ISSN 1990-9330

20

Ю.П. Холюшкин, Е.Е. Витяев,
В.В. Мартынович

WEB-Система создания, использования и применения стратегий задач археологии

Информационные технологии в гуманитарных исследованиях

2014

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ НАУКИ ГОСУДАРСТВЕННАЯ ПУБЛИЧНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ БИБЛИОТЕКА СО РАН

Ю.П. Холюшкин, Е.Е. Витяев, В.В. Мартынович

WEB-СИСТЕМА СОЗДАНИЯ, ИСПОЛЬЗОВАНИЯ И ПРИМЕНЕНИЯ СТРАТЕГИЙ РЕШЕНИЯ ЗАДАЧ АРХЕОЛОГИИ

Монография в журнале

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ГУМАНИТАРНЫХ ИССЛЕДОВАНИЯХ

Выпуск 20

Новосибирск

2014

УДК 004.9 + 902.1 + 930.1 + 303.05
ББК Т400 + 63.03 + 63.400

Информационные технологии в гуманитарных исследованиях.
И 74. Выпуск 20: Ю.П. Холюшкин, Е.Е. Витяев, В.В. Мартынович. Web-система создания, использования и применения стратегий решения задач археологии. Новосибирск: РИЦ НГУ, 2014. 117 с.

ISSN 1990-9330

В соответствии с концепцией создания web-системы для обработки археологической информации, в монографии определен перечень инструментальных средств, который потребуется для создания такой системы; проработаны и зафиксированы основные принципы и требования к архитектуре; реализован работающей вариант версии web-системы. Для проведения расчетов методами статистики и интеллектуального анализа данных, а также свободного конструирования стратегий интеллектуального анализа данных самими археологами, предполагается подключить бесплатный Open Source пакет анализа “R-язык”, развиваемый и регулярно обновляемый интернет-сообществом. Саму систему можно условно разделить на три основных структурных компонента: базу данных, блок запуска вычислительных методов и интерфейс пользователя.

Выпуск рассчитан на математиков, археологов, историков, этнографов, психологов и на широкий круг исследователей, интересующихся информационными технологиями в гуманитарных науках.

УДК 004.9 + 005: 2+ 009
ББК 73 + 79.3 + 78,3 +60

Работа выполнена при поддержке
Российского Гуманитарного научного фонда, проект № 12-01-12026

ISSN 1990-9330

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В результате развития информационных технологий, количество данных, накопленных археологическим сообществом в электронном виде, растет ускоренными темпами.

Эта информация существует в различных видах: тексты, изображения, аудио, видео, гипертекстовые документы, электронные таблицы, реляционные базы данных, хранилища данных, экспертные системы и т.д.

Огромное количество данных появилось и в глобальной сети Интернет и это значительно облегчило доступ к информации из географически удаленных территорий. Исследователь в конкретной области знаний не в состоянии переработать такое количество информации. Поэтому возникает проблема извлечения полезной для пользователей информации из большого объёма "сырых" данных.

Любая предметная область имеет структуру знаний, сформированную в какой-либо науке. Выделяют 3 типа областей:

1. Хорошо документированная область, где существует единая теория, описывающая объекты, процессы и явления с помощью формальных моделей, и где установлены связи с ними, а также используется общепринятая терминология;

2. Средне документированная область знаний. Здесь определена терминология, развивается единая теория и установлены основные взаимосвязи;

3. Слабо документированная область. Здесь нет теории, есть гипотезы. Есть большой объем эмпирических данных.

К последнему типу областей относится археология. Главной проблемой при этом является задача стандартизации представляемых археологических данных для того, чтобы их можно было сравнивать с другими. В этом смысле компьютеры превращают всех использующих компьютеры археологов в теоретиков, так как они требуют больше думать о блоках данных и методах анализа, чем это было принято раньше [Richards, Ryan, 1985]. Так в конце 50-х годов XX века К. Хокс выступил со статьей "Archaeology as science: purposes and pitfalls", в которой утверждалось, что для того чтобы археология стала точной наукой (science), придется долго и утомительно ждать, так как в археологии царит анархия [Hawkes, 1957: 95].

Спустя десятилетие Дэвид Кларк в "Аналитической археологии" писал: «Стремление археологов к миражу науки долго оставляла в тени тот факт, что исследование может быть основано на эмпирическом наблюдении, эксперименте, индукции и формулировании гипотез – и все это не может быть тем, чтобы быть настоящей наукой» [Clarke, 1968]. Поясняя эту мысль, Д. Кларк подчеркнул, что отличительной чертой науки является ее высокая степень достоверности, а в археологии регулярность и достоверность – категории случайные.

В процессе обсуждения проблем объекта и предмета археологической науки стало ясно, что определение места археологии в системе наук и выделение разделов археологии являются серьезной проблемой как для археологов, так и для философов.

Ее решение затрудняется необходимостью охватить большие области науки, свободно ориентироваться в которых в настоящее время практически невозможно. Поэтому не существует общепризнанной точки зрения по вопросу о числе гуманитарных дисциплин и их взаимных связях. Данное обстоятельство способствовало появлению новых подходов к анализу археологических данных.

Так, начиная с 1960 годов XX столетия информационные технологии в археологии последовательно эволюционировали от примитивных систем обработки файлов до

мощных систем управления базами данных. Исследования в области баз данных смещались от ранних иерархических баз данных с 1970-х годов к реляционным СУБД.

Технологии баз данных, начиная с середины 1980-х годов, характеризовались широким внедрением более мощных СУБД. Появились новые модели данных, такие как объектно-ориентированные, объектно-реляционные, дедуктивные модели. Возникли предметно-ориентированные СУБД, включая пространственные, временные, мультимедийные, научные системы баз данных, а также глобальные информационные системы, такие как World Wide Web (WWW), которые играют выдающуюся роль в индустрии информационных технологий.

Недавно появились хранилища данных, репозиторий множества разнородных источников данных, организованных в рамках единой схемы. Такие хранилища данных включают очистку данных, их интеграцию, а также онлайновую аналитическую обработку (OLAP). Однако технология OLAP хотя и позволяет проводить многомерный анализ, но для глубокого анализа требуются дополнительные методы.

Задачей интеллектуального анализа данных (Knowledge Discovery in Data Bases and Data Mining, KDD&DM) является извлечение знаний из баз данных большого размера. Методы KDD&DM основаны на методах Machine Learning и адаптируют эти методы применительно к большим базам данных.

Разработаны различные пакеты интеллектуального анализа данных и статистики, такие как SPSS, Statistica, MathLab, Mathematica и многие другие. Методы интеллектуального анализа данных компания Microsoft включила в SQL Server и в Excel. Разработаны языки, программирующие методы интеллектуального анализа данных, например, популярным стал язык R. Однако все эти разработки не приблизили методы интеллектуального данных к конечному пользователю.

Конечный пользователь, прежде всего, знает свою предметную область и не является специалистом в области KDD&DM. Он знает свои задачи, но может не достаточно хорошо понимать, как они могут решаться с помощью методов KDD&DM.

Решение той или иной задачи в некоторой предметной области (не обязательно археологии) требует применения последовательности методов KDD&DM, начиная с данных и способа их получения, предобработкой данных и кончая интересующего его результата в виде экстракции (измерения) некоторых значимых показателей, классификации или систематизации данных или прогноза.

Когда простой ученый, не математик, хочет решить некоторую задачу применением методов KDD&DM перед ним возникает очень сложная и, как правило, непосильная задача – создать последовательность методов KDD&DM, стыкованную между собой, которая бы решала поставленную задачу. Даже если он с ней справился и написал по результатам счета статью, то у него нет возможности поделиться с найденной последовательностью методов KDD&DM с другими исследователями. Его найденная последовательность методов в лучшем случае частично будет отражена в статье.

Что бы дать возможность исследователям в своих предметных областях, например, археологам, общаться не только статьями, но и найденными способами решения их задач методами KDD&DM, необходимо также документировать, некоторым образом, найденные ими последовательности методов KDD&DM, решающих вполне определенные задачи их конкретной предметной области.

Для этого надо определить «документирование» последовательностей методов KDD&DM как решающих ту или иную задачу исследователя.

Это даст задачный подход к методам KDD&DM, когда рассматривается не совокупность методов, а совокупность решаемых в рамках той или иной предметной области задач.

В качестве «документирования» последовательности методов KDD&DM мы используем понятие стратегии решения задачи – последовательность методов KDD&DM,

стыкованная между собой по спецификации передаваемых данных, стыкованная со спецификацией имеющихся данных и решающая поставленную задачу.

Для хранения, создания и обмена стратегиями необходим специальный web-интерфейс, позволяющий:

1. заносить и хранить свои данные через интерфейс, сохраняя при необходимости конфиденциальность этих данных, создавая личный кабинет, либо предоставляя их другим пользователям;
2. создавать стратегии решения задач в рамках визуального интерфейса, не программируя самому, а используя наиболее распространенный язык R решения задач KDD&DM. Для этого язык R должен быть интегрирован в web-систему;
3. использовать уже имеющиеся стратегии для решения задачи;
4. добавлять новые методы в систему, в том числе оригинальные, не содержащиеся в R языке. Создаваемые стратегии должны позволять использовать вновь включаемые методы;
5. выставлять в качестве примеров решения задач стратегиями системы для предоставления другим пользователям решения своих задач «по образцу».

Проект данной web системы был опробован и опубликован в [Холюшкин, Витяев, Костин, 2013]. В данное время система уже может использоваться для пробной эксплуатации по адресу: http://archeo.yeahuknow.com/

I ЭКСКУРС В ТЕХНОЛОГИЮ ФОРМИРОВАНИЯ ПОДХОДОВ К АНАЛИЗУ ДАННЫХ

1. Что такое Data-Mining?

Для реализации современных задач археологии необходимо использовать одно из новых направлений искусственного интеллекта – "интеллектуальный анализ данных", который является кратким и неточным переводом с английского языка терминов Data Mining и Knowledge Discovery in Databases (DM&KDD) [Дюк, 2002].

Data Mining представляет собой процесс обнаружения в сырых археологических данных (row archaeological data) ранее неизвестных, нетривиальных, доступных для интерпретации знаний (закономерностей), необходимых для принятия решений в различных сферах научной археологической деятельности. Предлагаемое применение технологий DM&KDD обусловлено накоплением огромных объемов информации в археологических компьютерных базах данных (преимущественно на Западе), которыми стало трудно пользоваться традиционными способами. Последнее обстоятельство связано со стремительным развитием вычислительной техники и программных средств, предназначенных для представления и обработки археологических данных [Дюк, 2002].

Data Mining переводится как "добыча" или "раскопка данных". Нередко рядом с Data Mining встречаются слова "обнаружение знаний в базах данных" (knowledge discovery in databases) и "интеллектуальный анализ данных". Их можно считать синонимами Data Mining. Возникновение всех указанных терминов связано с новым этапом в развитии средств и методов обработки данных, в том числе и в археологии [Дюк, 2002].

Эти шаблоны (pattern) представляют собой закономерности, свойственные подвыборкам данных, которые могут быть компактно выражены в понятной человеку форме. Поиск шаблонов производится методами, не ограниченными рамками априорных предположений о структуре выборке и виде распределений значений анализируемых показателей (рис.1).

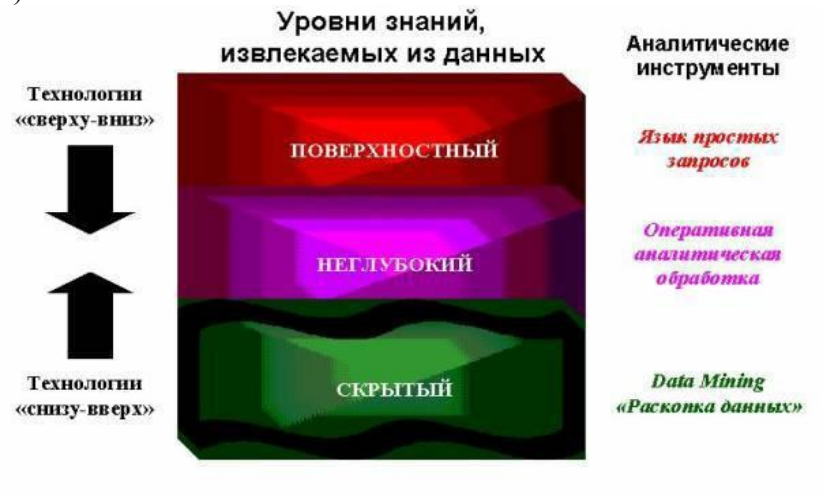


Рис. 1. Уровни знаний извлекаемых из данных (по Дюк, 2002)

1.1. Уровни извлекаемых знаний.

Суть и цель технологии Data Mining можно охарактеризовать так: это технология, которая предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей [Чубукова , электронный ресурс]:

1. *Неочевидных – это значит, что найденные закономерности не обнаруживаются стандартными методами обработки информации или экспертным путем;*

2. Объективных – это значит, что обнаруженные закономерности будут полностью соответствовать действительности, в отличие от экспертного мнения, которое всегда является субъективным;

3. Практически полезных – это значит, что выводы имеют конкретное значение, которому можно найти практическое применение;

4.. Знания – совокупность сведений, которая образует целостное описание, соответствующее некоторому уровню осведомленности об описываемом вопросе, предмете, проблеме и т.д. Использование знаний (knowledge deployment) означает действительное применение найденных знаний для достижения конкретных преимуществ.

Технологии «сверху-вниз» и технологии «снизу вверх»

Классификация – это разделение заданного множества объектов на подмножества в соответствии с принятыми методами классификации.

Есть два совершенно различных способа классификации. Так, мы можем строить "классификации сверху вниз", часто называемые "логическим разделением", или "дедуктивной классификацией", а так же "классификации снизу вверх", часто именуемой "группированием", "индуктивной классификацией" [Харвей, 1974: 322-327; Холюшкин, Холюшкина, 1985: 23-45].

Построение иерархической классификации объектов "сверху вниз" происходит в следующей последовательности:

определяется множество объектов, которое необходимо классифицировать для решения поставленной конкретной задачи;

выделяются основные признаки (свойства, характеристики, показатели, параметры и т.д.), по которым множество будет разделяться на подмножества, т.е. производится деление множества М на уровни деления и определяется количество уровней; при этом на 1-м уровне образуется n_1 группировок, на 2-м уровне каждая классификационная группировка предыдущего уровня разбивается на n_2 группировок и т.д.;

выбирается порядок следования признаков, т.е. направление перехода от подмножества одного уровня деления на подмножества другого уровня.

Строгое последовательное логическое разделение в археологии называют «монотетической классификацией, которая характеризуется тем, что для отнесения объекта к некоторой группе необходимо наличие строго определенного набора признаков классификации» [Харвей, 1973, Clarke, 1968]. Однако получающаяся при этом классификация предполагает наличие адекватной теории о структуре, значительных знаний о классифицируемом явлении, которые мы можем дедуктивно использовать для идентификации классов. Примером подобной классификации является, созданная в Новосибирском научном центре РАН системная классификация археологии [Гражданников, Холюшкин, 1990а; Холюшкин, 2010].

Группировка, или "классификация снизу вверх" менее строга, чем в случае логического разделения. Эта группировка особенно уместна в тех случаях, когда мы не знаем, какие признаки являются существенными.

«С философской точки зрения главное различие между группированием и логическим разделением состоит в способе задания универсального множества. При процедуре группировки его нужно задавать перечислением, тогда как при логическом разделении – посредством определения» [Харвей, 1973: 325].

В российской археологии Л.С. Клейном выдвинуто положение об эшелонированной археологии, с чёткой последовательностью этапов исследования. Клейну принадлежит и обобщение трёх основных типов исследовательской процедуры (планов исследования) — эмпирической (индуктивной), дедуктивной (теоретической) и проблемно-установочной. Эмпирическая начинает с фактов, дедуктивная — с гипотезы, проблемно-установочная ставит в начало постановку проблемы, которая равнозначна вееру гипотез.

В свое время М. Борилло были проведены испытания этих трех процедур классификации на материалах древнегреческой архаической скульптуры [Borillo, 1974]. Эксперимент показал, что более перспективным является группировка "снизу". Это обстоятельство не означает отказа от возможности построения классификаций "сверху". Но для её реализации требуется тонкое понимание исследуемых явлений, наличие

теоретических знаний о характере структуры исследуемых естественных групп. Однако, и при группировке "снизу" также возникают проблемы, требующие определенной организации данных и определения степени их релевантности. Эта проблема оценки "веса" признака пока далека от теоретического обобщения и тем более от формализованной теории.

Поверхностный уровень. При поверхностном подходе «некоторые исследователи стали попросту обходить различные аспекты таких реконструкций, ограничиваясь публикацией материала, определением его хронологии, этнокультурных или историко-культурных связей. Появились и теоретические разработки, ограничивающие задачи археологии такой первичной классификацией и публикацией добываемых объектов. С другой стороны, конкретное исследование археологических материалов как источника по вопросам истории хозяйства и общественных отношений начало подменяться трафаретными рассуждениями, традиционными штампами, механически присоединяемыми к археологическим публикациям. Об этом справедливо писал видный советский археолог-первобытчик С.Н.Замятнин: «Данные археологии часто привлекаются только как иллюстрации тех или иных положений, априорно принимаемых... Возможности, заложенные в археологических источниках, используются далеко не полностью и подчас заменяются приведением ставших уже привычными, неоднократно повторяемых традиционных выводов и немногих поверхностных этнографических сопоставлений» [Массон, 1976].

На поверхностный уровень подхода влияет и заблуждение археологов-традиционистов в том, что организационные характеристики археологических данных информируют их непосредственно о характере археологической культуры. На самом деле простая процентная демонстрация структурных характеристик и группировок археологических комплексов еще не дает полной информации о характере процессов в прошлом. Ссылки на различные картины распределения, как на доказательство данной атрибуции значения явления, представляются неубедительными. Вот как писал об этом Пиггот: «Мы должны признать, что в археологии не существует иных фактов, кроме "наблюдаемых данных"... То, что мы, как доисторики, имеем в своем распоряжении – это случайно сохранившиеся пережитки материальной культуры, которые мы интерпретируем так, как можем, и неизбежно специфика этого источника определяет тип информации, который мы можем извлечь из него» [Binford, 1972, p. 7]. Для подобного подхода характерен язык простых запросов.

Неглубокий уровень. Оперативная аналитическая обработка данных

Очень часто информационно-аналитические системы, создаваемые в расчете на непосредственное использование лицами, принимающими решения, оказываются чрезвычайно простыми в применении, но жестко ограничены в функциональности. Такие статические системы содержат в себе предопределенные множества запросов и, будучи достаточными для повседневного обзора, неспособны ответить на все вопросы к имеющимся данным, которые могут возникнуть при принятии решений. Результатом работы такой системы, как правило, являются многостраничные отчеты, после тщательного изучения которых у аналитика появляется новая серия вопросов. Однако каждый новый запрос, непредусмотренный при проектировании такой системы, должен быть сначала формально описан, закодирован программистом и только затем выполнен. Время ожидания в таком случае может составлять часы и дни, что не всегда приемлемо.

Таким образом, внешняя простота статических систе́м подде́ржки приня́тия реше́ний (СППР) (англ. Decision Support System, DSS), за которую активно борется большинство заказчиков информационно-аналитических систем, оборачивается катастрофической потерей гибкости.

Скрытый уровень. Связан с технологией Data Mining и представляет собой процесс обнаружения в сырых археологических данных (row archaeological data) ранее неизвестных, нетривиальных, доступных для интерпретации знаний. Динамические

СППР, в отличие от статических, ориентированы на обработку нерегламентированных (ad hoc) запросов аналитиков к данным. Наиболее глубоко требования к таким системам рассмотрел E. F. Codd в статье [Codd, 1993], положившей начало концепции OLAP. Работа аналитиков с этими системами заключается в интерактивной последовательности формирования запросов и изучения их результатов.

В основе концепции OLAP лежит принцип многомерного представления данных. В 1993 году в статье [Codd, 1993] E. F. Codd рассмотрел недостатки реляционной модели, в первую очередь, указав на невозможность «объединять, просматривать и анализировать данные с точки зрения множественности измерений, то есть самым понятным для корпоративных аналитиков способом», и определил общие требования к системам OLAP, расширяющим функциональность реляционных СУБД и включающим многомерный анализ как одну из своих характеристик.

В большом числе публикаций аббревиатурой OLAP обозначается не только многомерный взгляд на данные, но и хранение самих данных в многомерной БД. Вообще говоря, это неверно, поскольку сам Кодд отмечает, что «Реляционные БД были, есть и будут наиболее подходящей технологией для хранения корпоративных данных. Необходимость существует не в новой технологии БД, а, скорее, в средствах анализа, дополняющих функции существующих СУБД и достаточно гибких, чтобы предусмотреть и автоматизировать разные виды интеллектуального анализа, присущие OLAP». По Кодду, многомерное концептуальное представление (multi-dimensional conceptual view) представляет собой множественную перспективу, состоящую из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одновременный анализ по нескольким измерениям определяется как многомерный анализ. Каждое измерение включает направления консолидации данных, состоящие из серии последовательных уровней обобщения, где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению.

Задачи Data Mining.

В основу технологии Data Mining положена концепция шаблонов (pattern), представляющих собой закономерности. В результате обнаружения этих, скрытых от невооруженного глаза закономерностей решаются задачи Data Mining. Различным типам закономерностей, которые могут быть выражены в форме, понятной человеку, соответствуют определенные задачи Data Mining [Дюк. 2002]. Задачи (tasks) Data Mining иногда называют закономерностями (regularity) или техниками (techniques).

Единого мнения относительно того, какие задачи следует относить к Data Mining, нет. Большинство авторитетных источников выделяют пять стандартных типов закономерностей, которые позволяют выявлять методы Data Mining: ассоциация, последовательность, классификация, кластеризация и прогнозирование (рис. 2). Некоторые исследователи добавляют к ним визуализацию, анализ и обнаружение отклонений, оценивание, анализ связей, подведение итогов (Рис.2).

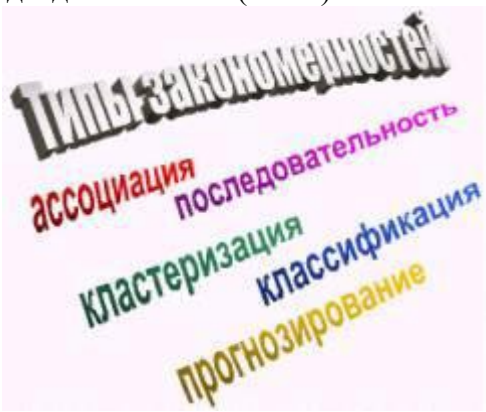


Рис. 2. Типы закономерностей, выявляемых методами Data Mining [Дюк, 2002].

Цель описания, которое следует ниже, – дать общее представление о задачах Data Mining, сравнить некоторые из них, а также представить некоторые методы, с помощью которых эти задачи решаются. Таким образом, все задачи Data Mining подразделяются по типам производимой информации, это наиболее общая классификация задач Data Mining.

1.2.1. Ассоциации (Associations)**.** Ассоциативное мышление – это мышление, которое происходит благодаря оперированию образами, возникающими в памяти человека. Каждый образ индивидуален и вызывает другие, связан с ними известными только их обладателю связями, и черпается из личного опыта человека. Любое слово может вызывать целую картину образов, с ним связанных. На этом свойстве разума базируется память и творческое мышление человека.

Из ассоциативно–образного мышления вытекает также способность человека творить что – то новое, генерировать новые идеи. Данный вид мышления способствует развитию памяти и внимания благодаря созданию ассоциативных связей между предметами и явлениями, а также позволяет понять новую информацию на основе уже имеющейся. Чем большее количество различных образов мы накапливаем, тем шире и разнообразнее возможность совершения в разуме операций с их использованием, и тем лучше мы можем развить память и творческое мышление.

С помощью упражнений для развития ассоциативного мышления можно увеличить количество ассоциативных образов и связей, а значит развить свой творческий потенциал.

В ходе решения задачи поиска ассоциативных правил отыскиваются закономерности между связанными событиями в наборе данных. Ассоциация имеет место в том случае, если несколько событий связаны друг с другом. В данном случае поиск закономерностей осуществляется не на основе свойств анализируемого объекта, а между несколькими событиями, которые происходят одновременно. Наиболее известный алгоритм решения задачи поиска ассоциативных правил – алгоритм Apriori [Дюк. 2002]. Кроме этого в археологии применяются:

Методы дисперсионного анализа – для выявления количественных связей между признаками объектов;

Методы факторного анализа и многомерного шкалирования – для выявления скрытых связей между объектами и признаками их описания. Метод главных компонент связан с представлением о пространстве, расстоянии и измерении. В методе использованы три основные методологические идеи:

Первая сводится к тому, что объекты рассматриваются как точки в пространстве, расстояние между которыми принимается равным расстоянию между объектами. Однако обычно факторы, полученные методом главных компонент, не поддаются наглядной интерпретации. Поэтому напрашивается следующий шаг.

Вторая основная идея была связана с выбором системы координат, при которой исследуемое пространство может вращаться вокруг фиксированного центра равновесия таким образом, чтобы облегчить их интерпретацию.

Третья идея основана на упрощении, т.е. на уменьшении числа показателей по сравнению с тем, которое требовалось для первоначального представления данных.

Наиболее активно факторный анализ стал применяться Л. Бинфордом [Binford&Binford, 1966]. Поскольку, по мнению Л.Бинфорда, каждый артефакт может выполнять функции разных уровней культуры – техномического, социотехнического и идеотехнического, то приходится рассматривать по отдельности разные его аспекты. Соответственно, в классификационных исследованиях приходится работать не с артефактами, а с их деталями и признаками. Бинфорд пришел к тому, что поставил под вопрос существование типов в археологии и выдвинул требование искать более сложные и более тонкие взаимоотношения признаков разных комплексов – с помощью Метода главных компонент.

Примером применения факторного анализа может служить анализ палеолитических орудий в статье четы Бинфордов в 1966 г. "Предварительный анализ функциональной вариабельности левалуазской фации" [Binford&Binford, 1966]. В этой и других работах

Бинфорд пришел к выводу, что фации мустье Франсуа Борда, являются сезонно обусловленными "структурными позами".

В отличие от ассоциации "Последовательность" (Sequence), или последовательная ассоциация (sequential association) позволяет найти временные закономерности между транзакциями. Задача последовательности подобна ассоциации, но ее целью является установление закономерностей не между одновременно наступающими событиями, а между событиями, связанными во времени (т.е. происходящими с некоторым определенным интервалом во времени). Другими словами, последовательность определяется высокой вероятностью цепочки связанных во времени событий. Фактически, ассоциация является частным случаем последовательности с временным шагом, равным нулю. Эту задачу Data Mining также называют задачей нахождения последовательных шаблонов (sequential pattern) [Дюк, 2002].

1.2.2. Классификация (Classification). Классификация является основной процедурой, посредством которой мы вносим некоторый порядок и связность в поток информации из реального мира [Харвей, 1973: 313].

Классификация наиболее простая и распространенная задача Data Mining. В результате решения задачи классификации обнаруживаются признаки, которые характеризуют группы объектов исследуемого набора данных – классы; по этим признакам новый объект можно отнести к тому или иному классу. Это делается посредством анализа уже классифицированных объектов и формулирования некоторого набора правил.

Методы решения. Для решения задачи классификации могут использоваться методы: ближайшего соседа (Nearest Neighbor); k-ближайшего соседа (k-Nearest Neighbor); байесовские сети (Bayesian Networks); индукция деревьев решений; нейронные сети (neural networks) и др. [Дюк, 2002].

В российской археологии Л.С. Клейн уделял особенно много внимания приёмам упорядочения, группирования — классификации и типологии. Он мотивировал различение этих понятий в археологии, увязав первое (и связанное с ним понятие класса) — с жёстким логическим членением, а второе (и связанные с ним понятия типа, типического и типизации) — с роением признаков вокруг идеальной нормы. В первом случае материал как бы раскладывается весь без остатка по "ящичкам" и их "отсекам", любой объект попадает в какой-то один "ящичек" и "отсек", по своим признакам. Это удобно, по его мнению, для упорядочивания и математической обработки. Во втором случае объект может по одним признакам тяготеть к одному идеальному образу, по другим — к другому (чётких границ между ними нет), а какие-то (атипичные) объекты — ни к какому. Это удобно для прослеживания реальных связей в материале. Л.С. Клейн показал, что эти виды группирования взаимоисключаются. Соединение преимуществ того и другого — очень сложная проблема и сопряжена с введением условности [Клейн, 1978].

Обычная процедура группирования предусматривает расчленение материала на элементарные ячейки, а затем эти ячейки объединяются по общности признаков во всё более крупные блоки: признаки артефактов, сгущаясь, дают разные виды деталей артефактов, те складываются в типы целых артефактов, типы в культуры и т.д. На практике, эмпирически выявляемых общностей может оказаться очень много, тогда как вопрос, какие из них имеют функциональное и вообще культурное значение, остаётся открытым. Л.С. Клейн доказывает, что выяснить это без привлечения посторонней информации принципиально невозможно: «исследователь только внешне поступал по обычному правилу: выделял элементарные признаки вещей, затем складывал их в типы, а типы группировал в культуру. На самом деле, он как бы тайно подсматривал вперёд — он заведомо знал, какие признаки культурно значимы, потому что в уме шёл противоположным путём: не от признаков через тип к культуре, а от культуры через тип к признакам» [Клейн, 1991].

Л.С. Клейн предложил противоположную стратегию группирования, в частности, типологии. Эта новая стратегия, которую он назвал системной, подразумевает опору на

предзнание: нужно заведомо иметь некое знание о культурном значении признаков и типов. Такое знание дают культуры, поэтому им было предложено двигаться от культур к типам, а от них — к признакам. Это предусматривает познание культур не через типы и признаки, а как-то иначе — целостным восприятием, выявлением эвидентных типов (очевидных до и без классификации) и т.п.

В исторических науках соответствующий процесс абстрагирования является иногда очень непростым. Основными его этапами является выделение понятий (процесс рождения которых уже не прост) и осуществление их т.н. операционализации. Процессу операционализации понятий посвящена обширная литература [Клейн, 1991; Холюшкин, 2010].

Так, в свое время Р. Даннел сказал, что палеоистория имеет обыкновение специально изобретать для себя термин и потом спорить двадцать лет о том, что он значит, вместо того, чтобы определить этот термин заранее [Dunnell, 1971: 4; Клейн, 1991: 125].

Такое заявление свидетельствует лишь о том, что археологи не всегда отчетливо осознают тот факт, что «всякое понятие, которым в данной предметной области выражают некоторый объект, свойство, явление или процесс, не является "элементарным кирпичиком мира"».

Нами была на основе метода Е.Д. Гражданникова создана системная классификация археологических понятий, в которой всякое понятие выражает некоторую структуру и системно организованный набор других понятий. Именно через этот системно организованный набор понятий мы осознаем семантику исходного понятия, объясняем, интерпретируем и используем это понятие в некоторых границах, которые также определяются через системно организованный набор понятий» [Деревянко, Фелингер, Холюшкин, 1989: 12].

1.2.3. Кластеризация отличается от классификации тем, что сами группы заранее не заданы. С помощью кластеризации средства Data Mining самостоятельно выделяют различные однородные группы данных.

Подходы авторов настоящей монографии основаны на идее информационного пространства археологических объектов как множества их описаний. В этом пространстве как описание, так археологические данные неоднородны, каждый артефакт уникален, существует множество пропусков и пробелов в информации об этих объектах. Поэтому в таком информационном пространстве приходится выбирать и исследовать обобщенные свойства и признаки археологических памятников и артефактов. На этой же идее строится гипотеза о возможности исследования археологических объектов на различных методологических основаниях, обусловленных использованием разных пространственно-временных метрик. На основе выбранных метрик предлагаются соответствующие процедуры, методы и технологии исследования археологических объектов. Среди приемов кластеризации можно выделить следующие:

а) Алгоритмы анализа структуры в анализе археологических данных. Информация, накопленная в ходе археологических исследований, требует обобщения и анализа. Ее анализ без использования ПК не может быть полным и охватит лишь небольшую часть огромного фонда данных. Эффективно эту работу можно сделать во многих случаях, лишь используя специальные алгоритмы и программы анализа данных на ЭВМ.

Среди множества методов анализа данных в настоящем подпункте мы рассмотрим методы прямой кластеризации данных и автоматизации типологического анализа связи, которые, как нам представляется, могут оказаться полезными для исследования сложных и не полностью определенных историко-археологических данных.

Прямой кластерный анализ матриц данных развивался во многих работах отечественных и зарубежных авторов [Hartigan, 1975; Браверман, Мучник, 1983; Жамбю, 1988; Миркин, 1980; Ростовцев, 1982, ряд других авторов].

Алгоритмы, описанные в этих работах, затрагивали преимущественно структуры, основанные на прямоугольных блоках. В данной работе мы рассматриваем менее ограниченный класс структур, основанный на связных областях.

Типологическое исследование данных состоит в построении логических классификаций объектов. Эта задача также является достаточно распространенной в применении к естественно-научным и гуманитарным исследованиям [Лбов, 1981; Устинов и Фелингер 1973 и др.]. Обычно рассматриваемые алгоритмы позволяют строить логические описания для детерминации классов разбиения совокупности объектов. В случае использования разнотипных переменных это описание ограничивается дихотомическим последовательным разбиением множества объектов. Здесь рассматривается логика последовательного разбиения (не обязательно дихотомического) с последующим объединением – "синтезом" типов. Целью построения классификации является минимизация остаточной обобщенной дисперсии по множеству переменных, в том числе и неколичественных. Работа является развитием разработок [Ростовцев, 1985], реализованных ранее на устаревшей в настоящее время технике.

При использовании многих методов анализа данных сложно делать выводы о статистической устойчивости результатов, поскольку до сих пор для части методов такие статистические оценки устойчивости не получены, для остальных методов эти оценки очень сложны или основаны на жестких предположениях о теоретическом распределении переменных. Для получения достоверных результатов в наших работах использовался метод Boot Strap [Efron, 1986];

б). Процедура выявления структуры таблицы. Археологу зачастую приходится иметь дело с пестрящими цифрами полотнами таблиц, при обращении к которым в поисках интересных фактов и обобщения нужно обладать исключительными интуицией, трудолюбием и опытом. В монографии приводятся специально разработанные средства для упорядочения неоднородной археологической информации и выявления ее структуры.

Эти средства дают возможность для анализа всевозможных статистик, соответствующих ячейкам приводимых в монографии таблиц, не требуя выделения общей характеристики связи. В монографии таблицы статистик, соответствующие ячейкам таблиц сопряженностей археологических комплексов, рассматриваются как матрицы сравниваемых между собой коэффициентов. Они предварительно подвергаются упорядочению перестановкой строк и столбцов, при которой строки (столбцы) рассматриваются как вершины графа и для этих вершин решается задача “коммивояжера” [Майника, 1983].

Выявление структуры таблицы происходит в два этапа. Первый этап состоит в упорядочении строк и столбцов таким образом, чтобы мало отличающиеся между собой строки (столбцы) оказались рядом. Критерием качества такого упорядочения является сумма расстояний между соседними строками (столбцами) – эта сумма минимизируется.

В ряде случаев имеет смысл отказаться от упорядочений строк и (или) столбцов, например, если строки являются временными рядами или элементы таблицы связаны с участками территории и требуется выявить однородные по какому-либо показателю регионы.

Второй этап – это разбиение элементов матрицы на связные области. Здесь критерий качества – дисперсионный: минимизируется остаточная дисперсия, получаемая при замене элементов матрицы на средние для элементов в соответствующих областях;

в) Упорядочение строк и столбцов. Естественно считать хорошо структурированной таблицу, в которой не очень часто происходят скачки по величине значений соседних элементов. Поэтому для лучшей структурированности таблицы целесообразно переставить строки и столбцы матриц так, чтобы расстояние между соседними строками, а также расстояние между соседними столбцами в сумме было небольшим. Благодаря такой перестановке строки, соответствующие памятникам, упорядочиваются по близости распределений по артефактам. Таким образом, обеспечивается сравнительно "плавный"

переход от одного типа памятников к другому. Аналогичная цель преследуется при перестановке столбцов-артефактов.

На основании этих соображений критерием качества упорядочения строк следует взять сумму расстояний между строками. Если отождествить строки матрицы с вершинами взвешенного графа, где вес ребра, соединяющего две вершины, совпадает с расстоянием между ними**,** то задача минимизации становится весьма похожей на общеизвестную задачу "коммивояжера" – поиска обхода вершин графа (гамильтонова контура) минимальной длины. Ее отличие лишь в том, что искомый путь не замкнут. Искусственное присоединение к графу "нулевой" вершины, равноудаленной от всех остальных вершин, превращает указанную задачу в точности в задачу коммивояжера. В этом случае первая и последняя строки таблицы будут связаны фиктивной вершиной.

Существующие методы решения задачи коммивояжера делятся на два класса:

методы, приводящие к полной оптимизации, но в худшем случае требующие полного перебора вариантов;

локально-оптимальные методы, не всегда приводящие к оптимуму.

Здесь предпочтение отдано методам, принадлежащим ко второму классу. Они основаны на последовательном улучшении некоторого произвольного первоначально выбранного порядка обхода вершин. Алгоритм состоит в том, что пара вершин меняются местами. В расчетах последовательно рассматриваются пары вершин и, если при смене их мест происходит уменьшение пути, они действительно меняются местами.

Для этих целей использован метод "вставки", при котором из пути исключается некоторая вершина и затем вставляется между другими вершинами. Процесс перемещения вершин происходит до тех пор, пока длина контура уменьшается. Проблема перестановки столбцов решается аналогично;

г) Разбиение таблицы на связные области. После того, как строки и столбцы упорядочены и таблица приняла непрерывный вид, она разбивается на однородные части так, чтобы эти части объясняли по возможности большую часть дисперсии элементов. Предварительно уточним: что такое связные области и каковы их основные свойства; какой требуется критерий разбиения таблицы на связные области; в чем состоит локально-оптимальный алгоритм поиска разбиения таблицы на связные области.

Определение связных областей таблицы соответствует определению связного подпространства в задачах элементарного районирования [Воронин, Градова, 1987].

Назовем областью произвольное подмножество элементов таблицы. Два элемента будем называть соприкасающимися, если они находятся рядом, притом в одной и той же строке или в одном и том же столбце. Таким образом, из числа соприкасающихся исключаются элементы, соседние по диагонали. Область называется связной, если для любой выбранной пары ее элементов можно выстроить такую последовательность элементов, в которой каждая пара членов-"соседей" по последовательности соприкасается, причем первым и соответственно последним членами являются выбранные элементы.

Связность свидетельствует о том, что любая пара элементов области взаимозависима хотя бы косвенным образом, т.е. можно построить внутри области цепочку "соприкасающихся" пар артефакт-памятник, характеризуемых близкими частотами, содержащую эту пару элементов. Две связные области называются соприкасающимися, если существует пара соприкасающихся элементов, один из которых принадлежит одной области, другой принадлежит другой. Очевидно, объединение соприкасающихся связных областей дает связную область. Разбиением таблицы на связные области называется множество непересекающихся связных областей, которые покрывают все элементы таблицы.

Критерием качества разбиения предполагается величина остаточного разброса при аппроксимации таблицы средними значениями элементов областей. При поиске разбиения эта величина должна, естественно, минимизироваться. Построение начинается с

разбиения на самые мелкие области – по одному элементу в каждой. Такое разбиение дает минимальный, нулевой остаточный разброс, однако это разбиение неприемлемо из-за сложности. Поэтому производится последовательное объединение соприкасающихся пар областей: вначале находится пара областей, слияние которых даст наименьший прирост критерия, затем следующая пара и т.д. Подобные агломеративные алгоритмы используются в кластерном анализе, в частности, при анализе изображений, в анализе геологических данных.

Алгоритм объединения может работать до тех пор, пока все элементы таблицы не объединятся в одну область; возможна также ситуация, когда на некотором шаге объединения полученные области невозможно объединить из-за имеющихся в таблице неопределенных клеток. Однако такое окончание процесса бесполезно. На современном уровне решения задачи наиболее целесообразным является задание исследователем определенного числа кластеров, соответствующего его представлениям о сложности таблицы. В случае неудовлетворительных результатов имеется возможность найти новую структуру с другим числом кластеров. Формальный критерий для числа кластеров базируется на следующих эвристических соображениях. Целесообразно ли увеличивать число кластеров, если при переходе к большему на единицу числу кластеров прирост объясненного структурой разброса составляет менее среднего разброса в пересчете на один элемент таблицы?

По-видимому, в большинстве случаев новый кластер, не объясняющий и доли дисперсии, приходящейся на элемент таблицы, не имеет смысла. Соответственно, если при слиянии кластеров потеря объясненного разброса превышает такую долю, то подобную потерю следует считать "существенной". Разумеется, приведенный критерий формален, приоритет имеют содержательные соображения.

В реализациях локально-оптимальных алгоритмов кластерного анализа целесообразно не ограничиваться агрегированием, а дополнить алгоритм процедурой перемещения объектов из класса в класс. Именно этим заканчивается работа по выявлению структуры таблицы.

Арсенал методов анализа данных, применяемых нами в серии монографий, далеко не ограничивается выявлением структуры таблиц археологических данных. Дополнительно в качестве отдельных методов и инструментов, встроенных в технологию обработки информации, нами включены и другие методы анализа данных:

д) Методы группирования и кластерного анализа – для выделения кластеров и типов объектов. Кластерный анализ представляет собой средство исследования топологической структуры совокупности объектов. Он позволяет разбить множество объектов в признаковом пространстве на классы близких между собой объектов. Обнаруженные этим методом "сгустки" объектов, называемые кластерами (таксонами, классами), позволяют сформулировать в конечном итоге гипотезы о логической структуре совокупности. В частности, этим методом можно изучать кластерную структуру множества археологических памятников по наличию и частоте встречаемости артефактов, исследовать информацию по другим совокупностям, представимым прямоугольными матрицами вида "объект — признак". Методы кластеризации разделяются на агломеративные и дивизивные.

В настоящей монографии применяются различные методы кластерного анализа (k-средних, иерархический кластерный анализ, кластерный анализ с использованием логики группирования по множеству "независимых" переменных).

Разнообразные типы кластерного анализа активно применялись и применяются в археологических исследованиях [Жамбю, 1988; Деревянко, Фелингер, Холюшкин, 1989; Деревянко, Холюшкин, Ростовцев, Воронин, 1998 и др.]. В них авторами осознавались недостатки процедур кластерного анализа, главными из которых являются три: 1) отсутствие четких рекомендаций по выбору числа классов; 2) невозможность индивидуального учета отдельных элементов при объединении классов. 3) результаты

кластеризации зависят от выбранного метода, и эта зависимость тем сильнее, чем менее явно изучаемая совокупность разделяется на группы объектов.

Кроме того, в ходе таких исследований обнаружено, что кластеры, замечательным образом, найденные в первый раз и разумно описанные исследователем, после повторного сбора информации (новых раскопок, других исследовательских приемов исследования и нового применения кластерного анализа) могут "рассыпаться" из-за случайности выявленной кластерной структуры.

Поэтому результаты кластеризации могут быть дискуссионны и часто служат лишь подспорьем для содержательного анализа.

Заметим также, что методы кластерного анализа не дают какого-либо способа для проверки статистической гипотезы об адекватности полученных классификаций.

Проблемы, с которыми мы сталкиваемся, связаны с вариабельностью в наших наборах данных и неспособностью классических методов помочь нам при нормальных обстоятельствах (т.е., при малых выборках, ненормальных распределениях, плохо обоснованных моделях и т.д.).

е) Метод повторной выборки с возвращением. Проблема получения устойчивых результатов всегда волнует добросовестного исследователя, ответственно относящегося как к научным, так и к практическим результатам. Однако данная задача едва ли может быть решена традиционными методами математической статистики из-за сложности алгоритмов поиска структур, из-за взвешенности данных.

В связи с этим используется метод повторной выборки с возвращением, известный как метод bootstrap [Efron and Diaconis, 1983]. Этот метод был применен более тридцати лет назад экономистом Ю. Саймоном. Он требует много вычислений для анализа данных, когда используется моделирование при создании многих (часто тысяч) объектов заданным набором данных, чтобы извлечь как можно больше информации, избегая применения статистических формул. Через этот процесс пользователь способен оценить степень, в которой результат эксперимента является пригодным или не пригодным. Поскольку он ориентирован на имеющиеся данные, он запрещает делать любые варианты статистических заключений (т.е., гауссовской кривой) и предоставляет право данным "говорить самим за себя". Критики этого метода утверждают, что качество анализа при этом полностью зависит от адекватности имеющейся выборки наблюдений (первичное условие повторной выборки), однако его защитники доказывают, что, хотя это и так, но повторная выборка вынуждает пользователя творчески и более тщательно думать относительно данных и их вариабельности вместо того, чтобы приспособить данные к возможно неверной формуле. Короче говоря, метод повторной выборки ставит на первый план скорее процесс рассуждения и использования интуиции, нежели детали формальных подходов [Simon, 1993, 1994: 290].

Саймон считал, что апатия и враждебность к повторной выборке частично связаны с поколением: большинство преподавателей вводных курсов по статистике вообще не видят никаких причин для перемен [Peterson, 1991: p. 58]. Правда со временем профессиональные статистики восприняли повторную выборку как подход для решения наиболее трудных проблем в математической статистике [Edgington, 1995; Efron & Tibshirani, 1993; Good, 1994]. Как уже было сказано выше, перед исследователем всегда стоит вопрос: не разрушится ли выявленная структура при последующих исследованиях археологических комплексов и повторном анализе данных. Задача избежать этого разрушения традиционными методами математической статистики из-за сложности алгоритмов поиска структур, из-за взвешенности данных едва ли может быть решена. К сожалению, понимание этой фундаментальной проблемы мало затронуло археологию. Мы можем привести лишь ряд экспериментов с повторной выборкой.

Так, Кинти [Kintigh, 1984] использовал выборку с помощью метода Монте Карло, чтобы генерировать псевдодоверительные интервалы для результатов анализа многообразий и k-значной кластеризации пространственных данных. Рингроуз [Ringrose,

1992] применил bootstrap для оценки подобным способом результатов анализа соответствия.

Суть метода повторной выборки с возвращением в нашем случае состоит в следующем: предполагается, что собранные данные репрезентативны, т.е. двумерные распределения для каждой изучаемой таблицы соответствуют (или почти соответствуют) распределению генеральной совокупности. При этом предположении, извлекая объекты из имеющейся совокупности и включая их в генеральный массив данных, мы будем имитировать повторный сбор данных. Следуя методу в каждом эксперименте, мы генерируем выборку, объем которой совпадает с исходными данными.

При этом мы должны подчеркнуть, что повторная выборка никак не противостоит классическим доказательным методам и эти два подхода могут работать вместе и весьма успешно. Необходимо, однако, заметить, что повторная выборка требует иной логики;

ё) Сравнение классификаций. При проведении автоматической классификации возникает проблема и в том, насколько выделенные программой классы отражают реальную структуру данных, а не случайную флуктуацию расположения точек в признаковом пространстве. В данной работе предлагается метод проверки неслучайности найденной кластерной структуры, основанный на сравнении классификаций, построенных на разных признаковых пространствах и возможно, разными методами [Костин, 2003: 57-65]. При этом определяется степень согласованности классификаций и статистическая значимость полученной величины путем построения функции ее распределения в условиях нулевой гипотезы. При этом были сформулированы требования, которым должен удовлетворять искомый показатель степени согласованности классификаций:

Во-первых, он должен быть нечувствителен к порядку нумерации классов. Это требование вытекает из того, что процедура автоматической классификации выделяет классы объектов, не учитывая их содержательной характеристики, а опираясь исключительно на особенности взаимного расположения объектов как точек в многомерном признаковом пространстве. Поэтому номер класса является не более, чем условным идентификатором.

Во-вторых, наш показатель должен измерять степень согласованности даже при несовпадении количества классов в сравниваемых классификациях, поскольку иначе его практическое применение будет неоправданно ограничено.

