?

Для окончания работы с системой и возврата в МС ДОС следует нажать клавишу F10. При окончании работы на экран выводится сообщение:

```
======= Сеанс работы закончен =======
========== до свидания ==============
```

Как начать работу с Блоком Анализа Данных?

Чтобы войти в БАД, следует в Главном меню выбрать пункт "Работа с Блоком Анализа Данных", т.е. подвести курсор к этой строке меню и активизировать ее, нажав клавишу ↵. После этого на экране появится меню БАД:

```
***   БЛОК АНАЛИЗА  ДАННЫХ   ***

Таксономия (автоматическая классификация, группировка)
ZET-комплекс (заполнение пробелов,  редактирование,
                                    прогнозирование)
Выбор информативных признаков, распознавание
Поиск аналогов, отбор объектов по заданным условиям
Анализ временных рядов
Вспомогательные программы
Подготовка файлов к работе
Работа с редактором текстов
```

Напомним, что для работы с программами БАД необходимо наличие файла данных WORK.DAT и файла WORK.PET, который описывает эти данные. Эти файлы желательно подготовить до входа в конкретный пункт БАД. Ознакомиться с правилами описания данных файлов, а также внести в них изменения, создать их заново можно в блоке "Подготовка файлов к работе", куда можно перейти из основного меню БАД. Для работы с алгоритмами выбора информативных признаков и распознавания необходимо выполнение ряда дополнительных требований (смотрите справки к данным режимам в меню БАД).

4. Правила создания файлов с данными, параметрами таблицы и наименованиями объектов и признаков

Для создания файлов с данными, параметрами таблицы и наименованиями объектов и признаков вы можете воспользоваться редактором текстов, встроенным в систему. Вам предлагается ознакомиться с правилами создания этих файлов (с расширением DAT, PET и INF соответственно). Их наличие необходимо для работы с Блоком Анализа Данных. Есть возможность совмещения просмотра правил с внесением информации в указанные выше файлы.

Правила создания файла с данными (с расширением .DAT).

Числа вводятся последовательно через пробел. Есть два способа ввода:

- "по объектам", т.е. сначала пишутся все показатели 1-го объекта, затем все показатели 2-го и т.д.

- "по признакам" - вначале пишется 1-й показатель для всех объектов, затем 2-й и т.д. Способ ввода данных необходимо отразить в файле описания параметров таблицы.

Правила создания файла параметров таблицы (с расширением .PET).

Любая таблица данных должна быть описана в файле ее параметров, в который заносится информация о числе объектов, признаков, наличии или отсутствии пробелов и т.д. Ниже мы приведем образец "стандартного" файла WORK.PET, в котором для наглядности пронумеруем все строки и дадим дополнительные подробные пояснения по правилам заполнения каждой из строк с номерами 13-18. В "Правилах создания файлов" использована эта же нумерация.

Обратите внимание на то, что единого для всех задач файла WORK.PET не существует. В зависимости от варианта вашей задачи вы должны заполнять те или иные строки этого файла. Строки 1-12 обязательны и присутствуют всегда. Присутствие информации в строках 13-18 не оказывает никакого влияния на работу программ, к которым эти параметры не относятся.

Стандартный образец файла WORK.PET.

```
1*  ┌В строках 1-4 вы можете поместить
2*  │любые комментарии к вашей таблице и
3*  │вашему варианту
4*  └решения.
5*  Количество объектов.
6*  Количество признаков.
7*  Способ записи (по признакам - 0, по объектам - 1).
8*  Обозначение пробела: вещественное число (0 - нет пробе -
    лов).
```

9* Тип задания целевого признака (-1 - задан в п.18, 0 - дополнительно проверяется п.12, > 0 - для некоторых программ номер признака,объявляемый целевым).
10* Количество образов.
11* Индекс типов признаков: 0 - разнотипные признаки (тип указан в файле параметров), >0 - признаки одного типа.
12* Индекс упорядочения по образам (1 - упорядочены, 0 - нет).
13* Заголовок к п.14.
14* Значения вектора типов признаков: 1 - булевы, 2 - в шкале наименований, 3 - в шкале порядка, 4 - в сильной шкале.
15* Заголовок п.16.
16* Вектор количества объектов в образах.
17* Заголовок к п.18.
18* Вектор распределения объектов по таксонам.

В пп.14,16,18 нет ограничений на длину векторов, информация каждого из остальных пунктов должна занимать не более одной строки.

При отсутствии информации в каких-либо из пп.13-18 резервировать для них место в файле .PET нет необходимости.

Если вы работаете с разными таблицами, то мы рекомендуем вам иметь для каждой из них отдельный файл .PET с именем,совпадающим с именем таблицы. Вектор распределения объектов по таксонам может быть получен в результате работы одного из алгоритмов таксономии. При каждом обращении к таксономии данный вектор будет изменяться!

Сведения о файле WORK.INF.

Наименования объектов и признаков должны храниться в файле WORK.INF в следующем порядке: вначале имена всех объектов,а затем имена всех признаков. Каждое имя длиной не более 70 символов пишется с новой строки.

Просмотр и изменение файлов.

1. Разделите экран на две части путем нажатия клавиш Ctrl-S, переведите курсор в правую часть экрана, нажав одновре-

менно клавиши Ctrl-W, переводящие курсор из одной части экрана в другую.

2. Если курсор не находится в командной строке, поставьте его туда, нажав Esc; наберите,начиная с первой позиции,команду "E WORK.PET" и нажмите ↵ . На правую половину экрана выведется файл WORK.PET.

Клавиша Esc служит для переключения текстового и командного режимов. Войдя в текстовой режим можно вносить изменения. Таким же образом можно корректировать и файл с данными или "заводить" новые файлы. Клавиши для работы в редакторе:
F3 (FILE) - сохранить файл на диске и убрать из памяти,
F4 (QUIT) - убрать файл из памяти без сохранения.

Пример одного из вариантов заполнения файла WORK.PET.

```
 1 | ВНИМАНИЕ! Параметры 13-18 таблицы нужны для работы алго-
 2 | ритмов распознавания и выбора информативных признаков
 3 | Эти данные не влияют на работу других алгоритмов и
 4 | автоматически появляются после режима таксономии
 5 |     20        (кол-во объектов)
 6 |     10        (кол-во признаков)
 7 |      1        (запись по объектам)
 8 |   9999        (обозначение пробела)
 9 |     -1        (вектор распределения указан)
10 |      2        (количество образов)
11 |      4        (все признаки в сильной шкале)
12 |      0        (объекты по образам не упорядочены)
13 |                  Отдельный целевой признак
14 |    1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
```

Как войти в любой пункт БАД?

Пользователь должен выбрать нужный ему пункт меню БАД.Для этого в меню БАД следует подвести курсор к нужной строке и активизировать ее, нажав клавишу ↵ . После этого на экране появится меню соответствующего Блока.

5. Основные принципы работы с программами

Работа со всеми программами БАД построена по единому принципу. Когда вы вошли в соответствующий раздел БАД, система

начнет задавать вам вопросы, чтобы уточнить постановку вашей задачи, выбрать конкретную программу и определить входные параметры для решения. Ряд ответов вам будет предложен "по умолчанию". Если предлагаемый вариант вас устраивает, то с помощью стрелок управления курсором вы можете пройти дальше, к следующему вопросу меню. Если вы желаете изменить значение, то вы должны набрать его на клавиатуре. При этом изменится цвет строки, в которую вносятся изменения. Нажатие клавиши ↵ фиксирует введенные значения и цвет строки восстанавливается.

До запуска программы вы можете изменять каждое значение многократно. Если вы не помните назначение какого-либо параметра или хотите уточнить что-либо относительно того пункта меню, где вы в данный момент находитесь, то вы можете получить справку, нажав клавишу F1.

Напоминаем вам, что до начала счета вы должны поместить таблицу исходных данных и файл с описанием ее параметров в рабочие файлы WORK.DAT и WORK.PET соответственно.

Когда все готово к решению, то для запуска программы следует нажать клавишу F5. Результаты решения всегда помещаются в файл TABL.DAT. Вы всегда можете посмотреть их и, если потребуется, отредактировать, отобрав ту информацию, которая вам может понадобиться для длительного хранения и дальнейшего использования. Система дает вам также возможность сразу же по окончании решения поместить результаты и в файл с указанным вами именем.

Если вы желаете получить на тех же исходных данных другой вариант решения, изменив частично условия рещения, то вам нужно будет только заменить соответствующее значение параметра и опять нажать клавишу F5. Не забудьте о том, что содержимое файла TABL.DAT при этом изменится.

6. Описание программ Блока Анализа Данных

Программы Блока Анализа Данных могут быть использованы и вне экспертной системы ЭКСНА в виде диалогового программного комплекса (ДПК) ОТЭКС, поэтому дальнейшее краткое описание программ БАД может служить в качестве документации сопровождения и для пользователей ДПК ОТЭКС. Библиотека БАД построена как открытая система, куда с помощью специально созданного конструктора диалога могут быть добавлены любые другие программы пользователя.

6.1. Программы таксономии (автоматической классификации).

Мы уже говорили о том, что данные, сведенные в таблицу, намного легче понять, если их удается описать более кратким способом, чем перечислением объектов со всеми их свойствами. Очень полезной для этой цели может оказаться геометрическая интерпретация табличной информации. Поскольку каждая строка таблицы - это объект, определенный значениями его характеристик, то этот объект можно представить себе в качестве точки много - мерного признакового пространства. В таком пространстве объекты с почти одинаковыми значениями свойств отобразятся в близ - кие точки, а объекты с сильно отличающимися свойствами будут далеки друг от друга.

Так как в таблице могут быть данные о достаточно похожих друг на друга, не очень похожих, или даже сильно различающихся объектах, то легко себе представить, что если бы мы смогли "заглянуть" в признаковое пространство, то увидели бы там сгустки-"облака", образованные похожими точками. Могут оказаться сгустки в разных местах пространства, близкие и далекие друг от друга, а также изолированные точки, структуры,похожие на линии и т.д.

Сгустки точек целесообразно выделить в отдельные струк - турные части множества - таксоны (классы, группы, образы,кла-

стеры). Таксоны могут иметь различную форму - сферическую,произвольную. Задачу выявления таких групп взаимосвязанных объектов в БАД решают программы ТАКСОНОМИИ (автоматической класси - фикации).

Что дает, зачем нужна таксономия?

Закономерности "групповой похожести" позволяют сильно сократить описание таблицы при малой потере информации. Если,например, были выделены таксоны сферической формы, то вместо перечисления всех объектов можно дать список "типичных" или "эталонных" представителей групп, указав допустимые отличия объектов таксона от эталона, т.е. значение радиуса сферы. В качестве таких эталонов могут быть взяты центры таксонов или наибо - лее близкие к центрам реальные объекты. При небольшом числе групп описание данных становится обозримым и легко интерпретируемым.

В каждой классификации имеются элементы субъективного и объективного. Все реальные объекты имеют бесконечное количество свойств. Выделение конечного числа свойств для описания объекта - акт в большой степени субъективный. Выбор цели также субъективен. Но если известна цель, для которой формируются таксоны из данного множества объектов по заданным их характе - ристикам, то можно достаточно объективно проверить качество таксономии для этой цели.

Иногда пользователь "обвиняет" программу таксономии за то, что на его данных она дала плохой, с его точки зрения, результат. Но следует иметь в виду, что нередко встречаются наборы данных, генеральная совокупность которых может быть описана, например, нормальным законом распределения. В этом случае невозможно выдать "хорошую" группировку точно на 5 или 10 таксонов: будет, как правило, получен один большой таксон, несколько маленьких и много отдельных точек.

Таксономический анализ данных полезен не только в том случае, когда получается "хорошая" таксономия. "Плохая" таксономия также несет информацию о структуре множества, она говорит о том, что при описании объектов данным набором признаков множество не расслаивается на обособленные подмножества. Характеристики, которые, как предполагается, должны быть связаны с целью, могут оказаться и неинформативными, не относящимися к делу.

Если желательная классификация известна пользователю хотя бы частично, но при делении на таксоны в состав одного и того же таксона попадают представители разных по мнению пользователя классов, то можно предположить, что или неудачно выбрано описание объектов, или прежние представления о сходстве и различии изучаемых объектов были ошибочными.

Есть еще один полезный аспект применения методов таксономии, в частности алгоритмов FOREL (FOREL2) и SKAT (SKAT2) - для выявления некоторых закономерностей во временных рядах. Этот вопрос будет подробнее рассмотрен в пункте "Анализ временных рядов".

Многолетний опыт использования методов таксономии говорит о том, что таксономический анализ данных является мощным средством познания закономерностей структуры множества изучаемых объектов или явлений.

Начало работы с программами таксономии.

Если вы переведете курсор к строке "Таксономия (автоматическая классификация, группировка)" в меню БАД и нажмете клавишу F1, то на экране появится информация о том, какие алгоритмы таксономии имеются в системе:

В системе реализованы следующие алгоритмы таксономии для количественных признаков:

** FOREL (FOREL2) - выделение таксонов сферической формы;
** SKAT (SKAT2) - выделение таксонов сферической формы с проверкой полученных таксонов на независимость;
** JOINT - укрупнение (объединение) таксонов;
** FOREL5 (FOR5R2) - выделение таксонов сферической формы в пространстве бинарных признаков.

6.1.1. Алгоритм FOREL.

Алгоритм FOREL является базовым для целого ряда алгоритмов таксономии, и в первую очередь для алгоритмов выделения таксонов простой сферической формы.

Разные алгоритмы семейства отличаются друг от друга следующими особенностями:

- задается ли точно требуемое число таксонов пользователем, или автоматически выбирается лучший вариант разбиения;

- с большими массивами данных могут работать программы, или только с теми, которые одновременно умещаются в оперативную память;

- производится ли для выделенных сферических таксонов дополнительная проверка их самостоятельности или нет;

- в каких шкалах представлены исходные данные и др.

Коротко главная идея, на которой построен алгоритм FOREL, основана на движении многомерной сферы (гиперсферы) в область наибольшей плотности точек. Как только "шарик" достигает области пространства с локально максимальной плотностью объектов, движение прекращается и все точки, охваченные остановившейся гиперсферой, объявляются принадлежащими одному таксону. Точки, попавшие в эту сферу, из дальнейшего рассмотрения исключаются. Та же процедура выделения таксонов применяется к оставшимся точкам до тех пор, пока все исходное множество объектов не будет распределено по таксонам. Таксонам присваиваются номера в порядке их выделения.

Понятно, что характер движения "шарика" определяется как расположением точек в пространстве, так и заданным радиусом сферы. Несколько слов следует сказать о том, что стоит за понятием радиуса R. Представьте себе любую сферу, в которой есть какие-то точки,и центр этой сферы расположен в их центре тяжести. Независимо от того, как точки расположены внутри сферы, вы всегда с уверенностью можете утверждать, что любая из этих точек находится не далее чем на расстоянии R от центра и расстояние между любыми двумя точками не более 2R. Таким образом, при такой процедуре выделения таксонов радиус R есть не что иное, как оценка максимально допустимого отличия объектов в пределах одного таксона от среднего, "типичного" объекта.

Изменяя радиус, можно получить разделение исходного множества объектов на разное число групп. При этом,чем меньше радиус, тем больше количество таксонов и тем более "похожими" будут объекты внутри таксона. Желательное для пользователя количество таксонов K может быть найдено соответствующим подбором радиуса R. С этой целью радиус последовательно уменьшается с равномерным шагом DR от R_{max}, при котором все исходные точки объединяются в один таксон до тех пор, пока не будет получено количество таксонов, возможно близкое K.

Программа FOREL (FOREL2).

Программы FOREL и FOREL2 реализуют алгоритм FOREL выделения таксонов сферической формы и отличаются процедурой изменения радиуса: FOREL - с постоянным шагом, а FOREL2 - с переменным, подбор которого осуществляется методом последовательных приближений. В программе FOREL возможны различные режимы работы: выполнение одного варианта разбиения с заданным значением радиуса; получение заданного числа ID вариантов разбиений при равномерном уменьшении радиуса в пределах от R до R2 или от R_{max} до 0.

Чтобы избежать лишних вычислений, предусмотрена возможность задания желательного количества таксонов. Процесс останавливается либо тогда, когда исчерпаны все значения радиусов, либо когда полученное число таксонов превышает заданное.

Обратите внимание на различие результатов, которые могут быть получены программами FOREL и FOREL2 при одних и тех же ограничениях на число итераций и желательное количество таксонов.

Как уже было сказано выше, программа FOREL выполняет разбиение, равномерно уменьшая радиус, и выбирает ближайшее из полученных решений. Программа FOREL2 не остановится, если уже получено таксонов больше заданного порога, и даже если их получено ровно столько, сколько требовалось, а будет подбирать методом последовательных приближений заданное число таксонов с наибольшей плотностью объектов в них.

При обращении к программе FOREL система запрашивает границы изменения радиуса. При задании R1 > R(I) > R2 таксономия выполняется ID раз, начиная с радиуса R1 до R2 с шагом DR = = (R1 - R2)/ID. При R1 = R2 > 0 таксономия проводится один раз с радиусом R. При R1 = R2 = 0 вычисляется максимальный радиус R_{max} и текущий радиус меняется в пределах $0 < R(I) < R_{max}$ при $DR = R_{max} / ID$.

Программы FOREL5 (FOR5R2).

Программы FOREL5 (FOR5R2) фактически являются аналогами программ FOREL (FOREL2), предназначенными для работы только с бинарными признаками (принимающими значения 0 или 1). Различие в алгоритмическом плане состоит в том, что при построении очередной гиперсферы ее центр смещается в ближайшую к вычисленному значению реальную вершину (с бинарными координатами).

6.1.2. Режим печати.

Возможности печати результатов одинаковы для всех алгоритмов таксономии. При вызове в головном меню строки "Режим печа-

ти" пользователю будут предложены следующие возможности получения информации о результатах решения задачи:

- без печати;

- итоговая таблица распределения объектов по таксонам;

- таблица распределения объектов по таксонам на каждом шаге изменения радиуса.

При выборе того или иного режима печати можно дополнительно с помощью строки "Режим печати содержимого таксона" выбрать следующие ее разновидности:

- короткая печать: печатаются только номера объектов, которые попали в таксон;

- полная печать: указываются не только номера объектов,но и значения признаков для каждого объекта, средние значения,дисперсии, максимум и минимум по каждому признаку для данного таксона.

На печать выдается информация только о таксонах с количеством точек больше "минимального" и меньше "максимального" значений, заданных пользователем.

С помощью дополнительной программы могут быть распечатаны имена объектов, попавших в таксон (см.п. "Выдача дополнительной информации о содержимом таксонов" в блоке "Вспомогательные программы"). Эта же программа может выдать и информа - цию о том, как ведет себя целевой показатель на объектах каждого из полученных таксонов.

Пример решения задачи по программе FOREL.

Рассмотрим конкретный пример - таблицу сведений об урожайности зерновых для 66 регионов РСФСР за последовательные 14 лет. В качестве "объекта" примем строку из 14 значений - дан - ные для одного региона за все рассматриваемые годы. Не забудьте поместить эту таблицу в файл WORK.DAT. Содержание таблицы приведено в приложении.

Информация, которую необходимо задать в файле WORK.PET для работы программы (не считая первых 4-х строк, отведенных для комментариев), будет для этой таблицы выглядеть следующим образом:

```
   66   количество объектов
   14   количество признаков
    1   тип записи (1-по объектам)
 9990   обозначение пропуска
   -1   (для программы FOREL безразлично)
    5   (для программы FOREL безразлично)
    4   все признаки в сильной шкале
```

При обращении к программе FOREL вам будут заданы вопросы о конкретном варианте решения. Ниже приведены эти вопросы и ответы на них под соответствующей строкой. Нажатие клавиши F1 обеспечит вам получение справочных сведений,относящихся к рассматриваемой в данный момент строке.

```
константа нормировки: 0 - нет, 1 - к [0,1], 2 - по дисперсиям
2
желательное количество таксонов
10
ограничение на число итераций (шагов работы программы) - ID
10
режим печати (0 - нет печати, 1 - итог, 2 - пошаговая)
2
константа обращения к программам: 1 - FOREL, 2 - FOREL2
1
начальное значение радиуса (R1)
0
конечное значение радиуса (R2)
0
режим печати содержимого таксонов (0 - короткая, 1 - полная)
0
минимальное число точек в таксоне для выдачи на печать
0
максимальное число точек в таксоне для выдачи на печать
300
```

После ответа на вопросы о конкретных значениях параметров решения и запуска программы (нажатием клавиши F5) вы увидите на экране информацию о ходе решения. Вся эта информация после

окончания решения будет находиться в файле TABL.DAT, может быть сохранена вами под другим именем или вызвана в редактор для предварительного просмотра и отбора интересующей вас информации. Программа прежде всего проинформирует о том, как пронормированы исходные данные:

```
Исходный массив нормируется по дисперсиям
```

Затем будут повторены заданные вами условия, т.е. входные параметры, и для облегчения расшифровки информации даны сведения о характере ее представления по ходу решения:

```
                ВХОДНЫЕ  ПАРАМЕТРЫ

      Количество:  признаков  N = 14     таксонов  K = 10
                   объектов   M = 66     итераций  ID = 10
      Диапазон значений радиуса RD1 = 0   RD2 = 0

   РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ  ПО  ТАКСОНАМ ПРЕДСТАВЛЯЕТСЯ
                СЛЕДУЮЩИМ ОБРАЗОМ:

На месте, соответствующем порядковому номеру  объекта,
ставится номер таксона, которому он  принадлежит.Знаком  '-'
  отмечены объекты, ближайшие к центру своего таксона.
```

По заданным в данном случае условиям ($R_1 = R_2 = 0$) максимальный радиус R_{max} будет вычисляться в программе, а текущий радиус R_i будет последовательно изменяться в пределах от R_{max} до 0 с шагом $DR = R_{max}/10$. Ниже приведен первый фрагмент решения для $R = R_{max} - DR$. Числа перед символом "*" соответствуют номеру первого элемента в строке.

```
      Радиус  .998E+01     Шаг по радиусу  .111E+01

          РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ТАКСОНАМ

*** 1 ** 2 ** 3 ** 4 ** 5 ** 6 ** 7 ** 8 ** 9 ** 0 ***
  1* 1    1    1    1    1    1    1    1   -1    1
 11* 1    1    1    1    1    1    1    1    1    1
 21* 1    1    1    1    1    1    1    1    1    1
 31* 1    1    1    1    1    1   -2    1    1    1
 41* 1    1    1    1    1    1    1    1    1    1
 51* 1    1    1    1    1    1    1    1    1    1
 61* 1    1    1    1    1    1
```

КОЛИЧЕСТВО ОБ'ЕКТОВ В ТАКСОНАХ

```
***  1 ** 2 ** 3 ** 4 ** 5 ** 6 ** 7 ** 8 ** 9 ** 0 ***
 1* 65    1
```

На первом шаге программа выделила всего 2 таксона: таксон 1, включивший в себя 65 точек, с наиболее близкой к центру точкой 9, и таксон 2, состоящий только из одной точки 37.

