МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

Н.В. ЗАРИКОВСКАЯ

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ

Учебно-методическое пособие для выполнения лабораторных работ и самостоятельной работы студентов

2018

Зариковская Н.В. Информационно-аналитические системы управления: Учебно-методическое пособие для выполнения лабораторных работ и самостоятельной работы студентов – Томск: Изд-во ТУСУР, 2018. – 27 с.

Учебно-методическое пособие для студентов вузов посвящено изучению аналитической платформы «Deductor» и «Аналитик». Описываются основные правила работы с данными платформами, а также особенности работы с «Deductor» и «Аналитик».

ОГЛАВЛЕНИЕ

Краткое содержание тем и результатов их освоения

Темы лабораторных работ	Деятельность студента. Решая задачи, студент:
Знакомство с аналитической платформой «Deductor»	• *изучает* платформу «Deductor» Smath Studio и Scilab;
Реализация алгоритма построения дерева решений	• *изучает* алгоритм «Построение дерева решений» • *учится* обрабатывать с его помощью данные
Логистическая регрессия и ROC-анализ	• *учится* обработке данных • *учится* прогнозированию • *использует* возможности логической регрессии и ROC анализ
Применение алгоритма кластеризации: самоорганизующиеся карты Кохонена	• *получает навыки* работы с функциями «Deductor»; • *учится* использовать метод обработки данных «самоорганизующиеся карты Кохонена»
Поиск ассоциативных правил	• *учится* выявлять ассоциативные правила • *получает навыки* работы с функциями «Deductor»;
Построение семантических сетей	• *учится* строить семантические сети • *получает навыки* работы с основными функциями «Аналитик»;

ВВЕДЕНИЕ

Назначение практикума по курсу «Информационно-аналитические системы управления» – привить навыки выполнения анализа данных с помощью средств информационно-аналитических систем управления (ИАСУ) в соответствующих предметных областях, участия в создании ИАСУ. В соответствии с этим занятия проводятся по двум направлениям:

1. Выполнение заданий по анализу массивов данных с использованием массовых и специализированных инструментальных средств методами оперативного и интеллектуального анализа.

2. Решение задач, которые стоят перед пользователями, в процессе создания и развития ИАСУ.

Работы могут выполняться студентами индивидуально или в составе группы из 2–3 человек самостоятельно или при консультировании преподавателем.

ЛАБОРАТОРНАЯ РАБОТА № 1. ЗНАКОМСТВО С АНАЛИТИЧЕСКОЙ ПЛАТФОРМОЙ «DEDUCTOR

Целью выполнения данной лабораторной работы является:

1) получение первоначальных сведений о возможностях аналитической платформы;

2) изучение основных модулей; работа с мастерами импорта, экспорта, обработки и визуализации данных.

Аналитическая платформа (АП) «Deductor» применима для решения большого спектра задач, таких как создание аналитической отчетности, прогнозирование, поиск закономерностей и пр. Можно сказать, что данная система применима в задачах, где требуется консолидация и отображение данных различными способами, построение моделей и последующее применение полученных моделей к новым данным. Рассмотрим некоторые задачи, решаемые АП:

1) Системы корпоративной отчетности. Готовое

хранилище данных и гибкие механизмы предобработки, очистки, загрузки, визуализации позволяют быстро создавать законченные системы отчетности в сжатые сроки.

2) Обработка нерегламентированных запросов. Конечный пользователь может с легкостью получить ответ на вопросы типа "Сколько было продаж товара по группам в Московскую область за прошлый год с разбивкой по месяцам?" и просмотреть результаты наиболее удобным для него способом.

3) Анализ тенденций и закономерностей, планирование, ранжирование. Простота использования и интуитивно понятная модель данных позволяет вам проводить анализ по принципу «Что, если...?», соотносить ваши гипотезы со сведениями, хранящимися в базе данных, находить аномальные значения, оценивать последствия принятия бизнес-решений.

4) Прогнозирование. Построив модель на исторических примерах, вы можете использовать ее для прогнозирования ситуации в будущем. По мере изменения ситуации нет необходимости перестраивать все, необходимо всего лишь дообучить модель.

5) Управление рисками. Реализованные в системе алгоритмы дают возможность достаточно точно определиться с тем, какие характеристики объектов и как влияют на риски, благодаря чему можно прогнозировать наступление рискового события и заблаговременно принимать необходимые меры к снижению размера возможных неблагоприятных последствий.

