Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Томский государственный университет систем управления и радиоэлектроники»

Кафедра компьютерных систем в управлении и проектировании

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ТЕХНИКО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

Методические указания к лабораторным работам

Томск 2018

Кочергин М.И, Ганджа Т.В.

Информационные технологии в технико-экономических системах / Методические указания к лабораторным работам. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2018. – 23 с.

Методическое пособие для студентов вузов технических направлений посвящено изучению таких разделов современных информационных технологий как кластеризация данных, анализ временных рядов, хранилища данных, а также их применению для решения задач в области логистики, энергетики, промышленности. Рассматривается работа в следующих программных продуктах: Excel, Matlab, KNIME, Deductor Studio (Deductor Warehouse).

ОГЛАВЛЕНИЕ

Лабораторная работа 1. Введение в KNIME Analytics Platform

1. Цель работы

Изучение основ работы в среде KNIME, формирование навыков построения моделей обработки данных.

2. Указания к выполнению работы

Knime Analytics Platform – платформа с открытым исходным кодом для анализа данных.

В Knime процедура построения алгоритма обработки данных осуществляется через создание визуальной схемы – *Workflow*. *Workflow* состоит из узлов которые выполняют ту или иную функцию (например чтение данных из БД, трансформация, визуализация). Узлы, соответственно, соединяются между собой стрелочками которые показывают направление движение данных.

Workflow представлет собой исполняемую программу и может быть запущен (с любого узла) для выполнения заложенного в него алгоритма (или последовательности алгоритмов) обработки данных. После того как *workflow* запущен на исполнение, в базовом сценарии узлы *workflow* начинают отрабатывать один за одним, начиная с самого первого. Если в ходе выполнения того или иного узла произошла ошибка, то исполнение всей ветки следующей за ним прекращается.

Workflow состоит из узлов (*nodes*). У каждого узла есть окно свойств (конфигурации), где можно произвести его настройку.

Все узлы разбиты на категории (рис. 1.1), при этом пользователю предоставляется возможность скачать любой дополнительный раздел (даже те, которые находятся на стадии доработки – раздел *KNIME Labs*) с официального сайта *KNIME* или создать набор своих узлов.

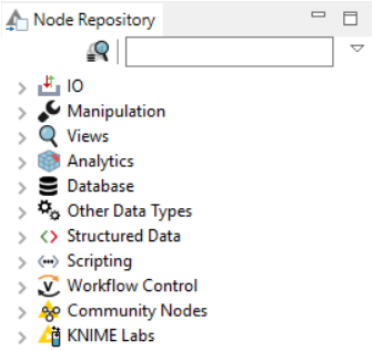

Рис. 1.1 – Окно категорий узлов в KNIME

Поддерживается следующие типы узлов: *IO* – ввод/вывод (*input*/*output*) данных (например чтение файлов *CSV*), *Manipulation* – преобразование данных (включая фильтрацию строк, столбцов, сортировку), *Views* – визуализация данных (построение различных графиков включая гистограммы, круговые диаграммы и пр.), *Database* – возможность чтения/записи данных из/в базу данных, *Workflow Control* – создание циклов, итерирование групп в ходе выполнения workflow и др.

В качестве примера узлов реализующих анализ данных можно привести следующие: статистические методы (линейная корреляция, проверку гипотез и пр.), методы интеллектуального анализа данных (*Data Minig*, например нейронные сети, построение деревьев решений, кластеризация).

Пример простого *workflow* приведён на рис. 1.2.

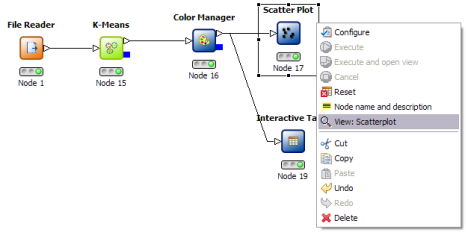

Рис. 1.2 – Пример worklflow

ETL включает все функции, необходимые для предварительной обработки и очистки данных.

Предварительная обработка и очистка данных являются наиболее важным шагом перед применением передовых методов аналитики к данным. Почти 80% работы во многих проектах аналитики занято предварительной обработкой и очисткой данных.