Выдача результата на каждом шаге для каждого R_i производится по единому образцу, так что повторять здесь ее всю нет необходимости. Рассмотрим только серию последовательных изме - нений количества выделенных таксонов и распределения объектов в них.

КОЛИЧЕСТВО ОБ'ЕКТОВ В ТАКСОНАХ

```
 Радиус    .998E+01                Шаг по радиусу    .111E+01
1*  65     1
 Радиус    .887E+01                Шаг по радиусу    .111E+01
1*  63     3
 Радиус    .776E+01                Шаг по радиусу    .111E+01
1*  63     3
 Радиус    .665E+01                Шаг по радиусу    .111E+01
1*  62     4
 Радиус    .554E+01                Шаг по радиусу    .111E+01
1*  60     1    2    3
 Радиус    .444E+01                Шаг по радиусу    .111E+01
1*  53     8    3    1    1
 Радиус    .333E+01                Шаг по радиусу    .111E+01
1*  47     9    3    4    1    1    1
```

```
 Радиус    .222E+01                Шаг по радиусу    .111E+01
  **  1 ** 2 ** 3 ** 4 ** 5 ** 6 ** 7 ** 8 ** 9 ** 10 *
 1*  24    3    2    1    1    3    1    4    1     1
11*   1    1    1    5    1    1    1    6    1     2
21*   1    1    1    1    1
```

Обратите внимание на то, как долго сохраняется разделение на 2 группы - одну очень большую и вторую совсем маленькую. Характер распределения практически сохранился на протяжении всего решения. Но тем не менее, если требуется скомпоновать объек-

ты на большее число таксонов, то программа предлагает лучший вариант.

Посмотрите на 2 последних результата в этой таблице. При радиусе 0.333 получено 7 таксонов, а при 0.222 уже 25. Мы просили разделить все множество на 10 групп. Так как 7 ближе к 10, чем 25, то программа выбирает вариант с разбиением на 7 групп и выдает этот результат как окончательный.

Радиус .333E+01 Шаг по радиусу= .111E+01

РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ТАКСОНАМ

	** 1	** 2	** 3	** 4	** 5	** 6	** 7	** 8	** 9	** 10	**
1*	1	1	1	2	1	1	1	1	-1	1	
11*	1	1	-6	1	1	1	1	1	1	1	
21*	1	1	2	2	-5	2	1	1	-7	1	
31*	1	1	1	1	4	1	3	2	2	2	
41*	-3	3	2	1	1	1	1	1	1	1	
51*	1	1	1	1	1	1	1	1	-4	1	
61*	4	1	1	1	4	-2					

КОЛИЧЕСТВО ОБ'ЕКТОВ В ТАКСОНАХ

	*** 1	** 2	** 3	** 4	** 5	** 6	** 7	** 8	** 9	** 10	**
1*	47	9	3	4	1	1	1				

Поскольку в больших массивах данных анализировать распределение объектов по рабочей выдаче довольно утомительно, то можно окончательный вариант дополнительно распечатать более подробно (см. программу TAKSON в блоке вспомогательных программ). Ниже приведен вариант такой печати для разбиения, полученного программой FOREL.

*** Печатается информация о таксонах, содержащих больше 0 и меньше 300 точек ***

Таксон	Номера точек, попавших в таксон									
1**	1	2	3	5	6	7	8	9	10	11
	12	14	15	16	17	18	19	20	21	22
:	27	28	30	31	32	33	34	36	44	45
:	46	47	48	49	50	51	52	53	54	55

```
      56    57    58    60    62    63    64
2 **  4     23    24    26    38    39    40    43    66
3 **  37    41    42
4 **  35    59    61    65
5 **  25
6 **  13
7 **  29
```

Если есть необходимость, то этот же результат можно вы - дать и в словесной форме, с использованием информации файла WORK.INF.

Пример решения задачи по программе FOREL2.

Мы уже говорили о том, что программы FOREL и FOREL2 различаются процедурой изменения радиуса. Ниже приводится таблица значений и приращений радиуса, полученных программой FOREL2, и распределения объектов по таксонам при этих значениях. Сравните ход решения и результаты программ FOREL и FOREL2 и вы еще лучше поймете возможности каждой из них и особенности их применения.

	Радиус	Шаг по радиусу	Количество объектов в таксонах
1	.554E+01	.554E+01	60 1 2 3
2	.277E+01	.277E+01	39 7 2 2 1 1 3 1 6 1 1 1 1
3	.416E+01	.139E+01	53 8 3 1 1
4	.347E+01	.693E+00	48 8 4 3 1 1 1
5	.312E+01	.347E+00	43 8 7 1 3 1 1 1 1
6	.295E+01	.173E+00	41 6 3 1 2 1 7 2 1 1 1
7	.303E+01	.866E-01	43 6 3 1 7 3 1 1 1
8	.299E+01	.433E-01	42 6 2 1 3 7 1 1 1 1 1
9	.301E+01	.217E-01	43 6 3 1 7 3 1 1 1
10	.300E+01	.108E-01	43 6 3 1 7 3 1 1 1
11	.295E+01	.173E+00	41 6 3 1 2 1 7 2 1 1 1

ИНФОРМАЦИЯ О ТАКСОНАХ

Таксон	Номера точек, попавших в таксон
1 **	1 2 3 5 6 7 8 9 0 11
	12 14 15 16 17 18 19 20 21 22
	27 30 31 32 33 34 26 46 48 49

	50	51	52	53	54	55	56	57	58	62
	63									
2 **	4	23	24	26	39	66				
3 **	37	41	42							
4 **	28									
5 **	40	43								
6 **	13									
7 **	35	45	59	60	61	64	65			
8 **	44	47								
9 **	29									
10 **	38									
11 **	25									

6.1.3. Алгоритм SKAT (SKAT2).

Алгоритмы семейства SKAT отличаются от алгоритмов FOREL добавкой, предназначенной для проверки каждого полученного таксона на "устойчивость" или "типичность". Для этого каждый таксон, сразу же после его получения по программе FOREL с радиусом R, запускается снова на всем множестве точек и с тем же радиусом R, а в качестве начальной точки принимается его центр. При этом некоторые ("неустойчивые") таксоны, обычно мелкие, лежащие по соседству с крупными таксонами, сдвигаются со своих мест и скатываются к соседним крупным ("устойчивым") таксонам. Подчеркнем, что при проверке каждого очередного таксона программа работает на всем множестве точек. Программа SKAT основана на использовании программы FOREL, а программа SKAT2 - программы FOREL2.

Вариант выполнения программы выбирается с помощью параметров, которые задаются в виде целых чисел. Возможны следующие варианты решений:

- проверка на устойчивость не осуществляется;
- мелкие таксоны проверяются на присоединение к ближайшим крупным таксонам;
- каждый новый таксон проверяется на устойчивость по отношению к ранее полученным;

- после объединения крупных таксонов мелкие таксоны при - соединяются к ближайшим крупным.

Алгоритмы выдают решение в виде перечня всех устойчивых таксонов и указания, какие из неустойчивых таксонов к ним тя - готеют. С помощью меню можно выбрать один из режимов печати.

Возможные режимы печати:

- без печати;

- выдается таблица распределения объектов по таксонам для итогового результата таксономии;

- выдается также информация о ближайших таксонах, номера мелких таксонов, находящихся от центров предшествующих на расстоянии > 2R,

- на каждом шаге выдается та же информация, что и в предыдущем режиме, только вместо таблицы распределения объектов по типичным таксонам печатаются номера таксонов, объединенных в типичном.

Пример решения по программе SKAT.

Рассмотрим тот же пример, что и в программе FOREL - таб - лицу сведений об урожайности (см. приложение 1). Файл WORK.PET для работы программы SKAT такой же, как и в предыдущем случае. При обращении к программе SKAT вам будут заданы вопросы о конкретном варианте решения:

```
константа нормировки: 0 - нет, 1 - к [0,1], 2 - по дисперсиям
 2
желательное количество таксонов
 10
ограничение на число шагов работы программы - ID
 10
количество таксонов, с которого проверяется типичность
 5
1-й  параметр для задания проверки на типичность
 1
2-й  параметр для задания проверки на типичность
 1
```

```
режим печати (0 - нет, 1;2 - итог, 3;4 - пошаговая)
 3
константа для обращения к программам: 1 - SKAT, 2 - SKAT2
 1
начальное значение радиуса  (R1)
 0
конечное значение радиуса (R2)
 0
режим печати содержимого таксонов (0 - короткая, 1 - полная)
 0
минимальное число точек в таксоне для выдачи на печать
 0
максимальное число точек в таксоне для выдачи на печать
 71
```

Начало решения по программе SKAT и FOREL одинаково,но программа SKAT дополнительно информирует о назначении ее специфических параметров:

```
Количество таксонов, с которого начинается проверка
     на типичность (устойчивость)   - 5
Условие 1 для проверки таксонов на типичность PK = 1
Условие 2 для проверки таксонов на типичность PS = 1

     Таксоны с количеством объектов ≤ PK названы мелкими
При PK = 0, PS = 0 объединение таксонов не происходит
При PK=1,  PS>0 проверяются на типичность все новые таксоны
При PK>0,PS=0 проверяются на типичн. только мелкие таксоны
При PK>1,PS>0 проверяются на типичность сначала только но-
        вые таксоны, а затем мелкие таксоны (таксоны,отстоя-
        щие от центров ближайших крупных на расстояние >2R,
        считаются устойчивыми)
```

До тех пор, пока число таксонов не достигнет заданного предела (в нашем примере 5), программа SKAT работает как FOREL, информируя после каждого шага о том, что "ПРОВЕРКА НА ТИПИЧНОСТЬ НЕ ПРОИЗВОДИТСЯ". При получении заданного числа "элементарных" таксонов программа SKAT начинает их проверку согласно заданным условиям. Результаты проверки сообщаются в виде таб - лицы:

* ИНФОРМАЦИЯ О БЛИЖАЙШИХ ТАКСОНАХ *

Номера ближайших таксонов		Расстоя - ние между центрами	Номера ближайших таксонов		Расстоя - ние между центрами
старые	новые		старые	новые	
1 5	1 3	5.0	2 4	1 1	5.0
3 2	2 1	6.5	4 2	1 1	5.0
5 1	3 1	5.0			

* СТАРЫЕ - номера таксонов до проверки их на типичность
НОВЫЕ - номера таксонов, полученных после проверки

РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ТИПИЧНЫМ ТАКСОНАМ*

типичный таксон	количество объектов	номера таксонов, объединенных в типичном
1	62	1 2 4
2	3	3

* в таблице выдается информация о крупных типичных таксонах
общее количество типичных таксонов K = 3

Вы помните, что в условиях задачи мы просили выделить 10 типичных таксонов. Поэтому программа выполнила еще несколько разбиений по программе FOREL, получив в конце концов 7 "элементарных" таксонов, из которых затем были сформированы 5 устойчивых. Сообщение о полученном решении при этом выдается в таком виде:

* И Т О Г О В А Я И Н Ф О Р М А Ц И Я *

Радиус .33264E+01 Шаг по радиусу .11088E+01
Количество выделенных таксонов = 7

РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ТАКСОНАМ

***	1 **	2 **	3 **	4 **	5 **	6 **	7 **	8 **	9 **	0 **
1*	1	1	1	2	1	1	1	1	-1	1
11*	1	1	-6	1	1	1	1	1	1	1

21*	1	1	2	2	-5	2	1	1	-7	1
31*	1	1	1	1	4	1	3	2	2	2
41*	-3	3	2	1	1	1	1	1	1	1
51*	1	1	1	1	1	1	1	1	-4	1
61*	4	1	1	1	4	-2				

КОЛИЧЕСТВО ОБ'ЕКТОВ В ТАКСОНАХ

***	1	**	2	**	3	**	4	**	5	**	6	**	7	**	8	**	9	**	0	**
1*	47		9		3		4		1		1		1							

* ИНФОРМАЦИЯ О БЛИЖАЙШИХ ТАКСОНАХ *

Номера ближайших таксонов				Расстояния между центрами	Номера ближайших таксонов				Расстояния между центрами
старые		новые			старые		новые		
1	4	1	1	3.5	2	1	1	1	4.3
3	6	2	4	6.2	4	1	1	1	3.5
5	2	3	1	5.1	6	2	4	1	4.8
7	1	5	1	5.1	6	2	4	1	4.8

* СТАРЫЕ - номера таксонов до проверки их на типичность
НОВЫЕ - номера таксонов, полученных после проверки

РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ТИПИЧНЫМ ТАКСОНАМ

Типичный таксон	Количество объектов	Номера таксонов, объединенных в типичном **-1**-2**-3**-4**-5**-6**-7**-8
1	60	1 2 4
2	3	3

* в таблице выдается информация о крупных типичных таксонах, общее количество типичных таксонов K = 5.

Сравнение результатов программ FOREL и SKAT говорит о том, что таксоны 2 и 4, полученные программой FOREL, могут и не считаться самостоятельными (устойчивыми). Ниже приведен результат, полученный при тех же исходных условиях программой SKAT2. При радиусе .249E+01 с шагом по радиусу .217E-01 на основе 18 элементарных было найдено 10 типичных таксонов:

ИНФОРМАЦИЯ О ТАКСОНАХ

Таксон		Номера точек, попавших в таксон									
1	**	1	2	3	5	6	7	8	9	10	11
		12	14	15	16	17	18	19	20	21	22
		23	24	26	27	28	30	31	32	33	34
		35	36	39	44	45	46	47	48	49	50
		51	52	53	54	55	56	57	58	59	60
		61	62	63	64	65	66				
2	**	38									
3	**	25									
4	**	13									
5	**	37									
6	**	4									
7	**	43									
8	**	41	42								
9	**	29									
10	**	40									

Сопоставление результатов работы программ SKAT и SKAT2 попробуйте провести самостоятельно.

6.1.4. Алгоритм KRAB (KRAB2).

Этот алгоритм предназначен для таксономии объектов на таксоны произвольной, не заданной заранее формы, и работает следующим образом. Вначале проводится кратчайший незамкнутый путь (КНП) между всеми точками (объектами). Если задано число таксонов K, то путем перебора находятся такие (K-1) ребер, проведение границ по которым дает максимальное значение функционала качества таксономии F, в качестве аргументов которого используется расстояние между объектами внутри таксона (r), расстояние между таксонами (d), характеристика равномощности таксонов (h) и характеристика локальной плотности распределения точек в пространстве признаков (l). Общий вид функционала качества: $F = \ln(d \cdot h) 3 (r \cdot l)$.

Расстояния r и d, а также величина l вычисляются с использованием ребер КНП; r - это средняя длина всех внутренних

ребер, а d - средняя длина граничных ребер КНП, т.е. тех ребер, по которым проходят границы между таксонами. Аргумент l тем больше, чем больше отличаются длины граничных ребер от примыкающих к ним внутренних ребер. Величина аргумента h меняется в пределах от 0 до 1. Наибольшее значение h получается в том случае, когда все таксоны имеют одинаковое число объектов.

В алгоритме возможно вычисление критерия качества таксономии как с учетом,так и без учета равномерности распределения количества объектов по таксонам.

Если числа объектов и таксонов велики, то перебор становится неприемлемо большим. Некоторое сокращение перебора достигается путем предварительного отбора ребер - претендентов, т.е. ребер, по которым с большой вероятностью могут пройти границы. Определение ребер-претендентов делается путем сравнения длин соседних ребер КНП. Из числа претендентов исключаются ребра, которые короче самого короткого из примыкающих к нему ребра.

При большом числе объектов (M > 50) обычно используется алгоритм KRAB2. Для ускорения процедуры таксономии в этом алгоритме вначале выполняется предварительная группировка M объектов в K1 таксон (K < K1 < M) программой FOREL2. Желательное количество K1 таксонов задается пользователем из содержательных соображений. В результате предварительной таксономии число таксонов должно быть больше, чем количество окончательных классов, и в то же время не слишком большим, чтобы последующее решение могло бы быть получено за приемлемое время. Затем центры этих мелких таксонов служат объектами таксономии по алгоритму KRAB. В этом алгоритме можно при вычислении критерия качества F учитывать центры таксонов как с равными весами,так и с весами, пропорциональными числу включенных в них объектов.

Возможные режимы печати:

- выдаются таблица распределения объектов по таксонам и значение критерия качества таксономии;

- выдаются также КНП в виде длин ребер и номеров связанных с ними вершин (объектов) и КНП, упорядоченный по длинам ребер;

- выдается также количество ребер, из которых выбираются граничные, количество объектов в каждом таксоне и номера этих объектов, граничные ребра и номера связанных ими вершин.

Пример решения по программе KRAB (KRAB2).

Рассмотрим вариант решения по программе KRAB2. При обращении к программе KRAB вам не будут задаваться вопросы относительно условий работы программы FOREL2, так как эти пункты нужны только для KRAB2. Все остальные параметры идентичны.

```
константа нормировки: 0 - нет, 1 - к [0,1], 2 - по дисперсиям
2
желательное количество таксонов
4
учитывать ли равномерность распред. объектов: 1 - нет,2 - да
1
проводить ли предварит. отбор претендентов: 0 - нет,1 - да
1
режим печати: 0 - нет печати, 1,2 - итог, 3 - полная
3
константа для обращения к программе: 1 - KRAB, 2 - KRAB2
2
количество такоонов, предварительно выделяемых п/п FOREL2
15
количество итераций для п/п  FOREL2
7
режим печати содержимого таксонов (0 - короткая, 1 - полная)
0
минимальное число точек в таксоне для выдачи на печать
0
максимальное число точек в таксоне для выдачи на печать
70
```

Программа FOREL2, выполнив заданные ей 7 итераций, выделила 16 предварительных таксонов, на основе центров которых далее нужно получить 4 таксона программой KRAB2. Кроме стандартной для программы FOREL2 информации, вы увидите сведения о ее результатах и в следующем виде:

ПРОГРАММА TIP

ВЕКТОР СООТВЕТСТВИЯ

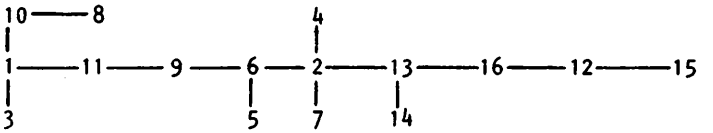
Новые номера	1	2	3	4	5	6	7	8
Исходные номера объектов	4	9	13	15	25	26	29	37
Исходные номера таксонов	9	1	10	13	6	2	7	16
Вектор весов	1	34	1	1	1	9	1	1
Новые номера	9	10	11	12	13	14	15	16
Исходные номера объектов	38	42	43	45	55	56	59	64
Исходные номера таксонов	15	12	11	5	8	4	3	14
Вектор весов	1	2	2	1	1	4	5	1

Далее для этих 16 объектов (в новой нумерации) будет построен граф связи (КНП) и информация о нем будет выдаваться для удобства в двух вариантах - в порядке построения и в упорядоченном по возрастанию длин связывающих ребер. К сожалению, на данный момент мы еще не имеем возможности представить его в графической форме, но при небольшом числе объектов это не очень обременительно сделать и вручную. Здесь мы приведем сведения о полученном пути, в порядке его построения.

Кратчайший незамкнутый путь (КНП)

Длина ребра; номера вершин (объектов)					
.375E+01	.299E+01	.336E+01	.340E+01	.249E+01	.290E+01
1	11	9	6	2	2
11	9	6	2	4	13
.244E+01	.374E+01	.282E+01	.231E+01	.385E+01	.487E+01
13	13	16	12	1	2
14	16	12	15	3	7
.498E+01	.572E+01	.316E+01			
6	1	10			
5	10	8			

Если нарисовать соответствующую данной выдаче графическую схему, то она будет выглядеть следующим образом:



После предварительного анализа построенного графа выяснено, что дальнейшему рассмотрению подлежат только 10 из 15 ребер. Перебор всех возможных комбинаций-претендентов (из 10 по 3) дал следующие результаты:

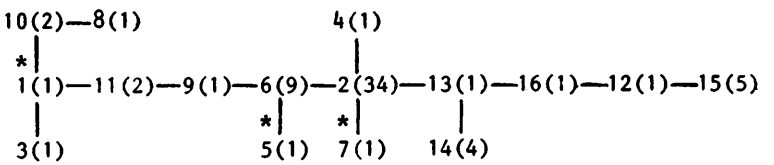
Количество объектов 16, таксонов 4

Количество ребер, из которых выбираются граничные 10

Граничные ребра	1**	2**	3**	4**	5**	6**	7**	8**	9** 0
Номера вершин	1	6	2						
	10	5	7						

Критерий качества таксономии F1 = 2.894

На схеме исходного графа значком "*" отмечены ребра, по которым произведено разделение, а в скобках приведены данные о количестве объектов в соответствующем элементарном таксоне.



В итоге получено следующее распределение исходных 66 объектов по 4-м таксонам: в первый таксон попала 61 точка, в 4-й - 3 точки (37, 41 и 42), точки 25 и 29 не объединились ни с кем и каждая представляет самостоятельный таксон.

6.1.5. Алгоритм JOINT.

