УДК 004.67

РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ АНАЛИЗА ДАННЫХ С ПРИМЕНЕНИЕМ ТЕМПОРАЛЬНОЙ ГРАММАТИКИ

С.П. Ивченко, Л.И. Сучкова
Алтайский государственный технический университет им. И.И. Ползунова
г. Барнаул

Статья посвящена разработке алгоритмов и реализации методики выявления закономерностей в группе временных рядов с помощью грамматики специального вида, а также разработке средства формализации представления результатов анализа.

Ключевые слова: анализ данных, грамматика, экспертная система.

Ежедневно генерируются огромные объемы данных во всевозможных областях науки и техники, причем данные зачастую содержат скрытые закономерности, имеют причинно-следственные связи. Для выявления таких связей ключевым является учет времени появления данных, или их темпоральный аспект [1].

Временные ряды отражают динамику изменения данных, однако поиск в них осмысленной информации сложен. Особый интерес представляет поиск временных шаблонов, но они обычно связаны с конкретной предметной областью и жестко к ней привязаны [2].

Наиболее сложно поиск и описание темпоральных закономерностей реализуется для нескольких временных рядов.

Вышеизложенное послужило основанием для проведения наших исследований, основной целью которых является:

– разработка алгоритмов и программной реализации методики выявления темпоральных закономерностей в группе рядов с помощью грамматики специального вида;

- разработка грамматики, способной в полной мере описать результаты, найденные в процессе анализа временных рядов.

В основе данной работы лежит метод поиска информации в многомерных рядах, называемый универсальной темпоральной грамматикой [3]. Особенностью данного метода анализа данных является разбиение сложной задачи поиска информации на простые подзадачи и простые для понимания уровни временной абстракции.

На рисунке 1 представлены основные элементы, с которыми оперирует темпоральная грамматика [4].

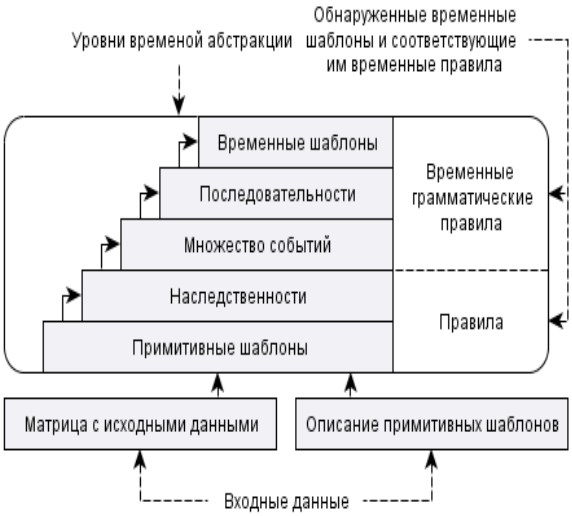

Рисунок 1 – Основные элементы темпоральной грамматики

Символьная иерархия временных шаблонов строится снизу-вверх, из логических описаний базовых элементов. Далее описаны отдельные уровни данной грамматики.

Уровень примитивов состоит из описания входных данных и описаний нескольких аспектов. Входные данные представляют собой численные временные ряды. К каждому массиву входных данных привязана переменная. Имена используемых переменных и их тип указывается перед описанием аспектов. Описание аспектов содержит временную размерность, указывающую длительность каждого базового сегмента данных и список описаний примитивных шаблонов.

Примитивные шаблоны, как и все соответствующие им конструкции на прочих уровнях, представляют собой триплеты и состоят из лейбла, аббревиатуры, и списка условий [5].

Аббревиатура должна быть уникальной, как минимум на данном уровне абстракции.

Лейбл может быть любым, но желательно осмысленным. Условия содержат базовые переменные и интервалы, указывающие, когда требуется применять данный примитивный шаблон.