Платформа *KNIME Analytics* предоставляет более 100 узлов для предварительной обработки, манипулирования и трансформации, что позволяет удовлетворить все требования *ETL*. Большинство этих узлов можно найти в категории «Манипуляции данными».

Манипуляция данными включает в себя: фильтрацию строк и столбцов, биннинг (бинаризация), конфигурацию данных, манипуляцию строк, замену ячеек (в особенности замещение отсутствующих значений), объединение и конкатенация, расщепление ячеек и комбинирование ячеек, сдвиг данных, сортировка, нормализация и др.

3. Содержание работы

1. Изучение и запуск схемы (workflow) в *KNIME*
2. Импорт данных
3. Предобработка данных

4. Преобразование данных
5. Визуализация данных
6. Составление отчётов

4. Порядок проведения работы

Задание 1. Запустите и изучите схему *003_Preprocessing/003002_StandardPreprocessing* в *KNIME*. Ответьте на вопрос: «Какие операции выполняет данная схема».

Задание 2. Импортируйте данные использованием: *CSV Reader node*, *File Reader node*. Произведите чтение *XLS* и *CSV* файлов. Запустите схему *004_Databases/004001_Database_SimpleIO* и проведите в нейпростые манипуляции (сортировка, визуализация данных).

Задание 3. Загрузите схему *003001_SimpleFlow_with_Statistics*. Осуществите выборку данных из этой схемы по критерию «Пол». Сгруппируйте данные. Объедините таблицы и визуализируйте данные.

Задание 5 Создание отчётов. Откройте схему *010001_simpleExample* из раздела *010_Reporting* . Модифицируйте схему так, чтобы отчёт формировался: 1) в файле *PDF*, 2) в формате *HTML*.

5. Контрольные вопросы

– Как осуществляется импорт данных в *KNIME*?
– Какие форматы можно использовать для формирования отчётов в KNIME?
– Как визуализировать данные в *KNIME*?

6. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов).

Лабораторная работа 2. Решение задач комбинаторного программирования

1. Цель работы

Изучение способов решения комбинаторных задач, формирование навыков решения комбинаторных задач с помощью программных средств.

2. Указания к выполнению работы

Целый ряд математических моделей процессов управления представляет собой дискретные модели комбинаторного типа, например, транспортные задачи, задачи по составлению расписаний; составление планов производства и реализации продукции и т.д. Рассмотрим основные типы задач:

1. Задачи определения очередности выполнения заданий. Для заданного множества заданий (работ, операций) исполнителю требуется выбрать наилучшую последовательность их выполнения. Понятие «наилучшей» последовательности зависит от конкретных условий и чаще всего определяется директивными сроками окончания отдельных заданий.

2. Задачи определения порядка обработки деталей. При заданных временах и последовательностях обработки деталей найти такой порядок их запуска, при котором суммарное время обработки минимально.

3. Задачи распределения заданий. Для заданного множества заданий и заданного множества исполнителей распределить задания наиболее рационально (с точки зрения затрат ресурсов или времени на их выполнение).

4 Задача о назначениях. Задача о назначениях является частным случаем задачи распределения заданий: она состоит, таким образом, в назначении каждому исполнителю ровно одного задания.

Для решения задачи в *Excel* рекомендуется использовать сервис «Поиск решения» (надстройка *Solver*).

Для решения задачи в *Matlab* необходимо создать соответствующие *m*-функции и *m*-скрипты, содержащие алгоритм решения задачи. Внесение входных данных может быть реализовано через чтение исходных данных из созданного в Excel файла.

3. Содержание работы

1. Решение задачи в *Excel*
2. Решение задачи в *Matlab*

4. Порядок проведения работы

Задание 1. Проведите формализацию задачи. Составьте и внесите в таблицу начальные данные.

Задание 2. Решите задачу в *Excel*.

Задание 3. Решите задачу в *Matlab*, создав требуемые *m*-функции и *m*-скрипты.

Задание 4. Сравните результаты. Сделайте выводы о проделанной работе. Составьте отчёт о проделанной работе.

5. Варианты заданий

1. Задачи определения очередности выполнения заданий.
2. Задачи определения порядка обработки деталей.
3. Задачи распределения заданий.
4 Задача о назначениях.

