Section 2. Data Mining and Knowledge Discovery

2.1. Actual Problems of Data Mining

АВТОМАТИЗАЦИЯ ПРОЦЕССОВ ПОСТРОЕНИЯ ОНТОЛОГИЙ

Николай Г. Загоруйко, Владимир Д. Гусев, Александр В. Завертайлов, Сергей П. Ковалёв, Андрей М. Налётов, Наталия В. Саломатина

Аннотация: *Описывается проект инструментальной системы "OntoGRID" для автоматизации построения онтологий предметных областей с использованием GRID–технологий и анализа текстов на естественном языке. Рассматривается содержание и текущее состояние разрабатываемых блоков системы "OntoGRID".*

Ключевые слова: *онтология, лингвистический процессор, пирамидальные Q-сети, GRID сети.*

Введение

Онтологией (О) называется краткое описание структуры предметной области (ПрО), которое включает в себя термины (Т), обозначающие объекты и понятия ПрО, отношения (R) между терминами и определения (D) этих понятий и отношений:

$$O=<T,R,D>.$$

В графическом представлении онтология имеет вид сети, вершины которой обозначены терминами и отношениями ПрО, а ребра указывают на связи между ними. Начальная вершина, которая содержит название ПрО, связана отношением «целое-часть» с вершинами следующего уровня, которые представляют собой базовые категории данной ПрО. Каждая категория связана с вершинами следующего уровня (понятиями) своими отношениями и т.д. Вершины сети могут быть связаны с соответствующими разделами метаинформации, содержащими указание на литературные источники. Построенная онтология предметной области будет полезна для совершенствования следующих областей деятельности:

1. Системы обучения. Действительно, для первого знакомства с предметной областью было бы очень полезно иметь в качестве «опорного сигнала» легко воспринимаемую структуру этой области. С помощью онтологии можно быстро находить ссылки на источники информации.
2. Поисковые системы. Наметившийся сейчас переход от поиска информации по ключевым словам к использованию семантически значимых фрагментов текстов существенно облегчается, если используется онтология ПрО.
3. Научные исследования. Большое значение имеет унификация терминологии ПрО. Наличие онтологии ПрО позволит автоматизировать процесс отслеживания полезных данных и знаний в потоке текущей информации.
4. Системный анализ предметной области. Онтология предоставляет структурированную и частично формализованную основу для проведения системного анализа предметной области.
5. Интегрирование данных и знаний. При объединении информационных баз онтология будет помогать устанавливать семантическую эквивалентность одинаковых фактов и понятий, сформулированных в разных терминах.

Почти все известные разработки инструментов для построения онтологий [Ontology] ориентированы на то, что источником знаний, которые нужно отобразить в онтологии, является эксперт в данной прикладной области, которого нужно лишь освободить от программистской работы. Между тем, как в процессе разработки, так и в ходе эксплуатации онтологии нужно постоянно отслеживать новые знания, которые появляются в информационных сетях обычно в виде текстов на естественном языке. Отсюда вытекает необходимость оснастить инструментальную систему лингвистическим процессором.

Онтология только тогда будет принята научным сообществом, если в ее разработке участвовали широкие коллективы экспертов данной ПрО. Это требует создания поддержки коллективной деятельности экспертных групп, географически удаленных друг от друга. Удобной технологической средой для реализации такого инструмента является GRID-сеть. В свете сказанного данный проект нацелен на создание системы автоматизации построения и развития онтологий предметных областей (системы OntoGRID), которая оснащена лингвистическим процессором, работающем с русскими и английскими текстами, и реализована при помощи GRID-технологии. Ниже описываются отдельные блоки разрабатываемой системы OntoGRID.

Создание лингвистической базы знаний

Любые работы, связанные с автоматическим анализом текстов, требуют определенного набора лингвистических и алгоритмических ресурсов, основу которых составляют машинные словари (толковые, словообразовательные и другие) и программы морфологического и локального синтаксического анализа, выделения терминологической лексики и т.д.

В настоящий момент нами разработаны и реализованы: морфологическая база русского языка; блок морфологического анализа; блок статистического анализа текстов; программа выделения устойчивых словосочетаний в тексте с учетом их морфологической и комбинаторной изменчивости; программа выявления аномалий в позиционном распределении лексических единиц по тексту.

Базой для процедуры морфологического анализа служит электронный словарь Д. Уорта, содержащий свыше 100 тыс. канонических форм [Уорт,1970]. Процедуру индексации (по Зализняку) для большей части словаря удалось автоматизировать, для чего было составлено порядка 200 правил. Полученная таким образом ***морфологическая база*** содержит 3,2 млн. словоформ с соответствующими значениями грамматических категорий рода, числа, падежа, времени, лица и т.п.

Основу статистического анализа текстов составляет процедура вычисления L – граммных характеристик текста. Термин L – грамма здесь означает цепочку из L подряд следующих слов текста. Частотной характеристикой порядка L текста T будем называть совокупность всевозможных представленных в нем L–грамм с указанием частот их встречаемости $\Phi_L(T)$. Совокупность частотных характеристик $\Phi(T) = \{\Phi_1(T), \Phi_2(T), ..., \Phi_{L_{\max}}(T)\}$, будем называть полным частотным спектром текста T. Совокупность совместных частотных характеристик со значениями L от 1 до $L_{\max}(\overline{T})$ образует совместный частотный спектр группы текстов $\overline{T}$. Здесь $L_{\max}(\overline{T})$ – длина максимальной цепочки слов, представленной, как минимум, в паре текстов из $\overline{T}$. Совместные частотные характеристики служат основой для вычисления различных теоретико-множественных мер близости для пар и групп текстов [Гусев, 1983].

Важную роль при анализе текстов играют устойчивые словосочетания [Белоногов, 2002]. В основе предложенного нами алгоритма выделения словосочетаний лежит последовательное вычисление частотных характеристик ($L = 2,3,..., L_{\max}$) и фильтрация повторяющихся L – грамм в соответствии с критерием устойчивости [Гусев, 2004]. Анализ комбинаторной вариативности выделенных «устойчивых» цепочек нацелен на выявление «*устойчивых конструкций*» типа образцов (или шаблонов): «не только *X*, но и *Y*», «целью … является», «особенность … заключается …».

Существенное значение при выявлении «ключевой лексики» играет информация о распределении слова по длине текста [Пащенко, 1083]. Нами предложен новый метод выявления в тексте сверхфразовых единств, образуемых сгущениями лексических единиц определенного типа [Гусев,2002].

Построение семантических сетей текстовых документов

Создание систем анализа текстов (САТ) в интересах построения онтологий включает в себя следующие задачи: выбор типа семантической сети для представления смысла текста; формирование лингвистической базы и начального объема экспертных знаний о ПрО; разработку механизмов использования семантических сетей для построения онтологии и анализа текстов в данной ПрО; создание интерфейса, обеспечивающего взаимодействие эксперта и САТ.

В качестве формализма для представления смысла текста удобно использовать семантические сети, которые должны удовлетворять требованиям однородности, иерархичности, функциональности, полноты и прозрачности. Нами разработан формализм, удовлетворяющий всем предъявленным требованиям. При этом использовались результаты работ В.П. Гладуна и И.П. Кузнецова. В.П. Гладуном [Гладун, 1987] был разработан аппарат построения растущих пирамидальных сетей (ПС). Пирамидальной называется сеть, не содержащая вершин с одним входным ребром. Пирамидой В называется вершина b и все те вершины, из которых существуют пути в эту вершину b. При построении сети в ней образуются вершины, пирамиды которых соответствуют отдельным объектам или общим частям нескольких объектов. Важным достоинством ПС является то, что в них реализованы процессы формирования понятий.

В семантическом представлении текстов, предложенном И.П. Кузнецовым [Кузнецов,1986] вершины сетей могут соответствовать объектам, понятиям, отношениям, логическим составляющим информации, комплексным объектам и др. Кроме того, вводятся вершины другого типа – вершины связи. Они соединяются помеченными ребрами с вершинами, упомянутыми выше. В результате образуется элементарный фрагмент (F), соответствующий объектам, связанным определенными отношениями. Для представления ситуаций, состоящих из множеств объектов и отношений, используются множества фрагментов, образующие семантическую сеть, которая записывается в виде $F=F_1 \circ \ldots \circ F_n$.

В разработанной нами Q-сети объединение достоинств обоих подходов удовлетворяет сформулированным выше требованиям [Загоруйко,2004]. Структура текста в Q-сети отображается в иерархическую структуру фрагментов, каждый из которых представляет некоторую семантическую цельность.

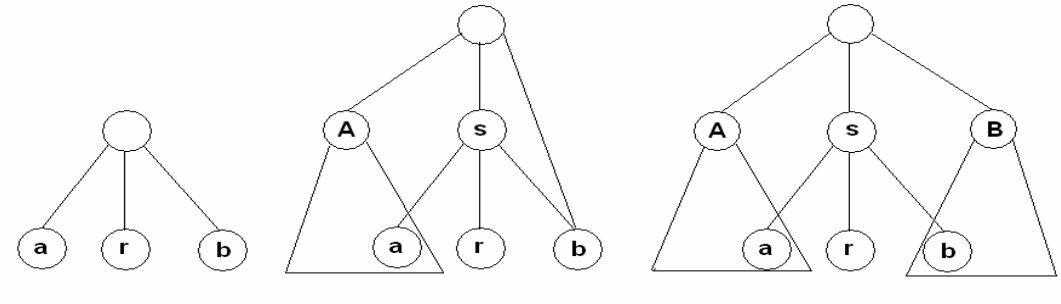
Пусть *D* – словарь ПрО, а *P* – набор отношений, реализации которых мы собираемся искать в текстах. $P = R_1 \cup R_2$, где R_1 – множество отношений с числом аргументов равным 2 (их реализациями являются словосочетания из двух значимых слов), R_2 – множество отношений с числом аргументов >2 (их реализации состоят более чем из двух слов).

По способу образования фрагменты Q-сети делятся на четыре типа:

1) **<_, r, _, a, b>≡a⊕rb** – словосочетание из двух значимых слов a,b∈D, связанных отношением r (например, a⊕rb= (анализ данных)).
2) **<_, r, s, A, b>≡Aa⊕rb** – расширение фрагмента A за счет присоединения знаменательного слова b через связь s=a⊕rb, где a∈D (например, Aa⊕rb=(интеллектуальный (анализ данных)), где A=(анализ данных), s=(интеллектуальный анализ)).
3) **<_, r, s, A, B>≡Aa⊕rbB** – объединение двух фрагментов A и B через связь a⊕rb, где a∈D, b∈D (например, Aa⊕rbB=((процесс таксономии) начинается) с (нормировки признаков)), где A=((процесс таксономии) начинается), B=(нормировка признаков), s=(начинается с нормировки)).
4) **<d, r, _, a1, …, an>** - фрагмент, соответствующий отношению r∈R2, $a_1,\ldots, a_n$ – аргументы этого отношения, *d* – имя фрагмента. Например, если *r*- родовидовое отношение, a_1=(задача интеллектуального анализа данных), a_2=(задача таксономии), a_3=(задача распознавания образов), то фрагмент <_, *r*, _, a_1, a_2, a_3> будет означать, что задачи таксономии и распознавания образов являются задачами интеллектуального анализа данных.

При анализе очередного предложения вначале выделяются (если они есть) фрагменты 4-го типа. В оставшейся части предложения выполняются следующие действия:

а) Образование фрагментов 1-го типа путем выбора словосочетаний вида a⊕rb.
б) Образование фрагментов 2-го типа Aa⊕rb, где A- фрагмент из разобранной части предложения, b- знаменательное слово из оставшейся части предложения.
в) Образование фрагментов 3-го типа Aa⊕rbB, где A, B- фрагменты из разобранной части предложения.



$a \oplus_r b$ $Aa \oplus_r b$ $Aa \oplus_r bB$

Рис. 1. Фрагменты 1-го, 2-го и 3-го типов.

В Q-сети реализованы механизмы, позволяющие описывать классы объектов в терминах как разделяющих, так и объединяющих признаков (критичных фрагменты сети). Для удобства работы, базу критичных фрагментов целесообразно хранить в виде пирамидальной сети, ассоциативные свойства которой сведут к минимуму затраты на операции поиска в ней.

САТ и база критичных фрагментов могут использоваться и для поддержки существующей онтологии. Соотнесение потока семантических портретов новых текстов с базой значимых фрагментов осуществляет наполнение элементов онтологии ссылками на текстовые документы. По степени «наполнения» эксперт может принимать решение о разделении «перегруженных» элементов сети и объединении «недогруженных». Вся лингвистическая база знаний (ЛБЗ) делится на список терминов ПрО, базу реализаций отношений (БРО) и набор правил выделения отношений в тексте. Список терминов содержит как однословные, так и многословные термины. Если многословный термин представляет собой наименование цельного понятия – в сети ему соответствует один рецептор.

После наполнения БРО производится формирование определений отношений. Под этим понимается обобщенное правило выделения отношения, не зависящее от конкретных лексем. Эти правила формулируются в терминах логических выражений от параметров, входящих в описания отношений, накопленных в БРО. При этом используется алгоритм Гладуна формирования понятий в пирамидальных семантических сетях. Предобработка текста включает в себя:

1) графематический анализ – разбиение текста на абзацы, предложения, слова.
2) морфоанализ (приписывание словоформам морфологической информации) и лемматизация (приведение текстовых форм слова к каноническим). В системе САТ используются морфологические базы для русского и английского языков [Саломатина, 2001, Сокирко, 2004]. Для учета таких отношений как синонимия, гиперонимия/гипонимия (родовидовые) и т.д. предусматривается выход на тезаурусы WordNet, RussNet.

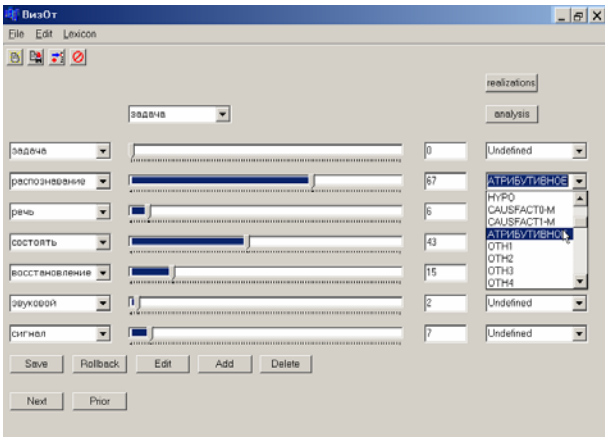


Рис. 2. Редактирование отношений

Для автоматизации построения базы реализаций семантических отношений, а также для построения Q-сетей по текстам нами разработана программа **«Визуализатор отношений» (ВизОт),** ориентированная на поддержку работы эксперта [Загоруйко,1999]**.** При загрузке текста, ВизОт проводит его нормализацию и предоставляет эксперту набор обнаруженных в тексте лексем. В верхнем окне на экране дисплея (см. рис. 2) эксперту предъявляется некоторая лексема s0, а в серии окон слева показывается ряд других лексем s_1, s_2,…, s_i,…, s_n. Эксперт может указать, какому отношению соответствует сочетание двух лексем (s_0, s_i), и с какой вероятностью эти две лексемы, встретившись совместно в тексте данной ПрО, реализуют данное отношение. Вероятность $\rho(s_0, s_i)$ указывается положением курсора на отрезке 0-100 и числовым значением (%) в окнах центральной части экрана, а имя отношения выбирается из списка в правом окне на той же строке, где стоит лексема s_i.

В ВизОт реализован алгоритм семантического анализ текста с использованием БРО и построение Q-сети текста по результатам этого анализа.

Автоматизации процессов создания и развития онтологии в GRID-сети

Структура системы автоматизированного построения онтологий «OntoGRID» должна отражать специфику трех типов ее клиентов: Эксперт, Пользователь и Администратор [Завертайлов,2004]. С точки зрения представления, создаваемая онтология - это комплект документов определённой структуры. Процесс построения онтологии состоит из итераций по дополнению и изменению этого комплекта документов. По результатам проведения ряда итераций администратор принимает решение о завершении очередного этапа процесса построения онтологии и публикации ее очередной стабильной версии.

Система, поддерживающая автоматизированное создание и обработку документов онтологии в распределенном режиме, характеризуется набором специфических требований, определяющих ее технологическую организацию. Эти требования определяются тем, что в создании онтологии участвуют многочисленные географически разделенные коллективы экспертов. Топология задействованных узлов сети разработчиков меняется по мере подключения новых коллективов или прекращения работы старых. Адекватную основу для построения систем, удовлетворяющих таким требованиям, предоставляют вычислительные технологии, известные под общим названием GRID [GRID,2003]. Из них наиболее развитый инструментарий предлагается консорциумом The Globus Alliance. К числу последних разработок этого консорциума относится архитектура OGSA (Open Grid Services Architecture), основанная на концепции веб-сервисов.

При разработке представления структуры онтологии были рассмотрены различные существующие на сегодняшний день подходы и стандарты. Как наиболее обоснованный и перспективный был принят стандарт **OWL** (Ontology Web Language) [Smith,2004], разработанный и рекомендованный консорциумом W3C. OWL обладает большей выразительной силой, чем такие структурные языки как XML, RDF и RDF-S, и может быть представлен в их форме. OWL-документ позволяет, используя лежащую в основе OWL дескриптивную логику, выводить такие факты о сущностях предметной области, которые не содержатся непосредственно в этом документе. В нашем проекте используется представление онтологии в нотации OWL-RDF.

Для упрощения разработки новых онтологий удобно создавать шаблоны онтологий различных групп предметных областей. Данный проект ориентирован на построение шаблона онтологий научно-технических предметных областей, связанных с процессами анализа, синтеза и преобразования информации о произвольных фрагментах реального мира. К числу таких процессов относятся измерение и накопление данных, обнаружение закономерностей (знаний), хранение, обработка и передача данных и знаний, использование знаний для прогнозирования и синтеза [Galunov, 2004]. На рис. 3 приведен перечень базовых категорий онтологий проблемных областей такого рода.

В дополнение к основному содержанию онтологии возможно формирование и хранение метаинформации об её элементах. Содержание метаинформации определяется в соответствии со стандартами Dublin (Guidelines for Implementing Dublin Core in XML) [Powell,2003], которые обеспечивают элементы онтологии такими данными, как реквизиты автора и соавторов, время создания и публикации, источники информации и т.д.

В качестве индивидуального средства работы эксперта с фрагментом онтологии планируется использовать редактор, разработанный группой Protege Project [Protégé,2003]. Это средство обеспечивает удобный визуальный контроль за процессом разработки фрагментов онтологии.

Описанная структура представления информации и архитектура ключевых сервисов были успешно апробированы в ходе создания **прототипа.** В настоящее время ведутся работы по дальнейшей реализации системы OntoGRID.

Рис. 3. Базовые категории онтологии

Заключение

Параллельно с описанными выше исследованиями по созданию инструментальной системы OntoGRID ведется подготовительная работа по организации виртуального коллектива экспертов из различных исследовательских центров, занимающихся проблемой «Интеллектуальный Анализ Данных» (Data Mining) для совместной разработки онтологии этой предметной области. С этой целью создается общедоступный двуязычный сайт, на котором будут помещены описания концепции и первого варианта предлагаемой онтологии. В настоящее время такой сайт (на русском и английском языках) создается на сервере Института Математики СО РАН.

Авторы выражают благодарность Борисовой И.А., Дюбанову В.В., Кутненко О.А., Соколовой А.П. и Чуриковой В.А. за активное и полезное участие в обсуждении вопросов, затронутых в этой статье.

Библиография

[Ontology] http://xml.com/2002/11/06/Ontology_Editor_Survey.html

[Dean,1070] Dean S. Worth, Andrew S. Kozak, Donald B. Johnson. Russian Derivational Dictionary. American Elsevier Publishing Company, Inc. New York, 1970.

[Гусев,1083] Гусев В.Д. Механизмы обнаружения структурных закономерностей в символьных последовательностях // Проблемы обработки информации. – Новосибирск, 1983. – Вып. 100: Вычислительные системы. – С. 47 – 66.

[Белоногов,2002] Белоногов Г.Г., Быстров И.И., Новоселов А.П. и др. Автоматический концептуальный анализ текстов // НТИ, сер. 2. – № 10, 2002. – С. 26 – 32.

[Гусев,2004] Гусев В.Д., Саломатина Н.В. Алгоритм выявления устойчивых словосочетаний с учетом их вариативности (морфологической и комбинаторной) // Труды международной конференции Диалог – 2004 "Компьютерная лингвистика и интеллектуальные технологии", Верхневолжский, 2 – 7 июня 2004. – М., Наука, С. 530 – 535.

[Пащенко,1983] Пащенко Н.А., Кнорина Л.В., Молчанова Т.В. и др. Проблемы автоматизации индексирования и реферирования // Итоги науки и техники. Информатика – Т. 7, 1983. – С. 7 – 165.

[Гусев,2002] Гусев В.Д., Немытикова Л.А., Саломатина Н.В. Выявление аномалий в распределении слов или связных цепочек символов по длине текста // Интеллектуальный анализ данных. – Новосибирск, 2002. – Вып. 171: Вычислительные системы. – С. 51 – 74.

[Гладун,1987] Гладун В.П. Планирование решений. Киев. Наукова думка, 1987. С.17-51.

[Кузнецов,1986] Кузнецов И.П. Семантические представления. Изд. «Наука», М. 1986 г.

[Загоруйко,2004] Загоруйко Н.Г., Налетов А.М., Соколова А.А., Чурикова В.А.. Формирование базы лексических функций и других отношений для онтологии предметной области //Труды конференции Диалог-2004. С.202-204.

[Саломатина,2001] Саломатина Н.В. Количественные характеристики вариативности морфемных моделей (на материале словаря канонических форм русского языка) // Методы обнаружения эмпирических закономерностей . – Новосибирск, 2001. – Вып.167: Вычислительные системы. – С. 93 – 114.

[Сокирко,2004] Сокирко А.В. Морфологические модули на сайте www.aot.ru //Труды конференции Диалог-2004. С.559-564.

[Загоруйко,1999] Загоруйко Н.Г. Метрологические свойства эксперта.// Обнаружение эмпирических закономерностей: Вычислительные системы, вып. 166, Тр. ИМ СО РАН, Новосибирск, 1999. С.119-128.

[Завертайлов,2004] Завертайлов А.В., Ковалев С.П. Система поддержки деятельности распределенных экспертных групп по разработке онтологий предметных областей // Труды Международной конференции по вычислительной математике <МКВМ-2004>. Рабочие совещания. Новосибирск: ИВМиМГ СО РАН, июнь 2004. С. 56-65.

[GRID,2003] Grid Computing: Making the Global Infrastructure a Reality. N. Y.: Wiley & Sons, 2003.

[Smith,2004] Smith M.K., Welty C., McGuinness D.L. OWL Guide. W3 Consortium, 2004. http://www.w3.org/TR/owl-guide/.

[Galunov, 2004] Valery I. Galunov, Boris M. Lobanov and Nikolay G. Zagoruiko. Ontology of the subject domain “Speech signals recognition and synthesis”// Proc. of 9-th International Conference “Speech and Computer” (SPEECOM’2004), Saint-Peresburg, September 2004, p.448-454.

[Powell,2003] Powell A., Johnston P. Guidelines for implementing Dublin Core in XML. DCMI, 2003. http://dublincore.org/documents/dc-xml-guidelines/.

[Protégé,2003] Protege Project. http://protege.stanford.edu.

Информация об авторах

Николай Г. Загоруйко – Институт Математики СО РАН, пр. Коптюга, 4, Новосибирск, 630090, Россия; e-mail: zag@math.nsc.ru

Владимир Д. Гусев – Институт Математики СО РАН, пр. Коптюга, 4, Новосибирск, 630090, Россия; e-mail: gusev@math.nsc.ru

Александр В. Завертайлов – Новосибирский Государственный Университет, ул. Пирогова, 2, 630090, Россия; e-mail: alzavmail@mail.ru

Сергей П. Ковалёв – Новосибирский Государственный Университет, ул. Пирогова, 2, 630090, Россия; e-mail: kovalyov@ccfit.nsu.ru

Андрей М. Налётов – Институт Математики СО РАН, пр. Коптюга, 4, Новосибирск, 630090, Россия; e-mail: naletov@ngs.ru

Наталия В. Саломатина – Институт Математики СО РАН, пр. Коптюга, 4,Новосибирск, 630090, Россия; e-mail: nataly@math.nsc.ru

APPLICATION OF THE MULTIVARIATE PREDICTION METHOD TO TIME SERIES [1]

Tatyana Stupina, Gennady Lbov

***Abstract**: An approach to solving the problem of heterogeneous multivariate time series analysis with respect to the sample size is considered in this paper. The criterion of prediction multivariate heterogeneous variable is used in this approach. For the fixed complexities of probability distribution and logical decision function class the properties of this criterion are presented.*

***Keywords**: the prediction of multivariate heterogeneous variable, multivariate time series, the complexity of distribution.*

Introduction

Let certain object (process) is described by the set of random features $X = X_1,...,X_n$, changing on time. On the base of analysis information, that presents features measurements in the consequent moments time series (prehistory), it is necessary to predict a values of features set $Y = Y_1,...,Y_m$ at certain future time moment (in particular, $Y \subseteq X$. Distinguishing feature of considered below prediction problems is the measured features heterogeneity: the variable set be able consist of binary, nominal and quantitative variables simultaneously. In this case, multivariate time series presents itself a set of binary, symbol and numeric random sequences. Classical methods are directed to the analysis of numeric sequences basically. Many methods allow analyse univariate binary or symbol sequences. However the most of important applied problems number are concerned with need to heterogeneous time series analyse. There is reason to suppose in some problems that time series is the realization of random processes, in which probabilistic characteristics (distribution) are saved on a time. At other times such suggestions to do it is impossible under the matter of problem (probabilistic characteristics of process are changed on time). There is possible to offer a different depending on specified suggestions targets setting and the different methods of their decision accordingly. The methods of heterogeneous time series analysis for different targets setting, including the logical deciding functions class for heterogeneous variable are considered in work [Lbov G.S., 1994].

The Target Setting

One is considered the n – measured heterogeneity random process $G = \{X_1(t),...,X_j(t),...,X_n(t)\}$. Let it set of predictable characteristic is $Y_j = X_j$, $j = 1,...,n$. Fix some consequent moments of the time, $1 \le R \le N$. Denote the value random variable X_j at a moment of the time t_d, $x_j^d \in D_{X_j}$, as this x_j^d, and x^d is the value random variable of X, $x^d \in D_X$, $D_X = \prod_{j=1}^{n} D_{X_j}$. The problem consist of that, it is necessary to predict the values set $y = (y_1,...,y_j,...,y_n)$ at certain future moment of the time t_{R+1}, where $y_j = x_j^{R+1}$ using the data, characterizing prehistory, $b = \{x_j^d\}$, $j = 1,..,n$, $d = 1,...,R$. It is necessary to build decision function, allowing predict a set of values $y = (y_1,...,y_j,...,y_n)$ on prehistory *b*.

The set of every possible all prehistory, that have line measure R denote as B, and the set of every possible all sets y denote as D_Y, $b \in B$, $y \in D_Y$, $D_Y = \prod_{j=1}^{n} D_{Y_j}$. Let us understand a prediction decision function as a f mapping of the B set on the D_Y set, i.e. $f : B \to D_Y$. At the building decision functions f is used following

[1] This work was financially supported by RFBR-04-01-00858

hypothesis: It is supposed that conditional distribution $P(y/b)$ does not depend on the shift on the time, i.e. distribution is specified for moments of the time $t_1,...,t_R,t_{R+1}$ is contemporized with distribution for moments of the time $t_1 \pm \Delta T,...,t_R \pm \Delta T,t_{R+1} \pm \Delta T$. If the conditional distribution $P(y/b)$ is known, then it is possible to find optimum prediction decision function $f_\circ$. Since specified distribution is unknown, decision function shall be constructed on the base of multivariate time series analysis.

Let the features $X_1,...,X_j,...,X_n$ are measured at consequent moments of the time with the gap $\Delta t = t_d - t_{d-1}$ for the random process G. Denote this set of moments as $T = \{t_1,...,t_k,...,t_N\}$. Thus, the empirical information is presented by n – measured heterogeneity time series $q = \{x_j^k\}, j = 1,...,n, k = 1,...,N$. The set of values $x^{k-d} = (x_1^{k-d},...,x_j^{k-d},...,x_n^{k-d})$ will is called prehistory with the number d, correlated with a moment of the time t_k, $k = R+1,...,N$. The prehistory with line measure R for a specified moment of the time t_k is denoted as a table $b^k = \{x^{k-d}\}$, $d = 1,...,R$. Note that univariate symbol sequence for $R = 1$ is the realization of simple Markoff process with the transfer probability matrix $P(y/x)$, $x \in A$, $y \in A$, A –an alphabet of symbols. Decision function $\bar{f}$, constructed on the base of set prehistory analysis with line measure *R*, is named sample decision function of prediction.