6) Анализ данных маркетинговых и социологических исследований. Анализируя сведения о потребителях, можно определить, кто является вашим клиентом и почему. Как изменяются их пристрастия в зависимости от возраста, образования, социального положения, материального состояния и множества других показателей. Понимание этого будет способствовать правильному позиционированию ваших продуктов и стимулированию продаж.

7) Диагностика. Механизмы анализа, имеющиеся в системе Deductor, с успехом применяются в медицинской диагностике и диагностике сложного оборудования. Например, можно построить модель на основе сведений об отказах. При ее

помощи быстро локализовать проблемы и находить причины сбоев.

Обнаружение объектов на основе нечетких критериев. Часто встречается ситуация, когда необходимо обнаружить объект, основываясь не на таких четких критериях, как стоимость, технические характеристики продукта, а на размытых формулировках, например, найти продукты, похожие на ваши с точки зрения потребителя.

Для начала работы необходимо создать новый сценарий, воспользуемся для этого мастером импорта (клавиша F6).

Импорт данных включает в себя:

– выбор типа источника данных;

– выбор файла источника данных;

– указание параметров импорта;

– указание параметров столбцов;

– выбор способа отображения данных (при выборе «Диаграммы», «Гистограммы» или «OLAP-куба» потребуется дополнительно указать параметры построения);

– указание имени, метки и описания данных.

Выполнив вышеуказанные действия по импорту данных, на панели «Сценарии» мы получим новый узел, с заданными именем, меткой и описанием.

Все способы разделены на четыре основные группы: очистка данных, трансформация данных, Data Mining, пр. Каждый способ обработки имеет название и краткое описание. Выбор способа зависит от целей обработки данных (например, сортировка и фильтрация данных, построение дерева решений и пр.).

Мастер визуализации позволяет определить способ отображения данных, указать метки и добавить описание к проекту. Запустить его можно с помощью кнопки либо клавишей F5.

Готовый проект можно экспортировать, воспользовавшись мастером экспорта (кнопка основного окна либо клавиша F8). Указав параметры, проект можно перенести в один из доступных форматов.

Задание

1. Опишите назначение и возможности АП «Deductor».

2. Запустите программу «Deductor Studio Academic», ознакомьтесь с назначением кнопок и контекстным меню главного окна программы.

3. Воспользуйтесь мастером импорта данных (импортируйте любой файл, например из C:\Program Files\ BaseGroup\ Deductor\ Samples\ *.txt).

4. Ознакомьтесь с доступными способами обработки данных.

5. Изучите возможности мастера визуализации и экспорта. Какие параметры доступны для мастера экспорта данных?

6. Создайте отчет.

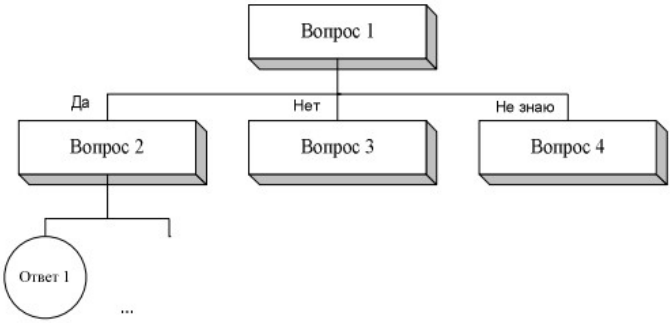
ЛАБОРАТОРНАЯ РАБОТА № 2. РЕАЛИЗАЦИЯ АЛГОРИТМА ПОСТРОЕНИЯ ДЕРЕВА РЕШЕНИЙ

Целью данной лабораторной работы является изучение алгоритма «Построение дерева решений» и научиться обрабатывать с его помощью данные.

Своевременная разработка и принятие правильного решения – это одна из главных задач работы управленческого персонала организации, т.к. необдуманное решение может дорого обойтись компании. Зачастую на практике результат одного решения заставляет нас принимать следующее решение и т. д. Когда же нужно принять несколько решений в условиях неопределенности, когда каждое решение зависит от исхода предыдущего, то применяют схему, называемую деревом решений.

Дерево решений – это графическое изображение процесса принятия решений, в котором отражены альтернативные решения, соответствующие вероятности, и выигрыши для любых комбинаций альтернатив.

Дерево решений представляет один из способов разбиения множества данных на классы или категории. Корень дерева неявно содержит все классифицируемые данные, а листья определенные классы после выполнения классификации. Промежуточные узлы дерева представляют пункты принятия решения о выборе.