Рассмотренные выше методы таксономии были предназначены для группировки независимо заданных объектов. Однако довольно часто бывает уже известно, или просто предполагается, что некоторые объекты взаимосвязаны и их не следовало бы распределять по разным таксонам. Для таких задач таксономии с предварительной информацией о принадлежности некоторых объектов к одному и тому же таксону и предназначен алгоритм JOINT.

Вначале вручную или по любой программе таксономии формируется K таксонов произвольной формы, состав которых удовлетворяет отмеченному выше требованию. Вектор распределения объектов по этим таксонам задается в файле WORK.PET. Затем производится объединение K таксонов в заданное число K1 более крупных таксонов (K1 < K). С этой целью строится КНП внутри каждого мелкого таксона, а затем КНП между ними - путем соединения ребрами ближайших точек разных таксонов. Вычисляется критерий качества F этой исходной таксономии. Затем поочередно объединяются все пары смежных таксонов (соседних по КНП). При каждом (i-м) объединении вычисляется величина критерия F(i). После проведения всех вариантов объединения выбирается тот из них, который дает $F(i)_{max}$. Такая процедура объединения смежных пар повторяется до получения требуемого числа таксонов K1.

На каждом шаге алгоритм JOINT гарантирует наилучший вариант уменьшения числа таксонов на единицу, но не гарантирует получения глобально оптимального варианта объединения K мелких таксонов в K1 более крупных. При очередном укрупнении программа сравнивает новое значение критерия качества с полученным на предыдущем шаге и информирует пользователя, если улучшения не происходит. Но укрупнение будет продолжаться до тех пор, пока не будет достигнуто требуемое число таксонов K1. Информация о поведении критерия в процессе укрупнения таксонов дает представление о качестве окончательного решения. Выбирая режим пе-

чати, можно получить информацию об итоговых таксонах, о каждом шаге объединения таксонов и о КНП внутри исходных таксонов и между ними.

Пример решения по программе JOINT.

В процессе диалога вам будут заданы следующие вопросы:

```
константа нормировки: 0 - нет, 1 - к [0,2], 2 - по дисперсиям
2
константа печати (0 - 3)
3
учитывать ли равномерность распред. объектов: 0 - нет, 1 - да
0
желательное количество крупных таксонов
4
```

Рассмотрим результат решения для тех же исходных данных, что и во всех предыдущих случаях. В качестве исходного разбиения примем результат программы FOREL2 при делении на 16 таксонов за 7 итераций, приведенный при рассмотрении предыдущего примера. Программа JOINT при печати входных условий информирует и о заданном предварительном разбиении:

ВЕКТОР РАСПРЕДЕЛЕНИЯ ОБ'ЕКТОВ ПО ТАКСОНАМ

На месте, соответствующем порядковому номеру объекта, ставится номер таксона, которому он принадлежит

** 1 **	2 **	3 **	4 **	5 **	6 **	7 **	8 **	9 **	10
1	1	1	9	1	1	1	1	1	1
1	1	10	1	13	1	2	1	1	1
1	1	2	2	6	2	2	2	7	1
1	1	1	2	3	1	16	15	2	11
12	12	11	4	5	1	4	1	4	1
1	1	1	1	8	4	1	1	3	3
3	1	1	14	3	2				

Исходное количество объектов в таксонах

** 1**	2**	3**	4**	5**	6**	7**	8**	9**	10**	1**	2**	3**	4**	5**	6
34	9	5	4	1	1	1	1	1	1	2	2	1	1	1	1

Далее будет документирован процесс последовательного ук - рупнения. Программа будет работать до тех пор, пока не получит требуемый результат, но по ходу решения будет информировать вас о том, как ведет себя при этом функция качества на каждом шаге.

Критерий F	Номера объединяемых таксонов		Полученному таксону присвоен номер
5.5670	9	12	9
6.0275	1	2	1
Дальнейшее объединение не приводит к улучшению критерия			
5.9461	1	4	1
Дальнейшее объединение не приводит к улучшению критерия			
5.2261	3	5	3
Дальнейшее объединение не приводит к улучшению критерия			
4.4373	1	15	1
4.6458	1	8	1
7.0243	1	3	1
7.2152	1	9	1
Дальнейшее объединение не приводит к улучшению критерия			
6.3329	1	10	1
Дальнейшее объединение не приводит к улучшению критерия			
5.6549	1	11	1
5.6630	1	13	1
5.8539	1	14	1

Закончив решение, программа выдает все необходимые сведения, в том числе строит график значений критерия F, полученных на каждом шаге объединения таксонов, и вектор распределения объектов по таксонам в двух вариантах - в исходных и новых номерах таксонов. Приведем здесь одну из выдач:

ИТОГОВЫЙ ВЕКТОР РАСПРЕДЕЛЕНИЯ ОБ'ЕКТОВ ПО ТАКСОНАМ
(В ИСХОДНЫХ НОМЕРАХ ТАКСОНОВ)

** 1	** 2	** 3	** 4	** 5	** 6	** 7	** 8	** 9	** 0
1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1
21	1	1	1	1	6	1	1	1	7	1

31	1	1	1	1	1	1	16	1	1	1	
41	1	1	1	1	1	1	1	1	1	1	
51	1	1	1	1	1	1	1	1	1	1	
61	1	1	1	1	1	1					

6.2. Сравнение и отбор объектов по заданным условиям. (Авторы алгоритмов - Ёлкина В.Н., авторы программ Киприянова Т.П., Кулакова Л.Г., Ефанова Н.В.). В этой группе программ имеются следующие программы: SIM (поиск аналогов), SELEKT (отбор объектов, удовлетворяющих заданным условиям) и DIFER (сравнение "образцового" объекта с остальными). Вход в программы данной группы осуществляется из основного меню выбором строки "Поиск аналогов; отбор объектов по заданным условиям".

6.2.1. Программа SIM.

Программа дает возможность:

- отобрать заданное количество объектов (аналогов), ближайших к указанному объекту-образцу;

- получить информацию о степени сходства аналогов с образцом;

- сравнить значения параметров образца со значениями тех же параметров у аналогов и со средними значениями для группы найденных аналогов;

- получить характеристику распределения расстояний от образцов до аналогов по 10 интервалам относительно max и min значений и относительно среднего значения;

- распечатать для каждой группы ОБРАЗЕЦ-АНАЛОГИ "паспорт" группы: информацию об интересующих дополнительных параметрах, в том числе и не участвовавших при выборе аналогов;

- оценить, как расположен образец относительно аналогов в данном признаковом пространстве - "внутри" группы или "вне" ее.

Для обращения к программе необходимо задать 8 параметров:

1 - количество заданных строк-образцов,
2 - номера строк-образцов,
3 - требуемое количество аналогов,
4 - режим печати результатов счета (0-8),
5 - количество признаков, учитываемых при поиске аналогов,
6 - номера учитываемых признаков,
7 - количество "дополнительных" признаков,
8 - номера "дополнительных" признаков.

Номера в пп.2,6,8 указываются через запятую или пробел.

Все возможные варианты меню, предлагаемые системой для программы SIM, могут быть получены из основного списка с помощью приведенной ниже таблицы. В столбце "нет вопросов" указаны номера вопросов, отсутствующих при данных значениях параметров.

Параметр	Значения	Нет вопросов
1	0	2
4	0 1 2 3 6	7 8
5	0	6
7	0	8

Программа имеет много возможных вариантов представления информации для печати. Нужный нам вариант можно выбрать с помощью приведенной ниже таблицы.

Сообщение	Режим печати
Таблица номеров строк-аналогов и их расстояния до строки-образца	1,2
Оценка отклонения параметров образцов от средних по группе аналогов	2

Характеристика распределения расстояний от образцов до из аналогов	3-8
Характеристика образца, аналогов и средних по группе аналогов	3,5
То же для дополнительных параметров	4,5
Характеристика образца, отклонение параметров аналогов и их среднего от параметров образца	6,8
Характеристика образца по дополнительным параметрам, отклонение дополнительных параметров аналогов и их среднего от дополнительных параметров образца	7,8
Без печати	0

Примеры решений по программе SIM.

Исходные данные представлены в таблице (см.приложение). В начале работы программа печатает необходимые информационные сведения об условиях решения.В данном случае было использовано полное описание объектов - все 14 признаков, поэтому верхняя и нижняя строки "Вектора соответствия нумерации признаков" совпадают.

Программа SIM - поиск аналогов

Режим печати 3. Количество: признаков - 14, объектов - 66, объектов-образцов - 2,аналогов - 3.Обозначение пробела .8E+18. Сходство объектов вычисляется по 14 признакам.

Вектор соответствия нумерации признаков:

верхняя строка - порядковый номер признака в программе, нижняя строка - номер признака в исходном массиве

```
 1   2   3   4   5   6   7   8   9   10   11   12   13   14
**************************************************************
 1   2   3   4   5   6   7   8   9   10   11   12   13   14
```

Поскольку заказан режим печати 3, то будут даны сведения о том, как распределяются расстояния от заданных образцов до найденных для них аналогов.

Характеристика множества расстояний от образцов до аналогов

средние	дисперсии	max	min
.4819	.1711E-02	.5263	.4251

Распределение объектов по интервалам расстояний от образцов до аналогов (от min до max значений)

min									max
.425	.436	.448	.459	.470	.481	.493	.504	.515	.526
1	0	1	0	0	0	2	0	0	2

То же - в долях от среднего расстояния до найденных аналогов

0.1Xcp	.25Xcp	0.5Xcp	.75Xcp	Xcp	1.25Xcp	1.5Xcp	1.75Xcp	2Xcp
.048	.120	.241	.361	.482	.602	.723	.843	.964
0	0	0	0	2	4	0	0	0

"Таблица ближайших объектов" содержит информацию для каждого объекта-образца в стандартном виде. Ниже мы приводим часть этой таблицы для 1-го объекта-образца. Порядок следования признаков указан согласно вектору соответствия.

Образец 1		Характеристики объекта-образца			
		12.0	11.80	20.00	14.20
		11.9	9.900	14.10	12.20
		15.8	15.10	20.90	19.10
		7.60	13.20		
№ объекта-аналога	Расстояние до заданного образца	Характеристики объектов-аналогов			
9	.487	13.60	13.00	22.00	16.20
		10.60	6.100	16.10	14.90
		14.90	12.50	16.50	19.60
		15.10	14.00		

19	.526	13.40 11.70 13.00 13.20	12.70 7.400 15.60 15.70	19.90 17.10 15.20	14.90 18.00 19.90
18	.526	12.90 11.70 14.80 8.800	17.20 10.10 12.20 13.80	23.50 16.10 17.40	10.90 13.30 12.80
центр объектов-аналогов	.414	13.30 11.33 14.23 12.36	14.30 7.866 13.43 14.50	21.80 16.43 16.36	14.00 15.40 17.43

В режиме печати 1 при тех же входных условиях вы получите только краткую информацию о номерах аналогов и их расстояниях до заданного образца:

Номер заданного образца	Расстояние до центра аналогов	Объекты-аналоги							
		№	расст.	№	расст.	№	расст.	№	расст.
1	.414	9	.487	19	.526	18	.525		

6.2.2. Программа SELECT.

С помощью этой программы можно сделать отбор объектов, характеристики которых удовлетворяют заданным условиям. Для обращения к программе необходимо задать только один параметр - имя файла, содержащего условия для отбора признаков.

Структура файла. В первых четырех строках текст комментариев; далее вводятся через пробел попарно пороговые значения признаков и коды условий от 0 до 6: например,

```
99.7 1  55.2 6  3.0 0  17.5 3  22. 5
```

Кодировка условий.

Код 0 - признак несущественный, в отборе не участвует

Код	1	2	3	4	5	6
Условие	>	<	≥	≤	=	≠

Пороговые значения должны быть заданы для всех признаков! Если признак не участвует в отборе, то на соответствующем ему месте в этом файле нужно поставить произвольное число и код условия 0.

Пример решения по программе SELECT.

Рассматривается таблица: 66 строк, 14 столбцов.

Пусть ОБРАЗЕЦ имеет следующие характеристики:

1	2	3	4	5	6	7
13.60	13.00	22.00	16.20	10.60	6.100	16.10

8	9	10	11	12	13	14
14.90	14.90	12.50	16.50	19.60	15.10	14.00

Заданы следующие условия:

НОМЕР ПРИЗНАКА	1	3	4	7	11	12
ТИП УСЛОВИЯ	>	≥	>	>	≥	>

Программа выдает следующую информацию о результатах решения: номера объектов, удовлетворяющих условиям, и полные сведения о каждом из этих объектов.

ЗАДАННЫМ УСЛОВИЯМ УДОВЛЕТВОРЯЮТ ОБ'ЕКТЫ:

4 13 23 24 37 39 40 41 42 66

Объект номер 4

1	2	3	4	5	6	7
20.80	21.00	40.80	22.00	14.00	19.10	25.20

8	9	10	11	12	13	14
24.20	25.60	22.90	25.30	23.00	13.50	25.00

Объект номер 13

.....

В случае отсутствия в таблице объектов, удовлетворяющих заданным условиям, программа информирует об этом.

6.2.3. Программа DIFER.

Эта программа служит для сравнения объекта-образца с ос - тальными объектами по признакам и вычисления отклонений значений признаков объектов от значений признаков образца. В таблице допускаются пробелы.

Для обращения к программе надо задать 2 параметра:

1 - номер объекта-образца

2 - режим работы программы: 1,2

В качестве объекта-образца выбирается один из объектов таблицы. Программа производит сравнение образца с каждым из оставшихся объектов таблицы.

Режимы работы программы различаются только формой выдачи результата:

В режиме 1 для каждого объекта печатаются значения,а в режиме 2 - отклонения значений тех показателей, которые не сов - падают со значениями показателей объекта-образца, если в них нет пробелов. Для каждого режима дополнительно отдельной строкой выдается перечень номеров показателей, в которых имеются пробелы.

Программа выдает входные данные и сведения об образце.

Входные данные

Число объектов в таблице 66, признаков 14

Кодировка пропуска .0

образец 1

1	2	3	4	5	6	7
12.00	11.80	20.00	14.20	11.90	9.900	14.10
8	9	10	11	12	13	14
12.20	15.80	14.10	20.90	19.10	7.600	13.20

Обратите внимание на то, что программа будет сравнивать с образцом все объекты таблицы (кроме себя самого).

Таблица отклонений параметров объектов-аналогов от параметров образца

(На печать выводится информация только о несовпадающих значениях параметров)

представление информации:
- порядковый номер параметра
- значение параметров образца
- значения параметров объекта

образец 1 объект 2

1	2	3	4	5	6	7
12.00	11.80	20.00	14.20	11.90	9.900	14.10
13.80	15.20	20.80	14.00	8.200	13.00	18.40
8	9	10	11	12	13	14
12.20	15.80	14.10	20.90	19.10	7.600	13.20
15.00	19.10	16.10	21.10	15.50	8.300	14.00

Во втором режиме печати для тех же объектов информация будет выглядеть следующим образом:

.....

(от значений параметров объекта вычитаются значения параметров образца)

.....

представление информации:
- порядковый номер параметра
- значение параметров образца
- разность между параметрами объекта и образца

образец 1 объект 2

1	2	3	4	5	6	7
12.00	11.80	20.00	14.20	11.90	9.900	14.10
1.800	3.400	.8000	-.2000	-3.700	3.100	4.300
8	9	10	11	12	13	14
12.20	15.80	14.10	20.90	19.10	7.600	13.20
2.800	3.300	2.000	.2000	-3.600	.7000	.8000

.....

6.3. Выбор информативных признаков и распознавание образов.

Определяя перечень показателей (признаков), характеризующих объекты, исследователь обычно предполагает, что эти признаки связаны с некоторой целью, которую он задает указанием принадлежности объектов обучающей выборки к тому или иному классу (образу). Но может оказаться, что не все признаки соответствуют подразумеваемой цели, а иногда и вообще среди данного опи -

сания не удается найти ни одного, связанного с этой целью. Если же исходное множество признаков содержит нужную информацию, то может оказаться, что она включает в свой состав лишние признаки,которые можно было бы и не измерять. В связи с этим часто возникают задачи проверки признаков на их информативность и выбора наиболее информативного их подмножества. Заметим, что информативность признаков не существует сама по себе, безотносительно к цели. То, что важно, например, для достижения высоких надоев молока, может не иметь никакого значения для производства других видов продукции.

Информативное подпространство признаков используется в дальнейшем для распознавания новых объектов, т.е. для определения принадлежности к тому или иному образу.

В блоке анализа данных есть несколько методов решения задачи выбора наиболее информативных подсистем признаков и задачи распознавания образов: NTPP, NTPP5, PROVINE, ACQUIS, RASP (RASPP), TRF (TRF2).

6.3.1. Алгоритм NTPP (Направленного Таксономического Поиска Признаков) предназначен для выбора из заданного набора признаков требуемого числа наиболее информативных. В БАД алгоритм NTPP реализован двумя программами - NTPP для признаков, измеренных в сильных шкалах, и NTPP5 - для бинарных признаков.

Для решения этой задачи вы должны указать целевой признак - вектор принадлежности каждого объекта таблицы к одному из известных вам классов. То, каким образом была установлена эта принадлежность, не имеет значения. Можно задать классы на основе какого-либо одного результативного показателя, например, по уровню урожайности или производительности труда. Можно по комплексу одних характеристик, результативных, выполнить

предварительно таксономию и задать в качестве целевого показателя вектор распределения объектов по полученным таксонам. С таким же правом классы могут быть заданы из теоретических соображений, согласно некоторой гипотезе пользователя. Алгоритм в любом из этих случаев даст ответ на вопрос: существует ли среди указанных исходных показателей такая их подсистема, которая обеспечила бы разделение объектов по заданным классам с достаточной надежностью.

Понятно, что проверить информативность всех подсистем можно было бы их полным перебором. К сожалению, полный перебор слишком трудоемок и при сравнительно небольших размерностях уже практически нереализуем. Мы пошли по пути предварительного сокращения исходного признакового пространства, пытаясь с возможно меньшей потерей информации выбрать такое количество признаков, при котором полный перебор всех подсистем уже станет возможным.

В алгоритме NTPP существенно используются методы таксономии, в частности, алгоритм FOREL. По аналогии с группировкой объектов, одинаково "ведущих себя" в признаковом пространстве, ставится задача объединения признаков (столбцов таблицы), которые примерно одинаково проявляют себя на данных объектах. Поскольку задача группировки признаков представляет и самостоятельный интерес для анализа структуры признакового пространства, то предусмотрен режим работы программы, при котором делается только группировка признаков без дальнейшего выбора наиболее информативной подсистемы.

В полной программе NTPP вначале производится таксономия всех исходных признаков по алгоритму FOREL. Тем самым определяются группы "наиболее похожих" свойств. Далее, из каждого таксона выбирается по одному "типичному" (ближайшему к среднему) представителю и тестируются различные подсистемы из типичных признаков. Лучшей подсистемой считается та, при которой полу -

чается наилучшее распознавание обучающей выборки по линейному решающему правилу: объект считается принадлежащим тому образу, к центру которого он ближе. Количество таксонов для предвари - тельной группировки признаков задается пользователем.

Даже при сокращении исходного количества признаков задача перебора остается достаточно трудоемкой и не всегда может быть выполнена за один раз. Поэтому в программе предусмотрен ряд возможностей для получения информации об уже найденных лучших вариантах на момент прерывания и для продолжения решения с точки прерывания. Информация для продолжения работы программы сохраняется в файле PRODOL.DAT.В случае прерывания работы программы рекомендуется запомнить этот файл под другим именем и восстановить его под именем PRODOL.DAT для возобновления работы программы.

Информация о нескольких "лучших" сочетаниях признаков,полученных к данному моменту, выдается периодически с заданным пользователем интервалом. Выбирать величину интервала нужно так, чтобы не вызвать потока заведомо избыточных сведений,но и не повторять слишком большую часть вычислений в случае прерывания решения и возобновления перебора признаков после перерыва. Период для промежуточной печати нужно задавать обязательно больше требуемого количества лучших сочетаний.

Желательное количество (К) признаков в информативной подсистеме задается пользователем. Можно указать пределы K_1-K_2 изменения длины проверяемых сочетаний ($K_1 < K < K_2$). В последнем случае программа начнет перебор с подсистем длины K_1 и остановится, только закончив просмотр подсистем длины K_2.

Уже было сказано, что для решения задачи оценки информа - тивности необходимо указать целевой признак. В программе NTPP в качестве целевого показателя должен быть задан вектор рас - пределения объектов по классам (образам) или вектор количества объектов в образах, если обучающая выборка упорядочена по клас-

сам. Для работы с программой NTPP дополнительно необходимо наличие следующих параметров в файле WORK.PET (см. "Стандартный образец файла"):

18* вектор распределения объектов по классам (если в п.9* указан тип -1);

или 16* значения вектора количества объектов в образах (если в п.9* указан тип 0 и в п.12* указан индекс =1);

Пункты 13* и 14* присутствуют только в том случае, если в п.11* указан индекс =0 (признаки разных типов).

Режимы печати:

- итоговая информация, информация для продолжения работы программы и периодические сведения о лучших сочетаниях признаков;

- выдается также и входная информация;

- выдается также информация о качестве распознавания для каждого найденного сочетания признаков.

Пример решения по программе NTPP.

Рассмотрим тот же пример, что и в предыдущих случаях: таблицу 66 на 14 (см. приложение), которую нужно поместить в файл WORK.DAT.