It is necessary to construct the sample decision function on the small sample in the multivariate heterogeneous space, so the most proper class is a class of logical decision functions [Lbov G.S., Starceva N.G, 1999]. Methods of time series analysis propose to decision of problem in two stages: It is constructing decision function for fixed prehistory with the number d ($d = 1,...,R$) it is constructing the generalise logical decision function (mapping $f : B \to D_Y$). The first stage is consist of decision the prediction multivariate variable problem Y on other multivariate variable *X*, i. e. for each prehistory *d* we have two data tables $\{x^{k-d}\}$, $\{y^k\}$, $k = R+1,...,N$, on base which necessary to construct the sample decision function (mapping $D_X \to D_Y$). Below it is considered a decision of this problem, in which is used criterion, introduced in work [Lbov G.S., Stupina T.A., 2002].

The Performance Criterion of Prediction

In the probabilistic statement of the problem, the value *(x,y)* is a realization of a multidimensional random variable *(X,Y)* on a probability space $< \Omega, \mathrm{B}, \mathrm{P} >$, where $\Omega = D_X \times D_Y$ is μ-measurable set (by Lebeg), B is the borel σ-algebra of subsets of Ω, P is the probability measure (probability distribution) on B, D_X is heterogeneous domain of under review variable, $\dim D_x = n$, D_y is heterogeneous domain of objective variable, $\dim D_y = m$.

Definition 1. The strategy of nature is *c={p(x,y)=p(x)p(y/x))}*, where a conditional probability *p(y/x)* is specified for any elements on B.

Let us put $\Phi_\circ$ is a given class of decision functions. Class $\Phi_\circ$ is μ-measurable functions that puts some subset of the objective variable $E_y \subseteq D_Y$ to each value of the under review variable $x \in D_X$, i.e. $\Phi_\circ = \{f : D_x \to 2^{D_y}\}$. This class of decision function is more total than class of logical decision functions [Lbov G.S., Starceva N.G, 1999]. In this paper, we will consider criterion for decision function from total class $\Phi_\circ$. So criterion was considered for logical decision functions in work [Lbov G.S., Stupina T.A., 2002]. But here we will achieve that class of logical decision functions is a universal class about relative to criterion.

The quality $F(c,f)$ of a decision function $f \in \Phi_\circ$ under a fixed strategy of nature *c* is determined as follows.

$$F(c,f) = \int_{D_x} (P(E_y(x)/x) - \mu(E_y(x)))dP(x),$$

where $E_y(x) = f(x)$ is a value of decision functions in *x*, $P(y \in E_y(x)/x)$ is a conditional probability of event $\{y \in E_y\}$ under a fixed *x*, $\mu(E_y(x))$ is measurable of subset E_y. Note that if $\mu(E_y(x))$ is probability measure,

than criterion $F(c,f)$ is distance apart distributions. If the specified probability coincides with equal distribution than such prediction does not give no information on predicted variable (entropy is maximum). The measure $\mu(E_y(x)) = \frac{\mu(E_y)}{\mu(D_Y)} = \prod_{j=1}^{m} \frac{\mu(E_{y_j})}{\mu(D_{y_j})}$ is the normalized measure of the subset E_y and it is introduced with taking into account the type of the variable. The measure $\mu(E_y(x))$ is measure of interval, if we have a variable with ordered set of values and it is quantum of set, if we have a nominal variable (it is variable with finite non-ordering set of values). Clearly, the prediction quality is higher for those E_y whose measure is smaller (accuracy is higher) and the conditional probability $P(y \in E_y(x)/x)$ (certainty) is larger.

For a fixed strategy of nature *c*, we define an optimal decision function $f_\circ(x)$ as function for which $F(c,f_\circ) = \sup_{f\in\Phi_\circ} F(c,f)$, where $\Phi_\circ$ is represented above class of decision functions.

As a rule, the strategy of nature is unknown; for this reason, a decision function is constructed from a training sample $v = (x^i, y^i)_{i=1,...,N}$ by sampling criterion $F(\bar{f})$ with the use of some algorithm $Q(v) = \bar{f}$, where $\bar{f}(x)$ is a sampling decision function and *N* is the size of the training sample. The sampling criterion $F(\bar{f})$ is empirical risk of the criterion $F(c,f)$.

When we solve this problem in practice the size of sample is very smaller and type of variables different. In this case is used class of logical decision function. The logical decision function *f* is assigned the pair $<\alpha,\beta>$, where $\alpha \in \Psi_M$ and $\beta \in R_M$. The class Ψ_M is the set of partitions $\alpha = \{E_x^1,...,E_x^t,...,E_x^M\}$ of the space D_X into disjoint subsets for which $E_x^t = \prod_{i=1}^{n} E_{x_i}^t$, $E_{x_i}^t \subseteq D_{X_i}$, $E_{x_i}^t \neq \varnothing$ and $E_{x_i}^t \in W_{X_i}$, where W_{X_i} is the set of all possible intervals if X_i is a variable with ordered set of values and W_{X_i} is the set of arbitrary subsets of D_{X_i} if X_i is a nominal variable, i.e. a variable with a finite unordered set of values; we have $E_x^t \in W_X$, where $W_X = \prod_{i=1}^{n} W_{X_i}$.

The class R_M is the set of decisions (arbitrary subset of the space D_Y) $\beta = \{E_y^1,...,E_y^t,...,E_y^M\}$ for which $E_y^t = \prod_{i=1}^{m} E_{y_i}^t$, $E_{y_i}^t \subseteq D_{Y_i}$, $E_{y_i}^t \neq \varnothing$ and $E_{y_i}^t \neq \varnothing$, where W_{Y_i} is defined so as W_{X_i}. The decision function is presented in simple form for understanding: *if* $x \in E_x^t$ *than* $y \in E_y^t$. The subsets E_x^t and E_y^t represented as above can be described in terms of conjunctions of simple predicates. Such a coarsening of the decision function is caused by the necessity to construct solutions from small samples. The class of logical decision function Φ_M can be represented as $\Psi_M \times R_M$.

Under the assumptions made, the **complexity of the class** Φ_M is only determined by the *M* parameter: $v(\Phi_M) = M$. Thus, the larger the number *M*, the more complex the class Φ_M. We achieve important property of this class by theorem.

Theorem. For a fixed type of the predicate, the class Φ_M of logic decision functions is a universal class in the problem of prediction multivariate heterogeneous value by criterion *F(c,f)*, i.e. for any strategy of nature *c* and any $\varepsilon > 0$ there exists a number *M (M=1,2,3,…)* and for some logical decision function $f \in \Phi_M$ (it is represented in the form of decision tree on *M* vertices) such that $|F(c,f) - F(c,f_\circ)| \leq \varepsilon$, where $f_\circ$ is optimal function in class $\Phi_\circ$.

The proof of this theorem readily follows from the property of μ-measurability and P-measurability of space *D* and its projections on the space D_X, D_Y correspondingly.

The proof for the case where *Y* is a discrete variable is given in [Lbov G.S., Starceva N.G, 1994]. The proof for the case where *Y* is a continuous variable is given in [Berikov V.,1995].

We can introduce a complexity of distribution (strategy of nature *c*) using the class logical decision function. It is necessary for solving statistical stability problem of decision function.

Statement 1. For any nature strategy *c* the quality criterion $F(c,f)$ (risk function) of logical decision function *f* belonging to Φ_M is presented by following expression:

$$F(c,f)=\int\limits_{D_X}\int\limits_{D_Y}(1-L(y,f(x)))p(x,y)dxdy=\sum_{t=1}^{M}p_x^t(p_{y/x}^t-\mu^t),$$

where the loss function $L(y,f)$ such as $L(y,f)=\begin{cases}p_o, & y\in\beta\\ 1+p_o, & y\notin\beta\end{cases}$, $p_o=\mu(E_Y^t)$, $\beta=f(\alpha),\alpha\in\Psi_M$.

Proof.

$$F(c,f)=\int\limits_{D_x}(P(E_y(x)/x)-\mu(E_y(x)))dP(x)=\sum_{t=1}^{M}\left[\int\limits_{E_X^t}\int\limits_{E_Y^t}p(x,y)dxdy-p_o\int\limits_{E_X^t}p(x)dx\right]=$$

$$\sum_{t=1}^{M}\left[\int\limits_{E_X^t}\int\limits_{E_Y^t}p(x,y)dxdy+\int\limits_{E_X^t}\int\limits_{D_Y}(-p_o)p(x,y)dxdy\right]=$$

$$\sum_{t=1}^{M}\left[\int\limits_{E_X^t}\int\limits_{E_Y^t}(1-p_o)p(x,y)dxdy+\int\limits_{E_X^t}\int\limits_{D_Y}(-p_o)p(x,y)dxdy-\int\limits_{E_X^t}\int\limits_{E_Y^t}(-p_o)p(x,y)dxdy\right]=$$

$$=\sum_{t=1}^{M}\int\limits_{E_X^t}\left[\int\limits_{E_Y^t}(1-p_o)p(x,y)dxdy+\int\limits_{\bar{E}_Y^t}(-p_o)p(x,y)dxdy\right]=\int\limits_{D_X}\int\limits_{D_Y}(1-L(y,f(x)))p(x,y)dxdy.$$

Definition 2. To each subclass Φ_M we put in correspondence the subset $L_\varepsilon(M)=\{c:\exists f\in\Phi_M,\left|F(c,f)-F(c,f_\circ)\right|\le\varepsilon\}$ of nature strategies; ε is an arbitrarily small number determining an admissible error level of this subset of strategies, where $f_\circ$ is optimal function in class $\Phi_\circ$.

The complexity measure of each subset $L_\varepsilon(M)$ is defined as the complexity measure of the corresponding subclass of decision functions: $v(L_\varepsilon(M))=v(\Phi_M)=M$. Accordingly, the nature strategy *c* belonging to $L_\varepsilon(M)$ has complexity measure *M*. The important statement follows from this theorem and definition.

Statement 2. The set of all possible strategies can be ordered according to complexity, i.e. $L_\varepsilon(1)\subset L_\varepsilon(2)\subset\ldots\subset L_\varepsilon(M)\subset\ldots\subset L_\circ$, and $\varepsilon^{M+1}\le\varepsilon^M$, where $v(L_\varepsilon(M))=M$ is the complexity and ε^M is the admissible error level of the strategy class $v(L_\varepsilon(M))$.

Proof. For an arbitrary *M*, let us prove the embedding $L_\varepsilon(M)\subset L_\varepsilon(M+1)$ i.e. show that $\forall c\in L_\varepsilon(M)$, $\exists f\in\Phi_{M+1}$ such that $\left|F(c,f)-F(c,f_\circ)\right|\le\varepsilon$. The definition of the class $L_\varepsilon(M)$ implies that $\exists g\in\Phi_M$ such that $\left|F(c,g)-F(c,f_\circ)\right|\le\varepsilon^M$. Since $\Phi_M\subset\Phi_{M+1}$, we can obtain *f* from *g* by partitioning some subset E_X^t into two subsets: if $g\sim<\alpha,\beta>$, $\alpha=\{E_X^t\}_{t=1,\ldots,M}$, $\beta=\{E_Y^t\}_{t=1,\ldots,M}$ than $f\sim<\alpha',\beta'>$, $\alpha'=\{E_X^1,\ldots,E_X^{t_1},E_X^{t_2},\ldots,E_X^M/E_X^t=E_X^{t_1}\cup E_X^{t_2}\}$, $\beta'=\{E_Y^1,\ldots,E_Y^{t_1},E_Y^{t_2},\ldots,E_Y^M/E_Y^t=E_Y^{t_1}\cup E_Y^{t_2}\}$, where $\mu(E_X^t)=\mu(E_X^{t_1})+\mu(E_X^{t_2})$ and $\mu(E_Y^t)\ge\mu(E_Y^{t_1})+\mu(E_Y^{t_2})$.

Therefore, $\left|F(c,f)-F(c,f_\circ)\right|\le\varepsilon=\varepsilon^{M+1}\le\varepsilon^M$, it is followed from the definition $F(c,f)$.

We can suppose that the true (optimal) decision function belongs to Φ_M it is followed from this statement 1.

Definition 3. Define a nature strategy c_M (generated by logical decision function $f\in\Phi_M$) such as set of parameters satisfying the following conditions:

1) $\sum_{t=1}^{M}p_x^t=1$,

2) $P(E_Y^t/E_X^t)=p_{y/x}^t$ (conditional distribution is same for any $x\in E_X^t$ and $y\in E_Y^t$),

3) $P(\overline{E}_Y^t / E_X^t) = 1 - p_{y/x}^t$,

where $E_X^t \in \alpha$, $E_Y^t \in \beta$, $<\alpha,\beta> \sim f \in \Phi_M$. The complexity of this strategy is *M*, i.e. $v(c_M) = M$. Note that c_M generated by logical decision function belongs to class $L_\varepsilon(M)$. Clearly, the decision function that generated this strategy is optimal function in class Φ_M.

Statement 3. For a fixed nature strategy $c_M \in L_\varepsilon(M)$ of complexity *M* the quality criterion $F(c_M, \widetilde{f})$ (risk function) of logical decision function $\widetilde{f} \in \Phi_{M'}$ of complexity M' is presented in following form:

$$F(c_M, \widetilde{f}) = F(\widetilde{\alpha}) = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} \rho^{t'} = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} (\widetilde{p}_{y/x}^{t'} - \mu_y^{t'}),$$

where $\widetilde{p}_x^{t'} = P(x \in \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} p_x^t \dfrac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)}$,

$$\widetilde{p}_{y/x}^{t'} = \frac{1}{\widetilde{p}_x^{t'}} \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)} \left(p_{y/x}^t \frac{\mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{\mu(E_Y^t)} + (1 - p_{y/x}^t) \frac{\mu(\widetilde{E}_Y^{t'}) - \mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{1 - \mu(E_Y^t)} \right).$$

Proof. Since the decision function $\widetilde{f}$ belongs to class $\Phi_{M'}$ than there exists partition $\widetilde{\alpha} = \{\widetilde{E}_X^1, ..., \widetilde{E}_X^{t'}, ..., \widetilde{E}_X^{M'}\}$ of space D_X and according to it the set of subsets $\widetilde{\beta} = \{\widetilde{E}_Y^{t'}, ..., \widetilde{E}_Y^{t'}, ..., \widetilde{E}_Y^{M'}\}$ of space D_Y. The expression of the criterion $F(c, \widetilde{f}) = \sum_{t'=1}^{M'} \widetilde{p}_x^{t'} (\widetilde{p}_{y/x}^{t'} - \mu_y^{t'})$ follows from statement 1, where $\widetilde{p}_x^{t'} = P(x \in \widetilde{E}_X^{t'})$, $\widetilde{p}_{y/x}^{t'} = P(y \in \widetilde{E}_Y^{t'} / x \in \widetilde{E}_X^{t'})$. Since the strategy $c = c_M$, $c_M \in L_\varepsilon(M)$ is generated by logical decision function $f \sim <\alpha,\beta> \in \Phi_M$, there is a partition $\alpha = \{E_X^1, ..., E_X^t, ..., E_X^M\}$ of space D_Y and according to it the set of subsets $\beta = \{E_Y^1, ..., E_Y^t, ..., E_Y^M\}$ of space D_Y, the sets of parameters $p_x^t = P(E_X^t)$, $p_{y/x}^t = P(E_Y^t / E_X^t)$ as provided by definition 3. Late for simplicity we will not write the mark '$\in$' and '$\cap$' in view of the events. Express the $\widetilde{p}_x^{t'}$ and $\widetilde{p}_{y/x}^{t'}$ by way of p_x^t and $p_{y/x}^t$ take account of the event distribution is inside of subsets E_X^t, E_Y^t:

$$\widetilde{p}_x^{t'} = P(\widetilde{E}_X^{t'}) = P(\cup E_X^t \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} P(E_X^t) P(\widetilde{E}_X^{t'} / E_X^t) = \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} E_X^t)}{\mu(E_X^t)};$$

$$\widetilde{p}_{y/x}^{t'} = P(\widetilde{E}_Y^{t'} / \widetilde{E}_X^{t'}) = \frac{P(\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'})}{P(\widetilde{E}_X^{t'})} = \frac{1}{\widetilde{p}_x^{t'}} P(\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}),$$

$$P(\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = P(D\widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = P(\cup E_X^t D_Y \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} P(E_X^t D_Y \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = \sum_{t=1}^{M} (P(E_X^t E_Y^t \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) + P(E_X^t \overline{E}_Y^t \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'})),$$

$$P(E_X^t E_Y^t \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = P(E_X^t E_Y^t) P(\widetilde{E}_X^{t'} \widetilde{E}_Y^{t'} / E_X^t E_Y^t) = p_{xy}^t \frac{\mu((E_X^t E_Y^t) \cap (\widetilde{E}_X^{t'} \widetilde{E}_Y^{t'}))}{\mu(E_X^t E_Y^t)} =$$

$$= p_x^t \frac{\mu(E_X^t \widetilde{E}_X^{t'})}{\mu(E_X^t)} p_{y/x}^t \frac{\mu(E_Y^t \widetilde{E}_Y^{t'})}{\mu(E_Y^t)},$$

$$P(E_X^t \overline{E}_Y^t \widetilde{E}_Y^{t'} \widetilde{E}_X^{t'}) = P(E_X^t \overline{E}_Y^t) P(\widetilde{E}_X^{t'} \widetilde{E}_Y^{t'} / E_X^t \overline{E}_Y^t) =$$

$$= p_x^t (1 - p_{y/x}^t) \frac{\mu((E_X^t \overline{E}_Y^t) \cap (\widetilde{E}_X^{t'} \widetilde{E}_Y^{t'}))}{\mu(E_X^t \overline{E}_Y^t)} = p_x^t \frac{\mu(E_X^t \widetilde{E}_X^{t'})}{\mu(E_X^t)} (1 - p_{y/x}^t) \frac{\mu(\overline{E}_Y^t \widetilde{E}_Y^{t'})}{\mu(\overline{E}_Y^t)},$$

where $\dfrac{\mu(\overline{E}_Y^t \widetilde{E}_Y^{t'})}{\mu(\overline{E}_Y^t)} = \dfrac{\mu(\widetilde{E}_Y^t) - \mu(E_Y^t \widetilde{E}_Y^{t'})}{1 - \mu(E_Y^t)}$ and $\overline{E}_Y^t = D_Y \setminus E_Y^t$.

Remark. If the nature strategy c_M such that some subset E_Y^t coincides with the space D_Y, than

$$\widetilde{p}_{y/x}^{t'} = \frac{1}{\widetilde{p}_x^{t'}} \sum_{t=1}^{M} p_x^t \frac{\mu(\widetilde{E}_X^{t'} \cap E_X^t)}{\mu(E_X^t)} p_{y/x}^t \frac{\mu(\widetilde{E}_Y^{t'} \cap E_Y^t)}{\mu(E_Y^t)}.$$

It is followed from that $p_{y/x}^t = P(D_Y / E_X^t) = 1$, $\mu(D_Y) = 1$.

Consequence 1. If the decision function $\widetilde{f}$ belonging to Φ_M coincides with the function f belonging to Φ_M, than $F(c, \widetilde{f}) = F(c, f)$.

Consequence 2. For the decision function $\widetilde{f}$ belonging to $\Phi_{M'}$ we have the expression $P(\overline{\widetilde{E}}_Y^{t'} / \widetilde{E}_X^{t'}) = 1 - \widetilde{p}_{y/x}^{t'}$.
Really, it is follows from the statement 3, where

$$\frac{\mu(\overline{\widetilde{E}}_Y^{t'} E_Y^t)}{\mu(E_Y^t)} = \frac{\mu(E_Y^t) - \mu(E_Y^t \widetilde{E}_Y^{t'})}{\mu(E_Y^t)}, \quad \frac{\mu(\overline{\widetilde{E}}_Y^{t'} \overline{E}_Y^t)}{\mu(\overline{E}_Y^t)} = \frac{1 - \mu(E_Y^t) - \mu(\widetilde{E}_Y^{t'}) + \mu(E_Y^t \widetilde{E}_Y^{t'})}{1 - \mu(E_Y^t)}.$$

Consequence 3. If we have M = 1 and the optimal function f generating c_1 such that $E_Y^1 = D_Y$, than $F(c_1, f) = 0$.

Really, for the express of criterion we have $F(c,f) = \sum_{t=1}^{M} \left(P(E_X^t E_Y^t) - P_o(E_Y^t) \right) = P_o(D_X D_Y) - P_o(D_Y) = 0$.

It means that we have the event distribution in D for the nature strategy of the complexity M=1. It is case when the entropy is maximum.

Consequence 4. If we have M = 1 and the optimal function f generating c_1 such that $E_Y^1 = D_Y$, than for any decision function $\widetilde{f} \in \Phi_{M'}$ the criterion $F(c_1, \widetilde{f}) = 0$.

Really, $\widetilde{p}_{y/x}^{t'} = \frac{\mu(\widetilde{E}_Y^{t'} D_Y)}{\mu(D_Y)} P_o(D_Y / D_X) = \mu(\widetilde{E}_Y^{t'})$, $\widetilde{p}_x^{t'} = \frac{\mu(\widetilde{E}_Y^{t'} D_X)}{\mu(D_X)} P_o(D_X) = \mu(\widetilde{E}_X^{t'})$,

$$F(c_1, \widetilde{f}) = \sum_{t'=1}^{M'} \mu(\widetilde{E}_X^{t'})(\mu(\widetilde{E}_Y^{t'}) - \mu(\widetilde{E}_Y^{t'})) = 0.$$

Consequence 5. If the decision function $\widetilde{f}$ belongs to Φ_1 and $\widetilde{E}_Y^1 = D_Y$, than we have $F(c_M, \widetilde{f}) = 0$ for any complexity $M \geq 1$.

Really, we have $\widetilde{p}_x = \sum_{t=1}^{M} p_x^t \frac{\mu(D_X E_X^t)}{\mu(E_X^t)} = 1$, $\widetilde{p}_{y/x} = \sum_{t=1}^{M} p_x^t \left(p_{y/x}^t \frac{\mu(D_Y E_Y^t)}{\mu(E_Y^t)} + (1 - p_{y/x}^t) \frac{1 - \mu(D_Y E_Y^t)}{1 - \mu(E_Y^t)} \right) = 1$.

As stated above when the nature strategy is unknown the problem of statistical stability of sample decision functions is appeared. The quality $F(c, \bar{f})$ of sample decision function depends on the size N of the sample, the complexity M of the distributions, and the complexity M' of the class of functions $\Phi_{M'}$ used by the algorithm $Q(v)$ and empirical criterion $F(\bar{f})$ for constructing sample decision functions $\bar{f}$. The empirical criterion $F(\bar{f})$ (empirical risk function) is presented by expression:

$$F(\bar{f}) = \frac{1}{N} \sum_{i=1}^{N} (1 - L(x^i, y^i)) = \sum_{t=1}^{M'} \frac{N(\widetilde{E}_X^t)}{N} \left(\frac{N(\widetilde{E}_Y^t \widetilde{E}_X^t)}{N(\widetilde{E}_X^t)} - \mu^t \right) = \sum_{t=1}^{M} \hat{p}_x^t (\hat{p}_{y/x}^t - \hat{\mu}^t),$$

where $N(*)$ is a number of sample spots belonging to the corresponding subset *, $\hat{\mu}^t = \mu(\widetilde{E}_Y^t)$, $\bar{f} \sim < \widetilde{\alpha}, \widetilde{\beta} > \in \Phi_{M'}$.

On the one hand, if the constraints on the class of decision functions are too strong, then this class may be inadequate to the true distribution, and the higher the degree of inadequacy, then poorer the quality of the decision function. On the other hand, using a complex class of functions on small samples also lowers the quality for the decision function.

At present time there are two well-known approaches solving this problem. The Vapnik -Chervonenkis approach uses the principle of uniform convergence [Vapnik V.N., Chervonenkis A.Ya, 1970]: the quality criterion $F(c,\bar{f})$ depends on VC-complexity of the decision function class Φ and the level of empirical risk $F(\bar{f})$. In the case of one discrete variable prediction was provided results [Nedelko V.M., 2004]. When the nature strategy *c* belongs to even probability distribution class such problem was decided by the method of statistical modelling for the case of several heterogeneous variable prediction [Lbov G.S., Stupina T.A., 2003]. It is the particular case of our problem. Really, we can provide the biased estimator of criterion (risk function) $E\varepsilon_N = E_{v_N}\left|F(c,\bar{f}) - F(\bar{f})\right|$ by the statistical modelling method for any nature strategy *c* belonging to the class $L(M)$. It is follows from the consequence 1-4 that we have the expression $E\varepsilon_N = E_{v_N}F(\bar{f})$ for $c \in L(1)$.

Another (Bayesian) approach to solving this problem consists in the construction of the evaluation $EF(c,\bar{f})$ that is obtained by averaging over all samples of *N*-size. Raudys in [Raudis Sh.Yu.,1976] used that (Bayesian) approach to solving pattern recognition problem that is admitted small samples, but is imposed a fairly strong constraint on the form of the distribution.

When the nature strategy is unknown, the quality of decision function is assigned by the expectation $E_c EF(c,\bar{f})$ of criterion $EF\ (c,\bar{f})$, which is obtained by averaging over all distributions. This problem was solved for pattern recognition problem in the case of one discrete variable prediction [Startseva N.G.,1995], [Berikov V.B., 2002] and for regression analysis in the case of one real variable prediction [Lbov G.S., Stupina T.A., 1999].

The problem concerned at this paper generalizes the problem of pattern recognition and the problem regression analysis. From the presented above properties of the quality criterion is followed that we can use both approaches solving statistical stability problem.

Conclusion

An approach to solving the problem of heterogeneous multivariate time series analysis with respect to the sample size was considered in this paper. The solution of this problem was assigned by means of presented criterion. The universality of the logical decision function class with respect to presented criterion makes the possible to introduce a measure of distribution complexity and order all possible distributions (nature strategies) according to this measure. The logical decision function class allows us to introduce such orderings in the space of heterogeneous multivariate variables. For the fixed complexities of probability distribution and logical decision function class, the properties of this criterion are presented by means of theorem, statements and consequences. The approaches to the solution of the statistical stability sampling decision function problem were considered.

Bibliography

[Lbov G.S., 1994] Lbov G.S. Method of multivariate heterogeneous time series analysis in the class of logical decision function. Proc. RBS, 339, Vol. 6, pp.750-753.

[Lbov G.S., Starceva N.G, 1999] Lbov G.S., Starceva N.G. Logical Decision Functions and Questions of Statistical Stability. Inst. Of Mathematics, Novosibirsk.

[Lbov G.S., Stupina T.A., 2002] Lbov G.S., Stupina T.A. Performance criterion of prediction multivariate decision function. Proc. of international conference "Artificial Intelligence", Alushta, pp.172-179.

[Lbov G.S., Starceva N.G, 1994] Lbov G.S., Starceva N.G. Complexity of Distributions in Classification Problems. Proc. RAS, Vol 338, No 5,pp 592-594.

[Berikov V.,1995] Berikov V. On the convergence of logical decision functions to optimal decision functions. Pattern Recognition and Image Analysis. Vol 5, No 1, pp.1-6.

[Vapnik V.N., Chervonenkis A.Ya, 1970] Vapnik V.N., Chervonenkis A.Ya .Theory of Pattern Recognition, Moscow: Nauka.

[Nedelko V.M.,2004] Nedelko V.M. Misclassification probability estimations for linear decision functions. Proceedings of the seventh International Conference "Computer Data Analysis and Modelling". BSU. Minsk. 2004. Vol 1. pp. 171–174.

[Lbov G.S., Stupina T.A., 2003] Lbov G.S., Stupina T.A. To statistical stability question of sampling decision function of prediction multivariate variable. Proc. of the seven international conference PRIP'2003, Minsk, Vol. 2, pp. 303-307.

[Raudis Sh.Yu.,1976] Raudis Sh.Yu. Limited Samples in Classification Problems, Statistical Problems of Control, Vilnus: Inst. Of Mathematics and Computer Science, 1976, vol. 18, pp. 1-185.