Структура дерева решений

Построение дерева решений

Рисунок 1 Построение дерева решений

Структура дерева решений Построение дерева решений пусть нам задано некоторое обучающее множество T, содержащее объекты, каждый из которых характеризуется m атрибутами, причем один из них указывает на принадлежность объекта к определенному классу.

Пусть через {C1, C2, ... Ck} обозначены классы, тогда существуют 3 ситуации:

– множество T содержит один или более примеров, относящихся к одному классу Ck. Тогда дерево решений для T – это лист, определяющий класс Ck;

– множество T не содержит ни одного примера, т.е. пустое множество. Тогда это снова лист, и класс, ассоциированный с листом, выбирается из другого множества отличного от T, скажем, из множества, ассоциированного с родителем;

– множество T содержит примеры, относящиеся к разным классам. В этом случае следует разбить множество T на некоторые подмножества. Для этого выбирается один из признаков, имеющий два и более отличных друг от друга значений O1, O2, ... On. T разбивается на подмножества T1, T2, ... Tn, где каждое подмножество Ti содержит все примеры, имеющие значение Oi для выбранного признака. Эта процедура будет рекурсивно продолжаться до тех пор, пока конечное

множество не будет состоять из примеров, относящихся к одному и тому же классу.

Вышеописанная процедура лежит в основе многих современных алгоритмов построения дерева решений, этот метод известен еще под названием «разделение и захват». Очевидно, что при использовании данной методики построение дерева решений будет происходить сверху вниз.

Дерево решений является прекрасным инструментом в системах поддержки принятия решений, интеллектуального анализа данных (Data Mining). В областях, где высока цена ошибки, они послужат отличным подспорьем аналитика или руководителя.

Дерево решений успешно применяется для решения практических задач в следующих областях:

1) Банковское дело. Оценка кредитоспособности клиентов банка при выдаче кредитов.

2) Промышленность. Контроль качества продукции (выявление дефектов), испытания без разрушений (например, проверка качества сварки) и т.д.

3) Медицина. Диагностика различных заболеваний.

4) Молекулярная биология. Анализ строения аминокислот.

Это далеко не полный список областей, где можно использовать дерево решений, т.к. еще многие потенциальные области применения не исследованы.

Для загрузки данных примера импортируйте файл C:\Program Files\BaseGroup\Deductor\Samples\CreditSample.txt в АП «Deductor» с помощью мастера импорта. Все параметры импорта примите установленными по умолчанию. В окне выбора способа отображения данных выберите «Таблица», если он не выбран по умолчанию.

В результате в основном окне появится таблица, заполненная из указанного файла.

Запустите мастер обработки данных. В появившемся окне в разделе Data Mining выберете метод обработки «Дерево решений» и нажмите «Далее». Мастер обработки данных

На вкладке «Настройка значения столбцов» необходимо задать назначения столбцов данных. Почти все столбцы

автоматически получили значение «Входные». Значение поля «Выдать кредит», которое принимает только два значения «Да» или «Нет», необходимо установить в «Выходное». Также необходимо обозначить столбцы «Код» и «№ паспорта» как «Неиспользуемые» (так как значения этих столбцов уникальны, а это не позволит их классифицировать). Окно настройки назначений столбцов

Далее следует окно настройки разбиения исходного множества данных на подмножества. Оставьте это окно без изменений и нажмите кнопку «Далее».

Следующий этап – настройка параметров обучения дерева решений. Необходимо учитывать, что чем больше значение параметра «Уровень доверия, используемый при отсечении узлов дерева», тем больше будет дерево решений в итоге.

С помощью кнопки «Пуск» запускаем процесс построения дерева решений. По окончании процесса вы увидите график, отображающий уровень распознавания данных, количество узлов созданного дерева и правил, полученных в результате обработки. Процесс построения дерева решений

В последующем окне выбора способа отображения данных выберите «Дерево решений». А в последнем окне мастера обработки, по желанию, укажите имя и метку.

Результатом всех вышеописанных действий будет построенное дерево решений, которое отобразится в основном окне программы. На основании этого метода можно ответить на вопрос «Давать ли человеку кредит и если да, то при каких условиях». Готовое дерево решений

Из полученного дерева можно вывести правила выдачи кредитов. Например:

1) Если срок проживания в данной местности меньше 6,5 лет, то кредит не давать.