В-третьих, он должен давать максимальное значение (например, 1) при сравнении классификации с собой.

ж**) Построение обобщенной классификации.** Кроме того, в работе реализован метод построения сводной обобщенной классификации, основанный на анализе совпадения разных классификаций одних и тех же объектов [Костин, Корнюхин, 2003: 65-72].

Исходные данные, для построения обобщенной классификации, представляются в виде таблицы «объект-свойство», где объектами выступают памятники, а свойствами – номера кластеров, к которым они были отнесены в результате проведения каждой из классификационных процедур).

Для перехода к методу выделения наиболее устойчиво совместно классифицирующихся объектов в ядра кластеров обобщенной классификации вводится статистика для измерения степени близости объектов по результатам классификаций [Костин, Корнюхин, 2003].

С помощью методики моделируются условия нулевой гипотезы на наших данных, проведя случайное перемешивание клеток внутри каждого из столбцов.

В случае, когда все параметры, за исключением асимметрии, соответствуют стандартному нормальному распределению, проверяется гипотеза о наличии кластерной структуры еще до выполнения процедуры классификации. Принятие нулевой гипотезы *H* будет означать, что все *N* объектов в среднем расклассифицированы по *m* классификациям независимо друг от друга и потому искомая обобщенная классификация будет не более, чем случайным результатом эвристической процедуры.

Если же гипотеза H отвергается, мы можем переходить к построению искомой обобщенной классификации.

Если стандартное отклонение достаточно далеко от единицы, то это несовместимо с выполнением нулевой гипотезы.

Тогда, согласно методике Костина и Корнюхина строится квадратная матрица, элементы которой представляют собой отклонения совпадений от ожиданий по m классификациям для пар i и j объектов.

Как обычно, кластер в матрице представляется подмножеством строк (или столбцов с теми же номерами). Пересечение элементов этих строк и столбцов определяет блок матрицы. Удобно рассматривать матрицу после такой совместной перестановки строк и столбцов, которая собирает все клетки блока вместе. Тогда матрица приобретает блочно-диагональный вид. Внутри блоков значения их элементов должны быть больше, чем за их пределами. В прямоугольных блоках на пересечении строк и столбцов из разных диагональных блоков, объединены показатели близости объектов, принадлежащих разным блокам, поэтому этот набор элементов характеризует степень контраста двух кластеров.

Для выделения кластеров задается некоторое пороговое значение α – уровня значимости. Будем считать, что объекты образуют кластер только в том случае, когда гипотеза о равенстве нулю среднего значения элементов внутри блока отвергается на уровне значимости α. Это позволяет учесть не только среднее элементов, но и объем кластера. Проверку гипотезы можем проводить по t-критерию Стьюдента.

Выделенные блоки могут пересекаться. Тогда в соответствии с методикой кластером считается объединение блоков.

Некоторые объекты могут не входить ни в один из выделенных кластеров. Они считаются выпавшими из обобщенной классификации.

Процедуру выделения кластеров состоит из двух последовательных шагов.

На первом шаге переставляются строки и столбцы матрицы, располагая вместе наиболее близкие объекты. С этой задачей успешно справляется программа поиска структуры в таблицах сопряженности [Ростовцев, Костин, Корнюхин, Смирнова, 1994: 60-61].

На втором шаге для всех возможных диагональных блоков вычисляется значимость гипотезы о равенстве нулю среднего значения Z_{ij} внутри блока. Если значимость меньше пороговой (α), блок присоединяется к уже найденным.

Уменьшая пороговое значение, получается ряд кластерных разбиений, выделяющих все более и более устойчивые ядра. Таким путем строится своеобразная "карта уровней", напоминающая рельеф горной местности на физических картах.

Результат построения обобщенной классификации создаются диагональные блоки выделенные оттенками серого в зависимости от пороговой значимости, на уровне которой среднее Z_{ij} в блоке отличается от нуля. Применение описываемой информационной технологии в целом и на достигнутом уровне научных знаний предполагает комплексное исследование пока лишь отдельных аспектов затронутых выше проблем. В ходе такого изучения определяется научная задача, исходя из тех археологических данных, которыми располагает исследователь.

В процессе количественного и экспертного анализа формулируется научная гипотеза (сначала ее предпосылки — с помощью дисперсионного анализа, затем — основные взаимозависимости археологических комплексов и артефактов, составляющие ее структурную основу, и, наконец, сама гипотеза о характере количественных закономерностей, определяющих структуру и поведение исследуемой совокупности объектов), затем эта гипотеза благодаря дальнейшим исследованиям на устойчивость выявленной структуры принимается или отвергается.

Методические подходы, лежащие в основе подобной технологии, строятся на том, что все научные гипотезы взаимосвязаны и в той или иной степени дают ответ

(положительный или отрицательный) на центральный вопрос проблемы, подтверждая или отвергая друг друга.

Основой для всевозможных систем прогнозирования служит историческая информация, хранящаяся в БД в виде временных рядов. Если удается построить найти шаблоны, адекватно отражающие динамику поведения целевых показателей, есть вероятность, что с их помощью можно предсказать и поведение системы в будущем.

Прогнозирование (Forecasting). Краткое описание. В результате решения задачи прогнозирования на основе особенностей исторических данных оцениваются пропущенные или же будущие значения целевых численных показателей. Для решения таких задач широко применяются методы математической статистики, нейронные сети и др.

2. Классы систем Data Mining.

Data Mining является мультидисциплинарной областью, возникшей и развивающейся на базе достижений прикладной статистики, распознавания образов, методов искусственного интеллекта, теории баз данных и др. (рис. 3). Отсюда обилие методов и алгоритмов, реализованных в различных действующих системах Data Mining. Многие из таких систем интегрируют в себе сразу несколько подходов. Тем не менее, как правило, в каждой системе имеется какая-то ключевая компонента, на которую делается главная ставка. Ниже приводится классификация указанных ключевых компонент на основе работы. Выделенным классам дается краткая характеристика.

Рис. 3. Data Mining — мультидисциплинарная область (по Дюк,2002)

2.1. Распознавание образов. Распознавание образов – это научная дисциплина, целью которой является классификация объектов по нескольким категориям или классам. Объекты называются образами. Классификация основывается на прецедентах. Прецедент – это образ, правильная классификация которого известна. Идея принятия решений на основе прецедентности – основополагающая в естественно-научном мировоззрении. Задача распознавания образов является основной в большинстве интеллектуальных систем. Работы в этом направлении проводились новосибирскими математиками в соавторстве с археологами НГУ. Теория искусственного интеллекта, ИИ (artifical intelligence, AI) – общее понятие, описывающее «способность вычислительной машины моделировать процесс мышления за счет выполнения функций, которые обычно связывают с человеческим интеллектом. Сюда не входят задачи, для которых известна процедура решения (интегрирование обыкновенных дифференциальных уравнений, решение системы линейных уравнений и т.д.). Обычно к сфере ИИ относят построение и использование экспертных систем, логический вывод (доказательство теорем и правильности программ), понимание естественных языков, зрительное и слуховое восприятие. Иногда элементы ИИ реализуются в некоторых пространственно-

аналитических и геомоделирующих блоках и причисляются к функциональным возможностям ГИС.

В настоящее время одними из наиболее перспективных методов анализа данных в археологии являются методы, основанные на классе логических решающих функций [Лбов, Бериков, 2005]. Отличительной особенностью этих методов является возможность построения логико-вероятностной модели изучаемых объектов, представленной в виде списка логических закономерностей («знаний»). Разработанные методы позволяют, в частности, проводить анализ форм защитного покрытия средневековых номадов Сибири [Борисенко и др., 2008].

2.3. **Визуализация** (Visualization, Graph Mining). В результате визуализации создается графический образ анализируемых данных. Для решения задачи визуализации используются графические методы, показывающие наличие закономерностей в данных. Пример методов визуализации – представление данных в 2D и 3D измерениях. В той или иной мере средства для графического отображения данных поддерживаются всеми системами Data Mining. Вместе с тем, весьма внушительную долю рынка занимают системы, специализирующиеся исключительно на этой функции. Примером здесь может служить программа DataMiner 3D словацкой фирмы Dimension 5 (5-е измерение).

В подобных системах основное внимание сконцентрировано на дружелюбности пользовательского интерфейса, позволяющего ассоциировать с анализируемыми показателями различные параметры диаграммы рассеивания объектов (записей) базы данных. К таким параметрам относятся цвет, форма, ориентация относительно собственной оси, размеры и другие свойства графических элементов изображения. Кроме того, системы визуализации данных снабжены удобными средствами для масштабирования и вращения изображений.

В задачу предварительного анализа входит проверка корректности данных. Ошибку в данных легче увидеть на графике, чем в таблице. Например, для количественной переменной ошибки (опечатки) часто проявляются в виде выпадающих значений, отстоящих на значительном расстоянии от основной массы значений. Другой, не менее важной задачей предварительного анализа данных является поиск ответа на вопрос, обладает ли какой-либо (явной или скрытой) структурой анализируемая таблица данных. Достаточно простым и эффективным средством является "серый" (или "спектральный") анализ (рис. 4). Его суть состоит в том, что анализируемая таблица дополняется графической схемой, которая представляет собой образ таблицы в виде прямоугольника, разделенного на ячейки, подобно клеткам исходной таблицы. При "сером" анализе каждая клетка схемы заполняется (заливается) оттенком серого цвета в зависимости от того, какие значения принимает соответствующий признак для данного объекта. Предварительно промежуток, в который попадают числовые значения всех признаков, разбивается на конечное число равных интервалов. Каждому интервалу сопоставляется определенный оттенок серого цвета по правилу – чем больше значения признаков, которые попадают в данный интервал, тем темнее окрашиваются в серый цвет соответствующие клетки таблицы. Результатом серого анализа является наглядный образ данных, где их структура представлена наиболее отчетливо (рис. 4).

Не менее интересной может быть демонстрацция модели трехмерного изображения проверки устойчивости результатов кластерного анализа (рис. 5).

Ключевое слово «визуализация» во многом определяет задачи виртуальной археологии. Компьютерная реконструкция позволяет сохранить утраченную информацию о памятнике или даже воссоздать его на основе специальных анализов и всесторонних исследований. На рис. 6 представлен один из залов этнографической коллекции виртуального музея. К виртуальной археологии следует отнести все виды современных компьютерных технологий для археологических исследований, обработки данных, моделирования, археологических и исторических реконструкций и визуализации результатов (технологии многомерного моделирования исторических ландшафтов,

археологических памятников, объектов и находок, GIS-моделирование природных и исторических процессов, мониторинг объектов культурного наследия, проектирование виртуальной реальности.

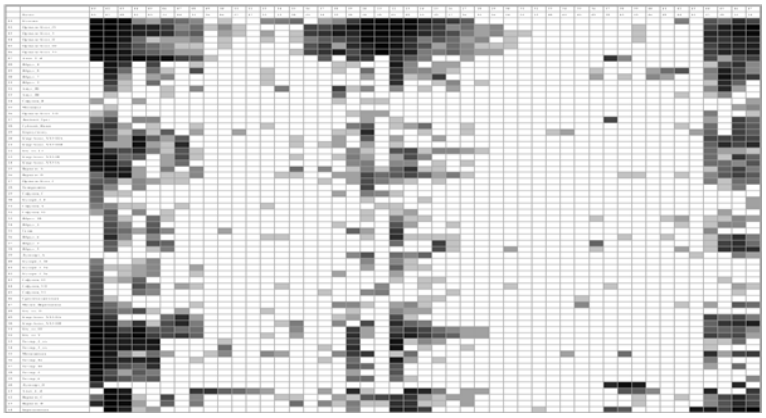


Рис. 4. Результаты проведения серого анализа

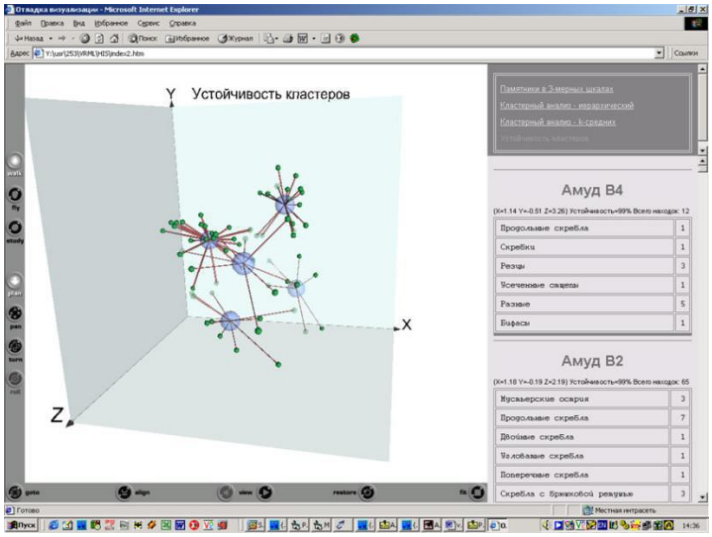


Рис. 5. 3d Проверка устойчивости результатов кластерного анализа

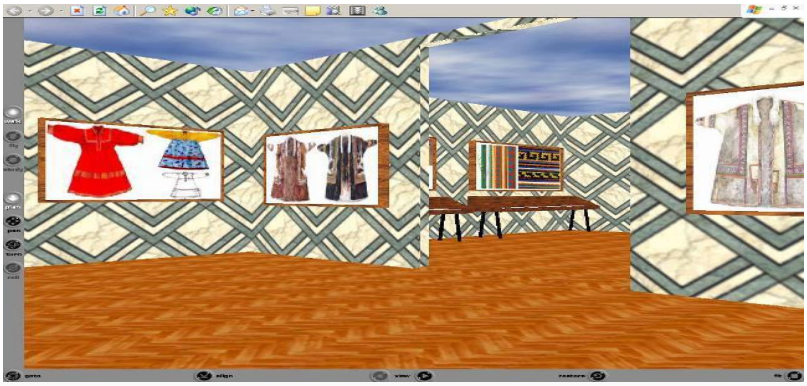


Рис. 6. Трехмерная модель виртуального музея «Древняя история, культура Северной Азии».

2.3. Экспертные системы. Теория экспертных систем Область разработок теории создания – систем искусственного интеллекта, включающая знания об определенной слабо структурированной и трудно формализуемой узкой предметной области и способная предлагать и объяснять пользователю разумные решения. Экспертная система состоит из базы знаний, механизма логического вывода и подсистемы объяснений.

Практические успехи в таких областях, как медицина, вычислительная техника, программирование, генетика, геология, спектральный анализ и, наконец, археология [Gardin, 1988] показывают, что ЭС являются одним из наиболее перспективных направлений, могущих разрешить ряд проблем в археологической исследовательской

процедуре, основанной на анализе археологической информации. Эвристический характер знания делает их приобретение весьма трудоемким процессом и это является наиболее узким местом в создании экспертных систем. Трудность этого процесса усугубляется тем, что многие из научных задач не могут быть заданы в числовой форме и выражены в точно определенных терминах. Эти проблемы в полной мере характерны и для археологии. В ней существуют области, для которых нет алгоритмического решения задач, а во многих областях, где такие решения существуют, применение алгоритмов ограничено ресурсами (время, память).

Таким образом, археология, как и ряд других наук, имеет широкий круг областей, для решения проблемных задач которых требуется применение неформализованных задач. Их особенностями являются ошибочность определений, неполнота исходных данных, противоречивость знаний о проблемных областях и решаемых задачах, а так же динамические изменяющиеся данные. Свежим примером является публикация В.Е. Щелинского, в которой показано различие между типологическим и функциональным определением каменного инвентаря пещеры Азых. И это не единственная область для создания экспертных систем в археологии. Не меньшую проблему представляет большая размерность пространства решения проблем хронологических параллелей и характер прямых аналогий в индустриях отдаленных территорий (диффузия, миграция, обмен, торговля и т.д.). Для решения их требуется конструирование цепочек описательных интерпретационных предположений.

Таким образом, в археологии неформализованные задачи представляют большой и важный класс задач [Дородницын, 1985], требующих эвристических поисков решения. Для их решения и требуется приложение ЭС, не уступающих по качеству и эффективности заключениям эксперта-археолога. В отличие от решений, предлагаемых статистическими методами, где всегда существует проблема интерпретации полученных результатов, решения ЭС обладают «прозрачностью», т. е. могут быть объяснены на качественном уровне (Попов, 1987: с. 9). Не менее важными представляются следующие особенности экспертных систем (Попов, 1987: с. 11):

а) способность вести диалог о решаемой задаче на естественном, документальном или научном языке, удобном для эксперта, и, в частности, приобретать в процессе этого диалога новые знания;

б) способность следовать линии рассуждения, понятной эксперту;

в) способность объяснять эксперту ход своего рассуждения.

В тоже время следует помнить об ограничениях, существующих в настоящее время, в круге задач, решаемых с помощью ЭС. Он ограничивается составлением смысловых данных по входным данным, определением вероятных последствий наблюдаемых ситуаций, планировании и конструировании конфигурацией объектов по заданным параметрам, интерпретацией, предсказанием, диагностикой и рядом других задач, связанных с управлением самой ЭС. К тому же научная база в этой области знания находится на начальной стадии развития, разработки в ней требуют серьезных трудозатрат. В зависимости от сложности задачи время разработки занимает от 3 до 10 человеко/лет, а успех не всегда приходит в ее конце. Поэтому всегда возникает вопрос: как наиболее эффективно реализовать ЭС в конкретной области знания.

В качестве примера можно привести ЭС Роджера Грейса «Exspert systems for lithic analysis» [http://www.hf.uio.no/iakh/forskning/sarc/iakh/lithic/expsys.html].

Эта экспертная система «LITHAN» была разработана для классификации типологии и технологии каменных орудий (Рис. 7).

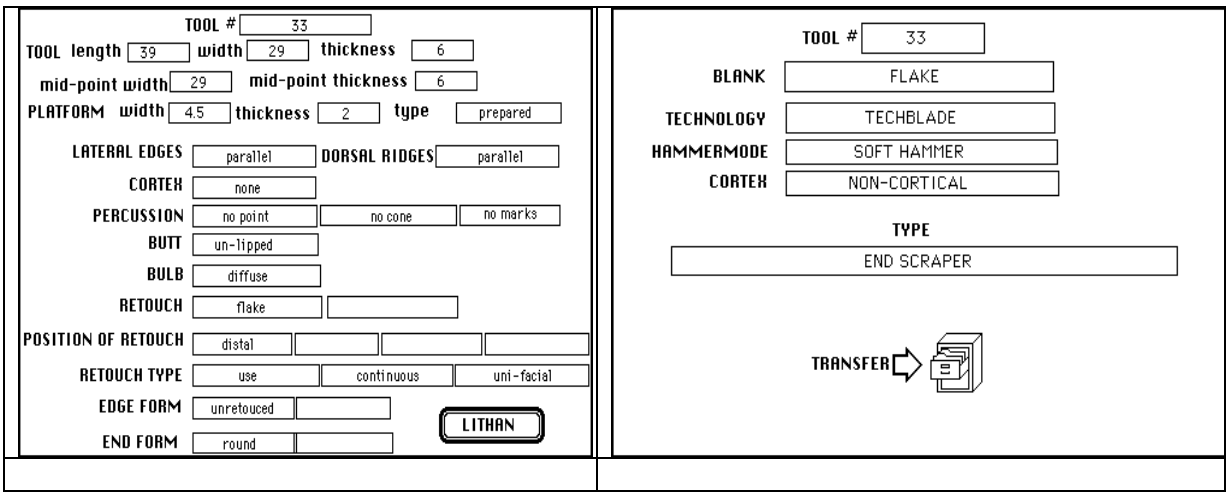


Рис. 7. Форма для ввода вывода результата информации TECH TYPE: if platform Thickness <5 and ButtType = "prepared" and Sides = "parallel" and Ridges = "parallel" then put "TECHBLADE" HAMMERMODE: if percussionCone = "no cone" and butt = "un-lipped" and bulb = "diffuse" then put "SOFT HAMMER" TYPE: if diff (length - width) > 0 and distalRetouch = "DISTAL" then put "END SCRAPER" General categories like endscraper are further subdivided by applying secondary rules.

1) if endForm = "ROUND" then put "END SCRAPER"

2) if endForm = "CARINATED" then put "CARINATED END SCRAPER"

Экспертные системы в отечественной археологии должны создаваться эволюционным путем, по мере возникновения естественных предпосылок и меры надобности в ее создании. В соответствии с этим и должна формироваться соответствующая архитектура ЭС.

2.4. Информационный поиск. Цель любого поиска заключается в потребности, необходимости или желании находить различные виды информации, способствующие получению лицом, осуществляющим поиск, нужных ему сведений, знаний и т.д. для повышения собственного профессионального, культурного и любого иного уровня; создания новой информации и формирования новых знаний и т.п. [Википедия].

Существуют различные толкования термина "поиск информации" или "информационный поиск".

Термин "**информационный поиск**" (англ. "information retrieval") ввёл американский математик К. Муэрс. Он заметил, что побудительной причиной такого поиска является *информационная потребность*, выраженная в форме информационного запроса. К объектам информационного поиска К. Муэрс отнес документы, сведения об их наличии и (или) местонахождении, фактографическую информацию. Решать проблемы фактографического поиска первыми стали представители библиотек. Они разработали средства информационного поиска, получившие название "*справочно-поисковый аппарат*" (каталоги, библиографические указатели и др.). В профессиональной отечественной печати данный термин используется с 1970-х годов. Библиотекари определяют "***информационный поиск***" как нахождение в информационном *массиве документов,* соответствующих *информационному запросу пользователей*.

С точки зрения использования компьютерной техники "***информационный поиск***" – совокупность логических и технических операций, имеющих конечной целью нахождение документов, сведений о них, фактов, данных, релевантных запросу потребителя.

В настоящее время главными проблемами доступа к знаниям, представленным в огромных объемах в сети Интернет, являются несистематизированность и слабая структурированность информации, ее распределенность по различным Интернет-сайтам, электронным библиотекам и архивам. Для решения данных проблем существует ряд подходов, одни из которых направлены непосредственно на унификацию или реорганизацию данных [Data Warehousing Technology], другие – ориентированы на унификацию средств доступа к ним [ANSI/NISO z39.50-2003].

Предлагаемый подход направлен на построение специализированных Интернет-порталов знаний [Боровикова, Загорулько. 2002: 76-82], ориентированных на работу с множеством разнородных ресурсов или источников данных по определенной научной тематике. Информационную основу портала знаний составляет онтология, применение которой позволяет сочетать принципы вышеперечисленных подходов: на ее основе обеспечивается как сведение ресурсов в единое информационное пространство, так и содержательный доступ к ним через Интернет.

Портал знаний предоставляет возможность поиска информации одновременно по различным аспектам научной деятельности (например, поиск информации об ученых, занимающихся определенной научной деятельностью). Использование онтологии для построения системы знаний портала позволяет не только целостно представить такие трудно формализуемые предметные области как гуманитарные науки, но и автоматизировать процесс сбора и накопления информации по выбранной тематике.

В данной работе описывается система знаний информационного Интернет-портала по археологии и этнографии [Андреева, Боровикова, Булгаков, Загорулько, Сидорова, Циркин, 2005].

Портал знаний представляет собой специализированную информационную систему, снабженную эргономичным пользовательским web-интерфейсом (рис.8).

С точки зрения пользователя, портал является тематическим Интернет-ресурсом, обеспечивающим возможность поиска и просмотра информации в рамках заданной предметной области (археология и этнография).

Как информационный ресурс портал:

•обеспечивает доступ к информации по различным аспектам и участникам научной деятельности, таким как: составляющие научной дисциплины (подразделы дисциплины, методы и техники исследования, используемые термины и понятия), персоналии исследователей, организации и т.п.;

•позволяет интегрировать в единое информационное пространство близкие по тематике ресурсы, представленные в Интернет (XML и HTML ресурсы, новостные каналы и т.п.);

•предоставляет средства поиска интересующей пользователя информации в рамках всего информационного пространства портала;

•обеспечивает информационную поддержку пользователей ресурса (например, анонсирование разного рода событий и мероприятий);

•поддерживает гибкий пользовательский интерфейс, позволяющий учитывать предпочтения пользователя по работе с ресурсом и предоставляемыми сервисами.

Рис. 8. Система знаний портала.

а). Компоненты системы знаний портала.

Система знаний портала представляет собой совокупность нескольких компонент, представленных на рис. 8.

Основу системы знаний составляет онтология и соотнесенное с ней описание соответствующих сетевых ресурсов. Онтология описывает структуру проблемной

области, и включает множество классов понятий и связывающих эти понятия отношений. Использование в качестве основы портала набора онтологий делает систему знаний портала легко расширяемой и настраиваемой – в нее могут интегрироваться как новые знания (например, о новых направлениях науки), так и новые типы информационных ресурсов.

Ядром системы знаний является онтология науки, которая фиксирует базовые содержательные структуры, используемые для построения онтологий более низкого уровня (онтологий предметных областей) и определяют структуру информационной базы портала. Онтология науки включает в себя две относительно независимые онтологии: онтологию научной деятельности и онтологию научного знания. *Онтология научной деятельности* включает общие классы понятий, относящиеся к организации научной деятельности, такие как Персона, Организация, Событие, Публикация, Информационный ресурс.

Онтология научного знания содержит следующие метапонятия, задающие структуры для описания рассматриваемой предметной области: Раздел науки (позволяет выделить в науке значимые разделы и подразделы), Метод исследования и Объект исследования (задают типизацию методов и объектов исследования и структуры для их описания), Научный результат (служит для типизации и описания результатов научной деятельности).

Онтология предметной области (ПО) отражает общие знания о предметной области, такие как иерархия классов понятий, семантические отношения на этих классах. Основой онтологии ПО для портала знаний по археологии и этнографии послужила предложенная в [Холюшкин, Гражданников, 2000: 58 с] и в последующем развитая Ю.П.Холюшкиным *системная классификация науки*, состоящая из фрагментов определенной универсальной структуры. Стандартный фрагмент данной классификации представляется в виде семантической карты, которая служит геометрической моделью фрагмента [Холюшкин, 2004: 99].

Экземпляры классов понятий и отношений, определенных в онтологии портала, образуют его *информационное наполнение.*

Онтология языка документов (словарь) представляет собой систему языковых средств выражения понятий онтологии портала. Лингвистическая информация представлена в словаре с помощью функциональных групп лексических единиц, выделенных классов понятий и набора дополнительных атрибутов, отражающих специфику выражений (фраз и отдельных слов), встречающихся в текстах документов данной тематики.

Исходными данными для системы знаний, характеризующими предметную область, являются *языковые ресурсы*, представленные в виде коллекции документов. Обеспечить автоматическое извлечение знаний из этих данных является главной задачей эксперта при наполнении и настройке системы знаний портала.

б). Онтология портала. Таким образом, разработанная нами система знаний включает такие онтологии как онтология науки, онтология ПО и онтология языка документов. Эти онтологии и образуют онтологию портала.

Для разработки, настройки и дальнейшей поддержки онтологии портала были разработаны: редактор онтологии и редактор системной классификации. Эти программные компоненты представляют собой пользовательские интерфейсы, позволяющие администратору или эксперту удаленно редактировать онтологию портала [Андреева, Сергеев, Холюшкин, 2005: 39-44].

в). Представление знаний. Онтология портала представляет собой иерархию понятий (или классов), связанных отношениями. Базовыми классами иерархии стали выделенные в процессе разработки онтологии и формально описанные 13 классов понятий (см. пп.3.2 и 3.3), связанные в иерархию с помощью отношения наследования. Различные свойства каждого понятия описываются на основе атрибутов понятий и ограничений, наложенных

на область их значений. При наследовании от родительского понятия передаются все отношения и атрибуты.

Отношения в онтологии портала являются бинарными (имеют два аргумента) и могут иметь собственные атрибуты. При разработке онтологии были выявлены следующие полезные для поиска информации типы отношений:

Отношения наследования;

Ассоциативные отношения, задаваемые пользователем. Наличие таких отношений позволяет осуществлять содержательный поиск;

Транзитивные отношения, к которым, в частности, относится отношение включения «часть-целое». При поиске информации, связанной отношениями такого типа, осуществляется транзитивное замыкание;

Ассоциативные отношения вида «класс-данные», позволяющие связывать конкретные экземпляры понятий с классом. Например:

Отношение: **Применяется-к-Классу-Объектов**

Аргумент 1: экземпляр класса – **Метод исследования.**

Аргумент 2: класс – **Объект исследования**

Частью онтологии портала является системная классификация, средства и методика построения которой описывается в [Андреева, Сергеев, Холюшкин, 2004: 39-44]. Поскольку системная классификация строится на принципах, отличающихся от применяемых в данном подходе, то для интеграции системной классификации в систему знаний портала введены отношения, позволяющие:

сопоставлять классу онтологии понятие системной классификации;

сопоставлять экземпляру класса онтологии понятие системной классификации;

осуществлять ассоциативную связь между экземпляром класса онтологии и понятием системной классификации.

г). Онтология научной деятельности. Онтология научной деятельности включает следующие классы понятий:

Персона. К этому классу относятся понятия, связанные с субъектами научной деятельности: исследователями, сотрудниками и членами организаций, исторически-значимыми персонажами и другими людьми. Атрибутами персоны являются: персональные данные, ученая степень, звание, направления научной деятельности, место проживания.

Организация. Понятия этого класса описывают различные организации, научные сообщества и ассоциации, институты, исследовательские группы и другие объединения. Атрибутами организации являются: название и место расположения.

Событие. В этот класс входят понятия, описывающие научно-организационную или научно-исследовательскую деятельность – научные мероприятия, конференции, исследовательские поездки, проекты, программы и т.п. К атрибутам события относятся: название, место проведения, дата начала, дата окончания, степень завершенности.

Научное мероприятие. Понятия этого класса описывают семинары, конференции, встречи, съезды, выставки и т.п. К атрибутам мероприятия, помимо наследуемых атрибутов события, относятся: язык, статус, дата основания, частота проведения.

Деятельность. Понятия класса Деятельность являются связующим звеном между методом и объектом исследования и полученным научным результатом. Класс описывает такие понятия, как проект, программа исследований.

Публикация. Этот класс служит для описания различного рода публикаций и материалов, представленных в печатном или электронном форматах (монографии, статьи, отчеты, труды конференций, периодические издания, фото- и видео-материалы и др.). К атрибутам публикации относятся: название, описание, дата публикации и язык публикации.

Географическое местоположение. Этот класс понятий позволяет описывать географическую и административно-территориальную локализацию объектов исследования, организаций и т.п. Атрибутами этого класса являются название местоположения и географический тип.

Информационный ресурс. Этот класс служит для описания информационных ресурсов, представленных в сети Интернет.

Понятия онтологии научной деятельности связаны как структурными («*общее-частное*», «*часть-целое*»), так и ассоциативными отношениями:

«быть автором» – используется для установления связи между персоной, являющейся автором публикации, и самой публикацией;

«состоять в» – связывает понятия организация и персона в случае, когда персона состоит в организации;

«быть участником» – связывает событие с персоной или организацией, участвующей в данном событии;

«*быть организатором»* – устанавливает связь между событием и персоной (или организацией), являющейся организатором события;

«научные труды» – задает связь между событием и публикациями, освещающими это событие;

«издан в» – связывает публикацию и организацию, являющуюся издателем;

«быть ресурсом» – связывает информационный ресурс с любым понятием онтологии.

д). Онтология научного знания. Онтология научного знания содержит следующие метапонятия:

Раздел науки. Этот класс отражает иерархию направлений научной деятельности. В частности, он может извлекаться из системной классификации науки.

Метод исследования. Данный класс служит для описания методов исследования, применяемых в археологии к определенному типу археологических объектов.

Объект исследования. Понятия этого класса задают типизацию объектов исследования и структуры для их описания. В археологии объектами исследования могут выступать как человек, или человеческое сообщество, так и различные объекты, созданные человеком в результате его деятельности: памятники, артефакты и т.п.

Научный результат. Понятия этого класса служат для описания результатов научной деятельности и их типизации. Например, выделяются следующие типы научных результатов: открытие, новый закон, теория, исторический факт и др. Обычно научные результаты находят свое отражение в публикациях.

Период. Основное назначение данного класса – датирование объектов исследования. Периоды образуют иерархию вложенности и исторического следования и задаются временным интервалом.

Понятия онтологии научного знания связаны следующими отношениями:

«научное направление» – связывает раздел науки с любым понятием онтологии научной деятельности;

«описывает» – связывает публикацию с любым понятием онтологии научного знания;

«часть деятельности» – связывает деятельность с объектами, методами и результатами исследований;

«применяется к классу объектов» – связывает метод и тип объектов исследования, к которым он применяется.

е) Онтология предметной области.

Онтология предметной области описывает археологию и этнографию в целом как раздел науки и включает формальное и неформальное описание понятий и отношений между ними. Эти понятия являются реализациями метапонятий онтологии научного знания и могут быть упорядочены в иерархию общее-частное и часть-целое. Так, например, Методам исследования в археологии соответствуют такие понятия, как методика раскопки, методика археологической разведки, а в качестве Объектов исследования выступают культуры, памятники, артефакты.

Классификация понятий онтологии по теоретическим разделам научной дисциплины, объектам исследования, применяемым методам исследования, временному и географическому признаку выполнялась на основе системной классификации науки.

Архитектурно, портал включает два описания предметной области по археологии и этнографии:

1. Системная классификация науки, предложенная Ю.П.Холюшкиным и Е.Д.Гражданниковым, в большей степени ориентирована на подготовленного пользователя. Классификация имеет многомерную сложную структуру, которая с одной стороны достаточно полно и профессионально описывает ПО, с другой стороны, она усложняет доступ к информационным ресурсам, например, при навигации по порталу.

2. Упрощенная классификация осуществляет быстрый доступ к выделенным понятиям системной классификации, что позволяет пользователю легко ориентироваться на сайте.

Взаимодействие упрощенной классификации с системной имеет свои особенности. Так, некоторые понятий системной классификации объявляются классами онтологии и образуют упрощенную иерархию наследования. Это могут быть: Одни классы в упрощенной классификации стали прямыми наследниками классов Метод Исследования и Научный результат.

Другие понятия системной классификации становятся экземплярами понятий онтологии портала. В частности, к таким понятиям относятся разделы науки.

Для начального наполнения словаря портала использовались языковые ресурсы, т.е. коллекции текстовых документов по археологии и этнографии, размеченные в соответствии с иерархией разделов науки портала.

Основная задача словаря – описать способ выражения фактов, извлекаемых из текстов на естественном языке. Специфика поставленной задачи определила требования, предъявляемые к словарю. Словарь должен содержать:

грамматическую информацию о терминах;

статистическую информацию; такая информация позволит использовать статистические методы классификации (рубрикации) для определения основной тематики ресурса (т.е. к какому разделу археологии относится данный ресурс) и его релевантности;

семантическую информацию, которая позволит связать элементы словаря с понятиями онтологии.

2.5. Способы аналитической обработки данных. Для того чтобы существующие хранилища данных способствовали принятию управленческих решений, информация должна быть представлена аналитику в нужной форме, то есть он должен иметь развитые инструменты доступа к данным хранилища и их обработки.

Очень часто информационно-аналитические системы, создаваемые в расчете на непосредственное использование лицами, принимающими решения, оказываются чрезвычайно просты в применении, но жестко ограничены в функциональности. Такие статические системы называются в литературе Информационными системами руководителя (ИСР), или Executive Information Systems (EIS). Они содержат в себе предопределенные множества запросов и, будучи достаточными для повседневного обзора, неспособны ответить на все вопросы к имеющимся данным, которые могут возникнуть при принятии решений. Результатом работы такой системы, как правило, являются многостраничные отчеты, после тщательного изучения которых у аналитика появляется новая серия вопросов.

Однако каждый новый запрос, непредусмотренный при проектировании такой системы, должен быть сначала формально описан, закодирован программистом и только затем выполнен. Время ожидания в таком случае может составлять часы и дни, что не всегда приемлемо. Таким образом, внешняя простота статических СППР, за которую активно борется большинство заказчиков информационно-аналитических систем, оборачивается катастрофической потерей гибкости.

2.6. Хранилища данных. Ключевым фактором деятельности научных структур является оперативное принятие эффективных решений. Однако естественное стремление усовершенствовать процессы принятия решений нередко наталкивается на труднопреодолимое препятствие — огромный объем и высокая сложность данных, содержащихся в разнообразных оперативных и других информационных системах.

Сделать такую информацию доступной для анализа — одна из наиболее серьезных задач, стоящих сегодня перед профессионалами в области информационных технологий. Большие объемы накопленных данных постоянно приходится модифицировать из-за быстрой смены аппаратного и программного обеспечения БД, при этом неизбежны потери и искажение информации.

Одним из средств, для преодоления подобных трудностей является создание информационных хранилищ археологических данных, доступ к которым не будет сильно зависеть от изменения данных во времени и от используемого программного обеспечения. Другой подход ориентирован на сжатие больших объемов данных путем нахождения некоторых общих закономерностей (знаний) в накопленной информации. Оба направления актуальны с практической точки зрения.

Второй подход более интересен для специалистов в области ИИ, так как связан с решением проблемы приобретения новых знаний. Однако, следует заметить, что наиболее плодотворным является сочетание обоих направлений. Наличие хранилища данных — необходимое условие для успешного проведения всего процесса KDD.

Хранилищем археологических данных называют интегрированное предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных, используемых для поддержки процесса принятия решений. Предметная ориентация означает, что археологические данные объединены в типы, фации, культуры и хранятся в соответствии с теми областями, которые они описывают, а не в соответствии с приложениями, которые их используют.

Такой принцип хранения гарантирует, что полевые и научные отчеты археологов, сгенерированные различными участниками процесса их создания, будут опираться на одну и ту же совокупность данных. Привязанность ко времени означает, что хранилище можно рассматривать как собрание исторических данных, т.е. конкретные значения данных однозначно связаны с определенными моментами времени. Атрибут времени всегда явно присутствует в структурах хранилищ данных. Данные, занесенные в хранилище, уже не изменяются в отличие от оперативных систем, где присутствуют только последние, постоянно изменяемые версии археологических данных. Для хранилищ данных характерны операции добавления, а не модификации данных.

Современные средства администрирования хранилищ данных обеспечивают эффективное взаимодействие с программным инструментарием DM и KDD. В общем случае зависимости, выявляемые в базах данных, могут быть представлены правилами, гипотезами, моделями нейронных сетей и т.п.

Интеллектуальные средства извлечения информации позволяют почерпнуть из БД более глубокие сведения, чем традиционные системы оперативной обработки транзакций (OLTP — On-Line Transaction Processing) и оперативной аналитической обработки (OLAP). Выведенные из данных закономерности и правила можно применять для описания существующих отношений и закономерностей археологических данных, и прогнозирования их последствий.

Извлечение знаний из БД является одной из разновидностей машинного обучения, специфика которой заключается в том, что реальные БД, как правило, проектируются без учета потребностей извлечения знаний и содержат ошибки. Современные подходы к решению этой задачи связаны с построением хранилища данных (data warehouse), позволяющего "высвободить" информацию из жестких рамок оперативных систем и лучше осознать проблемы реальной деятельности. Хранилище данных — это интегрированный накопитель информации, собранной из других систем, на основе которого строятся процессы принятия решений и анализа данных. Несмотря на то, что хранилища данных бывают различных типов и могут опираться на разные методологии, и даже философии, построения, все они имеют следующие общие признаки:

- Информация в хранилище данных организовывается вокруг базовых понятий, используемых в деятельности научных подразделений

- "Сырые" данные собираются из неинтегрированных оперативных и унаследованных приложений, очищаются от ошибок, затем агрегируются и представляются в виде, понятном конечным пользователям.

На основании откликов пользователей, а также закономерностей, обнаруженных с помощью соответствующих методов, архитектура хранилища данных со временем претерпевает изменения – то есть процесс создания хранилища является итеративным.

Иными словами, хранилище данных ориентировано на ключевые понятия (например, цели операций), а не на процессы (например, оформление какой-либо документации), и содержит всю существенную информацию, относящуюся к этим понятиям, которая собрана из различных обрабатывающих систем. Эта информация собирается и представляется за согласованные периоды времени и не подвержена оперативным изменениям [Дюк, 2002].

Одними из основных новых возможностей, появляющихся в результате построения хранилищ данных являются следующие:

- применение средств поддержки принятия решений на основе технологий интеллектуального анализа данных (Data Mining — добыча данных, knowledge discovery in databases — обнаружение знаний в базах данных), включающих методы логического вывода, нейронных сетей и нейрокомпьютеров, и др.
- использование средств, повышающих простоту поиска информации и обращения к конкретным прикладным функциям, например, гипертекстовым, естественного языка, речевого ввода.

2.7. Нейросети. Нейронные сети ищут более сложные (нелинейные) функции от тех же переменных, и потому обладают большей разрешающей способностью. Если же и нейронные сети не в состоянии правильно распознать классы, то, скорее всего, на основании использованных переменных в принципе невозможно построить представленную классификацию. В таком случае можно будет говорить о том, что авторы такой классификации либо неявно использовали дополнительную информацию, не учтенную при анализе, либо же где-то допустили ошибку. Таким образом, дискриминантный анализ и нейронные сети могут быть использованы в качестве своеобразного “детектора лжи”, осуществляющего процедуру фальсификации научных гипотез, сформулированных в виде классификаций эмпирических объектов.

3. Построение модели интеллектуального анализа данных является частью более масштабного процесса, в который входят все задачи, от формулировки вопросов относительно данных и создания модели для ответов на эти вопросы до развертывания модели в рабочей среде. Этот процесс можно представить как последовательность следующих шести базовых шагов.

1. Постановка задачи
2. Подготовка данных
3. Просмотр данных
4. Построение моделей
5. Исследование и проверка моделей
6. Развертывание и обновление моделей

На следующей диаграмме представлены связи между всеми шагами процесса и технологии Microsoft SQL Server, которые можно использовать для выполнения каждого шага.

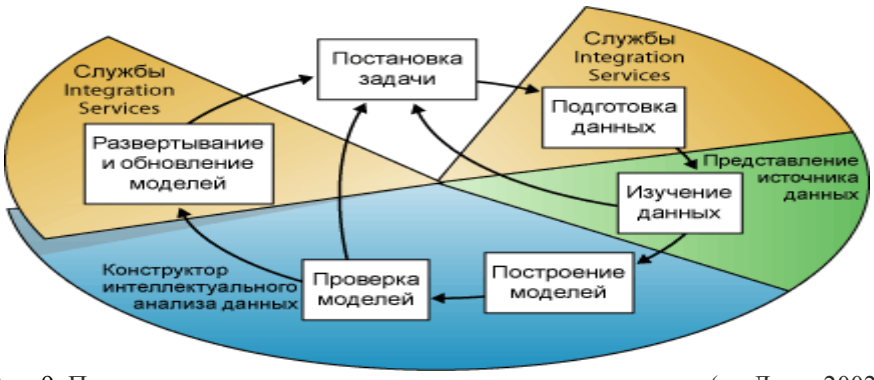


Рис. 9. Построение модели интеллектуального анализа данных (по Дюк, 2002)

3.1.Постановка задачи.

Первым шагом процесса интеллектуального анализа данных, как видно из диаграммы, является четкое определение проблемы и рассмотрение способов использования данных для решения проблемы. «Действительно от характера задач зависит выбор исследуемых археологических материалов, способ их видения, классификация, комментирование и, наконец, даже мера ценности результатов, о чем можно судить, только соотнеся их с целью и задачами построения» [Гарден, 1983: 209].

Этот шаг включает анализ требований, определение области проблемы, метрик, по которым будет выполняться оценка модели, а также определение задач для проекта интеллектуального анализа данных.

Все шесть шагов располагаются один за другим, однако, при следующих условиях:

а) Постановка задач одновременно или последовательно оказывает влияние на все остальные звенья исследования, включая оценку достоверности полученных результатов;

б) Если создается компиляция, процесс останавливается на шагах «подготовка данных» и «просмотр данных»;

в) Блок построения моделей играет двоякую роль: как динамическую связь между звеньями в пределах одного построения, либо между разными, альтернативными или дополняющими дуг друга построениями. В приблизительном виде это напоминает аналогичные построения Ж-К.Гардена [Гарден, 1983: 211, рис.28].

В монографии "О профессии исследователя в точных науках" Е.М. Регирер, так формулирует сущность научного метода: «Строить предположения о ближайших причинах – гипотезы – и проверять на соответствие фактам те выводы, которые из этих гипотез вытекают – это и есть научный метод… Во всех науках имелись и имеются ложные гипотезы, ошибочные предположения. Ни одна наука не становится от этого ненаучной, если только она ведет систематическую проверку и изучение выводов принятой гипотезы» [Регирер, 1966: 81].

Таким образом, логистическое требование, предъявляемое к первому базовому шагу схемы, состоит в перечислении возможных альтернативных задач и в обосновании выбора той или иной проблемы.

Именно из этого логистического требования вытекает вся совокупность приемов (частных методов) и этапов исследования, которую Л.С. Клейн зовёт методикой исследования [Клейн, 2005]. Ведь чтобы осуществить такую группировку фактов, из которой можно получать обобщения и предположения о закономерностях, причинах и зависимостях, надо затем извлекать следствия из этих гипотез и проводить систематическую проверку этих следствий на всё новых наблюдаемых фактах,

Именно из этого главного научного метода вытекает вся совокупность приемов (частных методов) и этапов исследования, которую мы зовём методикой исследования. Ведь чтобы осуществить такую группировку фактов, из которой можно получать обобщения и предположения о закономерностях, причинах и зависимостях, чтобы затем извлекать следствия из этих гипотез (ожидания) и проводить систематическую проверку этих следствий на всё новых наблюдаемых фактах [Клейн, 2005].

Эти задачи можно сформулировать в виде следующих вопросов:

- Что необходимо найти? Какие типы связей необходимо выявить?
- Отражает ли поставленная задача логические правила или процессы в прошлом?
- Надо ли делать прогнозы на основании модели интеллектуального анализа данных или просто найти содержательные закономерности и взаимосвязи?
- Какой результат или атрибут необходимо спрогнозировать?
- Какие виды данных нужно иметь и, какого рода информация находится в каждом столбце таблицы?
- Если существует несколько таблиц данных, то, как они связаны между собой?
- Нужно ли выполнять очистку данных от «шума», статистическую обработку, чтобы данные стали применимыми?
- Каким образом распределяются данные? Дают ли данные точное представление об исторических процессах далёкого прошлого?

Чтобы ответить на эти вопросы, возможно, потребуется исследовать уровень доступности данных, изучить возможности доступа к этим данным. Если данные не доступны, то может возникнуть необходимость в изменении самого проекта.

Также необходимо рассмотреть способы для учета результатов построения модели в ключевых показателях эффективности, которые используются для оценки ведения научного проекта.

3.2. Подготовка данных. Стереотипным требованием у большинства археологов на втором шаге процесса интеллектуального анализа данных является наличие исчерпывающего объема археологического материала, выраженное в той или иной наивной форме («в основном», «все», «возможно большее число»), имеет тенденцию превратиться в риторический оборот [Гарден, 1983: 210]. А этому требованию противоречит возможность пополнения коллекций новыми материалами. Что касается экспликаций, то они ориентируются на определенные источники точной исторической и археологической информации. Поэтому в этих источниках должны учитываться только материалы пригодные для исследования [Гарден, 1983: 210-211].

Так "Новые археологи" перешли к анализу репрезентативных выборок. Тогда, по мнению Л.С.Клейна, нужно было придерживаться строгих правил, как делать выборки, чтобы они были, в самом деле, репрезентативными. Но идея удовлетвориться выборками (примеры – участки для сборов по системе случайных цифр или в шахматном порядке даны у Уотсон с соавторами), да еще вслепую (а это обязательное условие репрезентативности) встретила непонимание и непринятие археологов традиционной ориентации [Клейн, 2005].