[Startseva N.G.,1995] Startseva N.G. Estimation of Convergence of the Expectation of the Classification Error Probability for Averaged Strategy, Proc. Ross. RAS, vol. 341, no. 5, pp. 606-609.

[Berikov V.B., 2002] Berikov V.B. An approach to the evaluation of the performance of a discrete classifier. Pattern Recognition Letters. Vol. 23 (1-3), 227-233

[Lbov G.S., Stupina T.A., 1999] Lbov G.S., Stupina T.A.. Some Questions of Stability of Sampling Decision Functions, Pattern Recognition and Image Analysis, Vol 9, 1999, pp.408-415.

Author's Information

Gennady Lbov – Institute of Mathematics SBRAS, 4 Koptuga St, Novosibirsk, 630090, Russia; e-mail: mailto:lbov@math.nsc.ru

Tatyana Stupina – Institute of Mathematics SBRAS, 4 Koptuga St, Novosibirsk, 630090, Russia; e-mail: mailto:stupina@math.nsc.ru

К ОПРЕДЕЛЕНИЮ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Ксения А. Найденова

Аннотация: *В работе рассматриваются ключевые проблемы интеллектуального анализа данных, анализируется содержание термина «интеллектуальный анализ данных (ИАД). Показывается роль методов машинного обучения для извлечения концептуальных знаний из данных. Рассматривается абстракция данных как метод формирования знаний. Взаимосвязь между базами данных и средствами машинного обучения выделяется как ключевая проблема реализации ИАД.*

Ключевые слова: *интеллектуальный анализ данных, машинное обучение, machine learning, извлечение знаний из данных, data mining.*

Содержательный анализ ИАД

Несколько лет тому назад термин «интеллектуальный анализ данных» (ИАД) не был широко распространен, но широко употреблялся термин «интеллектуальные системы». Интеллектуальными назывались любые прикладные системы, независимо от их назначения, в которых для принятия решений в явном виде использовались знания специалистов, представленные в виде правил, процедур, эвристик, классификаций, моделей объектов и т. д. Использование знаний в интеллектуальных системах охватывало и охватывает не только проблемы извлечения знаний эксперта, но и всю проблематику машинного обучения (Machine Learning) с целью автоматизированного извлечения знаний из данных. Так, на 6-ой национальной конференции по искусственному интеллекту (КИИ-98) [Труды, 1998] работали следующие секции: «Прикладные системы», «Интеллектуальные системы и виртуальная реальность», «Интеллектуальные системы и формализация синтеза познавательных процедур», «Интеллектуальные производства и предприятия». Но уже через два года, на 7-ой национальной конференции по искусственному интеллекту (КИИ-2000) работала секция «Интеллектуальный анализ данных» [Труды, 2000]. Эта секция была посвящена работам в сравнительно новой области Data Mining (DM). Этот термин переводится как «добыча» полезных данных (сродни добыче полезных ископаемых) из баз данных (БД) и по существу употребляется как синоним термина «ИАД».

Нередко наряду с Data Mining употребляются термины Knowledge Discovery (обнаружение знаний) и Data Warehouse (хранилище данных). ИАД и управление знаниями представляют сегодня самостоятельное направление в теории и приложении интеллектуальных систем [Забежайло, 1998а].

ИАД выделяет подкласс задач, которые имеют дело с извлечением из данных зависимостей нестатистического характера. Такие зависимости позволяют делать заключения не в среднем по некоторому множеству объектов, а для каждого изучаемого объекта в отдельности. Исследуются объекты, описываемые не просто совокупностью элементов, но имеющие внутреннюю структуру, представленную набором качественных и количественных отношений между элементами. Практическая потребность в ИАД очень велика во всех областях, в том числе и в медицине, так как к настоящему времени в БД, поддерживающих различные системы принятия решений, накоплено такое количество информации, которое невозможно осмыслить и применить без помощи компьютера.

Трудности анализа данных в больших хранилищах отягощаются не только объемом и разнородным характером данных, но и открытостью знаний специалиста и динамичностью практических потребностей. Какие именно знания необходимо получить специалисту из БД, невозможно сказать заранее, а тем более формализовать в виде стандартных запросов к БД.

ИАД призван уменьшать все увеличивающийся разрыв между сбором данных и их пониманием. Практические цели ИАД: хранить данные, иметь к ним доступ в различных ситуациях, получать информацию для разных целей, часто не ограничиваясь форматом запросов к БД, создавать новые концепты и эффективно их использовать.

Практические цели «извлечения знаний из данных» приводят к тому, что ИАД охватывает великое множество методов преобразования и анализа данных от первичной обработки данных до методов машинного обучения для выявления концептуальных знаний. Методы машинного обучения включают одновременно методы выявления логических правил (импликаций, причинных зависимостей) и методы аппроксимации различных зависимостей, которые не поддаются аналитическому описанию. ИАД охватывает также статистику, распознавание образов, нейронные сети, абстракцию данных (data abstraction), онтологии, средства визуализации для поддержки анализа данных и др. Трудно также разграничить ИАД и область, которая занимается построением БЗ на основе БД. ИАД и DM есть подзадачи одной общей задачи – **Преобразование Данных в Знания.**

Разные авторы предъявляют различные, подчас противоречивые, требования к средствам анализа данных. По мнению ряда авторов, цель Data Mining состоит в выявлении скрытых закономерностей в больших и очень больших объемах данных, содержащих несколько миллионов и даже биллионов записей [Hand, 1998]. Разворачиваются дискуссии по поводу применимости статистических методов анализа данных в Data Mining. Традиционные методы статистики считаются основным инструментом анализа данных. Статистика оперирует усредненными характеристиками выборки, которые часто являются малоинформативными величинами при решении практических задач. Например, средняя платежеспособность клиента не позволяет прогнозировать состоятельность клиента и его намерения в условиях риска для принятия решений. Средняя интенсивность сигнала не выявляет характерных особенностей сигнала, положений пика сигнала и т. п.

Вот взгляд на содержание термина «ИАД» российского выдающегося специалиста Н.Г. Загоруйко: «Методы интеллектуального анализа данных (Data Mining) применяются для автоматического обнаружения эмпирических закономерностей и использования их при решении задач классификации, распознавания образов и прогнозирования. Особенность этих методов состоит в их ориентации на задачи, для которых использование традиционных статистических методов вызывает значительные затруднения. Имеются в виду задачи анализа данных очень большого объема, плохо обусловленных таблиц (количество признаков сравнимо с количеством объектов), пораженных шумами и пробелами, с признаками, измеренными в разнотипных шкалах, при отсутствии оснований для выдвижения гипотез о законах распределения» [Загоруйко, 1999].

Можно заметить некоторые противоречия в требованиях к методам DM, например, традиционные статистические методы анализа данных не применимы к данным гигантского объема, являющимся конгломератом информации разнородного характера, поступившей от разных источников. Алгоритмы распознавания и машинного обучения в особенности имеют экспоненциальную сложность и плохо приспособлены, кроме специальных модификаций, к анализу данных гигантского объема. Тем не менее, они являются главным инструментом в DM.

Обнаружение знаний в данных это область исследований, которая находится на пересечении технологии баз данных, статистики, машинного обучения (Machine Learning), распознавания образов, конструирования БЗ и многих других дисциплин, так или иначе связанных с обработкой данных и знаний.

Все авторы обзоров по ИАД [Загоруйко, 1999], [Забежайло, 1998б], [Рощупкина и Шапот, 1997], [Lavrač, 1998] согласны в том, что это направление связано с анализом гигантских объемов данных, накопленных в БД, с целью извлечения из них знаний в форме неожиданных, интересных, ранее неизвестных зависимостей, связей, ассоциаций, фактов. Большинство авторов согласны также и в том, какие методы обработки данных включаются в ИАД. Наиболее полный список методов представляется следующим образом:

1) первичная обработка данных (фильтрация, выделение однородных областей, анализ временных рядов, извлечение признаков и отдельных структурных элементов;
2) дискретизация данных (бинаризация, шкалирование, интервальный анализ, кластеризация, квантификация, укрупнение диапазонов и др.);
3) преобразование данных (быстрое преобразование Фурье, быстрое преобразование Уолша, преобразование пространства признаков и др.);
4) таксономия и методы выделения признаков;
5) выявление закономерных отношений статистическими (факторный анализ, корреляционный анализ, регрессионный анализ) и логическими (Machine Learning) методами;
6) выявление информативных признаков;
7) методы анализа структурных объектов (динамическое программирование, скрытые марковские процессы, иерархические структуры и др.);
8) распознавание образов и прогнозирование (нейронные сети; генетические алгоритмы, моделирование правдоподобных рассуждений, заполнение пробелов в БД);
9) методы моделирования сложных объектов и систем;
10) нечеткие модели и мягкие вычисления.

Объединение в одном направлении такого большого количества различных методов требует объяснения. Но, что самое важное - цель ИАД не определена достаточно ясно. Какие именно знания извлекаются из данных, каждый исследователь понимает по-разному, исходя из задач своей специальной области. С точки зрения целей исследования главным является не объем данных, **а природа данных, степень их генерализации, степень их структурированности (организованности) и степень их активности – то есть возможность их непосредственного использования, при решении задач.** Таким образом, процесс извлечения знаний из данных при их анализе управляется **содержанием тех знаний, которые пользователь хочет получить из данных.**

Чтобы более точно определить цели ИАД, рассмотрим, какие данные он объединяет. Это данные, которые имеют различную природу и разную степень генерализации: измерения (например, записи ЭКГ), данные, полученные от сенсорных датчиков, данные мониторинга (которые могут включать изображения и временные последовательности сигналов), наконец, информация в БД (такая как истории болезней, данные психофизиологического тестирования, текстовая информация, описания сложных структур через признаки и др.). Информация в БД более структурирована, чем данные сенсорных датчиков. Обычное представление данных – таблицы, в которых записи соответствуют объектам, а столбцы атрибутам или характеристикам объектов. В общем случае значения атрибутов могут быть представлены в различных шкалах – непрерывных, бинарных, категориальных и т.д.

Данные разной природы, накопленные в БД различного назначения, потенциально содержат неограниченные знания, которые можно получить, «извлечь», «добыть», но которые не содержатся явно в этих «складах» информации. Например, БД может содержать истории болезней для пациентов, проходивших лечение в различных лечебных заведениях, в различных регионах и т. д. Целью исследования может быть зависимость времени излечения пациента при некотором заболевании от региона или возраста пациента или некоторого другого фактора. Понятие «время излечения» не содержится явно в записях. Однако его можно определить по тем данным, которые есть в записях: дата поступления больного в лечебное заведение, дата выписки из лечебного заведения, дата закрытия последнего бюллетеня перед выходом на работу. То есть необходимо создать новую «абстракцию», новый признак на основе имеющихся данных. Для этого потребуется явное формулирование нового термина, сведение его к имеющимся данным с помощью программно реализованных запросов и вычислительных и/или логических операций. Кроме того, потребуются программы, формирующие необходимые выборки записей, в соответствии с градациями управляющих факторов – конкретное заболевание, регион, возраст. Может понадобиться выделить интервалы значений факторов.

Взаимосвязь между временем лечения и факторами может быть исследована разными способами, в зависимости от того, какую форму взаимосвязи выберет исследователь: вероятностную или логическую. Безусловно, это очень простой пример.

Процесс извлечения нужных концептуальных знаний может потребовать создания целой системы взаимосвязанных процессов, которые включат как предварительную обработку «сырых» данных, так и процедуры обучения компьютера правилам, определяющим требуемый концепт. Характерный пример такого более сложного процесса можно взять из области анализа изображений с целью извлечения геологических пространственных структур, связанных с рудными месторождениями. К таким структурам относятся антиклинали, синклинали, кольцевые структуры. Для выделения структур необходимо на изображении выделить сначала элементарные пространственные составные части, такие как линия. Для выделения линий можно идти двумя путями: а) построить программу, в которой заложено определение линии и правила её выделения, б) задать примеры линий на изображении и с помощью индуктивного метода построить автоматически правила выделения линий. Когда линии выделены (прямые линии и линии, имеющие кривизну, волнистые линии – примитивы), можно формировать «компьютерное» понимание, что такое «кольцевая структура». Для этого, опять-таки, необходимо либо построить распознающую программу, в которой будут заложены сформулированные экспертом правила выделения «кольца», либо использовать метод «обучения с помощью примеров» и получить правила выделения кольцевых структур с помощью компьютера.

ИАД – это многоступенчатый процесс трансформации данных в знания с многозначным выбором средств на каждом этапе обработки. **Каждый метод извлечения знаний имеет определенные условия и пределы применения, которые включают степень генерализации данных, степень их структурности, форму их представления, точность, с которой они отображают объекты и, конечно, размерность.** Рост размерности данных приводит к необходимости декомпозиции данных и процедур. Огромное значение приобретают пошаговые процедуры ИАД, которые при поступлении новых порций данных корректируют ранее полученные решения.

В целом процесс извлечения знаний из данных можно представить как многоступенчатый процесс, на каждой ступени которого происходит преобразование концептов более низкого уровня к концептам более высокого уровня, причем это преобразование происходит на основе одних и тех же принципов не зависимо от уровня генерализации и природы данных. Концепты или паттерны более низкого уровня с их выделенными признаками служат исходным планом для выделения концептов или паттернов более высокого уровня. И есть принципиально два пути выделения концептов следующего более высокого уровня иерархии: использование уже готовых программных модулей, воплощающих известные математические методы и знания специалистов о свойствах выделяемых концептов, и использование индуктивных методов обучения концептам по примерам. Этот второй путь подразумевает большую активность эксперта в управлении процессом вывода новых знаний от задания обучающей выборки до использования процедур, моделирующих рассуждения специалистов.

Прозрачность результата для пользователя в DM считается обязательным. Для этого при разработке методов анализа данных опираются на модели концептов. Один и тот же результат может получить разную интерпретацию при разных концептуальных базах. Программа DM может быть представлена следующей схемой действий: (данные + концепт) $\Rightarrow$ (программа DM) = (прозрачный результат) [Загоруйко, 1999].

DM в качестве процесса преобразования данных в знание имеет немало применений как для конструирования БЗ на основе БД, так и для организации взаимодействия между БЗ и БД. Взаимодействие осуществляется не через фиксированные запросы, а через концепты, термины и задачи проблемной области. Вот почему приобретает все большее значение новое направление в искусственном интеллекте – создание онтологий для различных прикладных областей исследований. Под онтологией понимается смысловая теория о разновидностях, свойствах объектов и связях между ними. Онтологии предоставляют терминологию для описания знаний в предметных областях [Гаврилова, 2001; Левашова и др., 2002]. Онтологии необходимы для того, чтобы извлекаемые из данных концепты одинаково понимались многими пользователями. На основе онтологий возможно объединить разные источники данных для большого числа пользователей, интегрировать знания, извлекаемые из различных БД для их повторного использования.

Методы машинного обучения и извлечение концептуальных знаний из данных в ИАД

Остановимся более подробно на методах машинного обучения (Machine Learning), предназначенных для поиска в данных логических правил и закономерностей, на основе которых формируются концептуальные знания. Эти методы по существу моделируют человеческие способы правдоподобных (индуктивных, дедуктивных, абдуктивных, по аналогии и т. д.) рассуждений, с помощью которых любой человек, и специалист в том числе, приобретает и модифицирует свои знания. Именно на основе этих методов оказывается возможным построить «интерфейс» между исследователем и интеллектуальной системой анализа данных.

Методы машинного обучения применяются к данным, которые имеют наиболее распространенную на практике форму представления в виде таблиц «объект - атрибуты», где атрибуты могут иметь различные числовые и качественные области значений.

Можно выделить два относительно самостоятельных направления в машинном обучении: конструирование концептов или абстракций из данных (Data Abstraction) и выявление закономерных связей в данных в форме различных зависимостей выполняемых на множестве объектов между атрибутами (значениями атрибутов): функциональных, импликативных, ассоциативных, отношений «класс-подкласс», «часть-целое» и т.д.

Машинное обучение включает также кластерный анализ (символьных и числовых сигналов), вероятностные каузальные сети, обучение на основе прецедентов (Case-Based Learning), метод ближайшего соседа, байесовский классификатор. Однако извлечение концептов или абстракций из данных и выявление зависимостей в данных являются основными процессами, с помощью которых происходит формирование концептуальных знаний на основе имеющихся данных, что по сути и есть главное содержание ИАД.

Абстракции данных как метод формирования знаний

В рамках этого направления наиболее развиты методы формирования и извлечения временных абстракций [Keravnou, E.T. et.al., 1996a]. Временные абстракции включают тренды, периодические события, временные паттерны. Процесс формирования абстракций есть эвристический процесс. Например, пусть необходимо сформировать абстрактное понятие «лихорадка». В этом случае можно опираться на знание, что если температура больше 39^o, то это «лихорадка». Тогда если температура $t = 41^o$, то можно применить абстракцию «лихорадка» к этому данному. Другие примеры – формирование понятия интервала времени или понятия «динамика изменения некоторого признака». Для каждого понятия необходимо разработать процедуру его определения через имеющиеся данные, т.е. осуществить сведение понятия к данным.

Рассматривают генерализационную абстракцию, определительную абстракцию, абстракцию через слияние (данные можно слить, интервал расширить), расширительную абстракцию (можно продолжить во времени некоторое свойство, если предполагается, что это свойство сохранится во времени), трендовую абстракцию (выявляется направление изменения некоторого параметра и уровень его изменения), абстракцию периодичности (выявление повторяющихся событий).

Абстракции данных становятся компонентами систем принятия решений. Абстракции можно рассматривать как частный случай онтологий, то есть определений, разделяемых всеми специалистами в данной области. Примеры создания абстракций можно найти во многих исследованиях.

Создание абстрактных интервальных концептов из временных клинических данных рассматривается в работе [Shahar and Musen, 1996]. В этой работе определяются такие абстракции как «состояние пациента», «паттерн», «событие», «градиент», «уровень». Событие влечет некоторое терапевтическое действие. Интерпретация данных пациента контекстно-зависима.

В статье [Haimowitz and Kohane, 1996] даны определения шаблонов для трендов или спецификации временных моделей для динамических процессов. Интерпретация данных вовлекает выбор шаблона, который наилучшим образом подходит к «сырым» временным данным. Производится детектирование шума в данных. Абстракции применяются для диагностики нарушений детского роста, для выявления трендов в гемодинамике.

Временные абстракции используются в системе VIE-VENT для контроля и терапевтического планирования искусственного дыхания новорожденных [Miksh et. al., 1996]. Используются три типа трендов – очень короткого времени, короткого и среднего времени. Если интерпретация указывает на тревожную ситуацию, то вовлекается процесс рассуждений и производится оценка терапии. Режим пациента может быть изменен. Количественные данные преобразуются в качественные значения на уровне оперативного контекста.

Система M-HTP & T – IDDM осуществляет мониторинг пациентов с трансплантацией сердца. Система имеет временные абстракции нескольких уровней сложности, описанные в статьях [Miksh, 1984; Larizza et. al., 1992].

В работе [Bellazzi et. al., 1998] описан интернет - сервер временных абстракций (HTTP – based Temporal Abstraction Server), который используется для ведения и поддержки больных диабетом. Осуществляется мониторинг инсулин - зависимых пациентов через телемедицинскую систему, которая обеспечивает врачей системой распределенных устройств и служб для хранения, анализа и интерпретации данных. Имеется также и система поддержки решений на основе правил.

Выводу периодичностей в данных посвящены работы Керавноу Е.Т. (Keravnou, E.T.) [Keravnou, 1996b; Keravnou, 1996c; Keravnou, 1997; Gong, 1997; Xia, 1997]. Большинство медицинских явлений возникает периодически вновь. Болезнь вновь появляется, симптомы возвращаются, лечение имеет начало и конец. Часто событие из одного клинического эпизода дает ключ к пониманию того, что может проявиться в более отдаленном периоде времени. Способность мыслить о возвращающихся событиях есть существенная часть решения медицинских задач. Время интегрирует элементы решающих медицинских систем и формирует их процессы. Базовый элемент (примитив) онтологии есть временной объект, который рассматривается как связь между свойством и его существованием во времени. Определяются повторяющийся элемент, повторяющийся паттерн и прогрессирующий. Повторяющийся элемент, в свою очередь, может быть периодическим.

История пациента есть коллекция конкретных временных объектов. Выявляются все периодичности в истории пациента с помощью двух базовых алгоритмов, один из которых выделяет периодичности первого порядка, второй – периодичности более высоких порядков. Решаются проблемы шума, пропущенных данных, производится валидизация и верификация данных.

Абстракции применяются также для генерализации данных пациента путем индуктивного вывода свойств более высокого уровня из записей репрезентативных примеров пациентов. Профили пациентов сравниваются и определяются генерализации в терминах выведенных абстракций, таких как периодичности, тренды и другие временные паттерны.

Следует сказать, что как метод формализации знаний формирование абстракций и конструирование процедур для их распознания не является новым. Примеры применения этого метода можно найти, например, в психодиагностике или психометрической психологии: психологическая характеристика есть концепт, который, с одной стороны, определяет некоторые аспекты человеческого поведения и его личности, с другой стороны, он определяется через независимые непосредственно измеряемые характеристики. Измерение происходит с помощью специально сконструированных тестов, в основе которых чаще всего лежат серии вопросов. От содержания вопросов зависит содержательное «наполнение» оцениваемого психологом концепта.

В случае с временными абстракциями речь идет о конструировании понятий, определяемых математически точно. В психометрии дело обстоит иначе. Одна и та же психологическая характеристика может пониматься разными исследователями по-разному. Так существует большое количество психологических концепций интеллекта и соответствующих им тестов: интеллект «по Векслеру», «по Спирмену», «по Терстоуну», «по Кеттелу», «по Гарднеру» и т.д. Один из основоположников измерительной психологии крупнейший французский психолог Бине как-то сказал: «интеллект это то, что мерит мой тест».

Хорошим примером формирования нового концепта может служить работа [Карпов, 2003], в которой конструируется концепт «Уровень Развития Рефлексивности» (УРФ) и разрабатывается вопросник для диагностики этого психологического свойства. После стандартной процедуры создания измерительной шкалы вопросника, её нормализации, проверки её валидности, надежности и т.д. авторы провели исследование с целью получения новых знаний о связи свойства рефлексивности с а) эффективностью

управленческой и исполнительской деятельностями и б) личностными качествами испытуемых. Для этого одновременно с измерением рефлексивности были получены экспертные оценки эффективности деятельности испытуемых, а также измерялись с помощью диагностических тестов около 40 личностных качеств. Структура личностных качеств респондентов изучалась с помощью структурограмм и с помощью матриц интеркорреляций личностных качеств. Анализировалась степень интегрированности и дифференцированности матриц интеркорреляций: матрицы сравнивались по критерию χ^2 для оценки их однородности. Для получения классов респондентов со статистически значимыми различиями их личностных свойств, применялась таксономия матриц интеркорреляций. Изучались структурные различия личностных свойств для групп респондентов с низким и высоким значением УРФ.

Новое направление концептуализации анализа мнений в социологии разрабатывается в нашей стране В.К. Финном и его сотрудниками [Финн и Михеенкова, 2002, Гусакова и др., 2001]. В этой работе, авторы предлагают определение концепта - «рациональность» - как аргументированное принятие решений или аргументированное высказывание мнений. Средствами ДСМ – метода автоматического порождения гипотез [Финн, 1999] осуществляется распознавание рационального поведения в отличие от поведения нерационального. Специалистом – социологом по каждому возможному мнению конструируется тема мнения, раскрываемая с помощью системы вопросов. В результате опроса респондентов формируется эмпирическое отношение «субъект – мнение». ДСМ – метод позволяет при анализе полученного эмпирического материала выявлять детерминанты мнений, которые могут использоваться для прогнозирования мнений или построения модели изучаемого социума. Практическое применение предложенной технологии осуществлено для анализа и прогноза электорального поведения студентов Российского государственного гуманитарного университета [Бурковская и др., 2004].

В работе [Fiorini et. al., 2004] описывается программная система GEOGINE – онтологическая модель для оптимального синтетического описания текстурных и морфологических признаков изображений, получаемых при люминесцентном микроскопическом анализе кожных покровов при ранней диагностике раковых кожных заболеваний. Система GEOGINE используется как ядро «Генератора Онтологической Модели», который создает описание кожного участка в пошаговом режиме от менее точного к более точному уровню на основе формального языка (словаря онтологий). Математический аппарат для создания онтологий – тензорный анализ, с помощью которого вычисляются инварианты, характеризующие цвет, форму, геометрические признаки изображений. Более подробное изложение онтологической модели для диагностики раковых кожных заболеваний можно найти в [Duta et. al., 2001; Flusser et. al., 2003, Dacquino et. al., 2002].

Ключевая проблема ИАД

Центральной проблемой в ИАД является проблема взаимодействия БД с системами извлечения знаний из данных, и главным образом, с системами Machine Learning (ML). С этой точки зрения основное противоречие в ИАД видится не между объемом данных и возможностями алгоритмов ML эффективно обрабатывать данные, а между реляционной структурой хранения данных в БД и структурой, удобной для реализации процессов ML.

Наш опыт решения задач ML в рамках диагностического подхода (поиск хороших классификационных тестов) показывает, что наиболее удобной структурой для решения этих задач является алгебраическая решетка на множестве данных, которые представляют собой двойственные объекты, определяемые следующим образом. Пусть $S = \{1, 2,\ldots, N\}$ – множество индексов примеров или записей, $T = \{a_1, a_2, \ldots, a_j, \ldots a_m\}$ – множество значений атрибутов, появляющихся в записях. Обозначим множество записей через R, пример или запись через t_i, $i = 1,\ldots, N$, где N – число примеров. Тогда пример $t_i \subseteq T$ есть подмножество множества значений. Данное определяется с помощью двух отображений $S \rightarrow T$, $T \rightarrow S$: $t(s) = \{$пересечение всех t_i: $i \in s$, $t_i \subseteq T\}$ и $T \rightarrow S$: $s(t) = \{i: i \in S, t \subseteq t_i\}$. Каждое данное есть пара $\{s,t\}$, такая, что $s(t) = t(s)$.

Пара введенных нами отображений в математике известна как соответствия Галуа [Ore, 1944]. Соответствия Галуа лежат в основе определений концепта и концептуальной решетки, предложенных Рудольфом Вилле [Wille, 1982].

Практически все алгоритмы вывода импликативных, функциональных, ассоциативных зависимостей из данных опираются на генерацию алгебраической решетки двойственных объектов. Операции решетки и подзадачи, которые выделяются в процессах выявления зависимостей, оказываются операциями и подзадачами правдоподобных естественных человеческих рассуждений [Naidenova, 2004b]. Таким образом, процессы машинного обучения представляют собой модель правдоподобных индуктивных рассуждений [Naidenova, 2004a]. Но при их реализации возникает проблема создания новой структуры данных – алгебраической решетки и постоянного обмена между этой структурой и какой-нибудь БД, в которой хранится исходная информация.

Одно из решений проблемы взаимодействия между БД большого объема и процедурами машинного обучения предлагается в рамках направления **On-Line Analytical Mining** [Han, 1998]. Эта работа ставит своей целью интеграцию добычи данных (Data Mining) с OLAP (On-Line Analitical Processing) технологией с получением новой OLAM (On-Line Analitical Mining) технологии.

Истоком этого направления послужили методы атрибутивно-ориентированной индукции для извлечения знаний, предложенной в [Cai et. al., 1991]. Была создана система извлечения знаний DBMiner [Chiag et. al., 1997], которая интегрирует технологию OLAP с методами Data Mining. Функции Data Mining включают: характеризацию, сравнение, ассоциацию, классификацию, предсказание, кластеризацию. Извлечение знаний происходит интерактивно, то есть при управлении с помощью мыши и при быстрой реакции системы. При этом данные для извлечения знаний выбираются порциями из различных частей много размерных баз данных и на различных уровнях абстракции. В основе обмена с БД лежит эффективное вычисление кубов данных (data cubes) – подробнее об этом можно прочесть в [Chaudhuri et. al., 1997; Zha et. al., 1997]. Различают два метода построения кубов данных: ROLAP (relational OLAP) – применяется, когда строится куб небольшой размерности и данные генерализуются к высокому уровню [Zha et. al., 1997], и MOLAP (multidimensional OLAP) применяется для много размерных данных В последнем случае нарезается много кубов малой размерности и используется браузер для навигации среди этих кубов. Разрезание и сегментирование данных выполняется с помощью мыши, управление которой встроено в браузер.

Библиография

[Бурковская и др., 2004] Бурковская Ж. И., Михеенкова М. А., Финн В. К. Об интеллектуальной системе анализа электорального поведения // Труды 9-ой национальной конференции по искусственному интеллекту с международным участием. - М.: Изд. Физико-математической литературы, 2004. Том 1. С.120 - 128.