Исходные данные для работ предоставляются преподавателем в электронном виде и относятся к одной из областей: промышленность, энергетика, строительство, транспорт, логистика, услуги, агропромышленный комплекс.

6. Контрольные вопросы

– Постановка задачи определения очередности выполнения заданий.
– Постановка задачи определения порядка обработки деталей.
– Постановка задачи распределения заданий
– Постановка задачи о назначениях
– Решение задач комбинаторного программирования в *Excel*
– Решение задач комбинаторного программирования в *Matlab*

7. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов), 5) *приложение* (код программы с построчными комментариями).

Лабораторная работа 3. Определение оптимального маршрута на графе

1. Цель работы

Изучение методов решения задач оптимизации на графах, формирование навыков решения задачи поиска оптимального маршрута на графе.

2. Указания к выполнению работы

Задача определения кратчайшего пути в графе имеет большое значение в практических применениях. К ней сводятся многие задачи выбора наиболее экономичного (с точки зрения расстояния или стоимости) маршрута на имеющейся карте дорог, наиболее экономичного способа перевода динамической системы из одного состояния в другое и т.д. Существует много математических способов решения, но часто методы, основанные на теории графов, наименее трудоемки.

Наиболее часто для решения задачи поиска кратчайшего пути в графе используется метод Дейкстры. Алгоритм основан на приписывании вершинам временных пометок, причем пометка вершины дает верхнюю границу длины пути от s к этой вершине. Величины этих пометок постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации точно одна из временных пометок становится постоянной. Это означает, что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от s к рассматриваемой вершине.

Для решения задачи в *Excel* рекомендуется использовать сервис «Поиск решения» (надстройка *Solver*).

Для решения задачи в *Matlab* необходимо создать соответствующие m-функции и m-скрипты, содержащие алгоритм решения задачи. Внесение входных данных может быть реализовано через чтение исходных данных из созданного в Excel файла.

3. Содержание работы

1. Решение задачи в *Excel*
2. Решение задачи в *Matlab*

4. Порядок проведения работы

Задание 1. Проведите формализацию задачи. Составьте и внесите в таблицу начальные данные.

Задание 2. Решите задачу в *Excel*.

Задание 3. Решите задачу в *Matlab*, создав требуемые *m*-функции и *m*-скрипты.

Задание 4. Сравните результаты. Сделайте выводы о проделанной работе. Составьте отчёт о проделанной работе.

5. Варианты заданий

1. Поиск кратчайшего пути в графе
2. Решение задачи обхода всех вершин в графе
3. Решение задачи обхода всех ветвей в графе

Исходные данные для работ предоставляются преподавателем в электронном виде и относятся к одной из областей: промышленность, энергетика, строительство, транспорт, логистика, услуги, агропромышленный комплекс.

6. Контрольные вопросы

- Постановка задачи поиска кратчайшего пути на графе
- Постановка задачи поиска гамильтонова цикла в графе
- Постановка задачи поиска эйлерова цикла в графе
- Решение задач оптимизации на графах в *Excel*
- Решение задач оптимизации на графах в *Matlab*

7. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов), 5) *приложение* (код программы с построчными комментариями).

Лабораторная работа 4. Решение задач оптимизации в Excel и Matlab

1. Цель работы

Формирование навыков решения задач многомерной оптимизации в пакете Matlab и табличном редакторе Excel.

2. Указания к выполнению работы

Задачи оптимизации можно разделить на 2 типа: условные (с ограничениями) и безусловные (без ограничений).

Решение задач 2-го типа подразумевает поиск абсолютного минимума функции. Для решения таких задач могут применяться методы: покоординатного спуска, градиентного спуска, Левенберга-Маркгвардта и др.

Выбор метода для решения задач 1-го типа зависит от типа ограничений (линейные или нелинейные) и вида целевой функции.

В большинстве реальных задач оптимизации, представляющих практический интерес, целевая функция зависит от многих проектных параметров.

Во многих случаях никакой формулы для целевой функции нет, а имеется лишь возможность определения ее значений в произвольных точках рассматриваемой области с помощью некоторого вычислительного алгоритма или путем физических измерений. Задача состоит в приближенном определении наименьшего значения функции во всей области при известных ее значениях в отдельных точках.