Однако археологические данные могут храниться не только на стеллажах непосредственного исследователя, но и в разных базах данных музеев и НИИ, представленных в различных форматах хранения или содержать такие ошибки согласования, как неверные или отсутствующие записи. Поэтому процедура очистки данных — это не только удаление недопустимых данных или интерполяция отсутствующих значений, но и поиск в данных скрытых зависимостей, определение источников самых точных данных и подбор столбцов, которые больше всего подходят для использования в анализе.

Неполные данные, ошибочные данные и входные параметры, которые выглядят как независимые, но на самом деле имеют прочную взаимосвязь, могут непредвиденным образом повлиять на результаты построенной модели. Поэтому искусство отбора археологического материала в этом случае состоит в том, чтобы определить, исходя из задач исследования, такую методику отбора, которая позволяет свести к минимуму число необходимых для решения данной задачи археологических памятников [Гарден, 1983: 211]. Однако на пути реализации этой творческой работы в археологической практике

имеются существенные трудности, обусловленные информационными проблемами археологии. Все они в той или иной мере связаны со сбором и отбором наблюдений и фактов, их анализом и интерпретацией. Среди этих проблем следует отметить особо значимые:

Неполнота и фрагментарность археологической информации, объясняемая как дискретностью самих археологических данных, так и ограниченностью их использования. В первом случае неполнота зависит от степени сохранности и исследованности археологического памятника, а во втором определяется недостаточностью списка признаков, используемых в исследовательских процедурах. Очевидно, такая информация не дает адекватного представления о действительном состоянии исследуемой проблемы и, особенно, в тех случаях, когда остается неизвестной та ее часть, которая учтена и использована в каждой конкретной процедуре. Наиболее остро эта проблема возникает, когда археолог производит слабо обоснованную селекцию информации. Зачастую подобная селекция производится при отсутствии концепции у исследователя, а это в свою очередь не позволяет сделать выбор из большого ряда потенциальных характеристик, присутствующих в массиве изучаемого материала. Возникает задача восполнения данных, решение которой – самостоятельная проблема.

Несопоставимость данных. Одна из форм несопоставимости связана с использованием различных классификационных построений. Примером может служить использование процентных соотношений. На некоторые ошибки, связанные с применением процентных соотношений, указывал В.А. Ранов (отсутствие в публикациях сведений о базисных цифрах, от которых производились вычисления, проведение процентных сопоставлений, взятых от различных базисных цифр [Гинзбург, Гореншстейн, Ранов, 1980: 8]. Все это выдвигает более сложные и нестандартные задачи перед исследователями, так как одним приемом, одной процедурой их решить весьма затруднительно. Решение проблемы, требует, как правило, проведение комплексных исследований, в которых каждый аспект изучается на основе наполнения некоторой однородной формы данных.

Неадекватность применяемых процедур статистического анализа данных поставленной задаче. Часто археологи применяют статистические методы, не потому, что он необходим, а потому, что его знают. Отсюда возникают заблуждения относительно того, что применяемый метод дает сразу ответ положительный или отрицательный на поставленную задачу.

Устойчивость исходных и выделенных структур. В предлагаемом нами контексте, эта проблема в палеолитоведении ранее не рассматривалась. Что касается традиционной археологической парадигмы, то в ней предполагается постепенность культурных изменений, а в случае, когда они не наблюдаются, разрывы в структурных культурных образованиях объясняются сменой населения. При таком подходе обычно высказываются следующие предположения:

а) коллекции, относящиеся к одному и тому же времени, должны быть примерно одинаковы;

б) различия между коллекциями фиксируют направленные изменения [Deetz, 1967, p. 26-37], отражающие "развитие" форм артефактов из предшествующих форм [Clarke, 1968, p. 131-185]. Указанные взгляды покоятся на модели культуры, учитывающей лишь три показателя изменений в устойчивости структур: миграция, изобретение и диффузия, как процессы культурной истории [Trigger, 1968, p. 26-31]. При этом остаются в стороне другие факторы, связанные с контекстом памятника, сырьем, репрезентативностью используемых выборок, субъективностью, вносимой исследователем в источник исследования (специальная подборка материала и др.). Все эти факторы могут оказывать немаловажное влияние на устойчивость выделяемых археологических структур.

Внесение ошибок в археологические данные:

а) погрешности арифметических подсчетов в табличных данных;

б) недочеты измерений (устранение ошибок измерения углов, форм);

в) ошибки в определении типов артефактов (неправильные дефиниции). В свое время Д. Кларк указывал на возникновение опасности, что альтернативное или противоречащее определение типов артефактов коренным образом изменяет подробно рассмотренные процентные соотношения и соответственно их смысловую интерпретацию [Clarke, 1968, p. 188]. На Ближнем Востоке пытаются решить эту проблему преодоления ошибок, внутренне присущих типологическому анализу, на основе тип-листов Ф. Борда, путем использования в археологических штудиях только тех типов ретушированных орудий, которые всегда могут быть идентифицированы и отделены от других орудий любым исследователем. Для иллюстрации можно привести цитату из статьи А. Маркса: «...каждый, работавший в Леванте после Д. Гаррод, установил для себя разницу между продольными скреблами и концевыми скребками, между ножами с обушком и плоскоретушированными скреблами. Таким образом, эти группы орудий пригодны для нашего исследования. С другой стороны, ракле и псевдолеваллуазские острия, например, не всегда определимы и до сих пор еще не всеми единообразно включены в типологические списки комплексов. То же самое можно сказать о выемчатых орудиях и мустьерских транше» [Marks, 1992];

в) методические просчеты;

г) неверно поставленные задачи и т.д.

Здесь следует не только выделить такие этапы экспертизы подлинности археологического источника, как проверка сохранности памятника, степень его изученности, информативности и субъективизм выборки, произведенной археологом, но и учитывать проблему дальнейшей судьбы той или иной коллекции после раскопок

Поэтому перед началом построения моделей интеллектуального анализа данных следует выявить такие проблемы и определить, как их устранить. Во время интеллектуального анализа данных выполняется работа с большими наборами данных и нет возможности проверить каждую транзакцию на предмет качества данных. Поэтому иногда может потребоваться использовать некую форму профилирования данных и средства автоматической очистки и фильтрации данных, например средства, содержащиеся в Службы Integration Services, Службы Microsoft SQL Server 2012 Master Data Services или Службы SQL Server Data Quality Services, чтобы исследовать данные и определить несоответствия.

Важно заметить, что данные, используемые для интеллектуального анализа, не обязательно хранить в кубе аналитической обработки в сети (OLAP) или в реляционной базе данных, хотя оба эти типа объектов можно использовать в качестве источника данных. Интеллектуальный анализ данных можно проводить с помощью любого источника, определенного как источник данных служб Службы Analysis Services. Сюда могут относиться текстовые файлы, книги Excel или данные из других внешних распределенных хранилищ.

3.3. Просмотр данных. Третьим шагом процесса интеллектуального анализа данных является просмотр подготовленных данных.

Для принятия правильных решений при создании моделей интеллектуального анализа данных необходимо понимать данные. Методы исследования данных включают в себя расчет минимальных и максимальных значений, вычисление средневероятного и стандартного отклонения и изучение распределения данных. Например, по максимальному, минимальному и среднему значениям можно заключить, что выборка данных не является репрезентативной для имеющихся процессов, и поэтому необходимо получить более сбалансированные данные или изменить предположения, лежащие в основе ожидаемых результатов. Стандартное отклонение и другие характеристики распределения могут сообщить полезные сведения о стабильности и точности результатов. Большая величина стандартного отклонения может свидетельствовать о том, что добавление новых данных поможет усовершенствовать модель. Данные, которые

сильно отклоняются от стандартного распределения, могут оказаться искаженными или представлять точную картину реальной проблемы, которая делает сложным подбор соответствующей модели для данных.

Изучение данных в свете собственных представлений о проблеме может привести к выводу о наличии ошибок в наборе данных, и затем можно выработать стратегию для устранения проблем или получить более глубокое представление о моделях археологических данных.

Для просмотра доступных источников данных и определения их доступности для интеллектуального анализа данных можно использовать средства Master Data Services. Для анализа распределения данных и устранения проблем, таких как неверные или отсутствующие данные, можно воспользоваться таким средством, как Службы SQL Server Data Quality Services, или профилировщиком данных в службах Integration Services.

После определения источников их следует объединить в представлении источников данных с помощью конструктора представлений источников данных в SQL Server Data Tools. Конструктор содержит также ряд средств, которые можно использовать для просмотра данных и определения того, подходят ли они для создания модели. Обратите внимание, что во время создания модели службы Службы Analysis Services автоматически создают статистические сводки по данным, содержащимся в модели, и эти сводки можно запрашивать для использования в отчетах или при дальнейшем анализе.

3.4. Построение моделей. В методе моделирования исследуется не сам объект-оригинал, а замещающий его аналог (модель). Замена объекта-оригинала на модель позволяет получить следующие полезные эффекты. Во-первых, модели дешевле и доступнее оригиналов, следовательно, уменьшаются расходы на исследование. Благодаря этому при одних и тех же финансовых затратах методом моделирования можно провести гораздо больше наблюдений, чем при использовании традиционных научных методов. Во-вторых, модель гораздо компактнее, чем оригинал, что особенно наглядно проявляется в математических и вообще знаковых моделях. Благодаря компактности модель удобна для изучения и, что самое главное, обладает свойством конструктивности, проявляющемся в том, что она может играть роль конструктивного элемента блока, «кирпичика» в сложных научных построениях [Гражданников, Холюшкин, 1990].

Из таких компактных блоков можно строить чрезвычайно сложные научные теории (системные), которые при традиционных методах практически невозможны. В-третьих, можно проводить такие преобразования моделей, которым нельзя подвергать оригиналы. Обычно уже при замене объекта-оригинала на модель осуществляется какая-либо существенная трансформация, которую сознательно допускает исследователь. Это может быть уменьшение (или увеличение) размеров, деформация определенного вида, упрощение и т.д.

После построения модели с ней можно делать то, что в принципе недопустимо применительно к оригиналу: например, отправить модель в будущее, подвергнуть ее разрушительным нагрузкам и вообще проделать любую процедуру, которую придумает исследователь. Эта возможность неограниченных преобразований является самой ценной, наиболее фундаментальной и информативной стороной метода моделирования. Подвергая модель всевозможным преобразованиям, исследователь, с одной стороны, получает подробное и детальное описание существенных свойств объекта, а с другой стороны, находит способы воздействия, обеспечивающие достижение заданного состояния объекта или проявление какого-либо полезного эффекта, который можно использовать на практике.

Все исследовательские задачи, связанные с изучением объектов, можно разделить на два вида: задачи идентификации и квантификации.

Задача идентификации состоит в разработке процедуры, на основе которой исследователь может надежно распознавать отдельные объекты и фиксировать его

качественные характеристики (признаки). Признак представляет собой вид научных сведений об объекте, задаваемый словесно или наглядно.

Задача квантификации состоит в нахождении способа определения количественных характеристик. Количественная характеристика (показатель) представляет собой вид научных сведений об объекте, который может быть выражен числом или множеством.

Признак (качественная характеристика) и показатель (количественная характеристика) – это пара диалектически взаимосвязанных категорий. Они неразрывно связаны друг с другом, резкой грани между ними нет.

В соответствии с двумя видами исследовательских задач можно выделить два основных типа моделей: качественные (дескриптивные) и количественные (квантитативные) модели. Дескриптивные модели являются аналогами объектов-оригиналов по качественным характеристикам (признакам), квантитативные (количественные) модели – по количественным характеристикам (показателям). Строго говоря, ни одну из конкретных моделей нельзя отнести лишь к одному из этих типов потому, что в любую из них входят и качественные, и количественные характеристики, и признаки, и показатели. Используя это разделение моделей на два типа, можно, с одной стороны, выделить основной аспект модели, т.е. установить, для какой цели предназначена модель, для определения качественных или количественных характеристик, и в зависимости от этого отнести ее к соответствующему типу. С другой стороны, любую модель можно представить состоящей из двух моделей, одна из которых является дескриптивной (качественной), а вторая - квантитативной (количественной).

Нередко утверждают, что метод моделирования отражает лишь количественную сторону явлений. Между тем это совсем не так. Всегда учитываются и качество, и количество, надо лишь уметь правильно выделять и анализировать эти две стороны действительности.

В каждой модели следует различать содержательную и формальную стороны. Содержательная сторона модели связана с теми конкретными объектами, которые отражаются в данной модели. Формальная сторона модели связана с расположением этих объектов относительно друг друга и взаимосвязями между ними.

Совокупность взаимосвязей между элементами определенного целостного объекта (т.е. внутри системы) называется структурой. Соответственно можно говорить о содержательной и структурной моделях одного и того же объекта.

Широко распространенным типом содержательных моделей являются классификации, т.е. систематизированные перечни конкретных категорий и классов. Универсальная десятичная классификация (УДК), когда она представлена названиями рубрик, разделов и подразделов, – это содержательная модель, а перечень индексов или их изображение в виде графа – структурная модель.

Возможны два основных типа моделей классификационного фрагмента – знаковые и геометрические модели. В данном случае знаковая модель – это набор букв и цифр (индексная модель) или только цифр (цифровая модель); геометрическая модель – прямоугольник, разделенный на прямоугольные площадки (чертежная модель) (рис. 2), или набор координат этих площадок (координатная модель). Чертежную модель, на которой приведены названия понятий или их символические обозначения, будем называть семантической картой.

Что касается Западной археологии, то там понятие модель стало использоваться как чисто общая концепция для того, чтобы понять постоянно меняющиеся взаимоотношения между человеческими культурами и окружающей средой [Watson et al, 1984].

На основе разработок преимущественно американских антропологов было разработано ряд моделей:

Нормативная модель обязана своим возникновением Ф. Боасу. Это была первая концепция культуры, в основе которой возникло представление, что все поведение человека делится на модели, предусматривающие ряд правил и норм поведения.

Е. Радклиф-Браун пошел дальше Боаса, утверждая, что природа любого общества может быть понята только через изучение сети сложных взаимоотношений и механизмов, предназначенных для удовлетворения как социальных нужд, так и потребностей выживания всего общества в целом. Согласно такой модели каждый компонент культурной системы имеет специфическую функцию, будь это технология производства, или способы аграрных приемов, или правила жизни в браке. Процессуалисты внесли также свой вклад в созданием поведенческих моделей. Согласно их представлениям процесс представляет собой структурированную последовательность событий, которые ведут от одного состояния к другому. Занятия археологией тоже являются процессом. В этот процесс входят: планирование исследований, формулировка гипотез, вытекающих из предыдущих следований, сбор и интерпретация новых данных для проверки ранее выдвинутых гипотез.

Четвертым шагом процесса интеллектуального анализа данных, как видно из диаграммы выше, является построение моделей интеллектуального анализа данных. Знания, полученные при выполнении шага «Просмотр данных», помогают создать модели.

Пользователь определяет столбцы данных, которые должны быть использованы, путем создания структуры интеллектуального анализа данных. Структура интеллектуального анализа связана с источником данных, но не содержит никаких данных до обработки. Во время обработки структуры интеллектуального анализа создают статистические данные, которые могут использоваться в анализе. Эти данные могут использоваться любой моделью интеллектуального анализа данных, которая основана на этой структуре. Модель интеллектуального анализа данных перед обработкой структуры и модели является просто контейнером, который задает столбцы, используемые для входных данных, прогнозируемый атрибут и параметры, управляющие алгоритмом обработки данных. Такую обработку модели часто называют *обучением*. Обучение обозначает процесс применения некоторого математического алгоритма к данным в структуре с целью выявить закономерности. Закономерности, обнаруженные в процессе обучения, зависят от выбора обучающих данных, выбранного алгоритма и его конфигурации. SQL Server 2014 содержит множество различных алгоритмов, каждый из которых предназначен для задач различных типов и создает модель, отличную от других.

II WEB- СИСТЕМА ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

2.1. Обзор веб-системы интеллектуального анализа данных.

Стремительно развивающаяся веб-индустрия диктует современному рынку свои условия. Последнее непосредственно сказывается на привычках, потребностях и запросах современного общества, равно как и научного сообщества. Поэтому было решено разработать удобную для пользователя, и в то же время функциональную веб-систему решения задач археологии [Холюшкин, Витяев, Костин, 2013].

Со временем стало понятно, что такая система обладает большим потенциалом и не ограничивается задачами лишь одной археологии, она может служить в качестве системы интеллектуального анализа данных в целом. Цели, которые были успешно достигнуты при разработке этой системы можно обозначить следующим образом:

1. Простой, интуитивно понятный пользовательский интерфейс;
2. Удобная организация имеющейся в распоряжении пользователя информации;
3. Доступ к большому числу существующих статистических методов для решения задач анализа данных, в том числе очень специфических, например [Витяев, Москвитин, 1993];
4. Возможность самостоятельно программировать исследовательские стратегии с помощью визуального редактора, не прибегая к написанию программного кода;
5. Гибкая, дополняемая архитектура, возможность доработки системы как на уровне исходного серверного кода (PHP и JavaScript), так на уровне дополнения библиотеки методов вычислительными модулями сторонними разработчиками.

Прообраз системы, описанный в [Холюшкин, Витяев, Костин, 2013], оброс множеством особенностей, и таким образом превратился в готовый исследовательский инструмент, решающий задачи анализа данных произвольной природы. Чтобы внести ясность в вопрос о том, каким же образом функционирует разработанная нами система, мы ненадолго погрузимся во внутреннюю архитектуру программного комплекса. Затем мы рассмотрим видимую её часть – UI, так называемый пользовательский интерфейс.

2.2. Внутренняя архитектура сервиса.

Главенствующей концепцией, в рамках которой велись работы по реализации веб-сервиса, является так называемая схема Model-View-Controller (рис.10).

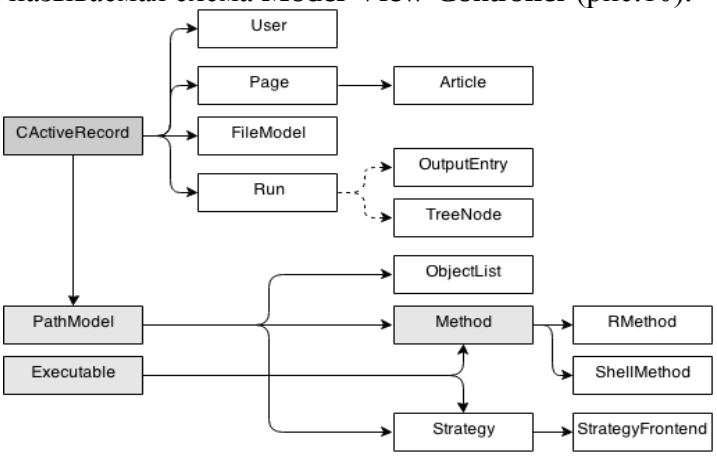


Рис. 10. Диаграмма моделей

Основные её тезисы берут начало из исторических особенностей развития интернета (роль баз данных, архитектура запрос-ответ и HTML представление):

Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.

Представление отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.

Контроллер обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

CActiveRecord – базовый объект, реализующий логику взаимодействия с базой данных;

Элемент архива (PathModel) – объект, предоставляющий логику взаимодействия с файловой системой сервера. Также позволяет хранить и управлять объектами в файловом менеджере;

Статья(Article), новость (Page) – структуры, описывающая информационные страницы (разделы теории, новости и главная);

Пользователь (User) –представляет зарегистрированного в системе пользователя;

Данные (ObjectList) – список объектов в виде таблицы объект-признак, со своими именами и типами данных. Каждый список хранится в отдельном файле на сервере;

Метод (Method), Р-метод (RMethod), исполняемый модуль (ShellMethod) – программы, осуществляющие анализ и преобразование данных и выступающие в качестве «кирпичика» в построении исследовательских стратегий. Р-метод является скриптом на языке R, в то время как исполняемый модуль может быть произвольной исполняемой программой, оформленной по особым правилам;

Стратегия (Strategy) – исследовательская стратегия, позволяет пользователю комбинировать различные методы, получая значимый научный результат. Стратегии могут быть нелинейными и в общем случае представляют собой граф в вершинах которого располагаются методы. Стратегии и методы в дальнейшем будут именоваться как исполняемые элементы;

Вычислительный узел (MethodNode) – вспомогательная сущность, отвечающая за организацию графа исполнения. Узлы используются на этапе исполнения стратегии;

Результат вычислений (OutputEntry) – содержит информацию, полученную в результате выполнения одного из методов. Помимо непосредственно данных включает в себя тип этих данных, а также классы всех подструктур данных, если он является комплексным. Подробнее структура данных описана в разделе «Типизация»;

Запуск (Run)–сведения о запуске исполняемого элемента, включая время запуска, время завершения, имя исполняемого элемента, параметры запуска, заданные данные и т.д. Ниже представлена схема организации компонентов системы с помощью контроллеров. Каждый Controller – это раздел сайта, каждый Action – доступное пользователю действие (рис.11).

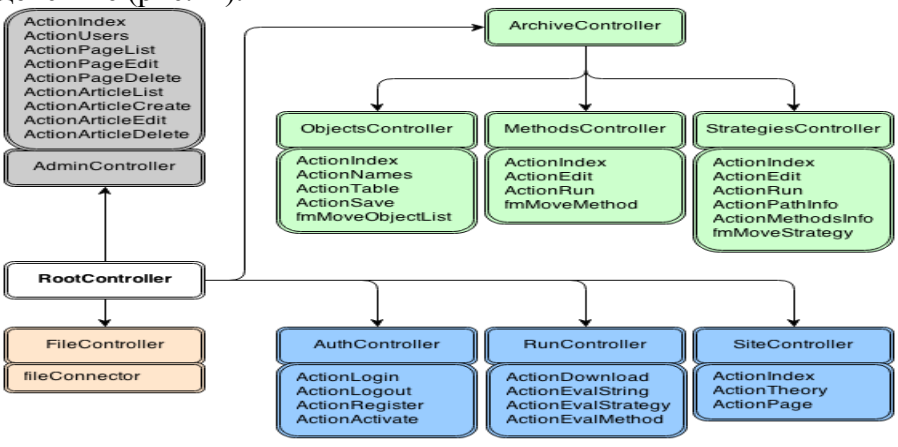


Рис. 11. Схема организации контроллеров.

Контроллеры из группы ArchiveController предоставляют доступ к файловым архивам для методов, данных и стратегий, а также интерфейс исполнения вычислительных модулей (методов или стратегий) и сводную информацию о запрашиваемой сущности по

требованию в режиме AJAX-запросов. Работа с файловой системой сервера осуществляется через контроллер-помощник FileController, связывающий браузер пользователя и серверную файловую систему воедино, также с помощью протокола AJAX.

AdminController отвечает за администрирование веб-сервиса. С помощью предоставляемых им действий, пользователь обладающий правами доступа уровня «редактор», создает и удаляет информационные страницы, редактирует новости, а также управляет пользователями (например, назначает им права доступа или запрещает участие в проекте в случае совершение неправомерных или злонамеренных действий).

AuthController позволяет пользователям выступать под конкретными учетными записями. Функционал – регистрация и авторизация пользователя.

Самый интересный компонент системы – RunController. Этот раздел функционала запускает на исполнение различные вычислительные модули, информирует пользователя об изменениях в статусах заданий, запущенных на исполнение, а также формирует и выдает ему результат счета. На данный момент пользователю доступно исполнение методов (Method), исполнение стратегий (Strategy) и запрос результатов в виде csv-файла с данными.

За отображение информационных страниц сайта (новостей, теории и главной) ответственен SiteController. Он также осуществляет постраничную разбивку информации, фильтрацию по интересующим тегам и поиск по сайту.

2.3. Пользовательский интерфейс веб-системы.

Перейдем теперь непосредственно к интерфейсу.

Навигация построена на сквозном боковом меню, всегда находящемся перед пользователем. Большой размер позволяет акцентировать внимание пользователя на важном элементе интерфейса и обеспечивает простоту переключения между следующими разделами (рис.12):

Домашняя страница – раздел с приветствием, новостями и актуальными статьями;

Теории – список статей, разбитый на страницы. Пользователь имеет возможность ознакомиться с обзорным текстом и перейти на страницу статьи;

Данные – архив файлов с данными. Пользователь может изменять файловую структуру представленного архива при наличии необходимых прав. Здесь же пользователь может разместить свои данные для последующего обсчета;

Методы – архив доступных для исполнения вычислительных модулей системы анализа данных. Пользователь может выбрать интересующий его метод, изучить его описание и опробовать на практике. Помимо этого пользователи, назначенные на роль «программиста» имеют доступ к редактированию этого раздела.

Само описание удобно представить в виде набора пользовательских сценариев. Мы рассмотрим четыре основных сценария, которые соответствуют четырем основным типам пользователей: авторизация (гость), администрирование (администратор), запуск (пользователь) и программирование (программист):

а). Гость начинает просмотр с главной страницы сайта, расположенной в данный момент по адресу [archeo.yeahuknow.com]. Здесь он видит актуальную информацию о работе проекта и имеет возможность ознакомиться с теоретическими разработками. Затем он решает, что ему это интересно и замечает кнопки «Вход» и «Регистрация» в верхней части экрана;

б). Гость кликает на кнопку «Регистрация и заполняет предложенную форму;

в). Гость получает сообщение об активации на почту, переходит по указанной ссылке;

г). Гость переходит по ссылке «Вход», вводит регистрационные данные и запоминается системой. С этого момента гость переходит в категорию «пользователь».

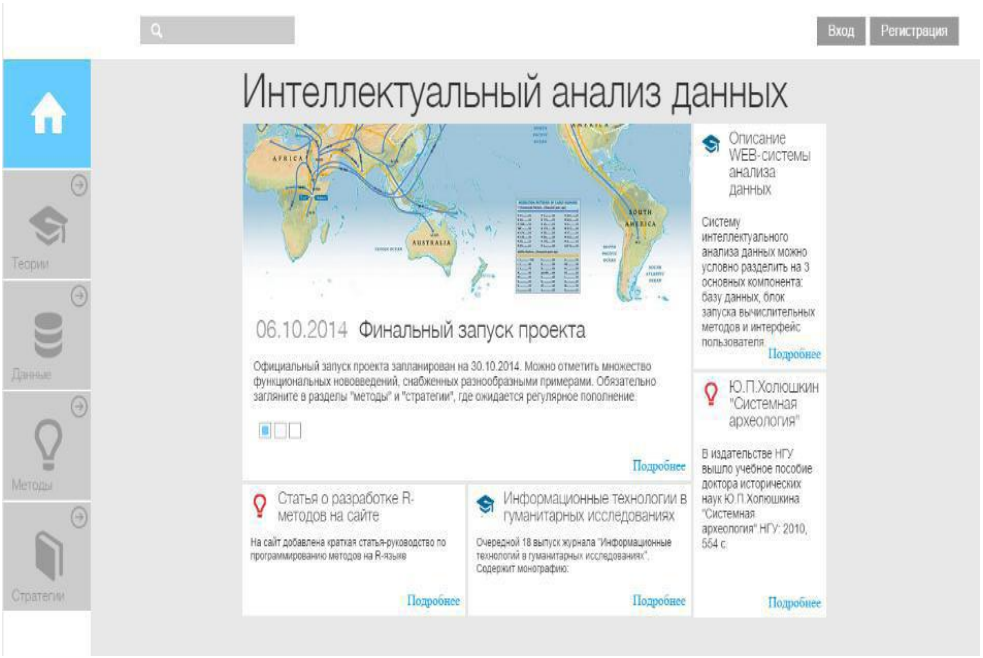


Рис. 12. Главная страница сайта

Сценарий 1 является типичным для большинства веб-сервисов и не заслуживает особого внимания. Перейдем сразу к сценарию 2, администрирование (рис.4):

а). Администратор переходит по скрытой ссылке: http://archeo.yeahuknow.com/?r=admin;

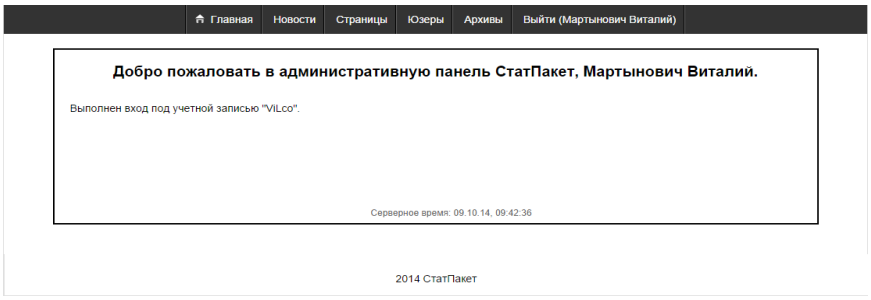


Рис. 13. Административная панель сервиса

б). Администратор кликает на пункт меню «Новости» (рис.14):

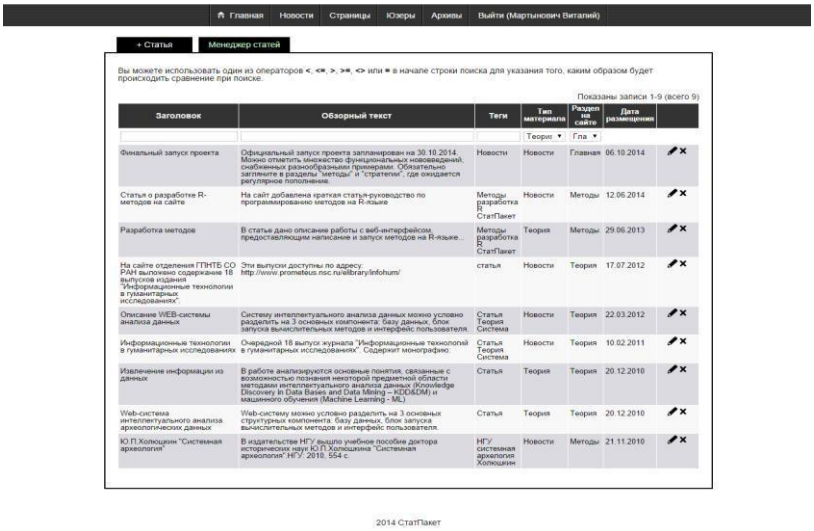


Рис. 14. Менеджер материалов на сайте

в). Администратор выбирает новости для удаления или выбирает режим добавления статьи (рис.15):

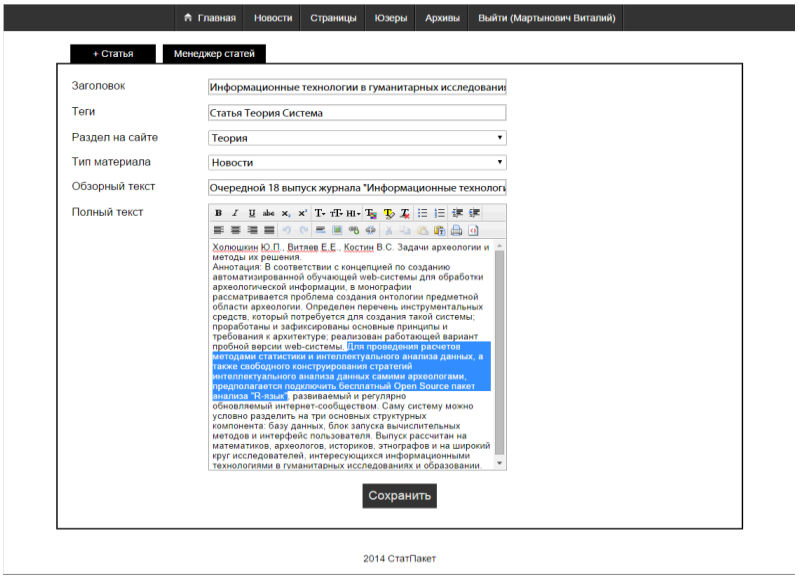


Рис. 15. Редактирование статьи

2.1) Администратор переходит в раздел управления пользователями сайта, выбирая пункт меню «Юзеры», здесь он может назначить пользователям права, заблокировать их или перейти к просмотру данных, загруженных пользователем (рис.16):

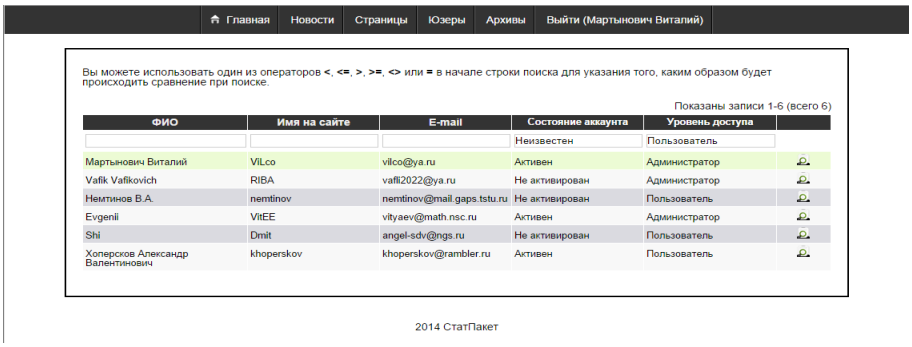


Рис. 16. Управление пользователями

Рассмотрим теперь использование исполняемых элементов, уже присутствующих в системе:

г) Пользователь переходит в архив «методы», используя боковое меню на сайте;

д) С помощью окна навигации, пользователь находит нужный файл метода (рис.17):

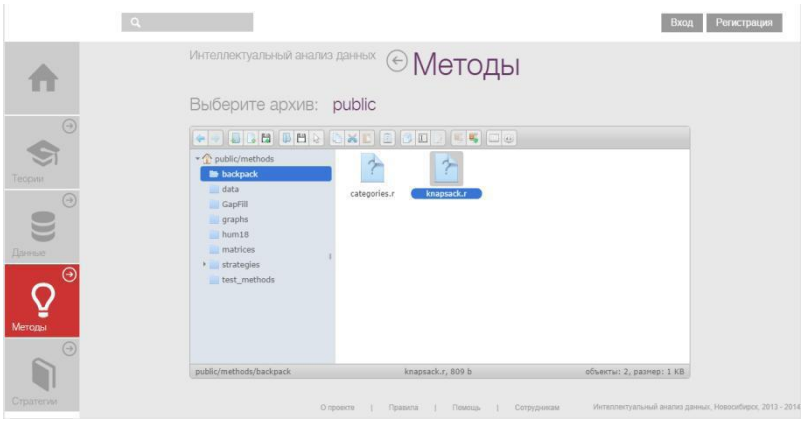


Рис. 17. Выбор метода для исполнения

е) Пользователь изучает сведения о выбранном методе и выбирает опцию «запуск» (рис.18):

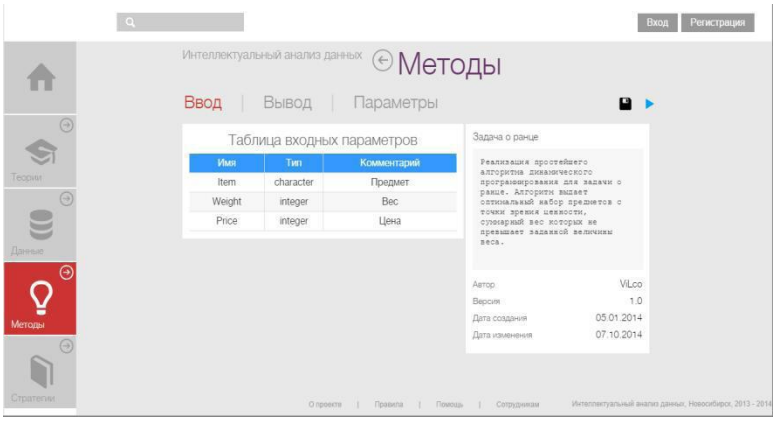


Рис. 18. Запуск метода на исполнение

ё). Далее следует выбор пользователем данных, на которых будет выполнен метод (рис.19):

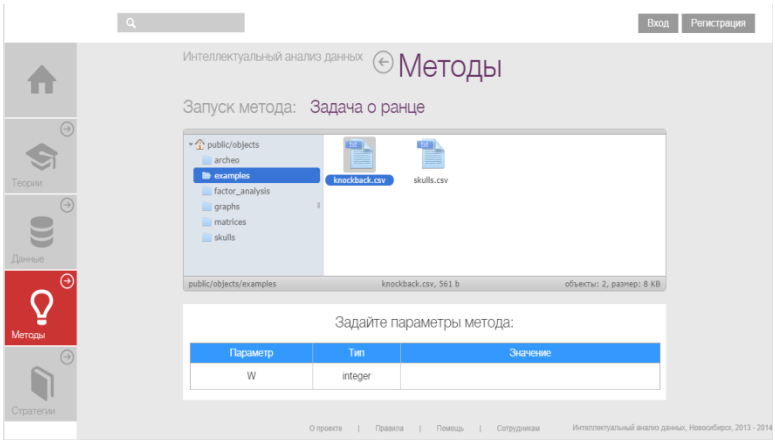


Рис 19. Выбор данных для метода

3.1) Наконец, пользователь устанавливает соответствие между элементами выбранного файла данных и входными переменными метода. Пользователь может поместить всю таблицу целиком в качестве единой переменной, используя «*» из колонки «переменные данных». Все соответствия устанавливаются перетаскиванием меток из колонки «переменные данных» в колонку «источник (из данных)». При необходимости пользователь может использовать несколько файлов данных, выбирая их аналогично п. 3.4. После этого, пользователь задает параметры метода и нажимает кнопку «запуск» (рис.20):

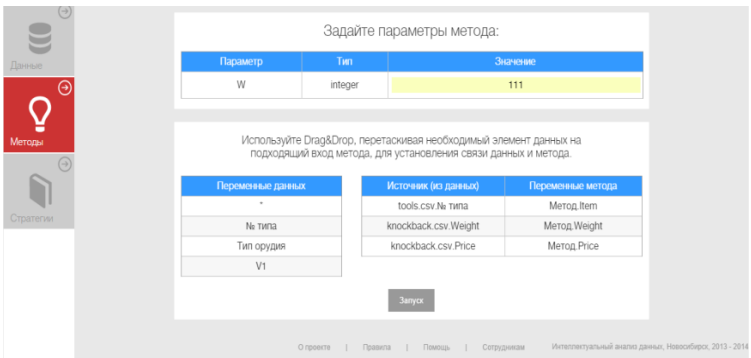


Рис 20. Задание входных переменных и параметров

3.2) На экране будут отображены результаты вычислений. Пользователь увидит таблицу вида «имя переменной – значение», если скалярные переменные присутствуют в выводе, таблицу «сводная таблица», если в результатах работы присутствуют векторные данные, а также набор таблиц для каждой табличной переменной. Любую таблицу можно скачать, кликнув на изображение дискеты и использовать в дальнейших исследованиях (рис.21):

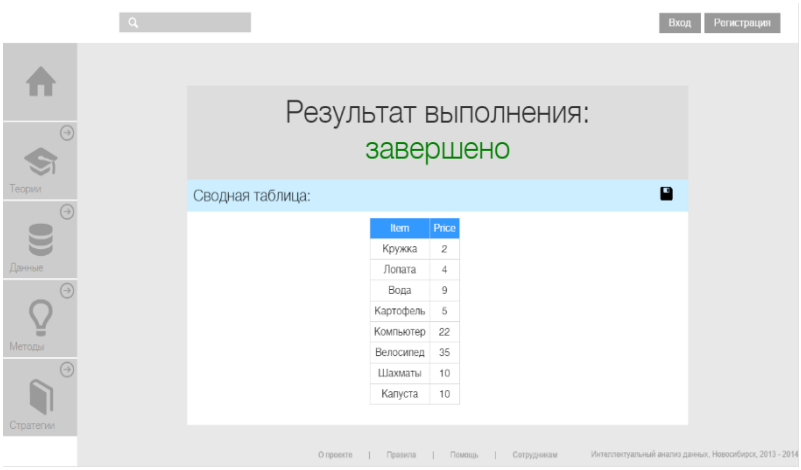


Рис. 21. Результаты выполнения исполняемого элемента

Разработчикам будет интересен сценарий добавления метода в систему:

4.1) Гость авторизуется под учетной записью, имеющей уровень «программист» (пп. 1.x);

4.2) Программист переходит в архив «Методы» используя боковое меню сайта;

4.3) Программист выбирает директорию, исходя из здравого смысла, в которой будет размещен файл метода. Теперь программист выбирает опцию «создать файл» и придумывает имя вновь созданному файлу (рис.22):

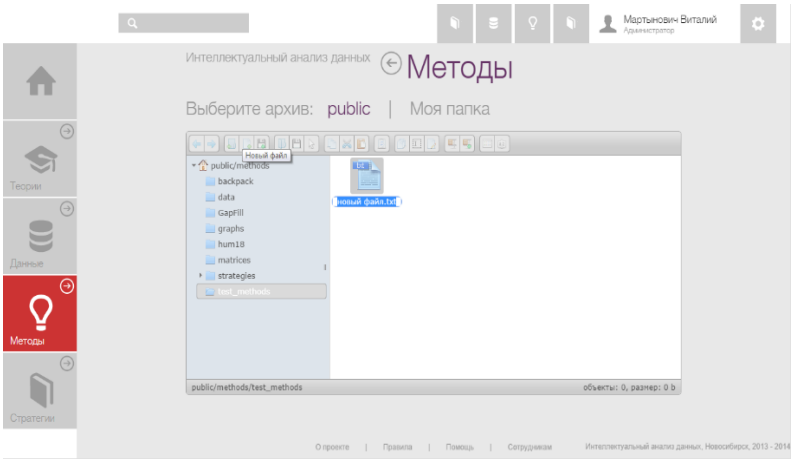


Рис. 22. Создание файла метода

4.4) Для того чтобы приступить к редактированию, программист открывает созданный файл двойным щелчком. В окне редактирования метода в окне «программа» необходимо заполнить всю мета-информацию о методе и вставить код метода на языке R в подокно «Текст программы» (рис.23):

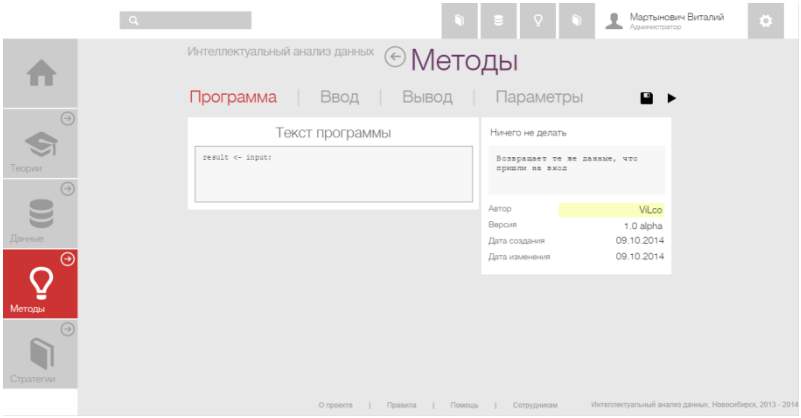


Рис. 23. Редактирование кода метода

4.5) На вкладках «Ввод», «Вывод» и «Параметры» программист заполняет таблицы, описывающие переменные, принимаемые методом на вход и выдаваемые на выходе, а также параметры метода. Кликая на пустые строки снизу программист задает спецификацию в виде имени переменной, её типа и неформального описания. Типы переменных подробно описаны в разделе «Типизация», а ввод-вывод методов – в разделе «Управление вводом и выводом исполняемых элементов». После завершения редактирования таблиц, программист кликает на кнопку «сохранить» (дискета расположенная на панели справа сверху) (рис. 24):

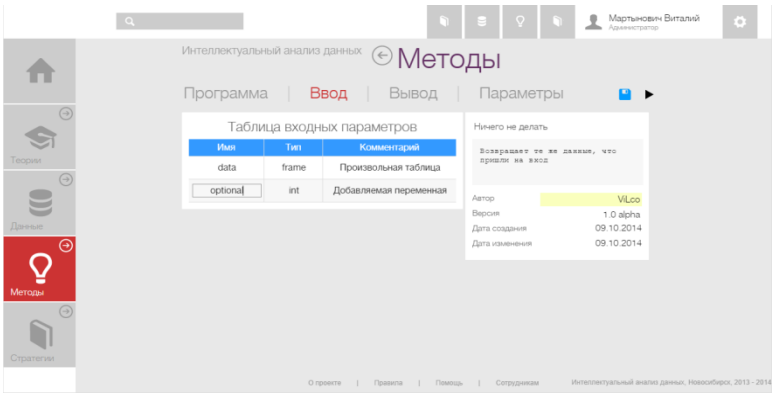


Рис. 24. Редактирование входных переменных (input) метода

Не рассмотренные разделы сайта действуют аналогично описанным, либо не представляют сложностей в самостоятельном освоении для пользователя.

2.4. Инфраструктура среды выполнения.

Основными типами методов, присутствующими в системе на данный момент, являются методы, написанные на языке программирования R. R был выбран нами за его богатый арсенал работы с данными, развитых инструментов работы со статистическими данными, и в то же время, в отличие от большинства других математических пакетов, R – полноценный язык программирования обладающий полнотой по Тьюрингу и внешней схожестью с семействами императивных и функциональных языков.

В веб-систему встроены механизмы запуска и обработки результатов выполнения R-скриптов. Для этого система предоставляет сервер вычислений Rserve [http: //www.r-project.org], обрабатывающий R-скрипты. Схема отработки любого метода примерно следующая:

1. RunController внутри действия EvalMethod обрабатывает данные пользовательского запроса, помещая их внутрь модели Run. EvalMethod загружает объект Method из базы данных;
2. EvalMethod разбирает данные запроса, извлекая информацию о входе в формате сопоставления:

```
Имя_файла => [
```

```
Имя_переменной => Колонка_Данных,
Имя_переменной => Колонка_Данных, …]
```

3. EvalMethod извлекает все задействованные в спецификации входа (п. 2) имена файлов данных и подгружает данные в объекты типа ObjectList;
4. EvalMethod вызывает функцию exec на объектеRun – Run::exec;
5. exec извлекает данные из файлов и преобразует их во внутренний формат, получая набор объектов типа OutputEntry. Кроме того, exec формирует карту заданных параметров;
6. Run::exec готовит среду исполнения: создает временные директории, формирует имена выходных файлов и файлов лога и выставляет им права доступа;
7. Run::exec записывает данные об исполнении в базу данных (время старта, имена выходных файлов и лог-файлов, сведения об исполняемом метода);
8. Run::exec запускает интерпретацию метода Method::exec;
9. Происходит выполнение метода (этот процесс полностью зависит от типа вычислительного модуля и протекает без контроля системы), метод возвращает результат выполнения в виде набора объектов типа OutputEntry;
10. Run::exec группирует объекты по их типу и возвращает результат контроллеру;
11. EvalMethod отображает полученный результат пользователю.

Важным моментом в этой схеме является передача параметров и переменных от веб-сервиса вычислительному серверу и обратно – передача результатов выполнения веб-сервису.

2.5. Типизация и обработка данных.

В любом из сценариев взаимодействия с исполняемым модулем происходит работа с данными в формате OutputEntry. Остановимся на этом моменте немного подробнее (рис.25).

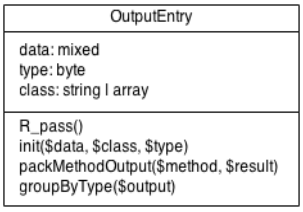


Рис. 25. Диаграмма класса OutputEntry

Результат вычислений выполняет несколько функций: во-первых, анализирует переданные ему данные, определяя один из типов ниже:

1. Значение (value) – скаляр;
2. Вектор (vector) – одно или более скалярное значение одного и того же типа;
3. Таблица (frame) – прямоугольная таблица, каждый столбец является вектором;
4. Список (list) – набор векторов, каждый из которых имеет собственное имя.

Во-вторых, результат вычислений форматирует данные для их непосредственной передачи серверу вычислений (за это ответственен метод R_pass). В-третьих, он осуществляет группировку данных на основе их типа для последующей обработки или выдачи пользователю.

Если тип данных описывает структуру переменной, то класс данных описывает природу элемента внутри этой структуры. Например, значение поля OutputEntry::class равное «character» для данных типа «значение» описывает символьную строку. Понимаемые системой классы:

1. character – символьная строка;

2. integer – целое значение;
3. double – дробное значение;
4. boolean – логическое да/нет (true/false).

