МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Омский государственный технический университет»

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ АНАЛИЗА ДАННЫХ

Методические указания

Омск
Издательство ОмГТУ
2015

Составители: А. Г. Белик, к.т.н., доцент;
В. Н. Цыганенко, к.т.н., доцент

Приведены цели, задачи и требования к выполнению лабораторных работ по дисциплине «Информационные технологии анализа данных». Описаны идеология анализа данных, реализованные механизмы, составные части и архитектура; представлены типовые задачи анализа бизнес-данных и способы их решения при помощи Deductor Studio.

Предназначены для студентов, обучающихся по направлениям подготовки бакалавриата: 27.00.00 «Управление в технических системах», 09.00.00 «Информатика и вычислительная техника».

Печатается по решению редакционно-издательского совета
Омского государственного технического университета

ОБЩИЕ ПОЛОЖЕНИЯ

1. Цели и задачи дисциплины

Анализ информации является неотъемлемой частью ведения бизнеса и одним из важных факторов повышения его конкурентоспособности. При этом в подавляющем большинстве случаев анализ сводится к применению одних и тех же базовых механизмов. Они являются универсальными и применимы к любой предметной области, благодаря чему имеется возможность создания унифицированной программной платформы, в которой реализованы основные механизмы анализа, например Deductor Studio.

Обычно анализ производят аналитики и эксперты предметной области предприятия. Они подготавливают данные к пригодному для анализа виду, применяют к ним различные методы анализа, приводят результаты к легко воспринимаемому виду. Результаты анализа необходимы лицам предприятия, принимающим решения, например, руководителям отделов, менеджерам. Они могут совершенно не разбираться в методах анализа, но у них есть потребность в их результатах.

Цель проведения лабораторного практикума по дисциплине «Информационные технологии анализа данных» – изучение корпоративной системы отчетности и многостороннего анализа любого вида деятельности на основе аналитической платформы Deductor Studio 5 Academic компании Base Group.

Deductor Studio Academic является платформой, ориентированной на решение задач анализа самого широкого спектра: от создания систем корпоративной отчетности до решения задач Data Mining.

В Deductor Studio используются самые мощные технологии, такие как многомерный анализ, нейронные сети, деревья решений, самоорганизующиеся карты, спектральный анализ и множество других. При этом акцент сделан на самообучающиеся методы и машинное обучение, что позволяет строить адаптивные системы, способные реагировать на изменение ситуации.

Реализованные в Deductor Studio технологии дают возможность на базе единой платформы пройти все этапы построения аналитической системы: от создания хранилища данных до автоматического подбора моделей и визуализации полученных результатов:

– системы аналитической отчетности;

– многомерный анализ;
– прогнозирование;
– поиск закономерностей;
– управление рисками;
– сегментация клиентов/товаров/услуг;
– построение профилей потребителей;
– оценка эффективности рекламы;
– анализ маркетинговых данных.

Deductor Studio – аналитическое ядро платформы Deductor. Deductor Studio содержит полный набор механизмов импорта, обработки, визуализации и экспорта данных для быстрого и эффективного анализа информации. В нем сосредоточены самые современные методы извлечения, очистки, манипулирования и визуализации данных. С ним становятся доступны моделирование, прогнозирование, кластеризация, поиск закономерностей и многие другие технологии обнаружения знаний *(Knowledge Discovery in Databases)* и добычи данных *(Data Mining)*.

В Deductor Studio включен полный набор механизмов, позволяющий получить информацию из произвольного источника данных, провести весь цикл обработки (очистку, трансформацию данных, построение моделей), отобразить полученные результаты наиболее удобным образом (OLAP, таблицы, диаграммы, деревья решений и т. д.) и экспортировать результаты. Вся работа по анализу данных в Deductor Studio базируется на выполнении следующих действий:

– импорт данных;
– обработка данных;
– визуализация;
– экспорт данных.

Отправной точкой для анализа всегда является процедура импорта данных. Полученный набор данных может быть обработан любым доступным способом. Результатом обработки так же является набор данных, который, в свою очередь, опять может быть обработан. Результаты обработки можно просмотреть множеством способов и экспортировать в наиболее популярные форматы.

Последовательность действий, которые необходимо провести для анализа данных, является сценарием, который можно автоматически выполнять на любых данных. Deductor Studio поддерживает множество источников данных – промышленные СУБД *(Oracle, MS SQL)*, текстовые файлы, офисные приложения *(Excel, Access)*, ADO и ODBC источники,

и полностью интегрировано с многомерным хранилищем данных Deductor Studio Warehouse.

Под обработкой подразумевается любое действие, связанное с преобразованием данных, например, построение моделей, очистка от шумов и аномальных значений. При этом механизмы обработки можно комбинировать произвольным образом так, чтобы достичь наилучшего результата.

Визуализация – это отображение импортированных и обработанных данных. Визуализировать можно любой объект в сценариях обработки. Программа самостоятельно анализирует, каким образом можно отобразить информацию, пользователь должен только выбрать нужный вариант.

Студент, успешно выполнивший лабораторный практикум, приобретает способность применять аналитические, вычислительные и системно-аналитические методы для решения прикладных задач в области управления объектами техники, технологии, организационными системами, работать с традиционными носителями информации, распределенными базами знаний.

Практическая подготовка студентов обеспечит получение ими основных знаний в области методов оперативного и интеллектуального анализа, принципов организации хранилищ данных и построения систем поддержки принятия решений.

В задачи дисциплины входит изучение классификации систем поддержки принятия решений и методов оперативного и интеллектуального анализа данных, формирование базовых знаний по принципам построения систем поддержки принятия решений, развитие практических навыков применения технологий анализа данных.

2. Варианты заданий

Для начала работы с Deductor Studio в программу следует импортировать данные из какого-либо источника. Для этого студенту необходимо создать исходные данные – базу данных.

Данные могут быть представлены в любом стандартном табличном виде: текстовые файлы и файлы DBF, MS Excel, базы данных MS Access, MS SQL, Oracle, InterBase, любой ODBC источник.

Academic версия предназначена только для образовательных целей. В ней ограничены возможности интеграции и автоматической обработки.

Поддерживается только три источника и приемника данных: Deductor Warehouse, Deductor Data File и текстовые файлы.

Варианты возможных видов деятельности, для проведения анализа данных с помощью аналитической платформы Deductor Studio, представлены в табл. 1.

Студент может предложить собственную тематику, если она связана с темой выпускной квалификационной или научно-исследовательской работы. В последнем случае, по завершении выполнения лабораторного практикума, студент обязательно предоставляет черновой вариант научной статьи.

Таблица 1

Варианты заданий к лабораторным работам

№	Сфера деятельности
1	Web деятельность, сайты, информационные отделы
2	Банковские операции и вклады
3	Бухгалтерская отчетность
4	Госавтоинспекция
5	Деканат вуза, студенты бюджетного отделения
6	Клубы любителей животных
7	Магазин «Детский мир»
8	Магазин «Компьютерная техника»
9	Магазин спортивной одежды и инвентаря
10	Налоговая инспекция
11	Отдел по защите прав потребителей
12	Отделы статистических исследований
13	Поликлиники и больницы города
14	Предприятия по бытовому обслуживанию
15	Предприятия по машиностроению
16	Предприятия по энергетике
17	Предприятия по производству хлебобулочных изделий
18	Служба занятости населения
19	Строительная фирма
20	Телефонная кампания
21	Товарищество собственников жилья, жилищно-эксплуатационные организации
22	Транспортные компании города

Исходная информация базы данных содержит в себе атрибуты, в том числе ключевые поля, характеризующие каждую описываемую сущность.

Так, например, сущность «Сотрудники» может иметь такие атрибуты, как индивидуальный номер, фамилия и инициалы, название отдела, в котором он работает.

Студенту необходимо создать как минимум:

1) три файла «Справочника» – информация в справочниках не меняется, она постоянна или может изменяться в редких случаях. В «Справочниках» не может присутствовать дата или другая информация, изменяющаяся во времени. В таких файлах студент создает примерно по 5–10 записей в каждом;

2) три файла «Процесса» – информация, изменяющаяся во времени, ежедневно, ежеминутно. Записей в данном файле не должно быть меньше чем 80, но и не более 150, так как большее количество записей Deductor Studio, используемый для учебной деятельности, не воспримет к анализу.

Пример выполнения лабораторного практикума, приведенный ниже, взят на основе деятельности торгово-закупочного предприятия.

Информация, используемая системой, сохраняется в базе данных, содержащих следующие таблицы, являющиеся источниками данных (табл. 2).

Таблица 2

База данных по торгово-закупочному предприятию

Таблица	Назначение	Поля
Справочник товаров	Измерение	Код товара (число); наименование товара (строка); группа товара (строка); категория товара (строка)
Справочник организаций	Измерение	Код организации (число); наименование (строка); регион (строка); тип организации (строка); категория организации (строка)
Справочник складов	Измерение	Код склада (число); наименование (строка)

Окончание табл. 2

Таблица	Назначение	Поля
Справочник менеджеров	Измерение	Код менеджера (число); фио (строка); отдел (строка)
Продажи	Процесс	Дата продажи (дата); код организации (число); код товара (число); код менеджера (число); количество (число); сумма в руб. (число); себестоимость (число)
Текущие остатки	Процесс	Код склада (число); код товара (число); количество (число); себестоимость (число)
Закупки	Процесс	Дата закупки (дата); код организации (число); код товара (число); количество (число); себестоимость (число)
Поступления	Процесс	Дата поступления (дата); код организации (число); сумма (число)
Оплата	Процесс	Дата оплаты (дата); код организации (число); сумма (число)

При создании таблиц рекомендуется заполнять их данными, характеризующими деятельность конкретного предприятия. Если таких данных нет, таблицы должны содержать тестовые наборы, достаточные для проведения тестирования разрабатываемой аналитической системы.

Образец файла, составленный по рассматриваемому предприятию в MS Excel, приведен на рис. 1.

	A	B	C	D	E	F	G
1	**Дата_прод**	**Код_орг**	**Код_тов**	**Код_мен**	**Кол-во**	**Сумма**	**Стоимость**
2	12.01.2001	2	3	4	6	300	200
3	13.01.2001	1	1	2	4	200	150
4	14.01.2001	1	1	3	7	300	250
5	15.01.2001	2	2	1	4	4000	3400
6	16.01.2001	3	3	5	2	3000	2700
7	17.01.2001	4	4	6	3	2000	1800
8	18.01.2001	5	6	4	4	500	480
9	02.02.2001	4	5	4	6	800	700
10	04.02.2001	3	5	3	5	250	240
11	05.02.2001	2	7	1	8	400	340
12	06.02.2001	2	3	2	3	300	240
13	07.02.2001	3	2	3	4	400	350
14	08.02.2001	4	4	2	5	600	560

Товары | Организации | Склады | Менеджеры | **Продажи** | Остатки | Закупки | Поступления

Рис. 1. База данных в MS Excel. Процесс «Продажи»

После создания исходных данных студент может приступить к непосредственному выполнению лабораторного практикума. Отдельные задачи, решаемые аналитической платформой, представляют собой сценарии и отчеты, которые следует сохранять в общем проекте Deductor Studio Academic.

Этапы разработки составляют содержание лабораторных работ:

1. Создание хранилища данных. Загрузка данных в хранилище.
2. Очистка данных.
3. Предварительный анализ данных.
4. OLAP-анализ.
5. Прогнозирование данных.
6. Поиск ассоциативных правил.
7. Построение деревьев решений.
8. Кластерный анализ.

Файлы, подготовленные на основе аналитической платформы Deductor Studio, и базу данных вашего варианта рекомендуется хранить в личном архиве до завершения аттестации по дисциплине «Информационные технологии анализа данных».

ПРАКТИЧЕСКАЯ ЧАСТЬ

Лабораторная работа 1

Создание хранилища данных и загрузка данных

Цель работы. Приобретение практических навыков по созданию хранилища данных Deductor Studio Warehouse, собирающего информацию из разнородных источников, импорту данных и настройке параметров хранилища.

Задание

1. Подготовить исходные таблицы данных в формате, допустимом для импорта в аналитическую платформу Deductor Studio, заполнить их тестовыми значениями.

2. Выполнить импорт всех таблиц в аналитическую платформу.

3. Выполнить отображение информации в виде таблицы, диаграммы, гистограммы.

4. Создать локальное хранилище данных, включив в него все таблицы.

5. Создать отчеты по всем таблицам.

6. Продемонстрировать проект преподавателю и защитить работу.

Краткая теория и методические указания

Мастер импорта. Мастер импорта системы Deductor Studio поможет в интерактивном пошаговом режиме выбрать тип источника данных и настроить соответствующие параметры. На первом шаге *Мастера импорта* открывается список всех предусмотренных в системе типов источников данных, сгруппированных по способу доступа к данным.

Для анализа необходимо получить табличные данные из стороннего источника. Природа источника данных значения не имеет. В полной версии программного продукта поддерживаются следующие типы источников:

– хранилище данных Deductor Studio Warehouse,

– текстовый файл с разделителями,

– Microsoft Excel,

– Microsoft Access,

– dBase,

– CSV-файлы,

– 1С:Предприятие,

– промышленные СУБД (Oracle, MS SQL),

– ADO источники данных,

– ODBC источники данных.

Выбор источника исходных данных. Для вызова *Мастера импорта* можно воспользоваться кнопкой «Мастер импорта» на панели инструментов «Сценарии» или выбрать соответствующую команду из контекстного меню.