В данной работе для визуализации поверхностей и построения трёхмерных графиков рекомендуется использовать среду *Matlab* или его аналоги (например, *GNU Octave*).

В среде *KNIME* также имеются средства для решения задач оптимизации (с использованием генетического алгоритма).

Создание массивов данных для трехмерной графики в *Matlab*

Поверхности как объекты трехмерной графики обычно описываются функцией двух переменных z(x,y). Специфика построения трехмерных графиков требует не просто задания ряда значений x и y, то есть векторов x и y. Она требует определения для X и Y двумерных массивов – матриц. Для создания таких массивов служит функция meshgrid. В основном она используется совместно с функциями построения графиков трехмерных поверхностей. Функция meshgrid записывается в следующих формах:

• [X,Y] = meshgrid(x,y) преобразует область, заданную векторами x и y, в массивы X и Y, которые могут быть использованы для вычисления функции двух переменных и построения трехмерных графиков. Строки

выходного массива *X* являются копиями вектора *x*; а столбцы *Y* – копиями вектора *y*;

• [X,Y] = meshgrid(x) аналогична [X,Y] = meshgrid(x,x);

• [X,Y,Z] = meshgrid(x,y,z) возвращает трехмерные массивы, используемые для вычисления функций трех переменных и построения трехмерных графиков.

Функция *ndgrid* является многомерным аналогом функции *meshgrid*.

Графики поверхностей в Matlab

Команда *plot3*(…) является аналогом команды *plot*(…), но относится к функции двух переменных *z*(*x*,*y*). Она строит аксонометрическое изображение трехмерных поверхностей и представлена следующими формами:

• *plot3*(*x*,*y*,*z*) строит массив точек, представленных векторами x, y и z, соединяя их отрезками прямых. Эта команда имеет ограниченное применение;

• *plot3*(*X*,*Y*,*Z*), где *X*, *Y* и *Z* – три матрицы одинакового размера, строит точки с координатами *X*(*i*,:), *Y*(*i*,:) и *Z*(*i*,:) и соединяет их отрезками прямых.

• *plot3*(*X*,*Y*,*Z*,*S*) обеспечивает построения, аналогичные рассмотренным ранее, но со спецификацией стиля линий и точек, соответствующей спецификации команды *plot*.

3. *Содержание работы*

1. Решение задачи безусловной оптимизации в *Excel*
2. Решение задачи условной оптимизации в *Excel*
3. Решение задачи безусловной оптимизации в *Matlab*
4. Решение задачи условной оптимизации в *Matlab*

4. Порядок проведения работы

Задание 1. Решите задачу минимизации целевой функции от нескольких аргументов в *Excel*, используя сервис «Поиск решения».

Задание 2. Решите задачу минимизации нелинейной целевой функции от нескольких аргументов при линейных ограничениях в *Excel*, используя сервис «Поиск решения».

Задание 3. Решите задачу минимизации целевой функции от нескольких аргументов в *Matlab*, используя метод согласно варианту.

Задание 4. Решите задачу минимизации линейной целевой функции от нескольких аргументов при линейных ограничениях в *Matlab*, используя встроенные функции.

Задание 5. Оцените и сравните полученные результаты. Сделайте выводы о проделанной работе.

5. *Варианты заданий*

Варианты к заданиям 1, 3 представлены в таблице 4.1.

Таблица 4.1 – Функции двух переменных

Вариант	Целевая функция
1	$y_1 = \frac{x_1 - x_2}{x_1 + x_2}$
2	$y_1 = 1{,}5 \cdot x_1 + x_2^{\,3}$
3	$y_1 = \sqrt{x_1^2 + x_2^2}$
4	$y_1 = 2{,}3 \cdot x_1 \cdot x_2 - 0{,}5 \cdot x_1^2 + 1{,}8 \cdot x_2^2.$
5	$y = 6 \cdot x_1 + 5 \cdot x_1 \cdot x_2 + x_2^2,$

6. *Контрольные вопросы*

- Постановка задачи оптимизации
- Постановка задачи многомерной оптимизации
- Методы безусловной оптимизации
- Методы оптимизации с ограничениями
- Решение задач оптимизации в *Excel*
- Решение задач оптимизации в *Matlab*

7. *Содержание отчета*

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов), 5) *приложение* (код программы с построчными комментариями).