Для комплексных типов данных, коими являются frame и list, необходимо описать классы всех присутствующих векторов в наборе. Так происходит потому, что векторы могут иметь разные классы. В этом случае классы передаются в OutputEntry::init в виде строки, элементы которой разделены запятой.

При задании входа метода с помощью файлов данных, возможными типами являются только vector (колонка из файла данных) и frame (файл данных целиком) – для этого служит поле «*» в таблице «переменные данных», указывающая системе на все колонки целиком (рис.26):

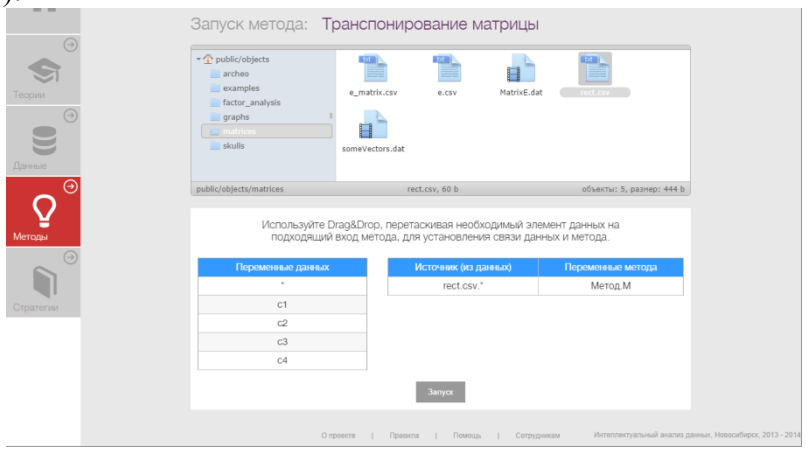


Рис.26. Запуск метода

При передаче данных от метода к методу, спектр возможных типов переменных расширяется. В этой ситуации для стыковки данных применяется описание входа и выхода методов на стыке. В случае если классы данных различаются (например, integer на выходе против double на входе) производится релевантное преобразование. В случае, когда типы фигурирующих данных не являются векторными значениями, применяются специальные описания для элементов входа и выхода методов (рис.27):

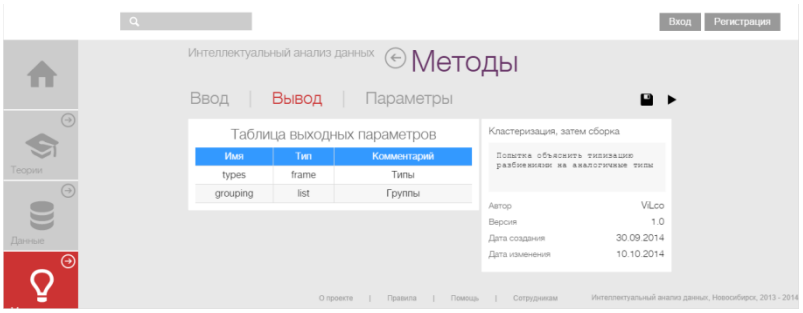


Рис. 27. Специальные типы переменных метода

Для обозначения того, что результат вычислений является списком, следует задать значение поля «тип» в спецификации вывода метода как «list». Аналогично для таблицы применяется значение «frame». То же самое справедливо для спецификации ввода метода.

В сценариях описанных выше, классы векторов, получаемых методом в случае поступления переменных типа «список» и «таблица» будут определяться автоматически. Если необходимо жестко задать классы таких векторов, программист может вместо специального значения «frame» вручную перечислить классы колонок ожидаемой таблицы.

2.6. Поддержка R-языка.

Как уже было отмечено, один из типов поддерживаемых системой вычислительных модулей – это методы, оформленные в виде скриптов на языке R. Для выполнения методов этого типа система предоставляет сервер вычислений Rserve, обрабатывающий R-скрипты.

Возможность написания собственных скриптов-модулей пользователями является чрезвычайно мощной и позволяет существенно расширить возможности системы, но вместе с тем накладывает определенные требования на разработчика таких модулей. Здесь мы расскажем об особенностях программирования R-методов, которое должны быть учтены при встраивании их в систему анализа данных. Часть из них касается среды, в которой выполняются данные, часть предъявляют требования к оформлению кода метода.

R-метод хранится в виде текстового файла в серверном файловом архиве и представляет собой скрипт, предназначенный для выполнения интерпретатором R-языка. Среда выполнения подготавливается в момент запуска метода компонентом Run.

Поток стандартного вывода (stdout), а также поток вывода ошибок (stderr) будут перенаправлены в специально заготовленный файл, файл лога. Имя файла лога сохраняется в базе данных моделей Run. Никаких специальных действий от программиста не требуется. Однако следует помнить, что весь вывод, отправляемый в обычной ситуации с помощью функций языка R на экран пользователя, попадает в лог-файл. Речь о функциях cat, str, и print.

После того, как код R-скрипта получает контроль, ему становится доступны переменные ввода. Для доступа к переменной data, описанной в спецификации ввода, программист использует следующую конструкцию:

```
data<- input[['data']]; # читаем переменную ввода «data»
```

Все переменные ввода доступны из списка input. Далее, все параметры доступны по именам, заданным им при спецификации параметров. Скажем, если параметр был назван FileName, то:

```
write.table(data, file = FileName);# запишем результаты в файл с именем FileName
```

Наконец, по окончании работы метода, все результаты должны быть записаны в список с именем result:

```
result <- list(data=output);# запишем переменную вывода «data» как результат
```

На остальную часть кода никаких ограничений не накладывается. Однако, при использовании специфических пакетов для R-интерпретатора, следует удостовериться в его доступности на сервере вычислений. При программировании R-метода позволяется использоваться любые лексические конструкции языка и вызывать встроенные в язык функции. Последнее расширяет возможности программиста в работе с данными практически до безграничных.

В заключение, простейший R-метод, оформленный по правилам выше, может выглядеть так:

```
M<- input[['M']];# прочитаем входящую переменную-таблицу
T<- as.data.frame(t(M)); # транспонируем таблицу
rownames(T) <- colnames(M);
colnames(T) <- rownames(M);
result<- list('M'=T);# запишем результат выполнения
```

2.7. Shell-методы.

Система предоставляет возможность расширение стандартного набора методов посредством исполняемых файлов. Сервер рассчитан на работу под управлением ОС Linux, соответственно вычислительные модули, подлежащие обсуждению в этой главе –

обыкновенные консольные *NIX-программы, целью которых является преобразование входящих данных в данные исходящие, возможность содержащие дополнительную, вычисленную информацию.

Каждый модуль располагается в одном из архивов статпакета (публичной или частной директории пользователя). В связи с потенциальной уязвимостью серверной системы, загрузка исполняемых утилит производится только администратором после проверки безвредности программы. Как только данные заполнены пользователем и нажат старт, сервер получает команду на запуск утилиты. Сначала проводятся подготовительные работы. Необходимо прочитать заданные пользователем файлы и вычленить из них данные. Затем сервер запускает утилиту с помощью proc_open, используя стандартные средства запуска posix [http://pubs.opengroup.org/onlinepubs/9699919799]. В будущем планируется переход на fork+exec механизм с использованием библиотек pcntl. Это позволяет создать мощную связку демона и утилиты, эффективно и динамично обменивающихся информацией о процессе вычислений и передачи результата.

Ввод, все общение программы и сервера, производится через стандартный поток ввода: STDIN. Механизм оповещений более детально освещен далее.

Первое, что должна сделать утилита – получить данные для работы от сервера. Для этого следует прочитать стандартный поток ввода (STDIN) на предмет сообщения следующего вида:

```
{'variable1':['val1.1', 'val1.2', …], …, 'variableK': ['valK.1', …, 'valK.N']}
```

Прочитать такое сообщение можно любым имеющимся JSON-парсером. Для большинства существующих на планете языков программирования имеются пакеты для работы со строками формата JSON [http://json.org]. Например, на языке PHP извлечение итогового пакета данных в формате ассоциативного массива [имя_переменной => значения_вектора, ...] выглядит примерно так:

```
$calcData = json_decode(file_get_contents('php://input'));
```

Помимо входных данных на STDIN, программа получает 3 параметра командной строки.

1) args[1] = имя log-файла с выполнением программы, который будет показан в случае ошибки и в административной панели сайта;
2) args[2] = имя output-директории, в которую следует записать результаты работы программы в виде таблиц. Эти файлы будут отображены в отформатированном виде в браузере клиента. В последующих версиях предполагается внедрение возможности форматирования этого документа самой работающей утилитой в виде HTML;
3) args[3] = имя result-файла, который должен хранить выходные данные в упакованном виде, в формате JSON (аналогично тому, как на входе).

Стандартная схема работы программы должна состоять в следующем:

1) Программа читает STDIN, помещая результат в переменную input;
2) Распаковать input, JSONDECODE, получив входные данные;
3) Открыть result-файл;
4) Совершить вычисления;
5) Записать результат в result-файл;

Как нетрудно заметить, работа с JSON [http://json.org] – по сути, единственное отличие от стандартной логики консольных программ. Программа сама ответственна за интерпретацию получаемых элементов данных, за их типизацию и преобразования.

В планах реализация Server-queue архитектуры и механизм отслеживания выполнения с помощью программы-демона. На данный момент опрос состояния будет производиться клиентом по таймеру и обрабатываться на сервер по ajax-запросу. Проверяется существование процесса с заданным PID. Если процесс завершил своё функционирование, отрабатывается логика запуска следующей части стратегии или выдается результат на терминал. Никаких действий со стороны выполняемой утилиты не требуется.

Результатом выполнения должен быть список, состоящий из векторов или матриц. Список должен быть упакован с помощью JSONENCODER, а именование и типизация элементов списка должна быть полностью согласована со спецификацией, указанной в веб-интерфейсе при размещении и регистрации метода в WEB-системе статпакета. Результаты выполнения помещаются в output-файлы, расположенные в директории с именем args[2], а итоговый вывод– в файл с именем args[3].

2.8. Пример разработки R-метода.

Для разработки метода и интеграции его в веб-систему, вам понадобятся права уровня «программист». Получить их можно связавшись напрямую с руководителям проекта, либо по e-mail, указанным на странице «Сотрудниками» в нижнем меню сайта.

Предположим, я зарегистрировался на сайте в разделе регистрация (доступно из верхнего правого меню по клику на кнопку «Регистрация») под именем «Programmer» (рис.28):

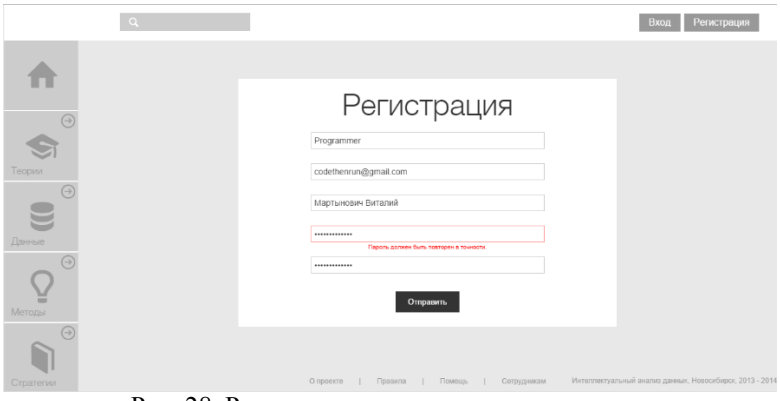


Рис. 28. Регистрация программиста в системе

Теперь я представляюсь системе, путем ввода имени и пароля пользователя, таким образом, подтверждая, что программист – это действительно я (рис.29):

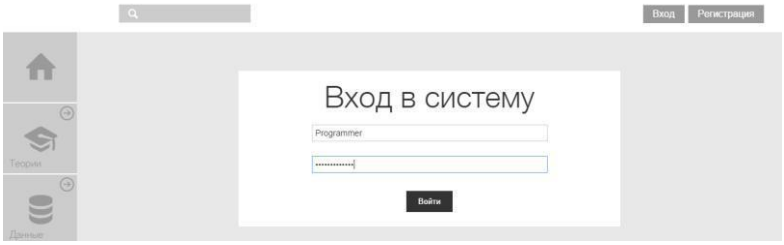


Рис. 29. Авторизация программиста

Теперь можно приступить к разработке метода. Вернемся ненадолго к сценарию 4 (добавление метода в систему) из раздела «Пользовательский интерфейс веб-системы». Выполняя шаги 4.1 – 4.4, создадим файл с именем cov.r, расположенный в общедоступном архиве по адресу /Statistics/Basic. Полное имя, которое получит такой файл будет выглядеть как/public/Statistics/Basic/cov.r (рис.30):

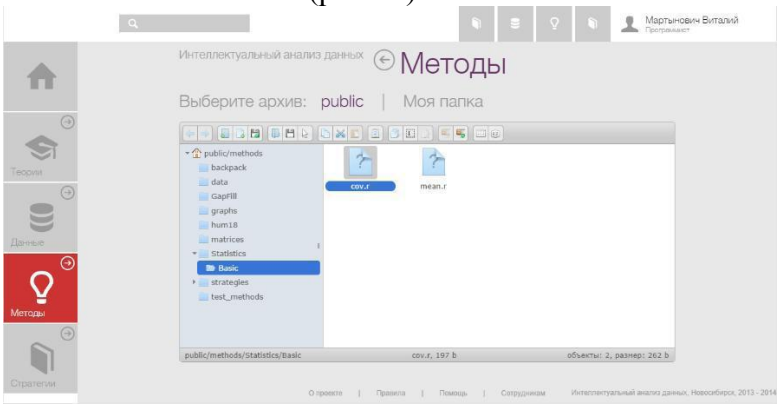


Рис. 30. Создание метода cov.r

Будем программировать подпрограмму, которая вычисляла бы для нас ковариацию случайных величин, выборка для которых представлена таблицей поступающей на вход метода.

Открываем вновь созданный файл cov.ru попадаем в редактор метода (если, конечно, наличествуют права доступа уровня «программист»).

Вооружившись сведениями из главы «Поддержка R-языка», первым делом извлекаем информацию, получаемую методом на вход:

```
data<- input[['data']]; # читаем переменную ввода «data»
```

Основная идея разработки R-методов состоит в том, чтобы использовать интеграцию R-языка в систему и активно пользоваться встроенными функциями этого языка. Поэтому наш генеральный план заключается в использовании функции *cov*, включенной в стандартный пакет R-языка. Однако, функция работает только с числовыми данными, поэтому необходимо привести наш ввод в надлежащий вид:

```
    nums<- sapply(data , is.numeric); # найдем только числовые колонки в таблице
    numericData <- data[,nums];# выберем и запомним их в переменной numericData
```

Теперь переменная numericData содержит то, что и требовалось. Применим к ней стандартную функцию cov:

```
cov<- as.data.frame(cov(numericData)); # таблица ковариаций
```

Здесь нужно отметить один момент. Веб-система работает только с таблицами и обрабатывает матрицы как вектор (в силу специфики R-языка, они таковыми и являются). Поэтому мы заставляем интерпретатор создать из матрицы таблицу, которую и ожидает среда выполнения. Теперь мы готовы записать результат:

```
result<- list('covariance'=cbind('Attribute' = colnames(cov), cov)); # вывод
```

cbind добавляет колонку с именем ‘Attribute’ к результирующей таблице для более удобного восприятия пользователем. После этого, согласно спецификации об интеграции R-языка, мы записываем вывод нашего метода в переменную result, которая будет передана среде исполнения. Итоговый код (без поясняющих комментариев) выглядит так (рис.31):

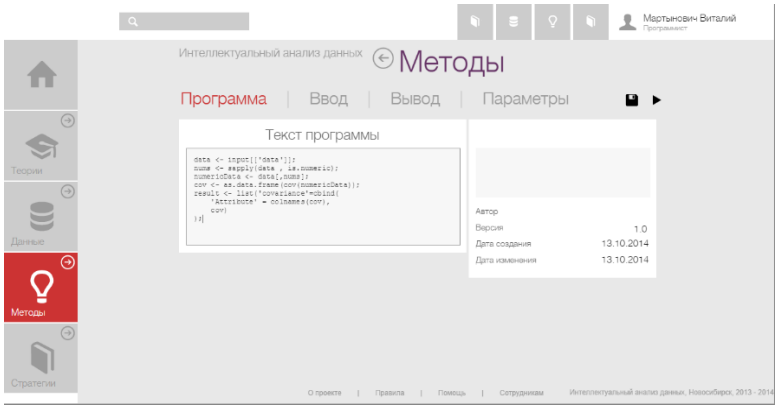


Рис 31. Итоговый код метода

Перейдем к заданию формата входных и выходных данных метода.

- Поскольку наш метод обращается к переменной ‘data’ и ожидает увидеть в ней таблицу с данными, следует задать элемент входа на вкладке «Ввод» с именем ‘data’ и типом ‘frame’; комментарий носит поясняющий характер и может быть произвольным
- Как видно из кода, на выходе метода будет список из одной переменной, которую и следует внести в список на вкладке «Вывод»: добавим элемент ‘covariance’ с типом ‘frame’. Если необходимо добавить больше элементов, заполните колонку «Имя» последней строки таблицы, после чего система сама предложит вам ещё одну пустую строку для заполнения;

- Вкладка «Параметры» заполняется аналогично, в данном случае у метода параметры отсутствуют и мы оставляем её без внимания.

Остается внести справочную информацию в поля «Имя», «Описание», «Автор» и «Версия». Их можно найти на вкладке программа. После внесения информации сохраняем метод. Обязательной является процедура тестирования метода, прежде чем завершать работу попробуйте запустить метод на данных, которые предполагаются пригодными для его работы.

2.9. Библиотека методов.

Прежде чем мы перейдем к составлению исследовательских стратегий, следует познакомить читателя со спектром предлагаемых системой методов. Программирование стратегий невозможно себе представить без ознакомления с этим инструментарием, ведь методы служат кирпичиками, или атомарными элементами в схеме стратегии.

Для упрощенной навигации, методы группированы по семантическому признаку и распределены соответствующим образом по директориям в общедоступном системном архиве (табл.1):

Табл. 1. Структура общедоступного архива методов.

Раздел	Путь в файловой системе	Описание
Matrix	/public/matrix	Методы для работы с числовыми матрицами: арифметические, собственные значения, ядра, детерминант
Data	/public/data	Преобразования исходных данных: сортировки, транспонирование, преобразование форматов, исключение нечисловых признаков и прочее
Graph	/public/graph	Операции на графах: составление графа по инцидентности, выделение остова, алгоритмы на графах
Generators	/public/generators	Генераторы случайных последовательностей (различные распределения)
FillGaps	/public/fillGaps	Заполнение пробелов в данных и устранение недочетов
Clustering	/publci/clustering	Кластеризации объектов в выборке по какому-либо признаку или их группам
OR	/public/or	Методы исследования операций (Operation Research)
SubOptimal	/public/suboptimal	Эвристические и неточные алгоритмы, вероятностные алгоритмы
Statistics	/public/statistics	Статистические методы
Statistics/Basic	/public/statistics/basic	Базовые операции: среднее, дисперсия, ковариация и прочее
Statistics/Criteria	/public/statistics/criteria	Статистические критерии

Следующую таблицу можно использовать как справочник при проектировании собственных стратегий. Однако не следует считать перечисленный ниже список исчерпывающим, поскольку система будет постоянно дополняться новыми методами (с помощью внедрения существующих в R-языке, либо согласно запросу пользовательской аудитории) (табл. 2):

Табл. 2. Библиотека методов системы

Метод (файл)	Вход	Выход	Параметры	Описание
Matrix				
sum.r	A: frame; B: frame;	M: frame	Нет	Выполняет поэлементное сложение 2 таблиц (A + B)
mul.r	A: frame; B: frame;	M: frame	Нет	Перемножает 2 числовых матрицы (AxB)
eigenValues.r	A: frame;	values: double	Нет	Вычисляет собственные значения матрицы A
Data				
sort.r	data: frame;	sorted: frame;	column: char;	Сортирует входную таблицу data по колонке, указанной в параметре "column"
transpose.r	M: frame;	T: frame;	Нет	Делает строки в таблицу T столбцами, а столбцы - строками
numericOnly.r	M: frame;	Numeric: frame;	Нет	Убирает все нечисловые столбцы из выборки M
parse.r	Нет	v: char	s: char;	Разбирает входящую строку s в формате "s1, s2, …, sN" и создает из неё вектор значений v
concat.r	M: frame; N: frame;	C: frame;	Нет	Склеивает таблицы или векторы M и N в одну таблицу C
Graph				
matrixToGraph.r	M: frame;	From: char; To: char; Weight: double;	Нет	Строит граф G по матрице весов M. Выходная конструкция является перечислением ребер графа
deikstra.r	From: char; To: char; Length: double;	To: char; Length: double; Path: char;	From: char;	Алгоритм Дейкстры для нахождения кратчайших путей на графе. В качестве параметра указывается исходная вершина, в качестве результата - длины и пути, на которых достигаются минимальные длины
vertex_cover.r	From: char; To: char;	Vertex: char	Нет	Быстро находит вершинное покрытие графа. Вершинное покрытие необязательно будет минимальным, т.е. полученное решение - субоптимально
Generators				

normal.r	Нет	Random: double;	u: double; s: double; N: integer;	Генерирует N чисел из нормального распределения с MO и дисперсией s
atomic.r	atoms: double; p: double;	Random: double;	N: integer	Генерирует N чисел из дискретного распределения с заданными атомами
bernoulli.r	Нет	Random: double;	p: double; N: integer;	Генерирует N чисел из распределения Бернулли с параметром вероятности p
FillGaps				
put.r	data: frame;	filled: frame;	value: char;	Пытается заполнить пустые или NA поля в data с помощью значения value
linearRegression.r	data: frame;	filled: frame;	base: char;	Пытается заполнить пустые или NA поля в data с помощью построения линейной регрессионной модели на базе колонок перечисленных в base
Clustering				
k_means.r	data: frame;	id: integer; cluser: integer; centers: frame;	k: integer;	Кластеризация объектов из data на основе алгоритма k-средних, число кластеров определяется параметром k
ttrees.r	data: frame;	id: integer; cluser: integer;	Нет	Кластеризация на основе критерия доли объясненной дисперсии
OR				
knapsack.r	Item: char; Weight: integer; Price: integer;	Item: char; Price: integer;	W: integer	Решает задачу о ранце: выбирает предметы из Item, суммарный вес Weight которых не превышает параметра W, максимизируя стоимость Price
SubOptimal				
Statistics				
quantile.r				
Statistics/Basic				
cov.r	data: frame;	cov: frame;	Нет	Вычисляет матрицу ковариаций для числовых признаков выборки data
mean.r	sample: double;	mean: double;	Нет	Вычисляет среднее mean по выборке sample

Statistics/Criteria				
fisher.r	sample: frame;	val: double;	Нет	Вычисляет значение критерия Фишера для представленной выборки sample
student.r	sample: frame;	val: double;	Нет	Вычисляет значений критерия Стьюдента для представленной выборки sample
chi.r	sample: frame;	val: double;	Нет	Вычисляет значений критерия Хи-квадрат для представленной выборки sample

Информацию о методах, присутствующих в архиве планируется поддерживать в актуальном состоянии по адресу [http:www. sibirica.spsl.nsc.ru].

2.9. Пример разработки стратегии

Наиболее интересный для исследователя инструмент – визуальный редактор стратегий.

Кликнув на вкладку «Стратегии» в боковом меню, пользователь найдет уже знаковый ему файловый архив по работе с методами и данными, однако содержащий уже исследовательские стратегии. Все те же операции, все та же навигация.

Сейчас нам понадобится опция «создать файл». Создадим файл стратегии «MatrixDeikstra.rs» в общем доступном архиве в секции examples. Полный путь до файла - /public/examples/MatrixDeikstra.rs (рис.32):

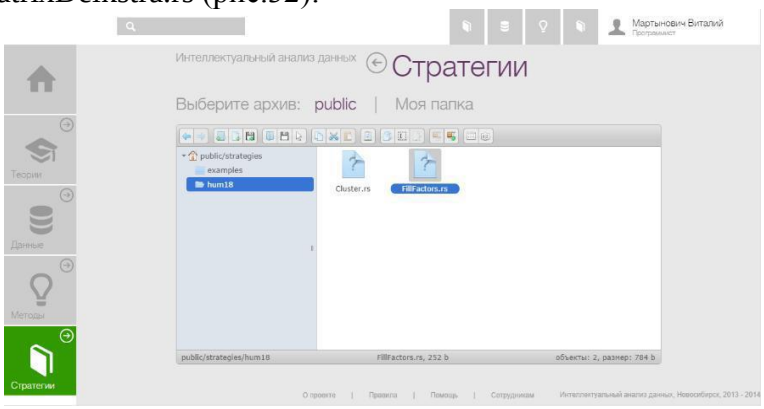


Рис 32. Архив стратегий

Наиболее интересный для исследователя инструмент – визуальный редактор стратегий.

III ПРОГРАММИРОВАНИЕ МЕТОДОВ И СТРАТЕГИЙ

3.1. Инфраструктура среды выполнения.

Основными типами методов, присутствующими в системе на данный момент, являются методы, написанные на языке программирования R. R был выбран нами за его богатый арсенал работы с данными, развитых инструментов работы со статистическими данными, и в то же время, в отличие от большинства других математических пакетов, R– полноценный язык программирования обладающий полнотой по Тьюрингу и внешней схожестью с семействами императивных и функциональных языков.

В веб-систему встроены механизмы запуска и обработки результатов выполнения R-скриптов. Для этого система предоставляет сервер вычислений Rserve [http://rforge.net/Rserve], обрабатывающий R-скрипты. Схема отработки любого метода примерно следующая:

1. RunController внутри действия EvalMethod обрабатывает данные пользовательского запроса, помещая их внутрь модели Run. EvalMethod загружает объект Method из базы данных;
2. EvalMethod разбирает данные запроса, извлекая информацию о входе в формате сопоставления:

```
Имя_файла => [
    Имя_переменной => Колонка_Данных,
    Имя_переменной => Колонка_Данных, …]
```

3. EvalMethod извлекает все задействованные в спецификации входа (п. 2) имена файлов данных и подгружает данные в объекты типа ObjectList;
4. EvalMethod вызывает функцию exесна объекте Run –Run::exec;
5. Exec извлекает данные из файлов и преобразует их во внутренний формат, получая набор объектов типа OutputEntry. Кроме того, exec формирует карту заданных параметров;
6. Run::exec готовит среду исполнения: создает временные директории, формирует имена выходных файлов и файлов лога и выставляет им права доступа;
7. Run::exec записывает данные об исполнении в базу данных (время старта, имена выходных файлов и лог-файлов, сведения об исполняемом методе);
8. Run::exec запускает интерпретацию метода Method::exec;
9. Происходит выполнение метода (этот процесс полностью зависит от типа вычислительного модуля и протекает без контроля системы), метод возвращает результат выполнения в виде набора объектов типа OutputEntry;
10. Run::exec группирует объекты по их типу и возвращает результат контроллеру;
11. EvalMethod отображает полученный результат пользователю.

Важным моментом в этой схеме является передача параметров и переменных от веб-сервиса вычислительному серверу и обратно – передача результатов выполнения веб-сервису.

3.2.Типизация и обработка данных

В любом из сценариев взаимодействия с исполняемым модулем происходит работа с данными в формате OutputEntry. Остановимся на этом моменте немного подробнее (рис.33).

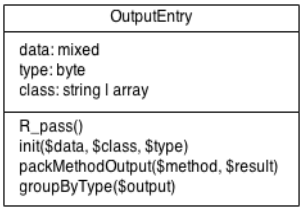


Рис. 33. Диаграмма класса OutputEntry

Результат вычислений выполняет несколько функций: во-первых, анализирует переданные ему данные, определяя один из типов ниже:

1. Значение (value) – скаляр;
2. Вектор (vector) – одно или более скалярное значение одного и того же типа;
3. Таблица (frame) – прямоугольная таблица, каждый столбец является вектором;
4. Список (list) – набор векторов, каждый из которых имеет собственное имя.

Во-вторых, результат вычислений форматирует данные для их непосредственной передачи серверу вычислений (за это ответственен метод R_pass). В-третьих, он осуществляет группировку данных на основе их типа для последующей обработки или выдачи пользователю.

Если тип данных описывает структуру переменной, то класс данных описывает природу элемента внутри этой структуры. Например, значение поля OutputEntry::class равное «character» для данных типа «значение» описывает символьную строку. Понимаемые системой классы:

1. character – символьная строка;
2. integer – целое значение;
3. double – дробное значение;
4. boolean – логическое да/нет (true/false).

Для комплексных типов данных, коими являются frame list, необходимо описать классы всех присутствующих векторов в наборе. Так происходит потому, что векторы могут иметь разные классы. В этом случае классы передаются в OutputEntry::init в виде строки, элементы которой разделены запятой.

При задании входа метода с помощью файлов данных, возможными типами являются только vector (колонка из файла данных) и frame (файл данных целиком) – для этого служит поле «*» в таблице «переменные данных», указывающая системе на все колонки целиком:

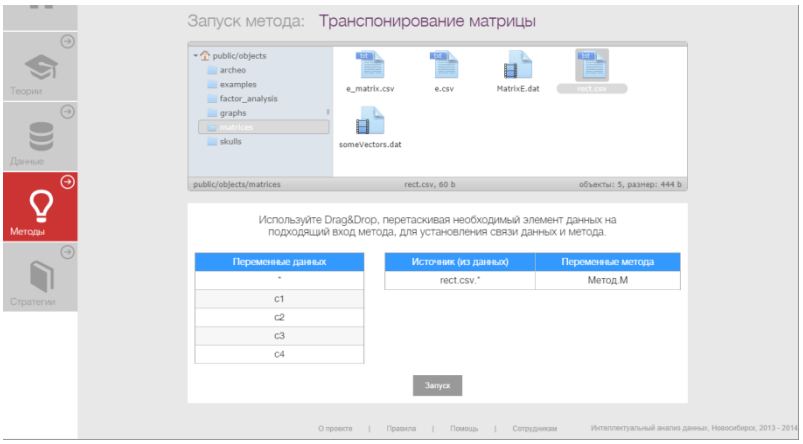


Рис. 34. Запуск метода

При передаче данных от метода к методу, спектр возможных типов переменных расширяется. В этой ситуации для стыковки данных применяется описание входа и выхода методов на стыке. В случае если классы данных различаются (например, integer на

выходе против double на входе) производится релевантное преобразование. В случае, когда типы фигурирующих данных не являются векторными значениями, применяются специальные описания для элементов входа и выхода методов (рис. 35):

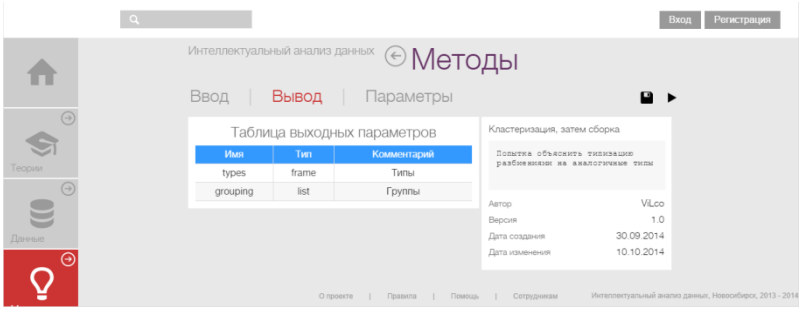


Рис. 35. Специальные типы переменных метода

Для обозначения того, что результат вычислений является списком, следует задать значение поля «тип» в спецификации вывода метода как «list». Аналогично для таблицы применяется значение «frame». То же самое справедливо для спецификации ввода метода.

В сценариях описанных выше, классы векторов, получаемых методом в случае поступления переменных типа «список» и «таблица» будут определяться автоматически. Если необходимо жестко задать классы таких векторов, программист может вместо специального значения «frame» вручную перечислить классы колонок ожидаемой таблицы.

3.3.Поддержка R-языка.

Как уже было отмечено, один из типов поддерживаемых системой вычислительных модулей – это методы, оформленные в виде скриптов на языке R. Для выполнения методов этого типа система предоставляет сервер вычислений Rserve, обрабатывающий R-скрипты.

Возможность написания собственных скриптов-модулей пользователями является чрезвычайно мощной и позволяет существенно расширить возможности системы, но вместе с тем накладывает определенные требования на разработчика таких модулей. Здесь следует рассказать об особенностях программирования R-методов, которое должны быть учтены при встраивании их в систему анализа данных. Часть из них касается среды, в которой выполняются данные, часть предъявляют требования к оформлению кода метода.

R-метод хранится в виде текстового файла в серверном файловом архиве и представляет собой скрипт, предназначенный для выполнения интерпретатором R-языка. Среда выполнения подготавливается в момент запуска метода компонентом Run. Поток стандартного вывода (stdout), а также поток вывода ошибок (stderr) будут перенаправлены в специально заготовленный файл, файл лога. Имя файла лога сохраняется в базе данных моделей Run. Никаких специальных действий от программиста не требуется. Однако, следует помнить, что весь вывод, отправляемый в обычной ситуации с помощью функций языка R на экран пользователя, попадает в лог-файл. Речь о функциях cat, str, и print.

После того, как код R-скрипта получает контроль, ему становится доступны переменные ввода. Для доступа к переменной data, описанной в спецификации ввода, программист использует следующую конструкцию:

```
data<- input[['data']]; #читаем переменную ввода «data»
```

Все переменные ввода доступны из списка input. Далее, все параметры доступны по именам, заданным им при спецификации параметров. Скажем, если параметр был назван FileName, то:

```
write.table(data, file = FileName); # запишем результаты в файл с именем
FileName
```

Наконец, по окончании работы метода, все результаты должны быть записаны в список с именем result:

```
result <- list(data=output);# запишем переменную вывода «data» как
                            результат
```

На остальную часть кода никаких ограничений не накладывается. Однако, при использовании специфических пакетов для R-интерпретатора, следует удостовериться в его доступности на сервере вычислений. При программировании R-метода позволяется использоваться любые лексические конструкции языка и вызывать встроенные в язык функции. Последнее расширяет возможности программиста в работе с данными практически до безграничных.

В заключение, простейший R-метод, оформленный по правилам выше, может выглядеть так:

```
M<- input[['M']];# прочитаем входящую переменную-таблицу
T<- as.data.frame(t(M)); # транспонируем таблицу
rownames(T) <- colnames(M);
colnames(T) <- rownames(M);
result<- list('M'=T);# запишем результат выполнения
```

3.4. Shell-методы

Система предоставляет возможность расширение стандартного набора методов посредством исполняемых файлов. Сервер рассчитан на работу под управлением ОС Linux, соответственно вычислительные модули, подлежащие обсуждению в этой главе – обыкновенные консольные *NIX-программы, целью которых является преобразование входящих данных в данные исходящие, возможность содержащие дополнительную, вычисленную информацию.

Каждый модуль располагается в одном из архивов статпакета (публичной или частной директории пользователя). В связи с потенциальной уязвимостью серверной системы, загрузка исполняемых утилит производится только администратором после проверки безвредности программы.

Как только данные заполнены пользователем и нажат старт, сервер получает команду на запуск утилиты. Сначала проводятся подготовительные работы. Необходимо прочитать заданные пользователем файлы и вычленить из них данные. Затем сервер запускает утилиту с помощью proc_open, используя стандартные средства запуска posix [http://pubs.opengroup.org/onlinepubs/9699919799]. В будущем планируется переход на fork+exec механизм с использованием библиотек pcntl. Это позволяет создать мощную связку демона и утилиты, эффективно и динамично обменивающихся информацией о процессе вычислений и передачи результата.

Ввод, все общение программы и сервера, производится через стандартный поток ввода: STDIN. Механизм оповещений более детально освещен далее.

Первое, что должна сделать утилита – получить данные для работы от сервера. Для этого следует прочитать стандартный поток ввода (STDIN) на предмет сообщения следующего вида:

```
{'variable1': ['val1.1', 'val1.2', …], …, 'variableK': ['valK.1', …,
                                    'valK.N']}
```

Прочитать такое сообщение можно любым имеющимся JSON-парсером [http: //json.org]. Для большинства существующих на планете языков программирования имеются пакеты для работы со строками формата JSON. Например, на языке PHP

извлечение итогового пакета данных в формате ассоциативного массива [имя_переменной => значения_вектора, ...] выглядит примерно так:

```
$calcData = json_decode(file_get_contents('php://input'));
```

Помимо входных данных на STDIN, программа получает 3 параметра командной строки.

1) args[1] = имя log-файла с выполнением программы, который будет показан в случае ошибки и в административной панели сайта;
2) args[2] = имя output-директории, в которую следует записать результаты работы программы в виде таблиц. Эти файлы будут отображены в отформатированном виде в браузере клиента. В последующих версиях предполагается внедрение возможности форматирования этого документа самой работающей утилитой в виде HTML;
3) args[3] = имя result-файла, который должен хранить выходные данные в упакованном виде, в формате JSON (аналогично тому, как на входе).

Стандартная схема работы программы должна состоять в следующем:

1) Программа читает STDIN, помещая результат в переменную input;
2) Распаковать input, JSONDECODE, получив входные данные;
3) Открыть result-файл;
4) Совершить вычисления;
5) Записать результат в result-файл.

Как нетрудно заметить, работа с JSON – содержит единственное отличие от стандартной логики консольных программ. Программа сама ответственна за интерпретацию получаемых элементов данных, за их типизацию и преобразования.

В планах реализация Server-queue архитектуры и механизм отслеживания выполнения с помощью программы-демона. На данный момент опрос состояния будет производится клиентом по таймеру и обрабатываться на сервер по ajax-запросу. Проверяется существование процесса с заданным PID. Если процесс завершил своё функционирование, то отрабатывается логика запуска следующей части стратегии или выдается результат на терминал. Никаких действий со стороны выполняемой утилиты не требуется.

Результатом выполнения должен быть список, состоящий из векторов или матриц. Список должен быть упакован с помощью JSONENCODER, а именование и типизация элементов списка должна быть полностью согласована со спецификацией, указанной в веб-интерфейсе при размещении и регистрации метода в WEB-системе статпакета. Результаты выполнения помещаются в output-файлы, расположенные в директории с именем args[2], а итоговый вывод– в файл с именем args[3].

3.5. Пример разработки R-метода.

Для разработки метода и интеграции его в веб-систему, вам понадобятся права уровня «программист». Получить их можно связавшись напрямую с руководителем проекта, либо по e-mail, указанным на странице «Сотрудникам» в нижнем меню сайта (рис. 36).

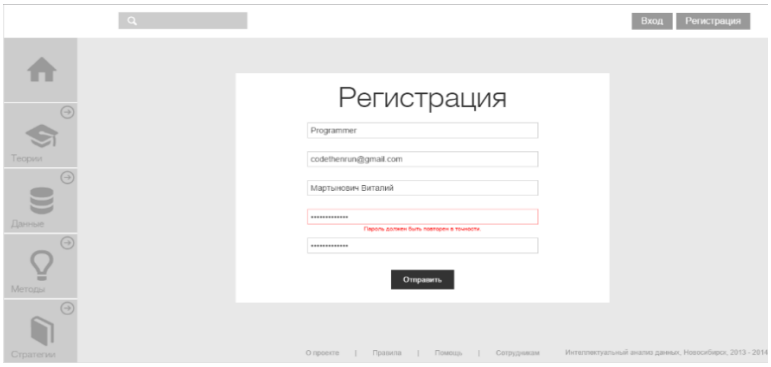


Рис. 36. Регистрация программиста в системе

Предположим, я зарегистрировался на сайте в разделе регистрация (доступно из верхнего правого меню по клику на кнопку «Регистрация») под именем «Programmer»:

Теперь я представляюсь системе, путем ввода имени и пароля пользователя, таким образом, подтверждая, что программист – действительно я (рис.37):

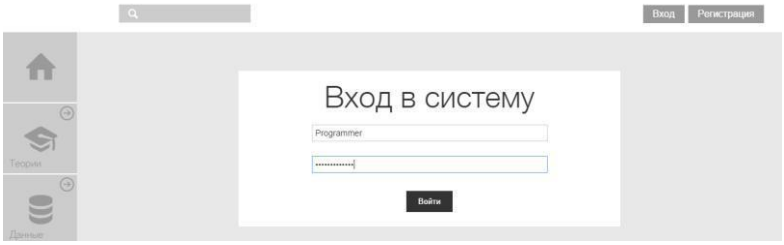


Рис. 37. Авторизация программиста

Теперь можно приступить к разработке метода. Вернемся ненадолго к сценарию 4 (добавление метода в систему) из раздела «Пользовательский интерфейс веб-системы». Выполняя шаги 4.1 – 4.4, создадим файл с именем cov.r, расположенный в общедоступном архиве по адресу /Statistics/Basic. Полное имя, которое получит такой файл будет выглядеть как/public/Statistics/Basic/cov.r (рис. 38):

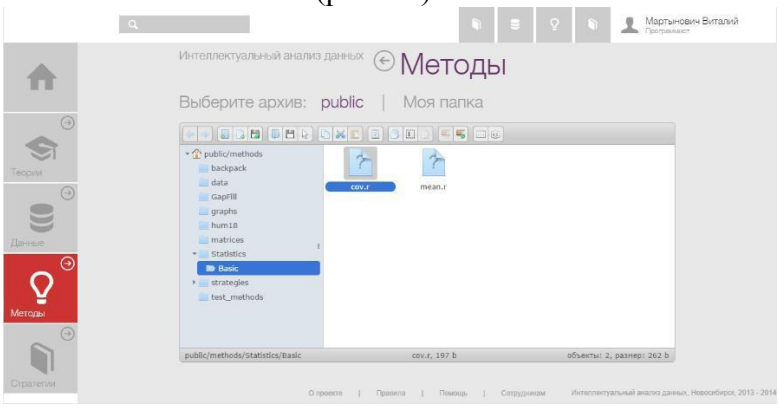


Рис. 38. Создание метода cov.r

Будем программировать подпрограмму, которая вычисляла бы для нас ковариацию случайных величин, выборка для которых представлена таблицей поступающей на вход метода.

Открываем вновь созданный файл cov.ri попадаем в редактор метода (если, конечно, наличествуют права доступа уровня «программист»).

Вооружившись сведениями из главы «Поддержка R-языка», первым делом извлекаем информацию, получаемую методом на вход:

```
data<- input[['data']]; # читаем переменную ввода «data»
```

Основная идея разработки R-методов состоит в том, чтобы использовать интеграцию R-языка в систему и активно пользоваться встроенными функциями этого языка. Поэтому наш генеральный план заключается в использовании функции *cov*, включенной в стандартный пакет R-языка. Однако функция работает только с числовыми данными, поэтому необходимо привести наш ввод в надлежащий вид:

```
nums<- sapply(data , is.numeric); # найдем только числовые колонки в
        таблице
numericData <- data[,nums];# выберем и запомним их в переменной
        numericData
```

Теперь переменная numericData содержит то, что и требовалось. Применим к ней стандартную функцию cov:

```
cov<- as.data.frame(cov(numericData)); # таблица ковариаций
```

Здесь нужно отметить один момент. Веб-система работает только с таблицами и обрабатывает матрицы как вектор (в силу специфики R-языка, они таковыми и являются). Поэтому мы заставляем интерпретатор создать из матрицы таблицу, которую и ожидает среда выполнения. Теперь мы готовы записать результат:

```
result<- list('covariance'=cbind('Attribute' = colnames(cov), cov)); #
        вывод
```

cbind добавляет колонку с именем ‘Attribute’ к результирующей таблице для более удобного восприятия пользователем. После этого, согласно спецификации об интеграции R-языка, мы записываем вывод нашего метода в переменную result, которая будет передана среде исполнения. Итоговый код (без поясняющих комментариев) выглядит так (рис.39):

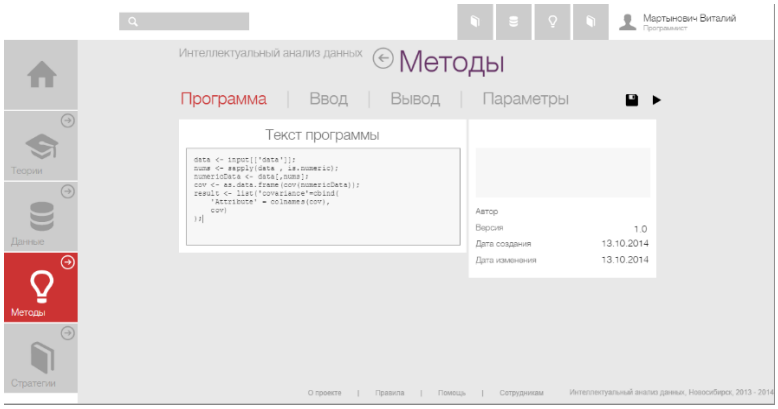


Рис 39. Итоговый код метода

Перейдем к заданию формата входных и выходных данных метода.

- Поскольку наш метод обращается к переменной ‘data’ и ожидает увидеть в ней таблицу с данными, следует задать элемент входа на вкладке «Ввод» с именем ‘data’ и типом ‘frame’; комментарий носит поясняющий характер и может быть произвольным
- Как видно из кода, на выходе метода будет список из одной переменной, которую и следует внести в список на вкладке «Вывод»: добавим элемент ‘covariance’ с типом ‘frame’. Если необходимо добавить больше элементов, заполните колонку «Имя» последней строки таблицы, после чего система сама предложит вам ещё одну пустую строку для заполнения;

- Вкладка «Параметры» заполняется аналогично, в данном случае у метода параметры отсутствуют, и мы оставляем её без внимания.

Остается внести справочную информацию в поля «Имя», «Описание», «Автор» и «Версия». Их можно найти на вкладке программа. После внесения информации сохраняем метод. Обязательной является процедура тестирования метода, прежде чем завершать работу попробуйте запустить метод на данных, которые предполагаются пригодными для его работы.

3.6. Библиотека методов.

Прежде чем мы перейдем к составлению исследовательских стратегий, следует познакомить читателя со спектром предлагаемых системой методов. Программирование стратегий невозможно себе представить без ознакомления с этим инструментарием, ведь методы служат кирпичиками, или атомарными элементами в схеме стратегии.