Из доступных источников щелчком мыши следует выбрать один из следующих:

1) *Deductor Studio Warehouse* – для осуществления импорта данных из хранилища данных платформы Deductor Studio;

2) *Бизнес-приложение* – для выполнения импорта данных из учетной системы 1С:Предприятие;

3) *Базы данных* – для загрузки данных из баз данных различных типов;

4) *Прямой доступ к файлам* – для доступа к данным, находящимся в текстовом файле с разделителями или в файле плоских баз данных типа DBF, который поддерживается такими приложениями, как dBase, FoxBase, FoxPro;

5) *Механизм MS ADO* – для обеспечения импорта данных:

– из книги Microsoft Excel (*.xls);

– файла СУБД Microsoft Access (*.mdb);

– плоских баз данных типа DBF, который поддерживается такими приложениями, как dBase, FoxBase, FoxPro;

– текстового файла с разделителями, доступ к которому производится через механизм ADO;

– системных настроек механизма ADO.

Число шагов *Мастера импорта*, а также набор настраиваемых параметров различен для разных типов источников. На каждом шаге *Мастера импорта* доступны кнопки «Далее» и «Назад», которые соответственно позволяют перейти к следующему шагу или вернуться на предыдущий шаг для внесения изменений в ранее настроенные параметры. Кнопка «Отмена» позволит отказаться от использования *Мастера импорта*.

Импортировать данные из выбранного файла можно двумя способами – из отдельной таблицы путем открытия файла или с помощью SQL-

запроса. Чтобы выбрать один из способов, нужно активизировать соответствующий пункт.

Если выбрать пункт «Запрос к базе данных», то в нижней части окна станет доступным поле, в которое следует ввести текст SQL-запроса.

После настройки параметров импорта запускается сам процесс импорта данных.

Настройка параметров столбцов. На данном шаге нужно настроить следующие параметры столбцов импортируемых данных, указав соответствующие значения в полях:

1. *Имя столбца* – отображается имя столбца, т. е. его идентификатор, используемый в базе данных. Изменить имя столбца здесь нельзя;

2. *Метка столбца* – указывается название (метка), под которым данный столбец будет виден в таблице, кросс-таблице или на диаграмме после импорта. Желательно, чтобы оно отражало содержание столбца;

3. *Размер* – указывается ширина столбца в символах;

4. *Тип данных* – указывается тип данных, содержащихся в столбце. Он также задается в базе данных, и изменить его здесь нельзя;

5. *Вид данных* – указывается вид данных, дискретный или непрерывный. Изменить здесь его нельзя;

6. *Назначение* – определяет порядок использования столбца при дальнейшей обработке импортированных данных:

– *неиспользуемое* – запрещает использование поля в обработке данных и исключает его из выходного набора. В отличие от непригодного поля, такие поля в принципе могут использоваться, просто в этом нет необходимости;

– *используемое* – поле будет использоваться в процедурах обработки данных;

– *непригодное* – данные в поле не пригодны для обработки;

– *входное* – поле таблицы, построенное на основе столбца, будет являться входным полем обработчика (нейронной сети, дерева решений и т. д.);

– *выходное* – поле таблицы, построенное на основе столбца, будет являться выходным полем обработчика (например, целевым полем для обучения нейронной сети);

– *информационное* – поле содержит вспомогательную информацию, которую часто полезно отображать, но не следует использовать при обработке;

– *измерение* – поле будет использоваться в качестве измерения в многомерной модели данных;

– *свойство* – поле содержит описание свойств или параметров некоторого объекта;

– *факты* – значения поля будут использованы в качестве фактов в многомерной модели данных;

– *транзакция* – поле, содержащее идентификатор событий, происходящих совместно (одновременно). Например, номер чека, по которому приобретены товары. Тогда покупка товара – это событие, а их совместное приобретение по одному чеку – транзакция;

– *элемент* – поле, содержащее элемент транзакции (событие).

Способы отображения данных. Для представления информации необходимо выбрать, в каком виде будут отображены импортированные данные. Для выборки данных, полученных в результате импорта из различных источников, доступны следующие виды отображения:

– *Таблица*. В таблице каждое поле выборки данных размещается в отдельном столбце. Столбцы озаглавлены метками полей, а если метка не была задана, то именами полей. Ширину столбцов можно менять.

– *Статистика*. В данном варианте представления будет отображаться набор основных статистических характеристик выборки данных текущей ветви сценария обработки (минимум, максимум, среднее, стандартное отклонение, сумма, сумма квадратов, количество уникальных значений, количество пустых значений).

– *Диаграмма*. При работе с диаграммой предусмотрена возможность увеличения масштаба просмотра всей диаграммы или ее произвольной области, а также доступен широкий набор различных действий и настроек, вызываемых с помощью кнопок на панели инструментов в окне диаграммы или в контекстном меню, вызываемом для поля диаграммы.

– *Гистограмма*. Действия при работе с гистограммой аналогичны действиям, выполняемым над диаграммой.

– *Куб*. Куб представляет собой один из распространенных методов комплексного многомерного анализа данных, получивших название *OLAP (On-Line Analyzing Process)*. В его основе лежит представление данных в виде многомерных кубов, называемых также OLAP-кубами или гипер-

кубами. По осям многомерной системы координат откладываются те или иные параметры анализируемого бизнес-процесса.

– *Описание.* Позволяет просмотреть все параметры, применяемые при выполнении того или иного процесса преобразования данных, в результате которого была сформирована новая выборка: импорт, обработка одним из методов или экспорт. Такими параметрами являются: время и длительность выполняемого процесса, условия остановки, наличие первичного ключа, ограничители столбцов, разделители целой и дробной частей чисел, элементов даты и т. д. В описании все параметры представлены компактно и наглядно, что позволяет оперативно анализировать текущие настройки и искать ошибки. Предусмотрено два вида представления описания: в виде дерева и текстовый. По умолчанию устанавливается вид дерева.

Отчеты. Панель *«Отчеты»* предусмотрена для того, чтобы конечный пользователь мог легко получить нужную информацию, даже не обладая специальными знаниями в области обработки данных и навыками работы в пакете Deductor Studio. Пользователю достаточно просто выбрать нужный отчет, и он будет автоматически сформирован по соответствующему сценарию.

Отчеты также представлены в виде древовидного иерархического списка, каждым узлом которого является отдельный отчет, или папка, содержащая несколько отчетов.

Чтобы добавить новый отчет нужно щелкнуть по кнопке *«Добавить узел»* или выбрать соответствующую команду из контекстного меню. В результате откроется окно *«Выбор узла»*, в котором следует выделить узел дерева сценария, где содержится нужная выборка данных, и щелкнуть по кнопке *«Выбрать»*. Следует отметить, что операция добавления нового отчета доступна только если выделена папка или корневой пункт *«Отчеты»* списка отчетов. Если выделить узел, содержащий отдельный отчет, команда создания нового отчета будет недоступна.

Чтобы добавить новую папку, нужно щелкнуть по кнопке *«Добавить папку»* или выбрать соответствующую команду в контекстном меню. В результате в списке отчетов появится новая папка с открытым полем имени, куда следует ввести имя папки. После ввода имени для его сохранения щелкнуть по любому узлу списка. Чтобы поместить отчет в папку, нужно перед вызовом команды *«Добавить узел»* выделить эту папку.

Создание файла хранилища и организация доступа к нему. Для создания и подключения хранилища данных необходимо выполнить следующие шаги:

1) в меню *«Вид»* выбрать команду *«Источники данных»*. В результате будет открыта панель *«Источники данных»*;

2) вызвать контекстное меню щелчком правой кнопки мыши в любом месте панели *«Источники данных»* и из списка *«Хранилище данных»* выбрать команду *«Создать локальное хранилище данных»*. В результате будет открыто окно настройки параметров для создания хранилища. В этом окне в поле *«Файл базы данных»* следует ввести имя файла, в котором должно быть создано новое хранилище. В полях *«Имя»* и *«Метка»* можно указать уникальный идентификатор и дать описание хранилища. Все эти действия можно проделать и в дальнейшем, в редакторе параметров источника данных.

После выполнения указанных действий выбранный файл хранилища будет отображен в качестве узла ветви *«Хранилище данных»* панели *«Источники данных»*. Для выделенного узла можно редактировать параметры подключения, вызвав окно редакторов параметров с помощью пункта всплывающего меню *«Показать»*. Для хранилища данных доступны просмотр и редактирование следующих настроек:

1) *«Имя»* – текстовое имя, под которым источник данных будет появляться в *Мастерах импорта и экспорта данных*. Это имя должно быть уникально в пределах одного типа источников.

2) *«Описание»* – пользовательское текстовое описание источника данных, содержащее любую дополнительную информацию.

3) *«Описание поставщика»* – текстовая строка с именем поставщика данных. Это поле не может быть изменено.

4) *«Хранилище данных»:*

а) *«Версия»* – версия подключаемого хранилища, это поле не может быть изменено, для внутреннего использования.

б) *«База данных»* – здесь указываются параметры подключения к базе данных поставщика. Так как любой источник представляется в виде БД, то эти поля есть у всех источников данных:

– «*База данных*» – путь к файлу базы данных или имя базы данных; если файл подключаемого хранилища находится на носителе удаленного компьютера (т. е. доступного только через сеть), то нужно установить

пункт «Удаленное», в поле «Сервер» ввести сетевое имя удаленного компьютера, а в списке «Протокол» выбрать используемый сетевой протокол. Если файл подключаемого хранилища расположен на дисках локального компьютера, то нужно установить пункт «Локальное»;

– «*Кодовая страница*» – используемая кодировка для хранения строковой информации;

– «*Логин/Пароль*» – имя пользователя и пароль для доступа к базе данных;

– «*Спрашивать логин/пароль при подключении*» – при каждом подключении к хранилищу будет выводиться диалоговое окно с запросом имени пользователя и пароля;

– «*Сохранять пароль*» – при снятом флаге при каждом подключении к базе данных у пользователя будет запрашиваться пароль. Если флаг установлен, то указанный в поле «*Логин/Пароль*» пароль будет сохранен в зашифрованном виде в файле настроек, и запрашиваться больше не будет.

Контрольные вопросы

1. Какие источники данных могут использоваться для создания хранилища данных?
2. С какой целью создается многомерное хранилище данных?
3. Какая структура данных называется процессом, что он описывает?
4. Какие данные называются измерениями, а какие свойствами?
5. Поясните порядок создания отчета.
6. Каким образом обеспечивается доступ к хранилищу данных?

Лабораторная работа 2

Очистка данных

Цель работы. Освоение основных способов очистки данных при подготовке их к анализу, приобретение практических навыков по использованию инструментария Deductor Studio по устранению ошибок в исходных данных.

Задание

1. Для исходных таблиц-справочников разработать и включить в систему сценарии определения дубликатов и противоречий.

2. Для всех исходных наборов данных разработать и включить в систему сценарии определения пропусков в данных.

3. Для полей-фактов наборов данных-процессов разработать и включить в систему сценарии определения и исправления аномальных значений.

4. Создать отчеты по всем разработанным сценариям.

5. Продемонстрировать проект преподавателю и защитить работу.

Краткая теория и методические указания

Ошибки данных. Необходимость предварительной обработки при анализе данных возникает независимо от того, какие технологии и алгоритмы используются. Более того, эта задача может представлять самостоятельную ценность в областях, не имеющих непосредственного отношения к анализу данных. При использовании же механизмов анализа, в основе которых лежат самообучающиеся алгоритмы, такие как нейронные сети, деревья решений и прочие, хорошее качество данных является ключевым требованием.

Очевидно, что исходные ('сырые') данные чаще всего нуждаются в очистке, осуществить которую можно с помощью большого набора математических методов, таких как алгоритмы робастной фильтрации, спектрального и Вейвлет-анализа, последовательной рекуррентной фильтрации, статистического анализа.

Мы не будем рассматривать ошибки такого рода, как несоответствие типов, различия в форматах ввода и кодировках, т. е. случаи, когда информация поступает из различных источников, где для обозначения одного и того же факта приняты различные соглашения. Характерный пример такой ошибки – обозначение пола человека. Где-то он обозначается как М/Ж, где-то как 1/0, где-то как True/False. С такого рода ошибками борются при помощи задания правил перекодировки и приведения типов. Такого рода проблемы более или менее сегодня решаются. Нас интересуют проблемы более высокого порядка, те, которые не решаются такими элементарными способами.

Рассмотрим основные виды ошибок, которые характерны для самых различных задач:

– противоречивость информации;

– пропуски в данных;

– аномальные значения;

– шум;

– ошибки ввода данных.

Конечно, ошибки можно править и вручную, но при больших объемах данных это становится довольно проблематично. Поэтому рассмотрим варианты решения этих задач в автоматическом режиме при минимальном участии человека.

Противоречивость информации. После того, как мы определимся с тем, что считать противоречием и найдем их, есть несколько вариантов действий.

1. При обнаружении нескольких противоречивых записей, удалять их. Метод простой, а потому легко реализуемый. Иногда этого бывает вполне достаточно. Тут важно не переусердствовать, иначе можно потерять и важные данные.

2. Исправить противоречивые данные. Можно вычислить вероятность появления каждого из противоречивых событий и выбрать наиболее вероятный. Это самый грамотный и корректный метод работы с противоречиями.

С помощью механизма обработки Deductor Studio в исходной выборке данных могут быть выявлены дублирующие и противоречивые записи. Настройка выявления дубликатов и противоречий заключается в выборе назначений полей исходной выборки данных.

Для настройки параметров поиска дубликатов и противоречий необходимо выполнить назначение полей исходной выборки. В левой части окна настройки представлен список всех полей исходной выборки. Если выделить поле, то для него будут отображены параметры: *Имя столбца, Тип данных, Вид данных.* Эти параметры определены для полей в источнике данных и здесь изменяться не могут.

Чтобы выбрать назначение для выделенного поля, нужно открыть список «Назначение» и выбрать один из возможных вариантов:

– *Входное* – поле будет содержать входные значения.

– *Выходное* – поле будет содержать выходные значения.