[Гаврилова, 2001] Гаврилова Т.А. Использование онтологий в системах управления знаниями. // Труды международного конгресса «Искусственный интеллект в XXI веке». - Россия, Дивноморск. - 2001. - С. 5-13.

[Гусакова и др., 2001] Гусакова С. М., Михеенкова М. А., Финн В. К. О логических средствах анализа мнений. // НТИ. Серия 2. 2001. №5. С. 4 – 22.

[Забежайло, 1998а] Забежайло М.И. Интеллектуальный анализ данных – новое направление развития информационных технологий //НТИ. Сер. 2.- 1998. - №8. - С. 6-17.

[Забежайло, 1998б] Забежайло М.И. Data Mining & Knowledge Discovery in Data Bases: предметная область, задачи, методы и инструменты // Труды 6-ой национальной конференции по искусственному интеллекту с международным участием. - Пущино. - 1998. - Том 1. - С. 592-600.

[Загоруйко, 1999] Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во Института Математики, 1999. – 270 с.

[Карпов, 2003] Карпов А.В. Рефлексивность как психическое свойство и методика её диагностики. //Психологический журнал. Том 24. №5. 2003. С. 45-57.

[Левашова и др., 2002] Левашова Т.В., Пашкин М.П., Смирнов А.В., Шилов Н.Г. Web-DESO: система управления онтологиями // Труды 8-ой национальной конференции по искусственному интеллекту с международным участием. - М.: Физматлит, 2002. - Том 1. - С. 437-445.

[Михеенкова, 1997] Михеенкова М. А. ДСМ – метод правдоподобного рассуждения как средство анализа социального поведения // Известия РАН: Теория и системы управления. 1997. №5. С. 62-70.

[Рощупкина и Шапот, 1997] Рощупкина Б.Д., Шапот М.Д. Интеллектуальный анализ данных в бизнес приложениях: подход фирмы Cognos //Новости искусственного интеллекта. – 1997. - №4. - С. 25 –53.

[Труды, 1998] Труды 6-ой национальной конференции по искусственному интеллекту с международным участием. - Пущино, 1998. - Том 1, 2.

[Труды, 2000] Труды 7-ой национальной конференции по искусственному интеллекту с международным участием. - М.: Изд. Физико-математической литературы, 2000. - Том 1, 2.

[Финн, 1999] Финн В.К. Синтез познавательных процедур и проблема индукции. // НТИ. Серия 2. 1999. №1-2. С. 8-44.

[Финн и Михеенкова, 2002] Финн В. К., Михеенкова М.А. О логических средствах концептуализации анализа мнений // НТИ. Серия 2. 2002. №6. С. 4 – 22.

[Bellazzi et. al., 1998] Bellazzi, R., Larizza, C., and Riva, A. Temporal Abstractions for Interpreting Chronic Patients Monitoring Data, Intelligent Data Analysis, 2(2), 1998. - (http://www.elsevier.com/locate/ida).

[Dacquino et. al., 2002] Dacquino, G., Aschedamini, R.A., Fiorini, A., and Meroni, A. Tensor Invariant Model for Target Discrimination // In: Proc. of SPIE, Targets and Backgrounds VIII: Characterization and Representation, Watkins, W., Clement, D., Reynolds, W.R. Editors, Vol. 4718, pp. 170-178, Orlando, Florida, USA, April 1-3, 2002.

[Cai et. al., 1991] Cai, Y., Cercone, N., and Han, J. Attribute-Oriented Induction in Relational Databases.// In Piatetsky-Shapiro, G. and Frawley, W.J., editors, Knowledge Discovery in Databases, pp. 213-228, AAAI/MIT Press, 1991.

[Chaudhuri et. al., 1997] Chaudhuri, S. and Dayal, U. An Overview of Data Warehousing and OLAP Technology. ACMSIGMOD Record, 26: 65-74, 1997.

[Chiag et. al., 1997] Chiag S., Han J., Chee J., Chen Q., Chen S., Gong W., Kamber M., Liu G., Koperski K., Lu Y., Stefanivic N., Winstone L., Xia B., Zaiane O. R., Zhang S. and Zhu H.: DBMiner: a System for Data Mining in Relational Databases and Warehouses // In Proc. CASCONE'97: Meeting of Minds, pp. 249-260, Toronto, Canada, Nov. 1997.

[Duta et. al., 2001] Duta, N., Jain, A.K., and Dubuisson-Jolly, M. P. Automatic Construction of 2D Shape Models // IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 23, №5, 2001, pp. 433-446.

[Fiorini et. al., 2004] Fiorini, Rodolfo A., Dacquino, G., e Laguteta, G. GEOGINE – A Formal Ontological Model for Shape/Texture Optimal Synthetic Description. //In: Mathematical Methods for Learning -2004. Advances in Data Mining and Knowledge Discovery. Conference Abstracts, 2004, pp. 75-76.

[Flusser et. al., 2003] Flusser, J., Boldis, J., and Zitova, B. Moment Forms Invariant to Rotation and Blur in Arbitrary Number of Dimensions // IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 25, №2, 2003, pp. 234-245.

[Gong, 1997] Gong, W. Periodic Patterns Search in Time Related Data Sets: M. Sc. Thesis, Simon Fraser Univ., B. C., Canada, Nov. 1997.

[Haimowitz and Kohane, 1996] Haimowitz, I.J. and Kohane, I.S. Managing Temporal Worlds for Medical Trend Diagnosis. Artificial Intelligence in Medicine, 8(3): 299-321 (1996).

[Han, 1998] Han J. Towards On-Line Analytical Mining in Large Databases. ACM SIG MOD Record, 27(1), 1998, pp. 97-107.

[Hand, 1998] Hand, David J. Data Mining: Statistics and More? //The American Statistician. - May 1998. - Vol. 52, No 2, pp. 112-119.

[Keravnou, et. al., 1996a] Keravnou, E.T. et.al. Special Issue on Temporal Reasoning in Medicine. AI in Medicine, 8(3): 187-326 (1996).

[Keravnou, 1996b] Keravnou, E.T., Engineering Time in Medical Knowledge-based Systems Through Time-Axes and Time-Objects. // In: Proc. TIME-96, IEEE Computer Society Press, 1996, pp. 160-167.

[Keravnou, 1996c] Keravnou, E.T., An Ontology of Time Using Time-Axes and Time-Objects as Primitives. Technical Report TR-96-9, Department of Computer Science, University of Cyprus, 1996 // In: Proc. TIME-96, IEEE Computer Society Press, 1996.

[Keravnou, 1997] Keravnou, E.T., Temporal Abstraction in Medical Data: Deriving Periodicity. Intelligent Data Analysis in Medicine and Pharmacology (Lavrac, N., Keravnou, E.T., and Zupan, B., eds.), Kluwer, 1997, pp. 61-79.

[Larizza et. al., 1992] Larizza et. al., 1992 Larizza, C., Moglia, A., and Stefanelli, M. M_HTTP: A System for Monitoring Heart Transplant Patients. Artificial Intelligence in Medicine, 4: 111-126 (1992).

[Lavrač, 1998] Lavrac, N. Data Mining in Medicine: Selected Techniques and Applications. // In.: Proceedings of Intelligent Data Analysis in Medicine and Pharmacology - IDAMAP-98, Brighton, UK, 1998, pp. 25-37.

[Miksh, 1984] Miksh, S., Towards a General Theory of Action and Time. Artificial Intelligence, 23: 123-154 (1984).

[Miksh et. al., 1996] Miksh, S., Horn, W., Popov, C., and Paky, F. Utilizing Temporal Data Abstraction for Data Validation and Therapy Planning for Artificially Ventilated Newborn Infants. Artificial Intelligence in Medicine, 8(3): 543-576 (1996).

[Mizoguchi et. al., 1997] Mizoguchi, F. Ohwada, H., Daidoji, M., and Shirato, S. Using ILP to Learn Classification Rules that Identify Glaucomatous Eyes, In: Intelligent Data Analysis in Medicine and Pharmacology (Lavrac, N., Keravnou, E., and Zupan, B., eds.), Kluwer, 1997, pp. 227- 242.

[Naidenova, 2004a] Naidenova, X. A Model of Common Sense Reasoning Based on the Lattice Theory. // In: Conference Abstracts of International Conference "Mathematical Methods for Learning 2004" (MML-2004), ed. By Carlo Vercellis and Giovanni Felici, Como, Italy, 2004, pp. 36-39.

[Naidenova, 2004b] Naidenova, X. An Incremental Learning Algorithm for Inferring Logical Rules from Examples in the Framework of Common Sense Reasoning Process // In: "Data Mining & Knowledge Discovery Based on Rule Induction", ed. By Evangelos Triantaphyllou and Giovanni Felici, Part 4, 60 pp. (in press).

[Ore, 1944] O. Ore, "Galois Connexions", Trans. Amer. Math. Society, Vol. 55, No. 1, pp. 493-513, 1944.

[Shahar and Musen, 1996] Shahar, Y. and Musen, M.A. Knowledge-based Temporal Abstraction in Clinical Domains. Artificial Intelligence in Medicine, 8(3): 267-298 (1996).

[Wille, 1992] R. Wille, "Concept Lattices and Conceptual Knowledge System", Computer Math. Appl., Vol. 23, No. 6-9, pp. 493-515, 1992.

[Xia, 1997] Xia, B. Similarity Search in Time Series Data Sets: M. Sc. Thesis, Simon Fraser Univ., B. C., Canada, Dec. 1997.

[Yao et al., 2002] Hong Yao, Howard J. Hamilton, and Cry J. Butz, FD_ Mine: Discovering Functional Dependencies in a Database Using Equivalences, University of Regina, Computer Science Department, technical Report CS-02-04, August, 2002,ISBN0-7731-0441-0.

[Zha et. al., 1997] Zha, Y., Deshpande, P. M., and Naughton J. F. An Array Based Algorithm for Simultaneous Multi Dimensional Aggregates // In Proc. 1997 ACM SIGMOD Int. Conf. Management of Data, pp. 159-170, Tucson, Arizona, May 1997.

Информация об авторе

Naidenova Xenia Alexandrovna - Military medical academy, Saint-Petersburg, Stoikosty street, 26-1-248, naidenova@mail.spbnit.ru.

THE DEVELOPMENT OF THE GENERALIZATION ALGORITHM BASED ON THE ROUGH SET THEORY

M. Fomina, A. Kulikov, V. Vagin

***Abstract**: This paper considers the problem of concept generalization in decision-making systems where such features of real-world databases as large size, incompleteness and inconsistence of the stored information are taken into account. The methods of the rough set theory (like lower and upper approximations, positive regions and reducts) are used for the solving of this problem. The new discretization algorithm of the continuous attributes is proposed. It essentially increases an overall performance of generalization algorithms and can be applied to processing of real value attributes in large data tables. Also the search algorithm of the significant attributes combined with a stage of discretization is developed. It allows avoiding splitting of continuous domains of insignificant attributes into intervals.*

***Keywords**: knowledge acquisition, knowledge discovery, generalization problem, rough sets, discretization algorithm.*

1. Introduction

Many enterprises in the various areas create and maintain huge databases with information about their activity. However without the productive analysis and generalization such streams of the "raw" data are useless. Due to the application of methods for information generalization in decision making systems, the construction of the generalized data models and processing of large arrays of experimental data are possible. There are sources of such large dataflows in many areas. Application domains of methods for generalization include marketing, medicine, the space researches and many others. Common for these data is that they contain a great many of the hidden regularities, which are important for the strategic solutions making. However, the discovery of these regularities lays outside the human possibilities mainly because of large and permanently increasing size of the data. Therefore the methods for generalization and computer systems implementing these methods are used to derive such regularities.

Concept generalization problem under redundant, incomplete or inconsistent information is very actual. The purpose of this paper is to consider opportunities of the using the rough set theory for solution of a problem of generalization, and to propose the methods improving work of known algorithms. The new discretization algorithm of continuous attributes and the search algorithm of the significant attributes which essentially increase an overall performance of algorithms for generalization will be proposed.

2. Statement of the Generalization Problem

For the description of object we will use features a_1, a_2, …, a_k , which are further called attributes. Each object x is characterized by a set of given values of these attributes: $x=\{v_1, v_2, \ldots, v_k\}$, where v_i is value of the *i-th* attribute. Such description of an object is called *feature description*. For example, the attributes may be a color, a weight, a form, etc.

Let we have a training set U of objects. It contains both the positive examples (which are concerning to interesting concept) and the negative examples. The concept generalization problem is the construction of the concept allowing the correct classifying with the help of some recognizing rule (*decision rule*) of all positive and negative objects of training set U. Here the construction of the concept is made on the basis of the analysis of a training set.

Let's introduce the following notions related with set U. Let $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty finite set of objects. $A = \{a_1, a_2, \ldots, a_k\}$ is a non-empty finite *set of attributes*. For each attribute the set V_a is defined which refers to the *value set* of attribute a. We will denote given value of attribute a for object $x \in U$ by $a(x)$. At the decision of a generalization problem often it is necessary to receive the description of the concept, which is specified by value of one of the attributes. We will denote such attribute d and call it *decision* or *decision attribute*. The attributes which are included in A are called *conditional attributes*. The decision attribute can have some values though quite often it is binary. The number of possible values of a decision attribute d is called the rank of the decision and is designated as $r(d)$. We will denote the value set of the decision by $V_d = \{v_1^d, v_2^d, \ldots, v_{r(d)}^d\}$. The decision attribute d defines the partition of U into classes $C_i = \{x \in U: d(x) = v_i^d\}$, $1 \le i \le r(d)$.

Generally the concept generated on the basis of training set U is an approximation to concept of set X, where the closeness degree of these concepts depends on the representativeness of a training set, i.e. how complete the features of set X are expressed in it.

3. Basic Notation of the Rough Set Theory

The rough set theory has been proposed in the beginning of 80th years of the last century by the Polish mathematician Z. Pawlak. Later this theory was developed by many researchers and was applied to the decision of various tasks. We will consider how the rough set theory can be used to solve concept generalization problem (also see [1-8]).

In Pawlak's works [1, 9] the concept of an information system has been introduced. An *information system* is understood as pair $S = (U, A)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is a non-empty finite set of objects named *training set* or *universe*, and $A = \{a_1, a_2, \ldots, a_k\}$ is a non-empty finite *set of attributes*. A *decision table* (or *decision system*) is an information system of the form $S = (U, A \cup \{d\})$, where $d \notin A$ is a distinguished attribute called *decision* or *decision attribute*, A is a set of *conditional attributes*.

Let us introduce the *indiscernibility* or *equivalence relation* on the training set U: $IND(A) \subseteq U \times U$. We will say, that if $(x, y) \in IND(A)$ then x and y are indiscernible by values of attributes from A. A set of equivalence classes of relation *IND(A)* is denoted by $\{X_1^A\ X_2^A\ , \ldots,\ X_m^A\}$. Then we can approximately define set X using attribute values by the constructing of the lower and upper approximations of X, designated by $\underline{A}X$ and $\overline{A}X$ respectively. As a *lower approximation* of set X we will understand the union of equivalence classes of an indiscernibility relation which belongs to X, i.e. $\underline{A}X = \bigcup \{X_i^A \mid X_i^A \subseteq X\}$. And as an *upper approximation* of set X we will understand the union of equivalence classes which part of objects belongs to X, i.e. $\overline{A}X = \bigcup \{X_i^A \mid X_i^A \cap X \neq \varnothing\}$. The set $U \setminus \overline{A}X$ will consist of *negative objects* for X. A set

$POS_A(d) = \underline{A}C_1 \cup \ldots \cup \underline{A}C_{r(d)}$ includes objects, which are guaranteed concerning to one of the decision classes, and this set is called *positive region* of the decision system S.

Rough set X is formed by pair $< \underline{A}X, \overline{A}X >$. If upper and lower approximations of *X* are equal then *X* is an ordinary set.

The equivalence relation can be associated not only with the full set of conditional attributes *A* but also with any attribute subset $B \subseteq A$. Further this relation is denoted as *IND*(*B*) and is called a *B-indiscernibility relation*. Formally the *B*-indiscernibility relation is defined as follows: $IND(B) = \{(x, y) \in U \times U: \forall a \in B (a(x) = a(y))\}$.

Thus two objects belong to same equivalence class, if they cannot be discerned by the given subset of attributes. The concepts of *B*-upper and *B*-lower approximations based on *IND*(*B*) are similarly introduced.

Since it is not always possible to find a single-valued decision for all objects of a decision system, we will introduce notion of a generalized decision. We will define function $\partial_B: U \to \mathbf{P}(V_d)$ which is called a *generalized decision* of S on a set of attributes $B \subseteq A$, as follows: $\partial_B(x) = \{v \in V_d : \exists x' \in U \ (x' IND(B) x \ \wedge \ d(x') = v)\}$. The generalized decision ∂_A of a system S is simply called the generalized decision of S. Instead of ∂_A we also will write ∂_S. The decision table S is *consistent*, if $|\partial_A(x)|=1 \ \forall x \in U$, otherwise S is *inconsistent*.

Since not all conditional attributes are equally important, some of them can be excluded from a decision table without loss of the information contained in the table. The minimal subset of attributes $B \subseteq A$ which allows to keep the generalized decision for all objects of a training set, i.e. $\partial_B(x) = \partial_A(x) \ \forall x \in U$, is called a *decision-relative reduction* of a table $S = (U, A \cup \{d\})$. In the sequel, when considering decision tables, instead of a decision-relative reduction we will use a *reduction*.

Now let us consider the methods for concept generalization.

4. Methods of the Rough Set Theory

Generally a work of the algorithm based on a rough set theory consist of the following steps: search of equivalence classes of the indiscernibility relation, search of upper and lower approximations, search of a reduction of the decision system and constructing a set of decision rules. Moreover, discretization is applied to processing attributes with a continuous domain. In the case of the incomplete or inconsistent input information the algorithm builds two systems of decision rules, one of them gives the certain classification, the second gives the possible one. Further, we will consider the most labour-consuming steps: search of reduction and discretization making.

4.1. The Problem of Search of Reduction

Let's consider the process of search of a reduction that is very important part of any method used the rough set approach. Quite often an information system has more than one reduction. Each of these reductions can be used in procedure of decision-making instead of a full set of attributes of original system without a change of dependence of the decision on conditions that is characteristic for original system. Therefore, the problem of a choice of the best reduction is reasonable. The answer depends on an optimality criterion related to attributes. If it is possible to associate with attributes the cost function, which expresses complexity of receiving attribute values then the choice will be based on criterion of the minimal total cost. Otherwise as a rule the shortest reduction is chosen. However, the complexity of a search of such reduction consists in that the problem for checking whether exist a reduction, which length is less than some integer *s* is NP-complete. The problem of searching for a reduction with minimal length is NP-hard [10].

Thus, the problem of a choice of relevant attributes is one of the important problems of machine learning. There are several approaches based on rough set theory to its decision.

One of the first ideas was to consider as the relevant attributes those attributes, which contain in intersection of all reductions of an information system.

Other approach is related to dynamic reductions [2], i.e. conditional attribute sets appearing "sufficiently often" as reductions of sub-samples of an original decision system. The attributes belonging to the "most" of dynamic

reductions are considered as relevant. The value thresholds for "sufficiently often" and "most" should be chosen for a given data.

The third approach is based on introduction of the notion of significance of attributes that allows by real values from the closed interval [0,1] to express how important an attribute in a decision table.

4.2. Discretization Making

The stage of discretization is necessary for the most of modern algorithms for generalization. The discretization is called a transformation of continuous domain of attributes in a discrete one. For example, the body temperature of the human being, which is usually measured by real numbers, can be divided into some intervals, corresponding to the low, normal, high and very high temperature. The choice of suitable intervals and partition of continuous domains of attributes is a problem, whose complexity grows in exponential dependence on the number of attributes to which discretization should be applied.

Let's give formal definition of a considered discretization task. Let $\mathrm{S}=(U,A\cup\{d\})$ is a consistent decision system. We will assume that the domain of any attribute $a\in A$ is a real interval, i.e. $V_a=[l_a,r_a)\subset R$. Any pair of the form p^a = (a, c) where $a\in A$ and $c\in R$, we will call *cut* on areas V_a. For each attribute $a\in A$ a set P_a = $\{[c_0^a,c_1^a),[c_1^a,c_2^a),...,[c_{s_a}^a,c_{s_a+1}^a)\}$ where s_a is some integer, l_a = c_0^a < c_1^a <... < $c_{s_a}^a$ < $c_{s_a+1}^a$ = r_a and V_a = $[c_0^a,c_1^a)\cup[c_1^a,c_2^a)\cup...\cup[c_{s_a}^a,c_{s_a+1}^a)$, we will call *partition* of a domain V_a. It is easy to notice, that the partition P_a is uniquely defined by C_a = $\{c_1^a\ c_2^a, \ldots, c_{s_a}^a\}$, which is called *set of cuts* of V_a. Therefore in the sequel we often will name P_a by a set of cuts and write down as P_a = $\{a\}\times C_a=\{(a,c_1^a),(a,c_2^a),...,(a,c_{s_a}^a)\}$. Then full set of cuts P can be presented as $P=\bigcup_{a\in A}\{a\}\times C_a$.

Any set of cuts P on the basis of an original decision system $\mathrm{S}=(U,A\cup\{d\})$ determines a new decision system $\mathrm{S}^P=(U,A^P\cup\{d\})$, where A^P=$\{a^P: a\in A\}$ and $a^P(x)=i \Leftrightarrow a(x)\in [c_i^a,c_{i+1}^a)$ for any object $x\in U$ and $i\in\{0, \ldots, s_a\}$. A decision table S^P is called *P-discretization* of the table S. Our purpose is that during discretization to construct such set of cuts P.

It is obvious, that is possible to construct many of sets of cuts. Therefore there is a question how among them to find set with the minimal number of elements. For this purpose, we will introduce the following concepts.

Two sets of cuts P and P' we will regard as equivalent, if $\mathrm{S}^P=\mathrm{S}^{P'}$. We will say that set of cuts P is consistent with S, if generalized decisions of systems S and S^P are equal, i.e. $\partial_{\mathrm{S}}(x)=\partial_{\mathrm{S}^P}(x)\ \ \forall x\in U$. The consistent set of cuts P^{irr} is *irreducible* in S if any its own subset is not consistent with S. Finally, the consistent set of cuts P^{opt} we will call *optimal* in S if it has the minimal cardinality among sets of cuts, which are consistent with S.

The problem of finding optimal set of cuts P for the given decision system S is NP-complete [11]. This fact clearly speaks about importance of development of effective heuristic algorithms for search of suboptimal set of cuts.

The general approach of the most of discretization algorithms is based that any irreducible set of cuts of a decision table S is a reduction of other decision table $\mathrm{S}^*=(U^*,A^*\cup\{d^*\})$ constructed on a basis of S as follows [11].

Let $\mathrm{S}=(U,A\cup\{d\})$ be an original decision table. An arbitrary attribute $a\in A$ defines sequence $v_1^a<v_2^a<...<v_{n_a}^a$, where $\{v_1^a,v_2^a,...,v_{n_a}^a\}$ = $\{a(x): x\in U\}$ and $n_a\le n$. Objects of new decision table S^* are all pairs of objects of S with different decisions, and the set of conditional attributes is defined as cuts of attribute domains of an original decision table, i.e. A^* = $\bigcup_{a\in A}\left\{p_i^a: p_i^a=(a,c_i^a),\ \text{where}\ c_i^a=(v_i^a+v_{i+1}^a)/2,\ 1\le i\le n_a-1\right\}$.

These attributes are binary. Set A^* is named an *initial set of cuts*. We will speak, that the cut $p_i^a=(a,c_i^a)$ *discerns* objects x and y of different decision classes, if $min(a(x), a(y)) < c_i^a < max(a(x), a(y))$. A value of the new attribute corresponding to a cut p_i^a for pair (x, y) is equal to 1 if objects x and y are discerned by this cut, and 0

otherwise. Moreover a new object $\perp$ for which all conditions and the decision *d** are 0 is added to the objects of a new decision table. For all other objects of a new decision table, the new decision value is equal to 1. Reductions of a new decision table S^* determine all irreducible sets of cuts of an original decision table S.

On the basis of this general layout the heuristic algorithms finding a suboptimal set of cuts are developed. Often the discretization algorithm based on straightforward implementation of Jonson's strategy [8,12] is used. Computational complexity of this algorithm is equal to $O(|P|\cdot kn^3)$. It does its inapplicable for processing large databases. Thus, the main problem of discretization stage of continuous attributes is its high computational complexity. Now we propose the effective modification that solves this problem.

4.3. The Modification of the Discretization Algorithm

Our algorithm is directed towards the decreasing of time and memory consumption. It is based on the Jonson's strategy and extension of idea of iterative calculation of number of pairs of objects, discerned by a cut. This idea has been offered in [4], however, originally, it is applicable only when some restrictions on the decision table are imposed. This idea is based on assumption that there is a close relation between two consecutive cuts. So, for example, it is possible to notice, that in each row of the table S^* all the cells with value 1 are placed successively within one attribute. Therefore some pairs of objects are discerned by both consecutive cuts, and changes in the number of discernible pairs of objects can be only due to objects which attribute values lay between two these cuts. In [4], the situation, when no more than one object lies in this interval, is considered. We generalize this idea on a case of the arbitrary number of such objects. Thus, our algorithm extends idea of iterative calculating number of pairs of objects discerned by a cut to an arbitrary decision table.

For some cut $p_t^a = (a, c_t^a) \in A^*$ for the attribute *a* where $a \in A$ and $1 \le t \le n_a$, and some subset $X \subseteq U$ we introduce the following notation: $W^X(p_t^a)$ is a number of pairs of objects from *X* discerned by a cut p_t^a; $I^X(p_t^a)$ and $r^X(p_t^a)$ is the number of objects from *X*, which have a value of the attribute *a* less (more) than c_t^a; $I_q^X(p_t^a)$ and $r_q^X(p_t^a)$ is the number of objects from *X*, which have a value of the attribute *a* less (more) than c_t^a and belong to the *q-th* decision class, where *q* = 1, …, *r*(*d*); $N^X(p_t^a, p_{t+1}^a)$ is the number of objects from *X*, values of the attribute *a* which lay in an interval (c_t^a, c_{t+1}^a); $N_q^X(p_t^a, p_{t+1}^a)$ is the number of objects from *X*, values of attribute *a* which lay in an interval (c_t^a, c_{t+1}^a) and belonging to the *q-th* decision class, where *q* = 1, …, *r*(*d*).

Now we formulate two our theorems which underlie proposed discretization algorithm. The first theorem will allow us to derive value $W^X(p_{t+1}^a)$ from $W^X(p_t^a)$, where p_t^a and p_{t+1}^a are two consecutive cuts of a domain of the attribute *a*.

Theorem 1. Let set $X \subseteq U$ consists of $N^X(p_t^a, p_{t+1}^a)$ objects which values of the attribute *a* belongs to an interval (c_t^a, c_{t+1}^a). Then

(*a*) $I_q^X(p_{t+1}^a) = I_q^X(p_t^a) + N_q^X(p_t^a, p_{t+1}^a)\ \forall q = 1,...,r(d)$;

(*b*) $r_q^X(p_{t+1}^a) = r_q^X(p_t^a) - N_q^X(p_t^a, p_{t+1}^a)\ \forall q = 1,...,r(d)$;

(*c*) $W^X(p_{t+1}^a) = W^X(p_t^a) + N^X(p_t^a, p_{t+1}^a)\cdot\left(r^X(p_t^a) - I^X(p_t^a)\right) -$

$$-\sum_{i=1}^{r(d)} N_i^X(p_t^a, p_{t+1}^a)\cdot\left(r_i^X(p_t^a) - I_i^X(p_t^a)\right) + \sum_{i=1}^{r(d)}\left(N_i^X(p_t^a, p_{t+1}^a)\right)^2 - \left(N^X(p_t^a, p_{t+1}^a)\right)^2.$$

Let's consider a case when during discretization we have a set of cuts $P \subseteq A^*$ that defines equivalence classes $X_1, X_2, \ldots, X_m$ of the indiscernibility relation $IND(A^P)$ of table S^P, and also two consecutive cuts p_t^a and p_{t+1}^a of the attribute *a*. Then we can calculate value $W_P(p_{t+1}^a)$ from $W_P(p_t^a)$ as follows:

Theorem 2. Let there are *K* equivalence classes $X_{\alpha_1}, X_{\alpha_2}, ..., X_{\alpha_K}$ to each of which belongs $N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a) \ge 1$ objects which values of attribute *a* are within an interval (c_t^a, c_{t+1}^a). Then

$$W_P(p_{t+1}^a) = W_P(p_t^a) + \sum_{i=1}^{K}\Big[N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\cdot\Big(r^{X_{\alpha_i}}(p_t^a) - I^{X_{\alpha_i}}(p_t^a)\Big) - \sum_{q=1}^{r(d)} N_q^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\cdot\Big(r_q^{X_{\alpha_i}}(p_t^a) - I_q^{X_{\alpha_i}}(p_t^a)\Big) + \sum_{q=1}^{r(d)}\Big(N_q^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\Big)^2 - \Big(N^{X_{\alpha_i}}(p_t^a, p_{t+1}^a)\Big)^2\Big].$$

Now we present steps of our algorithm. We will name its *GID* (**G**eneralized **I**terative algorithm for **D**iscretization).