Для упрощенной навигации, методы группированы по семантическому признаку и распределены соответствующим образом по директориям в общедоступном системном архиве (табл. 3):

Табл. 3. Структура общедоступного архива методов

Раздел	Путь в файловой системе	Описание
Matrix	/public/matrix	Методы для работы с числовыми матрицами: арифметические, собственные значения, ядра, детерминант
Data	/public/data	Преобразования исходных данных: сортировки, транспонирование, преобразование форматов, исключение нечисловых признаков и прочее
Graph	/public/graph	Операции на графах: составление графа по инцидентности, выделение остова, алгоритмы на графах
Generators	/public/generators	Генераторы случайных последовательностей (различные распределения)
FillGaps	/public/fillGaps	Заполнение пробелов в данных и устранение недочетов
Clustering	/publci/clustering	Кластеризации объектов в выборке по какому-либо признаку или их группам
OR	/public/or	Методы исследования операций (OperationResearch)
SubOptimal	/public/suboptimal	Эвристические и неточные алгоритмы, вероятностные алгоритмы
Statistics	/public/statistics	Статистические методы
Statistics/Basic	/public/statistics/basic	Базовые операции: среднее, дисперсия, ковариация и прочее
Statistics/Criteria	/public/statistics/criteria	Статистические критерии

Следующую таблицу можно использовать как справочник при проектировании собственных стратегий. Однако, не следует считать перечисленный ниже список исчерпывающим, поскольку система будет постоянно дополняться новыми методами (с помощью внедрения существующих в R-языке, либо согласно запросу пользовательской аудитории) (табл. 4):

Табл. 4. Библиотека методов системы

Метод (файл)	Вход	Выход	Параметры	Описание
Matrix				
sum.r	A: frame; B: frame;	M: frame	Нет	Выполняет поэлементное сложение 2 таблиц (A + B)
mul.r	A: frame; B: frame;	M: frame	Нет	Перемножает 2 числовые матрицы (AxB)
pow.r	A: frame	M: frame	n: double	Возводит матрицу в степень, в том числе для n=-1, вычисляет обратную к A матрицу
eigenValues.r	A: frame;	values: double	Нет	Вычисляет собственные значения матрицы A
cond.r	A: frame;	cond: double;	Нет	Вычисляет число обусловленности матрицы в стандартной R^2-норме
Data				
sort.r	data: frame;	sorted: frame;	column: char;	Сортирует входную таблицу data по колонке, указанной в параметре "column"
transpose.r	M: frame;	T: frame;	Нет	Делает строки в таблицу T столбцами, а столбцы – строками
numericOnly.r	M: frame;	Numeric: frame;	Нет	Убирает все нечисловые столбцы из выборки M
parse.r	Нет	v: char	s: char;	Разбирает входящую строку s в формате "s1, s2, …, sN" и создает из неё вектор значений v
select.r	data: frame; cols: char;	subdata: frame;	Нет	Выбирает указанные колонки cols из исходной таблицы данных data
param.r	Нет	v: char	v: char;	Преобразует параметр в результат элемента системы (Output Entry)
compose.r	M: frame; N: frame;	C: frame;	Нет	Склеивает таблицы или векторы M и N в одну таблицу C
concat.r	u: char; v: char;	C: frame;	Нет	Создает из двух векторов u и v один путем конкатенации
Graph				
matrixToGraph.r	M: frame;	From: char; To: char; Weight: double;	Нет	Строит граф G по матрице весов M. Выходная конструкция является перечислением ребер графа

graphToMatrix.r	From: char; To: char; Weight: double;	M: frame	Нет	Выдает матрицу весов по реберному описанию графа
deikstra.r	From: char; To: char; Length: double;	To: char; Length: double; Path: char;	From: char;	Алгоритм Дейкстры для нахождения кратчайших путей на графе. В качестве параметра указывается исходная вершина, в качестве результата - длины и пути, на которых достигаются минимальные длины
spantree.r	M: frame;	G: frame;	Нет	Находит минимальное остовное дерево на графе M. Выход представляет собой матрицу инцидентности не взвешенного итогового графа G
vertex_cover.r	From: char; To: char;	Vertex: char	Нет	Быстро находит вершинное покрытие графа. Вершинное покрытие необязательно будет минимальным, т.е. полученное решение – субоптимально
Generators				
normal.r	Нет	Random: double;	u: double; s: double; N: integer;	Генерирует N чисел из нормального распределения с MO и дисперсией s
uniform.r	Нет	Random: double;	a: double; b: double; N: integer;	Генерирует N чисел из равномерного непрерывного распределения с началом в точке a и концом в точке b
atomic.r	atoms: double; p: double;	Random: double;	N: integer	Генерирует N чисел из дискретного распределения с заданными атомами
bernoulli.r	Нет	Random: double;	p: double; N: integer;	Генерирует N чисел из распределения Бернулли с параметром вероятности p
FillGaps				
put.r	data: frame;	filled: frame;	value: char;	Пытается заполнить пустые или NA поля в data с помощью значения value

linearRegression.r	data: frame;	filled: frame;	base: char;	Пытается заполнить пустые или NA поля в data с помощью построения линейной регрессионной модели на базе колонок перечисленных в base
Clustering				
k_means.r	data: frame;	id: integer; cluser: integer; centers: frame;	k: integer;	Кластеризация объектов из data на основе алгоритма k-средних, число кластеров определяется параметром k
factor.r	data: frame; k: integer;	factors: frame;		Выполняет процедуру факторного анализа на выборке данных data; число факторных столбцов-нагрузок в factor есть переменная k
ttrees.r	data: frame;	id: integer; cluser: integer;	Нет	Кластеризация на основе критерия доли объясненной дисперсии
OR				
knapsack.r	Item: char; Weight: integer; Price: integer;	Item: char; Price: integer;	W: integer	Решает задачу о ранце: выбирает предметы из Item, суммарный вес Weight которых не превышает параметра W, максимизируя стоимость Price
tsp.r	Dist: frame;	Path: char;	Нет	Задача коммивояжера: ищет наиболее дешевый обход всех узлов графа. На входе – матрица расстояния, на выходе – путь
SubOptimal				
Statistics				
pnorm.r	x: double;	q: double;	u: double; s: double;	Вероятностная функция нормального распределения
qnorm.r	x: double; p: double;	q: double;	u: double; s: double;	Ищет квантили значений указанных в векторе x для нормального распределения
quantile.r	x: double; p: double;	q: double;	Нет	Ищет квантили значений указанных в векторе x для дискретного распределения
Statistics/Basic				

cov.r	data: frame;	cov: frame;	Нет	Вычисляет матрицу ковариаций для числовых признаков выборки data
mean.r	sample: double;	mean: double;	Нет	Вычисляет среднее mean по выборке sample
Statistics/Criteria				
fisher.r	sample: frame;	val: double;	Нет	Вычисляет значение критерия Фишера для представленной выборки sample
student.r	sample: frame;	val: double;	Нет	Вычисляет значений критерия Стьюдента для представленной выборки sample
chi.r	sample: frame;	val: double;	Нет	Вычисляет значений критерия Хи-квадрат для представленной выборки sample
kolm.r	sample: frame;	val: double;	Нет	Вычисляет значение статистики Колмогорова для выборки sample

Информацию о методах, присутствующих в архиве планируется поддерживать в актуальном состоянии по адресу [http: //sibirica.spsl.nsc.ru].

3.7. Техника разработки стратегии.

Наиболее интересный для исследователя инструмент – визуальный редактор стратегий.

Кликнув на вкладку «Стратегии» в боковом меню, пользователь найдет уже знаковый ему файловый архив по работе с методами и данными, однако содержащий уже исследовательские стратегии. Все те же операции, все та же навигация.

Мы будем действовать на примере. Поставим модельную задачу: создать стратегию, которая будет сравнивать две случайные независимые выборки и строить по ним матрицу корелляции.

Сейчас нам понадобится опция «создать файл». Создадим файл стратегии «compareRandom.rs»в общем доступном архиве в секции examples. Полный путь до файла - /public/examples/compareRandom.rs (рис.40:

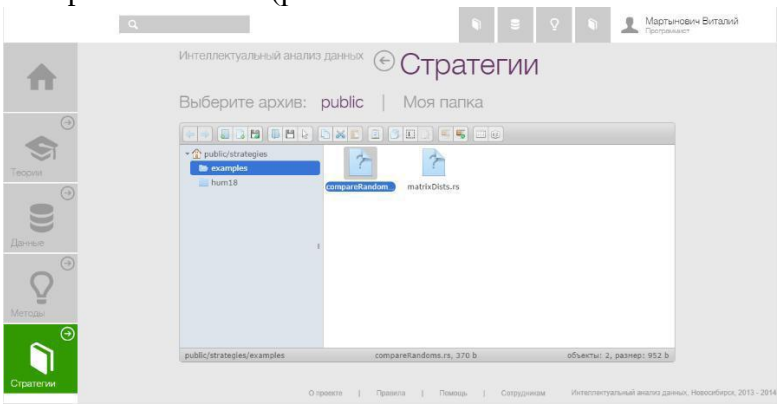


Рис 40. Архив стратегий

Двойной щелчок по вновь созданному файлу откроет визуальный редактор стратегий (рис.41).

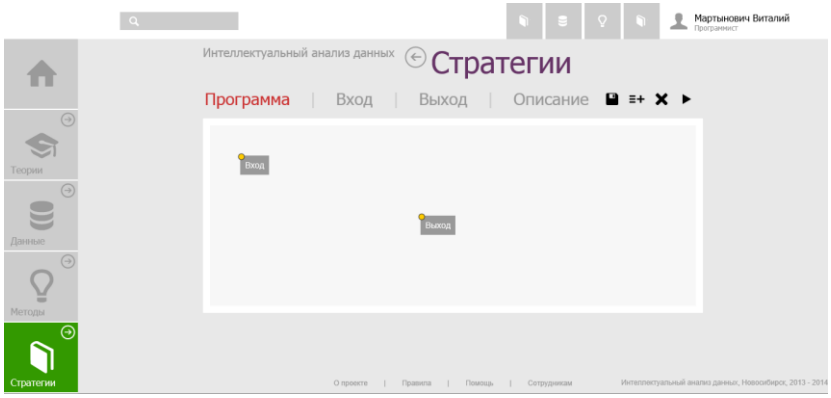


Рис 41. Начальный вид визуального редактора стратегий

Общая схемы работы с редактором состоит из следующих операций:

- Описание входа и выхода стратегии;
- Добавление методов на канву;
- Установка связи между методами;
- Задание соответствий на входах-выходах методов;
- Задание параметров запускаемых методов;
- Добавление справочной и обзорной информации о стратегии;
- Сохранение стратегии.

Все окно редактора поделено на три рабочих области:

1) Меню редактора. Навигация, как и в других разделах проекта, осуществляется переключением вкладок меню. Кроме того, здесь расположены кнопки для сохранения текущей стратегии, добавления метода на канву, удаления метода с канвы, а также кнопка запуска стратегии на исполнение (порядок сохранен);

2) Канва методов. Задача пользователя – расположить методы на этой канве таким образом, чтобы получить граф выполнения для конструируемой стратегии. Для этого он добавляет методы, перемещает их и соединяет друг с другом;

3) Панель свойств. Отображается в случае, если выбран метод или связь между двумя методами. Соответственно, становятся доступными таблицы параметров метода или схема передачи переменных между методами для заполнения;

Наиболее интересной является канва редактора. При запуске, как видно на Рис. 42, у стратегии, как и у всякого выполняемого элемента, присутствуют вход и выход. Схематично в редакторе это изображено двумя одноименными блоками.

4) Каждый серый блок – это отдельный метод, помещенный внутрь стратегии. Вход и выход можно понимать как методы, формирующие вход стратегии и разбирающие данные на её выходе. Клик по блоку с методом сделает этот блок активным для редактирования параметров метода в подокне «Панель свойств» ниже. Блоки можно свободно перемещать по канве, кликнув и зажав левую кнопку мыши (Drag&Drop) (рис.42);

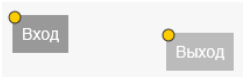


Рис. 42. Активный блок «Выход» подсвечен ярко-серым.

5) В левом верхнем углу каждого блока можно обнаружить желтую метку. Метка служит для установления связи между методами. Для того чтобы задать зависимость между входом одного метода и выходом другого, пользователь зажимает левую кнопку мыши на метке исходного метода и устанавливает конец возникающей стрелки на другой метод (рис. 43);

Рис. 43. Установление связи между методами

6) Каждый установленная связь между методами отображается в виде односторонней стрелки. Стрелка является активным элементом и при клике предлагает пользователю задать соответствие между входами одного метода и выходами другого (рис. 44):

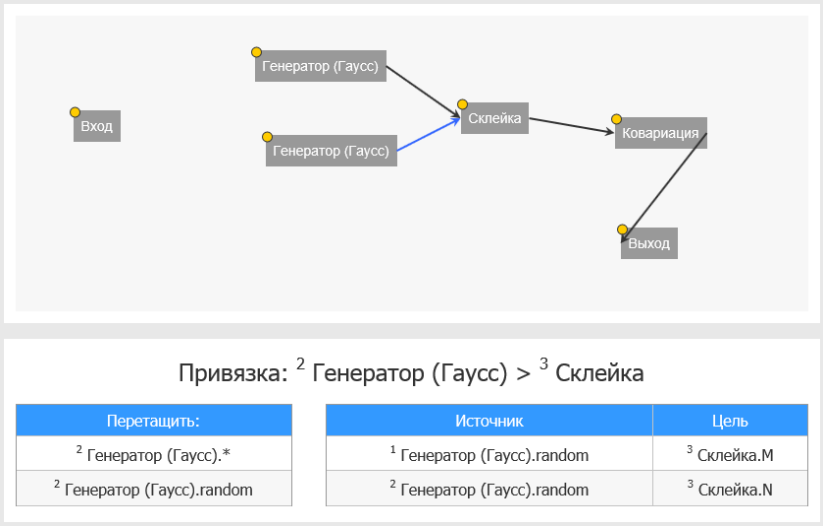


Рис. 44. Окно задания соответствия для активной связи.

Соответствие задает, какие выходные переменные исходного метода на какие входы целевого метода будут переданы при исполнении стратегии. Таблица слева содержит все выходные переменные плюс элемент, обозначенный как «*», который агрегирует весь выход исходного метода в одну таблицу. В целом установка соответствия аналогична процедуре выбора данных при запуске метода.

Методы «Вход» и «Выход» обеспечивают связь стратегии с миром данных. Для того, чтобы использовать переменные этих методов, следует задать описание входа и выхода стратегии на вкладках из меню стратегии. Мы не будет подробно останавливаться на этом моменте.

Для добавления методов на канву пользователь использует кнопку «Добавить метод» в меню редактора. После этого появляется диалоговое окно с до боли знакомым архивом:

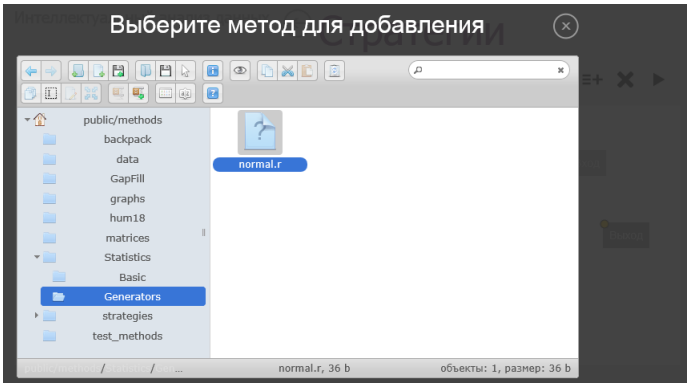


Рис. 45. Окно задания соответствия для активной связи.

Я выбрал метод /public/Statistics/Generators/normal.r. Взглянув на таблицу 4, нетрудно понять, что это генератор выборки из N элементов для нормального распределения.

Что нам потребуется для реализации поставленной задачи? Генеральный план состоит в раздельной генерации двух выборок u и v, а затем построении матрицы ковариации на векторах u и v. Поскольку библиотечный метод cov.r, строящий матрицу ковариаций работает с табличными данными, нам также понадобится вспомогательный метод «Склейка», сшивающий два вектора (или таблицы) в одну большую таблицу. Итого: две копии метода /public/Statistics/Generators/normal.r, /public/data/compose.ri /public/Statistics/Basic/cov.r. Добавив их, получим следующую картину (рис.46):

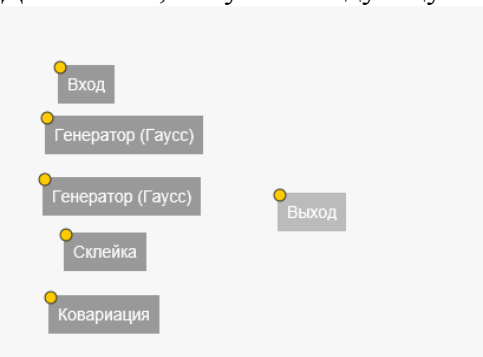


Рис. 46. Подготовленные для стратегии методы.

Всю стратегию согласно рассуждениям выше можно разбить на следующие этапы:

1) Генерация случайных выборок u и v;
2) Склейка случайных выборок u и v в таблицу M;
3) Вычисление ковариации переменных из таблицы M, получение матрицы ковариаций Cov;
4) Выдача Cov на выход стратегии;

Расположим методы согласно перечисленным этапам на канве редактора (рис.47):

Рис. 47. Перемещенные методы в порядке этапов исполнения

Однако методы которые мы добавили все ещё «висят в воздухе». Следует связать их между собой. Генераторы случайной выборки должны отдать свои ряды методу склейки, а склейка в свою очередь должна обеспечить методу «Ковариация» таблицу с двумя столбцами, по одному для каждой выборки. В конце «Ковариация» посылает результат работы стратегии на «Выход».

Рассмотрим, например, задание связи между «Генератор (Гаусс)» и «Склейка». Остальные связи устанавливаются аналогично:

1) Перетащим желтый маркер «Генератора» на блок с надписью «Склейка».
2) После этого кликает на появившуюся черную стрелку. Связь должна стать активной, что отмечается изменением цвета стрелки на синий (рис.47):

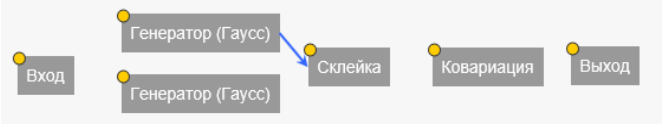
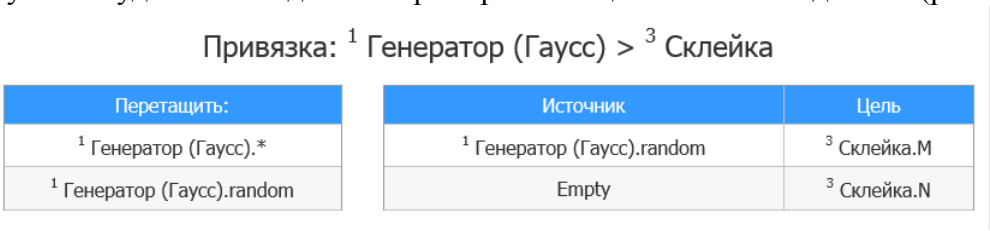


Рис. 48. Выбор связи для настройки

3) Заполняем таблицу связи. Без этого шага связь методов ничего не дает, и по существу является фиктивной. Система избавится от фиктивных связей при сохранении стратегии путем из удаления. В данном примере таблица связей выглядит так (рис.49):


Привязка: [1] Генератор (Гаусс) > [3] Склейка

Перетащить:	Источник	Цель
[1] Генератор (Гаусс).*	[1] Генератор (Гаусс).random	[3] Склейка.M
[1] Генератор (Гаусс).random	Empty	[3] Склейка.N

Рис. 49. Привязка выхода ко входу для связанных методов

Теперь последний штрих: зададим выход стратегии и её описание. Вход стратегии оставим пустым, так как наша модельная стратегия ничего не читает со входа. Оставим читателю эту часть редактирования стратегии в качестве простого упражнения.

После того, как все готово, сохраним нашу стратегию. Если будет обнаружена не состыковка в типах значений, система предупредит нас и выдаст соответствующее уведомление, давая возможность пользователю исправить обнаруженные неточности. На изображении ниже, я специально установил неверный тип данных для выходной переменной data (рис.50):

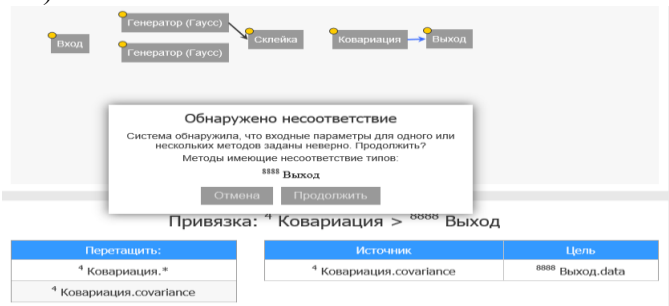


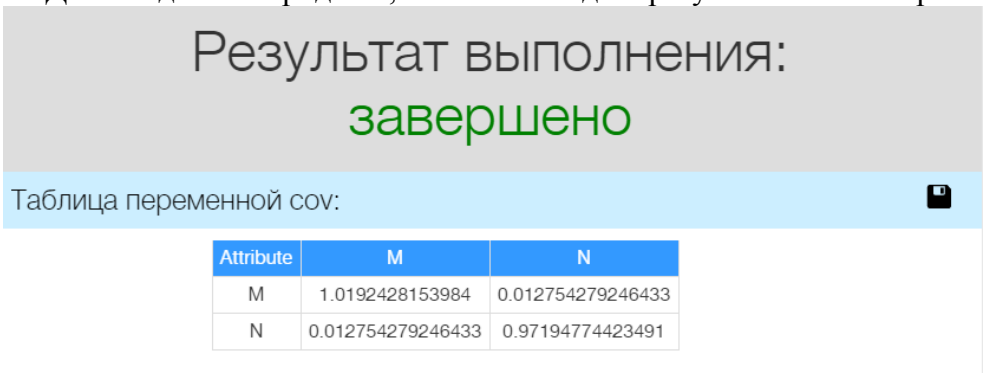
Рис. 50. Не состыковка типов данных при сохранении стратегии

Теперь стратегия готова к использованию. Можно убедиться, что все в порядке, если найти созданную стратегию в архиве и открыть её в визуальном редакторе ещё раз.

3.8. Работа с результатами анализа данных.

Пользователю может быть неинтересно внутреннее устройство методов или стратегий, но ему непременно должно быть важно, что же он получит в итоге. Возьмем, к примеру, только что созданную стратегию compareRandoms.rs из архива public/examples.

Используя инструкцию из сценария 3 «Запуск», отправим нашу стратегию на исполнение. Долго ждать не придется, и система выдаст результат нам на экран:



Attribute	M	N
M	1.0192428153984	0.012754279246433
N	0.012754279246433	0.97194774423491

Рис. 51. Результат выполнения тестовой стратегии.

Все результаты, согласно типизации элементов вывод OutputEntry можно разделить на четыре группы. Каждая из этих групп имеет свои особенности при отображении пользователю на странице «Результаты»:

1) Value (значение) – отображаются в виде секции «Значения переменных». Все значения собираются в один большой блок данных, в таблицу из двух столбцов. В первом столбце будут перечислены имена выходных переменных, а во втором – их значения;
2) vector (вектор) – составляют единую секцию «Сводная таблица». Каждый вектор занимает в ней свое место в виде отдельного столбца; имена столбцов выбираются согласно именам переменных из спецификации вывода метода / стратегии;
3) frame (таблица) – каждая переменная будет представлена в виде самостоятельной секции, содержащей данные переменной. Такие данные всегда будут прямоугольной таблицей, какой она была внутри метода её сгенерировавшего;
4) list (список) – не отображается на странице результатов. Список является вспомогательным типом данных, предназначенным для взаимодействия сложных методов внутри стратегии;

Каждая секция состоит из следующих частей: заголовок, кнопка сохранить и содержимое. Заголовок говорит об имени переменной, которая будет выведена. В содержимом всегда будет таблица одного из трех типов описанных выше.

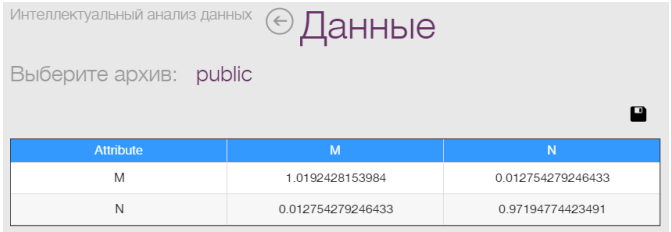
Веб-интерфейс предназначен для предварительного просмотра данных анализа, чтобы исследователь мог решить, повторить ли счетный эксперимент, выбрать ли другие параметры, или стоит все же использовать текущие результаты для подведения итого и совершения какого-либо рода выводов.

Для последнего существует кнопка «Сохранить», расположенная в заголовке каждой секции результатов. При её нажатии браузер предложит сохранить файл с результатами на компьютер пользователя. Его же можно повторно разместить в системе для совершения других операций. В нашем случае файл результатов для переменной вывода «cov» будет примерно следующего содержания (рис.52):

	A	B	C
1	Attribute	M	N
2	character	double	double
3	M	1.0192428153984	0.012754279246433
4	N	0.012754279246433	0.97194774423491

Рис. 52. Элемент вывода «cov»

Для наглядности я открыл скачанный документ с результатами cov.csvв Microsoft Excel. Однако, это обычный текстовый файл в формате CSV (Comma Separated Values), в котором записаны строки таблицы с разделителем «;» между двумя ячейками одной строки.



Attribute	M	N
M	1.0192428153984	0.012754279246433
N	0.012754279246433	0.97194774423491

Рис. 53. Размещенный в архиве файл с результатами вычисления стратегии

Единственной особенностью можно назвать тот факт, что вторая строка является специальной и описывает типизацию столбцов в таблице данных согласно схеме типов системы интеллектуального анализа данных. Таким образом, этот файл является «родным» для системы и может быть использован в качестве ввода для любого из методов в архиве.

IV СТРАТЕГИИ РЕШЕНИЯ АРХЕОЛОГИЧЕСКИХ ЗАДАЧ

4.1.Анализ данных черепов неандертальцев и построение классификации

Настало время перейти к реализации настоящих стратегий.

В качестве задачи была выбрана все та же стратегия, описанная в [Холюшкин, Витяев, Костин, 2013]. Она была выбрана потому, что большая часть подготовительной работы уже проделана авторами этой монографии.

Напомним, что исходным материалом служили данные и выводы монографии [Деревянко, Холюшкин, Ростовцев, Воронин, 2001]. Задача звучит примерно следующим образом: по исходным, неполным исходным данным, сведенным в таблицу измерений необходимо получить такое разбиение данных на группы, чтобы оно объясняло и подтверждало выводы [Холюшкин, Витяев, Костин, 2013], а также вычислить расхождение двух этих классификаций (исследовательской и алгоритмической). Таблица исходных данных приведена ниже (см. таблицу 5).

Сначала построим план работы стратегии.

1) Поскольку исходные данные являются неполными, а большинство существующих методов классификации подразумевают работу с цельными объектами, необходимо дополнить данные, заполнив пропуски в исходной таблице;

2) На дополненной таблице мы проведем линейный дискриминантный анализ, чтобы показать возможность разделения данных на группы;

3) С помощью алгоритма кластеризации TTrees получаем разбиение выборки Filled на группы, которые затем объединяются в теоретически предсказанные типы (получаем список групп и список типов);

4) Сравниваем разбиение, полученное на шаге 3 с теоретическим разбиением, в результате чего получаем статистику расхождений по предсказаниям типов содержащихся в таблицах групп и типов;

5) Выдаем полученную статистику на экран и завершаем вычислительный эксперимент, переходя к описательно-аналитической части;

Предлагаемый план работы стратегии без труда можно преобразовать в будущую схему стратегии.

Согласно спецификации, стратегия – это набор соединенных между собой блоков-методов, имеющая свой вход и выход, притом каждая связь между блоками-методами описывает какой из выходов исходного методами, каким из входов целевого метода соответствует. Поэтому логично представить схему функционирования стратегии (или просто «схему») в виде набора взаимосвязанных методов, указывая соответствия выход-вход для каждой связи.

По схеме стратегии практически на автомате строится стратегия в визуальном редакторе. Сценарий подробно описан в разделе «Пример разработки стратегии». Суть идеи в том, что следует добавить одноименные блоки на канву стратегии, расположить их аналогично схеме и достроить граф стратегии до графа изображенного на схеме.

Для примера можно рассмотреть схему стратегии из раздела «Пример разработки стратегии». На блок-схеме помимо графа стратегии представлены соответствия выхода-входа (табл.5):

Таблица 5. Исходные данные для стратегии анализа черепов

ип		Имя		ол	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
		Сванскомб	325							42	25							17			24	0	0						16	16	7	0		08	4	1	8				8.0							4.4							3.1	1.0	1.6	6.0	5.0			
		Гибралтар	300		90	83	86	76	75	46	17		3	5	9	06	07	25			10	8			22	76	42	24		06	0	6	07		2	5	5	6.8	1.6	9.0	0.1	3.7	5.7	3.5	3.3	5.6	7.3	13.6	5.6	1.0	6.3		3.2	6.3		7.4	1.7	7.8		4		
		Саккопасторе 1	200		82	73	75	73	76	42	09	07	01	9	1	02	01	16	25	16	17	5	8	20	03	82	38	10	6	07	5	2	06	3	7	1	9	8.0	9.9	5.5	6.8	1.1	5.1	9.5	1.1	7.1	6.3	9.2	3.8	0.2	8.2	5.6	7.8	6.4	6.5	1.3	2.7	4.2	0.0	1	2	0
		Саккопасторе II	300													14			30	16	12																																									
		Монте-Чирчео II	550		04	93	98	85	84	55	23	25	11	9	2	15	06	27	45	28	24	3		90		60	61	31	17	13	1	2	17	09	8	3	0	6.0	0.3	4.4	9.4	1.6	5.0	8.1	8.4	3.5	5.5	02.4	6.5	1.9	7.4	3.4	3.1	9.3	3.2	7.9	6.9	6.2		2	7	5
		Фонтешевад	470		95					54	17		08								26																	9.0	0.0	5.4	6.0	0.1																				
		Ля-Шапель	626		08	93	97	90	88	56	30	31	11	0	1	23	09	24	32		31	9	2	03	63	15	56	21	19	16	4	3	07	12	1	6	2	5.0	2.5	3.4	3.3	1.2	5.7	6.8	9.9	7.9	3.2	4.7	8.4	4.6	1.4	3.9	3.8	8.4	4.1	8.5	9.2	7.7	5.3	3	6	3
		Ляферасси 1	641		08	96	95	94	90	58	35	38	14	3	8	20	09	21	43	35	25	0	4	92	63	22	73	34	21	18	7	1	17	13	7	2	1	6.0	4.9	4.8	5.4	2.2	7.7	9.9	9.0	0.1	7.2	6.8	5.8	2.2	6.3	4.3	6.6	7.3	3.4	2.2	2.5	00.0	5.0	6	3	9
		ЛеМустье 1	564		96	88	89	84	82	50	28	27	14	1	6	24	07	21	33						34			20	22		3		08	09		7		6.5	5.3	8.2	5.3	6.0	8.2	0.4	1.3	8.4					5.1	5.6		0.0	2.6		0.5			9	2	0
	0	Спи 1	562		01	87	01	86	87	44			15	3	4		01		24		21			89	39	00		10	26		9		03	15		5		1.6		7.2		9.9	1.3	9.0	0.1		3.5				1.2	7.2		3.6	1.3		3.2			9	0	7
	1	Спи II	600		00	85				53			15				08	26	35		31				46	16			15		5			09		3		6.5		7.5		5.2			0.6	5.7	2.4	6.2				4.5			4.8		6.4					
	2	ЛяКинаУ	350		02	87	99	85	84	38			13	9	7		01	08	12		12			58	24	02		16	07		6		06	03		8		7.7		5.4		1.9	9.7	5.4	3.2	3.5	0.2	6.4			2.0	0.5		1.4	6.3		7.9			0	9	8
	3	Крапина D	450		98					69							10																					5.4							5.1																	
	4	Крапина С	200		78					49			05				9																					3.7		9.0		0.5			6.4																	
	5	Эрингсдорф IX	450		96	90	92	82	78	45				6	7		13	21			05						80	35	28	17	6	2	15	19	7	8	2	4.0					0.0	6.8	7.9	3.4	07.6	15.2	1.6		8.7	0.7	4.4	5.2	3.0	4.4	7.9	00.0		4	2	9
	6	Неандерталь	450		99	86	98	85	83	47				3	4		05	22						78	38			33	10		7		17	03		4		3.9					1.9	9.2	1.4	6.1					8.8	2.0		8.0	3.6		4.7			2	6	4
	7	Штайнхаим	070		85	73	79	73	68	32	11	13	8	5	1	9	02	18	16	07	07	9		46		00	41	20	08	14	0	4	00	7	9	5	3	1.4	0.0	3.0	4.1	4.2	7.5	9.5	7.3	6.4	5.3	10.3	4.3	9.0	4.0	2.4	8.1	3.3	9.8	8.1	2.9	7.7		9	4	0
	8	Гановице	320		00					45	15																											2.5	7.5		9.3																					
	9	Петралона	220		09	94	06	96	90	49	28			3	4	10	11	17	38					97		08	72	30	14	28				06	4			1.3	1.2		5.9		0.3	7.6	4.5	4.9			6.2	9.6		0.7	5.0		3.0	3.4				5	3	6
	0	ДжебелИрхуд 1	480		98	90	90			45	25			3						12				82	70			11	14	13								3.2	3.1		6.2		3.7																	7	6	0
	1	ДжебелИрхуд II	450		97		93			48	18			4					15			7	0	78	66								18	13	2			5.1	9.9		9.7		3.5								9.9	7.4	6.7						1.1			
	2	Ньярасса	100		91					42	09		8				1		32		32	0	4															4.4	7.1	1.3	6.8	9.0			4.1		8.9												5.0			
	3	Брокен-Хилл	325		09	88	09	99	93	45	31	30	06	0	2	12	8	19			31	0	9				73	38	17	18			21	12	9			9.4	2.7	0.7	0.3	3.1	8.3	1.2	7.6	2.4	4.8	0.8	6.0	0.0	7.9	3.6	2.6	7.7	5.7	5.4			2.5	0	8	5
	4	Салданья	225		00	85	00	92		44				4			02																	10				2.0					2.0		0.8															3		7
	5	Схул IV	554		06	93	02	99	97	48	28		12	03	6	10	06	21			32	0			80	15	88	32	34	22			18	22	6			1.8	2.1	4.4	6.5	5.7	1.0	3.2	1.6	7.6	0.3	1.7	3.1	8.4	7.3	9.2	1.8	9.4	1.0	0.5				7	4	1
	6	Схул V	588		92	79	90	83	74	44	26		12	5	4	8	00	14			22				23	03	77	21	37	19			06	23	6			5.0	5.6	8.3	7.5	7.8	0.0	5.0	9.4	7.7	2.0	3.4	0.9	6.0	5.2	4.1	0.0	7.6	9.8	0.7				8	6	1
	7	Схул VI	400							44											40																																									
	8	Схул IX	610		13	00	12	98	93	45	30	31	13	7	3	15	6	20			20				60	20	79	30	20	29			14	12	5			8.1	1.0	3.0	9.7	7.9	1.3	6.8	6.2	0.0	0.0	00.0	6.2	0.3	3.5	2.6	4.6	7.7	3.3	3.6				2	8	4
	9	Табун 1	271		83	73	79	74	74	41	15	15	8	4	7	08	8	22			20	6	8		00	92	33	07	17	08			6	05	0			7.0	2.8	3.5	1.6	9.5	6.9	7.0	9.5	0.3	0.3	01.7	5.0	2.4	2.5	7.4	9.2	9.7	9.7	3.3			7.8	8	7	4
	0	Зуттие	400							38			16				7	13										25					13									4.1			0.3	5.8								0.4								
	1	ДжебелКафзех VI	568		96	90	90			45			17	01			10	25			23			68	50		68	33					14					4.0		9.7		0.7	3.2		5.9	8.0	8.0	01.6	3.3		8.2			5.7						9	1	7
	2	Амуд 1	750		15	04	08	02	02	54	39		21	01	9		15	24	39		24			08		33	85	35	20	30	7	3	20	12	05	5		1.6	4.6	6.3	0.3	8.6	8.6	9.2	4.7	2.7	2.7	00.0	5.8		5.8	2.1	8.8	8.9	3.3	0.8	6.2			1	3	1
	3	Шанидар 1	600		07	97	01	99	99	52	33		16	02	8		10	25	33		25			91		09	73	17	34	22	2	0	09	28	7	5		3.4	4.3	6.0	7.5	6.3	0.7	4.2	2.4	8.0	8.0	00.0	5.5		2.7	1.8	6.9	3.2	5.5	9.5	0.3			2	7	4
	4	Мапа	225		96																							34					16																		9.2			6.6								5
	5	Тешик-Таш	545		98		87			51	39		25	3		08	04	26	30		24	6	8	54		20	70	23	14	34			10	04	03			6.3	0.2	3.1	2.1	2.8			8.9	2.5	3.9		3.5	9.2				9.4	1.2	6.9				4		2

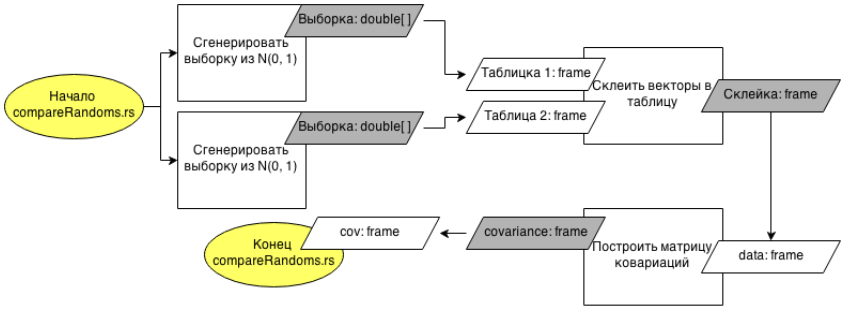


Рис.54. Схема тестовой стратегии.

Уточним план, предложенный выше, но уже в терминах схемы стратегии:

1) На вход стратегии (блок «Начало TTreeCluster.rs») подается переменная с именем data и типом frame (таблица);

2) Вход стратегии передает data методу «Заполнение пропусков»;

3) «Заполнение пропусков» дополняет данные (я буду использовать метод /public/FillGaps/LinearRegression.r, следуя выкладкам из [Холюшкин, Витяев, Костин, 2013]), генерируя выходную таблицу data типа frame;

4) «Заполнение пропусков» передает полную таблицу data методу «Линейный дискриминантный анализ»;

5) «Линейный дискриминантный анализ» вычисляет факторные нагрузки для выборки data и передает результат – factors типа frame–на выход;

6) Те же самые данные «Заполнение пропусков» передает в «Кластеризация, затем сборка»;

7) Наконец, «Заполнение пропусков» обеспечивает теоретической типизацией заключительный метод «Проверить кластеризацию»;

8) «Кластеризация, затем сборка» выполняет разбиение на группы, согласно алгоритму TTree, получая разбиение исходной выборки на группы groups, а затем пытается объединить их в типы types. Обе переменные groups и types являются списками. Метод расположен в архиве/public/clustering/ttrees.r;

9) «Кластеризация, затем сборка» передает эмпирически полученную типизацию types контрольному методу «Проверить кластеризацию» под именем classes;

10) «Проверить кластеризацию» сравнивает 2 типизации извлеченные из data и types и выдает таблицу-результат сравнения на выход (блок «Конец TTreeCluster.rs»);

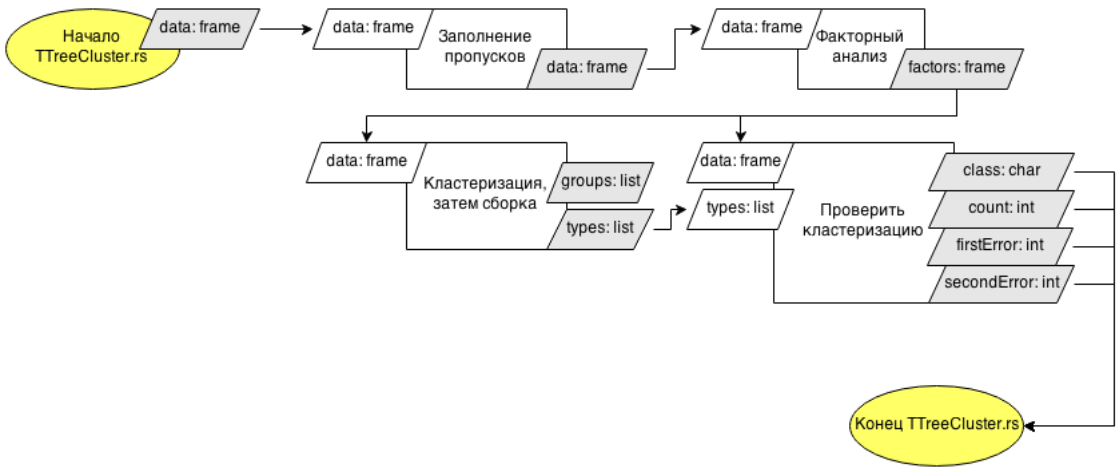


Рис. 55. Схема стратегии TTreeCluster.rs.

Разберем немного подробнее методы, используемые в рамках данной стратегии.

Первым на очереди метод является метод «Заполнение пропусков». Основой для метода является идея о том, чтобы представить целевой вектор значений u представить как линейную комбинацию базисных векторов x и y:

$v = \alpha x + \beta y + \gamma$ (1)

Рассматриваемый метод строит линейные модели вида (1), аппроксимирующие существующие (уже заполненные) значения вектора v. Для этого:

1) Строится источник линейной модели, задаваемый пользователем как база линейной регрессии в виде списка колонок исходной таблицы данных:

```
regressionNames = c(base1, base2);
regressionBasis = input[regressionNames];
```

2) Для каждой колонки с отсутствующими значениями вычисляется линейная регрессионная модель вида (1) с помощью R-функции `lm`:

```
for (name in names) {
  v <- input[[name]];
model<- lm(x ~ ., data.frame(x = v, regressionBasis));
```

3) Выполняется предсказание согласно полученной модели model и строится вектор предсказанных значений. За это отвечает функция R-языка `predict.lm`:

```
predictedValues<- predict.lm(model, regressionBasis);
```

4) Отсутствующие значения вектора v дополняются предсказанными значениями:

```
missed = is.na(v);
v[missed]<- ifelse(
     typeof(v) == "integer",
     as(predictedValues[missed] + 0.5, typeof(v)),
     as(predictedValues[missed], typeof(v)));
```

Линейный дискриминантный анализ построен на принципе разделения объектов с помощью введений нескольких (k) интегральных линейных показателей в k-мерном пространстве состояний R^k:

1) Из данных выбрасываются все нечисловые значения, а остальные приводятся к double:

```
toAnalyze<- data[,sapply(data, is.numeric)]
toAnalyze<-              as.data.frame(sapply(toAnalyze,
as.double));
```

2) Вычисляем факторные нагрузки для линейной дискриминации объектов в факторном пространстве, в этом нам поможет функция языка `lda`:

```
fa <- lda(Type ~ ., toAnalyze);
```

3) Строим векторы значений согласно построенным моделям факторных нагрузок:

```
factorVals <- predict(fa)$x;
```

Особое внимание следует уделить методу кластеризации. Авторский метод TTrees был предложен П.С.Ростовцевым [Ростовцев и др., 1994]. Суть идеи состоит в постоянном разбиении множества объектов на группы таким образом, чтобы это разбиение отличалось наибольшей объяснительной способностью в смысле критерия.

Разбиение является хорошим, если оно имеет максимальную объясненную, или межгрупповую дисперсию (и, соответственно, минимальную необъясненную, или внутригрупповую дисперсию). Отметим, что разбиения продолжаются до тех пор, пока размер группы не окажется пренебрежительно мал, либо группа станет однотипной с точки зрения теоретической таксономии.

Авторы попытались воспроизвести работу оригинального метода кластеризации при программировании метода-аналога на R языке:

1) Если размер текущей группы <=5 (в этом случае при разбиении получилась бы подгруппа неудовлетворительно малого размера =1) или группа полностью гомогенна, то она оставляется без изменений и возвращается «как есть»:

```
if (nrow(group) <= 5 | length(unique(group[[Y]])) == 1)
return(list(list(path=path, data=group)));
```

2) Иначе ищется оптимальное с точки зрения критерия дробление группы и для каждой подгруппы повторно запускается процедура дробления:

```
criterias<- sapply(X, varMax, group = group);
criteria<- criterias[,which.max(apply(criterias, …))];
dataPieces<- c(split(group[criteria$permutation,], …));
result<- list();
for (dataIndex in 1:length(dataPieces))
result<- c(result, breakGroups(…,dataPieces[[dataIndex]]));
```

3) Функция varMax из листинга п. 2 выполняет поиск оптимального разбиения для каждого признака присутствующего в переданной выборке. Результат возвращается в виде списка описаний, содержащих:

1. Сортировку объектов в виде отображения-подстановки (permutation);
2. Значение критерия объясненной дисперсии (fValue);
3. Размеры получающихся при дроблении групп (lengths);
4. Границы значений признака, разбивающие весь интервал значений признака на группы (borders);

```
permutation<- order(groups[[attribute]]);
dataVector<- groups[permutation, attribute];
borderCombos<- subset(expand.grid(
border2=3:(length(permutation)-3),
border3=5:(length(permutation)-1)),
    border3 - border2 > 1);
criterias<- apply(borderCombos, 1, checkGrouping,
target = groups[permutation, Y]);
opt<- criterias[[which.max(sapply(criterias, …)]];
opt$borders<- cbind(
start=dataVector[c(1, cumsum(head(opt$lengths, -1)) + 1)],
end=dataVector[cumsum(opt$lengths)]);
```

В листинге borderCombos – это все возможные разбиения, по которым будет искаться оптимум, а apply строкой ниже получает значение критерия для всех таких разбиением поочередным применением функции checkGrouping к таблице разбиений;

4) Функция checkGrouping вычисляет межгрупповую дисперсию для представленного разбиения напрямую:

```
fVal<- 0; start_next<- 1; avg<- mean(target);
lengths = c(
grouping[['border2']] - 1,
grouping[['border3']] - grouping[['border2']],
length(target) - grouping[['border3']] + 1);
for (group in 1:length(lengths)) {
start<- start_next;
start_next<- start + lengths[group];
part<- target[start:(start_next - 1)];
fVal<- fVal + lengths[group] * (avg - mean(part)) ^ 2;
}
```

Метод «Кластеризация, затем сборка» немногим отличается от описанного выше метода. Помимо операций разбиения исходной выборки на группы, присутствует часть кода, отвечающая за сборку воедино тех групп, которые относятся к одному типу. В итоге получается цельный список групп, образующих эмпирический тип.

```
for (group in groups) {
groupData<- group$data;
counts<- as.data.frame(table(groupData[['Type']]));
classIdx<- counts[which.max(counts[,'Freq']), 'Var1'];
classIdx<- as.character(levels(classIdx))[classIdx];
ids<- as.integer(rownames(groupData));
foundClasses[[classIdx]] <- c(foundClasses[[classIdx]], ids);
}
```

«Проверить кластеризацию» не заслуживает пристального внимания. Метод сравнивает две таксономии и выдает числовые показатели, описывающие расхождения в количествах объектов, попадающих под тот или иной тип.

Запрограммировав все необходимые методы, можно приступить к преобразованию схемы стратегии с Рис. 55 в исследовательскую стратегию в рамках системы анализа данных. Для этого следует обратиться к руководству раздела «Техника разработки стратегии».

Подведем некоторый итог. Что у нас на входе? Согласно схеме Рис. 55, при запуске стратегия получает набор данных data: frame с пропущенными значениями. А на выходе? После исполнения, стратегия должна вывести нам на экран сводную таблицу со сравнением двух классификаций: теоретической и эмпирической. В таблице должны присутствовать целочисленные колонки class, count, firstError, secondError.

Что ж, можно и проверить. Для этого выберем файл с исходными данными. Это таблица 3, несколькими страницами выше. Чтобы стратегия могла нормально функционировать необходимо загрузить файл с этой таблицей на сервер, воспользовавшись архивом в разделе «Данные» (рис.56):

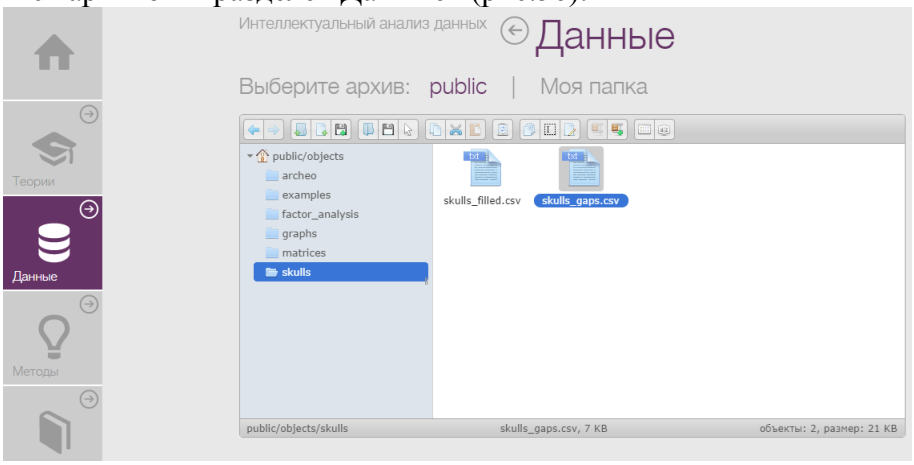


Рис. 56. Загрузка данных для обработки стратегией.