– *Информационное* – поиск дубликатов и противоречий для данного поля выполняться не будет, но само поле будет добавлено в результирующий набор без изменений.

– *Неиспользуемое* – поиск дубликатов и противоречий для данного поля выполняться не будет, и поле не будет представлено в результирующем наборе.

В результирующем наборе будут добавлены два поля логического типа *«Противоречие»* и *«Дубликат»*, где для каждой записи исходных полей будет указан признак дубликата или противоречия. Так, если запись содержит противоречие, то в поле *«Противоречие»* для нее будет установлено значение *«True»* (истина), в противном случае *«False»* (ложь). Аналогично и для поля *«Дубликат»*.

Кроме того, в набор будут включены два столбца целого типа *«Группа противоречий»* и *«Группа дубликатов»*, содержащие номер группы для противоречивых и дублирующихся записей соответственно. Для записей, не содержащих противоречий и дубликатов, эти два поля будут пустыми.

Обработка выявленных дубликатов может производиться с помощью других методов, доступных в Deductor Studio, таких как фильтрация и сортировка.

Пропуски в данных. Очень серьезная проблема. Большинство методов прогнозирования исходят из предположения, что данные поступают равномерным постоянным потоком. На практике такое встречается крайне редко. Поэтому одна из самых востребованных областей применения хранилищ данных – прогнозирование – оказывается реализованной некачественно или со значительными ограничениями. Для борьбы с этим явлением можно воспользоваться следующими методами:

1) *аппроксимация и экстраполяция.* Т. е. если нет данных в какой-либо точке, мы берем ее окрестность и вычисляем по известным формулам значение в этой точке, добавляя соответствующую запись в хранилище. Хорошо это работает для упорядоченных данных. Например, сведения о ежедневных продажах продуктов;

2) *определение наиболее правдоподобного значения.* Для этого берется не окрестность точки, а все данные. Этот метод применяется для неупорядоченной информации, т. е. в случае, когда невозможно определить, что же является окрестностью исследуемой точки.

Для решения этой задачи в аналитической платформе Deductor Studio в сценарий обработки необходимо включить восстановление пропущенных данных, где путем установки соответствующего пункта выбрать один из возможных методов: *«Аппроксимация»* или *«Максимальное правдоподобие»*. Пункт *«Отключить»* позволяет отказаться от восстановления пропущенных данных в сценарии обработки. Эти настройки необходимо определить для всех полей, к которым нужно применить алгоритм восстановления пропущенных данных.

Аномальные значения. Довольно часто происходят события, которые сильно выбиваются из общей картины, и лучше всего их откорректировать. Это связано с тем, что средства прогнозирования ничего не знают о природе процессов. Поэтому любое аномальное значение будет восприниматься как совершенно нормальное. Из-за этого будет сильно искажаться картина будущего. Какой-то случайный провал или успех будет считаться закономерностью.

Есть метод борьбы и с этой напастью – это *робастные* оценки. Это методы, устойчивые к сильным возмущениям. Мы оцениваем имеющиеся данные по отношению ко всему, что выходит за допустимые границы, и применяем одно из следующих действий:

1) значение удаляется;

2) значение заменяется на ближайшее граничное.

Чтобы добавить в сценарий обработки алгоритм редактирования аномальных данных, необходимо установить флажок *«Включить редактирование аномальных данных»*. В результате станет доступной настройка *«Степень подавления»*, которая позволит выбрать из списка возможную степень подавления аномальных значений – малую, среднюю или большую. Эту процедуру необходимо повторить для всех полей, к которым должно быть применено редактирование аномальных данных.

Шум. Почти всегда при анализе мы сталкиваемся с шумами. Шум не несет никакой полезной информации, а лишь мешает четко разглядеть картину. Методов борьбы с этим явлением несколько:

1) *спектральный анализ*. При помощи него можно отсечь высокочастотные составляющие данных. Проще говоря, это частые и незначительные колебания около основного сигнала. Причем, изменяя ширину спектра, можно выбирать, какого рода шум мы хотим убрать;

2) *авторегрессионные методы*. Этот довольно распространенный метод активно применяется при анализе временных рядов и сводится к нахождению функции, которая описывает процесс плюс шум. Собственно шум после этого можно удалить и оставить основной сигнал.

На шаге *«Спектральная обработка» Мастера обработки* пользователь может выбрать один из возможных методов спектральной обработки данных.

1. *Сглаживание данных* – если выбран данный пункт, то становится доступной настройка «Полоса пропускания». Чем больше требуется сгладить данные, тем меньше должно быть значение полосы. Однако слишком узкая полоса может привести к потере полезной информации.

2. *Вычитание шума* – если выбран данный метод, то становится доступной настройка «Степень вычитания шума» – малая, средняя и большая. Этим методом следует пользоваться с осторожностью, так как реализованный здесь эвристический алгоритм гарантирует удовлетворительные результаты лишь при выполнении двух условий: уровень шума мал и шум имеет нормальное распределение.

3. *Вейвлет преобразование* – если выбран данный метод, то необходимо задать глубину разложения и порядок Вейвлета. Глубина разложения определяет «масштаб» отсеиваемых деталей: чем больше эта величина, тем более «крупные» детали в исходных данных будут отброшены. При достаточно больших значениях параметра (порядка 7–9) выполняется не только очистка данных от шума, но и их сглаживание («обрезаются» резкие выбросы). Использование слишком больших значений глубины разложения может привести к потере полезной информации из-за слишком большой степени «огрубления» данных. Порядок Вейвлета определяет гладкость восстановленного ряда данных: чем меньше значение параметра, тем ярче будут выражены «выбросы», и наоборот – при больших значениях параметра «выбросы» будут сглажены.

Опция *«Отключить»* позволяет отключить все настройки и не использовать спектральную обработку в сценарии обработки.

Контрольные вопросы

1. Какие ошибки данных в источниках могут быть обнаружены средствами аналитической платформы Deductor Studio?

2. Какие методы восстановления пропущенных значений реализованы в аналитической платформе Deductor Studio?

3. Для чего используется спектральная обработка данных? Для каких наборов данных она может применяться?

4. В каких случаях целесообразно использование подавление шума?

5. Что представляет собой Вейвлет-преобразование? В каких целях оно может быть использовано?

Лабораторная работа 3

Предварительный анализ данных

Цель работы. Освоение основных методов и способов предварительного анализа при подготовке их к аналитической обработке, приобретение практических навыков по использованию инструментария Deductor Studio по корреляционному, факторному анализу, фильтрации данных.

Задание

1. Для таблицы, содержащей зависимые столбцы с числовыми данными разработать и включить в систему сценарии понижения размерного пространства факторов.

2. Для наборов данных-процессов разработать и включить в систему сценарии корреляционного анализа основных полей-факторов.

3. Для наборов данных-процессов разработать и включить в систему сценарии фильтрации данных по времени (например, за последний год, за последний месяц).

4. Создать отчеты по всем разработанным сценариям.

5. Продемонстрировать проект преподавателю и защитить работу.

Краткая теория и методические указания

Факторный анализ. Цель факторного анализа заключается в понижении размерности пространства факторов. Понижение размерности необходимо в случаях, когда входные факторы коррелированы друг с другом, т. е. взаимозависимы. В факторном анализе речь идет о выделении из множества измеряемых характеристик объекта новых факторов, более адекватно отражающих свойства объекта.

Первым этапом факторного анализа является выбор новых признаков, которые являются линейными комбинациями прежних и «вбирают» в себя большую часть общей изменчивости входных факторов. Поэтому они содержат большую часть информации, заключенной в первоначальных данных.

В нашем примере подходящей является таблица продаж, где стоимость определенного товара зависит от количества.

В обработчике *«Факторный анализ»* это осуществляется с помощью метода главных компонент. Данный метод сводится к выбору новой ортогональной системы координат в пространстве наблюдений. В качестве первой главной компоненты избирают направление, вдоль которого массив данных имеет наибольший разброс. Выбор каждой последующей главной компоненты происходит так, чтобы разброс данных вдоль нее был максимальным и чтобы эта главная компонента была ортогональна другим главным компонентам, выбранным прежде.

Обычно факторы, полученные методом главных компонент, не поддаются достаточно наглядной интерпретации. Поэтому следующим шагом факторного анализа служит преобразование (вращение) факторов таким образом, чтобы облегчить их интерпретацию.

Поле может быть использовано в факторном анализе, если выполнено несколько условий:

– оно имеет числовой тип данных;

– в нем не содержатся пропуски;

– стандартное отклонение столбца не равно нулю, то есть в столбце содержатся различные значения.

В противном случае поле будет автоматически помечено как непригодное. Для понижения размерности пространства факторов необходимо наличие хотя бы двух входных полей.

Если выделить в списке непрерывное (числовое) поле, для него будет отображен набор основных статистических характеристик в секции *«Статистика»* – минимальное, максимальное и среднее значения, а также стандартное отклонение. Если выделенное поле является дискретным, т. е. принимающим конечное число значений, для него в секции *«Уникальные значения»* будет указано количество уникальных значений в данном поле, а также список самих уникальных значений.

Корреляционный анализ. Корреляционный анализ применяется для оценки зависимости выходных полей данных от входных факторов и устранения незначащих факторов. Принцип корреляционного анализа состоит в поиске таких значений, которые в наименьшей степени коррелированы (взаимосвязаны) с выходным результатом. Такие факторы могут быть исключены из результирующего набора данных практически без потери полезной информации. Критерием принятия решения об исключении является порог значимости. Если корреляция (степень взаимозависимости) между входным и выходным факторами меньше порога значимости, то соответствующий фактор отбрасывается как незначащий.

На предыдущем шаге обработки были рассчитаны значения функции корреляции между каждым входным и каждым выходным столбцами. Эти значения отображаются в таблице в центре окна. На пересечении строки с именем входного поля и столбца с именем выходного поля находится значение рассчитанной между ними корреляции.

Исключение незначащих факторов производится на основании рассчитанной корреляции. Возможны два варианта принятия решения, определяемые выбором соответствующего пункта в нижней части окна:

– при ручном выборе незначащих факторов нужно отметить галочками те столбцы, которые будут включены в выходной набор, и снять пометки напротив тех столбцов, которые надо исключить из набора;

– в автоматическом режиме становится активной полоса *«Порог значимости»*. Передвигая по ней ползунок, можно задать необходимый уровень значимости. Столбцы, у которых максимальное из рассчитанных значений корреляции меньше порога значимости, будут исключены из выходного набора. Рекомендуемые значения порога значимости выделены синим цветом.

В выходной набор попадут информационные поля, столбцы, отмеченные на этом шаге, и все выходные столбцы.

Фильтрация данных. С помощью операции фильтрации можно оставить в таблице только те записи, которые удовлетворяют заданным условиям, а остальные удалить.

Параметры фильтрации задаются в виде списка условий, который содержит следующие столбцы:

1. *Операция* – позволяет установить функцию отношения *«И»* или *«ИЛИ»* между полями, для каждого из которых выполняется фильтрация. Возможна фильтрация по нескольким условиям для нескольких полей одновременно. В результате фильтрации по каждому из полей или условий будет получено отдельное множество значений. Тогда функция из поля *«Операция»* устанавливает отношение между этими множествами. Если используется отношение *«И»*, то в результирующий набор будут включены записи, удовлетворяющие условиям фильтрации по обоим полям. При использовании отношения *«ИЛИ»* в выходной набор включаются записи, удовлетворяющие хотя бы одному из условий. Установка отношений возможна, только если настроены два или более условий фильтрации. Для этого следует выполнить двойной щелчок в столбце *«Операция»* для соответствующего условия и из списка выбрать нужную функцию отношения. По умолчанию устанавливается отношение *«И»*.

2. *Имя поля* – позволяет выбрать поле, по значениям которого должна быть выполнена фильтрация. Для этого надо дважды щелкнуть в столбце *«Имя поля»* и с помощью кнопки открыть список полей текущей выборки, из которого выбрать нужное поле. Одно и то же поле может быть использовано в нескольких условиях.

3. *Условие* – указывается условие, по которому нужно выполнять фильтрацию для данного поля. Для выбора условия достаточно дважды щелкнуть мышью в соответствующей ячейке и в списке условий, открываемом кнопкой, выделить нужное условие. Доступны следующие условия фильтрации:

– *«=» (равно), «<» (меньше), «<=» (меньше или равно), «>» (больше), «>=» (больше или равно), «<>» (не равно)* – отбираются только те записи, значения которых в данном поле соответственно равны содержимому столбца *«Значение»*, меньше, меньше или равны, больше, больше или равны, не равны ему;

– *«пустой»* – отбираются только те записи, для которых в данном поле содержится пустое значение. В этом случае поле «Значение» не используется;

– *«не пустой»* – отбираются только те записи, для которых в данном поле не содержится пустое значение. В этом случае поле «Значение» не используется;

– *«в интервале», «вне интервала»* – отбираются только те записи, значения которых в данном столбце лежат в выбранном диапазоне (вне выбранного диапазона), то есть между (не между) верхней и нижней границами;

– *«в списке», «вне списка»* – отбираются только те записи, которые в данном столбце лежат в выбранном списке (вне выбранного списка);

– *«содержит», «не содержит»* – отбираются только строки, содержащие (не содержащие) указанную подстроку;

– *«начинается на», «не начинается на»* – для строковых полей отбираются записи, значения которых в данном столбце начинаются (не начинаются) на введенную последовательность символов;

– *«заканчивается на», «не заканчивается на»* – для строковых полей отбираются записи, значения которых в данном столбце заканчиваются (не заканчиваются) на введенную последовательность символов;

– *«первый», «не первый»* – для полей типа «Дата/время» – по данному полю отбираются первые (не первые) N периодов от выбранной даты. Периодом может быть день, неделя, месяц, квартал, год. Например, если выбрать условие «первые 3 дня от 29.11.2004», то будут отобраны записи, в которых значение данного поля равно «29.11.2004», «30.11.2004», «01.12.2004»;

– *«последний», «не последний»* – для полей типа «Дата/время» – по данному полю отбираются последние (не последние) N периодов от выбранной даты. Периодом может быть день, неделя, месяц, квартал, год. Например, если выбрать условие «последние 3 дня от 29.11.2004», то будут отобраны записи, в которых значение данного поля равно «29.11.2004», «28.11.2004», «27.11.2004».