Algorithm 1. *Algorithm GID*.

Input: The consistent decision table S.

Output: Suboptimal set of cuts *P*.

Used data structures: *P* is a suboptimal set of cuts, *L* = [*IND*(A^P)] – the set of equivalence classes of an indiscernibility relation of the table S^P; *A** - a set of possible cuts.

1. $P := \varnothing$; $L := \{U\}$; *A**: = initial set of cuts;
2. For each attribute $a \in A$ do:
   begin
   $W_P(p_0^a) := 0$;
   For each $X_i \in L$ do:
   $r^{X_i} := |X_i|$; $I^{X_i} := 0$;
   for *q* = 1,..., *r*(*d*) assign $r_q^{X_i} := |\{x \in X_i : d(x) = v_q^d\}|$; $I_q^{X_i} := 0$;
   For each cut $p_j^a = (a, c_j^a) \in A^*$ do:
   For all classes X_{α_i} which objects have a value of attribute *a* from an interval (c_{j-1}^a, c_j^a) to calculate $N^{X_{\alpha_i}}$ and $N_q^{X_{\alpha_i}}$.
   Find $W_P(p_j^a)$ according to the theorem 2.
   Count values $r^{X_{\alpha_i}}, I^{X_{\alpha_i}}$ and $r_q^{X_{\alpha_i}}, I_q^{X_{\alpha_i}}$ under the theorem 1.
   end;
3. Assume as $p_{\max}$ the cut with maximal value $W_P(p)$ among all cuts *p* from *A**.
4. Assign $P := P \cup \{p_{\max}\}$; $A^* := A^* \setminus \{p_{\max}\}$;
5. For all $X \in L$ do: if $p_{\max}$ divides the set *X* into X_1 и X_2 then remove *X* from *L* and add to *L* two sets X_1 and X_2.
6. If all sets from *L* consist of the objects belonging to same decision class then Step 7 otherwise go to the Step 2.
7. End.

Let's estimate computational complexity of offered algorithm. The most labour-consuming steps of algorithm are the second and the fifth.

On step 2, during calculation of number of pairs of objects discerned by a cut $p_j^a = (a, c_j^a)$ values $r^{X_{\alpha_i}}$, $I^{X_{\alpha_i}}$, $N^{X_{\alpha_i}}$ and $r_q^{X_{\alpha_i}}$, $I_q^{X_{\alpha_i}}$, $N_q^{X_{\alpha_i}}$ are changed, where *q* = 1, ..., *r*(*d*). These operations are carried out only for those equivalence classes X_{α_i}, even which one object satisfies to the condition of belonging of value of attribute *a* to interval (c_{j-1}^a, c_j^a). For one such equivalence class it will be executed $3 \cdot r(d)+3$ described operations. We will designate this number as α. It does not depend on the number of objects *n* and the number of attributes *k*. The number of such equivalence classes cannot exceed the number n_j of objects which belong to them and which value of attribute *a* are in interval (c_{j-1}^a, c_j^a). Hence, during calculation $W_P(p_j^a)$ for one cut p_j^a it is carried out no

more than $\alpha \cdot n_j$ operations. Therefore, during processing all cuts of one attribute it will be executed $\sum_{j=1}^{n-1} \alpha \cdot n_j \le \alpha \cdot n$ operations. For processing the cuts of all k attributes it is required $\alpha \cdot kn$ operations. The second step repeats $|P|$ times. It means, that its total computational complexity is equal to $O(|P| \cdot kn)$.

On step 5 splitting equivalence classes is carried out. We take the worse case when finally any class consists of exactly one object. Since there are n objects then during work of the algorithm it will be executed n-1 splitting operations. Hence, computational complexity of the fifth step is $O(n)$.

Thus total computational complexity of the proposed discretization algorithm is equal to $O(|P| \cdot kn) + O(n) = O(|P| \cdot kn)$. It is less on two orders than computational complexity of Jonson's algorithm.

Also we estimate the space complexity of our algorithm. It should be noticed that it does not build the auxiliary table S^*. It is required only $k(n\text{-}1)$ memory cells for a storing set of possible cuts from A^*, n cells for designating an equivalence class to which belongs each of the objects, and no more than $\alpha \cdot n$ cells for storing numbers r^{X_i}, I^{X_i}, N^{X_i} and $r_q^{X_i}$, $I_q^{X_i}$, $N_q^{X_i}$ for all equivalence classes $X_i \in L$ where $i \le n$ and $q = 1,\ldots, r(d)$ and the value α does not depend on k and n. Hence the space complexity of our discretization algorithm is equal to $O(kn)$. It is less on the order than space complexity of Jonson's algorithm. For more details about our algorithm see [5, 6].

4.4. The Modification of Algorithm for Searching the Significant Attributes

In the majority of the algorithms which are based on the rough set theory and carrying out splitting of continuous attribute domains into finite number of intervals, the stage of discretization is considered as preparatory before search of significant attributes. And consequently at a stage of discretization there is a splitting of the domains of all continuous attributes, including insignificant. In this work the combined implementation of discretization with the search of a reduction is offered to make discretization only for those quantitative attributes which appear to be significant during search of a reduction.

Besides, as significant attributes we will consider the attributes, which are included in approximate reductions with sufficiently high quality. The concept of an approximate reduction [8] represents generalization of concept of the reduction considered earlier. Any subset B of set A can be considered as an *approximate reduction* of set A, and value

$$\varepsilon_{(A,d)}(B) = \frac{dep(A,d) - dep(B,d)}{dep(B,d)} = 1 - \frac{dep(B,d)}{dep(A,d)}$$

is named a *reduction approximation error*. Here the value $dep(B, d)$ represents a measure of dependence between $B \subseteq A$ and d: $dep(B, d) = |POS_B(d)|/|U|$. The reduction approximation error shows how precisely the set of attributes B approximates whole set of conditional attributes A (relatively d). Application of approximate reductions is useful while processing inconsistent and noisy data.

Thus, the developed algorithm for search of significant attributes is based on two ideas: 1) combination of discretization of quantitative attributes with the search of significant attributes, 2) search for an approximation of a reduction, but no for reduction itself. Let's name it as **G**eneralized **I**terative algorithm based on the **R**ough **S**et approach, GIRS.

4.5. Results of the Experiments

The realized experiments show that the developed algorithm allows reducing time for search of significant attributes essentially, due to combination with discretization stage and use of proposed algorithm GID.

The results of the experiments executed on 11 data sets from a well-known collection UCI Machine Learning Repository [7] of the University of California are given in table 1.

For all data sets taken into the comparison, the developed algorithm has shown classification accuracy that not concedes to other generalization algorithms, and in some cases surpasses it. Average accuracy of classification is approximately 88.9 %. It is necessary to note that the classification accuracy received by our algorithm is much above that the classification accuracy achieved by methods of an induction of deciding trees (ID3, ID4, ID5R,

C4.5) at the solving the majority of the problems. It is explained by the impossibility of representation of the description of some target concepts as a tree. Moreover it is possible to note that combining of search of significant attributes and discretization procedure is very useful. Most clearly it is visible from the results received at the decision of the Australian credit task. It is possible to explain by the presence in these data the attributes both with continuous and with discrete domains. The modification of search procedure of significant attributes is directed namely to processing of such combination.

Data set	*Classification accuracy*				
	ID3	*C4.5*	*MD*	*Holte-II*	***GIRS***
Monk-1	81.25	75.70	100	100	**100**
Monk-2	65.00	69.91	99.70	81.9	**83.10**
Monk-3	90.28	97.20	93.51	97.2	**95.40**
Heart	77.78	77.04	77.04	77.2	**78.72**
Hepatitis	n/a	80.80	n/a	82.7	**84.51**
Diabetes	66.23	70.84	71.09	n/a	**81.00**
Australian	78.26	85.36	83.69	82.5	**88.71**
Glass	62.79	65.89	66.41	37.5	**70.10**
Iris	94.67	96.67	95.33	94.0	**96.24**
Mushroom	100	100	100	100	**100**
Soybean	100	95.56	100	100	**100**
Average	81.63	83.18	88.67	85.3	**88.89**

Table 1. Comparison of classification accuracy of the developed algorithm with other known generalization algorithms.

Conclusion

We have considered the concept generalization problem and the approach to its decision based on the rough set theory. The means provided by this approach have been shown. They allow solving the problem of processing of real-world data arrays. The heuristic discretization algorithm directed towards the decreasing of time and memory consumption has been proposed. It is based on Jonson's strategy and extension of idea of iterative calculating number of pairs of objects discerned by a cut. Computational and space complexities of the proposed algorithm have linear dependence on the number of objects of decision table. Also the search algorithm of the significant attributes combined with a stage of discretization is developed. It allows avoiding splitting into intervals of continuous domains of insignificant attributes.

Bibliography

[1] Pawlak Z. Rough sets and intelligent data analysis / Information Sciences, Elsevier Science, November 2002, vol. 147, iss. 1, pp. 1-12.

[2] Bazan J. A comparison of dynamic non-dynamic rough set methods for extracting laws from decision tables / Rough Sets in Knowledge Discovery 1: Methodology and Applications // Polkowski L., Skowron A. (Eds.), Physica-Verlag, 1998.

[3] Vagin V.N., Golovina E.U., Zagoryanskaya A.A., Fomina M.V. Dostovernyi i pravdopodobnyi vyvod v intellektual'nyh sistemah (Reliable and plausible inference in intellectual systems) / Pod red. V.N. Vagina, D.A. Pospelova. Moscow, Fizmatlit, 2004. – 704 p.

[4] Nguyen S.H., Nguyen H.S. Some efficient algorithms for rough set methods / Proc. of the Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems, Spain, 1996, pp. 1451-1456.

[5] Kulikov A., Fomina M. The Development of Concept Generalization Algorithm Using Rough Set Approach / Knowledge-Based Software Engineering: Proceedings of the Sixth Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2004) // V.Stefanuk and K. Kajiri (eds). – IOS Press, 2004. – pp.261–268.

[6] Vagin V.N., Kulikov A.V., Fomina M.V. Methods of Rough Sets Theory in Solving Problem of Concept Generalization / Journal of Computer and System Sciences International, Vol. 43, No. 6, 2004. – pp. 878 - 891.

[7] Merz C.J., Murphy P.M. UCI Repository of Machine Learning Datasets. – Information and Computer Science University of California, Irvine, CA, 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html.

[8] Komorowski J., Pawlak Z., Polkowski L., Skowron A. Rough Sets: A Tutorial. / Rough Fuzzy Hybridization, Springer-Verlag, 1999.

[9] Pawlak Z. Rough Sets / International Journal of Information and Computer Science. 1982, 11(5), pp. 341-356.

[10] Skowron A., Rauszer C. The Discernibility Matrices and Functions in Information Systems / Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, Kluwer, 1992, pp. 331-362.

[11] Nguyen H.S., Skowron A. Quantization of real value attributes / Second Annual Joint Conference on Information Sciences (JCIS'95) // Wang P.P. (ed.), North Carolina, USA, 1995, pp. 34-37.

[12] H.S. Nguyen, S.H. Nguyen. Discretization methods in data mining / Rough Sets in Knowledge Discovery 1: Methodology and Applications // Polkowski L. and Skowron A. (Eds.), Heidelberg, Physica-Verlag, 1998. pp. 451-482.

Authors' Information

Marina Fomina – Moscow Power Engineering Institute, Krasnokazarmennaya str, 14, Moscow, 111250, Russia; e-mail: fominhome@mtu-net.ru

Alexey Kulikov – Moscow Power Engineering Institute, Krasnokazarmennaya str, 14, Moscow, 111250, Russia; e-mail: kulikov@apmsun.mpei.ac.ru

Vadim Vagin – Moscow Power Engineering Institute, Krasnokazarmennaya str, 14, Moscow, 111250, Russia; e-mail: vagin@apmsun.mpei.ac.ru

EXTREME SITUATIONS PREDICTION BY MULTIDIMENSIONAL HETEROGENEOUS TIME SERIES USING LOGICAL DECISION FUNCTIONS[1]

Svetlana Nedel'ko

Abstract: *A method for prediction of multidimensional heterogeneous time series using logical decision functions is suggested. The method implements simultaneous prediction of several goal variables. It uses deciding function construction algorithm that performs directed search of some variable space partitioning in class of logical deciding functions. To estimate a deciding function quality the realization of informativity criterion for conditional distribution in goal variables' space is offered. As an indicator of extreme states, an occurrence a transition with small probability is suggested.*

Keywords: *multidimensional heterogeneous time series analysis, data mining, pattern recognition, classification, statistical robustness, deciding functions.*

Introduction

The specifics of multidimensional heterogeneous time series analysis consists in simultaneous prediction of several goal variables. But the most of known algorithms construct decision function for each goal variable separately. Such approach looses some information about features interdependencies [Mirenkova, 2002].

The next problem is strong increasing of dimensionality when analysing window length increases. So one has to either simplify decision functions class or make the window shorter.

The problem of insufficient sample appears much more essential [Raudys, 2001] when rare events are to be predicted.

[1] The work is supported by RFBR, grant 04-01-00858-a

In this work an algorithm of prediction multidimensional heterogeneous time series based on finding certain partitioning that maximizes informativity criterion [Lbov, Nedel'ko, 2001] for matrix of transitions between partitioning areas. This allows to avoid increasing complexity when a window get longer, but prediction looses accuracy.

Extreme situations are characterised by low number of precedents in a period under observation. Therefore, one need statistically robust methods of multidimensional heterogeneous time series forecast.

It might be interesting also to predict events having only a few precedents or may be no precedents at all. In this case it seems to be impossible to forecast extreme situations themselves, but one could catch changing a probabilistic model of time series and consider this as an indicator of abnormal process behaviour.

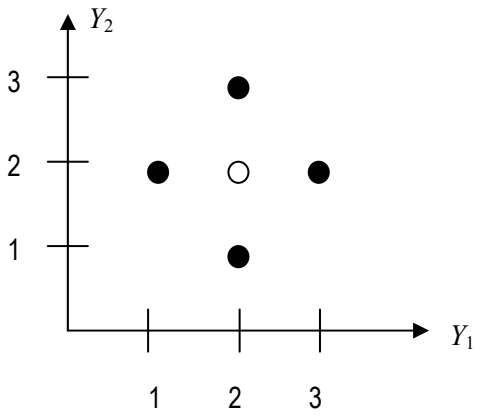


Fig. 1.

Problem Definition

Let a random n-dimensional process $Z(t)=(Z_1(t), \ldots, Z_n(t))$ with discrete time be given. Features may include both continuous and discrete (with ordered or unordered values) ones. Suppose that for a time moment t values of n variables depend on its values in previous l time moments, i. e. on a window of length l.

The most algorithms for prediction multidimensional time series use replacement of time series sample by a sample in form of data table. This is made via new notation: goal values are designated as $Y_j(t)=Z_j(t)$, and previous values (prehistory) as $X_j(t)=Z_j(t\text{-}1)$, $X_{j+n}(t)=Z_j(t\text{-}2)$, …, $X_{j+n(l\text{-}1)}(t)=Z_j(t\text{-}l)$ $j = 1, \ldots, n$.

Now any time series realization $Z(t)$, $t = \overline{1,T}$, may be represented like a sample $v = \left\{ \left(x^i, y^i \right) \middle| i = \overline{1,N} \right\}$, where $N = T - l$ — the sample size. Here $y^i = \left(y_1^i, \ldots, y_j^i, \ldots, y_n^i \right)$, $y_j^i = Y_j(i)$, $x^i = \left(x_1^i, \ldots, x_j^i, \ldots, x_m^i \right)$, $x_j^i = X_j\,(i+l)$, $m = nl$ — predictor space dimensionality. Note that the first l time moments have no prehistory of length l.

Such notation allows using a data mining methods to predict each feature separately. They may be for example classification or regression analysis methods in logical decision functions [Lbov, Startseva, 1999]. But this approach neglects features interdependence, so it is possible to construct an examples where separate decision functions give incompatible forecast [Mirenkova, 2002].

Let's consider an example that shows the weakness of separate feature forecast. Suppose two discrete features are given and probabilistic measure on them is like shown on figure 1. Each of black points has probability 0,25; another points have probability zero. Methods those make decision for every feature separately give predicted value marked by white circle. But such value combination will never occur.

This example shows necessity in methods constructing a decision rule for all features together because interdependencies are important. One need also to use decision in form of an area (in the example such area contains four black points), but not a single point.

In this work, we suggest not to separate features onto X and Y but to build partitioning in space Z directly.

Quality Criterion

Let's introduce quality criterion for decision in form of areas if goal features space. Such type criteria were proposed in [Rostovtsev, 1978].

It's suitable now to consider again separately D_X – space of predictors and D_Y – a goal features space. Let $P(E_Y)$ and $P\left(E_Y \middle| x\right)$ be unconditional and conditional measures for $E_Y \subseteq D_Y$. Suppose a set

$B_Y = \left\{E_Y^d \subseteq D_Y \,\middle|\, d = 1,...,k\right\}$ of non-intersected areas to be given. Then quality criterion will be $K(B_Y) = \sum_{d=1}^{k}\left(P\left(E_Y^d \middle| x\right) - P\left(E_Y^d\right)\right)$. Optimal decision in x will be $B_Y^* = \arg\max K(B_Y)$.

Quality criterion for conditional probabilistic measure may be defined as $K\left(P\left[D_Y \middle| x\right]\right) = \max_{B_Y} K(B_Y)$.

This criterion is some kind of distance between conditional by given x and unconditional measures on goal features space. There are known modifications those use uniform distribution instead of unconditional one.

If B_Y is a partitioning of D_Y one needs to use modified criterion:

$$K'(B_Y) = \sum_{d=1}^{k}\left| P\left(E_Y^d \middle| x\right) - P\left(E_Y^d\right)\right|. \qquad (1)$$

It differs in taking absolute values.

When the distribution is unknown and we have a sample only we can't estimate criterion for each $x \in D_X$, so need to build some partitioning λ of D_X.

Then $K(\lambda) = \sum_{E_X \in \lambda} K\left(P\left[D_Y \middle| E_X\right]\right) \cdot P(E_X)$ will be integral decision quality criterion.

All probabilities in expression may be estimated on sample.

Algorithm

Suggested algorithm makes partitioning directly in space $D_Z = \prod_{j=1}^{n} D_j$, where D_j – a set of feature Z_j all values.

Since partitioning $\lambda = \left\{E^i \in D_Z \,\middle|\, i = \overline{1,k}\right\}$ was fixed initial time series $Z(t)$ may be represented by one symbolic sequence $\beta(t) \in \left\{\beta^i \,\middle|\, i = \overline{1,k}\right\}$, where β_i – a symbol correspondent to area E^i, and $\beta(t) = \beta_i$ when $Z(t) \in E^i$.

Criterion (1) may be applied to transition matrix of process $\beta(t)$:

$$K'(\lambda) = \sum_{i_0=1}^{k}\cdots\sum_{i_l=1}^{k}\left| p_{i_0\ldots i_l} - \left(\sum_{j_0=1}^{k} p_{j_0 i_1\ldots i_l}\right)\left(\sum_{j_1=1}^{k}\cdots\sum_{j_l=1}^{k} p_{i_0 j_1\ldots j_l}\right)\right|, \qquad (2)$$

where $p_{i_0\ldots i_l} = \mathrm{P}\left(\bigwedge_{\tau=0}^{l}\left(\beta(t-\tau) = \beta^{i_\tau}\right)\right) = \mathrm{P}\left(\bigwedge_{\tau=0}^{l}\left(Z(t-\tau) \in E^{i_\tau}\right)\right)$ — the probability of given prehistory of length l.

To obtain sample estimation of the criterion need to replace $p_{i_0\ldots i_l}$ by $N_{i_0\ldots i_l}/N$ – a rate of prehistory appearance in the sample.

Transition probabilities for partitioning areas are a kind of multi-variant decision functions [Lbov, Nedel'ko, 2001].

Note that a partitioning λ may be constructed in any appropriate class, e. g. by linear discriminating functions or by logical deciding functions (decision trees).

Logical Decision Functions

For constructing a partitioning λ we shall use algorithm LRP [Lbov, Startseva, 1999] that builds a decision tree. This algorithm was designed first for classification task and applied then for various tasks of data analysis by using special quality criteria.

The algorithm builds a partitioning onto multidimensional intervals. Here an interval is a set of neighbour values when order is defined or any subset of values if feature values are unordered. Multidimensional interval is a Cartesian product of intervals.

Algorithm LRP makes sequential partitioning the space D onto given number of areas.

Since partitioning $\{E^1,...,E^i,...,E^s\}$, $E^i \subseteq D$, was constructed on step $s-1$, on step s the algorithm goes over the all areas and selects one that being split by all possible ways onto two sub-areas provides criterion maximum. Then these sub-areas replace initial area and the process is repeated until k areas been produced.

The partitioning may be represented by decision tree. Each non-terminal node ω is correspondent to some predicate $P^\omega \equiv \left(z_j \in E_j^\omega\right)$, $E_j^\omega \subseteq D_j$. Each terminal node corresponds to an area of the partitioning λ.

Rare Events Prediction

Extreme situations are characterised by low number of precedents in a sample. Therefore, statistical robustness of the methods used is especially actual. Proposed method of multidimensional heterogeneous time series prediction provide high robustness.

Nevertheless, it may be not enough if there are only several precedents. Moreover, it might be interesting to predict events having no precedents.

Obviously, in this case reliable prediction is impossible, but one could try to mark time moments where extreme situation is probable. One of indicators for such time moments may be changing a probabilistic model of time series.

Since we represent initial time series by correspondent Markov chain, all related mathematical results are available. So, a moment of changing a probabilistic model can be revealed.

Another indicator of process abnormality might be occurring in correspondent symbolic chain a transition with small probability.

Conclusion

Methods of simultaneous prediction the all variables of multidimensional heterogeneous time series allows using features interdependence information in comparison with method of separate constructing a decision function for each feature. It's possible also to build decision based on partitioning initial features space that decreases algorithm complexity. As quality criterion the method uses transition matrix informativity that was introduced.

The method proposed represents initial time series by correspondent Markov chain that allows avoiding great increasing complexity when considered prehistory length increases. This is especially important for predicting rare events. Such representation also allows applying all mathematical results related to Markov chains.

To predict time moments when extreme situations have higher probability here was suggested using changes in probabilistic model of time series.

Bibliography

[Lbov, Startseva, 1999] Lbov G.S., Startseva N.G. Logical deciding functions and questions of statistical stability of decisions. Novosibirsk: Institute of mathematics, 1999. 211 p. (in Russian).

[Rostovtsev, 1978] P. S. Rostovtsev. Typology constructing algorithm for big sets of social-economy information. // Models for aggregating a social-economy information. Proceedings, publ. IE and SPP SB AS USSR, 1978. (in Russian).

[Lbov, Nedel'ko, 2001] G.S. Lbov, V.M. Nedel'ko. A Maximum informativity criterion for the forecasting several variables of different types. // Computer data analysis and modelling. Robustness and computer intensive methods. Minsk, 2001, vol 2, p. 43–48.

[Raudys, 2001] Raudys S., Statistical and neural classifiers, Springer, 2001.

[Mirenkova, 2002] S. V. Mirenkova (Nedel'ko). A method for prediction multidimensional heterogeneous time series in class of logical decision functions // Artificial Intelligence, No 2, 2002, p. 197–201. (in Russian).

Author's Information

Svetlana Valeryevna Nedel'ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 630090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: nedelko@math.nsc.ru

CO-ORDINATION OF PROBABILISTIC EXPERT'S STATEMENTS AND SAMPLE ANALYSIS IN RECOGNITION PROBLEMS[1]

Tatyana Luchsheva

Abstract: *Considered in the paper is the method of the recognition problem decision on the base of an empirical information introduced with either probabilistic expert statements, or the sample, or expert statements and the sample simultaneously.*

Keywords*: pattern recognition, logical regularities, probabilistic expert statements.*

Introduction

The problem of construction of a decision function of recognition in the class of logical decision functions in the space of heterogeneous variables is considered [1,2]. In the given class, the probabilistic logical regularities reflecting cause-and-effect relations of the complex objects under study are constructed by the sample. By the logical regularity we mean a probabilistic expression (conjunction of values of objects' characteristics and of their combinations) having a large forecasting property. The set of such regularities introducing on a language close to a natural language of logical statements is logical-and-probabilistic model of the complex objects under study. At the same time, the extensive empirical information in the form of probabilistic expert's statements exists. Suggested method of analysis and reconciliation of different expert's statements is direct toward information can contain contradictions, reduplication, partial contradictions. Basic attention in the work will be given to the probabilistic expert statements' reconciliation problem especially when a repeated appeal to the experts for a contradiction correction is impossible.

Target Setting

Let Γ be a general population of objects under consideration. An object $a \in \Gamma$ is described with characteristics $X_1,...,X_n$; D_j is a range of variable X_j, $j = 1,...,n$. The target variable Y is made use of to indicate which pattern this object belongs to.

Let $J(a, E_j)$ denote as the predicate taking on a value "true" or "false". Predicate $J(a, E_j)$ is equal to the statement: "$X_j \in E_j$"; an object $a \in \Gamma$ is described with characteristics $X_1,...,X_n, Y$; E_j is the subset of range D_j, $j = 1,...,n$.

Let $S(a,E) = J(a, E_{j_1}) \wedge ... \wedge J(a, E_{j_d})$ name as *the conjunction of length d*. For any conjunction $S(a,E)$ in the table of data υ it is possible to determine a number of objects of the first pattern $N(1,S)$ and a number of objects of the second pattern $N(2,S)$, for which the given conjunction is true; and it can be defined the number of objects of the first pattern $N(1)$ and the number of objects of the second pattern $N(2)$ in the table of data.

Conjunction $S(a,E)$ we shall name as *the logical regularity,* describing the first pattern with the large probability (let it denote as *S**), if the following inequalities are fulfilled: $\frac{N(1,S)}{N(1)} \geq \delta$, $\frac{N(2,S)}{N(2)} \leq \beta$, where δ and β are parameters; $0 \leq \beta < \delta \leq 1$. The more δ and less β the stronger the logical regularity is. Let the set of all regularities denote as S^*.

[1] This work was financially supported by RFBR-04-01-00858

Conjunction $S(a,E)$ we shall name as *the potential logical regularity* for the first pattern (let it denote as *S'*), if the following inequalities are fulfilled: $\frac{N(1,S)}{N(1)} \geq \delta$, $\frac{N(2,S)}{N(2)} > \beta$. Let the set of all potential regularities denote as S'. Obviously, that from $S' \in S'$ it is possible to obtain the regularity *S** by a consecutive affiliation of predicates, i.e. $S' \wedge J(a,E_j) \wedge ...$; if for some conjunction $S(a,E)$ the inequality $\frac{N(1,S)}{N(1)} < \delta$ is fulfilled, conjunction S by definition is not the logical regularity and addition to S of any predicate will not give the regularity (let the set of similar conjunctions denote as S). Thus, any conjunction $S(a,E)$ can be one of three types: S^*, S', S.

Algorithm

The algorithm of finding the logical regularities consists of the consecutive execution of the following steps.

At the first step all possible conjunctions of the length 1 are considered, i.e. conjunctions of the type $S(a,E) = J(a,E_j)$, E_j - is the subset of range D_j, $j = 1,...,n$. If $S(a,E) \in S^*$, it is included in the list of the regularities and the appropriate subset E_j is excluded from further contemplation; if $S(a,E) \in S'$, the appropriate subset E_j is left for further contemplation; if $S(a,E) \in S$, the appropriate subset E_j is excluded from further contemplation. Let Q_j^1 denote as the set of subsets E_j that is left for further contemplation after execution of the first step of the algorithm.