Создадим целевой вектор в виде классов, в которые попадают регионы по урожайности последнего года. Классы установим следующим образом: урожайность до 10 центнеров - класс 1, далее - через каждые 5 центнеров. Всего получим 5 классов. Вектор распределения объектов по этим классам поместим в файл WORK.PET после строки ОТДЕЛЬНЫЙ ЦЕЛЕВОЙ ПРИЗНАК. Этот файл WORK.PET, не считая первых 4-х строк, отведенных для комментариев, будет выглядеть следующим образом:

```
      66   количество объектов
      14   количество признаков
       1   тип записи (1 - по объектам)
    9990   обозначение пропуска
      -1   тип задания целевого признака (задан в .PET):
```

```
:          5   количество образов                              :
:          4   все признаки в сильной шкале                    :
           0   индекс упорядочения по образам (0 - нет)
       Отдельный целевой признак
2 2 1 4 2 2 3 3 2 2 3 2 5 4 4 2 3 2 3 1 2 3 3 5 5 5 4
4 3 3 3 2 2 3 4 2 5 5 5 4 5 5 3 1 2 1 2 1 2 1 3 3 2 2
3 1 3 4 2 2 1 3 3 3 3 5
```

При обращении к программе NTPP вам будут заданы вопросы о конкретном варианте решения. Ниже приведены эти вопросы и ответы на них под соответствующей строкой. Как всегда, нажатие клавиши F1 обеспечит вам получение справочных сведений, относящихся к рассматриваемой в данный момент строке.

```
Желательное количество "типичных"  признаков
 14
Требуемое количество лучших сочетаний признаков
 3
Граничные размерности рассматриваемых подсистем признаков
 1 3
Константа перебора после перерыва (см. "Комментарии")
 0
Период для промежуточной печати при переборе подсистем
 100
Режим печати: 0 - 2
 1
Учет обратных зависимостей при группир.призн.: 1 - да,0 - нет
 0
```

После ответа на вопросы о конкретных значениях параметров решения и запуска программы (нажатием клавиши F5), вы увидите на экране информацию о ходе решения. Вся эта информация после окончания решения будет находиться в файле TABL.DAT,может быть сохранена под другим именем или вызвана в редактор для предварительного просмотра и отбора интересующей вас информации.

Программа прежде всего повторит заданные вами условия решения - входные параметры и вектор распределения объектов по образам и сообщит о том, сколько сочетаний она должна будет проверить для выполнения требуемых условий.

ВХОДНЫЕ ПАРАМЕТРЫ

Количество: признаков - 14, объектов - 66, образов - 5, "типичных" признаков - 14, лучших сочетаний - 3, границы длины сочетаний 1-3.

Информация выводится на печать через 100 сочетаний

Константа печати - 1

Группировка признаков без учета обратных зависимостей

ИСХОДНОЕ РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО ОБРАЗАМ

***** Всего будет обработано 469 сочетаний признаков, начиная с сочетания №1 и кончая сочетанием № 469

Так как существуют различные версии программ, то программа напомнит некоторые основные сведения, полезные для пользователя, чтобы избавить его от необходимости запоминания излишней информации:

******* ИНФОРМАЦИЯ К РЕШАЮЩЕМУ ПРАВИЛУ *******

РЕШАЮЩЕЕ ПРАВИЛО: Объект принадлежит тому образу, евклидово расстояние до центра которого минимально и не превышает порогового значения (радиуса). Евклидово расстояние вычисляется в нормированном пространстве.

Для нормировки координаты РАСПОЗНАВАЕМЫХ объектов следует умножить на соответствующие нормирующие коэффициенты.

Исходные номера признаков:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	

Нормирующие коэффициенты:

.1995	.2386	.1193	.2052	.3037
.1705	.1744	.1955	.1572	.2018
.1599	.1468	.1520	.1407	

ОШИБКА 1-го РОДА - отношение количества объектов данного образа, попавших при распознавании в другие, к исходному количеству объектов в образе.

ОШИБКА 2-го РОДА - отношение количества "чужих" объектов к общему числу попавших в данный образ при распознавании.

Повторим, что по ходу решения программа будет информировать об уже найденных лучших вариантах и полученном при этом числе ошибок, о количестве просмотренных сочетаний и о том, что нужно знать для продолжения решения с точки прерывания. Информация для продолжения работы программы сохраняется в файле PRODOL.DAT. В случае прерывания работы программы рекомендуется запомнить этот файл под другим именем и восстановить его под именем PRODOL.DAT при возобновлении работы программы.

Информация о нескольких "лучших" сочетаниях признаков, полученных к данному моменту, выдается периодически, с заданным пользователем интервалом, а также после окончания перебора сочетаний одной длины.

```
                ПРОМЕЖУТОЧНАЯ ИНФОРМАЦИЯ

 *** Просмотрено 14 сочетаний ***

 *** ЗАКОНЧЕН ПЕРЕБОР по 1 признаку ***

   Номер          Количество          Исходные номера
  сочетания       несовпадений           признаков

      14               2                 14
      27              13                  1     14
      77              13                  6     14

 Количество вхождений признаков в лучшие сочетания

 № признаков   1   6   14
 Кол. вхожд.   1   1    3

         ИНФОРМАЦИЯ ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ ПРОГРАММЫ

 Сбработано 101 сочетание признаков.
 Задавать  IKS#0 при возобновлении работы.

 Лучших сочетаний с количеством ошибок = 2 получено 1
```

```
   *** Просмотрено 105 сочетаний ***
   *** ЗАКОНЧЕН ПЕРЕБОР по 2 признака ***
         ИНФОРМАЦИЯ ДЛЯ ПРОДОЛЖЕНИЯ РАБОТЫ ПРОГРАММЫ
 Обработано 202 сочетания признаков
 Задавать IKS # 0  при возобновлении работы
 Лучших сочетаний с количеством ошибок = 2 получено 1

   *** Просмотрено 469 сочетаний ***
   *** ЗАКОНЧЕН ПЕРЕБОР по 3 признака ***
 Лучших сочетаний с количеством ошибок = 2 получено 1
```

Для каждого из найденных лучших сочетаний будет выдана полная итоговая информация. Здесь мы приведем выдачу только для одного из них. Анализ этой информации позволит вам полу - чить полное представление о качестве и ценности для ваших целей каждой подсистемы признаков.

```
******** ИТОГОВАЯ  ИНФОРМАЦИЯ ********

   Номер          Процент          Исходные номера
 сочетания        ошибок             признаков

     14            3.030                14

        ОБ'ЕКТЫ, ПОПАДАЮЩИЕ В ЧУЖОЙ ОБРАЗ

 N объектов          11              54
 N образов            2              11
 N исх. обр.          3               2
```

Первая часть итоговой информации указывает, что лучшая подсистема, состоящая из одного признака, включает в свой состав признак номер 14.

Напомним, что для данной версии программы NTPP принято следующее решающее правило: объект считается принадлежащим тому образу, евклидово расстояние до центра которого минимально

и не превышает заданного порогового значения (радиуса). Евклидово расстояние вычисляется в нормированном пространстве.

При использовании полученного для признака 14 решающего правила для распознавания новых объектов нужно предварительно выполнить предписания из раздела ИНФОРМАЦИЯ К РЕШАЮЩЕМУ ПРАВИЛУ: "Для нормировки координат РАСПОЗНАВАЕМЫХ объектов следует умножить их на соответствующие нормирующие коэффициенты". Признаку 14 соответствует нормирующий коэффициент .1407. Результаты распознавания обучающей выборки по этому решающему правилу приводятся в следующих таблицах. ОШИБКА 1-го РОДА - это отношение количества объектов данного образа, попавших при распознавании в другие, к исходному количеству объектов в образе, а ОШИБКА 2-го РОДА - отношение количества "чужих" объектов к общему числу попавших в данный образ при распознавании. Из таблицы видно, что ни один объект 1-го образа не был отнесен к другому классу, но в него зато попал чужой объект. Образы 4 и 5 распознаны безошибочно, 2-й образ имеет свои объекты в других и чужие у себя и т.д.

********* РЕШАЮЩЕЕ ПРАВИЛО *********

Исходные номера признаков: 14

Номер образа	Радиус	Ошибка		Центр образа
		2-го рода	1-го рода	
1	.7210E-01	11.11	.0000	1.183
2	.1315	5.000	5.000	1.848
3	.1998	.0000	5.000	2.422
4	.1248	.0000	.0000	3.107
5	.3432	.0000	.0000	4.367

Представление о судьбе "чужих" и "своих" объектов соответственно в "своих" и "чужих" образах дает следующая таблица. Это позволяет конкретнее понять результат решения: к 1-му

образу был отнесен 1 из 20 объектов 2-го образа, а ко 2-му - 1 объект (также из 20) 3-го образа.

РЕЗУЛЬТАТ РАСПОЗНАВАНИЯ:

Распределение числа объектов по образам

Номер образа	Исходное количество	Номера образов				
		1	2	3	4	5
1	8	8	0	0	0	0
2	20	1	19	0	0	0
3	20	0	1	19	0	0
4	8	0	0	0	8	0
5	10	0	0	0	0	10

Если вас интересует, какие конкретно объекты попали не в свои классы, то следует обратиться к началу итоговой информа - ции, где эти сведения имеются. В данном случае это объекты 11 и 54. Если вы затребовали печать в последнем, наиболее полном режиме, то в таблице "Промежуточная информация" будет присутствовать строка сведений о качестве распознавания для каждого найденного сочетания признаков.

ПРОМЕЖУТОЧНАЯ ИНФОРМАЦИЯ

Номер сочетания	Количество ошибок	Исходные номера признаков
.....	.....	.....
.....	.....	.....

Рассмотренное выше решение выполнялось при требовании предварительного поиска 14 типичных признаков, в результате чего был осуществлен полный перебор подсистем по 1, по 2 и по 3 признака из всего исходного их набора. Теперь продемонстрируем решение задачи при условии выбора 7 типичных признаков и полного перебора всех возможных подсистем из этих 7.

Основная вводная информация об условиях задачи остается той же, что и в предыдущем случае. Но так как теперь требуется отбор 7 типичных признаков, то появляется следующий фрагмент сообщений о выполняемых действиях:

```
ГРУППИРОВКА ПРИЗНАКОВ ВЫПОЛНЯЕТСЯ ПРОГРАММОЙ FOREL2

РАСПРЕДЕЛЕНИЕ ПРИЗНАКОВ ПО ТАКСОНАМ

 **  1 **  2 **  3 **  4 **  5 **  6 **  7 **  8 **  9 **  0 **
-------------------------------------------------------------------
 1-  -1    -5    -4    -3    -6     1     1     1     1     1
11-   1     2     2    -2
-------------------------------------------------------------------

                    ПРОГРАММА TIP

                --- ВЕКТОР СООТВЕТСТВИЯ ---

  НОВЫЕ НОМЕРА                1   2   3   4   5   6
  ИСХОДНЫЕ НОМЕРА ОБ'ЕКТОВ    1   2   3   4   5  14
  ИСХОДНЫЕ НОМЕРА ТАКСОНОВ    1   5   4   3   6   2
  ВЕКТОР ВЕСОВ                7   1   1   1   1   3

*** Всего будет обработано 63 сочетания признаков, начи-
ная с сочетания N1 и кончая сочетанием  N63
```

Программа смогла объединить 14 исходных признаков в 6 таксонов (лучший вариант при числе таксонов, ближайшем к 7) и соответственно выбрать 6 типичных признаков: 1, 2, 3, 4, 5 и 14. Всего из 6 номеров может быть получено 63 сочетания признаков. После проверки 63 возможностей получены следующие результаты:

```
******** ИТОГОВАЯ  ИНФОРМАЦИЯ ********

 Номер        Процент      Исходные номера
сочетания     ошибок          признаков

    6          3.030                14
   11         19.697             1  14
   15         25.758             2  14
```

Первые два результата полностью совпадают с тем, что было найдено полным перебором исходных признаков. Третий результат несколько слабее (признак 6 не попал в число типичных, поэтому сочетание 6, 14 не проверялось). При втором варианте решения число всех возможных сочетаний было 63, тогда как сочетаний из 14 только по 1, 2 и 3 уже 469.

6.3.2. Программа PROVINF.

Мы уже говорили о том, что в БАД системы ЭКСНА есть несколько программ, реализующих различные алгоритмы решения проблемы выбора наиболее информативных подсистем признаков. Эти программы оценивают информативность подсистем признаков "с разных точек зрения", т.е. проверяют выполнение условий по разным решающим правилам. Сейчас мы рассмотрим программу PROVINF. Эта программа выполняет простую, но очень полезную процедуру - проверяет, нет ли возможности различить заданные классы только по граничным в пределах класса значениям каждого из признаков - по их максимумам и минимумам. Если таких признаков нет,то тогда можно приступать к проверке информативности признаков по другим критериям.

Как и для программы NTPP, в файле WORK.PET должен быть подготовлен вектор распределения объектов по классам. Программа PROVINF анализирует данные, определяет по каждому признаку **max** и **min**, а затем проверяет степень пересечения классов. Программа производит подсчет количества ошибок 2-го рода при распознавании (отношение количества "чужих" к исходному количеству объектов в классе) по каждому признаку.

Режимы печати:

1 - печать итогового результата (суммарное отношение количества "чужих" объектов к исходному количеству объектов в классе);

2 - дополнительно к режиму 1 печатается "решающее правило";

3 - дополнительно к режиму 2 печатается количество ошибок 2-го рода при распознавании по каждому признаку ("чужие" объекты внутри "своих" интервалов).

Пример решения по программе PROVINF.

Программа напоминает значения входных параметров и заданное вами распределение объектов по таксонам. Затем для каждого признака будет дана информация о том, какое применяется решающее правило, и для каждого признака - значение его максимумов и минимумов в заданных классах. Обратите внимание на то, что в данной программе решение о принадлежности объекта к таксону принимается по попаданию в отрезок, определенный экстремальными значениями, в то время как в программе NTPP - по расстоянию до центра тяжести таксона.

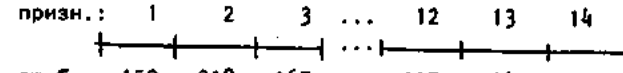
Решающее правило: $(X_{min} < X_i < X_{max})$

Образ	Признак 1	
	min	max
1	4.90	17.30
2	6.20	18.90
3	7.30	21.50
4	12.10	20.80
5	17.50	33.60

Признак 14	
min	max
6.90	10.00
10.10	15.00
15.10	19.60
20.20	25.00
27.10	41.00

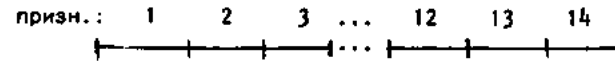
Эти таблицы дают возможность быстро определить, перекрываются ли интервалы значений признака для разных классов, и если да, то какие и насколько. Так, у 14-го признака ни один интервал не перекрыт, ни один максимум не больше чужого минимума. При больших размерностях внимательный просмотр и анализ данных по каждому признаку может быть достаточно утомительным. Поэтому предусмотрена также следующая обобщающая информация для каждого признака и образа:

Количество ошибок 2-го рода при распознавании по каждому признаку ("чужих" объектов внутри "своих" интервалов)



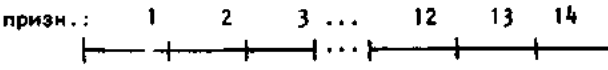
Отношение количества "чужих" объектов к исходному (%)

Образ 1

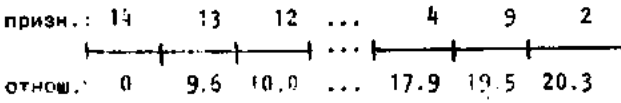


В завершение можно получить окончательные результаты, расположенные для удобства пользования как в порядке возрастания номеров признаков, так и по возрастанию значения ошибок, т.е. по убыванию "качества". Если среди проанализированных данных есть признаки, по которым разделение классов происходит безошибочно, программа дополнительно проинформирует вас об этом.

Суммарное отношение числа "чужих" объектов к исходному



Суммарное отношение количества "чужих" объектов к исходному, упорядоченное по возрастанию



По признаку 14 разбиение объектов на классы происходит без ошибок!

6.4. Распознавание образов.

В этом параграфе описаны алгоритмы и программы для анализа данных в такой ситуации: выбрана информативная система признаков, задано разбиение объектов обучающей выборки на классы (образы) и предъявлен некоторый новый объект или несколько объектов, принадлежность которых к этим классам необходимо установить. Процедура установления принадлежности объектов к образам и составляет предмет широко известной задачи распознавания образов.

Обычно решение задачи распознавания делится на два этапа: "обучение" и "принятие решения". На этапе обучения строится решающая функция, параметры которой определяются и жестко фиксируются по информации, содержащейся в обучающей таблице. Напомним, что обучающие объекты - это объекты, про которые заранее известно, каким образам (классам) они принадлежат. Какие бы контрольные объекты не предъявлялись впоследствии, сама решающая функция не изменяется.

Но возможен и другой подход: строить решающую функцию не заблаговременно, а прямо в процессе принятия решения, используя при этом одновременно информацию об обучающей и контрольной выборках. Построенная таким образом решающая функция будет более устойчивой по отношению к помехам, возникающим из-за непредставительности обучающей выборки. В БАД системы ЭКСНА имеются программы, реализующие оба этих подхода.

6.4.1. Программы RASP (RASPP).

Программа RASP предназначена для распознавания объектов по следующему решающему правилу: образы заданы координатами центров и их радиусами и упорядочены по возрастанию радиусов. Распознавание начинается с проверки попадания объекта в образ с наименьшим радиусом: если расстояние до центра образа меньше его радиуса, то объект считается принадлежащим рассматриваемо-

му образу; если больше, то вычисляется расстояние до центра следующего по порядку образа и т.д., пока не будут просмотрены все образы. Объекты, не попавшие при этом ни в один образ, отмечаются звездочками и указаниями номера образа, расстояние до центра которого было минимальным.

Решающее правило программы RASPP: объект считается принадлежащим тому образу, расстояние до центра которого минимально. Результаты распознавания по этим двум программам могут не совпадать.

Для работы программ необходимо иметь две таблицы (с одинаковым числом и порядком следования признаков!) - обучающую и контрольную. По обучающей таблице будут получены решающие правила и использованы для распознавания объектов второй (контрольной) таблицы.

Обратите внимание на то, что для работы программ необходимо, чтобы в файлах WORK.DAT и WORK.PET находились соответственно контрольная таблица и описание ее параметров. Обучающая таблица с данными и описание ее параметров должны храниться в файлах с расширением .DAT и .PET соответственно, но с именем, отличным от WORK.

Для обучающей таблицы должен быть создан файл ее описания, в котором записан вектор распределения объектов по образам (может быть получен при помощи одного из алгоритмов таксономии или набран вручную в редакторе текстов). В файле с расширением .PET обучающей таблицы должны содержаться следующие данные: номер целевого признака, количество образов, индекс упорядочения по образам и отдельный целевой признак.

Приведем перечень вопросов, которые могут быть заданы системой (в зависимости от конкретной ситуации).

```
Имя файла с данными обучающей таблицы (с расширением .DAT)
Имя файла параметров обучающей таблицы (с расширением .PET)
Способ задания номеров значимых признаков обучающей таблицы
Количество значимых признаков
Имя файла с номерами значимых признаков
          (если указан способ 1)
Номера значимых признаков (вводить через пробел)
          (если указан способ 2)
Режим печати (0-2)
Константа,задающая обращение к программам: 1 - RASP,2 -RASPP
```

Режимы печати:

0 - нет печати;

1 - выдается итоговый результат распознавания;

2 - выдается также решающее правило.

Способ задания номеров значимых признаков обучающей таблицы:

0 - все признаки обучающей таблицы являются значимыми, их количество должно совпадать с количеством признаков распознаваемой (контрольной) таблицы;

1 - номера значимых признаков должны быть заданы в отдельном файле и разделены пробелами;

2 - номера значимых признаков должны быть введены с терминала через пробел (длина строки не более 72 символов).

6.4.2. Пример решения по программам RASP и RASPP.

Чтобы программы RASP и RASPP могли работать автономно,вне зависимости от результатов других вспомогательных программ, перед программами RASP и RASPP работает всегда программа DORASP. Она по заданному вектору распределения объектов вычисляет нормирующие коэффициенты, координаты центров образов и их радиусы в заданном признаковом пространстве, упорядочивает радиусы по возрастанию и соответственно меняет порядок следования координат центров образов.Другими словами,программа DORASP заново строит решающее правило, подготавливает условия для работы программ RASP и RASPP и сообщает, как обычно, всю необходимую информацию о типе нормировки, нормирующих коэффициентах,

о векторе распределения объектов обучающей выборки по классам и т.д.

Рассмотрим все тот же пример (см. приложение), но отберем для решения из исходного набора только 3 признака - 1, 6 и 14. Программа DORASP в этом трехмерном пространстве получила сле - дующее решающее правило:

ПРОГРАММА DORASP

Вектор распределения объектов обучающей выборки по образам

```
**1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
 2  2  1  4  2  2  3  3  2  2  3  2  5  4  4  2  3  2  3  1
 2  3  3  5  5  5  4  4  3  3  3  2  2  3  4  2  5  5  5  4
 5  5  3  1  2  1  2  1  2  1  3  3  2  2  3  1  3  4  2  2
 1  3  3  3  3  5
```

РЕШАЮЩЕЕ ПРАВИЛО

Номер	Радиус	Координаты центров образов		
3	1.926	2.726	3.326	2.422
1	2.327	2.459	2.961	1.183
4	2.505	2.981	2.833	3.107
2	2.517	2.527	3.005	1.848
5	2.579	4.577	4.128	4.367

Для проверки качества обучения контрольной выборки служат те же 66 объектов, что использовались и при обучении. Программа RASP должна по полученному правилу отнести каждый из 66 объектов к одному из 5 образов. Напомним, что распознавание начинается с проверки попадания объекта в образ с минимальным радиусом: если расстояние до центра образа меньше его радиуса, то объект считается принадлежащим рассматриваемому образу; если больше, то вычисляется расстояние до центра следующего по порядку образа и т.д., пока не будут просмотрены все образы.

ПРОГРАММА RASP

ВЕКТОР РАСПРЕДЕЛЕНИЯ ОБ'ЕКТОВ ПО ОБРАЗАМ:

На месте, соответствующем порядковому номеру объекта,ставится номер образа, к которому он отнесен. Знаком '-' отмечены объекты, отнесенные к образам, расстояние до центра которых хотя и минимально, но превышает значение соответствующего радиуса.

```
**1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
  3  3  3  4  3  3  3  3  3  3  3  3  5  3  3  3  3  3  3  1
  3  3  3  4  3  3  3  3  3  3  3  3  3  3  3  3  5  4  3  4
  5  5  3  1  3  3 -3  3  3  3  3  3  3  3  3  3  3  3  1  1
  1  3  3  3  3  4
```

По программе RASPP распознаваемый объект будет отнесен к образу, расстояние до центра которого минимально. Результаты программ RASP и RASPP могут не совпадать, в чем вы можете легко убедиться, сравнив решения, полученные для одних и тех же данных.