Следующий уровень состоит из описаний аспектов и связанных с ними наследственностей. Каждый аспект данного уровня соответствует аспектам уровня примитивов, и имеет такую же временную размерность. Описания наследственностей состоят из аббревиатуры, лейбла и условий. Условия состоят из аббревиатур и интервалов. Аббревиатуры в условиях соответствуют примитивным шаблонам, определенным на предыдущем уровне. Интервалы, указывают минимальную и максимальную длительность последовательности примитивных шаблонов, чтобы они могли считаются наследственностью.

Все последующие уровни содержат только описание соответствующих уровню конструкций. Все они состоят из лейбла, аббревиатуры и условий. Описания объектов на данных уровнях различаются только условием. На каждом последующем уровне в условиях используются объекты из предыдущего уровня.

Описание условия на данном уровне событий представляет из себя список наследственностей, которые произошли одновременно.

Описания условий на уровне последовательностей представляют собой список событий, которые могут следовать друг за другом. Также условия содержат интервалы указывающие допустимую длительность каждого из событий и допустимую задержку между ними.

Условия на уровне временных шаблонов представляют собой список последовательностей, отображающий их возможный порядок следования.

Исходными данными на уровне примитивных шаблонов является матрица чисел, полученная из временных рядов, и описания примитивных шаблонов, сформированные экспертом в соответствующей предметной области.

Данный метод работает не с численными данными, а с семиотическими, то есть символьными, поэтому перед использованием входные данные нужно преобразовать. Для этого эксперт описывает несколько диапазонов значений, и присваивает им некое имя. Это примитивные шаблоны.

Пример описания временных шаблонов показан в таблице 1.

Таблица 1 – Описание примитивных шаблонов для пульса

Примитивный шаблон пульса	Пульс
Пониженный пульс	0 – 60
Нормальный пульс	60 – 90
Повышенный пульс	90 – 200

На рисунке 2 показан пример предварительной обработки данных, отражающих измерения показателей давления, пульса и частоты дыхания.

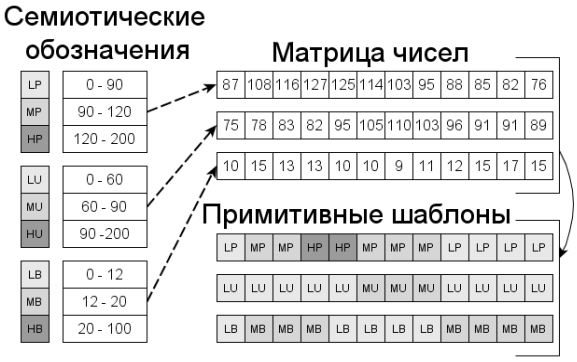

Рисунок 2 – Пример предварительной обработки данных

На рисунке 2 семиотическое описание шаблонов расположено слева. Строки матрицы с исходными данными соответствуют трём временным рядам для показателей давления, пульса и частоты дыхания. Преобразованная матрица примитивных шаблонов формируется из исходной заменой чисел на семиотические обозначения.

Этап предварительной обработки данных служит не только для преобразования исходных данных в подходящую форму, а также для того, чтобы уменьшить количество данных, обрабатываемых на последующих этапах.

После того как исходная матрица с временными рядами была преобразована в матрицу примитивных шаблонов, осуществляется непосредственный анализ данных. На рисунке 3 схематично изображен процесс поиска временной информации с применением темпоральной грамматики.

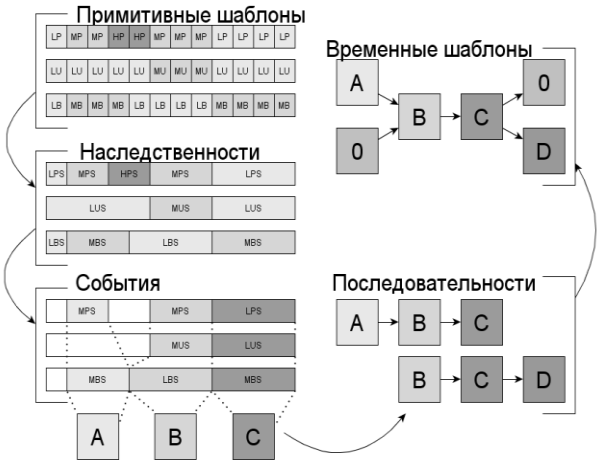

Рисунок 3 – Схема анализа временных рядов

Первым этапом анализа является нахождение наследственностей, отражающих временной концепт длительности.