Запускаем стратегию на свеже выгруженных данных.

Результат работы стратегии представлен на изображении ниже. В целом, это именно то, чего мы ожидали, исходя из схемы стратегии (рис.57):

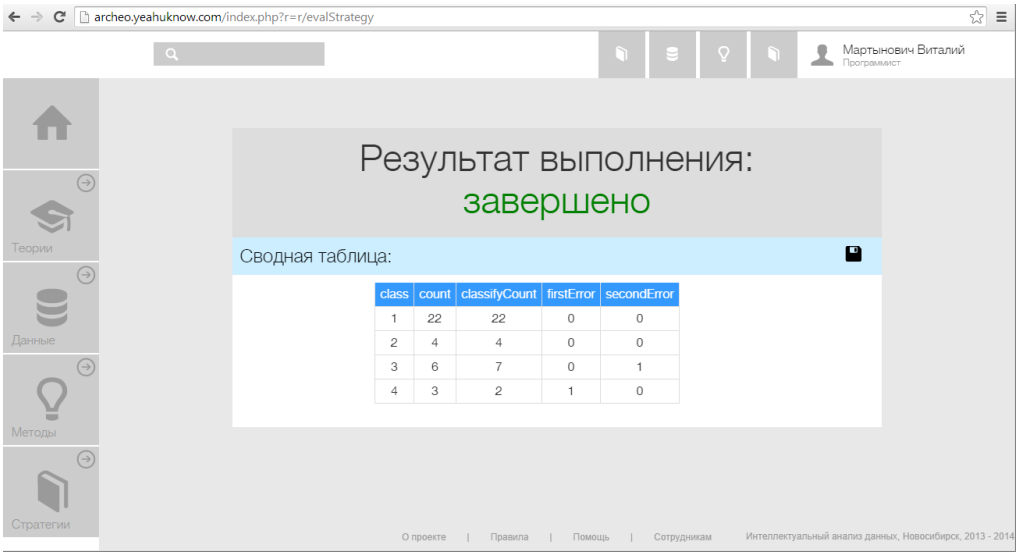


Рис. 57. Результаты анализа данных по черепам.

Из протокола можно увидеть, что почти все объекты были распределены правильно. Ошибочно определен только один объект, попавший вместо теоретически предсказанного типа 4 в тип 3. Все остальные объекты заняли своё место в эмпирической таксономии, совпадающей с таксономией теоретической.

Данные по классификации черепов: (указаны идентификаторы элементов исходной таблицы)

Тип `1` 4 22 28 29 16 7 10 5 14 12 18 2 6 13 21 8 11 15 3 9 17 20
Тип `2` 23 19 34 24
Тип `3` 1 27 30 35 25 31 26
Тип `4` 33 32

Таким образом, можно сказать, что теоретическая таксономия является абсолютно корректной и обоснованной. Заинтересовавшийся читатель может получить обоснование полученной иерархии групп, используя алгоритм TTrees из одноименного метода системы. TTrees дает разбиение исходной выборки в виде ветвящегося дерева, где каждая группа – это лист дерева, а метки на пути дерева позволяют восстановить «объяснение» произведенного разбиение в виде накладываемого на тот или иной признак ограничения. Ниже можно увидеть пример такого разбиения, произведенного методом /public/Clustering/TTrees.r (рис.58):

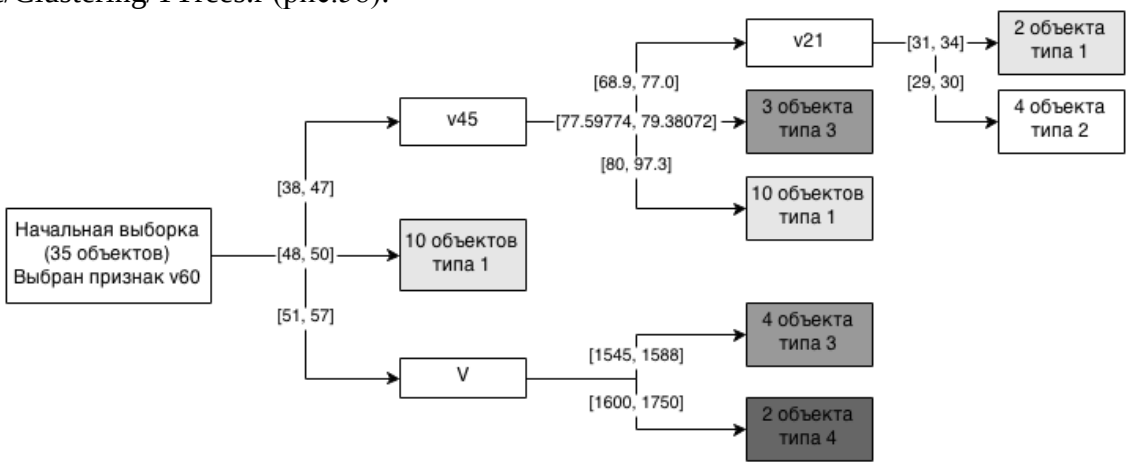


Рис. 58. Разбиение на группы для исходных данных.

В листьях дерева содержится описание полученных групп. Метки на не листовых узлах в совокупности с метками на ветвях дерева дают описание группы в терминах

исходных признаков. Цветом подсвечены группы, которые впоследствии будут собраны в единый тип.

4.2.Программирование стратегии для исследования данных по памятникам палеолита

Следующая стратегия нацелена на классификацию палеолитических орудий, найденных в различных локациях времен палеолита. Целью разработки стратегии будет демонстрация возможностей системы в автоматизация процессов преобразования данных в вид пригодных для анализа и совершения выводов, а также предоставления примера программирования сложных нестандартных методов системы интеллектуального анализа данных.

Для понимания и установления природы и характера различий палеолитических комплексов большое значение имеет исследование структурных характеристик археологических данных. Основой для исследований служит типологическая таблица, подготовленная авторами оригинальной работы [Деревянко и др., 2005] на основе данных, собранных по материалам публикаций (см. таблицы 4 и 5). Совокупность типы орудий – локации будут объединены в классы, объясняющие их схожесть. На основе полученной классификации могут без труда быть получены выводы о культурных и семантических взаимосвязях рассматриваемых объектов.

Теперь можно сформулировать исследовательскую задачу, а вместе с ней и задачу разрабатываемой стратегии. Имея набор типов орудий (таблица 6) и данные по количеству типов орудий в перечисленных локациях (таблица 7) получить классы орудий, связывающие родственные типы и культурно близкие локации. Класс должен представлять собой набор пар (тип орудия, локация), а также правила, описывающие внутреннюю структуру такого класса.

Таблица 6. Список орудийных комплексов Ближнего и Среднего Востока и Кавказа.

	леваллуазские сколы,	7	скребки,	3	рабо,
	леваллуазские острия,	8	резцы,	4	орудия с черешком,
	леваллуазские ретушированные острия,	9	проколки,	5	чопперы,
	псевдолеваллуазские острия,	0	ножи,	6	чоппинги,
	мустьерские острия,	1	ракле,	7	разные,
	лимасы,	2	усеченные отщепы,	8	бифасы листовидные,
	продольные скребла,	3	транше,	9	угловатые ,
	двойные скребла,	4	выемчатые,	0	долотовидные,
	конвергентные скребла,	5	зубчатые,	1	клювовидные,
0	угловатые скребла,	6	резцовые острия,	2	бифасы,
1	поперечные скребла,	7	сколы с брюшковой ретушью,	3	микроорудия,
2	скребла с брюшковой ретушью,	8	сколы ретушированные со спинки	4	диски,
3	скребла с крутой ретушью,	9	остроконечники тейякские,	5	кливер
4	скребла с утонченной спинкой,	0	треугольные орудия с выемкой,	6	пластины с притупленным краем
5	двусторонние скребла,	1	псевдорезцы,	7	NachrIbragim.
6	скребла с противолежащей ретушью,	2	сколы с выемчатым концом,		

Таблица 7. Список орудийных комплексов Ближнего и Среднего Востока и Кавказа

1	Амуд B4	14	ВарвазиA	27	Ябруд 4	40	Монашеская	53	Ортвала-Клде V
2	Амуд B2	15	Варвази B	28	Ябруд 5	41	Губский Навес	54	Ортвала-Клде VI
3	Кеу I сл. l	16	Варвази C	29	Ябруд 6	42	Малая Воронцовская	55	Ортвала-Клде VII
4	Кеу I сл. II	17	Варвази D	30	Ябруд 7	43	Таглар 2 сл.	56	Двойной Грот

5	Кеу I сл III	18	Сефуним А	31	Ябруд 8	44	Таглар 3 сл.	57	Азых 3 сл.
6	Кеу I сл V	19	Сефуним 12	32	Ябруд 9	45	Таглар 4а	58	Среднехаджохская
7	Кзар-Акил XXVIA	20	Сефуним 13	33	Ябруд 10	46	Таглар 4б	59	Азых 6 сл.
8	Кзар-Акил XXVIB	21	Сефуним VI	34	Кударо I 3а	47	Таглар 5	60	Медвежье
9	Кзар-Акил XXVIIA	22	Сефуним VII	35	Кударо I 3б	48	Таглар 6	61	Лусакерт D
10	Кзар-Акил XXVIIB	23	Сефуним В	36	Кударо I 3в	49	Ортвала-Клде I	62	Лусакерт А
11	Кзар-Акил XXVIII	24	Сефуним С	37	Кударо I 4	50	Ортвала-Клде II	63	Газма
12	Кзар-Акил XXVIII	25	Ябруд 2	38	Каркустакау	51	Ортвала-Клде III	64	Баракаевская
13	Кунджи	26	Ябруд 3	39	Тамарашени	52	Ортвала-Клде IV		

Основой для исследований служит типологическая таблица, подготовленная авторами оригинальной работы [Деревянко и др., 2005] на основе данных, собранных по материалам публикаций. Типы орудий были перенумерованы и представлены в таблице 6, мустьерские комплексы (локации) сведены в таблицу 7. В данных, используемых для исследовательской стратегии, содержатся уже лишь закодированные объекты, идентификаторы в первом столбце этой таблицы (таблица 8) соответствуют локациям из таблицы 7, идентификаторы в первой строке – это типы орудийных комплексов из таблицы 6.

Таблица 8. Распределение орудийных комплексов по локациям.

D									0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
																		1																				3											
	9	0					5		0																																					
	2																																														
	35	6					6	1	7	4	1																																				
	26	5	3	5	9		8	4	8	8	3						2								0																					
	7						8																																								
	0						6																		5																					
	8			8			3										2								0																					
0	1	6		1	9		0											6		0				8	9																				
1	6	9															0	1						1	4																						
2	4	0		0														9						9	0																					
3	4				1		93	3	5	4	0								7	6				1	9		4	6																	
4	8						0	5	6																2																					
5	1				7		19	3	6															1	0																					
6					5		1	9	5									0				0			3			0																	
7							8	2	0											0					2																					
8																																														
9																																														
0																																														
1	1																																													
2																																														
3																																														
4																																														
5																																														
6					8		3											0																												
7			1		3		6	6										0							7																					
8																		2	3						7																					
9					8		9																		7																					
0					7		1											6	3					4	5																					
1					7		4											9							1																					
2																		6						8	8																					
3																																														

4																																															
5																																															
6																																															
7																																															
8	2																			0					2																						
9	1																			1																											
0	02						2			4							8			5				4	1																						
1	5						5																		2																						
2	7																																														
3	54	8	5		7		0	8		3																																					
4	15	0	9		2		1	2		9																																					
5	83	4	5		1		9	0																																							
6	9		7				0																																								
7	6																																														
8	0						4																																								
9																																															
0	20				2		8	2	7	5							8	5		8				4	4			7																			
1	37				6		1	0	7	9				5			1	0		7				7	2			4																			
2	26	6	3		8		60	9	8	3	6						2	9	3	09		4		6	2			9				1					0										
3	65	7	4		4		25	4	0	3	0			7			2	6	3	5		8		5	5			09				4					4										
4	24				7		9	6	6					6			3	2		0				0	3			7				0					0										
5																																															
6																									2											0											
7	6	1	6	2	6		4	7		4										4		5		2												6											
8	1																																														
9							6	3	3	0	1	5					4							5										6													
0																																															
1	5																								1				9							1					8						
2																																															
3					5		0																																								
4					4		57	4	3		2						2	1						44	2											0				2							

План исследований в рамках стратегии может быть следующим:

1) Как видно из таблицы 8, многие ячейки оставлены пустыми. В данном конкретном случае это означает, что они просто опущены для простоты восприятия. Для работы алгоритма необходимо заполнить эти ячейки нулями;

2) Далее мы вычисляем степень родства строк исходной матрицы друг с другом. Родственные в смысле топологической близости в терминах стандартной Евклидовой метрики. В итоге получаем матрицу расстояний между строками матрицы Dist;

3) Dist используем для такой перестановки строк, чтобы они находились наиболее близкими друг к другу. Это позволит впоследствии объединить клетки матрицы в области, образующие кластеры. Задача перестановки строк сводится к задаче коммивояжера и решается стандартными субоптимальными алгоритмами;

4) Шаги 2-3 повторяем для столбцов исходной матрицы. В итоге получаем такие данные, где строки и столбцы располагаются наиболее близко друг к другу и могут быть объединены алгоритмом кластеризации;

5) Запускаем алгоритм кластеризации на матрице с выполненными перестановками;

6) Выдаем результат в виде таблицы, в которой в каждой клетке выписан номер класса, в который попадает пара (тип орудия, локация).

С точки зрения программной части работа стратегии будет выглядеть так (рис.59):

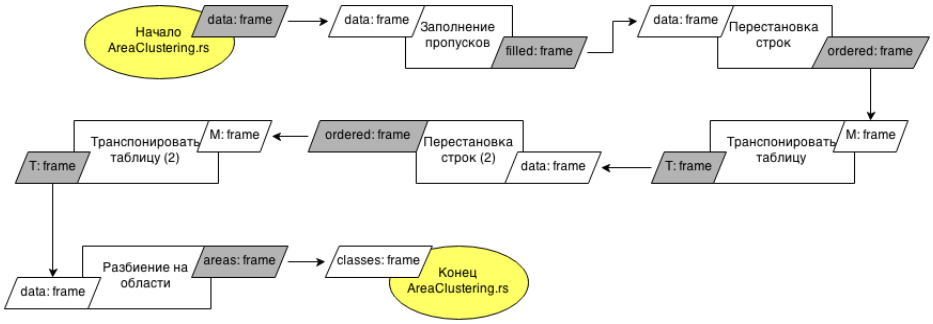


Рис. 59. Схема стратегии AreaClustering.rs.

1) На вход стратегии AreaClustering подается таблица data (типа frame), содержащая количества найденных орудий определенного типа в определенной локации;

2) Вход стратегии передает data методу «Заполнение пропусков»;

3) «Заполнение пропусков» вставляет нули в пропущенные поля таблицы;

4) «Заполнение пропусков» отдает цельную таблицу методу «Перестановка строк»;

5) «Перестановка строк» принимает таблицу (frame) data и действует следующим образом. Во-первых, по исходной таблице строится матрица расстояний между строками (расстояние между двумя строками вычисляется по стандартной евклидовой метрике). Далее исходная задача сводится к задаче коммивояжера, если положить строки – вершинами графа, а веса ребер – расстояниями между строками. В граф вводится «нулевая» вершина. После этого запускается алгоритм решения задачи коммивояжера, который дает субоптимальное решение path. Перестановка строк таблицы data согласно порядку строк в pathесть конечный результат работы методаordered.

6) «Перестановка строк» передает таблицу ordered методу «Транспонировать таблицу». Это сделано для того, чтобы вновь воспользоваться перестановкой элементов таблицы, но уже для столбцов а не для строк;

7) «Транспонировать таблицу» превращает столбцы матрицы в строки и инициирует исполнение второй части сортировки, выполняя метод «Перестановка строк (2)»;

8) «Перестановка строк (2)» отдает ordered методу «Транспонировать таблицу», которая возвращает таблицу в исходное состояние;

9) Все готово к выполнению метода «Разбиение на области». Разбиение на области получает упорядоченную таблицу (frame) areas и производит на ней кластеризацию (см. описание ниже).

10) Итог выполнения этого метода и всей стратегии – таблица classes, в которой каждой ячейке указан номер класса, к которому она принадлежит.

Основная сложность при разработке такого рода стратегий состоит в использовании нестандартных методов. В данном случае таковым является метод «Разбиение на области» (запрограммированый метод выложен в общий доступ по адресу /public/Data/AreaCluster.r). Для нестандартных методов необходима процедура программирования этих методов на R-языке с учетом спецификации системы. Все остальные части стратегии выбираются из общедоступного архива и разработка стратегии проводится в духе системы интеллектуального анализа данных на основе визуального редактора.

Метод «Разбиение на области» начинает свою работу с процедуры инициализации.

```
v <- list(); e <- list(); nu<- mean(input);
addEdge <- function (v1, v2) {
e[[v1]] <- c(e$v1, v2);
```

```
e[[v2]] <- c(e$v2, v1); }
idF <- function(i, j) { return(paste('v_', i, '.', j, sep='')); }
```

Изначально вся таблица представляется в виде графа, где вершины – это ячейки таблицы, а ребра соединяют две вершины тогда и только тогда, когда они являются в таблице соседними:

```
for (i in 1:nrow(input)) { for (j in 1:ncol(input)) { #ячейки
  v[[idF(i,j)]] <- data.frame(id=idF(i,j), val=input[i, j]);
if (j<ncol(input)) { # есть сосед справа
v1<-idF(i,j); v2<- idF(i,j+1); #имена вершин на ребре
e[[v1]] <- c(e[[v1]], v2); e[[v2]] <- c(e[[v2]], v1); }
if (i<nrow(input)) { # есть сосед снизу
v1<-idF(i,j); v2<- idF(i+1,j); #имена вершин на ребре
e[[v1]] <- c(e[[v1]], v2); e[[v2]] <- c(e[[v2]], v1); }
}}
```

Затем перебираются все доступные ребра и выбирается такое, для которого изменение суммарной объясненной дисперсии по итоговому разбиению – максимально:

```
q <- function(val) { return((mean(val) - nu)^2 * length(val));}

edgeDelta <- function (v1, v2) {
return(q(c(v1$val, v2$val)) - q(v1$val) - q(v2$val)); }

maxCrit <- function () {
  emax <- list(h='', m='', crit=-Inf);
  for (from in names(v)) { for (idx in 1:length(e[[from]])) {
    to <- e[[from]][idx]; #конец ребра
d<- edgeDelta(v[[from]], v[[to]]); #оценка критерия
    if (d> emax$crit) { emax <- list(h=from, m=to, crit=d); }
if (d == 0) return(emax);# дисперсия не может возрастать
}}

  return(emax);
}
```

Наконец, выполняется слияние вершин графа по оптимальному ребру:

```
v[[host]] <- rbind(v[[host]], v[[merged]]); # объединить области
e[[host]] <- setdiff(c(e[[host]],e[[merged]]), c(host,merged));
  v[[merged]] <- NULL; e[[merged]] <- NULL; #убрать вершину
if (length(v) <= 15) break;# повтор пока более 15 областей
```

Оставшаяся часть стратегии отвечает за отрисовку значений в итоговую матрицу:

```
for (vert in names(v)) {
  cells <- v[[vert]];
  for (i in 1:nrow(cells)) {
    cell <- cells[i,];
    input[cell$x, cell$y] <- vert;
}
}
```

С готовой схемой и набором запрограммированных методов достаточно просто реализовать стратегию. Визуальный редактор служит прекрасным инструментом для конструирования пользовательских стратегий:

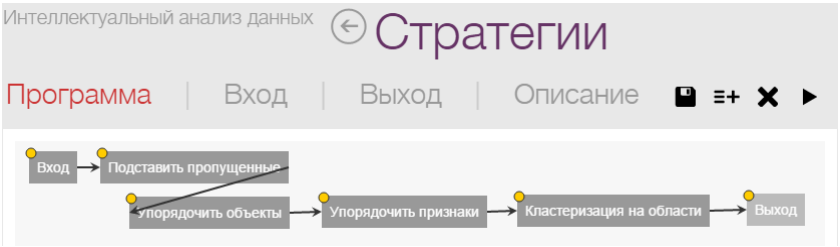


Рис 60. Реализация стратегии в визуальном редакторе стратегий

Результаты, полученные нами для приведенной стратегии представим в виде двух таблиц. Первая, таблица Ordered, получена на стадии перестановки строк и столбцов исходных данных, с целью минимизации суммарных расстояний между строками и столбцами таблицы. Вторая, Classified – результат процедуры классификации, каждая ячейка этой таблице соответствует ячейки таблицы Ordered. Внутри ячеек указан класс, к которому эта клетка принадлежит.

Результаты, полученные нами для приведенной стратегии представим в виде двух таблиц. Первая, таблица Ordered, получена на стадии перестановки строк и столбцов исходных данных, с целью минимизации суммарных расстояний между строками и столбцами таблицы. Вторая, Classified – результат процедуры классификации, каждая ячейка этой таблице соответствует ячейки таблицыOrdered. Внутри ячеек указан класс, к которому эта клетка принадлежит.

Таблица 9. Результат перестановки данных.

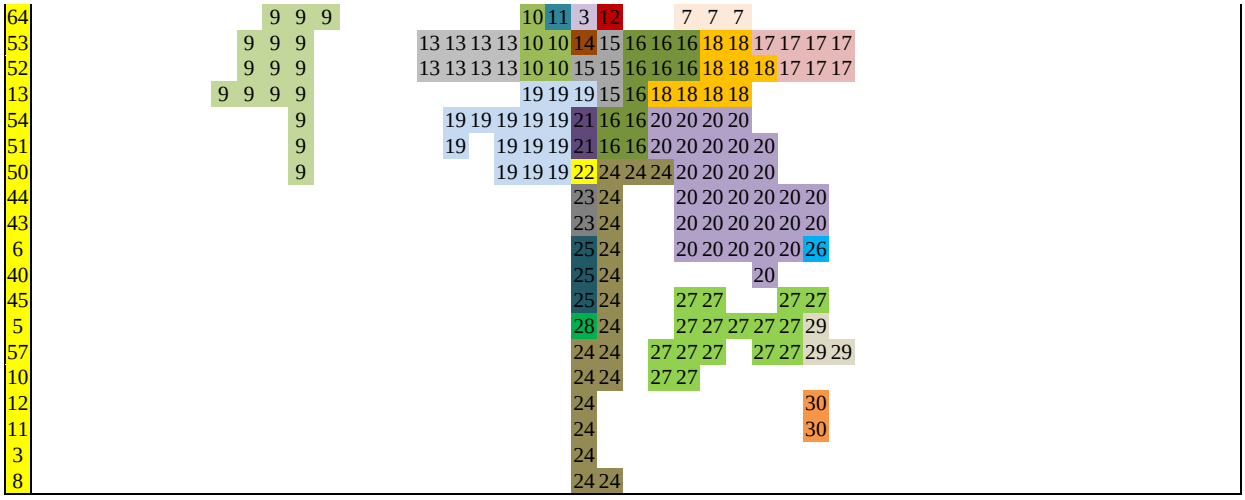
ID	33	26	15	13	21	30	16	27	19	11	17	40	36	41	29	32	14	37	18	25	24	1	7	28	20	5	8	9	10	3	2	22	4	12	34	6	35	45	44	46	42	47	43	39	38	23	31
46	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	5	0	89	30	0	5	8	8	0	7	17	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	8	0	0	8	2	0	2	4	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	50	14	0	5	5	7	1	5	7	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	1	0	0	1	6	2	0	0	0	0	0	0	1	0	12	7	52	7	0	20	0	1	0	1	1	6	7	0	2	0	0	2	0	0	0	0	0	0	2	1	0	0
9	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	1	5	10	4	38	13	0	0	9	0	1	2	0	0	5	18	2	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	22	7	0	0	3	4	2	1	1	9	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	4	0	0	2	0	1	3	0	0	0	0	0	0	0	13	21	2	0	0	7	0	3	3	1	0	1	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5	3	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
18	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	3	1	0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	2	0	1	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	6	1	3	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	11	0	4	3	0	0	0	0	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2	1	9	0	3	0	0	0	0	0	4	9	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	9	1	0	0	1	2	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	7	0	1	1	0	2	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	3	3	0	0	9	0	0	8	3	2	0	7	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	3	2	0	6	0	1	4	1	1	0	5	0	1	0	0	0	0	0	0	0	0	1	2	1	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	1	0	2	6	1	1	0	3	2	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	3	3	0	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	7	0	0	0	2	1	0	0	3	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	6	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	11	3	0	11	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	3	0	7	5	17	5	0	1	2	2	3	3	0	0	0	0	1	0	0	0	0	0	0	1	0	3	0	0	0	0	0
41	0	0	0	1	0	0	1	0	0	5	5	0	0	0	0	0	0	0	0	0	12	7	15	15	0	8	1	2	2	4	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	1	0	1	5	5	7	7	4	9	2	1	1	2	2	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1	1	3	2	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	6	0	1	0	0	0	0	0	0	0	4	2	0	0	0	8	5	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	10	0	2	15	5	0	6	8	2	3	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	2	0	10	0	0	0	0	0	0	12	6	4	9	0	2	1	0	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	21	58	29	0	0	0	0	21	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	2	0	0	0	0	0	0	1	9	0	1	0	0	0	0	0	0	0	16	18	18	0	6	0	0	0	0	0	0	1	0	4	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0
28	0	0	0	0	0	0	0	1	13	0	1	0	0	0	0	0	0	0	12	17	5	0	7	0	0	5	1	0	0	8	0	6	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
29	0	0	0	1	1	0	0	1	1	1	1	0	0	1	0	0	0	0	6	17	4	0	19	0	1	18	3	1	0	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
26	0	0	1	1	0	0	0	3	1	0	1	0	0	0	0	0	0	0	20	6	0	0	13	0	0	18	1	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	1	0	0	0	0	13	2	3	0	0	1	0	0	0	0	46	15	14	0	21	0	2	17	2	1	0	2	0	3	0	0	0	0	0	0	0	3	2	0	2	0	0	0	0	0
31	0	0	0	1	0	0	0	1	2	3	4	0	0	0	0	0	0	1	29	11	3	0	34	0	5	17	5	3	0	5	0	1	0	9	0	0	0	0	2	5	8	1	0	0	0	0	0	0
27	0	0	0	0	0	0	0	2	2	0	1	0	0	0	0	0	0	1	10	17	3	0	46	0	0	33	16	3	1	11	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	1	1	1	4	2	6	3	0	0	1	1	0	0	2	7	12	7	5	28	5	10	7	32	10	2	0	2	1	1	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	1	0	5	0	0	0	0	0	0	0	5	3	8	37	48	0	4	3	9	3	3	0	0	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	5	0	0	1	1	1	0	0	0	0	0	0	3	0	6	12	4	38	60	1	4	6	25	16	1	0	2	4	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	1	1	2	2	1	2	3	0	0	0	1	0	0	2	10	23	9	5	91	10	7	15	49	15	1	0	1	10	0	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0

15	0	0	0	0	0	5	2	0	1	1	3	0	0	0	1	0	3	1	8	20	11	21	119	1	1	17	63	16	4	0	2	6	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	2	2	0	0	2	0	4	21	24	0	0	8	7	0	2	0	0	0	15	0	76	0	1	0	33	33	20	0	0	0	0	15	16	6	5	3	3	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0	12	32	32	10	7	0	0	0	0	11	52	244	0	157	0	0	24	24	23	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
53	1	3	2	2	0	1	4	9	13	20	22	0	0	0	1	14	17	34	26	55	75	465	225	109	85	74	54	50	23	24	27	18	5	7	0	2	1	0	0	0	0	0	0	0	0	0	1
52	0	3	2	0	0	0	5	3	13	16	12	0	0	0	1	11	9	20	29	42	96	226	260	89	109	88	49	78	53	23	26	14	3	5	1	3	1	0	0	0	0	0	0	0	0	0	0
13	0	0	0	3	4	5	7	14	17	30	8	0	0	0	0	0	2	3	5	19	11	64	293	76	56	61	83	85	14	2	3	7	5	3	0	1	0	0	0	0	4	0	0	0	0	0	0
54	0	2	2	0	0	1	3	6	4	4	13	0	0	0	0	10	16	30	22	23	30	124	99	97	30	27	36	26	9	6	9	4	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
51	1	1	1	0	0	0	3	5	2	8	11	0	0	0	0	8	15	2	20	22	27	137	91	94	27	46	20	17	19	5	8	3	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
50	1	0	3	1	0	1	2	5	8	4	18	0	0	0	1	4	7	8	15	24	24	220	68	47	78	32	32	17	15	1	5	2	2	3	0	1	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0	315	41	0	0	42	32	0	19	29	30	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	2	3	1	0	0	0	0	0	0	0	1	5	0	354	40	0	0	47	28	1	13	45	38	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
6	0	0	2	1	0	2	2	6	8	13	12	0	0	0	0	0	0	0	8	10	2	226	78	0	2	19	24	28	18	13	85	9	15	6	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	3	2	0	0	0	0	0	18	0	0	5	0	0	0	3	2	11	14	202	32	0	15	1	3	5	14	5	1	2	0	0	1	3	0	0	0	0	0	0	0	2	0	0	0
45	0	0	0	0	0	0	0	0	1	0	4	0	0	0	0	0	1	0	0	0	0	183	29	0	0	31	20	2	5	25	24	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	2	0	0	3	2	3	5	11	7	0	0	0	0	0	0	0	6	8	3	135	46	0	2	9	11	17	14	8	56	8	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	2	0	0	4	9	0	16	3	0	0	7	0	2	9	12	56	54	0	14	16	37	7	14	36	71	55	12	8	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	1	4	0	0	0	4	6	0	0	0	0	0	0	3	16	19	28	61	40	0	10	19	3	6	0	1	16	9	21	4	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	1	0	0	0	0	6	0	0	0	0	0	1	2	19	10	19	64	2	0	2	1	2	0	0	2	50	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	2	0	10	0	0	0	0	0	0	2	11	14	11	66	8	0	7	2	2	0	0	6	39	2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	2	3	3	1	5	0	0	0	0	0	0	0	2	2	1	69	15	0	1	8	9	10	3	4	20	1	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0	3	5	0	0	0	0	0	0	1	1	15	6	70	36	0	2	2	5	9	4	0	7	2	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0

В таблице ниже представлены полученные в результате кластеризации области (30 штук). Каждая область окрашена в свой цвет, который соответствует цвету в таблице 8. Область 2 оставлена пустой для простоты восприятия (белое пространство).

Таблица 10. Результат разбиения матрицы на области

ID	33	26	15	13	21	30	16	27	19	11	17	40	36	41	29	32	14	37	18	25	24	1	7	28	20	5	8	9	10	3	2	22	4	12	34	6	35	45	44	46	42	47	43	39	38	23	31
46																						1																									
47																						1																									
48																						1																									
38																						1																									
9																																															
4																																															
2																																															
1																																															
18																																															
23																																															
24																																															
21																																															
22																																															
20																																															
19																																															
25																																															
33																																															
36																																															
35																																															
34																																															
37																																															
39																																															
42																																															
41																																															
49																																															
58																																															
55																																															
60																																															
62																																															
63																																															
56																				3																											
61														4	4					3																											
32																			3	3	3																										
28																			3	3																											
29																			3	3						5																					
26																			3							5																					
30																			3	3	3					5																					
31																			3	3	3		6			5																					
27																			3	3	3		6			5	5																				
17																			3	3	3	3	6				5																				
7																			3	3	3	6	6																								
14																			3	3	3	6	6				7	7																			
16																			3	3	3	3	8				7	7																			
15																			3	3	3	3	8				7	7																			
59										9	9										3	3	8				7	7	7																		



Интересен также анализ вкладов различных выделенных областей в общее распределение объектов. Для этого проанализируем таблицу долей объясненной дисперсии, которая приходится на ту или иную область. Таким образом, мы сможем показать, какие области объясняют полученное распределение хорошо, а какие – нет.

Область	Объясняет, %
15	18.05
23	16.18
14	15.76
25	8.84
16	6.23
11	4.26
22	3.45
24	3.28
2	2.63
21	2.36
18	2.30
8	1.83
12	1.73
10	1.31
28	1.26

Область	Объясняет, %
20	0.98
1	0.97
6	0.71
29	0.70
7	0.50
26	0.48
19	0.42
9	0.25
27	0.25
30	0.24
4	0.22
3	0.18
17	0.16
5	0.15
13	0.14

Таблица 11. Вклад областей в объяснение полученного разбиения

Следует отметить, что области выделялись таким образом, чтобы максимизировать суммарную объясненную дисперсию. Каждое объединение областей в алгоритме метода AreaCluster.r сопровождается потерей объясненной дисперсии, и эта потеря должна быть минимальной. Соответственно области – это наиболее экономные объединения объектов вида (Тип орудий, Локация) с точки зрения потерь объясненной дисперсии. Итоговая объясненная дисперсия составляет 95.8%, а её потери – 4.2%.

4.3. Анализ научных течений в новой археологии

Следуя за идеями, изложенными в работе [Холюшкин, Витяев, Костин, 2013], мы воспроизведем построение классификации различных археологических школ и течений. Суть идеи состоит в выделении трех достаточно ясно выраженных направлений на основе данных о взаимном цитировании авторов.

За основу была взята таблица 6 с количеством постраничных ссылок одного автора (указанного в строке таблицы) на другого (колонка таблицы). План разработки следующий:

1) Стратегия bib.rsполучает на вход таблицу (frame) перекрестных ссылок;
2) На полной таблице мы производим дискриминантный анализ, получая дополненную таблицу с возможностью выделения кластеров;
3) Строим точечный график, которые показал бы распределение объектов (авторов) в пространстве выделенных факторных нагрузок;
4) Анализируем график визуально, убеждаясь в качестве дискриминантного анализа;
5) Проводим повторный анализ с помощью нейронных сетей;
6) Сравниваем качество предсказанных нейронными сетями классов и теоретических;

Схема для Bib.rs достаточно очевидна:

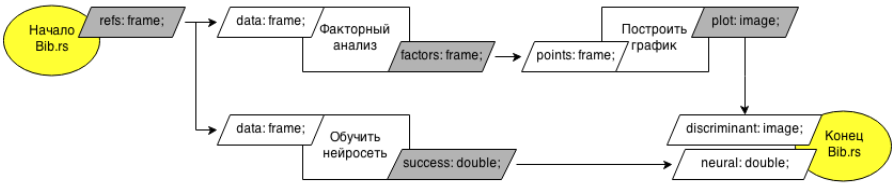


Рис. 60. Схема стратегии Bib.rs

Таблица 12. Исходные данные для стратегии Bib.rs

#	Автор	Класс	Бинфорд	Хилл	Фриц	Плог	Уотсон	ЛеБланк	Редмен	Уоллон	Стрювер	Лион	Лонгакр	Мартин	Айзек	Доран	Кларк_Д.	Фленнери	Ренфру	Дитц	Рауз
1	Бинфорд	1	82	38	12	6	5	1	2	10	3	5	2	6	9	0	34	3	24	0	2
2	Хилл	1	3	39	4	16	1	0	2	0	1	1	0	1	0	0	2	0	2	0	0
3	Фриц	1	0	12	5	0	3	0	1	0	0	0	0	1	0	1	1	1	0	0	0
4	Плог	1	0	4	10	1	3	0	1	0	0	0	0	1	0	1	5	1	0	0	0
5	Уотсон	1	3	2	2	0	21	2	4	0	5	0	2	0	0	0	4	5	0	0	3
6	ЛеБланк	1	0	2	2	0	8	0	3	0	4	0	1	0	0	0	4	4	3	0	2
7	Редмен	1	0	2	2	0	10	1	4	0	4	0	1	0	0	0	4	4	2	0	3
8	Уоллон	1	2	8	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0
9	Стрювер	1	2	1	1	0	0	0	1	0	2	0	0	0	3	0	1	0	1	0	0
10	Лион	1	0	0	6	2	1	0	1	0	0	0	0	0	0	0	0	1	2	0	0
11	Лонгакр	1	4	12	4	12	0	0	0	0	0	0	0	1	0	0	10	1	1	0	0
12	Мартин	1	1	1	5	15	3	0	0	0	0	0	0	6	0	0	4	0	2	0	0
13	Айзек	2	1	0	0	0	2	0	0	0	0	0	0	0	8	0	4	0	0	0	0
14	Доран	2	0	0	0	0	0	0	0	0	0	0	0	0	0	5	17	0	1	0	0
15	Кларк_Д.	2	8	7	1	5	1	0	0	0	1	0	2	0	3	3	138	5	6	0	1
16	Фленнери	3	6	2	2	1	1	0	2	1	0	3	0	0	0	0	3	17	27	0	4
17	Ренфру	3	0	0	0	0	0	0	1	0	0	0	0	0	2	0	8	0	8	0	0
18	Дитц	1	13	8	0	1	0	0	0	0	0	2	0	0	1	0	9	2	1	0	0
19	Рауз	1	12	57	1	2	0	0	0	0	0	0	3	0	0	1	5	1	0	0	0

Для дискриминантного анализа был применен уже готовый метод из системы public/Clustering/factor.r. Его результат (factors: frame) был передан следующему методу для отрисовки графика:

```
data<- input$data;
```

```
now<-format(Sys.time(), "%b%d%H%M%S");
plotFile<- paste("plot", "-", now, ".png", sep="");
png(filename=plotFile);
plot(data);
dev.off();
result<- list(plot=plotFile);
```

Метод plot.r рисует график по переданному ему набору точек и отдает на выход имя графического файла, содержащего этот график.

На этапе дискриминантного анализа была получена следующая картина:

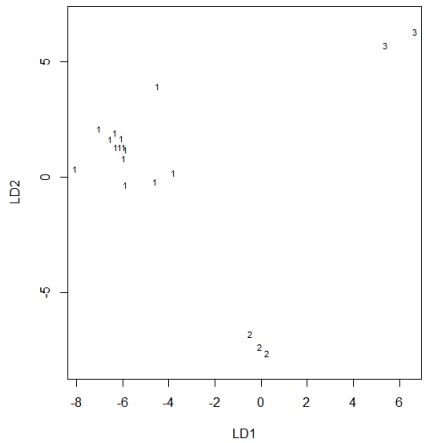


Рис. 61. Диаграмма распределения классов

Диаграмма распределения классов на основе дискриминтного анализа показывает полное соответствие предсказанных классов с классами теоретическими. В данном случае дискриминантный анализ оказывается в состоянии построить адекватную классификацию и является абсолютно состоятельным. Сравнимся теперь с нелинейными по своей природе моделями, построенными на основе нейронных сетей:

Таблица 13. Результат метода neural_nets.r

№	ФИО	Класс	Предсказан
1	Бинфорд	1	0.999997952874647
2	Хилл	1	1.00000796170678
3	Фриц	1	0.999999224949336
4	Плог	1	1.00000126993826
5	Уотсон	1	1.00000240437115
6	ЛеБланк	1	1.1428617571298
7	Редмен	1	0.898387496256107
8	Уоллон	1	0.999999990535727
9	Стрювер	1	1.00007883885016
10	Лион	1	0.999998937039829
11	Лонгакр	1	1.00000331525203
12	Мартин	1	1.0000153224157
13	Айзек	2	2.99997450346346
14	Доран	2	2.33332762107386
15	Кларк_Д.	2	2.90605050434245
16	Фленнери	3	0.999999717794561
17	Ренфру	3	2.99719792711429
18	Дитц	1	1.01162032166752
19	Рауз	1	1.00000031978014

Как видно из таблицы, нейронные сети достаточно плохо справляются с задачей предсказания на таком маленьком числе объектов. Путем оптимизации параметров архитектуры нейронной сети максимальное чего удается достичь – это точность порядка 84%.

Сам метод выглядит следующим образом:

```
  library("neuralnet");
  options(stringsAsFactors=FALSE);
  tr.in.init<-  input;
  tr.out.init<- input[target];
  tr.n<- setdiff(colnames(tr.in), c(target));
  f <- as.formula(paste(target, '~', paste(tr.n, collapse =
'+')));
  result<-
data.frame(id=character(0),exp=integer(0),n=integer(0));

  for (control in 1:nrow(input)) {
    tr.in <-  tr.in.init[-control,];
  tr.out<- tr.out.init[-control,];
  neuro<- neuralnet(f, data=tr.in, hidden=3, threshold=0.0001)
  testdata<- tr.in.init[control,-c('Class'];
  tr.predict<- compute(neuro, testdata) #Run
    net.res <- list(
       id=input[control,'Name'],
       exp=input[control,'Class'],
       n=tr.predict$net.result);
  result<- rbind(result, net.res);
  }
```

Алгоритм достаточно прост: из выборки выкидывается один объект, а для остальных строится модель на основе нейронной сети. Затем, для исключенного объекта выполняется предсказание на основе полученной модели. Иначе говоря, выполняется скользящий контроль.

Результаты всех шагов сводятся в единую таблицу, которая дана выше как Таблица 13.

ЗАКЛЮЧЕНИЕ

В ходе работ по проекту была создана web-система анализа данных в археологии. Проведенный анализ задач археологии и методов их решения показывает, что само по себе применение некоторого метода интеллектуального анализа данных не является решением какой-либо задачи археологии. Задача археологии означает, что мы хотим что-то новое УЗНАТЬ на основании имеющихся данных и имеющихся знаний. Сама постановка задачи определяет контекст задачи – ту точку зрения и систему понятий, в рамках которой надо решать задачу, интерпретировать имеющиеся данные и использовать знания, находящиеся в рамках данной системы понятий. Выбор методов интеллектуального анализа данных и интерпретация результатов их применения также определяется данным контекстом. Поэтому решением некоторой задачи археологии является такая последовательность применения методов, которая в рамках данного контекста и интерпретации данных, последовательно, каждым применяемым методом, дает некоторое новое знание, приводящее, в конце концов, к требуемому знанию. Вне рамок контекста решаемой задачи применение методов не имеет смысла. Отсюда возникает понятие стратегии решения задач археологии, используемое в монографии. На ряде приведенных примеров показано, как и какими последовательностями методов решаются задачи археологии. В монографии рассмотрен весь класс проблем, связанных с решением задач археологии методами интеллектуального анализа данных:

1) какие есть данные, их специфика, способы их записи и хранения;
2) какие есть методы интеллектуального анализа данных, применяемые в археологии и их классификация;
3) какие есть задачи археологии, которые возникают в разных контекстах и методологических подходах;
4) как решаются различные задачи археологии и, какими последовательностями методов (стратегиями);
5) примеры решения задач с помощью стратегий;
6) Программная web-система для решения задач археологии с помощью стратегий.

Мы надеемся, что применение данного подхода и программной системы откроет новую эру в использовании методов интеллектуального анализа данных для решения задач археологии.

ЛИТЕРАТУРА

Айвазян С. А., Бухштабер В. М., Юнюков И. С., Мешалкин Л. Д. Прикладная статистика: Классификация и снижение размерности. — М.: Финансы и статистика, 1989.

Брандт З. Анализ данных. Статистические и вычислительные методы для научных работников. - М.: Изд. «Мир», 2003: 688 с.

Витяев Е.Е. Классификация как выделение групп объектов, удовлетворяющих разным множествам согласованных закономерностей // Анализ разнотипных данных (Вычислительные системы вып. 99), Новосибирск, 1983: 44-50.

Витяев Е.Е. Естественная классификация как закон природы // Интеллектуальные системы и методология. ("Материалы научно-практического симпозиума "Интеллектуальная поддержка деятельности в сложных предметных областях", Новосибирск – 7-9 апреля 1992) // Новосибирск – 1992. Вып. 4.

Витяев Е.Е. Естественная классификация и систематика как законы природы // Анализ структурных закономерностей (Вычислительные системы Вып. 174), Новосибирск – 2005.

Витяев Е.Е. Извлечение знаний из данных. Компьютерное познание. Модели когнитивных процессов. Новосибирск, 2006: 293 с.

Витяев Е.Е. Извлечение информации из данных // Информационные технологии в гуманитарных исследованиях, Вып. 15, ИАЭТ СО РАН, Новосибирск, 2010: с. 9-16.

Витяев Е.Е., Костин В.С. Естественная классификация, систематика, онтология // Информационные технологии в гуманитарных исследованиях, Вып. 13, ИАЭТ СО РАН, Новосибирск, 2009: с. 65-75

Витяев Е.Е., Москвитин А.А. Введение в теорию открытий. Программная система DISCOVERY. // Логические методы в информатике (Вычислительные системы, вып. 148), Новосибирск, 1993, с.117-163

Гарден Ж.-К. Теоретическая археология. – М.: Прогресс, 1983: 296 с.

Гражданников Е.Д. Метод построения системной классификации наук. - Новосибирск, 1987.

Гражданников Е.Д. Проблема критериальной оценки научных результатов // Проблемы развития научно-образовательного потенциала. – Новосибирск: Наука, 1987: с. 24-47.

Гражданников Е.Д., Фелингер А.Ф., Холюшкин Ю.П. Системная классификация разделов археологии. // Методические проблемы реконструкций в археологии и палеоэкологии. – Новосибирск: Наука, 1989: с. 5 – 16.

Гражданников Е.Д., Холюшкин Ю.П. Системная классификация социологических и археологических понятий. Новосибирск: Наука. 1990.

Груздев Д.В., Журбин И.В. Компьютерное моделирование археологических объектов: методика и технология создания пространственной модели // Информационный бюллетень Ассоциации "История и компьютер", N 29, июнь 2002. http://kleio.asu.ru/aik/bullet/29/17.html 93

Грязнов М.П. Классификация, тип, культура // Теоретические основы советской археологии. – Л.,1969: с. 18-22.

Демин А.В., Витяев Е.Е. Метод построения «естественной» классификации // Информационные технологии в гуманитарных исследованиях, Вып. 15, ИАЭТ СО РАН, Новосибирск, 2010: с. 16-22

Деревянко А.П., Фелингер А.Ф., Холюшкин Ю.П. Методы информатики в археологии каменного века. - Новосибирск, 1989.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т. Предварительные результаты информационно-статистического анализа мустьерских индустрий Алтая. // Методология и методика археологических реконструкций.- Новосибирск, 1994.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. Некоторые статистические подходы к оценке фациальности позднего палеолита Енисея. // Методология и методика археологических реконструкций.- Новосибирск, 1994.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. и др. Математические методы в археологических реконструкциях. - Новосибирск, 1995а.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С. Неандертальская проблема как задача статистического анализа. // III Итоговая сессия Института археологии и этнографии СО РАН. Тезисы докладов. - Новосибирск, 1995б: 47-49.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. Некоторые статистические подходы к оценке фациальности мустьерских памятников Алтая. // Гуманитарные науки в Сибири, №3, - Новосибирск, 1996: 3-10.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С. Неандертальская проблема как задача статистического анализа (предварительные результаты).// Информационные технологии в гуманитарных исследованиях. - Новосибирск, 1998а:

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. Структурный анализ мустьерских памятников Алтая. // Каменный век Казахстана и сопредельных территорий. - Туркестан, 1998б: 93-111.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Статистический анализ позднепалеолитических комплексов Северной Азии. - Новосибирск, 1998в.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Статистический анализ среднепалеолитических индустрий Ближнего и Среднего Востока. - Новосибирск, 1999.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Неандертальская проблема как задача статистического анализа. – Новосибирск, 2001.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Статистический анализ мустьерских индустрий Кавказа. Часть 1. Технологические индексы. – Новосибирск, 2002а.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Статистический анализ среднепалеолитических индустрий Кавказа. Типология . – Новосибирск, 2002б.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Корреляция среднепалеолитических индустрий Ближнего Востока и Кавказа. – Новосибирск.: Изд. СО РАН, 2002в.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Статистический анализ индексов мустьерских памятников Средней Азии // Проблемы каменного века Средней и Центральной Азии. – Новосибирск, Изд. ИАЭТ СО РАН, 2002г.: 92-101.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. Статистический анализ технологических индексов мустьерских индустрий Кавказа // Информационные технологии в гуманитарных исследованиях. Вып.3. - Новосибирск: НГУ, 2002, с. 43-65.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С. Статистическое изучение мустьерских индустрий Кавказа и Ближнего Востока. Проблемы сопоставимости //Информационные технологии в гуманитарных исследованиях. Вып.5. - Новосибирск: РИЦ НГУ, 2003, с. 27-49.