4. *Значение* – значение, по которому будет производиться фильтрация записей в соответствии с заданным условием. Способ ввода значения будет различным в зависимости от типа данных и условия. Допустим, в качестве условия выбрана операция отношения *«=», «<>», «>»* и т. д. Если данные в поле являются непрерывными (т. е. числовыми), то достаточно дважды щелкнуть мышью в соответствующей ячейке, чтобы появился

курсор, затем ввести значение (число). Если поле, по которому выполняется фильтрация, имеет тип *«строка»* (т. е. является дискретным), то в результате двойного щелчка в столбце *«Значение»* появится кнопка выбора, которая откроет окно *«Список уникальных значений»*, где будут отображены все уникальные значения поля и их количество. Чтобы выбрать значение для условия отбора, достаточно выделить его и щелкнуть *«Ok»* либо просто выполнить двойной щелчок. Если выбрано условие *«между»* или *«не между»*, тогда при нажатии кнопки выбора (справа от поля) откроется окно, в котором необходимо выбрать верхнюю и нижнюю границы интервала. Если выбрано условие *«в списке»* или *«вне списка»*, тогда по кнопке выбора откроется окно, в котором необходимо выбрать список значений, установив галочки рядом с необходимыми значениями из списка. Если выбрано условие *«первый», «не первый», «последний», «не последний»*, тогда по кнопке выбора откроется окно, где необходимо указать дату, от которой вести отсчет, тип периода и количество периодов. Дата может быть текущей от имеющихся данных либо указанной вручную. Дата от имеющихся данных означает либо минимальную дату во всем наборе исходных значений обработчика (если выбрано условие *«первый», «не первый»*) либо максимальную (если выбрано условие *«последний», «не последний»*).

Изначально в окне настройки фильтрации появляется новая строка с пустым условием. Чтобы ввести новое условие фильтрации, нужно щелкнуть по кнопке на панели инструментов, расположенной справа от списка условий. При этом в окне появится новая пустая строка, в которой необходимо последовательно задать операцию отношения (кроме первой строки), имя поля, само условие и значение для отбора. Если хотя бы один из параметров задан не будет, при попытке перейти на следующий шаг *Мастера обработки* будет выдано сообщение об ошибке с указанием строки, в которой она была допущена. Для работы с уже введенными условиями можно использовать следующие кнопки (условие, в строке которого находится курсор или маркер является текущим):

– перемещает текущее условие на одну позицию вверх по списку;

– перемещает текущее условие на одну позицию вниз по списку;

– удаляет текущее условие;

– очищает список условий.

По мере заполнения списка условий в правой нижней части окна отображается общее выражение, описывающее параметры фильтрации. Установка флажка «Учитывать регистр» позволяет учитывать регистр при отборе записей по значению.

Контрольные вопросы

1. Поясните основные цели и назначение факторного и корреляционного анализа в предварительном анализе данных.

2. Для каких целей используется порог значимости? Каковы основные критерии его выбора?

3. Каким образом фильтрацию данных можно использовать для исключения записей, содержащих пропуски?

4. Каким должно быть количество входных и выходных полей при проведении факторного и корреляционного анализа?

5. Каким образом осуществляется настройка составных условий при фильтрации данных?

Лабораторная работа 4

OLAP-анализ

Цель работы. Освоение методов технологии OLAP и способов представления данных с использованием многомерных кубов. Изучение инструментария Deductor Studio по многомерному анализу данных.

Задание

1. Создать первую сводную таблицу (например товаров, включив в нее суммарные сведения о продажах, остатках и поставках). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.

2. Создать вторую сводную таблицу (например организаций, включив в нее суммарные сведения о продажах и поставках).

3. Создать третью сводную таблицу (например менеджеров, включив в нее сведения о продажах).

4. Для подготовленных сводных таблиц разработать сценарии OLAP-анализа на основе многомерного представления информации в виде куба и отображением ее с использованием кросс-таблиц и кросс-диаграмм.

5. Создать отчеты по всем разработанным сценариям.

6. Продемонстрировать проект преподавателю с использованием тестовых наборов данных и защитить работу.

Краткая теория и методические указания

OLAP-куб. Куб представляет собой один из распространенных методов комплексного многомерного анализа данных, получивших название *OLAP (On-Line Analyzing Process)*. В его основе лежит представление данных в виде многомерных кубов, называемых также OLAP-кубами или гиперкубами. По осям многомерной системы координат откладываются те или иные параметры анализируемого бизнес-процесса. Например, для продаж это может быть товар, регион, тип покупателя.

Обычно в качестве одного из измерений используется время. По осям (измерениям) многомерной системы координат находятся данные, количественно характеризующие процесс-факты. Это могут быть объемы продаж в штуках или в денежном выражении, остатки на складе, издержки, суммы и т. д. Пользователь, анализирующий информацию, может выполнять сечение куба по различным направлениям, получать сводные (например, по годам) или, наоборот, детальные (по неделям) данные и осуществлять другие операции, необходимые для эффективного анализа.

Чтобы получить на основе текущей выборки данных кросс-таблицу и соответствующую кросс-диаграмму, необходимо выполнить следующие шаги:

– настройка назначений полей (рассмотрена в лабораторной работе № 1).

– настройка размещения измерений.

Размещение измерений. Здесь следует определить расположение измерений, выбранных на предыдущем шаге, – по строкам или столбцам. Для работы с измерениями в окне имеются три поля:

1) *доступные измерения* – содержит заголовки измерений, размещение которых в кросс-таблице еще не определено (т. е. они в кросс-таблице отображаться не будут);

2) *измерения в строках* – для измерений, помещенных в это поле, факты в кросс-таблице будут располагаться горизонтально;

3) *измерения в столбцах* – для измерений, помещенных в это поле, факты в кросс-таблице будут располагаться вертикально.

Кроме полей, для настройки размещения измерений в окне имеется поле *«Факт»*, которое позволяет выбирать факты, отображаемые в кросс-таблице. Чтобы факт был отображен, слева от него должен быть установлен флажок. Сброс флажка позволяет скрыть факт. Для каждого факта можно установить функцию агрегации. Для этого дважды щелкнуть мышью в столбце *«Агрегация»* для соответствующего факта и из списка выбрать нужный пункт. Для вещественного и целого типов данных факта могут быть доступны следующие виды агрегации:

– сумма, среднее;

– минимум, максимум;

– количество.

Для остальных типов данных доступна только одна функция агрегации – *«Количество»*.

Кросс-таблица. Кросс-диаграмма. Кросс-таблица – удобное средство визуализации многомерных данных и получения необходимых форм отчетов. Кросс-таблица строится на основе многомерного представления в виде OLAP-куба и содержит измерения и факты, определенные при построении куба. Основной особенностью кросс-таблицы является то, что ее структура не является жестко определенной. Манипулируя с помощью мыши заголовками измерений, пользователь может добиться, чтобы кросс-таблица выглядела наиболее информативно.

Кросс-диаграмма представляет собой диаграмму заданного типа, построенную на основе кросс-таблицы. Основное отличие кросс-диаграммы от обычной диаграммы в том, что она однозначно соответствует текущему состоянию кросс-таблицы и при любых ее трансформациях изменяется соответственно.

При работе с кросс-диаграммой предусмотрена возможность увеличения масштаба просмотра всей кросс-диаграммы или ее произвольной области. Для этого следует, удерживая левую кнопку мыши нажатой, выделить ту область кросс-диаграммы, которую нужно просмотреть более детально, при этом двигая мышь слева направо. Как только кнопка мыши

будет отпущена, масштаб просмотра выделенной области будет увеличен. Для дальнейшего увеличения масштаба данную процедуру можно повторить. При выделении области диаграммы движением мыши слева направо масштаб просмотра диаграммы будет возвращен к исходному, независимо от размера выделенной области. Направляя указатель мыши в произвольную точку диаграммы и передвигая ее с нажатой правой кнопкой, можно перемещать диаграмму по экрану, делая доступными для просмотра различные ее части.

Контрольные вопросы

1. Поясните основные принципы технологии OLAP.
2. Какие операции используются в OLAP-анализе?
3. Какие манипуляции с кросс-таблицами и кросс-диаграммами используются для улучшения представления сводных данных?
4. Каким образом осуществляется слияние данных из нескольких наборов данных в платформе Deductor Studio?
5. В чем заключается группировка данных, и в каких целях она применяется?

Лабораторная работа 5

Прогнозирование с использованием метода скользящего окна

Цель работы. Освоение основных методов и способов прогнозирования временного ряда, приобретение практических навыков по использованию инструментария Deductor Studio и метода скользящего окна в задачах прогнозирования.

Задание

1. Создать сводную временную таблицу (например, продаж отдельных категорий товаров по количеству и по сумме, включив в нее суммарные сведения о продажах). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки. При необходимости выполнить сглаживание исходных данных.
2. Разработать сценарии прогнозирования (например, продаж) на заданный временной период вперед.

3. Создать сводную временную таблицу (например, сумм оплаты товаров). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.

4. Разработать сценарии прогнозирования (например, сумм оплаты) на заданный временной период вперед.

5. Создать отчеты по всем разработанным сценариям.

6. Продемонстрировать проект преподавателю с использованием тестовых наборов данных и защитить работу.

Краткая теория и методические указания

Метод скользящего окна. Прогнозирование позволяет получать предсказание значений временного ряда на число отсчетов, соответствующее заданному горизонту прогнозирования. Алгоритм прогнозирования работает следующим образом. Пусть в результате преобразования методом скользящего окна была получена последовательность временных отсчетов: *X(–n), ..., X(–2), X(–1), X0, X(+1)*, где *X(+1)* прогнозируемое значение, полученное с помощью предыдущего этапа обработки (например, линейной регрессии) на основе *n* предыдущих значений. Тогда, чтобы построить прогноз для значения *X(+2)*, нужно сдвинуть всю последовательность на один отсчет влево, чтобы ранее сделанный прогноз *X(+1)* тоже вошел в число исходных значений. Затем снова будет запущен алгоритм расчета прогнозируемого значения – *X(+2)* будет рассчитан с учетом *X(+1)* и так далее, в соответствии с заданным горизонтом прогноза.

Обработка данных методом скользящего окна применяется при предварительной обработке данных в задачах прогнозирования, когда на вход анализатора (например, нейронной сети) требуется подавать значения нескольких смежных отсчетов исходного набора данных. Термин *«скользящее окно»* отражает сущность обработки – выделяется некоторый непрерывный отрезок данных, называемый окном, а окно, в свою очередь, перемещается – «скользит» – по всему исходному набору данных.

В результате будет получена выборка, где в каждой записи будет содержаться поле, соответствующее текущему отсчету (оно будет иметь то же имя, что и в исходной выборке), а слева и справа от него будут расположены поля, содержащие отсчеты, смещенные от текущего отсчета в прошлое и в будущее соответственно.

Следовательно, обработка методом скользящего окна имеет два параметра: *глубина погружения* – количество отсчетов в «прошлое» и *горизонт прогнозирования* – количество отсчетов в «будущее».

Необходимо отметить, что для граничных положений окна (конец и начало исходной выборки) будут формироваться неполные записи – в начале исходной выборки формируются пустые значения для «прошлых» отсчетов, а в конце – для «будущих». В зависимости от конкретной ситуации пользователь может включать такие неполные записи в результирующую выборку или исключать их.

Настройка параметров прогнозирования. На данном шаге необходимо настроить связи столбцов для прогнозирования временного ряда. В графе *«Столбец»* представлены все поля исходной выборки. При этом для них установлено то же назначение, что и при настройке параметров предыдущего этапа обработки, где строилась модель прогноза. В графе *«При очередном шаге брать значения из»* выбираются поля, значения которых следует использовать для расчета прогнозируемого значения на очередном шаге прогнозирования:

1. *Горизонт прогноза* – в данном поле следует указать число шагов прогноза. Практически это определяет количество «будущих» отсчетов, которое будет рассчитано в результате прогноза.

2. *Добавлять горизонт прогноза* – установка данного флажка позволит добавить в результирующую выборку дополнительное поле *«Шаг прогноза»*, в котором для каждой записи будет указан номер шага прогноза, в результате которого она была получена.

3. *Исходные данные* – установка данного флажка позволяет включить в результирующую выборку не только записи, содержащие прогнозируемые значения, но и все записи, которые содержат исходные данные. В этом случае записи, содержащие прогноз будут расположены в конце результирующей выборки.

Диаграмма прогноза. Диаграмма прогноза становится доступной в списке способов представления только для тех ветвей сценария, которые содержат прогноз временного ряда. Основное отличие диаграммы прогноза от обычной диаграммы в том, что на ней, кроме исходных данных, отображаются результаты прогноза, при этом исходные данные и прогноз отличаются по цвету.

Все операции, доступные при работе с обычной диаграммой, применимы и к диаграмме прогноза.

При настройке параметров диаграммы прогноза можно выбрать поля результирующего набора данных, полученного в результате выполнения операции прогнозирования. Эти данные должны отображаться на диаграмме, назначить для них определенный цвет, определить тип диаграммы, а также настроить отображение подписей и значений по оси *X*. Для того чтобы поле отображалось на диаграмме прогноза, нужно установить флажок рядом с его именем в списке *«Имя поля»*. Сброс флажка наоборот позволит скрыть поле на диаграмме. Чтобы определить цвет, которым данное поле будет отображаться на диаграмме прогноза, нужно выполнить двойной щелчок мышью в графе *«Цвет»* напротив имени поля. В результате будет открыто стандартное окно Windows – *«Цвет»*, в котором пользователь может выбрать нужный цвет.