ПРОГРАММА RASPP

ВЕКТОР РАСПРЕДЕЛЕНИЯ ОБ'ЕКТОВ ПО ТАКСОНАМ

На месте, соответствующем порядковому номеру объекта,ставится номер образа, к которому он отнесен

```
**1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
  2  3  1  5  2  2  3  3  2  2  2  1  5  4  4  2  4  3  3  1
  3  3  3  5  5  4  4  4  2  4  3  2  2  3  4  3  5  4  4  4
  5  5  4  1  2  1  3  1  3  1  3  3  2  2  3  3  3  4  2  1
  1  3  3  3  2  5
```

6.4.3. Алгоритм TRF (Таксономические Решающие Функции).

По алгоритму TRF решение о принадлежности рассматриваемых объектов принимается на основании таксономического критерия,вы-

числяемого в процессе распознавания. Обучающая и контрольная выборки совместно подвергаются таксономии, и контрольный объект относится к тому таксону, в структуру которого он лучше вписывается, где максимально значение критерия качества таксономии.

Основные этапы алгоритма TRF1.

Точки обучающей выборки каждого образа соединяются кратчайшим незамкнутым путем (КНП). Полученные графы по тому же принципу соединяются в единый незамкнутый граф отрезками, проведенными между ближайшими точками разных образов. Вычисляется начальное значение критерия качества таксономии F_0. Вычисление критерия качества таксономии возможно как с учетом равномерности распределения количества объектов по таксонам, так и без учета.

Для распознавания контрольный объект поочередно присоединяется к каждому из К образов и при этом вычисляются критерии F_i ($i = 1,\ldots,K$). Если разрешено выделять новые таксоны, то проверяется и (К+1)-й вариант, в котором рассматриваемый объект считается самостоятельным таксоном. В качестве решения будет выбран тот вариант, где было получено максимальное значение функции качества: $F_{max} = \max\{F_i\}$.

Поскольку описанная выше процедура принятия решений достаточно трудоемка, то в существующей версии программы организовано сокращение лишних вычислений следующим образом. Вначале для рассматриваемого объекта определяются несколько (в данном варианте - 5) "ближайших соседей" из объектов обучающей выборки. Если все они принадлежат одному образу, то и распознаваемый объект считается принадлежащим тому же образу. Дополнительно проверяется, не превышает ли расстояние до ближайшего соседа размера максимального ребра КНП в этом образе. Если превышает, то об этом выдается сообщение или объект выделяется в самостоятельный таксон.

В случае, когда выбранные ближайшие соседи принадлежат разным образам, то производится по описанной процедуре присоединение его к каждому из конкурирующих таксонов и решение принимается по максимуму значения функции качества.

Режимы печати:

- нет печати;
- итоговый результат распознавания;
- выдаются также входная информация и результаты отнесения распознаваемого объекта к каждому из проверяемых образов;
- выдается также КНП внутри исходных таксонов.

Для работы программы TRF1 необходимо выполнение следующих правил:

1. Таблица для распознавания и описание ее параметров должны храниться в файлах WORK.DAT и WORK.PET соответственно.

2. Обучающая таблица с данными и описание ее параметров должны храниться в файлах с расширением .DAT и .PET соответственно, но с именем,отличным от WORK.

3. В файле с расширением PET обучающей таблицы должны быть следующие данные: номер целевого признака, количество образов, индекс упорядочения по образам и отдельный целевой признак.

6.4.4. Алгоритм TRF2.

В алгоритме TRF2 решение о принадлежности распознаваемых объектов тому или иному образу принимается на объединенном множестве всех распознаваемых и обучающих объектов.

Все объекты одновременно делятся алгоритмом FOREL2 на число таксонов, не меньшее количества исходных образов. Проверяется распределение обучающих объектов по таксонам. Если в таксоне имеются обучающие объекты только одного образа, то и все распознаваемые объекты, попавшие в таксон, относятся к этому образу. Если в таксоне имеются представители разных образов,то объекты этого таксона подвергаются дальнейшей таксономии.

Дробление таксонов повторяется до тех пор, пока в таксо - нах либо не останется обучающих объектов, принадлежащих разным образам, либо не будет превышено указанное число шагов.Таксон, содержащий только распознаваемые объекты, относится к тому исходному образу, расстояние до центра которого минимально.

Для работы с данным алгоритмом необходимо подготовить файл с номерами обучающих объектов, сгруппированных по образам. Объекты каждого образа должны быть упорядочены по возра - станию их номеров, разделитель между номерами - пробел. Разделитель между образами - 0 (ноль).

Режимы печати:

- без печати;

- выдается итоговая таблица распределения объектов по образам;

- на каждом шаге работы программы выдается также и результат таксономии.

Ниже приводится перечень вопросов, которые могут быть заданы системой (в зависимости от конкретной ситуации):

В программе TRF1

Имя файла с обучающей таблицей (с расширением .DAT)
Имя файла параметров обучающей таблицы (с расширением .PET)
Режим печати (0-3)
Учитывать ли равномерность распределения объектов: 1 - да, 2 - нет

В программе TRF2

Количество обучающих объектов
Количество образов, представленных обучающей выборкой
Число шагов для выделения однородных таксонов прогр. FOREL2
Количество таксонов, выделяемых программой FOREL2
Режим печати: 0,1,2
Имя файла с номерами обучающих объектов

6.4.5. Пример решения по программам TRF1 и TRF2.

Рассмотрим на тех же данных следующий пример. Возьмем одну и ту же таблицу как в качестве обучающей, так и в качестве контрольной (распознаваемой) таблицы. Мы помним, что по условиям задания информации эти таблицы должны находиться в разных файлах: контрольная - в файле WORK.DAT, а обучающая - в файле с таким же расширением .DAT, но отличным от WORK. именем.

Пусть объекты обучающей выборки будут распределены по 5 образам. Сообщение перед началом работы программы TRF1:

```
                    ПРОГРАММА TRF1

          Признаков  14, объектов  66, образов 5,
                распознаваемых объектов 66
          РЕЖИМ 2 (без учета равномерности) - критерий F2
          КОЛИЧЕСТВО ОБ'ЕКТОВ ОБУЧАЮЩЕЙ  ВЫБОРКИ В  ОБРАЗАХ

    Номер образа  ***1***2***3***4***5***6***7***8***9***0
Количество объектов  8  20  20   8  10

     РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ОБУЧАЮЩЕЙ ВЫБОРКИ ПО ОБРАЗАМ

     На месте, соответствующем порядковому номеру объекта,
        ставится номер образа, которому он принадлежит

 **1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
   2  2  1  4  2  2  3  3  2  2  3  2  5  4  4  2  3  2  3  1
   2  3  3  5  5  5  4  4  3  3  3  2  2  3  4  2  5  5  5  4
   5  5  3  1  2  1  2  1  2  1  3  3  2  2  3  1  3  4  2  2
   1  3  3  3  3  5
```

В процессе работы программа TRF1 будет поочередно решать судьбу каждого распознаваемого объекта. Так, первый объект может быть отнесен к образу 2 или 3, так как 5 ближайших к нему объектов обучающей выборки принадлежат этим образам. После вычисления критериев оказалось, что связь распознаваемого объек-

та со 2-м образом больше, чем 3-м, так что 1-й объект отнесен ко 2-му образу. Эти и другие сведения содержатся в следующей таблице:

```
                  Объект   Образ   Критерий
                    1        2       .687
                    1        3       .544

 ОБ'ЕКТ  1  ПРИНАДЛЕЖИТ ОБРАЗУ  2   КРИТЕРИЙ =     .687

                  Объект   Образ   Критерий
                    4        4       .704
                    4        5       .452
                    4        3       .478
 ОБ'ЕКТ  4  ПРИНАДЛЕЖИТ ОБРАЗУ  4   КРИТЕРИЙ =     .704

                  Объект   Образ   Критерий
                   66        5       .700
                   66        3       .451

 ОБ'ЕКТ 66  ПРИНАДЛЕЖИТ ОБРАЗУ  5   КРИТЕРИЙ =     .700

ХАРАКТЕРИСТИКА РАСПРЕДЕЛЕНИЯ ОБ'ЕКТОВ ВНУТРИ ИСХОДНЫХ ОБРАЗОВ
 Среднее расстояние между соседними (по графу) объектами
     1.710     1.815     2.194     2.948     2.970
Максимальное расстояние между соседними (по графу) объектами
     2.912     2.367     4.709     4.491     6.093
```

Итоговая информация к программе TRF1.

Если в итоговой таблице:

1) на месте критерия стоят символы ****, это значит, что объект отнесен к образу без вычисления критерия, так как все объекты, ближайшие к распознаваемому, принадлежат этому обра - зу;

2) номер образа помечен символом '-', то распознаваемый объект выделен в отдельный образ, на печать выдан номер ближай-

шего к нему образа, а на месте критерия стоит минимальное расстояние до этого ближайшего образа.

НОМЕР ОБ'ЕКТА	НОМЕР ОБРАЗА	КРИТЕРИЙ	НОМЕР ОБ'ЕКТА	НОМЕР ОБРАЗА	КРИТЕРИЙ
1	2	.687	34	3	.688
2	2	********	35	4	.704
3	1	.695	36	2	.687
4	4	.704	37	5	********
5	2	********	38	5	.700
6	2	********	39	5	.700
7	3	.688	40	4	.704
8	3	.867	41	5	********
9	2	.687	42	5	********
10	2	.687	43	3	.688
11	3	.688	44	1	.695
12	2	.687	45	2	********
13	5	.700	46	1	********
14	4	.704	47	2	1.001
15	4	.704	48	1	.695
16	2	.687	49	2	.687
17	3	.900	50	1	.695
18	2	********	51	3	.688
19	3	.688	52	3	.688
20	1	.695	53	2	.687
21	2	.687	54	2	.687
22	3	.912	55	3	.688
23	3	.688	56	1	1.012
24	5	.700	57	3	.688
25	5	********	58	4	.704
26	5	.914	59	2	.687
27	4	.887	60	2	.687
28	4	.704	61	1	.695
29	3	********	62	3	.688
30	3	.683	63	3	.688
31	3	.688	64	3	********
32	2	.910	65	3	.683
33	2	.687	66	5	.700

Рассмотрим процесс решения по программе TRF2. Для этой программы требуется задание сведений о тех объектах таблицы,ко-

торые будут использованы в качестве обучающей выборки. Напомним, что информация об обучающих объектах представляется списками их номеров, сгруппированных по образам. В каждом списке номера должны быть упорядочены по возрастанию, разделитель между номерами - пробел. Разделитель между списками - 0 (ноль).

Для решения нашей задачи зададим следующий список 23 объектов, представляющих 5 образов:

```
3 20 44 46 0 2 5 6 18 47 53 54 0 7 8 29 64 0 4 14 15 58 0
25 37 39 42 0
```

Программа проинформирует о заданных ей условиях:

```
                    ПРОГРАММА TRF2
   Признаков - 14.  Объектов - 66.  Образов - 5.

  ВЕКТОР РАСПРЕДЕЛЕНИЯ  ОБУЧАЮЩИХ ОБ'ЕКТОВ ПО ТАКСОНАМ

№ объектов   2   3   4   5   6   7   8  14  15  18  20
№ таксонов   2   1   4   2   2   3   3   4   4   2   1

№ объектов  25  29  37  39  42  44  46  47  53  54  58  64
№ таксонов   5   3   5   5   5   1   1   2   2   2   4   3
```

Затем программа обратится к программе FOREL2, которая при заданных ей условиях в конце концов получит 20 таксонов.

```
                    ПРОГРАММА FOREL2

        РАСПРЕДЕЛЕНИЕ  ОБ'ЕКТОВ ПО ПОЛУЧЕННЫМ ТАКСОНАМ

*1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
13  2  1  4  2  2  3  3  3  2  3  2 18  4  4  2  5  2  3  1
 8  3  5  5  5  5  5  5  3  7 10  3 12  5 14 10  5 20  5 19
 5  5 19  1 16  1  2  6 15  1  9 11  2  2 17  2  4  4 14 14
14  9  9  3 14  5

        КОЛИЧЕСТВО ОБ'ЕКТОВ В ПОЛУЧЕННЫХ ТАКСОНАХ

**1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
  5 11  9  5 13  1  1  1  3  2  1  1  1  5  1  1  1  1  2  1
```

Полученные 20 таксонов должны быть укрупнены до требуемого количества - 5 таксонов, поскольку в обучающей выборке были представлены объекты 5 образов.

```
                    УКРУПНЕНИЕ ТАКСОНОВ
              Исходное количество таксонов  20
              Требуемое количество таксонов  5

        РАСПРЕДЕЛЕНИЕ ОБ'ЕКТОВ ПО УКРУПНЕННЫМ ТАКСОНАМ
*1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
 1  2  1  4  2  2  3  3  3  2  3  2  5  4  4  2  5  2  3  1
 1  3  5  5  5  5  5  5  3  3  1  3  1  5  1  1  5  5  5  5
 5  5  5  1  1  1  2  1  1  1  1  2  2  2  2  2  4  4  1  1
 1  1  1  3  1  5

          КОЛИЧЕСТВО ОБ'ЕКТОВ В УКРУПНЕННЫХ ТАКСОНАХ
**1**2**3**4**5**6**7**8**9**0**1**2**3**4**5**6**7**8**9**0
 21 13 10  5 17
```

6.4.6. Алгоритм и программа ACQUIS.

Программа ACQUIS по обучающей таблице строит решающее правило в виде логического дерева и с помощью этого решающего правила выполняет распознавание контрольной таблицы. Содержание логического дерева легко преобразуется в список правил типа "Если..., то...". Эти правила есть ни что иное, как знания,которыми наполняется База Знаний экспертной системы. Следовательно, программа ACQUIS может использоваться как для распознавания образов, так и для автоматического обнаружения знаний,содержащихся в таблицах данных.

Построение решающего правила может быть выполнено как на всех признаках, так и только на указанных пользователем. Для того чтобы не изменять таблицу исходных данных, в программе предусмотрена возможность указания нужных признаков путем задания специального вектора. В этом векторе на месте, соответствующем порядковому номеру признака, должна быть поставлена 1, если признак не будет учитываться при построении решающего правила, и 0 - в противном случае.

Допускаются пропуски некоторых значений признаков, признаки могут быть разнотипными. Предположений о виде распределения в пространстве признаков не делается, но алгоритм обладает рядом положительных статистических свойств. Число классов k больше либо равно 2. Для наглядности поясним основные принципы работы алгоритма на примере, где каждый объект может принадлежать к одному из 2-х образов (k = 2).

Пусть задан набор $[a_1,\ldots, a_n]$ объектов, например, людей. Для каждого объекта определено значение признака $x(j)$, $j = 1,\ldots, m$, например: $x(1)$ - рост, $x(2)$ - вес, $x(3)$ - образование, $x(4)$ - температура, $x(5)$ - давление, $x(6)$ - наличие насморка.

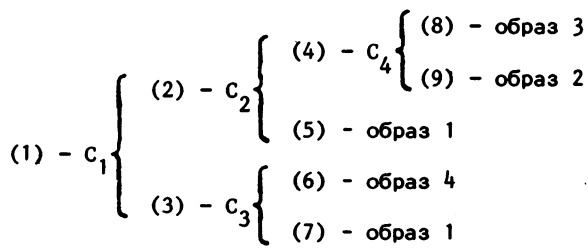
Каждый объект может принадлежать к одному из 2-х образов, т.е. целевой признак принимает только два значения: 1 и 2 (первый и второй образы). В нашем примере будем полагать: 1-й образ - больные гриппом, 2-й образ - здоровые люди.

Решающим правилом W будем считать любую процедуру, которая по описанию объекта в пространстве признаков, т.е. по вектору $x(1),\ldots, x(m)$, предсказывает образ. В зависимости от того, совпадает или нет предсказанное значение образа с предполагаемым, предсказание считается либо верным, либо ошибочным. В нашем примере мы в качестве W могли бы принять следующее правило: "если у человека насморк и температура > 37.1, то болен; если нет насморка и температура > 37.5, то болен; и, если нет насморка и температура меньше 37.5, то здоров".

Опишем теперь конкретный класс решающих правил - класс логических деревьев. Назовем элементарным высказыванием C высказывание вида $x(j) \leq p$, $x(j) > p$ для признака, замеренного в шкале порядка и в более сильных шкалах, и B - высказывание вида $x(j) = p$, $x(j) \neq p$ для признака, замеренного в шкале наименований. Здесь p - конкретное значение признака. Будем считать, что условие выполнено, если $x(j) \leq p$ (для шкалы поряд-

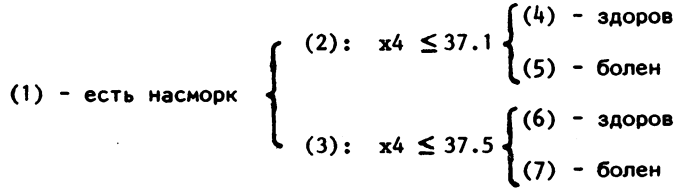
ка и более сильных шкал) или $x(j) = p$ (для шкалы наименований).

На наборе элементарных высказываний $C_1,\ldots,C_l$ можно построить конъюнкцию $D = C_1 \& C_2 \& \ldots \& C_l$. Конъюнкция D выполняется, если выполняется каждое C_i из D. Набор конъюнкций может быть представлен в виде дерева, например, такого:



Все вершины дерева занумерованы. В каждой не конечной вершине дерева указано некоторое элементарное высказывание C_i. Если оно истинно, то объект попадает в вершину i, а если ложно, то в вершину i+1, где i и i+1 - заданные номера вершин, выходящих из данной. Если вершина конечна, то в ней принимается решение о принадлежности к одному из k образов.

Указанное выше решающее правило о заболевании можно было бы представить в виде дерева следующим образом:



Алгоритм, реализованный в программе, строит на основе обучающей выборки решающее правило в виде логического дерева, описанного выше. Можно сказать, что алгоритм стремится отыскать

логическое дерево с небольшим числом вершин, которое дает малое число ошибок на обучающей выборке. Алгоритм перебора деревьев основан на принципе "лучший к лучшему". На первом шаге строится высказывание C_1, лучшее в смысле некоторого критерия F, связанного с качеством распознавания заданных образов. Объекты обучающей выборки разбиваются на две группы: удовлетворяющие C_1 и не удовлетворяющие C_1. Далее для каждой группы отыскивается опять свое лучшее в смысле F высказывание, получаем C_2 и C_3. Деление продолжается до тех пор, пока не достигнем заданного количества ветвей дерева или других условий останова.

Опишем критерий F и условия останова. Пусть $n_1, n_2, \ldots, n_k$ - количество объектов каждого образа в i-й вершине дерева, а $m_1, m_2, \ldots, m_k$ и $l_1, l_2, \ldots, l_k$ - количество объектов каждого образа в (i+1)-й и (i+2)-й вершинах, выходящих из i-й вершины:

$$(1) - [n_1, n_2, \ldots, n_k] \begin{cases} (2) - [m_1, m_2, \ldots, m_k] \\ (3) - [l_1, l_2, \ldots, l_k]. \end{cases}$$

Первый вариант критерия F отражает количество правильно распознаваемых объектов:

$$F = \max(m_1, m_2, \ldots, m_k) + \max(l_1, l_2, \ldots, l_k).$$

Второй вариант критерия F максимизирует оценку условной вероятности образа, при условии попадания в заданную вершину:

$$F = \max\left\{ \frac{\max(m_1, m_2, \ldots, m_k) - CF}{m_1 + m_2 + \ldots + m_k}, \quad \frac{\max(l_1, l_2, \ldots, l_k) - CF}{l_1 + l_2 + \ldots + l_k} \right\}.$$

Константа CF добавляется, чтобы избежать хорошей оценки (т.е. близкой к 1) в ситуации, когда все $m_i = 0$, кроме одного, либо все $l_i = 0$, кроме одного. Значения CF желательно задавать меньше либо равным 1.

Программа может строить решающее правило на основе любого из этих двух критериев: по количеству правильно распознавае - мых объектов или по максимуму отношения правдоподобия в каждой рассматриваемой вершине.

Условия останова.

В процессе построения решающего правила прекращение ветвления дерева в каждой вершине может происходить, если в данной вершине не превосходит указанного минимума одна из следующих трех величин:

- число неправильно распознанных объектов;
- пороговая величина приращения значения критерия;
- число объектов.

Критерием для окончания процедуры построения дерева также является предельно допустимое число вершин. В данной программе число вершин не должно превышать 70.

Режимы работы программы:

1 - построение решающего правила (с распознаванием объектов обучающей выборки);

2 - дополнительно к режиму 1 - распознавание объектов новой таблицы;

3 - распознавание объектов по известному решающему правилу.

В зависимости от режима работы программы при входе в соответствующий раздел меню системы Вам будут заданы вопросы об условиях решения. Общий список вопросов приведен ниже:

1	режим работы: 1,2,3
2	сохранять ли формализованное решающее правило: 1 - да,0-нет
3	имя файла для сохранения формализованного решающего правила
4	выводить ли на печать таблицу исходных данных: 1-да, 0 -нет
5	имя файла с решающим правилом
6	имя файла с данными распознав.табл. (с расширением .DAT)
7	имя файла параметров распознав.табл. (с расширением .PET)
8	количество вершин
9	имя файла с вектором, определяющим запреты на признаки

10 по какому критерию вести разбиение: 1,2
11 1-е условие прекращения ветвления в вершине
12 2-е условие прекращения ветвления в вершине
13 значение параметра смещения для вычисл.крит.правдоподобия
14 3-е условие прекращения ветвления в вершине
15 одинаков ли порог валидности для всех призн.: 0 - да,1-нет
16 порог валидности
17 имя файла с вектором значений порогов валидности

Система задает только те вопросы, которые нужны для данного конкретного варианта. Если Вы будете выполнять решение в режимах 1 или 2, то из этого списка не появится вопрос 5 об имени файла, где хранится готовое решающее правило. В режиме 1 не будут заданы вопросы 6,7. Для режима 3 (распознавание объектов по известному решающему правилу) не будут заданы вопросы 2, 3,6-17. Все варианты возможных меню могут быть получены на основе приведенной ниже таблицы.