2) Если срок проживания в данной местности больше 6,5 лет, займ обеспечен, возраст больше 20,5 лет, не имеется недвижимость, но имеется банковский счет, то кредит давать.

Задание

1. Постройте дерево решения для описанного выше примера. Попробуйте использовать различные значения

параметров обучения дерева решения и сравните полученные деревья.

2. Выведите 5 правил из построенного дерева решений.

3. Приведите 4-5 примеров, для которых можно использовать метод обработки дерево решений, реализуйте один из них.

4. Составьте отчет.

ЛАБОРАТОРНАЯ РАБОТА № 3. ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ И ROC-АНАЛИЗ

Целью данной работы является обучение обработки данных и прогнозирование событий используя возможности логистической регрессии и ROC-анализ.

Логистическая регрессия — метод построения линейного классификатора, позволяющий оценивать апостериорные вероятности принадлежности объектов классам.

Вообще, регрессионная модель предназначена для решения задач предсказания значения непрерывной зависимой переменной, при условии, что эта зависимая переменная может принимать значения на интервале от 0 до 1. В силу такой специфики ее часто используют для предсказания вероятности наступления некоторого события в зависимости от значений некоторого числа предикторов.

При изучении линейной регрессии мы исследуем модели вида

$$y = a + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n .$$

Здесь зависимая переменная y является непрерывной, и мы определяем набор независимых переменных x_i и коэффициенты при них b_i, которые позволили бы нам предсказывать среднее значение y с учетом наблюдаемой ее изменчивости.

Во многих ситуациях, однако, y не является непрерывной величиной, а принимает всего два возможных значения. Обычно единицей в этом случае представляют осуществление какого-либо события (успех), а нулем – отсутствие его реализации (неуспех).

Среднее значение y – обозначенное через p, есть доля

случаев, в которых y принимает значение 1. Математически это можно записать как p = P(y = 1) или p = P("Успех").

ROC-кривая или кривая ошибок – показывает зависимость количества верно классифицированных положительных объектов (по оси y) от количества неверно классифицированных отрицательных объектов (по оси x).

В терминологии ROC - анализа первые называются истинно положительным, вторые – ложно отрицательным множеством. При этом предполагается, что у классификатора имеется некоторый параметр, варьируя который, мы будем получать то или иное разбиение на два класса. Этот параметр часто называют порогом, или точкой отсечения. В зависимости от него будут получаться различные величины ошибок I и II рода.

В логистической регрессии порог отсечения изменяется от 0 до 1 – это и есть расчетное значение уравнения регрессии. Будем называть его рейтингом.

Введём ещё несколько определений:

TP (True Positives) – верно классифицированные положительные примеры (так называемые истинно положительные случаи);

TN (True Negatives) – верно классифицированные отрицательные примеры (истинно отрицательные случаи);

FN (False Negatives) – положительные примеры, классифицированные как отрицательные (ошибка I рода). Это так называемый «ложный пропуск» – когда интересующее нас событие ошибочно не обнаруживается (ложно отрицательные примеры);

FP (False Positives) – отрицательные примеры, классифицированные как положительные (ошибка II рода). Это ложное обнаружение, т.к. при отсутствии события ошибочно выносится решение о его присутствии (ложно положительные случаи).

Используя мастер импорта и файл с данными, например, C:\ProgramFiles\BaseGroup\Deductor\Samples\CreditSample.txt, создайте новый сценарий и импортируйте данные.

В мастере обработки выберите способ обработки «Логистическая регрессия».

Прежде чем начнется обработка данных, необходимо провести нормализацию полей и настроить обучающую выборку. Нормализация полей проводится с цель преобразования данных к виду, подходящему для обработки средствами АП «Deductor». Например, при построении нейронной сети, линейной модели прогнозирования или самоорганизующихся карт «Входящие» данные должны иметь числовой тип (т.е. непрерывный характер), а их значения должны быть распределены в определенном диапазоне. В этом случае при нормализации дискретные данные преобразуются в набор непрерывных значений.

Настройка нормализации полей вызывается с помощью кнопки «Настройка нормализации» в нижней левой части окна «Настройка назначения столбцов». Вызов окна настройки нормализации

В окне «Настройка нормализации данных» слева приведен полный список входных и выходных полей. При этом каждое поле помечено значком, обозначающим вид нормализации:

– линейная – линейная нормализация исходных значений;

– уникальные значения – преобразование уникальных значений в их индексы;

– битовая маска – преобразование дискретных значений в битовую маску.