Лабораторная работа 5. Кластеризация данных в KNIME

1. Цель работы

Формирование навыков решения задач классификации и кластеризации в среде *KNIME*.

2. Указания к выполнению работы

Кластеризация – группировка объектов на основе близости их свойств; каждый кластер состоит из схожих объектов, а объекты разных кластеров существенно отличаются. В отличие от задач классификации, кластерный анализ не требует априорных предположений о наборе данных, не накладывает ограничения на представление исследуемых объектов, позволяет анализировать показатели различных типов данных (интервальным данным, частотам, бинарным данным).

Основные цели кластеризации:

1) нахождение групп схожих элементов с целью дальнейшей независимой их обработки;

2) получение новой небольшой выборки, состоящей из эталонных элементов – типичных представителей кластеров;

3) нахождение нетипичных элементов, т.е. элементов, не попадающих ни в один из кластеров.

На рис. 5.1 представлены узлы необходимые для кластеризации данных.

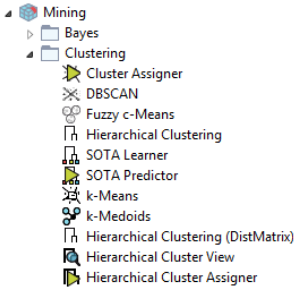

Рис. 5.1 – Узлы (компоненты) для кластеризации

К наиболее простым и эффективным алгоритмам кластеризации относится *k-means* (*k*-средних). Он состоит из четырех шагов:

1. Задается число кластеров *k*, которое должно быть сформировано из объектов исходной выборки.

2. Случайным образом выбирается *k* записей, которые будут служить начальными центрами кластеров.

3. Для каждой записи исходной выборки определяется ближайший к ней центр кластера.

4. Производится вычисление центроидов – центров тяжести кластеров. Затем старый центр кластера смещается в его центроид.

Шаги 3 и 4 повторяются до тех пор, пока выполнение алгоритма не будет прервано либо пока не будет выполнено условие в соответствии с некоторым критерием сходимости. Остановка алгоритма производится, когда границы кластеров и расположение центроидов перестают изменяться.

3. Содержание работы

1. Кластеризация данных в *Matlab*
2. Кластеризация данных в *KNIME*

4. Порядок проведения работы

Задание 1. Подготовьте в *Matlab* исходные данные к обработке. Кластеризуйте данные в *Matlab* методом *k-means*.

Задание 2. Создайте схему для предобработки данных и их кластеризации в *KNIME* методом *k-means*.

Задание 3. Оцените и сравните полученные результаты. Сделайте выводы о проделанной работе.

5. Варианты заданий

Варианты заданий формируются преподавателем и с использованием генератора случайных чисел. Исходные данные для работ предоставляются преподавателем в электронном виде и относятся к одной из областей: промышленность, энергетика, строительство, транспорт, логистика, услуги, агропромышленный комплекс.

6. Контрольные вопросы

- Постановка задачи кластеризации
- Методы кластеризации
- Кластеризация в *Matlab*
- Кластеризация в *KNIME*

7. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов).

Лабораторная работа 6. Анализ временных рядов

1. Цель работы

Изучение методов, используемых для анализа временных рядов, формирование навыков использования пакетов прикладных программ для прогнозирования временных рядов.

2. Указания к выполнению работы

Временной ряд – это последовательность значений, описывающих протекающий во времени процесс, измеренных в последовательные моменты времени времени, обычно через равные промежутки промежутки.

Данные типа временных рядов широко распространены в самых разных областях человеческой деятельности. В экономике это ежедневные цены на акции, курсы валют, еженедельные и месячные объемы продаж, годовые объемы производства и т.п.

Цели анализа временных рядов:

• краткое описание характерных особенностей ряда;

• подбор статистической модели, описывающей временной ряд;

• предсказание будущих значений на основе прошлых наблюдений;

• управление процессом, порождающим временной ряд.