Если параметр	принимает значение	нет вопроса
1	1	5,6,7
1	2	5
1	3	2,3,6-17
2	0	3
10	1	13
15	0	17
15	1	16

Учтите, что при работе в режиме 2 (с дополнительным к режиму 1 распознаванием объектов новой таблицы) при ответах на вопросы 6 и 7 об именах файлов с данными и параметрами распознаваемой таблицы может быть указано и имя WORK. При этом фактически выполняется решение режима 1.

Программа при построении решающего правила может выдать его в нескольких вариантах. Для дальнейшего использования при распознавании новых объектов требуется решающее правило в формализованном виде. Этот же вариант представления удобен и в

том случае, если Вы желаете построить соответствующее этому правилу дерево. Формализованное решающее правило формируется в виде таблицы:

1*	2*	3*	4*	5*
1	4	11.10	2	3
2	1	410.00	4	5
...	...	...	...	...
...	...	...	...	...

где 1* - номер вершины; 2* - номер признака, по которому строится высказывание; 3* - пороговое значение признака; 4* - номер вершины, в которую надо идти, если высказывание ИСТИННО; 5* - номер вершины, в которую надо идти, если высказывание ЛОЖНО.

Из списка вопросов, задаваемых системой, осталось пояснить понятие "порог валидности". Значения показателей в логических высказываниях, разделяющих классы, должны "достаточно существенно" отличаться друг от друга. Порог валидности определяет допустимое минимальное отличие между значениями одного и того же показателя при включении его в решающее правило для разделения разных классов. Значение этого параметра задается пользователем из содержательных соображений. Если не задать никаких дополнительных ограничений, то в результате работы программы может быть найдено решающее правило, "уловившее" очень незначительное расхождение в значении какого-либо показателя для объектов разных классов. Если заранее порог установить затруднительно, то можно решить задачу без ограничений, проанализировать полученное решение, а затем, если необходимо, повторить решение при ограничениях.

Если порог одинаков для всех признаков, то достаточно задать его только один раз (задавать нулем в случае несущественности критерия). В противном случае пороговые значения должны быть указаны для каждого признака в файле (с именем, отличным

от WORK.*) рядом последовательных чисел, разделенных пробелами. Если для каких-либо признаков валидность учитывать не нужно, то на соответствующем им месте следует поставить "0".

6.4.7. Пример решения по программе ACQUIS.

Рассмотрим результат решения при следующих условиях:

```
Объектов - 305, признаков - 30, образов - 2
Режим работы программы - 1. Обозначение пробела - .8E+18
Вариант критерия - 2. Минимум "чужих" объектов в вершине - 1
Величина приращения значения критерия в вершине - .0
Параметр смещения в критерии 2 - 1.0
Минимум числа объектов в конечных вершинах - 1
Предельно допустимое количество вершин - 60
Все признаки количественного типа
Порог валидности не задан
Номера признаков, не участвующих в решении: 1
```

Программа в результате обучения на 305 объектах,принадлежащих двум классам (вектор распределения этих объектов задан в файле), построит решающее правило и выполнит по этому правилу распознавание обучающей выборки, т.е. этих же 305 объектов.

РЕЗУЛЬТАТ ОБУЧЕНИЯ

РЕШАЮЩЕЕ ПРАВИЛО (формализованное)

1*	2*	3*	4*	5*
1	19	76.00	2	3
2	0	,00	0	0
3	5	12.00	4	5
4	9	4.00	6	7
5	0	.00	0	0
6	23	36.00	8	9
7	0	.00	0	0
8	2	87.00	10	11
9	0	.00	0	0
10	3	256.00	12	13
11	0	.00	0	0
12	18	-16.00	14	15
13	0	.00	0	0
14	0	.00	0	0
15	22	71.00	16	17
16	12	0160.00	18	19

17	0	.00	0	0
18	19	90.00	40	41
19	23	6.00	20	21
20	0	.00	0	0
21	12	0168.00	22	23
22	0	.00	0	0
23	28	5.00	24	25
24	7	.00	26	27
25	21	-8.00	48	49
26	25	1.00	28	29
27	0	.00	0	0
28	6	16.00	30	31
29	0	.00	0	0
30	0	.00	0	0
31	29	5.00	32	33
32	3	120.00	34	35
33	0	.00	0	0
34	0	.00	0	0
35	13	79.00	36	37
36	0	.00	0	0
37	13	103.00	38	39
38	0	.00	0	0
39	21	-15.00	44	45
40	0	.00	0	0
41	24	.00	42	43
42	9	1.00	46	47
43	0	.00	0	0
44	0	.00	0	0
45	15	108.00	54	55
46	0	.00	0	0
47	3	200.00	56	57
48	0	.00	0	0
49	2	61.00	50	51
50	0	.00	0	0
51	15	143.00	52	53
52	24	.00	58	59
53	0	.00	0	0
54	0	.00	0	0
55	0	.00	0	0
56	0	.00	0	0
57	0	.00	0	0
58	0	.00	0	0
59	0	.00	0	0

Количество вевтей дерева = 30
Количество ошибок на обучении = 12

ПОЯСНЕНИЯ К ТАБЛИЦЕ:

1* - номер вершины;

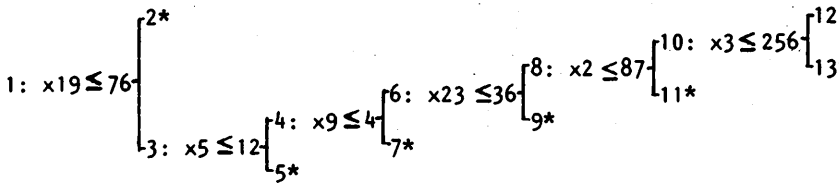
2* - номер признака, по которому строится высказывание;

3* - пороговое значение признака;

4* - номер вершины, в которую надо идти, если высказыва - ние ИСТИННО;

5* - номер вершины, в которую надо идти, если высказыва - ние ЛОЖНО.

Дерево, соответствующее этому решающему правилу, содержит 59 вершин. Для примера начнем строить это дерево:

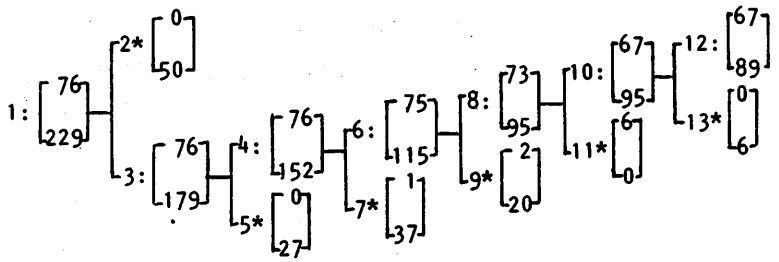


Вершины пронумерованы, и за теми вершинами, где происхо - дит дальнейшее ветвление, выписано взятое из приведенного выше решающего правила условие. Конечные вершины помечены * . Для каждой вершины в порядке их нумерации программа представляет также сведения о количестве относящихся к данной вершине объектов разных (в нашем случае 2-х) классов. Приведем эти сведения для тех же 13 вершин, для которых был построен фраг - мент дерева.

РАСПРЕДЕЛЕНИЕ ЧИСЛА ОБ'ЕКТОВ ПО ОБРАЗАМ В ВЕРШИНАХ

Номера образа	Номера вершин												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	76	0	76	76	0	75	1	73	2	67	6	67	0
	229	50	179	152	27	115	37	95	20	95	0	39	6

Ниже приведен тот же участок дерева с указанием в каждой вершине сведений об объектах.



Каждый из приведенных выше вариантов представления результатов решения обладает своей степенью наглядности. Формализованное правило необходимо также для дальнейшего использования при распознавании новых объектов (Вы помните, что для этих целей оно должно быть сохранено после его построения). Для дальнейшей "ручной" работы и облегчения понимания свойств исследуемого материала удобна следующая расшифровка результатов, в которой в явном виде выписаны решающие правила для конечных вершин, количество представленных ими объектов и полный перечень этих объектов.

РЕЗУЛЬТАТ РАСПОЗНАВАНИЯ ОБУЧАЮЩЕЙ ВЫБОРКИ

№	Кол-во объектов	Номер образа	Кол-во ошибок	Содержимое группы: номера объектов
1	50	2	0	Номера объектов 2-го образа
				77 78 79 3 85 19 22
				99 4 5 6 119 121 125
				126 127 128 130 8 9 158
				162 164 11 2 15 139 52
				210 216 56 57 230 [illegible] 59
				60 61 247 249 255 [illegible] 63

64 266 66 67 71 74 302
303

===== УСЛОВИЕ В ВЕРШИНЕ 2 =====
(x19 ≤ 76.0)

---

2 5 2 0 Номера объектов 2-го образа
115 53 223 224 225

===== УСЛОВИЕ В ВЕРШИНЕ 14 =====
(x2 ≤ 87.0) & (x3 ≤ 256.) & (x5 ≤ 12.0) & (x9 ≤ 4.0) &
& (x18 ≤ -16.0) & (x19 > 76.0) & (x23 ≤ 36.0)

---

4 4 1 0 Номера объектов 1-го образа
138 141 229 260

===== УСЛОВИЕ В ВЕРШИНЕ 46 =====
(x2 ≤ 87.0) & (x3 ≤ 256.) & (x5 ≤ 12.0) & (x9 ≤ 1.0) &
& (x12 ≤ 10160) & (x18 > -16.0) & (x19 > 90.0) & (x22 ≤ 71.0) &
& (x23 ≤ 36.0) & (x24 ≤ .0)

---

6 4 1 1 Номера объектов 1-го образа
134 182 207
Номера объектов 2-го образца
183

===== УСЛОВИЕ В ВЕРШИНЕ 57 =====
(x2 ≤ 87.0) & (x3 ≤ 256.) & (x3 > 200) & (x5 ≤ 12.0) &
& (x9 ≤ 4.0) & (x9 > 1.0) & (x12 ≤ 10160) & (x18 > -16.0) &
& (x19 > 90.0) & (x22 ≤ 71.0) & (x23 ≤ 36.0) & (x24 ≤ 0)

---

13 4 2 0 Номера объектов 2-го образа
92 101 35 239

===== УСЛОВИЕ В ВЕРШИНЕ 33 =====
(x2 ≤ 87.0) & (x3 > 120.) & (x3 ≤ 256.) & (x5 ≤ 12.0) &
& (x6 > 16.0) & (x7 ≤ .0) & (x9 ≤ 4.0) & (x12 > 10168.) &
& (x13 ≤ 103.) & (x13 > 79.0) & (x19 > 76.0) & (x18 > -16.0) &
& (x22 ≤ 71.0) & (x23 > 6.0) & (x23 ≤ 36.0) & (x25 ≤ 1.0) &
& (x28 ≤ 5.0) & (x29 ≤ 5.0)

По тем же решающим правилам выполнено распознавание 73-х новых объектов.

РЕЗУЛЬТАТ РАСПОЗНАВАНИЯ НОВЫХ ОБ'ЕКТОВ

№	Кол-во объектов	Номер образа	Содержимое группы: номера объектов
1	55	2	1 2 4 5 6 7 8 9 10 11 12 14 15 16 17 18 19 20 26 27 31 32 33 34 35 36 37 38 39 40 43 44 45 46 47 48 49 52 55 56 60 61 62 63 64 65 66 67 68 69 70 72 73 74 75 ===== УСЛОВИЕ В ВЕРШИНЕ 2 ===== (x19 ≤ 76.0)
5	1	2	21 ===== УСЛОВИЕ В ВЕРШИНЕ 56 ===== (x2 ≤ 87.0) & (x3 ≤ 200.0) & (x5 ≤ 12.0) & (x9 ≤ 4.0) & &(x9 > 1.0) &(x12 ≤ 10160) & (x18 > -16.0) & (x19 > 90.0)& &(x22 ≤ 71.0) & (x23 ≤ 36.0) & (x24 ≤ .0)
19	1	1	29 ===== УСЛОВИЕ В ВЕРШИНЕ 27 ===== (x2 ≤ 87.0) & (x3 ≤ 256.0) & (x5 ≤ 12.0) & (x7 > .0) & &(x9 ≤ 4.0) & (x 2 > 10160) &(x18 > -16.0) & (x19 > 76.0)& &(x22 ≤ 71.0) & (x23 ≤ 36.0) & (x23 > 6.0) & (x28 ≤ 5.0)
21	1	2	58 ===== УСЛОВИЕ В ВЕРШИНЕ 50 ===== (x2 ≤ 61.0) & (x3 ≤ 256.0) & (x5 ≤ 12.0) & (x9 ≤ 4.0)& &(x12 > 10168.) & (x18 > -16.0) & (x19 > 76.0) &(x21 > -8.0)& &(x22 ≤ 71.0) & (x23 ≤ 36.0) & (x23 > 6.0) & (x28 > 5.0)

27	1	1	30

===== УСЛОВИЕ В ВЕРШИНЕ 11 =====
(x2 > 87.0) & (x5 ≤ 12.0) & (x9 ≤ 4.0) & (x19 > 76.0) & & (x23 ≤ 36.0)

28	7	2	24	25	28	50	53	54	57

===== УСЛОВИЕ В ВЕРШИНЕ 9 =====
(x19 > 76.0) & (x5 ≤12.0) & (x9 ≤4.0) & (x23 > 36.0)

29	5	2	3	22	23	51	71

===== УСЛОВИЕ В ВЕРШИНЕ 7 =====
(x19 > 76.0) & (x5 ≤ 12.0) & (x9 > 4.0)

30	4	2	13	41	42	59

===== УСЛОВИЕ В ВЕРШИНЕ 5 =====
(x19 > 76.0) & (x5 > 12.0)

В данном случае нам было известно распределение этих объектов по тем же 2-м классам, так что была возможность дополнительно сравнить пригодность построенных правил для распознавания новых объектов.

НОМЕРА НЕПРАВИЛЬНО РАСПОЗНАННЫХ ОБ'ЕКТОВ:

Представление информации:
- Номер объекта
- Исходный номер объекта
- Предсказанный номер объекта

5	10	12	14	29	31	32	34	35	36	49	50
1	1	1	1	2	1	1	1	1	1	1	1
2	2	2	2	1	2	2	2	2	2	2	2

56	67	68	69	71	72	73
1	1	1	1	1	1	1
2	2	2	2	2	2	2

*ЧИСЛО ОШИБОК РАСПОЗНАВАНИЯ ПРИ КОНТРОЛЕ = 19

6.5. ZET-комплекс.

Таблицы "объект-свойство", с которыми приходится работать при решении реальных задач, как правило, имеют пробелы, т.е. по каким-либо причинам в таблицах могут отсутствовать значения отдельных параметров для некоторых объектов. Большинство известных методов анализа данных не могут обрабатывать такую информацию. Необходимость работы с неполными таблицами привела к разработке метода "заполнения эмпирических таблиц" - ZET. За годы активной эксплуатации алгоритм подтвердил свою высокую эффективность; он и реализующие его программы постоянно модифицировались с учетом накопленного опыта. К настоящему времени создано несколько базовых версий программы.

Заполнение пробелов в таблицах "объект-свойство" и сегодня остается одним из основных назначений алгоритма ZET. Однако в процессе его применения оказалось, что не менее существенна и возможность проверки данных, имеющихся в таблице. Если "закрыть" известный элемент таблицы, спрогнозировать его значение по алгоритму ZET и затем сравнить результат прогноза с реально имеющимся в таблице значением, то можно с достаточной степенью уверенности решить вопрос о том, "естественно" это значение для данной таблицы, подчиняется ли оно общим для нее закономерностям или "чужеродно", возможно, ошибочно. Очень часто в случае резкого расхождения исходного и спрогнозированного значений выявляются ошибки в задании информации. Поэтому такой вариант применения алгоритма ZET был назван "редактированием таблиц".

Поскольку решение вопроса о значении, которое должно быть поставлено в таблицу на место пробела, фактически и есть прогнозирование этого значения, то со временем алгоритм стал активно применяться для решения задач прогноза. С учетом специфики некоторых важных конкретных задач был разработан и успешно применяется для прогнозирования целый ряд алгоритмов и про-

грамм, использующих в качестве основного функционального блока программу ZET.

6.5.1. Алгоритм ZET. Алгоритм предназначен для прогнозирования значений пропущенных элементов (заполнения пробелов) в таблицах "объект-свойство" и для редактирования (проверки) всей таблицы или ее части. В реальных таблицах данных имеется избыточность, выражающаяся в том, что многие признаки (столбцы) связаны друг с другом определенной зависимостью, есть в таблице и объекты (строки), похожие друг на друга по значениям своих характеристик. В алгоритме выявляются такие связи и по - хожести, и на их основе выполняется предсказание искомого значения с высокой точностью.

Напомним, что в таблице "объект-свойство" строка соответствует рассматриваемым объектам, а столбцы есть не что иное, как значения признаков (свойств), характеризующих эти объекты. Таким образом, на пересечении строки с номером i и столбца с номером j будет находиться значение j-го признака для i-го объекта. Клеточка таблицы, расположенная на пересечении i-й строки и j-го столбца, обозначается символом "x(i,j)", где первый индекс всегда будет соответствовать номеру строки,а второй - номеру столбца.

Одна из основных идей, позволившая построить эффективный метод и получать хорошие результаты, заключается в том, что предсказание выполняется не на всей информации, имеющейся в таблице, а только на той ее части, которая наиболее тесно связана со строкой и столбцом, в которых этот пробел находится. Другими словами, в алгоритме ZET, в отличии от многих других алгоритмов заполнения пробелов, реализуется "локальный" подход к предсказанию (заполнению) каждого пропущенного значения, для вычисления этого значения строится своя "предсказывающая подматрица", содержащая только имеющую отношение к делу инфор-

мацию. В подматрицу отбираются в порядке убывания сходства строки - или "объекты-аналоги", т.е. строки, самые похожие на строку, содержащую интересующий нас пробел, а затем для выбранных строк отбираются также в порядке убывания сходства столбцы, "самые похожие" на столбец, содержащий этот пробел. Размер подматрицы (число строк и столбцов) задается пользователем.

Вопрос ограничения размера предсказывающих подматриц (мы не рекомендуем их более чем 10*10) связан не только с сокращением времени вычислений, но и с проблемой получения наиболее качественного решения. Если закономерность четко проявлена на многих элементах таблицы, то и использование части самой связанной с предсказанием информации даст хорошие результаты. Но если в таблице имеется немного объектов (строк), подчиняющихся единой закономерности, то использование для прогноза излишнего количества "чужой" информации заведомо не улучшит результат.

В алгоритме ZET производится оценка ожидаемой ошибки прогноза. Поскольку априорных критериев достаточности информации для каждого конкретного случая нет, в алгоритме качество прогноза оценивается косвенно по качеству предсказания известных элементов таблицы, связанных с интересующим нас пробелом (стоящих с ним в одной строке или в одном столбце). Если эти элементы предсказываются уверенно или, наоборот, плохо, то из этого можно сделать и вывод об ожидаемом качестве прогноза неизвестного элемента.

Достаточно подробно алгоритм опубликован в ряде изданий, в том числе: 1. Загоруйко Н.Г., Ёлкина В.Н., Емельянов С.В., Лбов Г.С. Пакет программ ОТЭКС для анализа данных.- М.: Финансы и статистика, 1986 и 2. Ёлкина В.Н., Загоруйко Н.Г. Новоселов Ю.А. Математические методы агроинформатики.- Новосибирск: Наука СО, 1987.

Программы ZETM1 и ZETM3. В настоящей версии Блока Анализа Данных имеются две программы (ZETM1 и ZETM3), реализующие ал-

горитм ZET. Они отличаются способом вычисления оценки точности прогноза.

В программе ZETM1 вычисление основано на относительной ошибке:

$$Y_1 = \mathrm{mod}\,[(\tilde{X}-X_o)/X_o],$$

где X_o - реальное значение, $\tilde{X}$ - прогноз.

В программе ZETM3 ошибка вычисляется как

$$Y_2 = \mathrm{mod}[(\tilde{X}-X_o)/(X_{max}-X_{min})],$$

где X_{max} и X_{min} - максимальное и минимальное значения оцениваемого параметра в группе объектов-аналогов.

При выборе программы следует учитывать, что относительная ошибка Y_1 неинвариантна к аддитивной составляющей и чувствительна к малым значениям X_o. Но результаты решения с относительной ошибкой несколько нагляднее и их проще интерпретировать.

Режим редактирования. Уже было сказано, что алгоритм ZET может быть использован для проверки данных, имеющихся в таблице. Если "закрыть" известный элемент таблицы, спрогнозировать его значение по алгоритму ZET и затем сравнить результат прогноза с реально имеющимся в таблице значением, то можно с достаточной степенью уверенности решить вопрос о том, "естественно" это значение для этой таблицы, подчиняется ли оно общим для нее закономерностям или "чужеродно", возможно, ошибочно. Очень часто в случае резкого расхождения исходного и спрогнозированного значений выявляются ошибки в задании информации. Такой вариант применения алгоритма ZET был назван "редактированием таблиц".

В программе, реализующей алгоритм ZET, имеется возможность при редактировании формировать предсказывающую подматрицу двумя способами: с учетом и без учета редактируемого элемента. Это означает, что при подборе "похожих" строк и столбцов в

первом случае элемент таблицы, который нужно перепроверить, находится на своем месте и участвует в вычислении мер сходства. В процессе редактирования с учетом редактируемого элемента символ пробела автоматически ставится на место редактируемого элемента в уже сформированной предсказывающей подматрице.

При редактировании без учета редактируемого элемента символ пробела автоматически ставится на место редактируемого элемента в исходной таблице до формирования предсказывающей подматрицы. В этом случае проверяемый элемент таблицы не участвует в вычислении мер сходства при отборе строк- и столбцов-аналогов.