At the second step all possible conjunctions of length 2, i.e. conjunctions of the type $S(a,E) = J(a,E_i) \wedge J(a,E_j)$, $j \neq i$, $E_i \in Q_i^1$, $E_j \in Q_j^1$ are considered. If $S(a,E) \in S^*$, the subsets E_i, E_j are excluded from further contemplation and the appropriate conjunction is included in the list of the regularities; if $S(a,E) \in S'$, appropriate subsets E_i, E_j are left for further contemplation; if $S(a,E) \in S$, the appropriate subsets E_i, E_j are excluded from further contemplation. Similarly, we denote set Q_j^2 including subsets E_j which is left for further contemplation after execution of the second step of the algorithm.

Further, similarly, the conjunctions of the length three, four, five etc. are considered.

As the result of the algorithm work, one can obtain the conjunctions of a small length only. For example, the maximal regularity length in the task described below is not more than 6.

Probabilistic Expert's Statements. Main Concepts

Let *L* experts have statements about *k* patterns. General number of logical regularities is *M* and each statement has a large forecasting property. Stage of a primary processing of the statements of an each expert separately is supplemented with the following co-ordination of the different expert's statements. Let *T* denote as the true domain of the logical regularity (statement) *S* in the space of variables $X_1,...,X_n$. Let a priori probability of two classes denote as P_1 and P_2; probability of true for the conjunction *S* as *P(S)* and conditional probabilities as *P(1/S), P(2/S).* Weights of experts we denote as $b_1,...,b_L$. If a priori information is absent then $P(S) = \frac{1}{M}$, $b_1 = ... = b_L = 1$. In this paper we consider statements co-ordination of two experts (*L*=2), and all statements will be processed together.

One Expert's Statements Agreement

Let us consider a procedure suggested of statement co-ordination on the sample of the first expert. Procedure is similar for co-ordination of other expert's statements. All the set of the first expert's statements is divided into subsets of statements so each one contains statements about the first and the second pattern with the true domains in one variables' subspace. And procedure is realized separately for each subspace.

Let $S_1^1,...,S_u^1$; $S_1^2,...,S_v^2$ are the first expert's statements about the first and the second pattern in the one variables' subspace. The statement S_i^1, $1 \le i \le u$ will be called a *contradictory statement* if the inequality is fulfilled (1) $P(2 \mid x \in T_j) \cdot \dfrac{\mu(T_i \bigcap T_j)}{\mu(T_j)} \ge \beta$ for some statement S_j^2, $1 \le j \le v$, where T_i is a true domain for the statement S_j^2, T_j is a true domain for the statement S_j^2, β is a parameter. We suggest $\beta = \theta \cdot P(2 \mid x \in T_j)$ where $\theta > 0$, $\theta \approx 1$. The set of similar statements we will denote as ω_1. The set Ω_i of statements of $\{S_1^2,...,S_v^2\}$ satisfied inequality (1) we will call as *contradictory set for the statement* S_i^1.

The statement S_i^1, $1 \le i \le u$ we will call as a *true statement* if the inequality (1) is not fulfilled for any statement S_j^2, $1 \le j \le v$. The set of such statements we will denote as ω_2.

Let $\omega_1 = \{S_{k_1}^1,...,S_{k_t}^1\}$, $\omega_2 = \{S_{k_{t+1}}^1,...,S_{k_u}^1\}$.

The following *algorithm 1* is suggested for co-ordination of true statements.

STEP 1. The different pairs $\{S_{k_i}^1, S_{k_j}^1\}$, $S_{k_i}^1, S_{k_j}^1 \in \omega_1$, $1 \le i < j \le t$ are considered.

Let $S_{k_i}^1$: " if $x \in T_1$ then the first pattern with the probability P^1".

$S_{k_j}^1$: "if $x \in T_2$ then the first pattern with the probability P^2".

The Cartesian product T_3 is compared to the pair $\{S_{k_i}^1, S_{k_j}^1\}$, so T_3 contains T_1, T_2, and T_3 is a minimal. If the inequality (2) $\dfrac{\mu(T_3 \setminus (T_1 \bigcup T_2))}{\mu(T_1 \bigcup T_2)} \ge \varepsilon$, ε is a parameter, is fulfilled for the pair $\{S_{k_i}^1, S_{k_j}^1\}$ then the statement of the type "if $x \in T_3$ then the first pattern with the probability $\dfrac{\mu(T_1) \cdot P^1 + \mu(T_2) \cdot P^2}{\mu(T_1) + \mu(T_2)}$" is included in the list $\Im^1$ of the statements left for further consideration after the first algorithm step. If for the statement $S_{k_i}^1$ and for any statement $S_{k_j}^1 \in \omega_1$, $1 \le i < j \le t$ the inequality (2) is not fulfilled then the statement $S_{k_i}^1$ is included in the list of a coordinated statements (let us denote such set as $\Im^*$) and is excluded out of ω_1.

If $\Im^1 \ne \varnothing$ STEP 2. The different pairs $\{S_{k_i}^1, S_{k_j}^1\}$, $S_{k_i}^1 \in \Im^1$, $S_{k_j}^1 \in \omega_1$ are considered.

Analogously to the step 1 the list $\Im^2$ of statements left for further consideration after the second algorithm step is found.

If $\Im^2 \ne \varnothing$ STEP 3. The different pairs $\{S_{k_i}^1, S_{k_j}^1\}$, $S_{k_i}^1 \in \Im^2$, $S_{k_j}^1 \in \omega_1$ are considered etc.

We suppose that the number of statements in the set $\Im^m$ is decreased with the steps number *m* increased. And the program stops after execution of a small number of steps.

The following *algorithm 2* is suggested for co-ordination of contradictory statements.

The statements $S^1_{k_i}$, $S^1_{k_i} \in \omega_2$, $t+1 \le i \le u$ and the corresponding contradictory sets Ω_i for the statement $S^1_{k_i}$ are considered. Let $S^1_{k_i}$: "if $x \in T_1$ then the first pattern with the probability P^1". Let $\Omega_i = \{S^2_{m_1}, ..., S^2_{m_p}\}$ and $S^2_{m_r}$, $1 \le r \le p$: " if $x \in T_{m_r}$ then the first pattern with the probability P^{m_r}". Let T is a Cartesian product so it contains T^{m_r}, $1 \le r \le p$ and T_3 is a minimal. Let us denote $T_2 = T_1 \setminus T$. If $\frac{\mu(T_2)}{\mu(T_1)} \ge \delta$, δ is a parameter, $0 < \delta \le 1$, $\delta \approx 1$, then the statement of the type: "if $x \in T_2$, then the first pattern with the probability P^1" is included in the list of coordinated statements $\Im^*$. Procedures of the true statement's co-ordination and the contradictory statement's co-ordination (algorithm 1 and algorithm 2) are similar for other pattern's processing.

Different Expert's Statements Agreement

At this step, the statements of each expert already have been co-ordinated by the procedure suggested above. For different experts' statements co-ordination the following procedure is suggested.
All the set of the *L* expert's statements is divided into subsets of statements so each one contains statements about the first and the second pattern with the true domains in the one variables' subspace. The procedure is realized separately for each subspace.

Let $S_1, ..., S_{L'}$ - are statements about the first and the second pattern in the one variable subspace. The different pairs $\{S_i, S_j\}$, $1 \le i < j \le L'$ are considered. We will find such sets of statements $S_{m_1}, ..., S_{m_p}$ of $S_1, ..., S_{L'}$ that for any pair $\{S_{m_i}, S_{m_j}\}$, $S_{m_i}, S_{m_j} \in \{S_{m_1}, ..., S_{m_p}\}$ the inequality is fulfilled: $\frac{1}{2}\left[\frac{\mu(T_{m_j} \setminus T_{m_i})}{\mu(T_{m_j})} + \frac{\mu(T_{m_i m_j} \setminus (T_{m_i} \cup T_{m_j}))}{\mu(T_{m_i} \cup T_{m_j})}\right] \le \gamma$, $\gamma \ge 0$, T_{m_i} - is a true domain of statement S_{m_i}, T_{m_j} - is a true domain of statement S_{m_j}, $T_{m_i m_j}$ - is a minimal Cartesian product contained T_{m_i} и T_{m_j}. Let e' is a number of statements of $S_{m_1}, ..., S_{m_p}$ about the first pattern, e'' - is a number of statements of $S_{m_1}, ..., S_{m_p}$ about the second pattern, T - is a minimal Cartesian product containing $T_{m_1}, ..., T_{m_p}$. To the statement with a true domain *T* impute weight $\frac{1}{L} \cdot (e' - e'')$, if $e' > e''$; weight $\frac{1}{L} \cdot (e'' - e')$, if $e'' > e'$.

Procedure is similar for other patterns.

Bibliography

[1] Lbov G.S., Starceva N.G. Logical Decision Functions and Questions of Statistical Stability. Inst. Of Mathematics, Novosibirsk, 1999.

[2] Lbov G.S., Nedelko V.M. (1997). Bayes approach to the decision of a prediction problem on the base of an statements and sample. Proc. RBS. T. 357, Vol. 1, pp. 29-32.

Author's Information

Tatyana Luchsheva – Institute of Mathematics, SB RAS, Acad.V.Koptyug St., bl.4, Novosibirsk-630090, Russia; e-mail: til@math.nsc.ru

EVALUATING MISCLASSIFICATION PROBABILITY USING EMPIRICAL RISK[1]

Victor Nedel'ko

***Abstract**: The goal of the paper is to estimate misclassification probability for decision function by training sample. Here are presented results of investigation an empirical risk bias for nearest neighbours, linear and decision tree classifier in comparison with exact bias estimations for a discrete (multinomial) case. This allows to find out how far Vapnik–Chervonenkis risk estimations are off for considered decision function classes and to choose optimal complexity parameters for constructed decision functions. Comparison of linear classifier and decision trees capacities is also performed.*

***Keywords**: pattern recognition, classification, statistical robustness, deciding functions, complexity, capacity, overtraining problem.*

Introduction

One of the most important problems in classification is estimating a quality of decision built. As a quality measure, a misclassification probability is usually used. The last value is also known as a risk. There are many methods for estimating a risk: validation set, leave-one-out method etc. But these methods have some disadvantages, for example, the first one decreases a volume of sample available for building a decision function, the second one takes extra computational resources and is unable to estimate risk deviation. So, the most attractive way is to evaluate a decision function quality by the training sample immediately.

But an empirical risk or a rate of misclassified objects from the training sample appears to be a biased risk estimate, because a decision function quality being evaluated by the training sample usually appears much better than its real quality. This fact is known as an overtraining problem.

To solve this problem in [Vapnik, Chervonenkis, 1974] there was introduced a concept of capacity (complexity measure) of a decision rules set. The authors obtained universal decision quality estimations, but these VC–estimations are not accurate and suggest pessimistic risk expectations.

For a case of discrete feature in [Nedel'ko, 2003] there were obtained exact estimations of empirical risk bias. This allows finding out how far VC–estimations are off.

The goal of this paper is to extrapolate the result on continuous case including linear and decision tree classifiers.

Formal Problem Definition

A classification task consists in constructing a deciding function that is a correspondence $f : X \to Y$, where X – a features values space and $Y = \{1, k\}$ – a forecasting values space. For simplicity let's assume a number of classes $k = 2$.

For the determination of deciding functions quality one need to assign a loss function: $L : Y^2 \to [0, \infty)$ that for classification task will be $L(y, y') = \begin{cases} 0, & y = y' \\ 1, & y \neq y' \end{cases}$, where $y \in Y,\ y' \in Y$.

By a risk we shall understand an average loss:

$$R(c, f) = \int L(y, f(x)) dP_c[D],$$

where C is a set of probabilistic measures on $D = X \times Y$ and $c \in C$ is a measure $P_c[D]$. The set C contains all the measures for those a conditional measure $P_c[Y/x]$ exists $\forall x \in X$.

[1] The work is supported by RFBR, grant 04-01-00858-a

Hereinafter we shall use square parentheses to indicate that the measure is defined on some σ-algebra of subsets of the set held, i. e. $P_c[D]: \mathrm{A} \to [0,1]$, where $\mathrm{A} \subseteq 2^D$ – a σ-algebra.

For building a deciding function there is a random independent sample $v_c = \left\{ \left(x^i, y^i \right) \in D \,\middle|\, i = \overline{1,N} \right\}$ from distribution $P_c[D]$ used.

An empirical risk will be sample risk estimation: $\widetilde{R}(v,f) = \frac{1}{N} \sum_{i=1}^{N} L\left(y^i, f\left(x^i \right) \right)$.

For the all practically used classification algorithms an empirical risk appears biased risk estimation, being always lowered, as far as the algorithms minimize an empirical risk. So, estimating this bias is actual.

Let

$$F(c,Q) = ER\left(c, f_{Q,v}\right), \quad \widetilde{F}(c,Q) = E\widetilde{R}\left(c, f_{Q,v}\right).$$

Here $Q: \{v\} \to \{f\}$ is an algorithm building deciding functions, and $f_{Q,v}$ – a deciding function built on the sample v by the algorithm Q.

An expectation is calculated over the all samples of volume N.

Introduce an extreme bias function:

$$S_Q\left(\widetilde{F}_0\right) = \hat{F}_Q\left(\widetilde{F}_0\right) - \widetilde{F}_0, \tag{1}$$

where $\hat{F}_Q\left(\widetilde{F}_0\right) = \sup_{c:\widetilde{F}(c,Q)=\widetilde{F}_0} F(c,Q)$.

We use a supremum because a distribution c is unknown and we assume the "worst" case.

Multinomial Case

In [Nedel'ko, 2003] there is reported the dependency $S_Q\left(\widetilde{F}_0\right)$ for the multinomial case when X is discrete, i. e. $X = \{1,\ldots,n\}$, and Q minimizes an empirical risk in each $x \in X$.

For the further comparison let's remember a dependency $S_Q\left(\widetilde{F}_0\right)$ in asymptotic case: $\frac{N}{n} = M = \mathrm{const}$, $N \to \infty$, $n \to \infty$. Though this is an asymptotic case, the results are applicable to real tasks because the asymptotic bias dependency is close to one for finite samples.

This asymptotic approximation is wholly acceptable already by $n = 10$, herewith it has only one input parameter M.

First, consider "deterministic" case when $\widetilde{F}_0 = 0$. In this case $S_Q(0) = \begin{cases} e^{-M/2}, & M \le 1 \\ \frac{1}{2Me}, & M \ge 1 \end{cases}$.

In general case of $\widetilde{F}_0 > 0$ there is no simple analytical formula for $S_Q\left(\widetilde{F}_0\right)$ and this dependence is given by plot.

Estimates by Vapnik and Chervonenkis

Now we can calculate an accuracy of Vapnik–Chervonenkis evaluations for the considered case of discrete *X*, as far as we know an exact dependency of average risk on the empirical risk for the "worst" probabilistic measure.

For $S\left(\widetilde{F}_0\right)$ in [Vapnik, Chervonenkis, 1974] there is reported an estimate $S'_V\left(\widetilde{F}_0\right) = \tau$, as well as an improved estimate: $S'_V\left(\widetilde{F}_0\right) = \tau^2 \left(1 + \sqrt{1 + \frac{2\widetilde{F}_0}{\tau^2}} \right)$, where τ asymptotically tends to $\sqrt{\frac{\ln 2}{2M'}}$, $M' = M \big/ \left(1 - e^{-M}\right)$.

By substitution $\widetilde{F}_0 = 0$ there is resulted $S'_V(0) = \frac{\ln 2}{M'}$.

Let's perform a simple inference of the last formula.

Consider a difference between risk and empirical risk:

$$P\left(\left|R-\widetilde{R}\right|>\varepsilon\right)=P\left(\widetilde{R}=0/R=\varepsilon\right)=(1-\varepsilon)^N .$$

Since the algorithm minimizes an empirical risk, it maximizes the distance between risks:

$$P\left(\sup_{f\in\Phi}\left|R-\widetilde{R}\right|>\varepsilon\right)<\left|\Phi\right|(1-\varepsilon)^N ,$$

where Φ is a set of all decision functions. This step implies a replacement of a probability of a sum by the sum of probabilities that makes the main contribution to VC-estimates inaccuracy. Assume right term to be equal to 1 (all probabilistic levels are asymptotically equivalent) and take logarithms:

$$\ln\left|\Phi\right|+N\ln(1-\varepsilon)=\ln 1 .$$

Since $\left|\Phi\right|=2^{n\left(1-e^{-M}\right)}$ and $\ln(1-\varepsilon)\approx-\varepsilon$ obtain:

$$S'_V(0)=\varepsilon=\frac{\ln 2}{M'} .$$

Factor $1-e^{-M}$ is a non-zero numbers probability from Poisson distribution and it appears because only "non-empty" values x contribute to capacity.

A rate:

$$\frac{S'_V(0)}{S_Q(0)}=\frac{2Me\ln 2}{M'}\xrightarrow[M\to\infty]{}2e\ln 2\approx 3{,}77$$

shows how far VC-estimates are off.

It is known that VC-estimates may be improved by using entropy as a complexity measure. Then the estimate inaccuracy will be:

$$\frac{S''_V(0)}{S_Q(0)}=2(e-1)\ln 2\approx 2{,}38 .$$

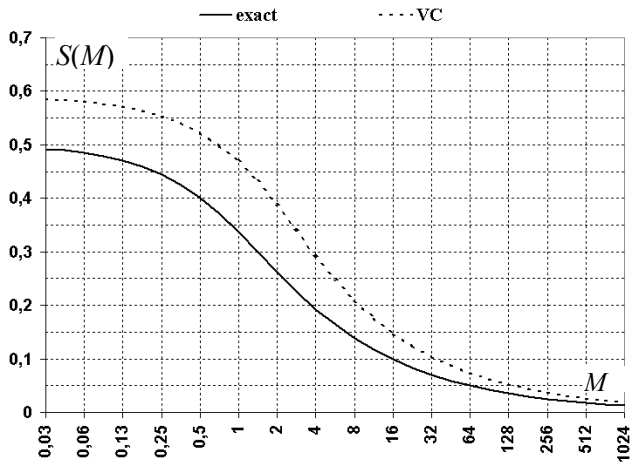


Fig. 1. Risk bias and VC–estimation. Multinomial case, $\mathrm{E}R=0{,}5$.

But in real tasks, entropy can't be evaluated and the last improvement has no use in practice.

On figure 1 there are drawn the dependency $S(M)=\max_{\widetilde{F}_0}S\left(\widetilde{F}_0\right)_M$ and its estimation $S_V(M)=\max_{\widetilde{F}_0}S_V\left(\widetilde{F}_0\right)_M$ = $=\sqrt{\frac{\ln 2}{2M'}}$. Plots demonstrate significant greatness of the last. Note that the accuracy of Vapnik–Chervonenkis estimation falls since $\widetilde{F}_0$ decreases.

By $M\le 1$ the "worst" distribution (that provides maximal bias) is uniform on X and the results obtained is consistent with results for multinomial case reported in [Raudys, 2001]. By $M>1$ and restricted $\widetilde{F}_0$ the "worst" distribution is not uniform on X.

Nearest Neighbors Method

This method assigns to each x a class that the most of nearest sample neighbours belongs to.

The number of neighbour objects taken into account is a parameter m that affects a statistical robustness.

Assume a measure on D to be uniform. Then misclassification probability for any decision function is 0,5 and empirical risk is:

$$\widetilde{F}(m)=\frac{1}{2}-\mathrm{C}_{m-1}^{\left[\frac{m-1}{2}\right]}\frac{1}{2^m} .$$

Here square parentheses denote an integer part of a value.

Figure 2 shows $S(M')$ for multinomial case (solid line) and $S(m) = 0{,}5 - \widetilde{F}(m)$ for nearest neighbours classifier, where $m = M'$.

Note that though there is no capacity concept defined for nearest neighbours method the number of neighbours m plays a role of M'.

So the case $m = 1$ corresponds to unbounded capacity (when a sample can be split via decision functions by all the ways). If capacity is unbounded, we can say nothing about expected risk using empirical risk only. But it does not mean that unbounded capacity methods can not be used, it means that they must use other risk estimators.

The fact that a risk bias for multinomial case is close to bias for nearest neighbours classifier is not accidental, because analytic expression for the first one appears to be some kind of averaging the bias for the second case.

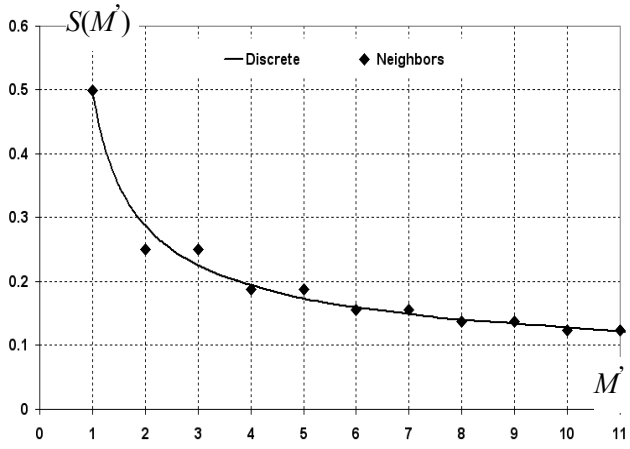


Fig. 2. Risk biases for multinomial and nearest neighbours classifiers.

Linear Decision Functions

Let us compare risk bias values for discrete case with bias for linear decision functions.

For simplifying, there was considered uniform distribution on features for both classes. For such c misclassification probability equals to 0.5 for every decision function, but empirical risk appears to be much lower.

Tab. 1. Risk bias for linear decision functions

d	N	M'	S	S_F	d	N	M'	S
1	3	1.16	0.4	0.4	1	10	2.31	0.27
1	20	3.75	0.2	0.2	1	50	7.53	0.13
1	100	13.1	0.1	0.1	2	4	1.05	0.47
2	10	1.53	0.36	0.27	2	20	2.33	0.27
2	50	4.44	0.18	0.13	2	100	7.53	0.13
3	5	1.02	0.48	0.35	3	10	1.25	0.41
3	20	1.79	0.32	0.2	3	50	3.28	0.22
3	100	5.46	0.16	0.09	4	10	1.11	0.45
4	20	1.5	0.36	0.19	4	50	2.66	0.25
5	10	1.04	0.48	0.27	5	50	2.27	0.28

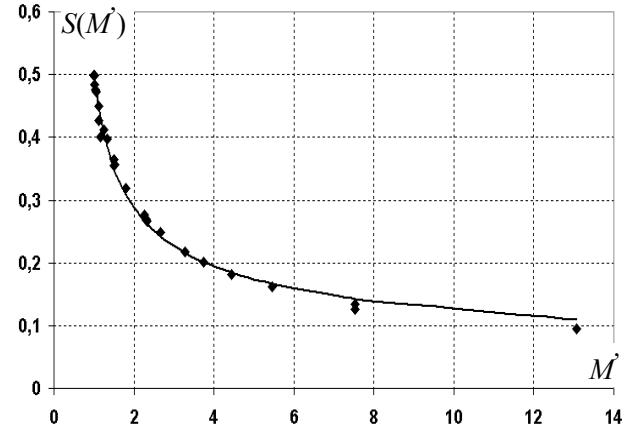


Fig. 3. Risk biases for multinomial and linear classifiers.

To find a dependence $S(M)$ for linear deciding functions in $X = [0,1]^d$ a statistical modelling was used. By the modelling there was for each combination of parameters a hundred of samples drawn from uniform distribution on D, for each sample the best linear classifier built by exhaustive search. Note that the uniform distribution on D provides maximum of empirical risk bias since we put no restrictions on $\widetilde{F}_0$.

A table 1 shows the result of modelling. Here d – features space X dimensionality, N – sample size, $M' = \dfrac{N}{\log_2 \mathbf{C}}$ – sample size divided by VC-capacity of linear functions class ($\mathbf{C} = 2\sum_{m=0}^{d} C_{N-1}^{m}$ is a total number of possible decision assignments to sample points by using linear decision functions), S – risk bias.

The same results are shown (by markers) on fig. 3 in comparison with $S(M')$ for discrete case (solid line).

Obtained results show that bias dependence on M' for linear functions is close to dependence for discrete (multinomial) case.

If an algorithm does not perform exhaustive search then a risk bias appears to be lower. This fact is illustrated in table 1 by value S_F that is a risk bias for the Fisher's discriminator.

Decision Tree Classifier

The goal now is to evaluate a risk bias for decision functions in form of binary decision trees [Lbov, Startseva, 1999].

Decision tree is a binary tree with terminal nodes marked by goal class (certain value *y*) and non-terminal nodes marked by predicates in form: $X_j < \alpha$, where α is a value. Two arcs starting from each non-terminal node correspond to true and false predicate values.

Each decision tree forms certain sequential partitioning in X.

There was the exhaustive search algorithm implemented. The search is performed over the all decision trees with L terminal nodes and the best tree minimizing an empirical risk is founded.

While searching, the algorithm counts $\mathbf{C}$ – the number of different assignments *y* to sample objects.

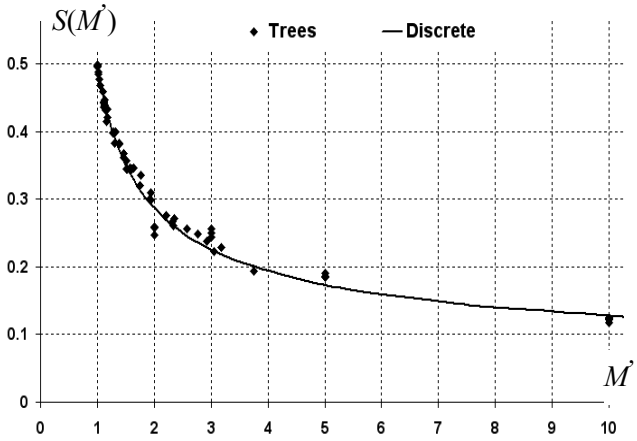


Fig. 4. Risk biases for multinomial and tree classifiers.

Tab. 2. Risk bias for tree decision functions

d	*N*	*L*	*M'*	*S*	*d*	*N*	*L*	*M'*	*S*
1	2	1	2	0.26	1	2	2	1	0.5
1	5	2	1.51	0.36	1	5	3	1.12	0.44
1	10	2	2.31	0.27	1	10	3	1.53	0.34
1	20	2	3.76	0.19	1	20	3	2.33	0.26
1	20	5	1.50	0.34	2	5	2	1.26	0.40
2	5	3	1.02	0.49	2	10	2	1.92	0.30
2	10	3	1.28	0.40	2	20	2	3.19	0.23
2	20	3	1.94	0.31	2	20	4	1.46	0.37
3	5	2	1.17	0.42	3	20	2	2.92	0.24
3	20	3	1.77	0.34	3	20	5	1.12	0.45
4	20	2	2.76	0.25	5	10	2	1.57	0.35

Since $\mathbf{C}$ essentially differs on different samples one need to evaluate entropy $H = \mathrm{E}\log_2 \mathbf{C}$.

Then $M' = \frac{N}{H}$.

Table 2 shows statistical robustness of decision trees by different parameters while uniform distribution on D assumed. The same result is shown on figure 4 in comparison with multinomial case.

One can see again that risk bias is caused and determined by M' (sample size per complexity) rather than any other factor.

Let's compare complexities (capacities) of decision trees and linear classifier.

Table 3 shows linear classifier dimensionality d' that provides the same entropy (average number of different assignments *y* to sample objects) like decision trees with L terminal nodes in d-dimensional space.

Though decision trees seem to be simple, they have essential capacity. For example if $L = d$ decision trees capacity exceeds capacity of linear classifier.

But, the most of algorithms do not perform exhaustive search in whole class of decisions and their capacities are expected to be lower.

Note that if an algorithm implements good heuristic search and always founds the best decision function, then its capacity will be nevertheless equal to the capacity of exhaustive search algorithm. So, there is no use to count a number of decisions being really tested by an algorithm, because this number is irrelevant to actual capacity.

Hence, calculation of effective capacity requires different approach. Effective algorithm capacity may be estimated by the following way.

First one need to perform statistical modelling using uniform distribution on D. In this case misclassification probability (risk) equals to 0,5 for any decision function. Expectation of empirical risk is estimated by modelling, so risk bias is estimated too.

Then via comparing the bias obtained by modelling with the bias for exhaustive search algorithm, the effective capacity of the algorithm under investigation is easily revealed.