Контрольные вопросы

1. Каким образом определяется достаточность данных для построения модели прогноза?
2. Какие математические модели могут быть использованы при прогнозировании?
3. Каким образом следует выбирать глубину погружения?
4. Следует ли применять сглаживание исходных данных при проведении прогнозирования? Почему?
5. Как связаны между собой глубина погружения и горизонт прогнозирования?

Лабораторная работа 6

Поиск ассоциативных правил

Цель работы. Освоение основных методов и способов проведения ассоциативного анализа и построения деревьев решений, приобретение практических навыков по использованию инструментария Deductor Studio по использованию метода скользящего окна в задачах поиска ассоциативных правил.

Задание

1. Разработать сценарии поиска ассоциативных правил и проведения анализа «что – если».

2. По таблице (например, продаж) создать таблицу транзакций (например, с полями: Менеджер, Организация, Вид товара). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.

3. Разработать сценарии поиска ассоциативных правил импликаций (например, Менеджер => Вид товара и Организация => Вид товара).

4. Создать отчеты по всем разработанным сценариям.

5. Продемонстрировать проект преподавателю с использованием тестовых наборов данных и защитить работу.

Краткая теория и методические указания

Ассоциативные правила. Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила служит утверждение, что покупатель, приобретающий «Хлеб», приобретет и «Молоко» с вероятностью 75 %. Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины *(market basket analysis)*.

Транзакция – это множество событий, произошедших одновременно. Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют потребительской корзиной. Пусть имеется список транзакций. Необходимо найти закономерности между этими событиями. Как в условии, так и в следствии правила должны находиться элементы транзакций.

Пусть $I=\{i1,i2,...,in\}$ – множество элементов, входящих в транзакции.

Ассоциативным правилом называется импликация $X => Y$ (читается «X дает Y» или «из X следует Y»), где X принадлежит I, Y принадлежит I и пересечение X и Y – пустое множество.

Обычные ассоциативные правила – это правила, в которых как в условии, так и в следствии присутствуют только элементы транзакций и при вычислении которых используется только информация о том, присутству-

ет элемент в транзакции или нет. Фактически все приведенные выше примеры относятся к обычным ассоциативным правилам.

Настройка назначения полей. Для определения ассоциативных правил нужно определить, как каждое поле будет использоваться при обработке данных. Слева на форме представлен список всех полей выборки, каждое из которых помечено значком в зависимости от их назначения:

Транзакция – поле, содержащее идентификатор событий, происходящих совместно (одновременно). Например, номер чека, по которому приобретены товары. Тогда покупка одного товара – это событие, а их совместное приобретение по одному чеку – транзакция.

Элемент – поле, содержащее элемент транзакции (событие).

Неиспользуемое – поле не будет использоваться при обучении и работе нейронной сети. В отличие от непригодного, такое поле может быть использовано, если в этом возникнет необходимость.

Непригодное – поле не может быть использовано при построении и работе алгоритма.

Поле может быть использовано в ассоциативных правилах, если выполнено несколько условий:

1) в нем не содержатся пропуски;

2) количество уникальных значений меньше количества строк таблицы;

3) дисперсия столбца не равна нулю, то есть в столбце содержатся одинаковые значения.

В противном случае поле будет автоматически помечено как непригодное.

Для начала работы с ассоциативными правилами необходимо указать, что является идентификатором *(ID)* транзакции, а что элементом транзакции. Например, идентификатор транзакции – это номер чека или код накладной, а элемент – это наименование товара в чеке.

Для указания назначения полю необходимо выделить его и в списке *«Назначение»* выбрать один из вариантов его использования.

Параметры построения ассоциативных правил. Далее следует указать параметры поиска правил:

– *Минимальная и максимальная поддержка*. Ассоциативные правила находятся только в некотором множестве всех транзакций. Для того чтобы транзакция вошла в это множество, она должна встретиться в исходной

выборке количество раз, большее минимальной поддержки и меньшее максимальной. Например, минимальная поддержка равна 1 %, а максимальная – 20 %. Количество элементов «Хлеб» и «Молоко» столбца «Товар» с одинаковым значением столбца «Номер чека» встречаются в 5 % всех транзакций (номеров чека). Тогда эти две строки войдут в искомое множество.

– *Минимальная и максимальная достоверность*. Это процентное отношение количества транзакций, содержащих все элементы, которые входят в правило, к количеству транзакций, содержащих элементы, которые входят в условие. Если транзакция – это заказ, а элемент – товар, то достоверность характеризует, насколько часто покупаются товары, входящие в следствие, если заказ содержит товары, вошедшие во всё правило.

– *Максимальная мощность искомых часто встречающихся множеств*. Если этот параметр указан (флажок установлен), то максимальная мощность (количество элементов) часто встречающихся множеств будет не больше значения этого параметра. Следовательно, любое результирующее правило будет состоять не более чем из *«максимальной мощности»* элементов.

Выявление действительно интересных правил – это одна из главных подзадач при вычислении ассоциативных зависимостей. Для того чтобы получить действительно интересные зависимости, нужно разобраться с несколькими эмпирическими правилами:

1. Уменьшение минимальной поддержки приводит к тому, что увеличивается количество потенциально интересных правил, однако это требует существенных вычислительных ресурсов. Одним из ограничений уменьшения порога минимальной поддержки является то, что слишком маленькая поддержка правила делает его статистически необоснованным.

2. Уменьшение порога достоверности также приводит к увеличению количества правил. Значение минимальной достоверности также не должно быть слишком маленьким, так как ценность правила с достоверностью в 5 % настолько мала, что это правило таковым считать нельзя.

3. Правило со слишком большой поддержкой с точки зрения статистики представляет собой большую ценность, но с практической точки зрения это, скорее всего, означает то, что, либо правило всем известно, либо товары, присутствующие в нем, являются лидерами продаж, откуда следует их низкая практическая ценность.

4. Правило со слишком большой достоверностью практической ценности в контексте решаемой задачи не имеет, так как товары, входящие в следствие, покупатель, скорее всего, уже купил.

5. Если значение верхнего предела поддержки имеет слишком большое значение, то в правилах основную часть будут составлять товары – лидеры продаж. При таком раскладе не представляется возможным уменьшить минимальный порог поддержки до того значения, при котором могут появляться интересные правила. Причиной тому является просто огромное число правил и, как следствие, нехватка системных ресурсов. Причем получаемые правила процентов на 95 содержат товары – лидеры продаж.

6. Если задать значение параметра максимальная мощность, то можно искать правила, которые состоят не более чем из *«максимальной мощности»* количества элементов. Например, если нужны только простые правила для оценочного анализа, то значение максимальной мощности следует установить либо в 2 либо в 3. При этом если максимальная мощность равна 2, то все найденные правила будут иметь вид: *«Если Товар I, то Товар J»*. Ограничение поиска часто встречающихся множеств по мощности (количеству элементов в множестве) может также понадобится, если при указанном значении минимальной поддержки количество часто встречающихся множеств, имеющих большую мощность, слишком велико.

Варьируя верхним и нижним пределами поддержки и достоверности, а также параметром *«максимальная мощность»*, можно избавиться от очевидных и неинтересных закономерностей. Как следствие, правила, генерируемые алгоритмом, принимают приближенный к реальности вид.

Дерево правил. *Дерево правил* – это всегда двухуровневое дерево. Оно может быть построено либо по условию либо по следствию.

При построении дерева правил по условию на первом (верхнем) уровне находятся узлы с условиями, а на втором уровне – узлы со следствием.

Второй вариант дерева правил – дерево, построенное по следствию. Здесь на первом уровне располагаются узлы со следствием. Дерево правил располагается в левой части окна. Для управления деревом правил служат следующие кнопки на панели инструментов:

1. *Группировать по условию* – построить дерево так, чтобы на первом уровне располагались элементы условия.

2. *Группировать по следствию* – построить дерево так, чтобы на первом уровне располагались элементы следствия.

3. *Сортировка*:

– по порядку;

– по следствию (по убыванию);

– по следствию (по возрастанию) – сортировать по наименованию элемента;

– по поддержке (по возрастанию);

– по поддержке (по убыванию);

– по достоверности (по убыванию);

– по достоверности (по возрастанию).

4. *Фильтрация…* – открывает окно настройки фильтра ассоциативных правил. Правила могут быть отфильтрованы по условию, по следствию, по поддержке и по достоверности. В случае большого списка можно оставить, например, наиболее достоверные правила.

5. *Показать проценты* – имеет два состояния: в нажатом состоянии справа от каждого элемента дерева отображается его поддержка в процентах и количестве.

6. *Показать правила (F12)* – имеет два состояния: в нажатом состоянии справа от дерева отображаются правила, соответствующие выбранному узлу дерева.

7. *Найти правило на дереве (Enter)* – при выборе мышкой правила из списка справа от дерева можно найти это правило в дереве, нажав эту кнопку.

Справа от дерева находится список правил, построенный по выбранному узлу дерева. Для каждого правила отображаются поддержка и достоверность. Если дерево построено по условию, то вверху списка отображается условие правила, а список состоит из его следствий. Тогда правила отвечают на вопрос, что будет при таком условии. Если же дерево построено по следствию, то вверху списка отображается следствие правила, а список состоит из его условий. Эти правила отвечают на вопрос, что нужно, чтобы было заданное следствие, или какие товары нужно продать, для того чтобы продать товар из следствия. Это может быть очень актуально, если какой-либо товар просто «завалялся» на складе.

Визуализация списка правил. Этот визуализатор отображает ассоциативные правила в виде списка правил. Этот список представлен таблицей со столбцами: номер по порядку, условие, следствие, поддержка в процентах, поддержка в количестве, достоверность. Список можно отсортировать, отфильтровать, экспортировать отчет в *MS Excel, MS Word, HTML* формат.

В поле *«Итого правил»* отображается количество правил, входящих в список с учетом фильтрации.

Популярные наборы. *Популярные наборы или часто встречающиеся множества* – это множества, состоящие из одного и более элементов, которые наиболее часто встречаются в транзакциях одновременно. На сколько часто встречается множество в исходном наборе транзакций можно судить по поддержке.

Данный визуализатор отображает множества в виде списка.

Анализ «что – если». Анализ по методу *«что – если»* позволяет исследовать то, как будет вести себя построенная система обработки при подаче на ее вход тех или иных данных. Проще говоря, проводится эксперимент, в котором, изменяя значения входных полей обучающей или рабочей выборки нейронной сети или дерева решений, пользователь наблюдает за изменением значений на выходе.

Возможность анализа по принципу *«что – если»* особенно ценна, поскольку позволяет исследовать правильность работы системы, достоверность полученных результатов, а также ее устойчивость. Под устойчивостью понимается то, насколько снижается достоверность полученных результатов при попадании на вход системы нетипичных данных – выбросов, пропусков данных и т. д. Такой анализ позволит определить, какую предварительную обработку данных нужно провести перед подачей на вход системы.

Система анализа *«что – если»* включает табличное и графическое представления, которые формируются одновременно.

Анализ *«что – если»* в ассоциативных правилах позволяет ответить на вопрос: *что* получим в качестве следствия, *если* выберем данные условия? Например, какие товары приобретаются совместно с выбранными товарами. В верхней части таблицы отображаются входные поля, а в нижней – выходные и расчетные. Изменяя значения входных полей, пользова-

тель дает команду на выполнение расчета и наблюдает рассчитанные значения выходов нейронной сети или дерева решений.

Расчетные поля отличаются от выходных тем, что они не существуют в исходном наборе данных и были созданы в ходе обработки. Такими полями являются, например, *«Номер правила»* или *«Поддержка»* для дерева решений.

Для каждого поля таблица *«что – если»* содержит столбцы:

– *поле* – имя входного или выходного поля;

– *тип* – указывается значок, соответствующий типу данных в поле;

– *значение* – указывается текущее значение поля.

Справа от таблицы можно вывести статистику по выделенному полю.

Знание диапазона входных данных (минимума и максимума), на котором обучалась сеть или дерево, позволит определить область устойчивости системы. Очевидно, что если подать на вход значения, существенно выходящие за диапазон, гарантировать правильную реакцию системы нельзя и достоверность полученных данных может быть снижена. Если значение, присвоенное полю, выходит за границы диапазона, это поле окрашивается в красный цвет.

В таблице пользователь может менять только содержимое столбца *«Значение»*. Это можно делать двумя способами – непосредственно вводя данные с клавиатуры или заполняя записями из текущей выборки (при этом вводятся записи целиком и заполняются одновременно все поля).

Чтобы ввести значения входов с клавиатуры, нужно щелкнуть мышью в столбце *«Значение»* в ячейке для соответствующего поля, затем ввести данные. Дискретные значения выбираются из выпадающего списка. Непрерывное значение может быть введено непосредственно с клавиатуры или с помощью ползунка, вызываемого кнопкой. Целые значения, кроме того, могут вводиться последовательным перебором вверх или вниз с помощью кнопок. Для перехода к предыдущим или последующим строкам можно использовать клавиши со стрелками. Если введенные вами значения выходят за диапазон значений выборки, соответствующая строка таблицы выделяется красным цветом.

При автоматической загрузке записей или в режиме *«Автоматически рассчитать выходы»* пересчет значений выходных полей производится автоматически.