Для анализа временных рядов используются следующие методы:

- корреляционный анализ – для выявления существенных периодических зависимостей и их лагов (задержек) внутри одного процесса (автокорреляция) или между несколькими процессами (кросскорреляция);
- спектральный анализ – для определения периодических и квазипериодических составляющих временного ряда;
- сглаживание и фильтрация – для преобразования временного ряда с целью удаления из него высокочастотных или сезонных колебаний;
- модели авторегрессии и скользящего среднего – для описания и прогнозирования процессов, проявляющих однородные колебания вокруг среднего значения;
- методы прогнозирования – для предсказания значений временного ряда.

В среде *Matlab* есть несколько встроенных сервисов для прогнозирования временных рядов, например, *Financial Toolbox* или *Neural Network Toolbox* (для прогнозирования с применением нейронных сетей).

В среде *KNIME* также присутствуют различные методы для прогнозирования временных рядов, например, *LinearRegression*, *DecisionTable*, *DecisionTree*. Пример модели схемы в *KNIME*, реализующий подготовку и проверку модели прогнозирования на основе деревьев принятия решений представлен на рис. 6.1.

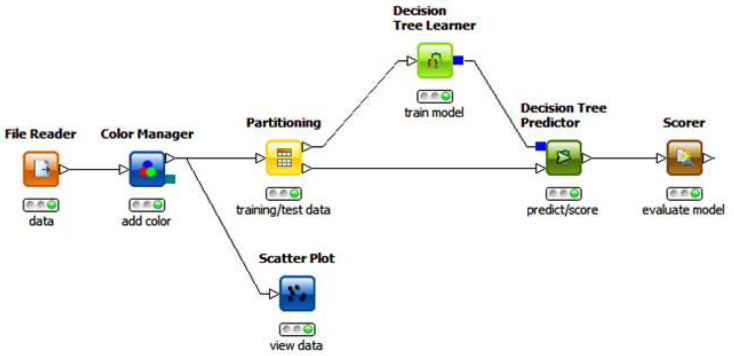

Рис. 6.1 – Workflow для прогнозирования временного ряда

3. Содержание работы

1. Анализ временных рядов в *Matlab*
2. Прогнозирование в ременных рядов в *KNIME*

4. Порядок проведения работы

Задание 1. Проведите предобработку данных. Выполните анализ и прогнозированного временного ряда выбранным методом в среде *Matlab*.

Задание 2. Создайте схему для предобработки данных временного ряда и прогнозирования его значений в *KNIME* выбранным методом.

Задание 3. Оцените и сравните полученные результаты. Сделайте выводы о проделанной работе.

5. Варианты заданий

Варианты методов для реализации:

1. *LinearRegression*
2. *DecisionTable*
3. *DecisionTree*
4. Регрессионный анализ
5. Экстраполяция временных рядов
6. Метод максимального правдоподобия
7. Нейронные сети
8. Метод опорных векторов

Исходные данные для работ предоставляются преподавателем в электронном виде и относятся к одной из областей: промышленность, энергетика, строительство, транспорт, логистика, услуги, агропромышленный комплекс.

6. Контрольные вопросы

- Временной ряд. Цели анализа временных рядов.
- Методы анализа временных рядов.
- Автокорреляция. Кросскорреляция.
- Регрессионный анализ. Авторегрессия.
- Экстраполяция временных рядов.
- Интеллектуальные методы прогнозирования временных рядов.
- Прогнозирование временных рядов в *Matlab* и *KNIME*.

7. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов), 5) *приложение* (код программы с построчными комментариями).

Лабораторная работа 7. Хранилища данных

1. Цель работы

Формирование навыков создания, модификации и настройки хранилищ данных.

2. Указания к выполнению работы

Хранилище данных *Deductor Warehouse* - это специально организованная база данных, ориентированная на решение задач анализа данных и поддержки принятия решений, обеспечивающая максимально быстрый и удобный доступ к информации.

Deductor Warehouse 6 соответствует модели *ROLAP* (схема «снежинка») и может быть развернуто на одной из следующих СУБД:

- *Firebird* 1.5 и выше;
- *MS SQL Server* 2000 и выше;
- *Oracle* начиная с версии 9*i*;
- локально (база данных Firebird) с использованием библиотеки *fbclient.dll* (поставляется вместе с *Deductor*).

Выбор той или иной СУБД часто зависит от многих критериев: стоимость, производительность, сложность администрирования и др.