Tab. 3. Correspondent dimensionality for tree and linear decision functions. Non-integer values of d' appears because of interpolation performed.

d	*N*	*L*	*d'*	*d*	*N*	*L*	*d'*
1	5	2	1	2	5	2	1.56
2	10	2	1.4	2	20	2	1.3
3	2	2	1	3	5	2	1.83
3	10	2	1.64	3	20	2	1.47
4	5	2	2.09	4	20	2	1.59
5	10	2	1.93	10	10	2	2.45
1	5	3	2	2	5	3	2.95
2	10	3	2.86	2	20	3	2.66
3	5	3	3.76	3	10	3	3.48
3	20	3	3.07	4	5	3	3.99
4	10	3	3.94	2	5	4	3.99
2	20	4	4.26	3	5	4	4
3	10	4	5.82	3	20	4	5.1
4	10	4	6.77	1	10	5	4
2	10	5	6.45	3	15	5	7.77

Conclusion

Risk estimates by Vapnik and Chervonenkis are known to be excessively pessimistic. But the approach based on complexity measure is very attractive because of universality. The work presented shows that the reason for such pessimistic estimates is an inaccurate inference technique, but not the worst case orientation. So, it is possible to obtain estimates assuming the “worst” distribution and the ‘worst’ sample but these estimates will be appropriate in practice.

For the multinomial case (a discrete feature) there was found how far Vapnik–Chervonenkis risk estimations are off. For continuous features the dependence of risk bias on complexity in considered cases is close to multinomial one that ensures a possibility to apply obtained scaling of VC-estimates to real tasks, e.g. linear decision functions and decision trees. The results obtained for multinomial case may be propagated on continuous one by using VC-capacity of decision function class instead of n.

Comparison of linear classifier and decision trees capacities is also performed.

There was also described a method for estimation an effective capacity of an algorithm that does not perform exhaustive search in the class of decision functions.

Bibliography

[Vapnik, Chervonenkis, 1974] Vapnik V.N., Chervonenkis A. Ja. Theory of pattern recognition. Moscow “Nauka”, 1974. 415p. (in Russian).

[Raudys, 2001] Raudys S., Statistical and neural classifiers, Springer, 2001.

[Lbov, Startseva, 1999] Lbov G.S., Startseva N.G. Logical deciding functions and questions of statistical stability of decisions. Novosibirsk: Institute of mathematics, 1999. 211 p. (in Russian).

[Nedel’ko, 2003] Nedel’ko V.M. Estimating a Quality of Decision Function by Empirical Risk // LNAI 2734. Machine Learning and Data Mining, MLDM 2003, Leipzig. Proceedings. Springer-Verlag. 2003. pp. 182–187.

Author's Information

Victor Mikhailovich Nedel’ko – Institute of Mathematics SB RAS, Laboratory of Data Analysis, 660090, pr. Koptyuga, 4, Novosibirsk, Russia, e-mail: nedelko@math.nsc.ru

2.2. Structural-Predicate Models of Knowledge

SCIT — UKRAINIAN SUPERCOMPUTER PROJECT

Valeriy Koval, Sergey Ryabchun, Volodymyr Savyak, Ivan Sergienko, Anatoliy Yakuba

***Abstract**: the paper describes a first supercomputer cluster project in Ukraine, its hardware, software and characteristics. The paper shows the performance results received on systems that were built. There are also shortly described software packages made by cluster users that have already made a return of investments into a cluster project.*

***Keywords**: supercomputer, cluster, computer structure.*

Introduction

To solve the most important tasks of an economy, technology, defense of Ukraine, that have large and giant computing dimensions, we need to be able to calculate extralarge information arrays. Such extremely large computations are impossible without modern high-performance supercomputers.

Unfortunately, such computational resources are almost unavailable in Ukraine today. This can cause a precarious situation development in a different country's life areas. We can lose leading positions in a science, science intensive products' development, complex objects and processes modelling and design technologies.

It is also impossible to import large supercomputers for the above mentioned tasks, because of embargo (for really powerful supercomputers), their extra-large prices, practically impossible upgrade, requirements to control the usage of imported supercomputers from abroad. In this situation Ukraine and other countries (India, China, Russia, Belarus) need to design its national supercomputers [1].

Today in Glushkov Institute of Cybernetics NAS of Ukraine, two high-performance and highly effective computational cluster systems SCIT-1 and SCIT-2 are running in an operation-testing mode. They are built on the basis of modern microprocessors INTEL® XEON™ и INTEL® ITANIUM® 2.

On the basis of these supercomputer systems, a powerful joint computer resource will be built. It will be available for access for users from different organisations from different regions from all the NAS of Ukraine. The systems built are focused to applications from the fields of molecular biology, genetics, science of materials, solid-state physics, nuclear physics, semiconductor physics, astronomy, geology.

Development Ideology

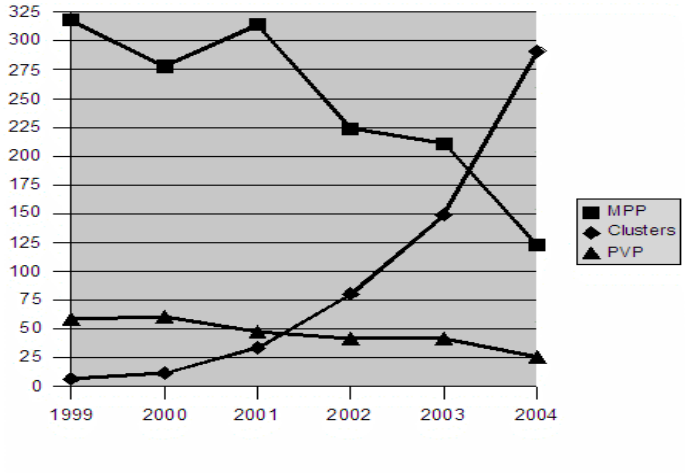
While developing a supercomputer, system scientists and engineers face, a great amount of questions that requires to run a different kind of experiments. The experiments are run to understand a performance, features and characteristics of architecture, hardware platform for computing node solution, node interconnections, networking interfaces, storage system [2].

To make a right **decision on system architecture** we have made an analysis of world supercomputer tendencies. One of major sources we used was top500 list of the largest supercomputer installations. An analysis we made proves us, that a solution of cluster architecture is a right one.

Cluster computer system – is a group of standard hardware and software components, coupled to solve tasks. Standard single processor or SMP (symmetric multiprocessor system) are used as processing elements in a cluster. Standard high-performance interconnect interfaces (Ethernet, Myrinet, SCI, Infiniband, Quadrics) are used to connect processing elements in a cluster system.

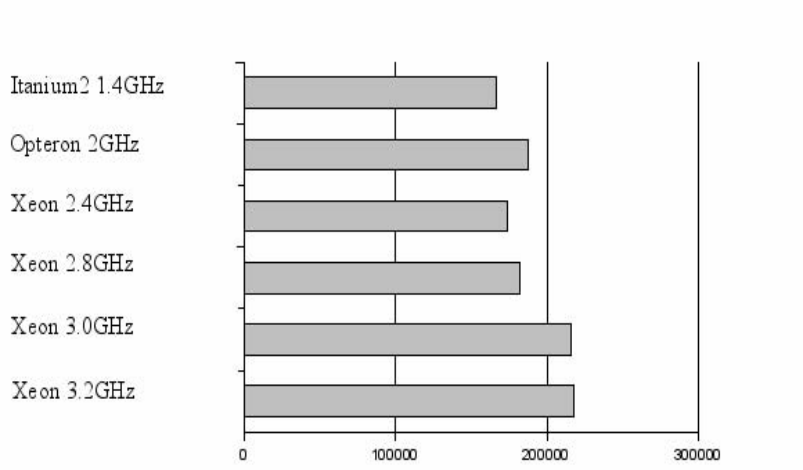
A development of supercomputer systems with cluster architecture is one of the most perspective ways in the world of high-performance computations today. The amount of supercomputer clusters installed in the world in increasing rapidly and the amount of finances spent for this direction is also increased.

Tendencies of a development of supercomputers in the world for **MPP** (Massively Parallel Processing), **PVP** (Parallel Vector Processor) and cluster systems are shown on a Picture 1. As shown on the picture below, clusters are dominated in top500 list. For the several last years, an amount of cluster systems in the list have grown and an amount of **MPP** and **PVP** systems is going down.



Picture 1. World supercomputer tendencies

When making a selection of a hardware platform of computational nodes we analyzed price/performance ratio. As LINPACK is rather narrow test, we choose SPECfp tests understand a performance of nodes on the basis of different kind of real applications. The prices we calculated were taken from Ukrainian IT market operators. The diagram received in analysis is shown on a Picture 2.



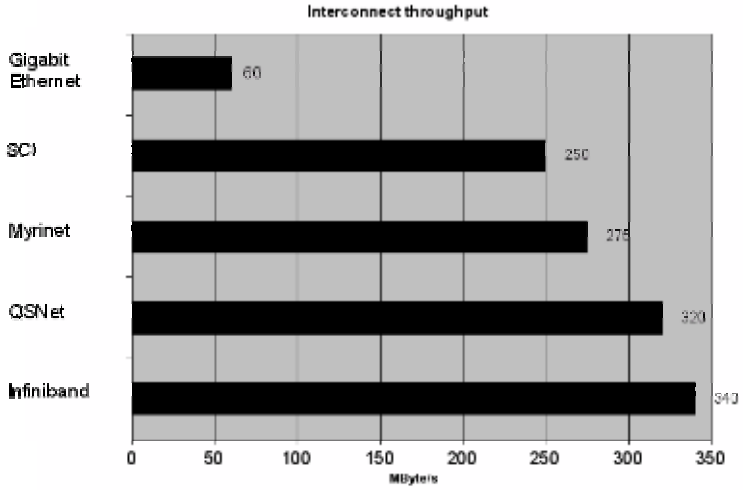
Picture 2. Price/performance ratio on the basis of SPECfp analysis

Today also new SPEChpc tests are available, that can give us an understanding of hpc computers performance for an applications from chemical, environment, seismic area and also OpenMP and MPI applications testing.

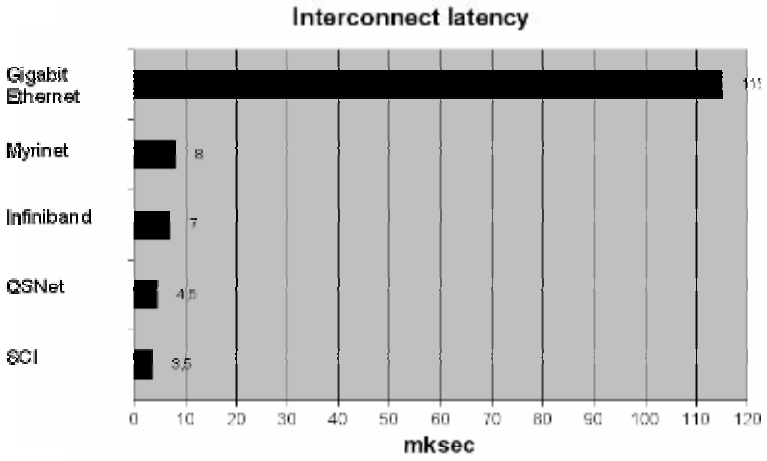
Price/performance analysis is made with a calculation of costs of all the main components of a system and its environment with a focus on a theoretical peak 300 GFlops performance, which is near 120 000 SPECfp. We have also take into consideration performance downsize for different platforms scaling on the basis of self made tests.

After the analysis, we choose an Itanium2 solution as the best scaling and best price/performance solution for floating point calculation intensive applications. But we understood that the selection of a newest Itanium2 architecture could cause problems with available 32-bit applications porting. So, we decided to build two systems. For SCIT-1 - a 32xCPU system we choose Xeon 2.67GHz platform and for SCIT-2 - a 64xCPU system we choose Itanium2 1.4GHz platform as the best one in 64-bit floating-point performer. It was also taken into account good perspective of Itanium2 architecture and its ability to operate faster with big precision operations and big memory. The other valued characteristics that cause better price/performance ratio of an Itanium2 systems is its best power/performance ratio between other well known general usage processors.

Design and a selection of internode communicational interfaces was done from the best performing ones. When making experiments with one of the software packages (Gromacs), we have found that a low latency is the most important issue for cluster scalability. We have seen from world published data and our own experiments that some of the tasks which don't scale more then 2-4 nodes on Gigabit Ethernet scales easily to 16 nodes on low-latency interconnect interfaces [3].



Picture 3. Interconnect throughput.



Picture 4. Interconnect latency

Understanding an importance of latency and throughput of an interface, we have made a price/performance analysis for interfaces available in Ukraine. The best one for 16x and 32x nodes' clusters we planned to build was SCI (Scalable Coherent Interface). Performance parameters of communicational interface received on 3rd quarter of the year 2004 for Intel Xeon platforms are shown on Picture 3 and Picture 4. Today these pictures will look different (because of changes in platforms and interfaces itself), but price/performance leaders for latency intensive applications are SCI and QSNetII; for throughput intensive applications they are QSNetII and Infiniband. For small clusters an SCI is a preferable interface. But it has also one more useful feature. An SCI system

network can be built on 2D mash topologies. Such architecture gives an ability to transfer data into two ways simultaneously. But to receive an advantage from this technology, software should be written with an understanding of this ability.

It is known that performance and intelligence are the most important factors promoting the development of modern universal high-performance computers. The first factor forced a development of parallel architectures. The rational base of this development is universal microprocessors, connected into cluster system architectures. The second factor becomes clear when the notion of **machine intellect** (MI) is used. The concept of MI is introduced by V.M.Glushkov. MI defines "internal computer intelligence" and the term "intellectualisation" is used to define an increase of machine intellect. During the last 5-6 years, V.M.Glushkov Institute of Cybernetics NAS of Ukraine carries out the research aimed at the development of cluster based, **knowledge-oriented architectures** called ***intelligent solving machines*** (ISM). ISM implementing high- and super-high-level languages (HLL and SHLL) and effective operation with large-size data- and knowledge bases. They operate as with traditional computation tasks (mathematical physics, modelling of complex objects and processes, etc.) as **artificial intelligence** (AI) tasks (knowledge engineering, pattern recognition, diagnosis, forecasting) [4].

Large-size complex data- and knowledge bases in these clusters are displayed as **oriented graphs** of an arbitrary complexity – trees, semantic networks, time constrained, etc. In ISM computers it is possible to build graphs with millions nodes and to represent various knowledge domains. It is also important that the developed architecture can be easily integrated with distributed database architectures, which are developed in Glushkov Institute of Cybernetics NAS Ukraine. This database architecture makes search processes and data processing much faster than solutions with traditional architectures.

The intellectual part of the cluster systems developed together with distributed databases is an advantage of this solution as compared with the systems developed in the other sites of the world.

Hardware and software of the systems developed. Today the following SCIT (**s**uper**c**omputer for **i**nformational **t**echnologies) supercomputers are built in the institute (Picture 5):

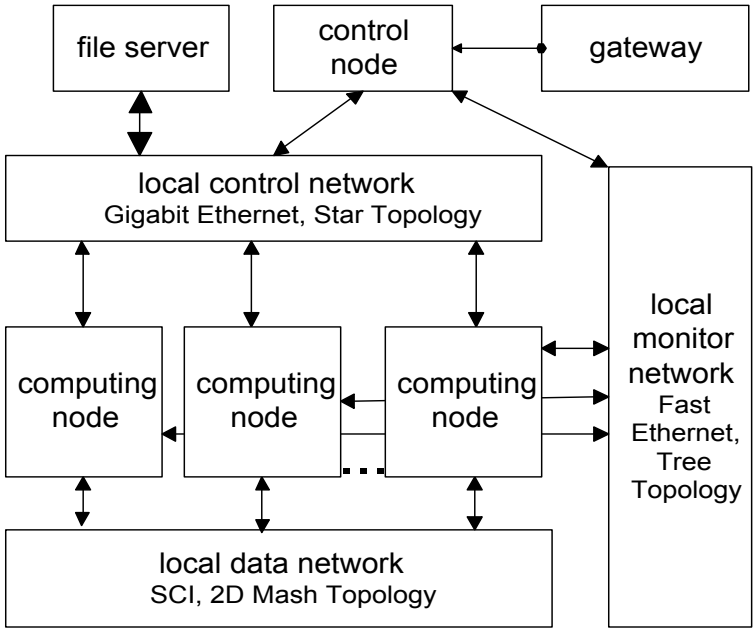
Picture 5. Photo of SCIT clusters.

SCIT-1 – 32xCPU, 16xNodes cluster on the basis of Intel Xeon 2.67GHz 32-bit processors. They are oriented to operate with 64-bit and 128-bit data. The peak performance of SCIT-1 is 170 GFlops with an ability to be upgraded to 0,5-1 TFlops (right on a photo).

SCIT-2 – 64xCPU, 32xNodes cluster on the basis of Intel Itanium2 1.4GHz 64-bit processors. They are oriented to operate with 128-bit and 256-bit data. The peak performance of SCIT-2 is 358 GFlops with an ability to be

upgraded to 2,0-2,5 TFlops. The storage system has capacity of 1 TByte and ability to be upgraded to 10-15 TBytes (left on a photo).

Each of two clusters is an array of computing nodes, connected together with three networks. The first one is a system network, based on SCI interface. The second one is a file data network, based on Gigabit Ethernet interface. The third one is a management network, based on Fast Ethernet interface. A general block-scheme of the SCIT supercomputer is shown on a Picture 6.



Picture 6. SCIT cluster structure.

A local data network is based on SCI and is used for a high-performance low-latency inter-node communication during a calculation process. A local data network is built as 2D mash. For 16x node cluster it is configured as 4x4 or 2x8 2D mash. For 32x node cluster it is configured as 4x8 or 2x16 2D mash. On data transfers based on MPI the throughput for Xeon E7501 platforms is 250 MB/s, for Itanium2 8870 platforms – 355 MB/s.

A local control network is based on Gigabit Ethernet and is used to handle all cluster-computing nodes and to transfer data files between nodes and file server.

A local monitor network is used for service information transfer and monitoring of all the cluster system.

On a table 1 performance parameters of SCIT-1 and SCIT-2 systems are described.

Table 1. 64-bit performance parameters of SCIT-1 and SCIT-2 systems.

	SCIT-1	SCIT-2
1 Processors	P-IV Xeon 2,67 GHz	Itanium2 1,4 GHz
2 Peak performance of a single processor		
Integer operations per second, 10^9IPS	1,34	5,6
Floating point operations per second, GFLOPS	5,34	5,6
Node system bus performance, GB/s	4,2	6,4
3 Total peak performance of a system		
Integer operations per second, 10^9IPS	43	358
Floating point operations per second, GFLOPS	170	358
Total system bus performance, GB/s	67,2	204,8
4 Linpack performance of a system, GFLOPS	112,5	280

Performance characteristics of developed systems SCIT-1 and SCIT-2 are on the one stage with world best systems. They are also one of the best systems in a world mathematical supercomputing construction.

The creation of cluster systems SCIT-1 and SCIT-2 and their integration and finally launch was made due to a fruitful cooperation of Glushkov Institute of Cybernetics NAS of Ukraine with USTAR scientific and manufacturing company (based in Kiev) and Intel corporation (International). The partners of the institute delivered a technical support and consulting of a project.

System Level Software

Components of system level software of a cluster support all stages of user-level parallel software development. They also provide execution of users processes of substantial processing on a solving field. They run on all the nodes of a cluster and a control node as well. Operating system used are **ALT Linux** for SCIT-1 and **Red Hat Enterprise Linux AS** for SCIT-2. Message Passing Interface (MPI) over SCI is used for programming in a message-passing model. In addition, system level software includes optimized compilers of C, C++, Fortran languages for parallel programming, fast Math libraries, etc.

Application Level Software

The powerful hardware, system level, service and specific cluster software integrated in a system is a strong ground for an application level software development. It gives an ability to solve new extra large tasks in a fields of science, economy, ecology, agriculture, technology, defense, space industry, etc.

Due to successful implementations of SCIT systems for the several months after the system was installed a lot of applications were developed and deployed on a supercomputer in Glushkov Institute of Cybernetics NAS of Ukraine.

The software packages for the following tasks were developed:

- soil ecology problems solution;
- seismic data processing;
- dynamical travelling salesman in a real time;
- modelling a structural-technological changes in a developing economy;
- a search for an optimal service center placement;
- construction of an interference-tolerant code;
- risk classification and evaluation decisions;
- data clusterization with genetics algorithms;
- decomposition, calculation, verification and solving of a theorems;
- software component for linear algebra;
- low-energy orbit selection;
- software package for a natural and technogenic processes analysis.

Conclusion

The supercomputer cluster project, as a first stage of a national supercomputer resources development, made a great impact on an intellectualisation of information technologies in Ukraine. The next stage will be devoted to improvement of performance characteristics of supercomputers designed and their software. This should allow extending an amount of large complex tasks that would be solved on the systems.

Bibliography

1. Koval V.N., Savyak V.V., Sergienko I.V., “Tendencies of modern supercomputer systems development”, Control Systems and Computers, Vol.6, November-December 2004, 31-44 pp (In Russian).
2. Koval V.N., Savyak V.V., “Multiprocessor cluster systems: planning and implementation”, “Nauka osvita”, Artificial Intellect, Vol.3, 2004, 117-126 pp (In Russian).

3. www.gromacs.org
4. Koval V., Bulavenko O, Rabinovich Z Parallel Architectures and Their Development on the Basis of Intelligent Solving Machines // Proc. Int. Conf. on Parallel Computing in Electrical Engineering. — Warsaw (Poland).- 2002.- P.21-26

Authors' Information

Valeriy N. Koval – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; email: icdepval@ln.ua

Volodymyr V. Savyak – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; email: Volodymyr.Savyak@ustar.kiev.ua

Ivan V. Sergienko – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine

Sergey G. Ryabchun – Ustar Corp. ; office 1, 16A, Dovzhenko str., Kiev, 03057, Ukraine; email: sr@ustar.ua

Anatoliy A. Yakuba – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova, 40, Kiev, 03680 MCP, Ukraine; email: ayacuba@voliacable.com

DISCOVERY OF NEW KNOWLEDGE IN STRUCTURAL-PREDICATE MODELS OF KNOWLEDGE

Valeriy N. Koval, Yuriy V. Kuk

***Abstract**: The effective mathematical method of finding new knowledge of structure of complex objects with required properties is developed. The method comprehensively takes into account the information on properties and relations of the primary objects, which are included in complex objects. It is based on measurement of distances between groups of predicates at their some interpretation. The optimum measure for measurement of these distances with the maximal discernibleness of different groups of predicates is constructed. The method is approved on the decision of a problem of discovery of the new compound possessing electrooptical properties.*

***Keywords**: new knowledge, predicates, measure, complex objects, primary objects, maximal discernibleness.*

Introduction

The given work is devoted to the further development of methods of practical extraction of knowledge from experimental data. Its purpose - development of an effective mathematical method of discovery of new knowledge of structure of the complex objects possessing those or other properties. Work is focused on the decision of the important applied problem - designing of structure of compound with the necessary properties.

In previous our works [1] - [2] for discovery of new knowledge in the form production rules the concept of a variable predicate which can accept set of values - so-called predicate constants - predicates in the standard sense and the concept of distance between predicates were used. These both concepts have received the further development in the present work. However as against the above mentioned works in given article predicates with the subject domains consisting of objects, having internal structure are considered, that is objects from subject domains of predicates are assumed complex while earlier they were considered integral. Components of complex object we shall name primary objects [3], and the predicates designating properties and the relations of primary objects, we shall name primary predicates. About properties and relations of the primary objects included of complex objects, as a rule, also some information, which should be used in procedures of discovery of new knowledge of structure of the complex objects possessing those or other properties, is known. Procedure of discovery of such knowledge suggested in work is based on measurement of distances between groups of properties and relations of primary objects, or in language of logic, between groups of predicates at their some interpretation. The measure entered by us in works [1] - [2] for measurement of a degree of affinity of predicates,

cannot be directly transferred on groups of predicates. Therefore, in the given work the special measure, optimum by criterion of the maximal discernibleness of different groups of predicates, for measurement of distances between them is entered.

1. Structural-predicate Model of Knowledge

It is conveniently knowledge of complex objects to represent in the form which we have named *structural - predicate model of knowledge*. It is the further generalization of structural - attributive model of knowledge [3]-[4]. Generalization will be, that their relations, and not just properties of objects are considered also. For example, the two-place predicate « a difference of temperatures of fusion of two substances more Δ » describes some relation between two objects.

The structural - predicate model of knowledge (SPMK) is four-layer columns of a pyramidal network which separate layers form its tops. For presentation on fig. 1 it is resulted SPMK, containing knowledge of properties of chemical compounds with various types of structures of a crystal lattice: such as LiCaAlF6 (L-structure of a lattice), such as Na2SiF6 (N-structure of a lattice), such as Trirutile (T-structure of a lattice). We shall designate P, A, S, V the following sets of tops SPMK. The first layer P corresponds to predicate constants (values of variable predicates), designating properties and relations of primary objects. Elements P we shall name *primary predicates*. On fig. 1 primary variable predicates are: Tm - a melting point, So - standard $\mathrm{entropy}$ for corresponding simple oxides, H - standard $\mathrm{enthalpy}$ formations for corresponding simple oxides, Rs - radius of ions, C - an isobaric thermal capacity. On fig. 1 each of these predicates accepts on 2 values, and predicate constants corresponding to them are designated by numbers 1 and 2.

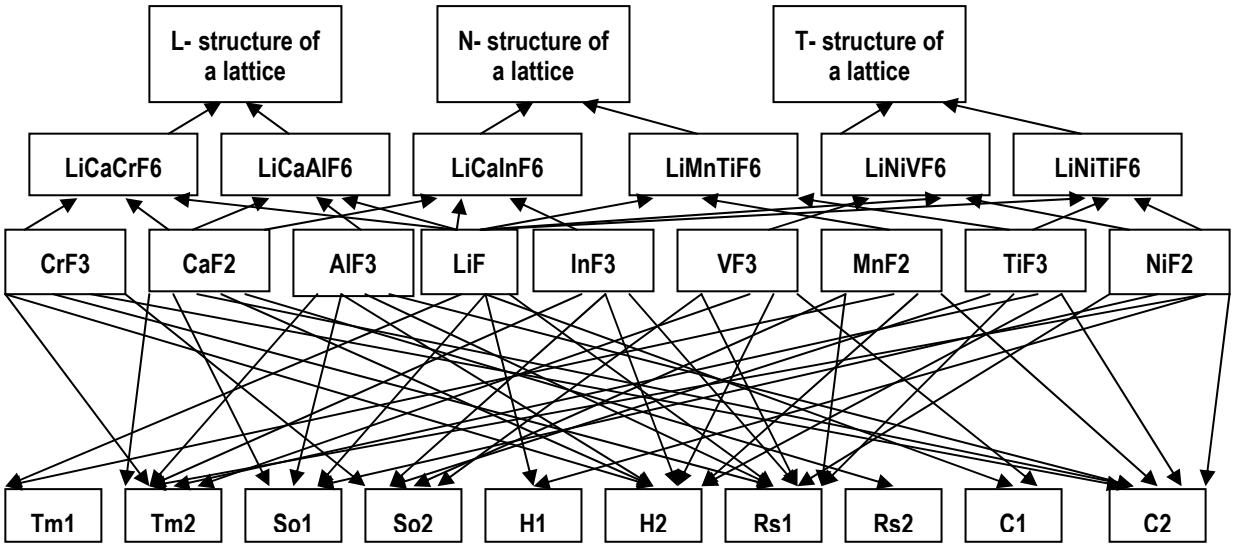


Fig. 1 the Example of structural - predicate model of knowledge

The second layer A corresponds to names of primary objects. They make subject domains of primary predicates at their interpretation. The third layer S corresponds to names of compound objects, the fourth V - to values of the variable predicates designating properties and the relations of compound objects. Elements V we shall name *predicates of compound objects*. Their subject domains are compound objects. On fig. 1 predicates of compound objects are 3 values of a variable predicate «to have the certain type of a crystal lattice». Arches of the bottom and top circles connect the tops representing objects, to the tops, representing predicate constants, and are directed from primary and compound objects to predicate constants. They are used at interpretation of predicates. Let ω designates multiplicity of some predicate constant. Then presence ω of the arches proceeding from ω objects and converging in the given predicate, corresponds to logic value of a predicate "true" at substitution of these objects in a predicate, and to value "lie" - at substitution of object in a predicate in a case absence of the arch, connecting given object with a predicate. Thus, at substitution of objects A and S from which to predicate constants arrows proceed from sets P and V, instead of arguments of these predicate

constants two sets of knowledge R_1 and R_2 in the form of true statements about properties and relations of primary and compound objects are formed. Arches of an average circle connect the tops corresponding to primary objects, to the tops representing compound. Primary elements, from which arches proceed, are part of those complex objects in which these arches comes to an end.