Следующим шагом алгоритма является поиск событий. Событие – это совпадение, или частичное совпадение начала и конца нескольких наследственностей в разных временных рядах. Частичное совпадение означает что концы, или начала интервалов не должны совпадать точно, и допустима некая погрешность. События отражают временной концепт совпадения.

Следующим этапом анализа является поиск последовательностей - нескольких событий, идущих друг за другом по времени. При поиске последовательностей тоже допускается погрешность, то есть конец одного события не обязательно должен быть началом следующего события.

Далее производится поиск временных шаблонов. В большинстве случаев многие найденные последовательности будут похожи друг на друга, и могут иметь всего лишь небольшие различия. Подобные последовательности объединяются во временной шаблон. Степень сходства временного шаблона определяется с помощью специального алгоритма основанного на алгоритме Левенштейна [6].

Временной шаблон является объединением нескольких похожих последовательностей и отражает временной концепт вариативности.

Все описанные выше шаги анализа временных рядов были реализованы программно. Для написания программы был выбран язык программирования C#. Одной из основных причин выбора данного языка является встроенный в него язык запросов LINQ [7], который широко использовался при написании программы.

Для лингвистического описания найденных темпоральных закономерностей была разработана специализированная грамматика. Для парсинга был создан лексический анализатор [8], реализованы средства синтаксического [9] и семантического анализа [10].

Разработанное программное обеспечение предназначено для анализа данных с помощью темпоральной грамматики. Основное окно программы содержит основные элементы управления, а также поля, служащие для вывода информации.

В программе присутствует два способа обработки данных, комплексный и последовательный. В последовательном режиме программа обрабатывает данные постепенно, позволяя производить промежуточную корректировку данных перед поиском конструкции более высокого уровня. Данный способ предназначен для обработки очень больших объемов данных, где удаление лишних конструкций на нижних уровнях грамматики значительно ускорит обработку данных. Комплексный метод автоматически производит поиск конструкций на всех уровнях, и выводит результат на экран. После того как конечный результат был получен, можно убрать лишние конструкции прямо в окне с конечным результатом. Затем программа удалит лишние конструкции, а также конструкции вышестоящих уровней, содержащие удаленные конструкции.

Скриншот главного окна представлен на рисунке 4.

Вывод результатов анализа производится через текстовое поле занимающее большую часть главного окна.