По горизонтальной оси диаграммы отображается весь диапазон значений текущего поля выборки, а по вертикальной – значения соответствующих выходов сети. По диаграмме *«что – если»* пользователь может легко увидеть, при каком значении входа изменяется значение на соответствующем выходе. Если, например, во всем диапазоне входных значений выходное значение для данного поля не изменялось, то диаграмма будет представлять собой горизонтальную прямую линию.

Важной особенностью диаграммы *«что – если»* является возможность комбинирования входных и выходных полей, отображаемых на диаграмме, т. е. для любого входного поля, выбранного для отображения по горизонтали, можно выбрать любое выходное поле, которое будет отображаться по вертикали.

В окне слева расположен список всех элементов транзакций. Справа от каждого элемента выделена поддержка – сколько раз данный элемент встречается в транзакциях. Список содержит также те элементы, которые не вошли в правила и в часто встречающиеся множества. Для них также указана поддержка, но она выделена красным цветом и с «волной» перед числом, это означает, что она вычислена примерно и может отличаться от реального значения.

Для управления списком предназначена панель инструментов.

Правила могут быть отфильтрованы по условию, по следствию, по поддержке и по достоверности. В случае большого списка можно оставить, например, наиболее достоверные правила.

Контрольные вопросы

1. Какие требования к полям исходных данных предъявляются при проведении ассоциативного анализа?

2. Поясните понятие поддержки ассоциативного правила.

3. Что характеризует достоверность ассоциации?

4. Поясните понятие мощности искомых часто встречающихся множеств.

5. С какой целью производится анализ «что – если»: что он позволяет исследовать?

Лабораторная работа 7

Построение дерева решений

Цель работы. Освоение основных методов и способов построения деревьев решений, приобретение практических навыков по использованию инструментария Deductor Studio.

Задание

1. Разработать сценарии построения дерева решений и проведения анализа *«что – если»*.

2. По таблице (например, продаж) создать таблицу транзакций с полями (например, Менеджер, Организация, Вид товара). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.

3. Разработать сценарии построения дерева решений с представлением правил, наиболее популярных наборов и анализа «что – если» с входными полями (например, Менеджер и Организация) и выходным полем (например, Вид товара).

4. Создать отчеты по всем разработанным сценариям.

5. Продемонстрировать проект преподавателю с использованием тестовых наборов данных и защитить работу.

Краткая теория и методические указания

Деревья решений. Деревья решений *(decision trees)* являются одним из самых мощных средств решения задачи отнесения какого-либо объекта (строчки набора данных) к одному из заранее известных классов. *Дерево решений* – это классификатор, полученный из обучающего множества, содержащего объекты и их характеристики, на основе обучения. Дерево состоит из узлов и листьев, указывающих на класс.

Результатом работы алгоритма является список иерархических правил, образующих дерево. Каждое правило – это интуитивно понятная конструкция вида *«Если…то…» (if – then)*. Дерево может использоваться для классификации объектов, не вошедших в обучающее множество. Чтобы принять решение, к какому классу следует отнести некоторый объект или ситуацию, требуется ответить на вопросы, стоящие в узлах этого дерева, начиная с его корня. Вопросы имеют вид *«значение параметра А больше*

B?». Если ответ положительный, осуществляется переход к правому узлу следующего уровня; затем снова следует вопрос, связанный с соответствующим узлом и т. д.

Настройка назначения полей. Необходимо определить, как будут использоваться поля исходного набора данных при обучении дерева и дальнейшей практической работе с ним. В левой части окна представлен список всех полей исходного набора данных. Для настройки поля следует выделить его в списке, при этом в правой части окна будут отображены текущие параметры поля:

1. *Имя поля* – идентификатор поля, определенный для него в источнике данных. Изменить его здесь нельзя.

2. *Тип данных* – тип данных, содержащихся в поле (вещественный, строковый, дата). Он также задается в источнике данных и здесь не может быть изменен.

3. *Назначение* – здесь необходимо выбрать порядок использования данного поля при обучении и работе дерева решений. Выбор производится с помощью списка, открываемого кнопкой и содержащего следующие варианты:

– *Входное* – значения поля будут являться исходными данными для построения и дальнейшей практической работы дерева решений, на их основе будет производиться классификация.

– *Выходное* – будет содержать результаты классификации. Выходное поле может быть только одно, и оно должно быть дискретным.

– *Информационное* – поле не будет использоваться при обучении дерева, но будет помещено в результирующий набор в исходном состоянии.

– *Неиспользуемое* – поле не будет использоваться при построении и работе дерева решений и будет исключено из результирующей выборки. В отличие от непригодного такое поле может быть использовано, если в этом возникнет необходимость.

– *Непригодное* – поле не может быть использовано при построении и работе алгоритма, но будет помещено в результирующий набор в исходном состоянии.

4. Вид данных – указывает на характер данных, содержащихся в поле (непрерывный или дискретный). Изменить это свойство здесь нельзя.

Статус непригодного поля устанавливается только автоматически и в дальнейшем может быть изменен только на неиспользуемое или информационное. Поле будет запрещено к использованию, если:

– поле является дискретным и содержит всего одно уникальное значение;

– непрерывное поле с нулевой дисперсией;

– поле содержит пропущенные значения.

В случае если текущее поле содержит непрерывные (числовые) данные, отображается секция *«Статистика»*, где показываются максимальное и минимальное значения поля, его среднее значение и стандартное отклонение. Если выделенное поле содержит дискретные (строковые) данные, то для него открывается секция *«Уникальные значения»*, в которой отображается общее число уникальных значений поля, а также список самих уникальных значений.

Нормализация полей. Целью нормализации значений полей является преобразование данных к виду, наиболее подходящему для обработки средствами пакета Deductor Studio. Для дерева решений данные, поступающие на вход, должны иметь числовой тип. В этом случае нормализатор может преобразовать дискретные данные к набору уникальных индексов.

Окно настройки нормализации полей вызывается с помощью кнопки *«Настройка нормализации»*. В окне слева приведен полный список входных и выходных полей. При этом каждое поле помечено значком, обозначающим вид нормализации поля:

– *линейная* – линейная нормализация исходных значений;

– *уникальные значения* – преобразование уникальных значений в их индексы.

Для числовых (непрерывных) полей с *линейной нормализацией* дополнительные параметры недоступны. В полях *«Минимум»* и *«Максимум»* секции *«Диапазон значений»* можно посмотреть минимальное и максимальное значения этого поля.

Для *дискретных полей* справа находится список уникальных значений поля, где для каждого значения указывается количество его повторений. Поле *«Количество значений»* показывает общее число уникальных значений, принимаемых полем.

Настройка обучающей выборки. Обучающая выборка может быть разбита на три множества – обучающее, тестовое и валидационное.

1. *Обучающее множество* включает записи (примеры), которые будут использоваться в качестве входных данных, а также соответствующие желаемые выходные значения.

2. *Тестовое множество* также включает записи, содержащие входные и желаемые выходные значения, но используется не для обучения модели, а для проверки его результатов.

3. *Валидационное множество* множество примеров, используемое как для оценки результатов обучения модели, так и для определения ее параметров.

Для разбиения исходного множества на обучающее, тестовое и валидационное необходимо настроить несколько параметров:

1. Из списка *«Способ разделения исходного множества»* выбирается порядок отбора записей во все три множества. Если выбран вариант *«по порядку»*, то порядок следования записей при их разделении не меняется. Множества последовательно формируются в соответствии с определенным для них числом записей. Если выбран вариант *«случайно»*, то отбор записей происходит случайным образом.

2. Затем необходимо указать, какие множества будут использоваться. Для того чтобы множество было сформировано, нужно установить флажок слева от его названия. Если флажок сброшен, то множество использовано не будет. Обучающее множество используется всегда, поэтому сбросить флажок для него нельзя.

3. Для каждого из используемых множеств необходимо задать его размер. Размер может быть задан непосредственно количеством записей или в процентах от объема исходной выборки. Для этого достаточно дважды щелкнуть мышью в соответствующей клетке и ввести нужное значение с клавиатуры. При этом размер, введенный в процентах, автоматически пересчитывается в количество строк и наоборот. В поле *«Количество строк (всего)»* отображается общее количество записей в исходной выборке данных, которое может быть задействовано для формирования множеств. Если суммарное число строк для всех используемых множеств меньше полного числа строк исходной выборки, то размеры множеств можно задавать произвольно. Можно, например, использовать не все записи, а только часть из них. Если же суммарное указанное число строк превышает максимальное для данной исходной выборки, то автоматиче-

ски включается баланс множеств, т. е. при указании для одного из множеств размера, в результате которого будет превышено максимальное число записей в исходной выборке, размер остальных множеств будет автоматически уменьшен таким образом, чтобы суммарный размер множеств не превышал доступного числа записей. В строке *«Итого»* указывается количество записей, задействованных во всех используемых множествах, а также процент от полного числа записей исходной выборки, который они составляют.

4. В столбце *«Порядок сортировки»* можно определить порядок следования записей внутри каждого множества. Для этого необходимо дважды щелкнуть мышью в столбце *«Порядок сортировки»* для соответствующего множества и с помощью появившейся кнопки выбора открыть список, в котором выбрать один из возможных вариантов:

– *по возрастанию* – записи в данном множестве будут следовать в порядке возрастания;

– *по убыванию* – записи в данном множестве будут следовать в порядке убывания;

– *случайно* – записи в данном множестве будут следовать в случайном порядке.

Для того чтобы обучающее множество было репрезентативным необходимо, чтобы все уникальные значения всех дискретных столбцов содержались в данном наборе данных.

Настройка параметров обучения. Необходимо установить параметры, в соответствии с которыми будет проводиться обучение дерева:

1. *«Минимальное количество примеров, при котором будет создан новый узел»* – задается минимальное количество примеров, которое возможно в узле. Если примеров, которые попадают в данный узел, будет меньше заданного – узел считается листом (т. е. дальнейшее ветвление прекращается).

2. *«Строить дерево с более достоверными правилами в ущерб сложности»* – установка данного флажка включает специальный алгоритм, который, усложняя структуру дерева, увеличивает достоверность результатов классификации. Сброс данного флажка хотя и приводит к упрощению дерева, снижает достоверность результатов классификации.

3. *«Уровень доверия, используемый при отсечении узлов дерева»*. Значение этого параметра задается в процентах и должно лежать в пределах от 0 до 100. Эти значения выбираются из списка. Чем больше уровень доверия, тем более ветвистым получается дерево, и, соответственно, чем меньше уровень доверия, тем больше узлов будет отсечено при его построении.

Контрольные вопросы

1. С какой целью проводится нормализация значений полей?
2. Для чего используется обучающая выборка? Из каких множеств она состоит?
3. Какие критерии используются для выбора параметров обучения?
4. Какие требования предъявляются к исходным данным при построении дерева решений?
5. Поясните смысл расчетных полей при анализе «что – если».

Лабораторная работа 8

Кластерный анализ с использованием карт Кохонена

Цель работы: освоение основных методов и способов кластеризации с использованием самоорганизующихся карт Кохонена, освоение принципов построения и использования простейших нейронных сетей, приобретение практических навыков по использованию инструментария Deductor Studio.

Задание

1. Разработать сценарии построения самоорганизующихся карт Кохонена.
2. Создать сводную таблицу (например, организаций), включив в нее суммарные сведения (например, о продажах, оплате и поставках). Таблицу получить путем слияния соответствующих полей из разных таблиц и последующей группировки.
3. Для подготовленной сводной таблицы разработать сценарий кластеризации с использованием самоорганизующихся карт Кохонена.
4. Создать отчеты по всем разработанным сценариям.

5. Продемонстрировать проект преподавателю с использованием тестовых наборов данных и защитить работу.

Краткая теория и методические указания

Самоорганизующиеся карты признаков Кохонена. Самоорганизующиеся карты признаков (СКП) являются разновидностью неуправляемых нейросетей. Они были предложены Тьюво Кохоненом в начале 80-х годов прошлого века и нашли широкое применение в инженерной области (для распознавания речи, в робототехнике и др.).

Технология СКП представляет собой набор аналитических процедур и алгоритмов, позволяющих преобразовывать традиционное описание множества объектов, заданных в многомерном $(n > 3)$ пространстве признаков плоской базы данных, в двумерную карту. Полученная карта устроена таким образом, что близким объектам в многомерном пространстве отвечают рядом стоящие точки (их образы) на карте.

В результате, трудно анализируемые в совокупности многомерные объекты получают простой и наглядный вид на двумерной карте, которая сохраняет их основные свойства (топологию и распределение в многомерном пространстве).

Применение технологии СКП дает ряд преимуществ:

– обнаружение групп объектов с одинаковыми характеристиками (далее – кластеров) по их локализованному расположению на специально создаваемой карте кластеров;

– проверка содержательного описания обнаруженных групп по специфическим особенностям, обнаруженным на карте признаков, а также на проекциях карты кластеров на каждый признак в отдельности;

– выявление неявных связей и закономерностей между признаками;

– проведение оценки объектов в динамике, оценка изменений как в целом по структуре кластеров, так и по отдельности;

– позиционирование на карту новых объектов для придания им статуса (рейтинга);

– прогнозирование значений одних признаков объектов через другие;

– фильтрация объектов за счет поисковых уникальных критериев, формируемых в терминах СКП.

Разберем действие нейросетевой модели самоорганизующихся карт Кохонена для маркетингового анализа.

Как уже было указано, характеристики организаций на рынке можно получить, анализируя различные показатели их работы и связи между ними. Для этого следует использовать данные финансовых отчетов. Из них эксперты извлекают значения различных параметров (активы, капитализацию, прибыль и т. д.). Но для получения достаточно достоверной информации приходится анализировать много взаимосвязей большого количества параметров. Эта задача непростая. Часто для описания субъекта рынка используется несколько десятков различных показателей, а человек обычно не может оперировать более чем тремя параметрами одновременно. Поскольку информации для анализа нужно много и чаще всего она разнородна, то невозможно окинуть одним взглядом весь этот набор.