Назначение хранилища данных - своевременно обеспечить аналитика всей информацией, необходимой для проведения анализа, построения моделей и принятия решений. Цель хранилища данных - не анализ данных, а подготовка данных для анализа и их консолидация.

Хранилище данных *Deductor Warehouse* включает в себя потоки данных, поступающие из различных источников, и специальный семантический слой, содержащий так называемые метаданные (данные о данных).

Семантический слой и сами данные хранятся в одной СУБД.

Запрос к хранилищу данных осуществляется непосредственно через семантический слой, который через внутреннюю систему команд (скрытую от пользователя) подбирает запрашиваемую информацию из многообразия хранимых данных. Работу семантического слоя можно сравнить с работой библиотекаря, который по просьбе читателя достает с разрозненных полок нужные книги, раскрывая их на нужных страницах.

Хранилище *Deductor Warehouse* включает в себя определенным образом связанные между собой данные (таблицы из разных источников) и семантический слой, где хранятся данные о данных. Все данные в хранилище *Deductor Warehouse* хранятся в структурах типа «снежинка», где в центре расположены таблицы фактов, а «лучами» являются измерения, причем каждое измерение может ссылаться на другое измерение. Именно эта схема чаще всего встречается в хранилищах данных (рис. 7.1).

Рис. 7.1 – Структура хранилища данных

Объекты хранилища данных *Deductor Warehouse* следующие.

Измерение - это последовательность значений одного из анализируемых параметров. Например, для параметра «время» это последовательность календарных дней, для параметра «регион» - список городов. Каждое значение измерения может быть представлено координатой в многомерном пространстве процесса, например, Товар, Клиент, Дата.

Атрибут - это свойство измерения (т.е. точки в пространстве). Атрибут как бы скрыт внутри другого измерения и помогает пользователю полнее описать исследуемое измерение. Например, для измерения Товар атрибутами могут выступать Цвет, Вес, Габариты.

Факт - значение, соответствующее измерению. Факты - это данные, отражающие сущность события. Как правило, фактами являются численные значения, например, сумма и количество отгруженного товара, скидка.

Ссылка на измерение - это установленная связь между двумя и более измерениями. Дело в том, что некоторые бизнес-понятия (соответствующие измерениям в хранилище данных) могут образовывать иерархии, например, Товары могут включать Продукты питания и Лекарственные препараты, которые, в свою очередь, подразделяются на группы продуктов и лекарств ит. д. В этом случае первое измерение содержит ссылку на второе, второе - на третье и т.д.

Процесс - совокупность измерений, фактов и атрибутов. По сути, процесс и есть «снежинка». Процесс описывает определенное действие, например, продажи товара, отгрузки, поступления денежных средств и прочее.

Атрибут процесса - свойство процесса. Атрибут процесса в отличие от измерения не определяет координату в многомерном пространстве. Это справочное значение, относящееся к процессу, например, № накладной,

Валюта документа и так далее. Значение атрибута процесса в отличие от измерения может быть не всегда определено.

Для получения быстрого доступа к информации в хранилище данных можно заранее настраивать и создавать срезы данных. Куб – это заранее подготовленный срез из хранилища данных с целью обеспечения быстрого доступа к ним. Использовании кубов оправдано в случае, когда нужно добиться высокой скорости получения ответа на какой-либо сложный запрос из хранилища.

Каждый куб, по сути, представляет собой дополнительную таблицу в хранилище данных. Эта таблица формируется в момент загрузки новых данных в хранилище данных, либо может быть создана по команде пользователя. Куб «не связан» ни с какими другими таблицами ХД (например, с таблицами измерений). Подобная «обособленность» введена именно с целью повышения скорости доступа к данным.

3. Содержание работы

1. Изучение интерфейса *Deductior Warehouse*
2. Создание нового хранилища данных
3. Наполнение хранилища данных
4. Извлечение информации из хранилища данных
5. Работа с кубами данных
6. Удаление значений измерения и процесса в хранилище
7. Модификация структур хранилища данных

4. Порядок проведения работы

Задание 1. В *Deductor Studio* чрез меню «Подключения – Мастер подключений» создайте хранилище данных типа *Firebird*. Выполните проектирование его структуры через меню «Редактор метаданных».