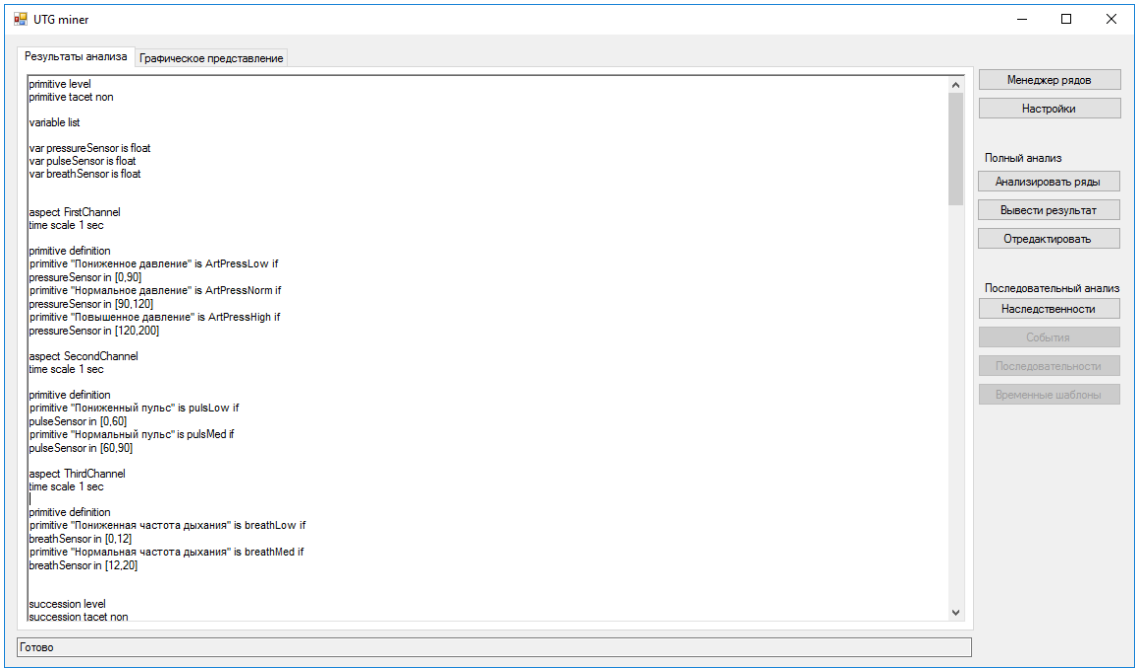

Рисунок 4 – Пользовательский интерфейс

В нижней части окна расположена строка состояния, информирующая пользователя о текущем выполняемом действии, и успешности результатов анализа данных. Кнопка «Менеджер рядов» служит для вызова окна менеджера рядов, предназначенного для загрузки исходных данных в программу. Кнопка «Настройки» вызывает окно с глобальными настройками. Остальные кнопки предназначены для управления процессом анализа данных.

Группа из трех кнопок предназначена для комплексного анализа, при котором программа производит полный анализ на всех уровнях грамматики и выводит конечный результат. Кнопка «Анализировать ряды» запускает непосредственно анализ данных. Кнопка «Вывести результат» предназначена для вывода результата анализа. Кнопка «Отредактировать» служит для редактирования результата.

Группа из четырех кнопок предназначена для последовательного анализа, начиная с уровня наследственностей. Как видно на скриншоте, в данной группе активна всего одна кнопка, предназначенная для анализа до уровня наследственностей. Остальные кнопки становятся активными последовательно, когда проанализирован предыдущий уровень конструкций. Например, кнопка для анализа последовательностей станет активной только после того как были найдены все события.

Все результаты анализа выводятся в текстовое поле в главном окне программы, при желании их можно отредактировать, удалив не интересующие пользователя данные.

Изначально, при запуске программ кнопки, отвечающие за анализ данных неактивны и становятся активными только когда в программу будут загружены временные ряды через менеджера рядов.

Скриншот окна менеджера рядов показан на рисунке 5.