В современном маркетинге достаточно часто возникает задача анализа данных, которые с трудом можно представить в математической числовой форме. Это случай, когда нужно извлечь данные, принципы отбора которых заданы нечетко: выделить надежных партнеров, определить перспективный товар, выявить основных конкурентов.

Предположим, что имеется информация о деятельности нескольких десятков фирм на рынке (их открытая финансовая отчетность) за некоторый период времени. По окончании этого периода исследователю известно, какие из этих фирм обанкротились, а какие продолжают стабильно работать (на момент окончания периода). И теперь необходимо решить вопрос о том, какие из них являются приоритетными с точки зрения сотрудничества. Значит, следует каким-то образом решить задачу анализа рисков сотрудничества с различными коммерческими структурами.

На первый взгляд, решить эту проблему несложно – есть данные о работе фирм и результат их деятельности. Но при этом возникает сложность, связанная с тем, что существующие данные описывают прошедший период, а исследователю интересно то, что будет в дальнейшем. Таким образом, необходимо на основании имеющихся априорных данных получить прогноз на дальнейший период. Для решения этой задачи можно использовать различные методы.

Так, например, наиболее очевидным является применение методов математической статистики. Однако недостатком подобных методов явля-

ется потребность в большом объеме априорных данных, а в выбранном примере может быть ограниченное их количество. При этом статистические методы зачастую не могут гарантировать успешный результат.

Следовательно, нужно попытаться найти эти закономерности, с тем, чтобы использовать их в дальнейшем. И тут возникает вопрос: как найти эти закономерности? Для этого, если будут применяться методы статистики, исследователь должен определить, какие критерии «похожести» использовать, а это может потребовать от него каких-либо дополнительных знаний о характере задачи.

Другим путем решения этой задачи может быть применение нейронных сетей. Метод анализа с использованием самоорганизующихся карт Кохонена – это метод, позволяющий автоматизировать все действия по поиску закономерностей. Рассмотрим, как решаются такие задачи и как карты Кохонена находят закономерности в исходных данных. Для общности рассмотрения здесь и далее будем использовать термин *объект* (например, *объектом* может быть фирма-клиент, как в рассмотренном выше примере, но описываемый метод без изменений подходит для решения и других задач, например, анализа конкуренции, поиска оптимальной стратегии поведения на рынке). В данной работе описывается способ применения указанного метода для анализа клиентов на рынке (реальных и потенциальных).

Каждый объект характеризуется набором различных параметров, которые описывают его состояние. Для примера по анализу фирм-клиентов параметрами можно взять данные из финансовых отчетов. Эти параметры часто имеют числовую форму или могут быть приведены к ней.

Как уже было указано выше, решение задачи предполагает на основании анализа параметров объектов выделение схожих объектов и представление результата в форме, удобной для восприятия. Все эти подзадачи успешно и эффективно решаются самоорганизующимися картами Кохонена. В целях упрощения рассмотрения будем считать, что объекты имеют три признака (на самом деле их может быть любое количество).

Предположим, что все эти три параметра объектов представляют собой их координаты в трехмерном пространстве. Например, для промышленного предприятия это могут быть следующие показатели: капитализация, объем реализованной продукции, прибыль. Тогда каждый объект

можно представить в виде точки в этом пространстве, что и сделаем (чтобы не было проблем с различным масштабом по осям, пронормируем все эти признаки в интервал [0,1] любым подходящим способом). В результате проведенной нормировки все точки попадут в куб единичного размера. Отобразим эти точки (рис. 2).

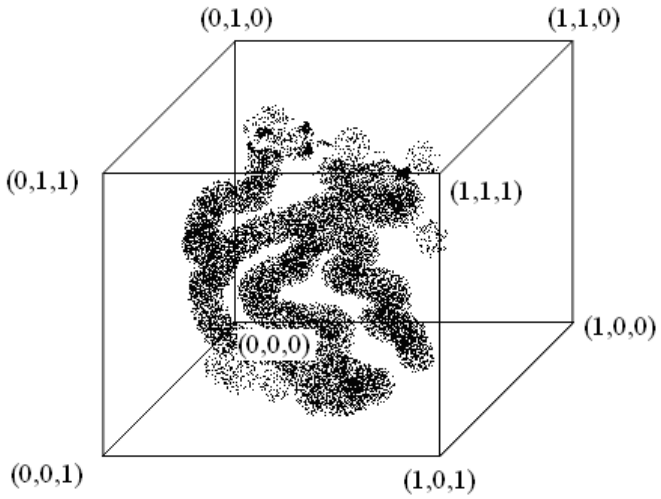

Рис. 2. Расположение объектов
в трехмерном пространстве

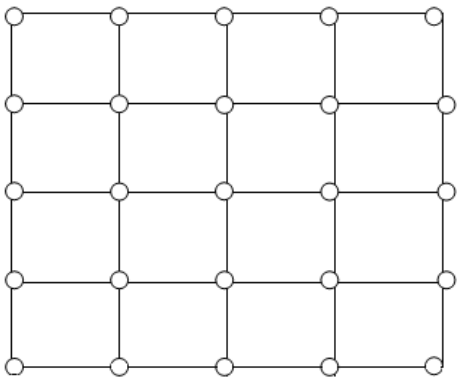

Рис. 3. Карта Кохонена

Анализ полученного рисунка позволяет увидеть, как расположены объекты в пространстве, причем легко заметить участки, где объекты группируются (сгущения). Распределение объектов таким образом означает, что у них схожи параметры, значит, и сами эти объекты принадлежат одной группе. Очевидно, что такой легкий способ можно применить только в том случае, когда признаков немного, поскольку человеческий разум не может представить изображение четырехмерного пространства. Следовательно, необходимо найти способ, которым можно преобразовать дан-

ную систему в простую для восприятия, желательно двумерную систему (потому что уже трехмерную картинку невозможно корректно отобразить на плоскости) так, чтобы соседние в изучаемом пространстве объекты оказались рядом и на полученной картинке. Для этого используем самоорганизующуюся карту Кохонена. В первом приближении ее можно представить в виде гибкой сети (рис. 3).

Эластичную сеть карты исследователь помещает в пространство признаков, где уже имеются объекты, которые необходимо проанализировать. Далее система работает следующим образом: берется один объект (точка в исследуемом пространстве) и выявляется ближайший к нему узел сети. После этого данный узел подтягивается к объекту (сетка эластична, поэтому вместе с этим узлом так же, но с меньшей силой подтягиваются и соседние узлы). Затем выбирается другой объект (точка), и процедура повторяется. В результате строится карта, расположение узлов которой совпадает с расположением основных скоплений объектов в исходном пространстве. Кроме того, полученная карта обладает следующим замечательным свойством – узлы ее расположились таким образом, что объектам, похожим между собой, соответствуют соседние узлы карты (рис. 4). Теперь следует определить, в какие узлы карты попали те или иные объекты. Это также определяется ближайшим узлом – объект попадает в тот узел, который находится ближе к нему. В результате всех этих операций объекты со схожими параметрами попадут в один узел или в соседние узлы. Таким образом, можно считать, что благодаря системе самоорганизующихся карт Кохонена исследователь решает задачу поиска похожих объектов и их группировки.

Самоорганизующиеся карты Кохонена обладают и другими возможностями. Они позволяют также представить полученную информацию в простой и наглядной форме путем нанесения раскраски. Для этого исследователь раскрашивает полученную карту цветами, соответствующими интересующим признакам объектов. Возвращаясь к примеру с анализом фирм-клиентов на рынке, можно раскрасить одним цветом те узлы, куда попала хотя бы одна фирма, у которой наблюдаются убытки. Тогда после нанесения цвета мы получим зону, которую можно назвать зоной риска, и попадание интересующей нас фирмы в эту зону говорит о ее ненадежности.

С помощью карт можно также получить информацию о зависимостях между параметрами. Отмечая на карте различные статьи финансовых и экономических отчетов отдельными цветами, менеджер-исследователь получит атлас, хранящий в себе информацию о состоянии рынка. Сравнивая расположение цветов на раскрашенных картах, подготовленных таким образом, руководитель получает полную информацию о финансовом и экономическом портрете фирм-клиентов – банкротов, неудачников, процветающих фирм, «середняков». Например, таким показателем может быть чистая прибыль фирмы.

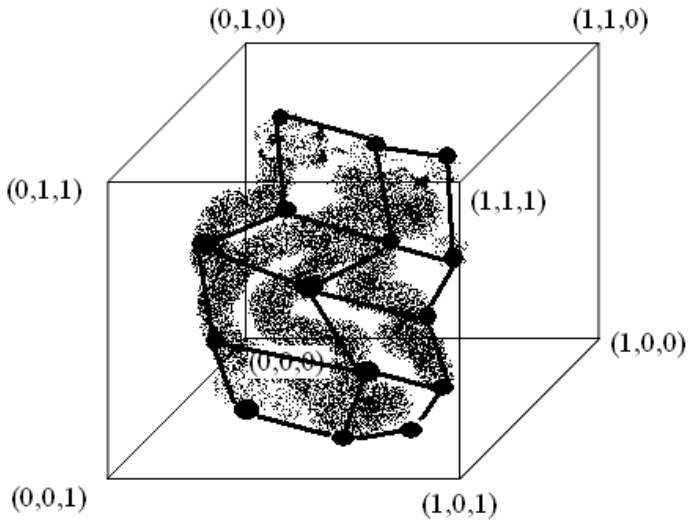

Рис. 4. Вид пространства после наложения карты

При всем этом, описанная технология является универсальным методом анализа. С ее помощью можно анализировать различные стратегии деятельности, производить анализ результатов маркетинговых исследований, проверять кредитоспособность клиентов и т. д. В данной работе технология самоорганизующихся карт Кохонена применяется для анализа клиентской базы. Этот универсальный многофункциональный инструмент анализа способен представить достаточно четкую картину реальных и потенциальных клиентов-фирм, работающих на рынке. Эта технология также полезна и потому, что в России большая часть деловой информации является засекреченной, и в результате информация, на основании которой приходится работать, крайне искажена и часто носит неправдоподобный характер.

Таким образом, имея перед собой карту, исследователь может достаточно достоверно судить об объектах, даже если имеет неполную инфор-

мацию об этих объектах. В результате, можно извлекать информацию из базы данных, основываясь на нечетких характеристиках.

В отличие от классических методов, самоорганизующиеся карты обеспечивают простую визуализацию данных, навязывают несколько меньшее количество предположений и ограничений и обнаруживают изолированные структуры в данных, оперируя с большим количеством комплексных данных.

Учитывая показанные возможности самоорганизующихся карт, можно определить следующие основные области применения этих карт в маркетинге:

– анализ товарных рынков на основании потребительских предпочтений;

– сегментирование покупателей и клиентов;

– информационное обеспечение выработки маркетинговых решений и анализа рынка;

– конкурентный анализ.

Алгоритм функционирования самоорганизующихся карт *(Self Organizing Maps – SOM)* представляет собой один из вариантов кластеризации многомерных векторов – алгоритм проецирования с сохранением топологического подобия.

Примером таких алгоритмов может служить алгоритм *k*-ближайших средних *(k-means)*. Важным отличием алгоритма *SOM* является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При этом в ходе обучения модифицируется не только нейрон-победитель (нейрон карты, который в наибольшей степени соответствует вектору входов, и определяет, к какому классу относится пример), но и его соседи, хотя и в меньшей степени. За счет этого *SOM* можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. При использовании этого алгоритма векторы, схожие в исходном пространстве, оказываются рядом и на полученной карте.

SOM подразумевает использование упорядоченной структуры нейронов. Обычно применяются одно- и двумерные сетки. При этом каждый нейрон представляет собой *n*-мерный вектор-столбец, где *n* определяется размерностью исходного пространства (размерностью входных векторов).

Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

При реализации алгоритма *SOM* заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (*neighborhood* в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, успех использования алгоритма в большой степени зависит от подходящего выбора начального радиуса обучения. Однако в случае, когда размер карты составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач. Таким образом, необходимо достигать допустимый компромисс при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов.

Существуют три способа инициирования начальных весов:

– *инициализация случайными значениями*, когда всем весам даются малые случайные величины;

– *инициализация примерами*, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;

– *линейная инициализация*, в этом случае веса инициируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных.

Обучение карты заключается в последовательности коррекции векторов, представляющих собой нейроны. На каждом шаге обучения из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве.

После того, как найден нейрон-победитель, производится корректировка весов карты. При этом вектор, описывающий нейрон-победитель, и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Обучение состоит из двух основных фаз: на первоначальном этапе выбирается достаточно большое значение скорости обучения и радиуса обучения, что позволяет расположить векторы нейронов в соответствии с распределением примеров в выборке, а затем производится точная подстройка весов, когда значения параметров скорости обучения много меньше начальных. В случае использования линейной инициализации, первоначальный этап грубой подстройки может быть пропущен.

Самоорганизующиеся карты могут использоваться для решения таких задач, как моделирование, прогнозирование, поиск закономерностей в больших массивах данных, выявление наборов независимых признаков и сжатие информации.

В результате обучения самоорганизующейся карты в исходную выборку данных будут добавлены следующие поля:

1. *<ИМЯ ПОЛЯ>_OUT* – содержит значения выходных полей, рассчитанные картой.

2. *Номер ячейки* – содержит номер ячейки карты, в которую попала данная запись.

3. *Расстояние до центра ячейки* – содержит значение расстояния от данной записи до центра ячейки, в которую эта запись попала.

4. *Номер кластера* – указывается номер кластера, где расположена ячейка, в которую попала данная запись исходной выборки.

5. *Расстояние до центра кластера* – указывается значение расстояния от ячейки, куда попала данная запись исходной выборки, до центра кластера.