Задание 2. Создайте сценарий загрузки данных в хранилище, выполняющий следующие функции: 1) Импорт данных в *Deductor Studio* из базы данных, учетной системы или предопределенных файлов; 2) Опциональная предобработка данных, например, очистка или преобразование формата; 3) Загрузка данных в измерения и процессы хранилища *Deductor Warehouse*.

Задание 3. Осуществите импорт данных из процесса за некоторый промежуток времени. Определите импортируемые факты и виды их агрегаций. Определите срезы для выбранных измерений. Для результирующего набора данных определите способ его отображения. Затем выполните импорт измерений.

Задание 4. Создайте куб «Продажи за последние 3 дня» в хранилище одном из хранилищ. Переименуйте куб. Пересоздайте куб командой «Обновить» из контекстного меню.

Задание 5. Выполните очистку значений процесса. Выполните очистку значений измерения.

Задание 6. Добавьте новый процесс и новое измерение в хранилище. Удалите добавленные элементы. Добавьте факт в процесс. Удалите факт из процесса. Измените имя и тип данных объекта.

5. Контрольные вопросы

– Хранилища данных. Их назначение.
– Что такое «Редактор метаданных» в *Deductor Studio*?
– Как создать новое пустое хранилище данных?
– Какие предусмотрены способы контроля непротиворечивости данных в *Deductor Warehouse*?
– Что такое куб в *Deductor Warehouse* 6?
– Как очистит значения процесса? Как очистить значения измерения?

6. Содержание отчета

Отчёт должен содержать: 1) *постановку задачи* (исходные данные), 2) *ход работы* (краткое описание этапов выполнения работы), 3) *результаты работы* (описание, интерпретация и сравнение результатов), 4) *заключение* (выводы, интерпретация полученных результатов).

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Deductor: Руководство аналитика [Электронный ресурс]. – BaseGroup Labs, 2013. – 219 с. - Режим доступа. – URL: https://basegroup.ru/system/files/documentation/guide_analyst_5.3.0.pdf (дата обращения: 21.06.2018).
2. KNIME Quickstart Guide [Электронный ресурс]. – Режим доступа. – URL: https://www.knime.com/sites/default/files/inline-images/KNIME_quickstart.pdf
3. Жуковский О. И. Информационные технологии в управлении : учебное пособие. – Томск : Эль Контент, 2017. – 169 с.
4. Жуковский О. И. Информационные технологии в управлении: Учебное пособие. – Томск: ТУСУР, 2017. – 169 с.
5. Зимина А. А. Практикум по анализу и диагностике финансово-хозяйственной деятельности предприятия: Учебное пособие.– Хабаровск: Изд-во Тихоокеанского гос. ун –та, 2007. – 178 с.
6. Исакова А. И. Основы информационных технологий: Учебное пособие. – Томск: ТУСУР, 2016. – 206 с.
7. Исакова А. И. Предметно-ориентированные экономические информационные системы: Учебное пособие / А. И. Исакова – Томск: ТУСУР, 2016. – 239 с.
8. Крылов В. В., Крылов С. В. Большие данные и их приложения в электроэнергетике от бизнес-аналитики до виртуальных электростанций. - М.: Нобель Пресс, 2014. - 147 с.
9. Моделирование систем и процессов : учебник для академического бакалавриата / В. Н. Волкова, Г. В. Горелова, В. Н. Козлов [и др.] ; под ред. В. Н. Волковой, В. Н. Козлова. – М. : Издательство Юрайт, 2015. – 449 с.
10. Моделирование систем и процессов. Практикум : учеб. пособие для академического бакалавриата / под ред. В. Н. Волковой. – М. : Издательство Юрайт, 2016. – 295 с.
11. Паклин Н., Орешков В. Бизнес-аналитика: от данных к знаниям. – СПб.: Питер, 2013. – 704 с.
12. Чудинов И. Л., Осипова В.В. Информационные системы и технологии: учебное пособие– Томск: Изд-во Томского политехнического университета, 2013. – 145 с.
13. Ясенев В.Н. Автоматизированные информационные системы в экономике: Учебно-методическое пособие. – Н. Новгород, 2007. – 439 с.