Деревянко А.П., Холюшкин Ю.П., Воронин В.Т., Ростовцев П.С., Костин В.С., Корнюхин Ю.Г. Статистические методы в изучении каменных индустрий //Информационные технологии в гуманитарных исследованиях. Выпуск 6. - Новосибирск: Редакционно-издательский Центр НГУ, 2003, с. 30-46.

Деревянко А.П., Холюшкин Ю.П., Костин В.С., Воронин В.Т. Структурный анализ орудийных комплексов Ближнего и Среднего Востока и Кавказа //Информационные технологии в гуманитарных исследованиях. Выпуск 7. - Новосибирск: Редакционно-Издательский Центр НГУ, 2004, с. 78-90.

Деревянко А.П., Холюшкин Ю.П., Ростовцев П.С., Воронин В.Т. Пример исследования устойчивости кластеризации на материалах мустье Алтая // Информационные технологии в гуманитарных исследованиях. Выпуск 7. – Новосибирск: Редакционно-издательский Центр НГУ, 2004, с. 91-93.

Долуханов П.М. Верхний палеолит и мезолит Европы: опыт многомерного анализа // Проблемы реконструкций в археологии. – Новосибирск: Наука, сиб. Отд., 1985: 62-73.

Дюк В.А. Обработка данных на ПК в примерах. — СПб: Питер, 1997.**Клейн Л.С.** История археологической мысли. – СПб, 2005.

Дюк В.А. Data Mining – интеллектуальный анализ данных. Санкт-Петербург, 2002

Жамбю М. Иерархический кластерный анализ и соответствия. - М.: Финансы и статистика, 1988.

Жданов А.С., Костин В.С. Значимость и устойчивость автоматической классификации в задаче поиска оптимального разбиения // Информационные технологии в гуманитарных исследованиях. - 2002. - Вып. 3. - С. 36-42.

Каменецкий И.С., Маршак Б.И., Шер Я.А. Анализ археологических источников. – М., 1975.

Классификация в археологии. Терминологический словарь-справочник. – М., 1990: 156 с.

Клейн Л.С. Теории в археологии // Новое в археологии Сибири и Дальнего Востока. - Новосибирск, 1979.

Клейн Л.С. О предмете археологии (в связи с выходом книги В.Ф. Генинга “Объект и предмет науки в археологии” // СА, М.,1989, 3 : 209 – 219.

Клейн Л.С. Археологические источники. Л.: ЛГУ. 1978. 119 с.

Клейн Л.С. Археологическая типология – Л., 1991а : 446 с.

Клейн Л.С. Рассечь кентавра. О соотношении археологии с историей в советской традиции // Вопросы истории естествознания и техники. № 4, 1991б: 3–12.

Клейн Л.С. Археологические источники. – Л., 1995 (2 изд.).

Клейн Л. С. Введение в теоретическую археологию. Санкт-Петербург, Бельведер. 2004.

Клейн Л.С. История археологической мысли. – СПб, 2005.

Клейн Л.С. Новая археология. – Донецк, 2009.

Колпаков Е.М. Проблема специфичности понятия «археологические источники // Категории исторических наук. Л., 1988.

Колпаков Е.М. Теория археологической классификации. – СпБ, 1991: 112 с.

Костин В.С. Статистика для сравнения классификаций // Информационные технологии в гуманитарных исследованиях. Вып. 6. – Новосибирск, 2003: с. 57-65.

Костин В.С., Корнюхин Ю.Г. Построение обобщенной классификации // Информационные технологии в гуманитарных исследованиях. - 2003. - Вып. 6. - С. 65-72.

Окунь Я. Факторный анализ. М. 1974

Регирер Е. М. О профессии исследователя в точных науках. Москва: Наука, 1966.

Ростовцев П.С. Алгоритмы анализа структуры прямоугольных матриц «пятна» и «полосы» // Статистика. М. 1985.

Ростовцев П.С. Статистическое согласование мер связи в анализе социально-экономической информации. // Экономика и математические методы, 1991, т. 27, вып. 1: 150-156.

Ростовцев П.С., Костин В.С. Автоматизация типологического группирования. Препринт №137. - Новосибирск, 1995

Федоров-Давыдов Г.А. Статистические методы в археологии. М.: Высшая школа.1987. 216 с.

Холюшкин Ю.П. О возможности проверки эффективности археологических гипотез // Археология эпохи камня и металла Сибири – Новосибирск, 1983: с. 143-149.

Холюшкин Ю.П. Системная археология. Новосибирск: РИЦ НГУ, 2010: 554 с.

Холюшкин Ю.П., Ростовцев П.С. Проблема статистического обоснования критериев выделения мустьерских фаций Средней Азии //Гуманитарные исследования. Итоги последних лет. – Новосибирск, 1997, с. 11-12.

Холюшкин Ю.П, Васильев С.А., Воронин В.Т, Костин В.С., Нуртдинов А.Н. Динамика развития позднепалеолитической культуры на верхнем Енисее: опыт статистического исследования. - Новосибирск, 2005. - 114 с.

Холюшкин Ю.П., Витяев Е.Е., Костин В.С. К вопросу о поисках закономерностей в археологии // Информационные технологии в гуманитарных исследованиях. №17 – Новосибирск: НГУ, 2012: С. 4-39.

Холюшкин Ю.П., Гражданников Е.Д. Системная классификация археологической науки (элементарное введение в археологическое науковедение). – Новосибирск: НГУ, 2000, 60с.

Холюшкин Ю.П., Костин В.С. Проверка гипотезы о существовании групп комплексов среднего палеолита Алтая. // Дальневосточные сибирские древности. Сборник научных трудов, посвященный70-летию со дня рождения В.Е.Медведева. – Новосибирск: изд. ИАЭТ СО РАН, 2012: С. 104-110.

Холюшкин Ю.П., Холюшкина В.А. Методические проблемы исследования археологических культур каменного века Сибири // Проблемы реконструкций в археологии. - Новосибирск: Наука, 1985. - С.23-45.

Шер Я.А. Интуиция и логика в археологическом исследовании (к формализации типологического метода в археологии). // Статистико–комбинаторные методы в археологии. – М., 1970: 8–24.

Binford L.R., and Binford S.R. A Preliminary Analysis of Functional Variability in the Mousterian of Levallois Facies // American anthropologist - 1966, № 68:238-295.

Binford S.R. & Binford L.R., Archaeological theory and Method // Binford S.R.& Biford L.R. (eds). New perspectives in archaeology. –Chicago, 1968: 373 p.

Binford L. R. An archaeological perspective. - NY, London, 1972.

Binford L. R. Behavioral Archaeology and the Pompeii Premise // Journal of Anthropological Research. V. 37. 1981a: p. 195—208.

Binford L. R. Bones: Ancient Men and Modern Myth. – Orlando: Academic Press, 1981b.

Binford L. R. In pursuit of the past: Decording the archaeol. Rec. – London, 1984: 256 p.

Bordes F. Le Paleolithique inferieur et moen de Jabrud (Syrie) et la question du pre-Aurignacien. // L'Anthropologie, 59 (5-6), 1955: 486-507.

Bordes F. Principes d'une methode d'etude des techniques de debitage et de typology du Paleolithique ancien et moyen // L'anthropologie, 1950, v. 54:19-34.

Borillo M. Construction of a deductive model by simulation of a traditional archaeological study // American antiquity – 1974, V. 39, № 2: p. 243-252.

Brézillon M. La dénomination des objets de pierre taillée. // Matériaux pour un vocabulare de préhistoriens de langue francaise. - Paris, 1968.

O'Brien M.J. and Lyman LR Seriation, Stratigraphy, and Index Fossils: The Backbone of Archaeological Dating. New York: Kluwer Academic/Plenum Publishers. 1999.

Chang K. C. Rethinking archaeology. New York: Random House, 1967..

Chenhall R.G. The impact of computers on archaeological theory: an appraisal and projection // Computers and the Humanities, 1968, 3(1): 15-24.

Codd E. F., Codd S. B., Salley C. T. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate. - E. F. Codd & Associates, 1993.

Clarke D.L. Analytical archaeology. – L: Methuen, 1968: 684 p.

Clarke D.L. (ed). Models in archaeology. – L: Methuen, 1972: 1055 p.

Derevianko A.P., Kholiouchkine Y.P., Voronine V.T., Rostovtsev P.S. L'Analyse statistique des Ensembles de paléolithique moyen du Proche-et Moyen-Orient. –Novosibirsk, 2001.

Derevianko A.P., Kholuchkin Y.P., Rostovtsev P.S., Voronin V.T. Corrélation des industries paléolithique moyen du Proche-Orient du Caucase. – Novosibirsk, 2004.

Dibble, H.L. Middle paleolithic scraper reduction: Background, clarification, and review of the evidence to date. "Journal of Archaeological Method and Theory" № 2:, 1995:: 299-368.

Dibble, H.L. The interpretation of Middle Paleolithic scraper reduction patterns, in L'Homme de Neandertal Volume 4: La Technique". Edited by L. R. Binford and J.-P. Rigaud. Liège: Etudes et Recherches Archéologiques de l'Université de Liège, 1988..

Dibble H.L., Holdaway S.J. A Middle Paleolithic Industries of Warwasi. // The Paleolithic of the Zagros-Taurus. – Philadelphia, 1993: 73-99.

Deetz J. Invitation to Archaeology. – N.Y., 1967.
Doran J. Systems theory, computer simulations and archaeology // World archaeology, 1970, 1: p. 289–290.
Doran, J.E. & Hodson, F.R. Mathematics and Computer in archaeology.– Edinburg, 1975: 381 p.
Dunnell R.C. Systematics in prehistory. - N.Y., 1971.
Edgington E. Randomization Test. – N.-Y., 1995.
Efron B. & Diaconis P. Computer intensive methods in statistics. // Scientific American, 1983: 116-130.
Efron B. Better bootstrap condidence intervals // J. American Statist. Association, 1986, 81
Efron B. and Tibshirani R. An Introduction to the Bootstrap. N.-Y., 1993.
Eving J.F. Preliminary Note on the Exavational the Paleolithic Site of Kzar Akil, Republic of Lebanon. // Antiquity, 21, 1947: 186-196.
For theory building in archaeology: Essays on faunal remains, aquatic resources, spatial analysis, and Systematic modeling // Ed. by Binford L.R. – N.Y., 1977 – XVII: 419 p.
Good P. Permutation Test: A Practical Guide to Resampling Methods for Testing Hypotheses
Hill J.N, Evans R.K. A model for classification and typology. // Models in archaeology. –L., 1972: p. 231–273.
Hymes D. Lynguistic models in archaeology // Archeologie et calculateurs. Problemes semiologiques et mathematiques. –
Paris, 1970: p. 91-120.
Jelinek A.J. Technology, Typology, and Culture in the Middle Paleolithic. // Upper Pleistocene Preghistory of Western Eurasia, 1988: 199-214.
Jelinek A.J. Tabun Cave and Paleolithic Man in the Levant.// Scince, 282, 1982: 1369-1375.
Jelinek A.J. The Middle Paleolithic in the southern Levant.// Préhistoire du Levant. - Lyon, 1981.97
Jelinek A.J. A Preliminary report on some Lower and Middle Paleolithic industries from the Tabun Cave, Mount Carmel.
Jochim M.A. Hunter-gatherer subsistence and Settlement. A predictive model. –NY, 1976.
Kendall M.G. A Course in multivariate analysis. L, 1957.
Kintigh K. Measuring archaeological diversity by comparison with Simulated assemblages.// American Antiquety, vol. 49, 1984: 44-54.
Knowledge Discovery Through Data Mining: What Is Knowledge Discovery? — Tandem Computers Inc., 1996.
Binford L. R. An archaeological perspective. New York – London, Seminar Press.1972.
Kovalerchuk B., Vityaev E. Data Mining in Finance: Advances in Relational and Hybrid methods. (Kluwer international series in engineering and computer science; SECS 547), Kluwer Academic Publishers, 2000, p.308.
Leone M.P.(ed.) Contemporary archaeology. A guide to theory and contribution. – Carbondale&Edwardsville, 1972a.
Leone M.P. Issuesin anthropological archaeology. – Leone, 1972b.
Liiv I. Seriation and matrix reordering methods: An historical overview. Statistical Analysis and Data Mining 3(2) 2010.:70-91.
Marks A.E. Typological Variability in the Levantine Middle Paleolithic.// The Middle Paleolithic: Adaptation, Behavier and Variability. University Museum series, vol.2., 1992: 127-142.
Marks A.E. Early Mousterian Settlement Patterns in the Sentral Negev, Israel: Their Social and Economic Implications. // L' Homme de Neandertal. - Liege, 1989, vol. 6: 115-126.
Marks A.E. The Levantine Middle to Upper Paleolithic Transition: the past and present.// Studi di paletnologia in onore di Salvatore M. Puglisi. - Roma, 1985: 123-136.
Marks A.E. The Middle Paleolithic of The Negev. // Préhistoire du Levant. - Paris: CNRS, 1981: 287-298.
Marks A.E., Monigal K. The Production of Elongated Blanks from the Early Levantine Mousterian at Rosh Ein Mor: A Technological Perspective// The Definition and Interpretation of Levallois Technology. International Conference (11.05.93-15.05.93), The University of Pennsylvania and Harvard University., 1993.
Marks A.E., Monigal K. Modeling the Production of Elongated Blanks from Early Levantine Mousterian at Rosh Ein Mor // The definition and Interpretation of Levallois Technology.-Madison, 1995: 267-277.
Marks A.E., Volkman P. The Mousterian of Ksar Akil: levels XXVIA through XXIIIB.// Paleoorient, 1986, vol. 12/1: 5-20.
Meltzer D.J. Paradigms and the nature of change in American archaeology // American antiquity, 1979. - v.44, №4.
Models in archaeology (ed.Clarke)– L: Methuen, 1972: 1055 p.
Peterson I. Pick a sample// Science News, 140, 1991: 56-57.
Redman Ch. L. (ed.). Research and Theory in Current Archaeology. – N.Y., L., Sydney, Toronto, 1973: 390 p.
Renfrew A.C. (ed.). The explanation in culture change: models in prehistory– London: Duckworth, 1973: 788 p.
Ringrose T. Bootstrapping and correspondenting analysis in archaeology. // Journal of Archaeological Sciences, vol. 19, 1992: 615-629.
Roland N.I., Dibble H.L. A new synthesis of middle paleolithic variability // American Antiquity. - 1990, v. 55, № 3: 480-499.
Salmon M.H. Philosophy and archaeology. – N.Y., 1982 – XI: 203.
Schiffer M.B. Archaeological Method and Theory, Volume 1, reviewed by T.G. Baugh in Journal of Field

Archaeology. // Journal of Field Archaeology, 1991, V.18, №4.
Simon J. Resampling: The New Statistics, Resampling Stats. - Arlington, VA, 1993.
Simon J. What some puzzling problems teach about the theory of simulation and the use of resampling. // American Statistician, vol. 48, 1994: 290-293.
Taylor W.W. A study of archaeology // American Anthropologist. Vol. 50. N 3, Pt 2, Memoir 69, 1948
The archaeology of contextual meanings / Ed by Hodder – Cambridge, 1987 – VII: 144 . (New direction in archaeology).
Trigger B.C. Settlement archaeology – its goals and promise // American antiquity – 1967, V. 32, № 2.
Trigger B. G. 1978. No Longer from another Planet. - Antiquity, LII, 1978: 193-198.
Trigger B. G. Marxism and archaeology. - Macquet J. and Daniels N. (eds.). On Marxian perspectives in anthropology: Essays in honor of Harry Hoijer 1981. Malibu, Undena Publications. 1984b.: 59 - 97.
Zweig Zachi. Using Data Mining Techniques for Analyzing Pottery Databases. Bar-Ilan University. 2007
Vityaev E.E, Kovalerchuk B.Y. Relational Methodology for Data Mining and Knowledge Discovery. Intelligent Data Analysis. Special issue on "Philosophies and Methodologies for Knowledge Discovery and Intelligent Data Analysis" eds. Keith Rennolls, Evgenii Vityaev. v.12(2), IOS Press, 2008, pp. 189-210
Watson P.J., LeBlanc S., Redman Ch.L. Explanation in archaeology. An explicitly scientific approach. –N.Y, London: Columbia Univ. Press, 1984.
Whilley G.R, Phillips Ph. Methods and theory in American archaeology. – Chicago: University of Chicago Press, 1958.

ПРИЛОЖЕНИЯ

Бартаханова Н.Д.,
Неупокоев Н.В.

Естественная классификация объектов через неподвижные точки предсказаний

Аннотация. *В статье описывается метод для «естественной» классификации в качестве стратегии анализа данных с пропусками. Проведена классификация археологических данных: орудийных комплексов Ближнего и Среднего Востока и Кавказа.*

Ключевые слова: *«естественная» классификация, классификация, предсказание, интеллектуальный анализ данных, неподвижная точка.*

В настоящее время известно много принципов и алгоритмов построения классификаций, при этом мало алгоритмов, основанных на принципах «естественной» классификации. Принцип «естественной» классификации сформулирован следующим образом [Витяев, 1983].

Разбиение объектов на классы должно производиться в соответствии с теми закономерностями, которым удовлетворяют объекты, т.е. объекты одного класса подчиняются одним и тем же закономерностям, объекты разных классов подчиняются разным группам закономерностей. Кроме того, объекты одного класса имеют группу закономерностей, которая предсказывает различные свойства объектов этого класса, что подтверждает взаимную согласованность объектов и целостность класса.

Наиболее точной формализацией принципа «естественной» классификации являются неподвижные точки предсказаний. Они выделяют множества объектов, удовлетворяющие разным множествам закономерностей, которые участвуют в предсказаниях неподвижной точки и характеризуют некоторую целостность, так как в неподвижной точке закономерности согласованы по предсказанию. В некотором смысле неподвижные точки определяют семантику множества закономерностей, т.к. они определяют классы объектов, на которых определённые множества закономерностей истинны.

Далее мы покажем, каков алгоритм нахождения неподвижных точек предсказаний по вероятностным закономерностям, чем определим классификацию, и проверим метод обнаружения неподвижных точек на реальных данных.

Неподвижные точки

Для нахождения неподвижных точек нужно извлечь из данных закономерности, описывающие свойства этих объектов. Для этого воспользуемся семантическим вероятностным выводом [Витяев, 2006], который из некоторого множества признаков X извлекает вероятностные закономерности $LP(X)$. Используя $LP(X)$, можно определить оператор предсказания Pr:

$$\Pr(\mathrm{X}) = \Phi_{\mathrm{Krit}}(\mathrm{X}),$$

где $\Phi_{\mathrm{Krit}}(\mathrm{X})$ – модификация множества X, увеличивающая специальный критерий $Krit$ максимальной согласованности предсказаний. Применяя Pr к некоторому множеству из X, оператор будет осуществлять предсказания по всем закономерностям $LP(X)$. Если обозначить n-кратное применение оператора Pr через $\Pr^{\mathrm{n}}$, тогда неподвижная точка оператора Pr, будет определяться равенством $\Pr^{\mathrm{n}+1}(\mathrm{X}) = \Pr^{\mathrm{n}}(\mathrm{X})$.

Стоит отметить, что семантический вероятностный вывод формализует образование условных связей и может рассматриваться как формальная модель нейрона [Витяев, 2007; Витяев, Перловский, Ковалерчук, Сперанский, 2011]. Работа оператора предсказаний Pr, который использует закономерности, найденные семантическим вероятностным выводом, имеет схожесть с тем, как происходит восприятие образа [Витяев, Неупокоев, 2014].

Критерий взаимной согласованности предсказаний

Для получения полной классификации необходимо найти неподвижные точки для всех объектов исследуемых данных. За первоначальный набор признаков X берутся признаки классифицируемого объекта. Определим критерий $Krit$ согласованности предсказаний. Пусть $S \subset LP(X)$ – множество закономерностей, подтверждающихся на наборе X, а $F \subset LP(X)$ - множество закономерностей, опровергающихся на наборе X. Тогда критерий $Krit$ есть сума весов подтверждающихся закономерностей минус сумма весов опровергающихся закономерностей:

$$Krit(X) = \sum_{R \in S} \mu(R) - \sum_{R \in F} \mu(R), \text{где } \mu(R) = -\log\big(1 - p(R)\big).$$

Функция $\mu(R) = -\log\big(1 - p(R)\big)$ усиливает вклад закономерностей, вероятность которых близка к единице.

Модификацию Φ_{Krit} исходного набора признаков X, увеличивающую критерий согласованности $Krit$, будем осуществлять пошагово. Будем либо расширять множество X на один элемент, которого еще нет в наборе, либо удалять какой-то элемент из множества X. При этом в обоих случаях критерий согласованности предсказаний $Krit$ должен строго увеличиваться на максимально возможную величину. Итак, определим модификацию Φ_{Krit} как

$$\Phi_{Krit}(X) = \begin{cases} X \cup x, \text{если } \delta^+ > \delta^-, \delta^+ > 0, x = arg\max\limits_x\big(Krit(X \cup x)\big), \\ X \backslash x, \text{если } \delta^- \geq \delta^+, \delta^- > 0, x = arg\max\limits_x\big(Krit(X \backslash x)\big), \\ X, \text{иначе}, \end{cases}$$

$$\text{где } \delta^+ = \max_x(Krit(X \cup x) - Krit(X)),\ \delta^- = \max_x(Krit(X \backslash x) - Krit(X))$$

Модификация дает *неподвижную точку*, когда мы имеем третий случай. Тогда у нас будет выполнено равенство $Pr^{n+1}(X) = Pr^n(X)$.

Каждая неподвижная точка определяет некоторый *класс*. К этому классу относятся те, и только те объекты o^i, которые *n*-кратным применением оператора $Pr^n X\big(o^i\big)$ приводятся к данной неподвижной точке.

Алгоритм получения закономерностей

Алгоритм поиска закономерностей подробно изложен в [Витяев, Неупокоев, 2014]. Стоит только добавить уточнение касательно обработки пропусков в данных. Выполнимость предиката задается в известном смысле, т.е.

$$P_i\big(o_i^j\big) = \begin{cases} \text{истина}, & \text{если } o_i^j = (o^j)_i \\ \text{ложь}, & \text{если } o_i^j \neq (o^j)_i \end{cases},$$

где $(o^j)_i$ - элемент кортежа на i -ом месте. Но если в объекте o^j на i -ой позиции нет конкретного значения, т.е. на данной позиции пропуск, то фактически мы рассматриваем правило R без предиката $P_i(x)$. Обозначим пропуск как ?, тогда $P_i(?) = \text{ложь}$, аналогично и для предиката в заключении правила будет выполняться $P_0(?) = \text{ложь}$.

Оператор проверки LP необходим для того, чтобы избежать добавления закономерностей, условная вероятность которых возрастает не статистически значимо. Теоретические выкладки по оператору проверки даны ранее в [Витяев, 2006, §46].

Оператор проверки использует критерий Фишера, который основан на рассмотрении предельных случаев расположения данных (какие только возможны) и вычислении вероятности для каждого из них [Кендал М., Стюарт А., 1973].

Для правила $R = P_i \Rightarrow P_0$ введем 2 гипотезы: о независимости предикатов P_0 и P_i

$$H_0: p(P_0 \,\&\, P_i) = p(P_0)p\,(P_i),$$

против альтернативы

$$H_1: p(P_0 \,\&\, P_i) \neq p(P_0)p\,(P_i).$$

И составим таблицу сопряженности 2x2:

	P_0	$\neg P_0$	Сумма
P_i	m_{11}	m_{10}	$n_{1\_}$
$\neg P_i$	m_{01}	m_{00}	$n_{2\_}$
Сумма	$n_{\_1}$	$n_{\_2}$	n_*

При малых значениях n_*, когда наблюдаемые частоты в какой-либо ячейке меньше 5 нужно использовать критерий Фишера [Аптон Г., 1982], иначе следует применять дополнительно критерий Юла.

Взаимодействие критерия Юла и критерия Фишера следующее:

1. Если $m_{11} < n_{1\_} \cdot n_{\_1}$, что говорит об отрицательной корреляции и что проверяется в процессе вычисления критерия Фишера, то ответ относительно закономерности отрицательный;
2. Если ответ положительный, но критерий Фишера дает отрицательный ответ, то ответ относительно закономерности также отрицательный;
3. Если первые два ответа положительны, то, если частоты больше 5 (тогда критерий будет приближаться к распределению χ^2), дополнительно проверяется условие $Q_\beta > 0$. В отличие от критерия Фишера критерий Юла не является точным критерием, который бы выдавал вероятность. В нашем случае нужно выбирать значение наиболее близкое к 1. Пользователем задается нижняя граница $Q > 0$. Для большого числа n_* критерий Фишера, как правило, выполнен, тогда корреляцию нужно оценивать по критерию Юла, по которому закономерность будет приниматься, если выполнено $Q_\beta > Q$.

Правила, прошедшие проверку, будут являться вероятностными закономерностями. Если не все подправила имеют положительную корреляцию, значит в посылке присутствуют несущественные предикаты. Дополнительно проверяем соблюдение ограничений, задаваемых программой, для условной вероятности CP_{min} и критерия Фишера и Юла: $F(R,O) < \alpha, Q_\beta(R,O) > Q$ и $p(R) > CP_{min}$, где $F(R,O)$ и $Q_\beta(R,O)$ - значения статистических критериев для правила R на множестве объектов O.

Алгоритм классификации

Пусть мы получили множество закономерностей Reg при помощи алгоритма поиска закономерностей. Следующим этапом построения естественной классификации является нахождение идеализированных классов для всех объектов исходных данных.

Как отмечалось ранее объект – это набор признаков, на котором можно считать $Krit(X)$. Если исходный объект $o^i =< o_1^i, \dots, o_n^i >, o^i \in O$ описывается набором значений признаков $X_{1i}, \dots, X_{ni}$, то исходный набор равен $< Y_1, \dots, Y_n >$, т.е. составляется из $Y_k = \{X_{ki}\}, k = 1, \dots, n$.

Число значений конечно, обозначим это число A, оно задает мощность алфавита, с которым мы работаем: $|Y| = A, Y = \{a_1, \dots, a_A\}$.

Берем начальный набор значений признаков $Y =< Y_1, \dots, Y_n >$. Для каждого значения признака (цикл по j_0 от 1 до A, цикл по i_0 от 1 до n) $X_{i_0 j_0} \in \Omega \times Y$ определим оценку $V(X_{i_0 j_0})$, которая будет складываться следующим образом. Возьмем множество Reg и зафиксируем одну закономерность $R \in Reg$. Если R не применима к набору Y, то ее не рассматриваем; иначе она применима к набору [Витяев, 1983]. Пусть R имеет вид $P_{i_1} \& \dots \& P_{i_k} \Rightarrow P_{i_0}$, если $P_{i_0} = \left(x_{i_0} \approx o_{i_0}^{j_0}\right)$, то:

1. $V\left(X_{i_0 j_0}\right) = \mu(R)$, если закономерность R подтверждается на нем, т.е. $X_{i_0 j_0} \in Y_{j_0}$;
2. $V\left(X_{i_0 j_0}\right) = -\mu(R)$, если закономерность R опровергается на нем, т.е. $X_{i_0 j_0} \notin Y_{j_0}$;

Отсюда для вычисления $Krit$ получаем

$$Krit < Y_1, \dots, Y_n > = \sum_{x \in \Omega \times Y} V(x)$$

Для нахождения локального максимума нам не обязательно для каждого нового Y' считать критерий. Посчитав критерий $Krit < Y_1, \dots, Y_n >$ для начального набора, дальше можно считать только его модификацию Φ_{Krit} при добавлении к набору $Y =< Y_1, \dots, Y_n >$ значения, которое не принадлежит ему, или при удалении значения ему принадлежащего. Поэтому нужно вычислять некоторую другую оценку

$$\sum_{x \in \Omega \times Y} W(x) = Krit(Y') - Krit(Y)$$

Оптимизированный критерий

Рассмотрим возможные случаи, возникающие в процессе локализации максимума критерия. Изменить набор можно добавлением нового для набора признака или удалением некоторого признака из набора X. В закономерностях выбранный признак встречается в посылке, либо в заключении, также закономерность изменит свойство быть применимым в первом случае (изменение посылки), свойство подтверждаться или опровергаться - во втором случае соответственно.

Если алфавит состоит всего из двух символов ($A = 2$), то мы используем закон исключенного третьего. $Y = \{a_1, \dots, a_2\}, (x_i \approx a_1) = \neg(x_i \approx a_2)$. При $A > 2$ если истинно $(x_i \approx a_1)$ то, следовательно, истинны и $\neg(x_i \approx a_2)$, и $\neg(x_i \approx a_3)$,…, и $\neg(x_i \approx a_A)$. Значит, кроме рассмотрений x и $\neg x$, нужно следить за $\acute{x}$ и $\neg\acute{x}$, где $X \ni \acute{x} \neq x$.

Кратко новый признак можно записать как ${}^{\varepsilon_1}x^{\varepsilon_2}, \varepsilon_i \in \{0,1\}$, ${}^0x = x$, ${}^1x = \neg x, x^0 = x, x^1 = \acute{x}$..

Изменяя набор, мы изменяем множество закономерностей. Определим множество применимых закономерностей Reg_{aplic} и множество сильнейших вероятностных закономерностей (СВЗ, Strong Probability Regularity, SPR) Reg_{spr}:

$$Reg_{aplic} = \{R \in Reg: base(R) \subseteq Y\},$$
$$Reg_{spr} = \{R \in Reg_{aplic}: \forall R' \neq R, R' \in Reg_{aplic}, \quad R \not\subset R'\}.$$

Рассмотрим закономерности из Reg_{aplic}. Это множество разбивается на подтверждающиеся и опровергающиеся закономерности:

$$S = \{R \in Reg_{aplic}: target(R) \in Y\},$$
$$F = \{R \in Reg_{aplic}: target(R) \notin Y\},$$
$$Reg_{aplic} = S \cup F.$$

Рассмотрим случай, когда добавляется новое значение признака $x \in X, (x \notin Y) = (x \notin \{Y_1, \dots, Y_m\})$, где $Y =< Y_1, \dots, Y_m >$, и покажем, как изменятся множества S, R, Reg_{aplic} при переходе к набору $Y' = x \cup Y =< x, Y_1, \dots, Y_m >$.

Пусть добавленный признак лежит в заключении закономерностей $x \in target(R), \forall R \in Reg_{aplic}$. Это закономерности, применимые к набору, предсказывающие отсутствие или присутствие x и влиявшие отрицательно или положительно, соответственно, на критерий. После добавления x их знаки изменятся на противоположные:

$$F_x = \{R: target(R) = x \notin Y\}; \quad F_x \subseteq F, F_x \not\subseteq F', F_x \subseteq S'; \quad (3)$$
$$S_x = \{R: target(R) \neq x \notin Y\}; \quad S_x \subseteq S, S_x \not\subseteq S', S_x \subseteq F'; \quad (4)$$
$$S' = S + F_x - S_x$$
$$F' = F - F_x + S_x$$

Здесь “+” и “−” теоретико-множественные операции объединения и разности в их обычном смысле.

$$Krit(Y') - Krit(Y) = \sum_{S'} - \sum_{F'} - \left(\sum_{S} - \sum_{F} \right)$$
$$= \sum_{S} + \sum_{F_x} - \sum_{S_x} - \sum_{F} + \sum_{F_x} - \sum_{S_x} - \sum_{S} + \sum_{F} =$$

$$= \sum_{F_x} - \sum_{S_x} + \sum_{F_x} - \sum_{S_x} = 2\left(\sum_{F_x} - \sum_{S_x} \right) = -2 \times V(x) = S_0. \quad (5)$$

В силу включений (3) и (4), показывающих, что F_x, S_x изменят подтверждающиеся закономерности на опровергающиеся, и, наоборот, при использовании значения критерия $V(x)$ для Y, в суммах нужно поменять знаки

$$\sum_{F_x} = -\sum_{R \in F_x} \mu(R) \text{ и } -\sum_{R \in S_x} = \sum_{R \in S_x} \mu(R)$$

Следовательно, изменение критерия в (5) составит $(-2 \times V(x))$.

В другом случае $x \in base(R)$. Для краткости введем обозначение множеств закономерностей, содержащих рассматриваемый признак: ${}^{\varepsilon_1}x^{\varepsilon_2} = \{R: {}^{\varepsilon_1}x^{\varepsilon_2} \in base(R)\}$.

Пока выполнялось $x \notin Y$

$$Reg_{aplic} = Z + \neg x + x' + \neg x',$$

где множество Z никак не задействует x, т.е. $\{R: {}^{\varepsilon_1}x^{\varepsilon_2} \notin base(R)\} \subseteq Z$.

В новом наборе $x \cup Y$ будет

$$Reg'_{aplic} = Z + x + \neg x'.$$

Выразим множество применимых закономерностей для нового набора через множество для данного набора:

$$Reg'_{aplic} = Reg_{aplic} + x + \neg \acute{x} - \neg x - x' - \neg \acute{x} = Reg_{aplic} + x - \neg x - \acute{x};$$

$$Krit(Y') - Krit(Y) = \sum_{Reg'_{aplic}} - \sum_{Reg_{aplic}} =$$

$$= \underbrace{\sum_{x} - \underbrace{\sum_{\neg x} - \sum_{x'}}_{S_2^+}}_{S_1^+}.$$

Положительная сумма S_1^+. Добавятся закономерности R, применимые к набору $x \cup Y =< x, Y_1, \dots, Y_m >$, посылка которых содержит значение признака $x = X_{ji}$, т.е. $P_j = (x_j \approx o_j^i) \in R$. Эти закономерности будут подтверждаться или опровергаться на наборе $x \cup Y$. Как и в случае оценок $V(x)$ они будут учитываться соответственно: $W(x) = \sum_{R \in S} \mu(R)$ для подтверждающихся закономерностей и $W(x) = -\sum_{R \in F} \mu(R)$ для опровергающихся закономерностей.

$$S_1^+ = \sum_{x} \mu(R).$$

Отрицательная сумма S_2^+. Кроме того, исчезнут закономерности, применимые к набору Y, содержащие в посылке $\neg P_j = \left(x_j \not\approx o_j^i\right)$, а значит $x = X_{ji}$ и закономерности, посылка которых содержит $P_j = (x_j \approx o_j^z)$, т.е. $\forall z: o_j^z \neq o_j^i, (x_j, o_j^z) \in Y$. Эти закономерности вносили вклад в общий критерий $Krit < Y_1, \dots, Y_m >$, с соответствующим знаком. Поэтому изменение критерия $Krit < Y_1, \dots, Y_m >$, при добавлении значения x равно

$$S_2^+ = -\left(\sum_{R \in \neg x \cup \acute{x}} \mu(R) \right),$$

где подтверждающиеся закономерности берутся со знаком "+", а опровергающиеся со знаком "−". Удаление этих закономерностей дает знак "−" перед суммой.

В итоге, выигрыш в критерии взаимной согласованности $Krit < Y_1, \dots, Y_m >$, при добавлении значения x^+ равен

$$Krit^+ = W(x^+) = S_0 + S_1^+ + S_2^+.$$

Теперь, рассмотрим случай, когда удаляется одно из значений $x \in \{Y_1, \dots, Y_m\}$ в наборе. Это также может улучшить взаимную согласованность закономерностей.

Аналогично первому случаю, если $x \in target(R)$, определим F_x, S_x. Закономерности $P_{i_1} \& \dots \& P_{i_k} \Rightarrow P_{i_0}$ и $P_{i_0} = (x_{i_0} \approx o_{i_0}^{j_0})$, применимые к набору $Y \backslash x = \{Y_1, \dots, Y_m\} \backslash x, x = X_{i_0 j_0}$ и предсказывающие значение x перестанут на нем подтверждаться после удаления этого значения.

$$S_x = \{R: target(R) = x \notin Y\}; \quad S_x \subseteq S, S_x \nsubseteq S', S_x \subseteq F';$$

Закономерности $P_{i_1} \& \dots \& P_{i_k} \Rightarrow P_{i_0}$ с заключением $\neg P_{i_0} = (x_{i_0} \not\approx o_{i_0}^{j_0})$ или $P_{i_0} = (x_{i_0} \approx o_{i_0}^{z})$, т.е. $\forall z: o_{i_0}^{z} \neq o_{i_0}^{j_0}, (x_{i_0}, o_{i_0}^{z}) \in Y$, применимые к набору $Y \backslash x, x = X_{i_0 j_0}$ и предсказывающие отсутствие значения x, станут на нем подтверждаться после удаления значения x.

$$F_x = \{R: target(R) \neq x \notin Y\}; \quad F_x \subseteq F, F_x \nsubseteq F', F_x \subseteq S';$$

Множество S_x вносило положительный вклад в критерий, а в $Krit(\Gamma)Y'$

$$S' = S + F_x - F_x,$$

значит изменение критерия $Krit < Y_1, \dots, Y_m >$, при удалении значения x дополнительно равно

$$S_{01}^{-} = -2 \times \sum_{R \in S_x} \mu(R),$$

где сумма берется для всех закономерностей, применимых к набору $Y \backslash x$ и предсказывающих значение x.

Закономерности из множества F_x были применимы к набору, но не подтверждались на нем. На уменьшенном наборе, наоборот

$$F' = F - F_x + S_x,$$

Следовательно, изменение критерия $Krit < Y_1, \dots, Y_m >$, при удалении значения x дополнительно равно

$$S_{02}^{-} = -2 \times \sum_{R \in F_x} \mu(R),$$

где сумма берется для всех закономерностей, применимых к набору $Y \backslash x$ и предсказывающих отсутствие значения x. Аналогично первому случаю, можем воспользоваться оценкой V: $S_{01}^{-} + S_{02}^{-} = -2 \times V(x)$.

Рассмотрим Reg_{aplic}:

$$x \cup Y': Reg_{aplic} = Z + x + \neg \acute{x};$$

Если мы удалим последнее значение признака для переменной x_j, то

$$x \notin Y: Reg_{aplic} = Z$$

Иначе останется некоторое значение признака и, следовательно,

$$Reg'_{applic} = Z + \neg x + \neg \acute{x} + \acute{x}.$$

Имеем два аналогичных случая как в пункте 2, только с набором осуществляется обратно действие и знаки перед суммой меняются:

$$Krit(Y') - Krit(Y) = -\sum_{x} - \sum_{\neg \acute{x}} = -\sum_{x} + \sum_{-x} + \sum_{\acute{x}}.$$

В случае $x \in base(R)$ меньше закономерностей становятся применимыми к набору Y', которые подтверждались или опровергались на Y и вносили вклад в общий критерий $Krit < Y_1, \dots, Y_m >$, с соответствующим знаком. Поэтому изменение критерия при удалении значения x равно

$$S_1^{-} = -\left(\sum_{R \in x} \mu(R)\right),$$

где подтверждающиеся закономерности берутся со знаком “+”, а опровергающиеся со знаком “−”. Удаление этих закономерностей дает знак “−” перед суммой.

Кроме того, если какой-то X_{ji} принадлежит начальному набору, то добавятся закономерности имеющие в посылке $\neg P_j = (x_j \not\approx o_j^i)$ для $x = X_{ji}$ или $P_j = (x_j \approx o_j^i)$. Закономерности будут подтверждаться или опровергаться на этом наборе. Поэтому изменение критерия $Krit < Y_1, \dots, Y_m >$ при удалении значения X_{ij} равно

$$S_2^- = -\left(\sum_{R \in \neg x \cup \acute{x}} \mu(R) \right),$$

где подтверждающиеся закономерности берутся со знаком “+”, а опровергающиеся со знаком “−”.

В итоге, изменение критерия взаимной согласованности $Krit < Y_1, \dots, Y_m >$ при удалении некоторого значения $x^- \in \{Y_1, \dots, Y_m\}$ равно

$$Krit^- = W(x^-) = S_{01}^- + S_{02}^- + S_1^- + S_2^-.$$

Заметим, что при использовании свойства сильных закономерностей в оценке W его следует соблюдать и в оценке V.

При расширении множества СВЗ или объединении нескольких СВЗ-множеств свойство сильных закономерностей может быть потеряно. Чтобы сформировать множество применимых СВЗ, нужно сначала выбрать множество просто применимых закономерностей, затем для каждой закономерности искать подправила. Заметим, что будет ошибочным, проводить выборку СВЗ только для подмножеств оценки W, а оценку V считать по всем применимым законам.

Возьмем некоторый исходный набор $< Y_1, \dots, Y_m >$. Найдем максимальное значение увеличения критерия $Krit < Y_1, \dots, Y_m >$ при добавлении некоторого значения x^+ к набору и при удалении некоторого значения x^-.

Максимальные оценки $W(x^+)$ и $W(x^-)$ могут быть как положительными, так и отрицательными или равняться нулю.

1.Если обе величины $W(x^+)$ и $W(x^-)$ меньше либо равны 0, то локальный максимум найден, т.е. найдена *неподвижная точка,* и мы переходим к следующему исходному набору на первый шаг, предварительно удалив все признаки, дающие нулевой вклад в изменение критерия. Незначащие признаки мы удаляем на последнем шаге изменения набора, когда уже достигнут локальный максимум критерия.

2.Иначе выполнены условия: $W(x^+) > 0, W(x^-) > 0$. Сначала необходимо убедиться, что x^+ и x^- есть признаки, которые явно предсказываются, т.е. для $x^+ = X_{i_0 j_0}$ должна найтись закономерность из множества применимых к набору, вида $P_{i_1} \& \dots \& P_{i_k} \Rightarrow P_{i_0}$, и $P_{i_0} = (x_{i_0} \approx o_{i_0}^{j_0})$, а для $x^- = X_{v_0 t_0}$ должна найтись закономерность вида $P_{v_1} \& \dots \& P_{v_l} \Rightarrow \neg P_{v_0}$, и $\neg P_{v_0} = (x_{v_0} \not\approx o_{v_0}^{t_0})$.

Явное предсказание быстрее всего искать перебором корневых вершин СВВ деревьев.

- Если найдутся закономерности, где в заключении присутствует x^+, и закономерности с x^- в заключении, то смотрим какая из величин $Krit^+$ или $Krit^-$ больше.
- Если какой-то из признаков, допустим x^+, явно не предсказывается, то прирост критерия $Krit^+$ берется для того признака, который был рассмотрен на шаге 3 и явно предсказывается. Далее, если новое значение $Krit^+ > 0$, то снова сравниваем величины $Krit^+$ и $Krit^-$ как ранее, иначе получаем случай одной положительной оценки, рассматриваемый далее.
- Если для обоих случаев (вставки и удаления) нет признаков, для которых бы нашлись закономерности их предсказывающие, то $Krit^+ = Krit^- = 0$ и *найдена неподвижная точка* и $Pr^{n+1}\left(X\left(o^i\right)\right) = Pr^n X\left(o^i\right)$.

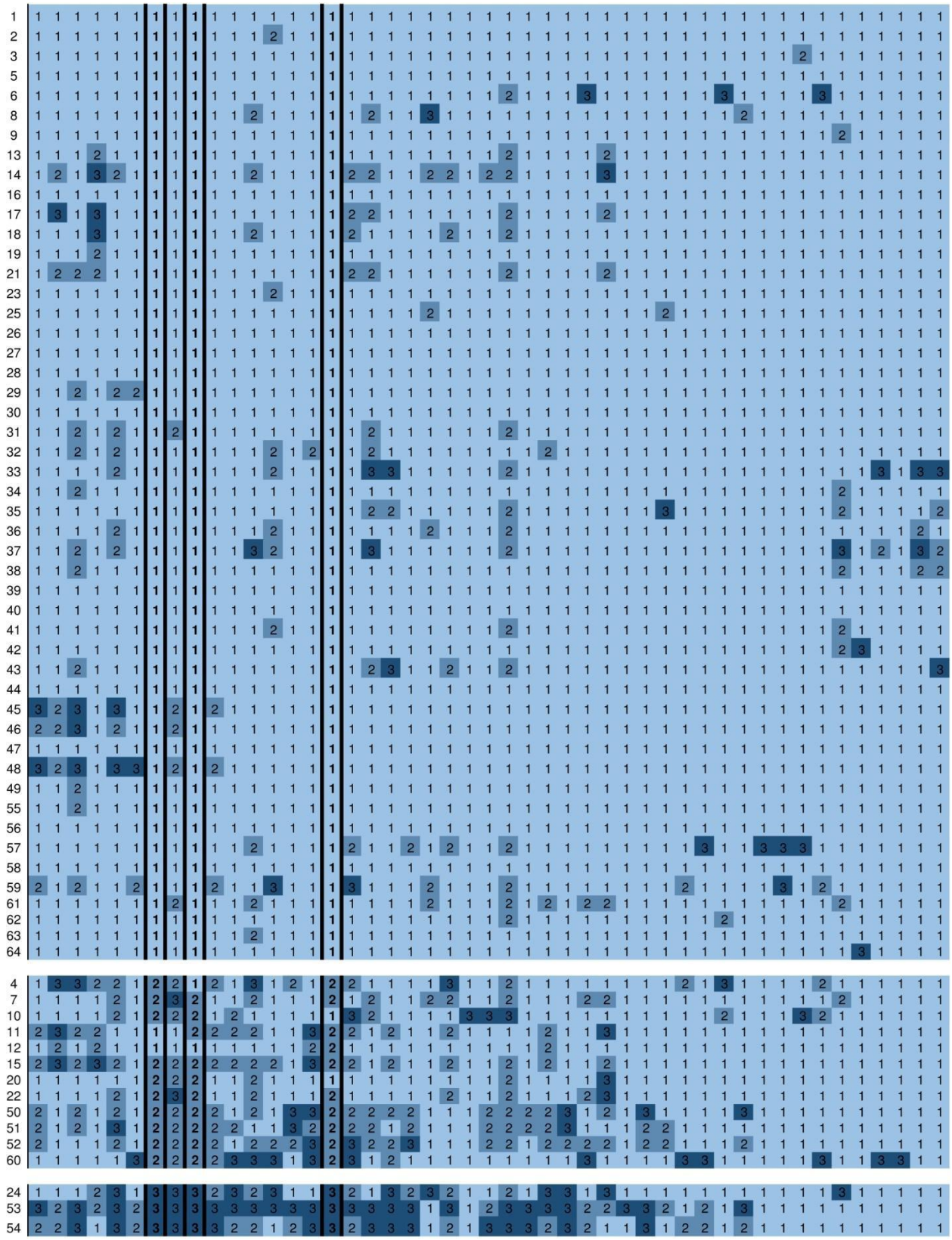


Рисунок 2. Данные после интервалирования

3.Если выполнено одно из двух условий, к примеру, $W(x^+) > 0$, то либо x^+ явно предсказывается и мы добавляем его к набору Y, либо найдется $\tilde{x}: W(\tilde{x}) > 0$ и закономерность ее предсказывающая, либо поиск идеального набора закончен.

Чтобы достичь максимума критерия $Krit$ *нужно*: сначала удалить/вставить одно значение, которое дает максимальное увеличение критерия, т.е. в зависимости от того какая из величин $W(x^+)$ или $W(x^-)$ больше вставить или удалить значение, затем

пересчитать обе оценки и снова вставить/удалить то значение, для которого одна из величин $W(x^+)$ или $W(x^-)$ больше.

Классификация археологических данных

Покажем как метод «естественной» классификации применим к реальной задаче. Исследуем структуру 64 орудийных комплексов Ближнего и Среднего Востока и Кавказа. Мы знаем, какое количество артефактов определенного типа орудий было найдено на каждом из них. Всего типов орудий 47.