Возможные режимы работы программы для варианта редактирования приведены ниже и в особых пояснениях не нуждаются.

Режимы работы программы:

1 - редактирование всех элементов таблицы,

2 - редактирование элементов с заданными координатами,

3 - редактирование элементов столбца с заданным номером,

4 - редактирование элементов строки с заданным номером,

5 - формирование предсказывающих подматриц для заданных элементов.

Чтобы можно было использовать не только прямую, но и обратную зависимость между признаками, вычисляется величина расстояний D от признака X_o до признака X_j и его обратного значения $(X_{max,j}-X_j)$ и выбирается минимальное из них:

$$\min\ D(X_o,X_j),\ D(X_o,(X_{max,j}-X_j))\ ,$$

где $D(X_o,X_j)$ - мера различия между j-м и нулевым столбцами, $X_{max,j}$ - максимальное значение в столбце X_j, X_o - столбец с предсказываемым элементом.

Количество ближайших строк и столбцов для формирования предсказывающих подматриц задается пользователем. При отсутст-

вии надежных аналогов требуемое количество строк может быть программой сокращено.

Иногда используется дополнительное ограничение, заключающееся в том, что в группу аналогов включаются только такие строки, значения элементов которых в редактируемом столбце отличаются не более чем на заданный процент от редактируемого элемента.

Режимы печати:

0 - программа работает без печати;

1 - печатается информация о редактируемых элементах;

2 - дополнительно к режиму 1 печатается матрица исходных данных;

3 - дополнительно к режиму 1 печатаются номера отобранных строк и столбцов для формирования подматрицы (в режимах работы программы 2,3,4,5). Пре редактировании элементов одного столбца в режимах работы программы 2,3,5 выдаются на печать также списки номеров строк и признаков (в исходной нумерации) и частота их встречаемости в предсказывающих матрицах;

4 - печатается вся информация, перечисленная в предыдущих режимах.

Возможно словесное документирование результата формирования предсказывающих подматриц. Программа нумерует объекты и показатели и пользуется при работе этими номерами. Если есть необходимость, то по этим номерам могут быть восстановлены исходные наименования объектов и признаков. Наименования должны храниться в файле WORK.INF в следующем порядке: имена всех объектов, а затем всех признаков; каждое имя длиной не более 70 символов пишется с новой строки.

Допустимый процент ошибки для режимов редактирования задает ограничение для выдачи на печать - информация выдается только об элементах, для которых расхождение между исходным и предсказанным превышает указанный процент.

Заполнение пробелов.

Режимы заполнения:

1 - все значения, отмеченные символом пробела, заполняются независимо друг от друга;

2 - предсказываются все элементы, отмеченные символом пробела; заполнение каждого пробела ведется с учетом всех предсказанных к этому моменту элементов;

3 - заполняются только элементы с указанными координатами независимо друг от друга;

4 - заполняются только элементы с указанными координатами. Заполнение каждого пробела ведется с учетом всех предсказанных к этому моменту элементов.

Особенности работы программы в каждом из вариантов достаточно понятны. Обращаем ваше внимание на различие решений с независимым заполнением и с учетом уже предсказанных элементов. В первом случае, независимо от того, выполнено или нет предсказание какого-либо элемента таблицы, в самой таблице на его месте остается символ пробела до окончания прогнозирования всех требуемых значений. Таким образом, каждый пробел заполняется только на основе информации, данной в исходной таблице. Во втором варианте каждое полученное значение сразу же помещается в таблицу и учитывается при прогнозировании последующих элементов.

Режим печати:

0 - программа работает без печати;

1 - печатается информация о заполняемых элементах;

2 - дополнительно к режиму 1 печатаются матрица исходных данных и матрица результатов;

3 - дополнительно к режиму 1 печатаются номера отобранных строк и столбцов для формирования подматрицы;

4 - печатается вся информация, перечисленная в предыдущих режимах.

Допустимый процент ошибки в режимах заполнения задает ограничение на ожидаемый допустимый процент ошибки - если ожидаемая ошибка предсказания превысит указанный процент, то элемент не будет предсказан.

Для оценки величины предсказания пропущенного элемента используются результаты предсказания (в режиме редактирования) всех известных элементов столбца и строки, на пересечении которых стоит пропущенный элемент.

Если получаемый массив нужен для дальнейшей работы,то надо сообщить об этом системе и задать имя файла для записи. Под таким же именем будет сформирован и файл описания данной таблицы с расширением .PET.

Прогнозирование. Мы уже говорили о том, что поиск значения, которое должно быть поставлено в таблицу на место пробела, есть прогнозирование этого значения. Естественно, что со временем алгоритм стал активно применяться для решения задач прогноза. Был разработан и успешно применяется для прогнозирования целый ряд алгоритмов и программ, использующих в качестве основного функционального блока программу ZET.

Довольно часто возникает проблема, например, при решении задач планирования, фактически предсказать значение ряда показателей для "нового объекта" - следующего года или даже ряда последующих лет.Оказалось,что эти задачи вполне разрешимы с помощью ZET-метода. Если известен ряд планируемых показателей,то задача довольно легко сводится к заполнению пробелов в столбце целевого показателя. Несколько сложнее ситуация в случае, когда надо предсказать,по существу,"пустую строку" таблицы. Но и эта задача разрешима, если есть соответствующая ретроспективная информация. Если события не случайны и существуют взаимосвязи между предыдущей и последующей информациями, то удается выявить и использовать такие закономерности и на их основе выполнить прогноз.

Рассмотрим такую ситуацию подробнее. Пусть имеются ретроспективные данные за непрерывный ряд M временных интервалов (например, за M лет) о N характеристиках для одних и тех же объектов.

Исходная (старая) таблица

Объект (строка)	Признак (столбец)				
	1	2	...	N-1	N
1	1,1	1,2	...	1,N-1	1,N
2	2,1	2,2	...	2,N-1	2,N
...	...	...	...	...	...
i	i,1	i,2	...	i,N-1	i,N
...	...	...	...	...	...
M-1	M-1,1	M-1,2	...	M-1,N-1	M-1,N
M	M,1	M,2	...	M,N-1	M,N

Например, рассмотрим таблицу данных урожайности зерновых за 14 последовательных лет (1976-1990 гг.) для 9 территорий РСФСР.

Год	Территория								
	1	2	3	4	5	6	7	8	9
1	20.0	19.4	21.2	17.5	18.5	18.8	18.6	21.5	20.1
2	21.0	15.6	17.9	15.2	12.8	13.4	7.1	11.9	17.9
3	40.8	22.8	25.3	21.9	24.2	24.0	45.5	38.8	27.0
.	...	...	...	...	...	...	...	...	...
12	23.0	22.5	28.4	27.0	20.9	21.5	21.4	18.7	24.3
13	13.5	19.8	25.4	21.4	26.5	23.0	23.8	20.2	20.9
14	25.0	18.9	31.3	27.1	29.7	28.5	23.4	19.6	28.6

Клеточка таблицы, расположенная на пересечении i-й строки и j-го столбца, обозначается символом "x(i,j)", где первый индекс всегда будет соответствовать номеру строки, а второй -

номеру столбца. Так, индексами (2,1) обозначено значение первого показателя для второго объекта (строки таблицы), а индексами (M,N) - значение N-го показателя для M-го объекта.

В качестве объекта в нашей таблице выступает некоторый временной интервал (здесь - год), а признаками являются значения производственного показателя (урожайности) для N территорий.

Чтобы предсказать их значения на несколько шагов вперед, прежде всего необходимо переформировать исходные данные по одному из следующих двух способов.

Способ 1.

Обозначим через K количество временных интервалов, периодов ("старых" строк, т.е. строк исходной таблицы) в строке новой таблицы.

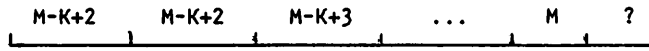
В первой строке новой таблицы разместим K первых строк (периодов) из старой. Во второй строке - K старых строк, начиная со второй, в третьей - начиная с третьей и т.д. Обратите внимание на то, что номера в новой таблице обозначают номера строк старой таблицы. За каждым таким номером стоит содержимое всей соответствующей строки, т.е. значения всех N размещенных в ней показателей.

N п/п	Старая строка номер ...				
1	1	2	...	K-1	K
2	2	3	...	K-2	K-3
...	...	...	...	...	...
i	i	i+1	...	i+K-2	i+K-1
...	...	...	...	...	...
M-K+1	M-K+1	M-K+2	...	M-1	M
M-K+2	M-K+2	M-K+3	...	M	?

Так как в строке старой таблицы было N элементов, а строка новой таблицы содержит K прежних строк, то в ней будет уже (K*N) чисел. При таком формировании, как легко видеть, новая

таблица будет состоять из меньшего числа строк. В итоге мы получим новую таблицу из (K*N) столбцов и (M-K+2) строк. На место последних N элементов должны быть поставлены символы пробела (GAP), которые будут затем предсказаны (заполнены) программой ZET.

Чтобы облегчить восприятие, рассмотрим небольшой пример. Но сначала несколько слов об интерпретации информации, имеющейся в новой таблице. Если рассматривать ее последнюю колонку в качестве целевой, то можно легко заметить, что мы имеем в каждой строке информацию о "результате" в К-м году и описание ситуации за предшествующие ему К-1 лет. В последней строке



результат нам пока неизвестен. Если верна гипотеза о том, что результат зависит от предшествующей ситуации, то программа по всей таблице выявит такие закономерности и на их основе выполнит нужный нам прогноз. Естественно, качество прогноза будет зависеть от того, насколько сильны эти взаимосвязи.

Возьмем приведенную выше таблицу урожайностей и построим на ее основе новую таблицу для К = 2.

Год	Территория							
	1	2	...	9	1	2	...	9
1-2	20.8	19.4	...	20.1	21.0	15.6	...	17.9
2-3	21.0	15.6	...	17.9	40.8	22.8	...	27.0
...	...	...	...	...	...	...	...	...
12-13	23.0	22.5	...	24.3	13.5	19.8	...	20.9
13-14	13.5	19.8	...	20.9	25.0	18.9	...	28.6
14-15	25.0	18.9	...	28.6	?	?	?	?

В строках этой таблицы находится информация уже о данных по урожайности за два последовательных года, а в последней

строке на месте нового, пятнадцатого года, будут поставлены символы пробела. Заполнив эти пробелы, мы получим прогноз на следующий год на основе закономерностей, выявленных на данных за предыдущие периоды.

Способ 2.

Каждый столбец старой таблицы (из M строк) нормируется к интервалу [0,1], и программа добавляет к этой таблице (M+1)-ю строку, состоящую из символов пробела.

Исходная (старая) таблица

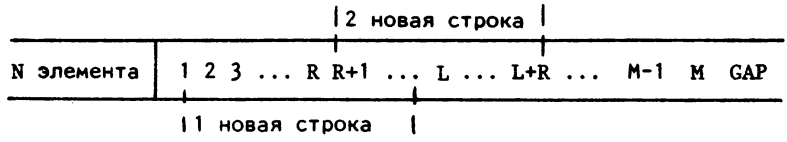
Номер строки	Признак номер ...				
	1	2	...	N-1	N
1	1,1	1,2	...	1,N-1	1,N
2	2,1	2,2	...	2,N-1	2,
...	...	...	...	...	...
i	i,1	i,2	...	i,N-1	i,N
...	...	...	...	...	...
M-1	M-1,1	M-1,2	...	M-1,N-1	M-1,N
M	M,1	M,2	...	M,N-1	M,N
M+1	GAP	GAP	...	GAP	GAP

Затем из каждого столбца вместе с символом пробела программа конструирует новую таблицу данных следующим образом. Рассмотрим столбец старой таблицы для первого признака:

Номер элемента 1 2 3 4 5 6 7 ... i ... M-1 M M+1(=GAP)

Пусть мы хотим создать новую таблицу при следующих условиях: новая строка должна быть длины L и каждая следующая строка должна начинаться на R элементов позже, чем предыдущая. Таким образом, первая строка будет состоять из первых L элементов старого столбца, вторая строка будет содержать также L элементов старого столбца, но начиная с (R+1)-го элемента и

т.д. Учтите, что длина строки (количество элементов в строке) должна быть не меньше константы сдвига (R):



Новая таблица для одного исходного столбца

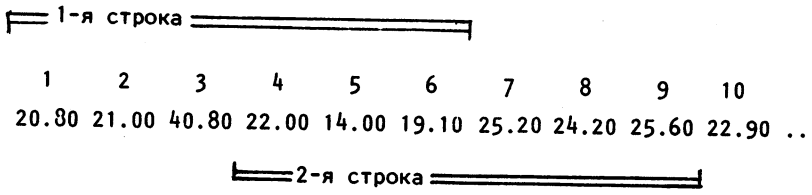
Строка	Элемент исходного столбца				
1	1	2	...	L-1	L
2	R+1	R+2	...	R+L-1	R+L
...	...	...	...	...	...
j	jR+1	jR+2	...	jR+L-1	jR+L
...	...	...	...	...	...
M1	M-L	M-L-1	...	M	GAP

Если число M_1 новых строк, построенных на основе одного столбца, не целое, то тогда первые элементы исходного столбца не включаются в новую таблицу.

Берем 1-й столбец начальной таблицы рассматриваемого примера:

1	2	3	4	5	6	7	8	9	10
20.80	21.00	40.80	22.00	14.00	19.10	25.20	24.20	25.60	22.90
11	12	13	14	15					
25.30	23.00	13.50	25.00	?					

Условия: новая строка должна быть длины 6 и каждая следующая строка должна начинаться на 3 элемента позже, чем предыдущая.



В данном случае используются все элементы исходного столбца, из которого при заданных условиях получено 4 строки с пробелом в нижнем правом углу:

1	2	3	4	5	6
20.80	21.00	40.80	22.00	14.00	19.10
22.00	14.00	19.10	25.20	24.20	25.60
25.20	24.20	25.60	22.90	25.30	23.00
22.90	25.30	23.00	13.50	25.00	?

Таблицы,построенные на основе каждого из N столбцов исходной таблицы, соединяются вместе одна под другой, и таким путем мы получаем окончательную новую таблицу, которая содержит M1*N строк, по L элементов в каждой. Среди всех M1*N*L элементов имеется N пробелов (GAPS) - по одному на каждый старый столбец - которые должны быть предсказаны программой ZET.

Так как в нашей исходной таблице было 9 столбцов, а на основе каждого из них построено по 4 новых строки, содержащих по 6 элементов, то в итоге мы получаем таблицу из 36 строк и 6 столбцов.

ВНИМАНИЕ! Если вы используете этот вариант переформирования, установите 0 для режима печати!

Количество периодов, включаемых в новую строку, выбирается только на основе содержательных соображений и существа задачи. Следует иметь в виду, что программа будет искать законо-

мерности между новыми строками, т.е. прогноз будет выполняться на основе взаимосвязей, выявленных на периодах указанной вами длины.

По аналогии с основными режимами программы ZET и в этих случаях прогноз заданного количества временных периодов (строк исходной матрицы) может осуществляться как в режиме чистого прогнозирования ("вперед"), так и в режиме, аналогичном редактированию - в режиме ретроспективной проверки ("назад").

При прогнозировании "вперед" вычисляются значения для заданного количества дополнительных строк. Возможные варианты приведены в таблице:

Режимы прогнозирования

1. Прогнозируются все значения одного очередного периода независимо друг от друга.
   Затем вычисленные значения вносятся в таблицу и считаются известными. Процедура повторяется для заданного количества периодов.
2. При прогнозировании значений очередного периода вычисление каждого нового значения ведется с учетом всех предсказанных к этому моменту и сразу же заносится в таблицу.

При вычислении "назад" заданное количество известных последних строк исходной таблицы будет программно заменено символами пробелов, и после их заполнения будет проведено сравнение с истинными значениями. Последний вариант прогнозирования полезно использовать для выбора оптимальных параметров работы программы, дающих наименьшую ошибку при сравнении известных и вычисленных значений.

При ретроспективной проверке ("назад") вы можете выполнить два варианта:

Режимы прогнозирования

1. Элементы каждого периода предсказываются независимо, т.е. только на основании исходных данных.
2. Элементы каждого периода предсказываются с учетом уже полученных результатов для предыдущих периодов.

Немного поясним разницу в применении этих вариантов. Если вы хотите, например, получить решение для 10 периодов и будете прогнозировать каждый период независимо от предыдущих, то тем самым вы решите задачу о том, каковы результаты 10 прогнозов, выполненных на 1 период вперед. Если же вы будете прогнозировать каждый период с учетом полученных результатов, то в итоге вы получите ответ на вопрос о том, как выполнен 1 прогноз на 10 лет вперед.

Поскольку все эти программы используют в качестве основного функционального блока программу ZET, то при прогнозировании каждого конкретного значения сохраняются и предусмотренные в программе ZET возможности их независимого заполнения или с учетом заполнения предыдущих (см. режимы заполнения пробелов).

6.5.3. Пояснения к решениям по программе ZETM1.

Задача этого раздела - помочь в обращении с программой и ознакомить с типом получаемых сведений.

Редактирование (проверка) таблицы алгоритмом ZET.

Для работы в режиме редактирования программе требуется 16 параметров. Мы приведем их полный список, но напоминаем, что система задает в каждом конкретном случае только те вопросы,которые нужны для требуемого варианта:

1 по какой программе выполняется счет: ZETM1 - 1, ZETM3 - 2
2 режим редактирования: 1 - 5 (смотрите комментарии)
3 количество ближайших столбцов для формирования предсказывающих подматриц
4 учет обратных зависимостей при выборе ближайших столбцов: 1 - есть, 0 - нет

5 количество ближайших строк для формирования предсказывающих подматриц
6 учитывать ли ограничение для выбора строк-аналогов: 0 - нет, 1 - да
7 процент ограничения для выбора аналогов
8 режим печати: 0-4
9 нужно ли словесное документирование ? (0 - нет, 1 - да)
10 допустимый процент ошибки
11 количество редактируемых элементов
12 номер редактируемого столбца
13 номер редактируемой строки
14 имя файла с номерами столбцов редактируемых элементов
15 имя файла с номерами строк редактируемых элементов
16 учитывать ли редактируемый элемент при выборе ближайших (0 - нет, 1 - да)

Все возможные варианты меню, предлагаемых системой в режиме редактирования, могут быть получены из основного списка с помощью приведенной ниже таблицы.

Параметр	Значение	Не задаются вопросы
2	1	9,11,12,13,14,15,16
	2	12,13
	3	11,13,14,15
	4	11,12,14,15
	5	12,13,16
6	0	7
8	0	9
	1	9
	2	9

Рассмотрим таблицу данных урожайности зерновых за 14 последовательных лет (1976-1990 гг.) для 66 территорий РСФСР.

Как и все программы БАД, программа ZET прежде всего выдаст сведения о тех условиях, при которых будет выполняться решение, например:

ПРОГРАММА ZET (РЕЖИМ М1)

Значения параметров режима:

количество признаков - 66; объектов - 14; редактируемых элементов - 14; подматрица: из 4 столбцов, 4 строк без учета обратных зависимостей

Обозначение пробела - .9000E+18

Выдается на печать информация только о тех элементах, для которых расхождение прогнозного и исходного значения выше .1000E-02%.

РЕДАКТИРОВАНИЕ

без учета редактируемого элемента

Итак, программа при указанных условиях будет выполнять редактирование. В этом режиме на печать будут выдаваться сведения только о тех элементах, для которых расхождение вычисленного и реального значений превышает заданный порог. При редактировании мы рекомендуем задавать его на всякий случай очень маленьким - 0.02, чтобы получить полные сведения.

Редактируемый элемент обязательно сопровождается информацией о его месте в исходной таблице - всегда будет указан номер столбца и строки, на пересечении которых этот элемент расположен.

Для того чтобы получить наиболее полную информацию, зажжем режим печати с выдачей данных о предсказывающих подматрицах, использованных программой для вычислений предполагаемых значений. Пример такой информации приведен ниже.

Информация о строках и столбцах,отобранных для прогнозирования

Номера объектов-аналогов:	1	10	4	8
Номера ближ.показателей:	4	8	7	6
Номера объектов-аналогов:	2	5	1	4
Номера ближ.показателей:	4	3	9	1

... ...

```
: Среднее расстояние между:  строками  .279,  столбцами   .242 :
        Среднее расстояние меняется от нуля до единицы.
     При отсутствии надежной информации для прогнозирова-
                  ния отбирается не более 5 строк.
```

В этой таблице объекты и признаки представлены их номерами. Такое представление компактно и достаточно удобно,но при желании, как вы помните, есть возможность получить те же сведения в более привычном, словесном варианте (если задан файл WORK.INF с наименованиями объектов и признаков). Пример такой информации приведен ниже.

```
Перечень аналогов, отобранных для объекта (признака),
    расположенного на первом месте в каждой группе

     1 - 1976
    10 - 1985
     4 - 1979
     8 - 1983
  ------------------
     4 - КУРСКАЯ
     8 - ЧЕЧЕНО-ИНГУШСКАЯ
     7 - ДАГЕСТАНСКАЯ
     6 - РОСТОВСКАЯ
  ==================
    ...
```

В большом списке номеров достаточно трудно ориентироваться и делать какие-либо обобщающие выводы. Поэтому программа подводит некоторые итоги о частоте встречаемости признаков в предсказывающих подматрицах.

```
     Информация, упорядоченная по убыванию значений
     частоты встречаемости признаков среди ближайших

                Представление информации:

   1  Номера признаков в программе
   2  Частота встречаемости признаков среди ближайших
   3  Среднее место среди ближайших

 1      4      9      7      6  ...      2      1      8      5
 2     66     53     47     37  ...      5      5      3      1
 3   1.00   3.00   3.29   3.71  ...   2.40   3.20   2.33   4.00
```

Так как мы выполняем редактирование элементов 4-го столбца, то номер этого столбца появится в подматрице М раз (в данном случае 66).Для прогнозирования значений 4-го столбца соответственно 53,47 37 раз (из 66) были использованы данные 9-го, 7-го, 6-го.столбцов,по 5 раз - столбцов 2 и 1, 33 раза - 8-го и только 1 раз - 5-го,к тому же только на последнем,4-м месте. Это позволяет судить об относительной важности признаков для предсказания элементов редактируемого столбца.