В правой части окна для выделенного поля отображаются параметры нормализации. Окно настройки нормализации данных

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 могут быть использованы два вида нормализации - уникальные значения и битовая маска.

Если дискретные значения преобразуются в битовую маску (т.е. каждому уникальному значению ставится в соответствие уникальная битовая комбинация), то возможны два способа такого преобразования, выбираемые из списка «Способ кодирования»:

1) Позиция бита - поле в этом случае представляется в виде n битов, где n - число уникальных значений в поле. Каждый бит

соответствует одному значению. В 1 устанавливается только бит, соответствующий текущему значению, принимаемому полем, все остальные биты равны 0. Этот способ кодирования используется при малом числе уникальных значений.

2) Комбинация битов - каждому уникальному значению соответствует своя комбинация битов в двоичном виде.

Настройка обучающей выборки - разбиение обучающей выборки на два множества - обучающее и тестовое - для построения линейной модели. Пример настройки обучающей выборки

Обучающее множество - включает записи, которые будут использоваться в качестве входных данных, а также соответствующие желаемые выходные значения.

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

Примечание. Обучение может с большой долей вероятности считаться успешным, если процент распознанных примеров на обучающем и тестовом множествах достаточно велик.

Следующий этап – настройка параметров остановки обучения, которая включает определение максимального числа итераций (заданная точность), задание функции правдоподобия, порога отсечения и допустимость ошибки.

Итогом проведения регрессионного анализа будет построенная ROC-кривая.

Задание

1. С помощью мастера импорта откройте файл (например, C:\ProgramFiles\BaseGroup\Deductor\Samples\ CreditSample.txt).
2. В мастере обработки выберите «Логистическая регрессия».
3. Проведите настройку нормализации полей.
4. Настройте обучающую выборку.
5. Проанализируйте полученные данные.
6. Создайте отчет.

ЛАБОРАТОРНАЯ РАБОТА № 4. ПРИМЕНЕНИЕ АЛГОРИТМА КЛАСТЕРИЗАЦИИ: САМООРГАНИЗУЮЩИЕСЯ КАРТЫ КОХОНЕНА

Целью данной лабораторной работы является обучение использования метода обработки данных «Самоорганизующиеся карты Кохонена».

Иногда возникают задачи 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 и т.п. Таким образом, необходимо на основании имеющихся у нас априорных данных получить прогноз на дальнейший период. Для решения этой задачи можно использовать различные методы.

Так, например, наиболее очевидным является применение методов математической статистики. Но тут возникает проблема с количеством данных, ибо статистические методы хорошо работают при большом объеме априорных данных, а у нас может быть ограниченное их количество. При этом статистические методы не могут гарантировать успешный результат.

Другим путем решения данной задачи может быть применение нейронных сетей, которые можно обучить на имеющемся наборе данных. В этом случае в качестве исходной информации используются данные финансовых отчетов различных банков, а в качестве целевого поля – итог их деятельности.

Но при использовании описанных выше методов мы навязываем результат, не пытаясь найти закономерности в исходных данных. Можно попытаться найти эти закономерности с тем, чтобы использовать их в дальнейшем. И тут перед нами возникает вопрос о том, как это сделать.

Существует метод, позволяющий автоматизировать все действия по поиску закономерностей – метод анализа с использованием самоорганизующихся карт Кохонена.

Самоорганизующаяся карта Кохонена (англ. Self-organizing map — SOM) — нейронная сеть с обучением без учителя,

выполняющая задачу визуализации и кластеризации. Является методом проецирования многомерного пространства в пространство с более низкой размерностью (чаще всего двумерное), применяется также для решения задач моделирования, прогнозирования и др.

Рассмотрим, как решаются такие задачи и как карты Кохонена находят закономерности в исходных данных. Для общности рассмотрения будем использовать термин «объект» (например, объектом может быть банк, однако описываемая методика без изменений подходит для решения и других задач – например, анализа кредитоспособности клиента, поиска оптимальной стратегии поведения на рынке и т.д.).

Каждый объект характеризуется набором различных параметров, которые описывают его состояние. Например, параметрами будут данные из финансовых отчетов. Эти параметры часто имеют числовую форму или могут быть приведены к ней.

Таким образом, нам надо на основании анализа параметров объектов выделить схожие объекты и представить результат в форме, удобной для восприятия.