6. *<ИМЯ ПОЛЯ>_ERR* – содержит среднеквадратичную ошибку рассогласования реального значения поля и значения, рассчитанного картой.

Настройка назначения полей. Здесь необходимо определить, как будут использоваться поля исходного набора данных при обучении самоорганизующейся карты и практической работе с ней. Для настройки поля следует выделить его в списке, при этом в правой части окна будут отображены его параметры:

1. *Имя поля* – идентификатор поля, определенный для него в источнике данных. Изменить его здесь нельзя.

2. *Тип данных* – тип данных, содержащихся в поле (вещественный, строковый, дата). Он также задается в источнике данных и здесь изменен быть не может.

3. *Назначение* – здесь необходимо выбрать порядок использования данного поля при обучении и работе самоорганизующейся карты Кохонена. Выбор производится с помощью списка, открываемого кнопкой и содержащего следующие варианты:

– *Входное* – поле будет использовано как одна из координат входного вектора, которые алгоритм построения карты Кохонена будет кластеризовать. По этому полю можно будет впоследствии посмотреть карту распределения значений этого поля;

– *Выходное* – при построении карты это поле использоваться не будет, однако после построения по этому полю будет собрана статистика для каждой ячейки и для каждого кластера полученной карты. Таким образом, можно говорить о выходе (классе) ячейки по этому выходному полю. Например, если выходное поле – это дискретное поле, то выходом ячейки (по этому выходному полю) будет являться самое распространенное значение выходного поля тех строчек данных, которые «попали» в данную ячейку. Если же выходное поле – это непрерывное поле, то выходом ячейки (по этому выходному полю) будет являться среднее значение выходного поля тех строчек данных, которые «попали» в данную ячейку. При таком подходе это поле можно рассматривать как целевое, как если бы мы рассматривали задачу регрессии или классификации;

– *Информационное* – поле не будет использоваться при обучении карты, но будет помещено в результирующий набор в исходном состоянии;

– *Неиспользуемое* – поле не будет использоваться при построении и работе с картой и будет исключено из результирующего набора. В отличие от непригодного, такое поле может быть использовано, если в этом возникнет необходимость;

– *Непригодное* – поле не может быть использовано при построении и работе алгоритма, но будет помещено в результирующий набор в исходном состоянии.

4. *Вид данных* – указывает на характер данных, содержащихся в поле (непрерывный или дискретный). Изменить это свойство здесь нельзя.

Статус непригодного поля устанавливается только автоматически и в дальнейшем может быть изменен только на неиспользуемое или информационное. Поле будет запрещено к использованию, если:

– поле является дискретным и содержит всего одно уникальное значение;

– непрерывное поле с нулевой дисперсией;

– поле содержит пропущенные значения.

Настройка обучающей выборки осуществляется так же, как при построении дерева решений (см. лабораторную работу 7).

Параметры карты Кохонена. В секции *«Параметры карты»* задается размер карты, т. е. количество ячеек, из которых она будет состоять. Для этого в полях *«Размер по оси X»* и *«Размер по оси Y»* следует указать количество ячеек по соответствующим координатам.

В поле *«Количество ячеек»* отображается общее число ячеек карты. Оно определяется как произведение значений полей *«Размер по оси X»* и *«Размер по оси Y»* и меняется только при их изменении.

В списке *«Форма ячеек»* выбирается один из вариантов конфигурации ячейки – прямоугольная или шестиугольная. При задании формы ячеек нужно учитывать, что шестиугольники дают более корректные результаты, так как расстояние между центрами ячеек ближе к Евклидову, чем между центрами прямоугольников. Скорость обучения выше для прямоугольной формы ячеек.

На данном шаге необходимо задать условие, при выполнении которого обучение карты будет прекращено.

«Считать пример распознанным, если ошибка меньше» – критерием останова в данном случае является условие, что рассогласование между эталонным и реальным выходом карты становится меньше заданного значения.

«По достижению эпохи» – указывается количество эпох, по достижении которого процесс обучения будет остановлен, даже, если не достигнута заданная ошибка. Позволяет избежать *«зацикливания»* в ситуациях, когда ошибка не достижима. Кроме этого, для обучающего и тестового множества в соответствующих секциях окна могут независимо устанавливаться следующие критерии останова обучения:

– *«Средняя ошибка меньше»* – средняя квадратичная ошибка на обучающем или тестовом множестве меньше заданного значения;

– *«Максимальная ошибка меньше»* – максимальная квадратичная ошибка на обучающем и тестовом множестве меньше заданного значения;

– *«Распознано примеров (%)»* – количество распознанных примеров на обучающем и тестовом множестве больше заданного процента.

При выборе нескольких условий останов процесса обучения происходит по достижении хотя бы одного из них.

Обучение карты Кохонена. Обучение карты производится итерационными циклами, каждый из которых называется эпохой. Во время каждой эпохи происходит подстройка весов нейронов карты Кохонена. Подстройка весов во время одной эпохи происходит следующим образом: каждый входной вектор (строчка таблицы) обучающей выборки «подтягивает» к себе ближайший по расстоянию нейрон (нейрон-победитель) карты Кохонена с определенной силой (скорость обучения). Вместе с нейроном-победителем подтягиваются и его соседи. Соседство определяется положением нейронов на двухмерной четырехугольной или шестиугольной сетке. Здесь расстояние – это обычное Евклидово расстояние между двумя точками в многомерном пространстве (входной вектор и веса нейрона победителя).

Способ начальной инициализации карты позволяет определить, как будут установлены начальные веса нейронов карты. Возможны три варианта:

– *случайными значениями* – начальные веса нейронов будут случайными значениями;

– *из обучающего множества* – в качестве начальных весов будут использоваться случайные примеры из обучающего множества;

– *из собственных векторов* – начальные веса нейронов карты будут проинициализированы значениями подмножества гиперплоскости, через которую проходят два главных собственных вектора матрицы ковариации входных значений обучающей выборки.

При выборе способа начальной инициализации следует руководствоваться следующей информацией:

1) объемом обучающей выборки;

2) количеством эпох, отведенных для обучения;

3) размерами обучаемой карты.

Между указанными параметрами и способом начальной инициализации существует много зависимостей. Однако можно выделить несколько главных:

1) если объем обучающей выборки значительно (раз в 100) превышает количество нейронов карты и время обучения не играет первоочередную роль, то лучше выбрать инициализацию случайными значениями, – это снизит вероятность попадания в локальный минимум ошибки кластеризации;

2) если объем обучающей выборки не очень велик, или ограничено время обучения, или необходимо уменьшить вероятность появления после обучения «пустых» нейронов (в которые не попало ни одного экземпляра обучающей выборки), то следует использовать инициализацию примерами из обучающего множества;

3) инициализацию из собственных векторов можно использовать при любом стечении обстоятельств. Единственное, вероятность появления после обучения «пустых» нейронов выше, чем если была бы использована инициализация примерами из обучающего множества. Именно этот способ необходимо выбирать при первом ознакомлении с данными.

Элементы группы *Скорость обучения* позволяют задать скорость обучения в начале и в конце обучения карты Кохонена. Значения можно выбрать из списка или ввести вручную с клавиатуры. Процесс обучения можно условно разделить на две фазы – *грубую подстройку* и *точную подстройку*. Для этапа грубой подстройки характерна достаточно большая коррекция весов нейронов сети по прохождении каждой эпохи.

На этапе точной подстройки величина коррекции значительно уменьшается. При этом коэффициент (скорость обучения), с которым многомерные координаты (веса) нейрона победителя и его соседей будут двигаться в сторону очередного экземпляра данных, изменяется в зависимости от текущей эпохи обучения по правилу, определяемому следующей функцией:

$$V = V_{начало} \times (V_{конец} / V_{начало}) \wedge (T / T_{\max}), \qquad (1)$$

где V – текущий радиус обучения;

$V_{начало}$ – начальная скорость обучения;

$V_{конец}$ – конечная скорость обучения;

$T_{\max}$ – количество эпох обучения (задается на предыдущем шаге);

T – текущая эпоха обучения.

Рекомендуемые значения для скорости обучения:

– в начале обучения 0.1–0.3;

– в конце обучения 0.05–0.005.

Элементы группы *Радиус обучения* позволяют задать *радиус обучения* в начале и в конце обучения карты Кохонена, а также *тип функции соседства*. *Радиус обучения* – это параметр, который определяет, сколько нейронов, кроме нейрона-победителя, участвуют в обучении. В процессе обучения радиус обучения обычно должен постепенно уменьшаться так, чтобы на заключительных этапах в обучении участвовал только нейрон-победитель. При этом радиус обучения изменяется в зависимости от текущей эпохи обучения по правилу, определяемому следующей функцией:

$$r = r_{начало} \times (r_{конец} / r_{начало}) \wedge (T / T_{\max}), \qquad (2)$$

где r – текущий радиус обучения;

$r_{начало}$ – начальный радиус обучения;

$r_{конец}$ – конечный радиус обучения;

$T_{\max}$ – количество эпох обучения (задается на предыдущем шаге);

T – текущая эпоха обучения.

Радиус обучения в начале должен быть достаточно большой – примерно половина или меньше размера карты (максимальное линейное рас-

стояние от любого нейрона до другого любого нейрона); а в конце должен быть достаточно малым – примерно единица или меньше. Чем больше текущий радиус обучения, тем более грубо подстраивается карта, так как приходится корректировать большое количество весов нейронов, и чем меньше текущий радиус обучения, тем более точно подстраивается карта. *Начальный радиус* обучения подбирается автоматически в зависимости от размера карты. Автоматически подобранный радиус – это всего лишь рекомендуемое значение.

Параметр *Функция соседства* определяет, какие нейроны и в какой степени будут считаться соседними по отношению к нейрону-победителю. Этот параметр может принимать два значения: *«Ступенчатая», «Гауссова»*.

Если функция соседства *Ступенчатая*, то «соседями» для нейрона-победителя будут считаться все нейроны, линейное расстояние до которых не больше текущего радиуса обучения. При этом варианте функции соседства процесс обучения происходит немного быстрее, но качество результата может быть немного хуже, чем если бы использовалась Гауссова функция соседства. Если используется *Гауссова* функция соседства, то «соседями» для нейрона-победителя будут считаться все нейроны карты, но в разной степени полноты. При этом степень соседства определяется следующей функцией:

$$h = \exp((-d \times d)/(2 \times r)), \qquad (3)$$

где h – значение, определяющее степень соседства;

d – линейное расстояние от нейрона-победителя до нейрона-«соседа»;

r – текущий радиус обучения.

После определения степени соседства очередного нейрона, его веса будут изменены не с текущей скоростью обучения, а со скоростью, равной текущей скорости обучения, умноженной на коэффициент, определяющий степень соседства – h.

При использовании Гауссовой функции соседства обучение проходит более плавно и равномерно, так как одновременно изменяются веса всех нейронов, что может дать немного лучший результат, чем если бы использовалась ступенчатая функция. Однако времени на обучение требуется

немного больше по причине того, что на всех эпохах корректируются все нейроны.

Способ определения количества кластеров может быть установлен как автоматический, так и ручной.

Уровень значимости – параметр автоматического определения кластеров. Чем больше этот параметр, тем большее количество кластеров будет получено.

Фиксированное количество кластеров – параметр, доступный при ручном определении количества кластеров. Собственно задает желаемое количество кластеров, на которое будут разбиты нейроны карты Кохонена.

Контрольные вопросы

1. Поясните необходимость использования карт Кохонена при кластеризации.

2. Объясните общий принцип построения самоорганизующейся карты признаков Кохонена.

3. Каким образом производится назначение размеров и формы ячеек (нейронов) карты Кохонена?

4. Как осуществляется назначение начальных значений весовых коэффициентов нейронов?

5. Поясните понятия скорости и радиуса обучения нейросети.

6. Какие критерии используются для остановки процесса обучения?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Анализ данных и процессов / А. А. Барсегян [и др.]. – СПб. : БХВ-Петербург, 2009. – 512 с.

2. Афонин, А. Ю. Оперативный и интеллектуальный анализ данных / А. Ю. Афонин, П. П. Макарычев. – Пермь : Изд-во ПГУ, 2010. – 142 с.

3. Компания BaseGroup™ Labs : офиц. сайт. – М., 1995–2013. URL: http:// www.basegroup.ru.

4. Малла, С. Вейвлеты в обработке сигналов / С. Малла. – М. : Мир, 2005. – 672 с.

5. Методы и модели анализа данных: OLAP и Data Mining / А. А. Барсегян [и др.]. – СПб. : БХВ – Петербург, 2004. – 336 с.

6. Сергиенко, А. Б. Цифровая обработка сигналов : учеб. для вузов / А. Б. Сергиенко. – СПб. : Питер, 2006. – 751 с.

7. Цыганенко, В. Н. Компьютерные системы поддержки принятия решений : конспект лекций / В. Н. Цыганенко, А. Г. Белик. – Омск : Изд-во ОмГТУ, 2007. – 96 с.

8. Цыганенко, В. Н. Компьютерные системы поддержки принятия решений : метод. указания к лаб. работам / В. Н. Цыганенко, А. Г. Белик. – Омск : Изд-во ОмГТУ, 2007. – 56 с.

ОГЛАВЛЕНИЕ

Редактор *О. В. Маер*

Компьютерная верстка *Ю. П. Шелехиной*

Сводный темплан 2015 г.
Подписано в печать 05.03.15. Формат $60\times84^{1}/_{16}$. Отпечатано на дупликаторе.
Бумага офсетная. Усл. печ. л. 4,25. Уч.-изд. л. 4,25.
Тираж 50 экз. Заказ 147.

Издательство ОмГТУ. 644050, г. Омск, пр. Мира, 11; т. 23-02-12.
Типография ОмГТУ.