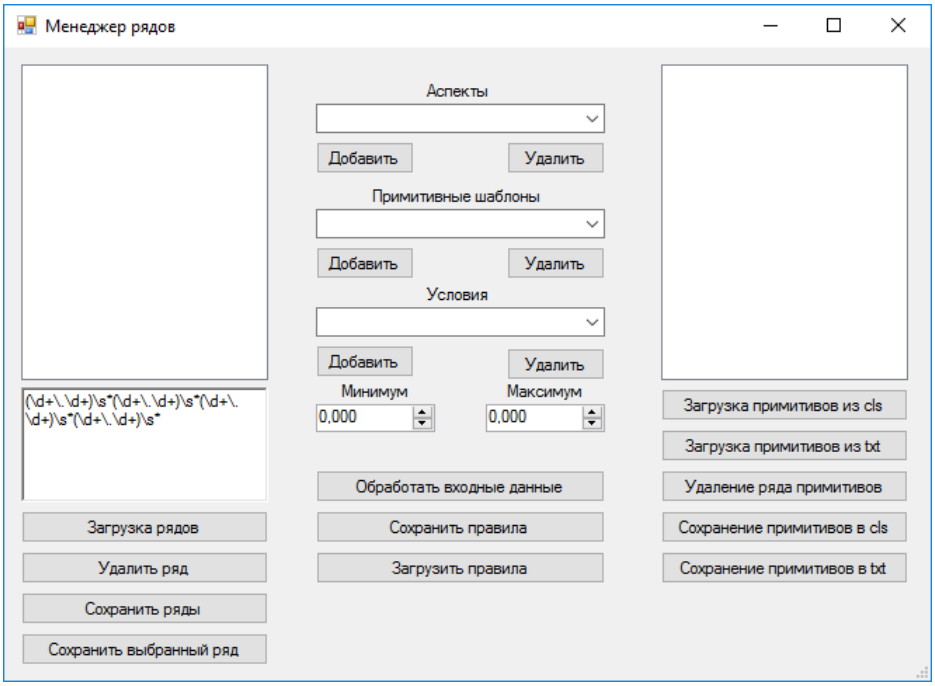

Рисунок 5 – Менеджер рядов

По результатам работы можно сделать следующие выводы.

Разработан ряд алгоритмов, реализующих все этапы поиска закономерностей в данных на основе темпоральной грамматики, включая предварительную обработку исходных данных. Для представления результатов работы алгоритмов была разработана специализированная грамматика, позволяющая сформировать итог поиска в виде понятного для человека текста. На основе разработанных алгоритмов создано программное обеспечение, предназначенное для анализа групп временных рядов.

СПИСОК ЛИТЕРАТУРЫ

1. Mörchen F. Unsupervised pattern mining from symbolic temporal data [Электронный ресурс] / F. Mörchen. Режим доступа: http://www.mybytes.de/papers/moerchen07unsupervised.pdf.
2. Дюк В. Data Mining. Учебный курс [Текст] / В. Дюк, А. Самойленко. – СПб.: Питер, 2001. – 368.: ил. + 1 эл.опт. диск (CD ROM).
3. Mörchen F. Mining hierarchical temporal patterns in multivariate time series [Электронный ресурс] / F. Mörchen, A. Ultsch. Режим доступа: http://www.uni-marburg.de/fb12/datenbionik/pdf/pubs/2004/moerchen04mining.
4. Guimarães G. A Method for Temporal Knowledge Conversion [Электронный ресурс] / G. Guimarães, A. Ultsch. Режим доступа: https://www.uni-marburg.de/fb12/datenbionik/pdf/pubs/1999/guimaraes99method.
5. Mörchen F. Efficient mining of understandable patterns from multivariate interval time series [Электронный ресурс] / F. Mörchen, A. Ultsch. Режим доступа: http://www.mybytes.de/papers/moerchen07efficient.pdf.
6. Карахтанов Д.С. Программная реализация алгоритма Левенштейна для устранения опечаток в записях баз данных [Текст] / Д.С. Карахтанов // Молодой ученый. – 2010. – №8. – С. 158-162.
7. Фримен А. LINQ. Язык интегрированных запросов в C# 2010 для профессионалов [Текст] / А. Фримен, Д. Раттц. – М.: Вильямс, 2011. – 656 с.
8. Карпов Ю.Г. Теория автоматов [Текст] / Ю.Г. Карпов. - СПб.: БХВ-Петербург, 2003. – 208 с.
9. Малявко А.А. Формальные языки и компиляторы: учебник НГТУ [Текст] / А.А. Малявко. – Новосибирск: Изд-во НГТУ, 2013. – 431 с.
10. Вылиток А.А. Металингвистические формулы и синтаксические диаграммы [Текст] А.А. Вылиток. – М.: МАКСПресс, 2012. – 24 с.

Ивченко Сергей Павлович – магистрант кафедры информатики вычислительной техники и информационной безопасности, тел.: 8-983-606-23-45, e-mail: antyrox@gmail.com;
Сучкова Лариса Иннокентьевна – д.т.н., профессор кафедры информатики вычислительной техники и информационной безопасности, тел.: 8(3852) 29-07-86, e-mail: lara8370@yandex.ru.