Импортируйте в АП «Deductor» исходные данные из файла C:\Program\Files\BaseGroup\Deductor\Samples\CreditSample.txt.
Процесс построения карты Кохонена состоит из 10 этапов. Далее рассмотрим эти этапы подробнее.

Затем запустите мастер обработки, в котором в разделе «Data Mining» выберете способ обработки данных «Карта Кохонена», нажмите «Далее».

В окне настройки назначения столбцов необходимо обозначить столбцы «Код» и «№ паспорта» как «Неиспользуемые» (так как значения этих столбцов уникальны, а это не позволит их классифицировать по общим признакам). Определите поле «Давать кредит» как «Выходное».

Настройку обучающей выборки и параметров карты Кохонена можно оставить без изменений.

Настройте параметры остановки обучения, указав уровень допустимой погрешности, если он будет превышен, анализ данного множества будет прекращен. Можно оставить значения

«по умолчанию».

Далее запустите процесс построения карты Кохонена, нажав кнопку «Пуск».

На вкладке «Выбор способа отображения данных» поставьте галочку напротив пункта «Самоорганизующаяся карта Кохонена».

Теперь необходимо провести настройку отображения карты: отметьте разделы «Давать кредит» и «Кластеры» и другие разделы по желанию

Щелкнув левой клавишей мыши по любому шестиугольнику на любой карте, выделятся соответствующие ему ячейки на остальных картах, в том числе на картах «Давать кредит» и «Кластеры». При этом на шкалах в нижней части карт отобразятся значения соответствующих параметров.

Задание

1. Выполните описанные выше действия по построению карт Кохонена. Проанализируйте результаты, что можно сказать о вероятности возврата кредита для групп 2, 3 и 4?

2. Используя различные отображения карты Кохонена, постройте 3-4 правила выдачи кредитов.

3. Ответьте на вопросы: – для чего используются карты Кохонена? – по какому принципу происходит перенос многомерного пространства на пространство меньшей размерности?

4. Подготовьте отчет.

ЛАБОРАТОРНАЯ РАБОТА № 5. ПОИСК АССОЦИАТИВНЫХ ПРАВИЛ

Целью данной лабораторной работы является обучение выявлению ассоциативных правил с помощью АП «Deductor»

В последнее время неуклонно растет интерес к методам «обнаружения знаний в базах данных». Объемы современных баз данных, которые весьма внушительны, вызвали устойчивый спрос на новые масштабируемые алгоритмы анализа данных. Одним из популярных методов обнаружения знаний стали алгоритмы поиска ассоциативных правил.

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила, служит утверждение, что покупатель, приобретающий хлеб, приобретет и молоко с вероятностью 75 %. Первый алгоритм поиска ассоциативных правил, называвшийся AIS, был разработан в 1993 году сотрудниками исследовательского центра IBM Almaden. На середину 90-х годов прошлого века пришелся пик исследовательских работ в этой области.

Ассоциативные правила (Association Rules) Впервые эта задача поиска ассоциативных правил была предложена для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины.

Пусть имеется база данных, состоящая из покупательских транзакций. Каждая транзакция – это набор товаров, купленных покупателем за один визит. Такую транзакцию еще называют рыночной корзиной.

Покажем на конкретном примере: «75 % транзакций, содержащих хлеб, также содержат молоко. 3 % от общего числа всех транзакций содержат оба товара». 75 % – это достоверность правила, 3 % – это поддержка, или хлеб, молоко с вероятностью 75 %.

Другими словами, целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов x, то на основании этого можно сделать вывод о том, что другой набор элементов y также должен появиться в этой транзакции. Установление таких зависимостей дает нам возможность находить очень простые и интуитивно понятные правила.

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил x y, причем поддержка и достоверность этих правил должны быть выше некоторых наперед определенных порогов, называемых соответственно минимальной поддержкой и минимальной достоверностью.

Задача нахождения ассоциативных правил разбивается на две подзадачи:

1. Нахождение всех наборов элементов, которые удовлетворяют порогу минимальной поддержки. Такие наборы элементов называются часто встречающимися.

2. Генерация правил из наборов элементов, найденных согласно п.1 с достоверностью, удовлетворяющей порогу минимальной достоверности.

Один из первых алгоритмов, эффективно решающих подобный класс задач, – это алгоритм APriori. Кроме этого алгоритма в последнее время был разработан ряд других алгоритмов: DHP, Partition, DIC и другие.