Иногда можно получить полезные сведения и из анализа встречаемости объектов среди ближайших, поэтому программа выдает такую информацию, упорядоченную по убыванию значений частоты встречаемости. Представление информации такое же, как и для признаков.

В режиме редактирования программа выдает следующие сведения:

№ столбца	№ строки	Процент ошибки (ожидаемый)	Предсказанное значение	Исходное значение	Процент отклонения (фактический)
4	4	8.5674	18.501	18.900	2.1129
...	...	...	...	...	...
4	8	3.0537	17.915	18.600	3.6854
...	...	...	...	...	...
Средние:		4.8079	19.745	19.136	2.009

6.5.4. Заполнение пробелов.

Для работы в режиме заполнения программе требуется 12 параметров. Приводим их полный список:

1 по какой программе выполнять счет: ZETM1 - 1, ZETM3 - 2
2 режим заполнения: 1 - 4
3 количество ближайших столбцов для формирования предсказывающих подматриц
4 учет обратных зависимостей при выборе ближайших столбцов: 1 - есть, 0 - нет

5 количество ближайших строк для формирования предсказывающих подматриц
6 режим печати: 0-4
7 допустимый процент ошибки
8 количество пробелов
9 имя файла с номерами столбцов заполняемых элементов
10 имя файла с номерами строк заполняемых элементов
11 нужно ли сохранить полученную таблицу в файле? (0 - нет, 1 - да)
12 имя файла, в который будет записан заполненный массив

Возможные варианты меню для режима заполнения пробелов:

Параметр	Значение	Не задаются вопросы
2	1	9, 10
	2	9, 10
11	0	12

Работа с программой в режиме заполнения пробелов по сравнению с режимом редактирования практически не имеет дополни - тельных сложностей и особых пояснений не требует. Обратите внимание на то, что при заполнении предсказываются только те элементы, для которых ожидаемая ошибка не больше указанного порогового значения.

Выдача результата также похожа на то, что вы видели при редактировании. Естественное различие заключается только в том, что не может быть сравнения полученных значений с фактическими.

№ п/п	№ столбца	№ строки	Процент ошибки (ожидаемый)	Предсказанное значение
1	19	10	7.2159	25.498
...	...	...	...	...
7	25	10	2.3357	22.185
8	26	10	1.6439	22.477
9	27	10	5.8508	22.073

ЗАПОЛНЕНО 9 ПРОБЕЛОВ, НЕ ЗАПОЛНЕНО 0
Средний ожидаемый процент ошибки = 6.461

Прогнозирование.

Для работы в режиме прогнозирования программе требуется 16 параметров.

1 по какой программе выполнять счет: ZETM1 - 1, ZETM3 - 2
2 способ переформирования таблицы для прогноза: 1 или 2 (см.комментарии)
3 количество периодов в формируемой строке
4 на сколько периодов будет прогноз
5 как осуществляется прогноз: вперед - 0, назад - 1
6 прогнозировать периоды независимо (0), с учетом предыдущих (1)
7 константа сдвига (R)
8 количество элементов в формируемой строке
9 режим прогнозирования: 1 - 2 (см. комментарии)
10 количество ближайших столбцов для формирования предсказывающих подматриц
11 учет обратных зависимостей при выборе ближайших столбцов: 1 - есть, 0 - нет.
12 количество ближайших строк для формирования предсказывающих подматриц
13 режим печати: 0 - 4 (см.комментарии)
14 допустимый процент ошибки
15 нужно ли сохранить полученную таблицу в файле? (0 - нет, 1 - да)
16 имя файла, в который будет записан заполненный массив

Возможные варианты меню для режима прогнозирования:

Параметр	Значения	Не задаются вопросы
2	1	7, 8
	2	3
5	0	6
15	0	16

Чтобы предсказать значения на несколько шагов вперед, прежде всего необходимо переформировать исходные данные по одному из 2-х способов. Рассмотрим на тех же данных вариант прогнозирования "на 3 периода назад" (из 14), переформировав таблицу по 1-му способу.

Зададим количество временных интервалов, периодов ("старых" строк, т.е. строк исходной таблицы) в строке новой таблицы K = 3. В первой строке новой таблицы будет 3 первых строки (периода) из старой. Во второй строке - 3 старых строки, начиная со второй, затем начиная с третьей и т.д. Если, например,в исходной таблице было 9 столбцов и 14 строк, то в новой столбцов будет 27, а строк - 12.

Программа информирует о варианте работы:

```
Прогноз выполняется на 3 периода  назад

В таблицу вносятся исходные значения
 В строке  новой таблицы 3 периода

      ПРОГРАММА ZET (РЕЖИМ M1)
    Значения  параметров режима
```

Программа выдает достаточно полную информацию, которая, как обычно, хранится в файле tab1.dat и может быть сохранена в указанном вами файле без изменений или после обработки в редакторе. Результаты прогноза также хранятся в файле PROG.ZET.

Прогноз выполняется на 12-й период

Номер признака	Прогноз (B)	Исходное значение (A)	B-A	Относит. ошибка,%	Ср. % ошибки за все периоды
1	22.37	20.90	1.473	7.046	7.046
2	22.18	21.40	.7847	3.667	3.667
		Средний процент ошибки = 5.85			5.85

Прогноз выполняется на 13-й период

. . . . . .

		Средний процент ошибки = 4.61			5.23

Прогноз выполняется на 14-й период

Номер признака	Прогноз (B)	Исходное значение (A)	B-A	Относит. ошибка,%	Ср. % ошибки за все периоды
1	19.28	18.90	.3759	1.989	4.172
2	25.83	27.10	-1.267	4.675	10.04
3	22.74	23.40	-.6617	2.828	3.64
	Средний процент ошибки =			3.45	4.64

Поскольку в последнем столбце дается информация о среднем проценте ошибки за все спрогнозированные периоды, то в первом из предсказываемых периодов числа в двух последних колонках будут совпадать, а далее будет выполняться усреднение.

Теперь рассмотрим на тех же данных вариант прогнозирования "на 3 периода назад", переформировав таблицу по 2-му способу.

Построим новую таблицу при следующих условиях:

Константа сдвига = 1
Количество элементов в формируемой строке = 7
В исходной таблице 14 объектов (строк), 9 признаков (столбцов)
В прогнозируемой подматрице 5 строк, 5 столбцов
Прогноз выполняется на 3 периода назад
В таблицу вносятся исходные значения

Берем 1-й столбец начальной таблицы:

1	2	3	4	5	6	7	8	9	10
20.80	21.00	40.80	22.00	14.00	19.10	25.20	24.20	25.60	22.90

11	12 ?	13 ?	14 ?
25.30	23.00	13.50	25.00

Для предсказания 12-го периода получим на основе 1-го столбца табличку с символом пробела в нижнем правом углу.

20,80	21.00	40.80	22.00	14.00	19.10	25.20
21.00	40.80	22.00	14.00	19.10	25.20	24.20
40.80	22.00	14.00	19.10	25.20	24.20	25.60
22.00	14.00	19.10	25.20	24.20	25.60	22.90
14.00	19.10	25.20	24.20	25.60	22.90	25.30
19.10	25.20	24.20	25.60	22.90	25.30	?

Так как в исходной таблице было 9 столбцов, то окончательно новая таблица будет иметь 54 строки и 9 пробелов. По условиям решения каждый период будет предсказываться независимо (в таблицу вносятся исходные значения). После предсказания всех значений очередного периода программа будет строить новую таблицу для новых вычислений. Форма выдачи результата такая же, как и в предыдущем случае.

З а к л ю ч е н и е

Как уже было сказано во введении, Блок Анализа Данных является одной из составных частей экспертной системы ЭКСНА - инструментального комплекса, предназначенного для построения прикладных экспертных систем, обладающих некоторыми элементами систем второго поколения или "партнерских" систем. Одним из элементов,повышающих уровень интеллекта системы, и является Блок Анализа Данных, который работает с данными, представленными в виде таблицы "объект-свойство".

Система ЭКСНА снабжена конструктором диалога, простым в обучении и легким в пользовании. Входящий в систему генератор диалоговых интерфейсов дает пользователю возможность создавать собственный сценарий диалога, не прибегая к услугам программистов.

При создании инструментальной экспертной системы ЭКСНА впервые реализованы новые оригинальные алгоритмы логического вывода. Логический вывод моделирует человеческие рассуждения по аналогии и базируется на специально разработанной мере бли-

зости в пространстве знаний. Знания в нашей системе представ - ляют собой набор продукций вида: "Если А, то В", где А и В - некоторые элементарные высказывания, которые могут содержать в себе "фактор неопределенности".

С использованием этой меры близости реализован на IBM PC прямой вывод, работает блок поиска противоречий, позволяющий выявлять ошибки при вводе информации в Базу Знаний.

Одной из важнейших особенностей системы ЭКСНА является возможность автоматического извлечения знаний из данных и проверка имеющихся в Базе Знаний сведений на новых фактических данных. Знания, полученные программным путем, уже оформлены в виде, требуемом для ввода в Базу Знаний, и не нуждаются в дальнейшей обработке.

Блоки системы ЭКСНА, реализующие названные выше функции, будут подробно рассмотрены в последующих публикациях.

Поступила в ред.-изд.отд.
10 октября 1991 года

Файл WORK.DAT

12.00	11.80	20.00	14.20	11.90	9.90	14.10
12.20	15.80	14.10	20.90	19.10	7.60	13.20
13.80	15.20	20.80	14.00	8.20	13.00	18.40
15.00	19.10	16.10	21.10	15.50	8.30	14.00
10.90	9.80	21.10	16.40	10.50	6.80	10.20
8.50	11.90	12.00	16.00	14.00	7.90	7.00
20.80	21.00	40.80	22.00	14.00	19.10	25.20
24.20	25.60	22.90	25.30	23.00	13.50	25.00
9.90	14.60	19.40	11.90	9.00	10.00	11.80
11.40	14.20	11.30	13.90	11.10	6.60	14.70
9.90	9.50	19.30	11.40	7.40	8.70	12.30
11.20	14.00	11.30	11.90	11.10	7.80	15.00
11.80	13.50	20.40	15.50	12.80	10.10	12.30
10.80	13.20	11.10	12.70	19.30	14.70	19.40
16.20	13.30	22.80	14.80	12.10	7.30	17.70
15.80	15.60	15.60	17.50	21.90	16.80	17.50
13.60	13.00	22.00	16.20	10.60	6.10	16.10
14.90	14.90	12.50	16.50	19.60	15.10	14.00
10.90	9.80	22.90	17.50	.6.90	10.50	13.80
13.00	13.90	11.40	16.70	14.40	9.10	14.70
11.60	8.30	19.20	17.10	10.60	7.40	12.50
11.70	12.80	11.90	14.20	20.70	14.50	15.10
11.00	10.70	19.30	11.90	9.10	7.50	13.10
13.00	12.80	9.90	14.60	13.60	9.20	10.90
22.30	24.50	41.60	27.10	19.50	12.30	22.30
21.70	25.90	25.40	28.40	36.10	26.50	28.10
12.40	10.60	16.10	17.10	10.20	7.60	14.40
14.20	15.30	12.40	17.10	22.20	18.60	22.30
12.10	9.60	19.40	15.40	17.00	4.30	15.40
13.10	11.70	13.20	13.70	21.40	15.10	20.20
12.80	9.90	17.90	14.90	7.50	8.80	12.80
12.10	15.10	12.10	13.60	15.70	10.00	13.60
14.00	12.20	18.50	18.20	11.80	9.70	19.40
16.70	20.20	17.10	20.90	26.80	18.30	19.60

Файл WORK.DAT (продолжение 1)

12.90	17.20	23.50	10.90	11.70	10.10	16.10
13.30	14.80	12.20	17.40	12.80	8.80	13.80
13.40	12.70	19.90	14.90	11.70	7.40	17.10
18.00	13.00	15.60	15.20	19.90	13.20	15.70
11.60	7.10	16.70	18.20	8.80	6.30	14.60
12.10	11.70	12.90	17.80	13.70	10.00	9.30
16.60	13.30	18.50	19.00	11.20	6.90	19.40
17.70	18.00	19.40	22.60	15.80	11.40	14.10
12.30	13.70	18.30	17.90	12.40	7.80	17.70
17.20	10.10	14.40	13.00	18.30	13.30	16.90
19.40	15.60	22.80	22.50	15.70	11.00	26.80
23.50	18.30	22.60	23.00	22.50	19.80	18.90
21.10	17.90	25.30	23.10	13.30	14.00	21.00
22.50	14.00	18.40	24.30	28.40	25.40	31.30
18.70	15.90	30.80	40.20	16.60	11.60	15.20
20.40	7.50	17.60	17.10	25.10	23.00	27.40
17.50	15.20	21.90	18.90	12.80	12.20	17.60
18.60	16.30	19.30	22.10	27.00	21.40	27.10
14.40	13.10	20.40	17.80	12.10	7.20	15.90
16.60	12.40	15.60	17.60	24.40	22.00	23.00
15.70	14.10	19.00	22.10	13.20	6.40	19.70
20.90	9.30	17.00	15.00	23.70	18.00	20.40
10.40	10.20	55.60	8.40	10.50	11.10	9.20
11.90	9.60	13.70	11.50	16.20	15.10	16.30
13.60	9.70	19.60	14.00	8.60	8.20	10.20
13.10	4.10	14.10	10.60	13.20	18.50	18.40
15.10	13.80	25.10	18.50	8.90	6.80	12.50
19.00	10.80	16.20	19.30	11.00	10.50	16.10
13.70	12.50	18.00	16.00	13.50	7.50	14.50
16.90	7.80	14.40	11.70	17.30	12.90	12.90
12.20	10.10	20.70	14.20	7.90	6.30	11.90
15.80	5.30	13.50	11.00	10.20	9.30	14.80
16.00	13.80	15.50	18.60	14.40	8.30	19.90

Файл WORK.DAT (продолжение 2)

19.70	9.70	17.10	18.30	17.00	17.20	17.50
13.10	6.70	20.80	10.70	4.70	14.90	10.10
8.10	9.10	7.80	7.40	12.60	12.50	21.20
14.00	14.00	19.40	20.00	9.60	7.80	19.50
18.10	14.60	18.30	20.90	11.40	12.10	14.30
33.60	19.00	35.60	26.20	14.00	31.10	34.20
32.20	37.00	29.80	41.40	38.70	37.70	41.00
18.50	12.80	24.20	15.00	9.40	25.80	20.20
19.10	21.10	14.30	26.00	20.90	26.50	29.70
18.80	13.40	24.00	17.80	8.90	16.70	17.80
15.70	16.30	17.40	18.00	21.50	23.00	28.50
18.60	7.10	45.50	17.00	14.70	19.10	16.90
17.70	19.40	20.30	23.60	21.40	23.80	23.40
30.80	18.00	25.70	22.70	10.90	29.50	31.70
31.70	34.90	33.60	34.50	33.30	36.50	36.20
28.00	18.40	25.50	22.90	9.10	30.80	30.60
30.00	33.30	31.00	34.90	34.20	32.40	32.50
21.50	11.90	38.80	17.50	9.10	21.60	20.30
16.50	20.30	20.20	25.70	18.70	20.20	19.60
17.30	19.40	13.00	19.30	6.20	12.70	16.90
12.50	11.00	17.60	17.10	8.00	11.20	6.90
13.10	9.80	9.10	14.50	4.10	8.40	7.30
14.90	6.90	10.00	15.50	8.30	8.40	12.10
11.70	12.70	16.40	19.10	8.40	6.20	12.60
14.00	13.00	13.60	16.90	11.40	9.90	7.30
18.90	21.70	17.60	17.00	5.90	13.90	19.50
16.30	17.80	17.50	21.40	11.60	13.40	11.30
14.50	14.10	17.60	21.00	6.40	9.70	14.00
13.40	6.20	15.30	19.10	9.30	9.50	9.00
17.30	14.60	14.50	19.40	5.50	10.50	14.00
18.60	14.50	19.80	22.90	8.50	12.40	11.90
12.30	11.20	16.30	18.90	9.50	6.30	13.00
14.90	12.50	14.40	18.30	10.70	9.70	10.00

Файл WORK.DAT (продолжение 3)

11.60	15.30	13.00	16.70	5.90	6.50	8.20
11.60	14.50	14.60	14.70	15.30	12.40	15.60
12.30	19.30	17.90	14.60	6.90	9.70	11.70
14.50	16.70	16.00	16.70	18.70	18.60	18.40
12.10	15.80	13.80	12.70	5.40	8.40	8.10
11.10	16.70	15.40	16.10	15.10	14.30	11.30
15.00	16.60	12.60	17.80	6.40	12.30	9.50
10.30	15.30	16.40	16.80	14.40	12.60	10.10
12.70	21.90	14.40	13.30	10.10	12.30	10.60
13.90	16.00	14.90	16.30	18.70	18.40	15.40
15.40	20.10	14.50	16.20	7.20	15.70	18.40
13.30	16.70	18.30	17.30	12.40	15.70	9.90
11.80	14.30	12.70	13.80	4.70	11.30	16.90
16.30	15.20	15.30	14.40	18.40	17.80	17.00
12.40	12.80	12.00	17.90	8.00	14.90	16.40
15.40	14.50	15.30	13.80	18.80	19.70	21.20
7.40	6.50	9.80	10.40	3.90	8.80	8.30
9.60	6.90	9.40	11.00	12.10	13.90	13.80
6.20	5.30	11.50	9.50	9.20	5.80	11.60
12.80	9.70	13.00	13.10	13.60	18.00	12.20
4.90	4.90	15.40	11.20	6.90	1.50	5.40
6.70	9.70	12.40	12.90	9.00	11.60	7.90
15.50	17.80	20.70	17.20	10.70	10.30	12.00
16.10	13.10	13.20	19.40	14.30	19.90	17.70
13.40	17.50	16.90	18.50	9.10	9.60	6.70
13.20	10.00	12.30	13.30	15.50	14.00	17.20
13.30	15.80	14.10	14.70	7.50	12.30	8.40
9.30	5.10	8.10	6.60	10.00	10.90	16.30
7.30	8.20	13.70	8.20	7.50	9.50	9.30
6.50	9.30	7.20	5.00	10.60	13.30	15.80
20.10	17.90	27.00	22.30	14.00	14.90	19.00
19.50	23.20	20.00	21.10	24.30	20.90	28.60

Файл WORK.PET

ВНИМАНИЕ! Параметры таблицы, расположенные в строках 9, 10, 12, 13 и далее, нужны для работы алгоритмов распознавания и выбора информативных признаков. Эти данные не влияют на работу других алгоритмов и автоматически появляются после режима таксономии.

```
   66     ;количество объектов                          (5)
   14     ;количество признаков                         (6)
    1     ;0-запись по признакам (1-по объектам)        (7)
    0     ;обозначение пропуска:0-нет пробелов          (8)
   -1     ;номер целевого признака                      (9)
    5     ;количество образов                           (10)
    4     ;все признаки в сильной шкале                 (11)
    0     ;индекс упорядочения по образам               (12)
          ;отдельный целевой признак                    (13)
 2 2 1 4 2 2 3 3 2 2 3 2 5 4 4 2 3 2 3 1 2 3 3
 5 5 5 4 4 3 3 3 2 2 3 4 2 5 5 5 4 5 5 3 1 2 1
 2 1 2 1 3 3 2 2 3 1 3 4 2 2 1 3 3 3 3 5
```

Файл WORK.INF

```
1   Архангельская
2   Вологодская
3   Коми
4   Ленинградская
5   Новгородская
6   Псковская
7   Брянская
8   Владимирская
9   Ивановская
10  Калининская
11  Калужская
12  Костромская
13  Московская
14  Орловская
15  Рязанская
16  Смоленская
17  Тульская
18  Ярославская
19  Горьковская
20  Кировская
21  Марийская
22  Мордовская
23  Чувашская
```

Файл WORK.INF (продолжение)

24 Белгородская
25 Воронежская
26 Курская
27 Липецкая
28 Тамбовская
29 Астраханская
30 Волгоградская
31 Куйбышевская
32 Пензенская
33 Саратовская
34 Ульяновская
35 Калмыцкая
36 Татарская
37 Краснодарский
38 Ставропольский
39 Ростовская
40 Дагестанская
41 Кабардино-Балкарская
42 Северо-Осетинская
43 Чечено-Ингушская
44 Курганская
45 Оренбургская
46 Пермская
47 Свердловская
48 Челябинская
49 Башкирская
50 Удмуртская
51 Алтайский
52 Кемеровская
53 Новосибирская
54 Омская
55 Томская
56 Тюменская
57 Красноярский
58 Иркутская
59 Читинская
60 Бурятская
61 Тувинская
62 Приморский
63 Хабаровский
64 Амурская
65 Якутская
66 Калининградская

Файл WORK.INF (продолжение)

```
1  урожайность зерновых во всех категориях хозяйств,ц/га, 1976
2  урожайность зерновых во всех категориях хозяйств,ц/га, 1977
3  урожайность зерновых во всех категориях хозяйств,ц/га, 1978
4  урожайность зерновых во всех категориях хозяйств,ц/га, 1979
5  урожайность зерновых во всех категориях хозяйств,ц/га, 1980
6  урожайность зерновых во всех категориях хозяйств,ц/га, 1981
7  урожайность зерновых во всех категориях хозяйств,ц/га, 1982
8  урожайность зерновых во всех категориях хозяйств,ц/га, 1983
9  урожайность зерновых во всех категориях хозяйств,ц/га, 1984
10 урожайность зерновых во всех категориях хозяйств,ц/га, 1985
11 урожайность зерновых во всех категориях хозяйств,ц/га, 1986
12 урожайность зерновых во всех категориях хозяйств,ц/га, 1987
13 урожайность зерновых во всех категориях хозяйств,ц/га, 1988
14 урожайность зерновых во всех категориях хозяйств,ц/га, 1989
```