Исходные данные имели интервальную природу, и чтобы применить метод описанный выше, нужно разбить интервал всех возможных значений на небольшое число интервалов и в дальнейшем работать не с точными значениями, а с интервалами. Разбиение такого типа можно построить с помощью автоматического группирования [Ростовцев, Костин, 1995]. Задаем максимум 3 группы для каждого интервала. Для каждого типа орудий первая группа (интервал содержащий 0), интерпретируем как пропуск в данных. На этом этапе некоторые минимальные значения будут отсеяны. Мы полагаем, что если они были отнесены к группе с 0, то их влияние можно больше интерпретировать как шум, нежели признаки привносящие качество в классификацию.

Класс 1				Класс 2	Класс 3
Таглар 4а	Сефуним А	Ябруд 10	Таглар 4б	Кеу I сл. II	Сефуним С
Амуд B4	Сефуним 12	Кударо I 3а	Таглар 5	Кзар-Акил XXVIA	Ортвала-Клде V
Амуд B2	Сефуним VI	Кударо I 3б	Таглар 6	Кзар-Акил XXVIIB	Ортвала-Клде VI
Кеу I сл. l	Сефуним В	Кударо I 3в	Ортвала-Клде I	Кзар-Акил XXVIII	
Кеу I сл III	Ябруд 2	Кударо I 4	Ортвала-Клде VII	Кзар-Акил XXVIII	
Кеу I сл V	Ябруд 3	Каркустакау	Двойной Грот	Варвази В	
Кзар-Акил XXVIB	Ябруд 4	Тамарашени	Азых 3 сл.	Сефуним 13	
Кзар-Акил XXVIIA	Ябруд 5	Монашеская	Среднехаджохская	Сефуним VII	
Кунджи	Ябруд 6	Губский Навес	Азых 6 сл.	Ортвала-Клде II	
ВарвазиА	Ябруд 7	Малая Воронцовская	Лусакерт D	Ортвала-Клде III	
Варвази С	Ябруд 8	Таглар 2 сл.	Лусакерт А	Ортвала-Клде IV	
Варвази D	Ябруд 9	Таглар 3 сл.	Газма	Медвежье	
			Баракаевская		

Таблица 1. Разбиение орудийных комплексов на классы

На рисунке 2 изображены данные после интервалирования, где различными цветами обозначены полученные группы. Группа 1 содержит 0 и интерпретируется как пропуск. Более темный цвет соответствует большему числу орудий. Найденные вероятностные закономерности опираются на 3 типа орудий: угловатые скребла (x_7), усеченные отщепы (x_9) и резцы (x_{16}) – столбцы соответствующие этим типам выделены на рисунке вертикальными границами и жирным начертанием. Выпишем закономерности, которые образуют второй класс орудийных комплексов (в описании закономерностей используются интервалы для исходных данных):

$N(x_7) \in [54; 157] \Rightarrow N(x_9) \in [15; 33]$, $N(x_7) \in [54; 157] \Rightarrow N(x_{16}) \in [2; 3]$,

$N(x_9) \in [15; 33] \Rightarrow N(x_7) \in [54; 157]$, $N(x_9) \in [15; 33] \Rightarrow N(x_{16}) \in [2; 3]$,

$N(x_{16}) \in [2; 3] \Rightarrow N(x_7) \in [54; 157]$, $N(x_{16}) \in [2] \Rightarrow N(x_9) \in [15; 33]$,

где $N(x)$ - число орудий типа x.

Приведенное множество закономерностей согласовано по предсказанию. В частности это значит, что в памятнике Кеу I сл. II число усеченных отщепов (x_9) должно принадлежать к группе 2, так как из $N(x_7)$ и $N(x_{16})$ для этого памятника следует, что $N(x_9) \in [15; 33]$.

Данные на рисунке упорядочены по строкам так, чтобы нагляднее увидеть результат классификации. В первом столбце указана начальная нумерация орудийных комплексов. Классы разделены пустой строкой.

Опираясь на выделенные 3 типа орудий «естественная» классификация разделила орудийные комплексы на 3 класса (см. таблицу 1).

ЛИТЕРАТУРА

Аптон Г. Анализ таблиц сопряженности. М.: Финансы и статистика, 1982, 143 с.

Витяев Е.Е. Классификация как выделение групп объектов, удовлетворяющих разным множествам согласованных закономерностей // Анализ разнотипных данных (Выч. сист. 99), Новосибирск, 1983, с.44-50.

Витяев Е.Е. Извлечение знаний из данных. Компьютерное познание. Модели когнитивных процессов. НГУ, Новосибирск, 2006, с.293.

Витяев Е.Е. Синтез логики, вероятности и обучения в семантическом вероятностном выводе // Сборник научных трудов IV Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (28-30 мая, Коломна, 2007), т.7, с.133-140.

Витяев Е.Е., Неупокоев Н.В. Формальная модель восприятия и образа как неподвижной точки предвосхищений // Подходы к моделированию мышления. (сборник под ред. д.ф.-м.н. В.Г. Редько). УРСС Эдиториал, Москва, 2014, с.152-168.

Витяев Е.Е., Перловский Л.И., Ковалерчук Б.Я., Сперанский С.О. Вероятностная динамическая логика мышления // Нейроинформатика, том 5, № 1, 2011, с.1-20.

Кендал М., Стюарт А. Статистические выводы и связи. М.: Наука, 1973, с.899.

Ростовцев П.С., Костин В.С. Автоматизация типологического группирования. Новосибирск: ИЭ и ОПП СО РАН, 1995, с.46.

Kovalerchuk B., Vityaev E. Data Mining in Finance: Advances in Relational and Hybrid methods. New York: Kluwer Academic Publishers, 2000, p.308.

Halpern J.V. An analysis of first-order logic of probability.//Artificial Intelligence 46: pp.311-350, 1990.

Черепанов Е.М

Некоторые замечания к понятию «мысль»

В предлагаемой статье рассмотрены некоторые аспекты логического анализа Г.Фреге понятия мысль. Рассмотрен вопрос – что означает, что два утверждения выражают одно и ту же мысль, как в одном и том же языке, так и в различных языках. Рассмотрен вопрос о простоте выражения мысли. Автор также делает попытку анализа некоторых пунктов возражений к подходу Фреге. Кроме того, на основе понятия мысль автор предлагает некоторое дополнение к определению понятия знание.

Ключевые слова: *Мысль, мышление, смысл, содержание, простота, знание.*

1. Подход Г. Фреге.

Несомненно, что самым фундаментальным логическим анализом понятия *мысль* является работа одного из столпов математической логики Г.Фреге. В своей статье «Мысль: логическое исследование» [Фреге, 1997: 50-75] он предлагает основательный анализ этого понятия и предлагает вполне обоснованную концепцию для понимания того, что мы имеем в виду, говоря, что некоторый текст выражает вполне определенную мысль. Мысль, как таковая, является результатом мышления, но понятие мысль и понятие мышление не всегда многими различается. На сегодняшний день вполне устоявшейся формулировкой соотношения этапов мышления является следующая:

1. постижение мысли - мышление ;
2. признание истинности мысли - суждение ;
3. выражение этого суждения - утверждение.

Под мыслью в концепции Г.Фреге понимается не субъективный акт мышления, а объективное содержание результата мыслительного акта, способное стать достояние многих. Результатами мышления могут быть самые различные акты. Результатом мышления может быть действие, эмоциональная реакция (возмущение, восхищение, смех и т.д.), вопрос и прочее, к чему может привести мыслительный акт, но нас, как и Г.Фреге в первую очередь будет интересовать результат мышления, выраженный с помощью

утверждения в фиксированном языке некоторого содержания. Поэтому, в концепции Фреге к утверждениям, выражающим мысль, можно относить лишь те утверждения, которые могут быть соотнесены с понятием истинность (или ложности), а также с понятием возможности такого соотнесения (к таким относятся предположительные утверждения). Как утверждает сам Г.Фреге: «Когда мы называем предложение истинным, мы имеем в виду, собственно, его смысл. Отсюда следует, что та область, в которой применимо понятие истинности, – это смысл предложения» [Фреге, 1997: 53]. В этом отношении становится возможным говорить о том, что мысль должна иметь отношению к чему-то конкретному. Ясно также, что утверждение, выражающее мысль, это интерпретированное утверждение. Хотя термины *мысль* и *смысл* однокоренные термины, но, тем не менее, следует отметить, что не всякое осмысленное утверждение выражает мысль. С помощью предложений некоторого языка можно выразить широкий спектр осмысленных утверждений. Такие, например, осмысленные утверждение как выражение желания, приказ, выражение восхищения и т.п. мысль не выражают, так как к этим утверждениям понятие истинности не применимо. Так, например, утверждение «Я так рад Вас видеть», выражающее эмоциональную реакцию, решительным образом не выражает никакой мысли, так как это утверждение является актом эмоциональной реакции, но не является результатом мышления и, кроме того, невозможно придать этому утверждению однозначный смысл, то есть достоверно убедиться – соответствует это действительности или нет. В равной мере можно как верить такому сообщению, так и не верить. Именно по этой причине Г.Фреге считает необходимым различать утверждения относительно внутреннего мира человек от утверждений относительно внешнего мира. Внутренний мир образуют чувственные впечатления, создание воображения, ощущения, эмоции, настроения, а также мира склонностей и решения. Все эти компоненты, за исключением решений, он называет «представлениями». Представление о предмете есть его внутренний образ и даже для одного человека определенное представление не всегда связано с одним и тем же смыслом. Представление субъективно и представление одного человека не то же самое, что представление другого человека. Отличие представлений ль вещей внешнего мира, по мнению Фреге, состоит в следующем:

а) представления не могут быть восприняты органами чувств;

б) представление, которым обладает некоторый человек, составляет содержание его сознания;

в) представления требуют существования носителя представления. Вещи же внешнего мира в этом отношении являются автономными;

г) всякое представление имеет только одного носителя и никакие два человека не обладают одним и тем же представлением.

Быть содержанием сознания какого-то человека – настолько существенное свойство любого представления, что уже в силу этого факта принадлежности отличается от представления другого человека. Для каждого человека невозможно сравнение чужих представлений с его собственными. Различные люди, наблюдая одну и ту же вещь, формируют различные представления о ней. Боль, испытываемая одним человеком, не может принадлежать никому другому. Кто-то другой может по этому поводу испытывать сострадание, но при этом всегда боль принадлежит одному, а сострадание другому.

Исключение решений из мира представлений основывается на том, что решения, являясь результатом мыслительной деятельности, могут относиться как к внешнему, так и внутреннему миру. Например, можно принять решение сформулировать мысль или совершить тот или иной поступок, но и в том и другом случае эти акты нельзя соотнести с представлением. Главное отличие представлений от вещей внешнего мира состоит в том, что представление имеет только одного носителя – никакие два человека не обладают одним и тем же представлением – иначе бы представления существовали независимо от людей. По этой причине мысль не может являться представлением, так как, к примеру,

мысль, выраженная в теореме Пифагора, признается истинной многими людьми. В противном случае смысл теоремы Пифагора был различен, в зависимости от носителя.

В классификации предложений могущих иметь отношение к выражению мысли особое место занимают обще вопросительные предложения, имеющие непосредственное отношение к научной деятельности. Такого рода предложения состоят из двух компонент. Первая компонента вопросительная, которая требует ответа «да» или «нет», а вторая компонента утвердительная, к которой собственно вопрос и относится. Как правило, утверждения, выражающие ту или иную мысль, могут быть преобразованы в вопросительное предложение в тех случаях, когда требуется установление истинности или ложности этого утверждения. Если не удается такого рода установление, то вполне возможно, что мы имеем дело с вопросом, который возможно не имеет смысла. Однако следует различать два случая. Первый – это принципиальная невозможность установления истинности или ложности утвердительной части общего вопроса и, второй – невозможность установления этого статуса в данный момент. Примером второго случая могут быть вопросы, имеющие отношение к гипотезам, которые выражают мысль, которую на данный момент не предоставляется возможным ни подтвердить, ни опровергнуть. К таким утверждениям, к примеру, можно отнести современную физическую гипотезу теории суперструн, или же утверждения подобные таким как, что «на Марсе будут яблони цвести». Такого рода утверждения и являются *предположительно* истинными или ложными. То есть, такого рода утверждения выражают вполне определенную мысль, в то время как утверждения принципиально не верифицируемые, следуя этой логике, мысль не выражают и поэтому являются бессмысленными утверждениями. Прогресс в науке обычно происходит так, что вначале постигается мысль, выражаемая, например, в виде общего вопроса; и только впоследствии, после необходимых исследований, эта мысль признается истинной. Признание истинности мы выражаем в форме утвердительного предложения. При этом слово «истинный», в общем-то, и не требуется, так как, к примеру, принимаемая гипотеза может быть в дальнейшем фальсифицирована. К обще вопросительным утверждениям близки по сути и предположительные утверждения типа «я полагаю, что утверждение А истинно», «я думаю, что утверждение А истинно », «я верю, что утверждение А истинно » и тому подобные, которые можно трансформировать в обще вопросительное предложение вида «действительно ли, что…?» . Утвердительное предложение помимо собственно утверждения и выраженной с с помощью него мысли может содержать и еще один компонент, на который само утверждение не распространяется – это компоненты, облегчающие восприятие утверждения. Это, к примеру, языковые конструкции «следует отметить, что…», «следует обратить внимание…» и тому подобное. Кроме этого, в качестве фактора, облегчающего восприятие мысли, может служить как порядок слов в предложении, так и интонационные оттенки – если что-то утверждается голосом. В точных науках такого рода обороты встречаются заметно реже, нежели в гуманитарных науках или в поэзии и литературе. Такого рода компоненты не влияют на утвердительную силу предложения, но, тем не менее, они вполне оправданы для любой области мыслительной деятельности. Все это показывает, что содержание предложения может быть «шире», чем выраженная в нем мысль. Возможно также и обратное, что представленное утверждение является недостаточным для выражения требуемой мысли. Здесь естественно возникает вопрос: каким образом и в каких случаях имеется возможность определить, что представленное утверждение является недостаточным для выражения вполне определенной мысли? Из обыденного опыта мы знаем, что относительно непонятного для нас утверждения мы задаем дополнительные вопросы для уточнения и прояснения содержания утвердительного предложения. В том случае, когда требуемая мысль не выражена, может также и оказаться, что выражена некоторая другая мысль и мы воспринимаем совсем не то, что было намерено высказать. Ситуация в каком-то смысле проясняется тем обстоятельством, что в концепции Г.Фреге мысль

понимается как *утверждение о положении дел в мире*. Например, рассмотрим, в не строгом смысле, утверждение: «Кошки серые». Это утверждение является недостаточным для выражения одной из двух мыслей – «Все кошки серые» или «Существуют серые кошки». Одно из этих утверждений является ложным. Может оказаться, что высказываемое утверждение не является достаточным для выражения истинного положения дел. Например, то же самое утверждение «Кошки серые» является недостаточным для выражения мысли «Ночью все кошки кажутся серыми». В приведенных примерах мы сталкиваемся с тем, что утверждение, которого не достаточно для выражения требуемой мысли, является подформулой утверждения, выражающего требуемую мысль адекватно. Однако всегда остается вопрос о том, насколько адекватно утверждение, выражающее мысль, фактическому положению дел и существуют ли критерии такого рода адекватности. Это собственно вопрос о степени адекватности модели, описывающей некий фрагмент реальности. Кроме этого, возможен и такой случай, когда утверждение, являющееся недостаточным, для выражения одной мысли, является вполне достаточным для выражения другой мысли. Однако определить это обстоятельство не во всех случаях является возможным.

Резюмируя все вышесказанное, приведем рассуждения на эту тему Л.Виттгенштейна [Витгенштейн, 1958: 47-53]:

3.Логический образ фактов есть мысль.

3.04 Априори верной мыслью была бы такая, возможность которой обеспечивала ее истинность.

3.05. Априори знать, что мысль истинна мы могли бы только тогда, когда ее истинность познавалась бы из самой мысли (без объекта сравнения).

3.328. Если знак *не необходим* то он не имеет значения. В том смысл «бритвы» Оккама.

4.Мысль есть осмысленное предложение.

2. Содержательность мысли.

Таким образом, становится понятным, что содержанием мысли является суждение о некотором фактическом положении дел в мире или о факте. Термин мир в данном случае является достаточно широким. В зависимости от контекста, в котором выражена та или иная мысль это может быть мир физических объектов, мир математических объектов, мир поэзии и многие другие области, относительно которых мы делаем те или иные утверждения. Во многих случаях мы сталкиваемся с утверждением о *содержательности* той или иной мысли. Естественным представляется понимать это стандартным образом, то есть определять содержательность мысли множеством всех дедуктивных следствий, выражающего эту мысль утверждения. При таком определении содержательности мысли приобретает вполне определенный смысл утверждение, что одна мысль более содержательна, чем другая, естественно, при условии, что обе эти мысли претендуют на адекватное суждение об одном и том же факте реальности. Причем следует отметить, что если суждение не является верным, то это обстоятельство также порождает свои следствия, которые могут оказаться важными. Достаточно проблематичными являются области суждений, в которых понятия истинности или ложности носят в достаточной мере условный характер. Как, к примеру, быть с утверждениями, выражающими вполне определенную мысль в области искусства? Допустим, мы имеем дело с суждением, что какое-то произведение искусства имеет эстетическую ценность. Это есть суждение о некотором факте, например, в мире поэзии. Это суждение выражает вполне определенную мысль? Несомненно – да. Каким же образом в подобных областях решается вопрос – верна эта мысль или нет, что условно можно соотносить с истинностью? На взгляд автора в этом случае мы сталкиваемся с чем-то отдаленно напоминающее понятие форсинга в теории множеств и истинность или некое подобие истинности такого рода утверждений основывается на схеме: мнение субъекта A, выраженное суждением φ определяет (вынуждает) истинность суждения ψ. Истинность подобных рода утверждений носит

экспертный характер. Таким образом, всякую мысль, выраженную в фиксированном языке L утверждением φ, можно связать с понятием истинности или ее неким подобием и содержанием этой мысли является суждение о некотором факте интересующего нас фрагмента реальности.

3.Сравнение утверждений, выражающих мысль

Что значит, что два утверждения выражают одну и ту же мысль и во всех ли случаях это возможно? Естественно полагать, что два утверждения φ и ψ выражают одну и ту же мысль, если это суждения об одном и том же факте и они эквивалентны, то есть:

$Q_1 x_1 ... Q_n x_n (\varphi(x_1,...,x_n) \leftrightarrow \psi(x_1,...,x_n))$

где $Q_1,...,Q_n \in \{\forall, \exists\}$. Такое понимание эквивалентности суждений, выражающих одну и ту же мысль, справедливо как в случае научного, так и обыденного языка. Для того, чтобы понять в каких случаях эта эквивалентность возможна рассмотрим дедуктивные замыкания обоих утверждений – $T_1 = Th(\varphi)$ и $T_2 = Th(\psi)$ и рассмотрим все случаи сравнения этих теорий. . Приведем все необходимые определения [Смирнов, 1987: 32-36].

3.1Сравнение теорий в одном языке.

Рассмотрим отношения между теориями, сформулированными в одном и том же языке, т.е. с одними и теми же правилами образования выражений и с одним и тем же словарем внелогических терминов.

Две теории T_1 и T_2 будем называть *несовместимыми*, если и только если теория $Th(T_1 \cup T_2)$ противоречива.

Две теории T_1 и T_2 будем называть *независимыми*, если и только если $T_1 \cap T_2 = Th(\varnothing)$. Независимые теории не имеют общего нелогического содержания. Из этого определения следует, что $T(\varnothing)$ независима от любой теории, в том числе и от самой себя.

Возможны следующие взаимоисключающие отношения между двумя теориями (сформулироваными в одном и том же языке с одним и тем же словарем нелогических терминов):

1) T_1 эквивалентна T_2, т.е. $T_1 = T_2$;

2) T_1 является собственной подтеорией T_2, т.е. $T_1 \subset T_2$;

3) T_2 является собственной подтеорией T_1, т.е. $T_2 \subset T_1$;

4) Ни одна из теорий не является подтеорией другой - $\exists\varphi\exists\psi(\varphi \in T_1 \,\&\, \varphi \in T_2 \,\&\, \psi \notin T_1 \,\&\, \psi \in T_2$, т.е. $T_1 \perp T_2$.

Нетрудно видеть, что если T_1 и T_2 несовместимы, то $T_1 = T_2$ тогда и только тогда, когда T_1 и T_2 противоречивы; $T_1 \subset T_2$ тогда и только тогда, когда T_1 непротиворечива и T_2 противоречива; $T_1 \perp T_2$ тогда и только тогда, когда T_1 и T_2 непротиворечивы.

Аналогично, если T_1 и T_2 независимы, то $T_1 = T_2$ тогда и только тогда, когда $T_1 = Th(\varnothing)$ и $T_2 = Th(\varnothing)$; $T_1 \subset T_2$ тогда и только тогда, когда $T_1 = Th(\varnothing)$ и $T_2 = \neg Th(\varnothing)$; $T_1 \perp T_2$ тогда и только тогда, когда $T_1 = \neg Th(\varnothing)$ и $T_2 = \neg Th(\varnothing)$.

3.2 Сравнение теорий с помощью определений.

До сих пор мы сравнивали теории, имеющие один и тот же язык. Теперь мы имеем возможность сравнивать теории, сформулированные с одной и той же грамматикой, но с разными словарями. Прежде всего введем понятие дефинициального расширения.

T_2 есть *дефинициальное расширение* T_1, если и только если существует множество Δ определений терминов теории T_1, отсутствующих в T_2, в терминах теории T_2, такое что $T_2 = Th(T_1 \cup \Delta)$.

Будем говорить, что теория T_1 *дефиницаально эквивалентна* теории T_2 если и только если существуют определения D_{T_2} терминов теории T_2 в терминах теории T_1 и определения D_{T_1} терминов теории T_1 в терминах теории T_2, такие, что $Th(T_1 \cup D_{T_2})$ эквивалентна $Th(T_2 \cup D_{T_1})$.

Очевидно, что теория T_1 дефиниционально эквивалентна теории T_2, если и только если существуют дефиниционные расширения этих теорий, эквивалентные друг другу.

Если $T_1 \subseteq T_2 + D_{T_1}$, то будем говорить, что T_1 *дефиниционально интерпретируема* в T_2. Встает следующий вопрос: если T_1 дефиниционально интерпретируемы в T_2 и T_2 дефиниционально интерпретируема в T_1, то следует ли отсюда, что T_1 дефиниционально эквивалентна T_2? Ответ отрицательный (теории дефиниционально сравнимы).

Будем говорить, что T_1 дефиниционально вложима в T_2, если и только если существует система определений D_{T_1} терминов теории T_1 в термина теории T_2, такая, что $T_2 + D_{T_1}$ является консервативным расширением теории T_1.

Пусть T_1 дефиниционально вложима в теорию T_2. Единственно ли такое вложение? Может оказаться, что существует система определений D_1 такая, что $T_2 + D_{T_1}$ есть консервативное расширение T_1 и есть система определений D_2, такая, что $T_2 + D_2$ так же есть также консервативное расширение T_1. Возникает вопрос: какое вложение является «правильным»? Ответ Куайна: нет привилегированного перевода, привилегированного способа вложения. Но можно показать, что существуют способы перехода от одного вложения к другому.

Сформулируем это более точно. Пусть T_1 - некоторая теория и T_1^* - ее переформулировка в других терминах, отличных от терминов теории T_1. Пусть $T_2 + D_1$ есть консервативное расширение T_1 и $T_2 + D_2$ есть консервативное расширение T_1^*. Тогда $T_2 + D_2$ дефиниционально эквивалентна $T_2 + D_1$. Ситуация аналогична ситуации с системами измерений. Можно измерять в футах, метрах, аршинах, но ни одна из систем сама по себе не имеет привилегированного положения. Однако мы можем переходить от одной системы к другой. То же самое имеет место относительно вложения одной теории в другую с помощью определений. В связи с обсуждаемой проблемой следует отметить, что каждый раз следует пользоваться только одной системой определений, как и одной системой измерений. Существование различных систем определений, с помощью которых одна теория вкладывается в другую, не приводит ни к каким противоречиям.

Таким образом, из приведенных определений видно, в каких возможных случаях рассматриваемые утверждения дефиниционально эквивалентны, а, следовательно, эти утверждения выражают одну и ту же мысль. Кроме этого из этих определений следует, что в дефиниционально сравнимых теориях не существует гарантий, что сравниваемые утверждения выражают одну и ту же мысль. В этом и есть суть тезиса Куайна о невозможности радикального перевода.

4. Простота утверждений, выражающих мысль.

Рассмотрим еще один способ сравнения утверждений. Теперь к факторам, облегчающим понимание утверждения выражающего мысль, помимо не формализуемых факторов, таких как интонационные, акцентирующие, эмоциональные и прочие, можно добавить вполне формализованный фактор простоты выражения мысли. Постулировав, что во многих случаях в более простом утверждении легче понять выраженную в нем мысль, мы приходим к заключению, что из двух эквивалентных утверждений, выражающих одну и ту же мысль, следует предпочесть более простое. То есть, если

обозначить через $\nu(\varphi)$ значение сложности формулы φ, то если $\nu(\varphi) < \nu(\psi)$, то следует предпочесть утверждение φ. Способ измерения сложности утверждений предложен автором в статье «Простота как критерий убедительности доказательства» [Черепанов, 2010]. В связи с этим, находит точное выражение принцип «экономии мышления» в случае, если процесс мышления представим некоторой последовательностью утверждений, отображающих этот процесс. Допустим, к одной и той же мысли мы можем придти различными способами. Пусть есть две последовательности рассуждений $\{\varphi_1,...,\varphi_n\}$ и $\{\psi_1,...,\psi_m\}$, причем $(\varphi_n \leftrightarrow \psi_m)$ есть эквивалентные утверждения, выражающие одну и ту же мысль. Тогда если $\nu(\varphi_1,...,\varphi_n) < \nu(\psi_1,...,\psi_m)$, то к выражению мысли φ мы пришли более экономным способом $\{\varphi_1,...,\varphi_n\}$. Способ измерения сложности, такого рода последовательностей формул, также представлен в статье автора [Черепанов. 2010]. Такого рода последовательности могут являться и доказательствами, обосновывающими истинность или ложность утверждения, выражающего мысль. Тем не менее, следует отметить, что не всякая такого рода последовательность является валидной, то есть проводящая к правильному умозаключению. Этот аспект и является основным в обосновании знания, которое и выражается с помощью суждений о некотором фрагменте реальности.

4. О критике подхода Г.Фреге.

Несмотря на естественность и логичность концепции Г.Фреге существуют упреки в «нестандартности» понимания им понятия мысль. Основные пункты возражения в статье В.А.Бочарова «Понятие, суждение и мысль» [Бочаров, 2001: 110] следующие:

«… Во-первых, здесь необычно употребляется термин «мысль». В самом деле, если мы различаем мысль и содержание мысли, то термином «мысль» следовало бы называть именно суждение, а не его содержание, Во-вторых, Фреге сужает сферу использования термина, так понятия, вопросы и императивы тоже относятся к области мыслей. В-третьих, говоря об объективном характере мыслей, Фреге вынужден утверждать их независимое существование, что вряд ли философски оправдано, В-четвертых, у Фреге существует несогласованность в трактовке суждений и понятий, а ведь если и то и другое – мысли, то тогда они (как мысли) характеризоваться взаимоподобно.»

Относительно первого возражения можно лишь повторить высказанное автором этой статьи его понимания концепции Фреге, состоящее в том, что Мысль выражается интерпретированным утверждением, а содержанием мысли является суждение о некотором положении дел.

Ошибочность второго возражения, требует более детального рассмотрения.

Рассмотрим языковые конструкции, выражающие императив. Для начала следует уточнить смысл понятия «императив». Итак, «императи́в» (лат. imperativus) — требование, приказ, закон. С появлением кантовской «Критики практического разума» императив — это общезначимое предписание, в противоположность личному принципу (максиме); правило, выражающее долженствование (объективное принуждение поступать так, а не иначе). Императивы разделяются на категорические и гипотетические. Гипотетический императив имеет силу лишь при известных условиях; категорический императив выражает безусловное, неуклонное долженствование, он устанавливает форму и принцип, которым нужно следовать в поведении. Категорический императив, или императив нравственности, формулируется Кантом следующим образом: «Поступай так, чтобы максима твоей воли в любое время могла стать принципом всеобщего законодательства». Таким образом, императив есть некоторая интерпретированная языковая конструкция, которая выражает либо долженствование действия, либо запрет какого-либо действия как, например, одна из библейских заповедей «Не убий». Выражение императива всегда является результатом мыслительного акта, но является ли выражение императива выражением вполне определенной мысли? Если принять допущение, что императив как языковая конструкция выражает мысль, то компьютерная программа, предписывающая

компьютеру действие как некоторую вычислительную процедуру, выражает вполне определенную мысль, то есть компьютер «понимает» мысль человека. Такое допущение автору кажется сомнительным. Несомненно, что некоторые императивные выражения допускают их преобразования в утверждения, выражающие мысль. Как, например, императив «Не убий» как запрет на убийство себе подобных может быть преобразован в обосновывающее этот императив утверждение «Убийство себе подобного приводит, с точки зрения эволюционной теории, к исчезновения вида». Это утверждение выражает вполне определенную мысль и может быть либо истинным, либо ложным, так как отражает положение дел в этом мире. Это как раз пример того, как мысль порождает императив. В подобных случаях императив есть следствие мысли, но не тождественен ей. Императив может быть также выражением желания, принятого решения и т.д.

Относительно вопросительных предложений аргументация Г.Фреге автору кажется вполне достаточной. В вопросительном предложении, в общем случае, выражается в его вопросительной части желание обоснования его утвердительной части.

Рассмотрим теперь вопрос – выражает ли *понятие* мысль, являясь результатом мыслительной деятельности?

В общем случае, формулированное понятие является результатом мышления, выраженное вполне определенной формой, с помощью которой объекты выделяются и обобщаются по их существенным признакам. Понять нечто, то. есть составить понятие об объекте мышления, означает возможность выражения сущности этого объекта. Этим понятие отличается от других познавательных форм – ощущения, восприятия, представления, которые не обладают такой обобщающей и абстрагирующей силой и, следовательно, в своем содержании не могут выразить закономерностей.

Как логическая форма понятие характеризуется двумя важнейшими параметрами – содержанием и объемом.

Содержание понятия – это совокупность существенных признаков объектов, на основании которых они выделяются и обобщаются вполне определенным образом. Объем понятия – это объект или совокупность объектов, обладающих признаками, составляющими содержание понятия. Совокупность предметов, охватываемая объемом понятия, называется логическим классом, или множеством, а отдельный предмет объема понятия – элементом класса (множества). Класс (множество) может включать в себя подклассы, или подмножества.

Содержание понятия задается с помощью определения. Определение — это логическая процедура придания строго фиксированного смысла терминам языка. Термин, над которым проводится операция дефиниции, называется дефидентом. Таким образом, определение или дефиниция – это логическая операция, раскрывающая содержание понятия.

Поскольку содержание понятия представляет собой совокупность существенных признаков предмета, то определить понятие означает раскрыть его существенные признаки.

Из всех возможных видов определений существенными для нашего рассмотрения являются два принципиально отличающихся вида – это номинальные и реальные определения.

Номинальным называется определение, посредством которого взамен описания какого-либо предмета вводится новый термин, объясняется значение термина, его происхождения и т.п. Эти определения отвечают на вопрос, что обозначает то или иное слово. Например: «Флорой называется видовой состав растений, произрастающих на той или иной территории.

Реальным называется определение, раскрывающее существенные признаки предмета. Например: «Трапеция – четырёхугольник, у которого две стороны параллельны, а две другие не параллельны.

В научной практике реальные определения задают классы объектов, а номинальный вид определений служит редукцией в области рассуждений.

Определяя понятие «планета» мы задаем класс космических объектов, удовлетворяющих содержанию этого понятия. Этому понятию будут соответствовать вполне определенное количество космических объектов, относительно которых утверждения типа «х есть планета» есть утверждение, выражающее вполне определенную мысль и которое либо истинно, либо ложно. Содержание этого понятия можно не существенно изменить, в результате чего объем этого понятия может быть либо шире, либо уже. Однако, при всем этом, остается проблематичным понять какую мысль выражает сам термин «планета». Таким же образом можно определить понятие «функция» и тем самым задать класс функций, выделить в этом классе класс непрерывных функций и т.д. и рассматривать принадлежность математических объектов этому классу.

Второй тип определений определяет новые термины языка через базовые термины. К примеру, в арифметике определимо понятие «больше», которое само по себе не задает класса объектов, а является отношением между числами натурального ряда. Так, например, формула $x > y$ говорит лишь о том, что какое-то число x может быть больше, а может и не больше какого-то числа y. Утверждение же $7 > 5$ выражает уже вполне определенную мысль, которое может быть либо истинным либо ложным, в зависимости от интерпретации цифр. Подобным же образом выглядит ситуация с понятием «брат», которое выражает отношение на классе людей.

К основными правилами дефиниции, позволяющим избегать ошибок при определении понятия, относится одно существенное требование, а именно – определение не должно содержать круга, когда дефиниция определяется через дефидент, а дефидент был определён через дефиницию. Пример ошибки: «Халатность заключается в том, что человек халатно относится к своим обязанностям. Считать же *понятие,* являющееся результатом мышления, *мыслью* как раз и приводит к такого рода ошибке. Сам термин «мысль» является классифицирующим языковые выражения понятием и если «понятие» является мыслью, то мы попадает в порочный круг – «мысль» есть «мысль».

Упрек, состоящий в том, что «…говоря об объективном характере мыслей, Фреге вынужден утверждать их независимое существование, что вряд ли философски оправдано», представляется сомнительным, так философски оправданной можно считать лишь ту позицию, которая философски обоснована, а философское обоснование позиции основывается на безупречном в логическом отношении анализе, в отсутствии которого вряд ли Фреге можно обвинить.

И последнее. Вследствие рассмотренного выше понятия представления о предметах и явлениях внешнего и внутреннего мира следует, что утверждения о субъективных явлениях выражением мысли не являются. Так, например, утверждение «У меня болит голова», являясь информационным утверждением, мысль не выражает, так как невозможно говорить о содержании этого утверждения (ощущение боли у разных людей различно), а также установление истинности или ложности этого утверждения невозможно. Даже если возможны некие измерения медицинских параметров, при которых такая боль возможна, у одних индивидов головная боль ощущается – у других же нет. Аналогичная аргументация применима и к другим актам субъективной деятельности, например, утверждениям типа «Я чувствую, что…», так как невозможно установить истинность или ложность подобного утверждении, а так же невозможность установления содержания такого утверждения.

6. О знании.

Из всего выше сказанного следует, что утверждения, выражающие мысль, являются, согласно кантовской классификации, синтетическими апостериорными утверждениями. Кроме этого, понятие «мысль» существенным образом существенным образом влияет на понимание того, что является «знанием». Все вышесказанное приводит нас к

формулировке, что *знание в какой-либо области познания есть множество утверждений, выражающих мысль*. Ложное утверждение в таком определении также является знанием, состоящем в том, что мысль, выраженная этим утверждении не верна. Это обстоятельство, как отмечалось выше, может иметь значимые в познании следствия. Наверное, в этом и состоит значимость отрицательных результатов. Принцип фальсифицируемости научных гипотез имеет к этому так же непосредственное отношение.

ЛИТЕРАТУРА

Бочаров В.А Понятие, суждение и мысль. Смирновские чтения. 3 международная коференция. – М., 2001.

Витгенштейн Л. Логико-философский трактат. Изд. Иностранной литературы. М.1958.

Фреге Г. Мысль: логическое исследование. Избранные работы. М.:Дом интеллектуальной книги. Русское феноменологическое общество, 1997.

Смирнов В.А. Логические методы анализа научного знания. - М.:Наука, 1987 с.32-66.

Черепанов Е.М. Простота как критерий убедительности доказательства. Философия науки №1(44), 2010 г., Новосибирск. С.91-101

Комментарий к статье Е. М. Черепанова: Некоторые замечания к понятию «мысль».

Рецензируемая статья состоит в близком родстве с тенденцией математической и философской мысли, известной в литературе под именем «логицизм» и восходящей к исследованиям Фреге и Рассела по обоснованию математики. Отстраненная впоследствии от магистральной лини разработки указанной проблематики (без достаточных к тому оснований), эта тенденция сохраняет и в наши дни общенаучную значимость и привлекает всё большее внимание в прикладном аспекте, ввиду ускоренного прогресса компьютерной науки. И потому интерес к логицизму в современном мире нужно признать объективно оправданным. Конспективное, но внятное изложение ряда моментов, присущих непростой логико-семиотической концепции Фреге, наряду с их развернутым обсуждением применительно к нынешней обстановке. Это обсуждение явно нацелено на уяснение непроявленных познавательных резервов логицизма, – наперекор гильбетову финитизму, не возбуждающему безусловного доверия, даже когда речь идет, об элементарной арифметике и семантических парадоксов путем ее арифметизации была изначально обречена на неуспех: кошмар противоречивости попросту сменился кошмаром неполноты. В принципе, можно было заранее ожидать чего-то подобного: «арифметизация арифметики» – звучит парадоксально. Однако к концу XX-го века назрела потребность в пересмотре дотоле незыблемых ментальных предпосылок.

Рецензируемая статья закономерно вписывается в сложившуюся ситуацию. Не углубляясь в частности, скажу лишь, что упорная (почти сакральная) приверженность догмату неограниченно продлеваемого натурального ряда сыграла злую шутку с устоявшейся математической интуицией и, видимо, послужила толчком к эффективно выразимому (и тоже подлежащему арифметизации) осознанию разницы между нескончаемым и необозримым, вычислимым и вычисленным. При этой позиции, заявлять, что логически выстроенное доказательство выражает какую-то мысль, оставаясь своего рода вычислением, представляется рискованным.Уместен вопрос: что́ если однажды обнаружится, что существование истинного, но недоказуемого предложения – всего лишь малозначащая вариация парадокса лжеца ?

Теоретически это может случиться и в рамках гильбертовой установки. Критический анализ теорем о неполноте и рефлексии наводит на подобные догадки. Похоже, что так и есть. Догмат натурального ряда открывает дорогу для диагонализации и тем самым начинает работать против себя: исподволь возникая (и уже опробованная в прикладной логике) идея «формализации неформализуемого» служит тому подтверждением. Ее креативный смысл заключается в узаконенном нарушении запретов, формально как бы сохраняющих свой запретительный статус.

Но анализировать подобные вещи: дефинициальные расширения, самоприменимость и пр. – лучше по Фреге, нежели по Геделю (чтó и делает автор рецензии). На мой взгляд, автору стоило бы уделить внимание понятию форсинга, близкому содержанию его статьи, поскольку оно представляет значительный интерес благодаря созвучию с понятием «мысли». Речь идет о корректном логическом и математическом осмыслении бытовых представлений касательно отличия истинных суждений от общепринятых мнений.

Статья хорошо изложена, так что я полностью поддерживаю ее публикацию.

д.ф.-м.н., с.н.с. ИМ СО РАН Н. В. Белякин

Кулемзин В. М. — К вопросу о методике этнографических исследований (по материалам экспедиций)

Аннотация. *Автор данной статьи говорит о чрезмерной абсолютизации экономического фактора. Именно этот фактор, по мнению ряда исследователей, имеет решающее значение при деформации традиционных культур. На основании собственных материалов и полевых исследований автор показывает действие многих других важных факторов, таких как экологический, демографический, политический, мировоззренческий, психологический, а также научных знаний. В ряде случаев имеет место сложное переплетение множества факторов, и экономический далеко не всегда является равнодействующим. Он также не всегда является вектором культурных изменений. По мнению автора, наибольшей устойчивостью обладает мировоззренческий фактор, частью которого является верование или какая-либо мировая религия. Именно они играют роль ритуала, предохраняющего обычай от изменений.*

Ключевые слова: *culture, tradition, innovation, stability, variability, customs, ceremonies.*

Abstract. *The author of this article talks about the excessive absolutization of economic factor. This factor, according to some researchers, is of crucial importance in the deformation of traditional cultures. On the basis of his own data and field research, the author shows the influences of many other important factors, such as environmental, demographic, political, philosophical, psychological, and the factor of scientific knowledge. In some cases, there is a complex interplay of many factors, and the economic is not always equal in effect. It is also not always the vector cultural change. According to the author, the greatest stability has ideological factor, part of which is a belief or any world religion. They play the role of ritual, which protects custom against changes.*

С 1969 по 1988 гг. мне пришлось работать среди самых разных групп хантов – васюганских, ваховских, казымских и т. д.

Это время интенсивного промышленно-индустриального освоения края, а потому больших демографических, культурных, мировоззренческих и прочих изменений.

Многие изменения происходили и сейчас происходят столь быстро, что их фиксирует не обязательно острый глаз этнографа. Вчерашние охотники, рыболовы, оленеводы, их дети и внуки сидят за компьютером в светлых кабинетах научных и учебных учреждений.

Многие уже не помнят деталей традиционной культуры и инновацию воспринимают как традицию.

Однажды я спросил подростка, из чего делали рыболовную морду, когда не было проволоки? Он ответил, что проволока была всегда. Другой пояснил, что его отец, дед и прадед коптили рыбу в брошенном топливном баке трактора.

Вероятно, такую же ошибку делают и этнографы, когда в силу традиции все или почти все изменения традиционной культуры объясняют экономическим или фактором здравого смысла.

При более внимательном подходе обнаруживается, что экономический фактор не всегда является ведущим. Он часто бывает в тесном переплетении с другими, иногда – следствием этих «других». Какие же факторы могут вызывать деформацию традиций?

Современная методика не столь совершенна, чтобы безошибочно дифференцировать ведущие и второстепенные факторы (причины) деформации культуры. Однако некоторые из них очевидны. В частности, это:

- экологический
- мировоззренческий
- демографический
- научные знания (школьное образование)
- экзогенный (влияние чуждых культур)
- эндогенный (внутреннее развитие культуры)
- политика государства
- энергоемкость (использование физической силы)

Рассмотрим некоторые показательные примеры.

В охотничьем стойбище «Озерное», расположенном на берегу озера «Тух-Эмтор» издавна существует обычай рыболовный запор делать с таким расчетом, чтобы сквозь него проходила рыбья молодь. В 1970 г. этот запрет был нарушен и несколько центнеров мелкой рыбы стали добычей Петра Милимова. Он ее собирался сдать на звероферму как корм песцам. Свои действия Петр объяснил тем, что ему нужны деньги для приобретения телевизора, бензопилы и прочего привозного.

Другой охотник, самовольно продлив срок охоты на соболя, продал соболей русским скупщикам пушнины. Кстати, несколькими годами позднее он устроился на буровую установку, подсчитав, что это экономически более выгодно, поскольку зарплата идет круглогодично.

В этих же юртах, в это же время существовал строгий запрет на всякое вмешательство в природу, в том числе охоту и рыбную ловлю вокруг озера Весэмтор по-русски Мамонтово озеро. Дело в том, что со дна этого озера всплывали огромные куски торфа. Считалось, что их выбрасывает наверх бивнями мамонт, обитающий в воде. Это озеро и прилегающая к нему местность являлись как бы заповедником: здесь не было никаких нарушений.

С берега можно было рассматривать рыб, гусей, лебедей, а в хвойном прибрежном лесу порхали непуганые рябчики и глухари.

Экономический фактор здесь уступал мировоззренческому, силе традиций. На всех реках без исключения, включая самые северные, в конце 1960 – нач. 1970-х гг. появились подвесные лодочные моторы, а через год-два металлические лодки. Пока их не было, или они являлись дефицитом, ханты устанавливали моторы на корме долбленых лодок, для чего последние делали шире и длиннее, чем обычные весельные. Так, мускульная энергия уступила место силе мотора. Снегоход «Буран» на Югане, Вахе, Казыме вытеснил оленей, которые прежде тянули нарты. Казалось бы, все понятно, но понятно далеко не все. Например, применение любых механических средств, новых технологий при копании мерзлой земли для могилы умершего недопустимо: нельзя нарушать покой умерших. Энергоемкость здесь уступает традиционному мировоззрению.

Ханты, особенно пожилого возраста, всякий раз удивлялись, когда их знакомили с новым постановлением, запрещающим вылов рыбы ценных пород (осетр, стерлядь, нельма). Дело в том, что традиционная пища включала как раз эти породы, но никак не щуку, окуня, язя, налима. Из многочисленных объяснений, возмущений понятно было одно: закон выполняет регулирующую функцию только тогда, когда он является продолжением, юридическим оформлением традиции. То же самое можно сказать относительно квот на отстрел лося. Из сухожилий этого животного делали нитки, из рогов – клей, из шкуры, снятой с передней части передних ног – обшивку для лыж. Квота на одного лося для семьи не покрывала потребностей. Нарушения наказывались уголовно.

Мне приходилось быть неоднократно свидетелем того, как дети убегали из школы-интерната пос. Корлики на Вахе к своим родителям на стойбище. С большим трудом их собирали и на вертолете отправляли обратно. Свой поступок они объясняли тем, что надо бросать любую работу, если появляется много белки. В то же время дети старших классов или окончившие школу убеждали шаманов-провидцев, предсказателей, лекарей в наличии безграничного космического пространства. Шаманы так же тщетно доказывали существование бога-творца и что некоторым из них удавалось подняться до места обитания бога и увидеть золотую юрту, подвешенную на золотых цепях к Полярной звезде.

В конце 1980-х гг. обнаружилась вещь, которая не являлась неожиданностью. Дети, закончившие или обучавшиеся в школе, и дети, проводившие все время на стойбище, – это были разные люди. Первые не могли решать задачи, которые ставила перед ними жизнь, то есть воспроизводить традиционную культуру. Вторые не имели возможности поступать в средние и высшие учебные заведения. Однако эта сложность не являлась главной, непреодолимой. Главной сложностью, к которой общество не было готово, являлось следующее. В результате больших демографических, культурных, мировоззренческих изменений опыт, приобретенный родителями, оказался невостребованным. Сложилась так называемая патовая ситуация, когда трансляция культуры не есть сознательное творчество, в тоже время она не есть бессознательное творчество. Ситуация менялась столь быстро, что события развивались быстрее, чем одно поколение сменяется другим. Мне лично приходилось многократно быть свидетелем того, как дети обучают своих родителей: «Не доверяйте дом, детей, деньги, меха незнакомым людям, не оставляйте детей без присмотра, не отправляйте детей в лес без взрослых». Ясно, что дети обменивались новостями в школе и приносили эти новости домой.

Часто экономическим фактором объясняют смену традиционного типа жилища на современный, благоустроенный. При этом упускают из вида то обстоятельство, что радикально изменилась экологическая ситуация: проведены дороги, нефте и газопроводы, вырублены леса, уничтожены ягодники, перегорожены реки, сокращены оленные пастбища. Лесоперерабатывающие предприятия поставляют брус как основной строительный материал. В связи с отдаленностью доступного бытового строительного материала и установленного местной администрацией порядка строить типовые дома население практически не имеет иного выхода, как подчиниться местной администрации. Надо иметь в виду еще одно очень важное обстоятельство. По моим наблюдениям, ханты, селькупы, ненцы русскую культуры в сравнении со своей рассматривают как престижную. Русские же, напротив, только лишь охотничий опыт, умение ориентироваться в пространстве рассматривают как для себя недосягаемый. Во всем остальном традиционный образ жизни аборигенов не достоин внимания, подражания, изучения, описания. Как это видно из всей обширной научной, художественной и научно-популярной литературы, разница между двумя типами культур слишком велика, чтобы говорить о каком-то компромиссном, обобщающем типе культуры. В своем время Н. М. Ядринцев и Г. Н. Потанин, по моему мнению, явно преувеличили моменты сходства.

Совершенно неслучайно разного рода движения в защиту национальных культур начинаются именно с противопоставления и требований экологического порядка.

Ю.П. Холюшкин, Е.Е. Витяев, В.В. Мартынович

Web-система создания, использования и применения стратегий решения задач археологии

Монография в журнале

Информационные технологии в гуманитарных исследованиях

Выпуск 20

Подписано в печать Формат 60x84/8
Уч-изд.л. 14,4 Усл. печ.л. 13,4 Тираж 100 экз. Заказ №

Редакционно-издательский центр НГУ.
630090, Новосибирск-90, ул. Пирогова 2.