Значения для параметров минимальная поддержка и минимальная достоверность выбираются таким образом, чтобы ограничить количество найденных правил. Если поддержка имеет большое значение, то алгоритмы будут находить правила, хорошо известные аналитикам или настолько очевидные, что нет никакого смысла проводить такой анализ. С другой стороны, низкое значение поддержки ведет к генерации огромного количества правил, что, конечно, требует существенных вычислительных ресурсов. Тем не менее большинство интересных правил находится именно при низком значении порога поддержки. Хотя слишком низкое значение поддержки ведет к генерации статистически необоснованных правил.

Поиск ассоциативных правил совсем не тривиальная задача, как может показаться на первый взгляд. Одна из проблем - алгоритмическая сложность при нахождении часто встречающихся наборов элементов, т.к. с ростом числа элементов в I (| I |) экспоненциально растет число потенциальных наборов элементов.

Рассмотрим еще некоторые понятия.

Транзакция - множество событий, произошедших одновременно. В нашем случае, каждая транзакция - набор товаров, купленных покупателем за один визит.

Минимальная и максимальная поддержка. Ассоциативные правила ищутся только в некотором множестве всех транзакций. Для того чтобы транзакция вошла в это множество, она должна встретиться в исходной выборке количество раз, большее минимальной поддержке и меньше максимальной.

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.

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.

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.

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.

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

Правило со слишком большой достоверностью практической ценности в контексте решаемой задачи не имеет, т.к. товары, входящие в следствие, покупатель, скорее всего, уже

купил.

Максимальная мощность искомых часто встречающихся множеств. Если данный 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 больше, чем из элементов.

Если задать значение параметра максимальная мощность, то можно искать правила, которые состоят не более чем из 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 ТоварI, то ТоварJ». Ограничение поиска часто встречающихся множеств по мощности может также понадобиться, если при указанном значении минимальной поддержки количество часто встречающихся множеств, имеющих большую мощность, слишком велико.

Импортируйте в АП «Deductor» данные файла C:\Program Files\BaseGroup\Deductor\Samples\Supermarket.txt, изменив при этом вид данных столбца «Номер чека» на дискретный.

В разделе «Data Mining» мастера обработки выберите пункт «Ассоциативные правила».

Установите назначение поля «Номер чека» – транзакция, а поля «Товар» – элемент.

Установите параметры построения ассоциативных правил, используя информацию, изложенную в теоретической части данного раздела.

Далее запустите процесс поиска ассоциативных правил, нажав кнопку «Пуск».

Выбираем способ отображения данных «Правила» в разделе «Data Mining». В завершение указываем значения полей «Имя», «Метка», «Описание».

Задание

1. Выполните действия, описанные выше, используя различные параметры построения ассоциативных правил. Сравните полученные результаты, объясните их.

2. Ответьте на вопросы: – какой товар с наибольшей достоверностью берут с вафлями? – человек взял мед и сыры, какой один из товаров он скорее всего не возьмёт? – назовите 5 самых популярных наборов товаров (в наборе может быть один или несколько товаров).

3. Опишите 4-5 ассоциативных правил, полученных в ходе выполнения работы.

4. Где еще, кроме торговли, можно использовать ассоциативные правила? Приведите примеры.

5. Составьте отчет

ЛАБОРАТОРНАЯ РАБОТА № 6. ПОСТРОЕНИЕ СЕМАНТИЧЕСКИХ СЕТЕЙ

Целью данной лабораторной работы является обучение построению семантических сетей АРМ «Аналитик»

Визуальное отображение данных исследования позволяет наиболее ясно увидеть картину. Проследить прямые, косвенные связи, выявить причинно-следственные взаимодействия. Такая картина сама должна быть максимально ясной и логически интуитивно понятной. Временные процессы следует располагать слева направо, иерархии строить сверху вниз. Процессы, являющиеся подпроцессами одного единого, – объединять рамками. Следует избегать множественного пересечений связей. Связь один ко многим следует отображать одной связью, объединив объекты рамкой.

Анализ собранных данных по объекту начинается с изучения связанных с ним фактов в инспекторе свойств или на динамической раскладке семантической сети. Удобно разбираться в данных, создав экземпляр рабочей области и формируя раскладки с фрагментами данных. Далее в такие раскладки постепенно добавлять новые «куски» сети.

В витрине АРМ «Аналитик» создайте «Рабочую область» «Аффилированные лица».

Для этого нажмите на панели инструментов кнопку «Поиск элементов по типу».

При помощи кнопки «Создание экземпляра» создайте

экземпляр типа «Рабочая область» с заданным наименованием. Нажмите кнопку «ОК»

Для перехода в режим «Семантическая сеть» выберите данный тип визуализатора в витрине «Аналитика».

Создана новая рабочая область для построения семантической сети. Перед началом работы нажмите кнопку «Выделение элементов», т.к. по умолчанию стоит настройка защиты от редактирования.

Для вызова панели редактирования нажмите кнопку  . В нижней части экрана появится панель редактирования, которая необходима для построения семантической сети.

Начинаем строить семантическую сеть с добавления главного объекта, в нашем примере «Иванов Сергей Владимирович». Нажмите кнопку «Добавить объект» на панели редактирования. Откроется стандартная панель выбора экземпляров. В строке поиска наберите «Иванов» и нажмите кнопку «Поиск по имени». Из найденных экземпляров выберите нужный, с помощью стрелочки переместите его в правую часть окна и нажмите кнопку «Ок».

Экземпляр «Иванов Сергей Владимирович» отобразится на семантической сети. Далее выделите объект (щелчком мыши), справа находится всплывающая панель инспектора свойств. Чтобы закрепить ее, щелкните по скрепке.

На закладке «Связи» отображены все связанные с объектом экземпляры. Выделите один или несколько экземпляров и нажатием правой клавишей мыши выберите «Вставить в семантическую сеть».

Таким образом, постепенно наполняется семантическая сеть, распределяются экземпляры.

Для сохранения результата работы воспользуйтесь кнопкой «Сохранить».

Дайте наименование раскладке, например, «Аффилированные лица Иванова С.В.».

Для придания раскладке эстетичного вида воспользуйтесь кнопкой «Форматирование».

Из выпадающего списка выберите пункт «Внешний вид элементов». Откроется окно «Внешний вид элементов».

В этом окне можно редактировать внешний вид экземпляров: изменять размеры, цвет и т.д. По умолчанию настроены цвета в соответствии с типом объекта.

Чтобы сохранить раскладку в формате программы MS Visio, нажмите кнопку «Экспорт в VISIO». Последовательно выполняйте шаги мастера экспорта. Вы получите изображение в формате «.vsd».

Можно воспользоваться шаблоном оформления заголовка и легенды, либо создать свой фирменный шаблон и экспортировать раскладки в него. На семантической сети есть возможность расположения экземпляров в виде таблицы.

Задание

1. Запустите АРМ «Аналитик».
2. Найдите в БД объект для построения семантической сети.
3. Создайте рабочую область «Аффилированных объектов».
4. Постройте семантическую сеть.
5. Экспортируйте полученный результат в MS Visio.

Указания к самостоятельной работе студентов (СРС) и контрольные вопросы для оценивания

Вид самостоятельной работы:

1. Выполнение индивидуальных заданий.
2. Текущая проработка теоретического материала учебников и лекций, в том числе тем, вынесенных для самостоятельного изучения:

- построение семантических сетей;
- построение карт Кохонена;
- поиск ассоциативных правил.

Изучение программы курса:

На лекциях преподаватель рассматривает вопросы программы курса, составленной в соответствии с государственным образовательным стандартом. Из-за недостаточного количества аудиторных часов некоторые темы не удается осветить в полном объеме, поэтому преподаватель, по своему усмотрению, некоторые вопросы выносит на самостоятельную работу студентов, рекомендуя ту или иную

литературу.

Кроме этого, для лучшего освоения материала и систематизации знаний по дисциплине, необходимо постоянно разбирать материалы лекций по конспектам и учебным пособиям. В случае необходимости обращаться к преподавателю за консультацией.

Список использованной литературы

1. Методы и модели анализа данных: OLAP и Data Mining. / А.А. Барсегян, М.С. Куприянов, В.В.Степаненко, И.И. Холод – СПб.: БХВ-Петербург, 2004. – 336 с.: ил

2. Джексон П. Введение в экспертные системы: [пер. с англ.] / П. Джексон. - 3-е изд. – М.: издательский дом «Вильямс», 2001. – 624 с.

3. Data Mining и аналитическая платформа Deductor [Электрон. ресурс]: [статья]. М., 2008. – Режим доступа: http://sttc.ru/index.php?option=com_content&task=view&id=56&Itemid=90

4. И. Ю. Нежданов. «Аналитическая разведка для бизнеса». – М.: издательство «Ось-89», 2008.