2. A Measure of Affinity of Groups of Predicates

Let's consider a problem of construction of a measure for measurement of a degree of affinity of groups of variable predicates. This measure should possess the following natural property: the distance between groups of the predicates, measured with its help, should not be equal to zero when these groups of predicates are various. From here follows, that it should possess property of the maximal discernibleness of different groups of predicates. We shall construct a measure satisfying this property.

Let's designate N - number of primary predicates in structural - predicate model, M - number of predicates of compound objects, $n(k)$ - number of the primary objects, which are included in complex object s_k. Symbols $p_1,\cdots p_N$ we shall designate primary variable predicates of model. In case of numerical values of variable predicates we shall adhere to the following rule: indexes for their predicate constants get out so that the order of their following corresponded to the order of following of numerical values of variable predicates. Thus, at dividing an interval of change of numerical values of predicates into segments (digitization), indexes of predicate constants, which correspond to them, should coincide with numbers of these segments.

Definition 1. *The label* x_{ik} *of a primary variable predicate* p_i *for complex object* s_k is understood as an index of that predicate constant of a predicate p_i which accepts logic value "True" at substitution in it instead of arguments of the primary objects included in s_k and connected by arches with this predicate constant.

Distribution of labels $x_k=(x_{1k},x_{2k},\cdots,x_{Nk})$ *of primary predicates for compound* s_k we shall name a vector which elements are labels for complex object s_k of all primary predicates which are included in structural - predicate model of knowledge. *Typical distribution of labels* for group of complex objects $G_1=\{s_1^{(1)},s_2^{(1)},\cdots,s_K^{(1)}\}$ we shall name a vector $h^{(1)}=(\bar{x}_1^{(1)},\bar{x}_2^{(1)},\cdots,\bar{x}_N^{(1)})$ which coordinates are equal to average values of components of vectors of labels in the group. We shall name a vector $\tilde{x}_k=(x_{1k}-\bar{x}_1^1,x_{2k}-\bar{x}_2^1,\cdots,x_{Nk}-\bar{x}_N^1)$ *a centralized vector of labels of primary predicates for the complex object* s_k belonging to group of complex objects $G_1=\{s_1^{(1)},s_2^{(1)},\cdots,s_K^{(1)}\}$.

Distributions of labels of primary predicates for some group of complex objects represent set of points in space R_N. If there are two groups of complex objects, we shall receive two such sets of points, which are mixed among them in a random way. There is a following problem: it is required to find such characteristics of these sets that it was possible to measure a degree of affinity of groups with their help. The following decision arises: to take as characteristics of each of sets of points corresponding points with the average coordinates which represent that other as typical distribution of labels for corresponding groups, and Euclidean length of distance between them to take as a measure of affinity for these groups. However, this decision is not optimum. Really, we shall consider the following example. Let points of distributions of labels of primary predicates for both groups are located on two perpendicular straight lines symmetrically concerning their center of crossing, and points of each group lay on a separate line. It is easy to see, that typical distributions of labels for both groups will coincide, and, hence, the distance between them is equal to zero though groups of predicates are various. It is possible to look at the above-stated not optimum decision from other point of view, allowing finding the optimum decision. We shall lead a direct line through two points which are in R_N typical distributions of labels of primary predicates for complex objects of both groups, and we shall project on it sets of points for both groups. It is easy to see that average values of projections for both sets of points, the so-called centers of projections $\bar{z}^1$ and $\bar{z}^2$ corresponding groups, coincide with the points representing typical distributions of labels. As the measure should possess property of the maximal discernibleness of different groups of predicates from here there is a following optimization problem: to construct in R_N the direct line c which is not necessarily taking place through typical

distributions labels, such that the distance center to center projections of both sets of points was maximal. The criterion of optimization of the given problem looks like: $\bar{z}^1 - \bar{z}^2 \to \max$. The distance $\bar{z}^1 - \bar{z}^2$ received as a result of optimization should be taken as a measure of affinity of complex objects for both groups.

3. The Mathematical Apparatus for Constructing the Measure

From the previous section follows, that for finding the optimum decision of a problem of construction of a measure for measurement of a degree of affinity of groups of variable predicates it is necessary to construct some auxiliary direct line c in space R_N and to project distributions of labels for both groups of predicates on it. In result, we shall receive two crossed sets of points on a direct line. As the choice of a direction of a line c influences distances between projections of distributions of labels and, hence, on their affinity the direct line should be chosen so that projections of distributions of labels from different groups of complex objects would be removed from each other so far as far as it is possible. Such choice of a direction of a line will allow distinguishing different groups of complex objects in the optimum way. A direct line c, on which distributions of labels of primary predicates for complex objects are projected, we shall name *a projective* line.

Let's result without the proof a number of auxiliary statements necessary for finding the required projective line.

Lemma 1. The projection of distributions of labels $x_k = (x_{1k}, x_{2k}, \cdots, x_{Nk})$ of primary predicates for complex object s_k to the projective line c, which is taking place through the beginning of coordinates in space R_N, is defined by the formula $\Pr_c x_1 = c_1 x_{11} + c_2 x_{21} + \cdots + c_N x_{N1}$, where $(c_1, c_2, \cdots, c_N)$ - cosines of the corners formed by a straight line with axes of coordinates.

Definition 2. *Disorder* concerning any point z of projections of distributions of labels of primary predicates for group of complex objects $G_1 = \left\{s_1^{(1)}, s_2^{(1)}, \cdots, s_K^{(1)}\right\}$ we shall name *total distance* $D_1(z)$.

$D_1(z) = \sum_{\nu=1}^{K} \left\| z_\nu^{(1)} - z \right\|$, where $z_1^{(1)} = c_1 x_{11}^{(1)} + \cdots + c_N x_{N1}^{(1)}, \ldots, z_K^{(1)} = c_1 x_{1K}^{(1)} + \cdots + c_N x_{NK}^{(1)}$.

The center of projections for group of complex objects $G_1 = \left\{s_1^{(1)}, s_2^{(1)}, \cdots, s_K^{(1)}\right\}$ we shall name average value of projections of distributions of labels of primary predicates for group G_1: $\bar{z}^{(1)} = \frac{1}{K} \sum_{\nu=1}^{K} z_\nu^{(1)}$.

Lemma 2. *The disorder* concerning any point z of projections of distributions of labels of primary predicates for group of complex objects $G_1 = \left\{s_1^{(1)}, s_2^{(1)}, \cdots, s_K^{(1)}\right\}$ is minimal, when z is equal to the center of their projections: $z = \bar{z}^{(1)}$, thus $D_1(z) = D_1(\bar{z}) = \sum_{\nu=1}^{K} (z_\nu^{(1)} - \bar{z}^{(1)})^2$.

Let $G_1 = \left\{s_1^{(1)}, s_2^{(1)}, \cdots, s_K^{(1)}\right\}$ and $G_2 = \left\{s_1^{(2)}, s_2^{(2)}, \cdots, s_L^{(2)}\right\}$ - two groups of the complex objects consisting accordingly from K and L complex objects. For each complex object of these groups, we shall construct on the basis of structural - predicate model of knowledge distribution of labels of its primary predicates. We shall receive $K + L$ vectors, which in space R_N will be displayed by two sets of vectors: X_1 - set of vectors $x_1^{(1)}$ $x_2^{(1)}$, ..., $x_K^{(1)}$ and X_2 - set of vectors $x_1^{(2)}$ $x_2^{(2)}$, ... $x_L^{(2)}$. We shall project sets X_1 and X_2 on a projective line c. On the basis of a lemma 1, we shall find values of projections:

$z_1^{(1)} = \Pr_c x_1^{(1)} = c_1 x_{11}^{(1)} + c_2 x_{21}^{(1)} + \cdots + c_N x_{N1}^{(1)}, \ldots, z_1^{(2)} = \Pr_c x_1^{(2)} = c_1 x_{11}^{(2)} + c_2 x_{21}^{(2)} + \cdots + c_N x_{N1}^{(2)},$

$z_2^{(2)} = \Pr_c x_2^{(2)} = c_1 x_{12}^{(2)} + c_2 x_{22}^{(2)} + \cdots + c_N x_{N2}^{(2)}, \ldots z_L^{(2)} = \Pr_c x_L^{(2)} = c_1 x_{1L}^{(2)} + c_2 x_{2L}^{(2)} + \cdots + c_N x_{NL}^{(2)}.$

Let's designate sets of projections X_1 and X_2 accordingly Z_1 and Z_2, and their centers:

$$\bar{z}^{(1)} = \frac{1}{K}(z_1^{(1)} + z_2^{(1)} + \cdots + z_K^{(1)}),\ \bar{z}^{(2)} = \frac{1}{L}(z_1^{(2)} + z_2^{(2)} + \cdots + z_L^{(2)}).$$

Definition 3. Disorder concerning an any point z of projections of distributions of labels of primary predicates for the pooled group of complex objects $G = G_1 \bigcup G_2$ we shall name *the general disorder* of both groups $D(z)$.

It is obvious, that it is equal $D(z) = \sum_{\nu=1}^{K} \left\| z_\nu^{(1)} - z \right\| + \sum_{\nu=1}^{L} \left\| z_\nu^{(2)} - z \right\|$. *The general center* of the pooled set of projections $Z = Z_1 \bigcup Z_2$ we shall name size $\bar{z} = \frac{1}{K+L}(z_1^{(1)} + \cdots + z_K^{(1)} + z_1^{(2)} + \cdots + z_L^{(2)})$.

Let's result without the proof the following theorem about the disorder of projections of distributions of labels of primary predicates.

Theorem 1. The general disorder $\overline{D} = \overline{D}(\bar{z})$ concerning the general center $\bar{z}$ of projections of distributions of labels of primary predicates of the pooled group of complex objects $G = G_1 \bigcup G_2$ is calculated under the formula $\overline{D} = \overline{D}_1 + \overline{D}_2 + \hat{D}_1 + \hat{D}_2$, where

$$\overline{D}_1 = \sum_{\nu=1}^{K} (z_\nu^{(1)} - \bar{z}^{(1)})^2,\ \overline{D}_2 = \sum_{\nu=1}^{L} (z_\nu^{(2)} - \bar{z}^{(2)})^2,\ \hat{D}_1 = K(\bar{z}^1 - \bar{z})^2,\ \hat{D}_2 = L(\bar{z}^{(2)} - \bar{z})^2.$$

From the theorem 1 follows that to maximize a measure of discernibleness of both groups of predicates - distance $\bar{z}^1 - \bar{z}^2$ it is necessary to maximize the sum $\hat{D}_1 = K(\bar{z}^1 - \bar{z})^2$ and $\hat{D}_2 = L(\bar{z}^2 - \bar{z})^2$. We shall name the sum $D_1 + D_2$ *full discernibleness*.

We shall receive expressions for the full discernibleness and the sums of disorders of projections of distributions of labels of both groups of predicates, which are used at the further calculations.

Vector of a difference of typical distributions of labels for groups of complex objects G_1 also G_2 we shall designate $h = h^{(1)} - h^{(2)} == (\bar{x}_1^{(1)} - \bar{x}_1^{(2)}, \bar{x}_2^{(1)} - \bar{x}_2^{(2)}, \cdots, \bar{x}_N^{(1)} - \bar{x}_N^{(2)})$.

Let's construct a square matrix $H = h^T h$ where the top index T designates operation of transposing. Dimension H is equal $N \times N$. It looks like

$$H = \begin{pmatrix} (\bar{x}_1^{(1)} - \bar{x}_1^{(2)})^2 & (\bar{x}_1^{(1)} - \bar{x}_1^{(2)})(\bar{x}_2^{(1)} - \bar{x}_2^{(2)}) & \cdots & (\bar{x}_1^{(1)} - \bar{x}_1^{(2)})(\bar{x}_N^{(1)} - \bar{x}_N^{(2)}) \\ (\bar{x}_2^{(1)} - \bar{x}_2^{(2)})(\bar{x}_1^{(1)} - \bar{x}_1^{(2)}) & (\bar{x}_2^{(1)} - \bar{x}_2^{(2)})^2 & \cdots & (\bar{x}_2^{(1)} - \bar{x}_2^{(2)})(\bar{x}_N^{(1)} - \bar{x}_N^{(2)}) \\ \cdots & \cdots & \cdots & \cdots \\ (\bar{x}_N^{(1)} - \bar{x}_N^{(2)})(\bar{x}_1^{(1)} - \bar{x}_1^{(2)}) & (\bar{x}_N^{(1)} - \bar{x}_N^{(2)})(\bar{x}_2^{(1)} - \bar{x}_2^{(2)}) & \cdots & (\bar{x}_N^{(1)} - \bar{x}_N^{(2)})^2 \end{pmatrix}.$$

Let's consider a matrix H' with elements $h'(\nu, \mu) = \frac{KL}{K+L} h(\nu, \mu)$ and designate $A^{(1)}$ and $A^{(2)}$ matrixes which columns will consist of components *centralized* distributions of labels of primary predicates for corresponding groups of complex objects. They contain N lines and accordingly K and L columns.

$$A^{(1)} = \begin{pmatrix} x_{11}^{(1)} - \bar{x}_1^{(1)} & \cdots & x_{1K}^{(1)} - \bar{x}_K^{(1)} \\ \cdots & \cdots & \cdots \\ x_{N1}^1 - \bar{x}_1^1 & \cdots & x_{NK}^{(1)} - \bar{x}_K^{(1)} \end{pmatrix},\ A^{(2)} = \begin{pmatrix} x_{11}^{(2)} - \bar{x}_1^{(2)} & \cdots & x_{1L}^{(2)} - \bar{x}_{lK}^{(2)} \\ \cdots & \cdots & \cdots \\ x_{N1}^{(2)} - \bar{x}_1^{(2)} & \cdots & x_{NL}^{(2)} - \bar{x}_L^{(2)} \end{pmatrix}$$

Let's consider matrixes $B^{(1)} = A^{(1)} A^{(1)^T}$ and $B^{(2)} = A^{(2)} A^{(2)^T}$. They look like:

$$B^{(1)} = \begin{pmatrix} \sum_{\nu=1}^{K}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})^2 & \sum_{\nu=1}^{K}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})(x_{2\nu}^{(1)} - \bar{x}_2^{(1)}) & \cdots & \sum_{\nu=1}^{K}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})(x_{N\nu}^{(1)} - \bar{x}_N^{(1)}) \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{\nu=1}^{K}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})(x_{1\nu}^{(1)} - \bar{x}_1^{(1)}) & \sum_{\nu=1}^{K}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})(x_{2\nu}^{(1)} - \bar{x}_2^{(1)}) & \cdots & \sum_{\nu=1}^{K}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})^2 \end{pmatrix},$$

$$B^{(2)} = \begin{pmatrix} \sum_{\nu=1}^{L}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})^2 & \sum_{\nu=1}^{L}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})(x_{2\nu}^{(1)} - \bar{x}_2^{(1)}) & \cdots & \sum_{\nu=1}^{L}(x_{1\nu}^{(1)} - \bar{x}_1^{(1)})(x_{N\nu}^{(1)} - \bar{x}_N^{(1)}) \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{\nu=1}^{L}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})(x_{1\nu}^{(1)} - \bar{x}_1^{(1)}) & \sum_{\nu=1}^{L}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})(x_{2\nu}^{(1)} - \bar{x}_2^{(1)}) & \cdots & \sum_{\nu=1}^{L}(x_{N\nu}^{(1)} - \bar{x}_N^{(1)})^2 \end{pmatrix}$$

In the following theorems, the formulas for calculation of the full discernibleness and the sum $\overline{D}_1 + \overline{D}_2$ of disorders are resulted. The theorems we shall result without the proof.

Theorem 2. The full discernibleness is equal

$$\hat{D}_1 + \hat{D}_2 = \frac{KL}{K+L}\sum_{\nu=1}^{N}\sum_{\mu=1}^{N} c_\nu c_\mu h(\nu,\mu),$$

where $h(\nu,\mu)$ - the element of a matrix H which is taking place in a ν line and in a μ column.

Theorem 3. The sum of disorders of projections for distributions of labels of primary predicates $\overline{D}_1 + \overline{D}_2$ is equal $\overline{D}_1 + \overline{D}_2 = \sum_{\nu=1}^{N}\sum_{\mu=1}^{N} c_\nu c_\mu b(\nu,\mu)$, where $b(\nu,\mu)$ - elements of a matrix $B = B^{(1)} + B^{(2)}$.

We shall result without the proof the basic theorem, which allows distinguishing groups of predicates thus that full discernibleness was so big as far as, it is possible.

Theorem 4. The full discernibleness $\hat{D}_1 + \hat{D}_2$ reaches the maximum at the fixed value of the sum of disorders of groups when values cosines $(c_1, c_2, \cdots, c_N)$ of the corners formed by the projective line with axes of coordinates are the components of an eigen vector W for a nonzero eigen value of a matrix $B^{-1}H'$.

6. Stages of Designing Structure of Complex Objects

Let's consider stages of the decision of a problem on designing structure of compounds with the set properties [3]. For presentation with this decision we shall accompany an example based on the data, resulted in [4]. Let it is required to design the new compounds possessing electrooptical properties. It is known, that electrooptical property crystals of fluorides, which have crystal structures of types LiCaAlF6 and Na2SiF6 possess, and the crystals of fluorides having structure such as Trirutile by such properties do not possess. For simplicity, we shall be limited to consideration of compounds with structures such as Na2SiF6 and Trirutile. At the first design stage SPMK, describing properties of compounds of fluorides, is under construction. The fragment of such model is resulted on fig. 1. At the second stage in SPMK the set of predicate constants of compound objects – V^+ to which there correspond required properties of projected compound, and set of predicate constants to which there correspond undesirable properties of projected compound is allocated – V^-. In our example V^+ is a predicate «to have structure such as Na2SiF6», and V^- is a predicate «to have structure such as Trirutile». At the third stage in SPMK the set of tops of the group G_1 corresponding to known compounds which have connections with predicates of set V^+ is allocated, and have no connections with predicates of set V^-, and set of the tops G_2 corresponding to known compounds which have connections with predicates of set V^-, and have no connections with predicates of set V^+. Let the first group included 10 compounds which are resulted in the first column of the table 1, and the second group included 17 compounds resulted in the first column of table 2.

Table 1

LiMgAlF6	MgF2	AlF3	1536	13,68	268,7	0,72	14,72	1545	15,8	361	0,39	17,95
LiMnAlF6	MnF2	AlF3	1133	22,25	202,4	0,83	16,24	1545	15,8	361	0,39	17,95
LiCaInF6	CaF2	InF3	1691	16,36	291,8	1	16,02	1445	33,5	250	0,8	15,93
LiMnTiF6	MnF2	TiF3	1133	22,25	202,4	0,83	16,24	1500	21,1	342	0,67	15,93
LiMnVF6	MnF2	VF3	1133	22,25	202,4	0,83	16,24	1679	23,1	271	0,64	21,62
LiMnCrF6	MnF2	CrF3	1133	22,25	202,4	0,83	16,24	1677	22,5	277	0,61	18,82
LiMnRhlF6	MnF2	RhF3	1133	22,25	202,4	0,83	16,24	1460	26	175	0,66	15,93
LiFeGaF6	FeF2	GaF3	1375	20,79	158	0,78	16,28	1225	28	255	0,62	15,93
LiCoInF6	CoF2	InF3	1400	19,59	159,1	0,745	16,44	1445	33,5	250	0,8	15,93
LiNiInF6	NiF2	InF3	1430	17,6	157,2	0,69	15,31	1445	33,5	250	0,8	15,93
		h1	1309	19,92	204,7	0,808	15,99	1496	25,30	279,2	0,64	17,33

Table 2

LiMgCrF6	MgF2	CrF3	1536	13,68	268,7	0,72	14,72	1677	22,5	277	0,615	18,82
LiMgGaF6	MgF2	GaF3	1536	13,68	268,7	0,72	14,72	1225	28	255	0,62	15,93
LiMgRhF6	MgF2	Rh3	1536	13,68	268,7	0,72	14,72	1460	26	175	0,665	15,93
LiNiTiF6	NiF2	TiF3	1430	17,6	157,2	0,69	15,31	1500	21,1	342,2	0,67	15,93
LiNiVF6	NiF2	VF3	1430	17,6	157,2	0,69	15,31	1679	23,1	271	0,64	21,62
LiCoCrF6	CoF2	CrF3	1400	19,59	159,1	0,745	16,44	1677	22,5	277	0,615	18,82
LiCuCrF6	CuF2	CrF3	1043	16,4	128,5	0,73	16,8	1677	22,5	277	0,615	18,82
LiZnCrF6	ZnF2	CrF3	1148	17,61	183	0,74	15,69	1677	22,5	277	0,615	18,82
LiNiFeF6	NiF2	FeF3	1430	17,6	157,2	0,69	15,31	1300	25	239	0,645	15,93
LiNiCoF6	NiF2	CoF3	1430	17,6	157,2	0,69	15,31	1230	27	187,2	0,61	15,93
LiZnCoF6	ZnF2	CoF3	1148	17,61	183	0,74	15,69	1230	27	187,2	0,61	15,93
LiCoGaF6	CoF2	GaF3	1400	19,59	159,1	0,745	16,44	1225	28	255	0,62	15,93
LiNiGaF6	NiF2	GaF3	1430	17,6	157,2	0,69	15,31	1225	28	255	0,62	15,93
LiCuRhF6	CuF2	RhF3	1043	16,4	128,5	0,73	16,8	1460	26	175	0,665	15,93
LiZnRhF6	ZnF2	RhF3	1148	17,61	183	0,74	15,69	1460	26	175	0,665	15,93
LiMgVF6	MgF2	VF3	1536	13,68	268,7	0,72	14,72	1679	23,1	271	0,64	21,62
LiFeCrF6	FeF2	CrF3	1375	20,79	158	0,78	16,28	1677	22,5	277	0,615	18,82
		h2	1352,8	16,96	184,9	0,722	15,60	1474	24,7	245,4	0,63	17,45

At the fourth stage in SPMK the set of the tops corresponding to primary objects for compounds of groups G_1 and G_2, and also set of the tops corresponding to primary predicates to which arrows from these primary objects approach is allocated. Primary objects are submitted in 2-nd and 3-rd columns in tables 1 and 2. Primary object LiF is included into all compounds, therefore its primary properties do not influence a belonging of compound to this or that group and consequently it further is not taken into account. At the fifth stage there are distributions of labels of primary predicates for groups of compounds G_1 and G_2. Each of primary variable predicates accepts accounting set of values - predicate constants. As their labels numerical values of properties of primary objects, which correspond to them were accepted. In an example the following of 5 primary variable predicates were considered: Tm - a melting point, So - standard entropy for corresponding simple oxides, H - standard enthalpy formations for corresponding simple oxides, Rs - radius of ions, C - an isobaric thermal capacity. Their values are submitted in 4-8 columns for a primary element of 2-nd column and at 9-13 columns for a primary element of 3-rd column, thus labels of predicates get out so that they coincided with these values.

At the sixth stage typical distributions of labels h_1 and h_2 are calculated by averaging values in columns 4-13 each tables. h_1 and h_2 are submitted in last lines of tab. 1 and 2. Further are centralized distributions of labels by subtraction of the received average values of each column from actual values of their cells. In result in numerical cells of both tables we shall receive values of the transposed matrixes $A^{(1)T}$ and $A^{(2)T}$. At the seventh design stage matrixes $B^{(1)} = A^{(1)}A^{(1)^T}$ $B^{(2)} = A^{(2)}A^{(2)^T}$ $B = B^{(1)} + B^{(2)}$, a return matrix B^{-1},

and matrixes $H = (h^{(1)} - h^{(2)})^T (h^{(1)} - h^{(2)})$ and $H' = H * \frac{KL}{K+L}$, where $K = 10, L = 17$ are calculated. At the eighth stage, there is an eigen vector W for a nonzero eigen value of a matrix $B^{-1}H'$ (for example, with the help of program Matlab). For a considered example an eigen vector looks like $W = (c_1, c_2, \cdots, c_{10}) =$ (-0.0002 0.0359 0.0017 0.6283 -0.0276 0.0004 0.0363 0.0015 -0.7754 -0.0242). Cosines of the corners formed by an optimum projective line c with coordinate corners are proportional to values of this vector; thus, the coefficient of proportionality does not play any role. At the ninth stage, there are projections of typical distributions of labels to this line: $\bar{z}^{(1)} = W * h_1^T$ and $\bar{z}^{(2)} = W * h_2^T$. We have $\bar{z}^{(1)} = 1.889$, $\bar{z}^{(2)} = 1.6201$. Their common center is equal $\bar{z} = 0.5(\bar{z}^{(1)} - \bar{z}^{(2)}) = 1.7545$. At the tenth stage gets out in SPMK primary objects for projected compound as follows. The objects having connections with primary predicates with which have also connections primary objects of group of compounds G_1 get out, and there are no connections with primary predicates with which have connections primary objects of group of compounds G_2, thus possible restrictions on structure of compounds are taken into account. We shall assume that the compounds submitted in table 3 have been chosen.

Table 3

LiMgInF6	MgF2	InF3	1536	13,68	268,7	0,72	14,72	1445	33,5	250	0,8	15,93
LiMnFeF6	MnF2	FeF3	1133	22,25	202,4	0,83	16,24	1300	25	239	0,645	15,93
LiMnGaF6	MnF2	GaF3	1133	22,25	202,4	0,83	16,24	1225	28	255	0,62	15,93
LiMnInF6	MnF2	InF3	1133	22,25	202,4	0,83	16,24	1445	33,5	250	0,8	15,93
LiZnInF6	ZnF2	InF3	1148	17,61	183	0,74	15,69	1445	33,5	250	0,8	15,93
LiCdInF6	CdF2	InF3	1345	20	167,4	0,95	15,93	1445	33,5	250	0,8	15,93
LiMgTiF6	MgF2	TiF3	1536	13,68	268,7	0,72	14,72	1500	21,1	342,2	0,67	15,93
LiMgFeF6	MgF2	FeF3	1536	13,68	268,7	0,72	14,72	1300	25	239	0,645	15,93
LiMgCoF6	MgF2	CoF3	1536	13,68	268,7	0,72	14,72	1230	27	187,2	0,61	15,93
LiFeTiF6	FeF2	TiF3	1375	20,79	158	0,78	16,28	1500	21,1	342,2	0,67	15,93
LiCoTiF6	CoF2	TiF3	1400	19,59	159,1	0,745	16,44	1500	21,1	342,2	0,67	15,93
LiZnTiF6	ZnF2	TiF3	1148	17,61	183	0,74	15,69	1500	21,1	342,2	0,67	15,93
LiZnVF6	ZnF2	VF3	1148	17,61	183	0,74	15,69	1679	23,18	271	0,64	21,62
LiNiCrF6	NiF2	CrF3	1430	17,6	157,2	0,69	15,31	1677	22,5	277	0,615	18,82
LiFeFeF6	FeF2	FeF3	1375	20,79	158	0,78	16,28	1300	25	239	0,645	15,93
LiCoFeF6	CoF2	FeF3	1400	19,59	159,1	0,745	16,44	1300	25	239	0,645	15,93
LiCuFeF6	CuF2	FeF3	1043	16,4	128,5	0,73	16,8	1300	25	239	0,645	15,93
LiZnFeF6	ZnF2	FeF3	1148	17,61	183	0,74	15,69	1300	25	239	0,645	15,93
LiCuCoF6	CuF2	CoF3	1043	16,4	128,5	0,73	16,8	1230	27	187,2	0,61	15,93
LiCoRhF6	CoF2	RhF3	1400	19,59	159,1	0,745	16,44	1460	26	175	0,665	15,93
LiNiRhF6	NiF2	RhF3	1430	17,6	157,2	0,69	15,31	1460	26	175	0,665	15,93
LiCuGaF6	CuF2	GaF3	1043	16,4	128,5	0,73	16,8	1225	28	255	0,62	15,93

For check of correctness of a choice, the projection $z = c_1 x_1^{(3)} + c_2 x_2^{(3)} + \cdots + c_N x_N^{(3)}$ of distribution of labels of each chosen connection to a projective straight line is calculated. If $\left|\bar{z}^{(1)} - z\right| < \left|\bar{z}^{(2)} - z\right|$, than the choice is considered correct. For compounds of table 3 consistently from top to down we find: z equally 1.8395, 1.9060, 2.0089, 2.1339, 1.8838, 2.0870, 1.6272, 1.6115, 1.5784, 1.7448, 1.6329, 1.6715, 1.6521, 1.6135, 1.7291, 1.6173, 1.5107, 1.6558, 1.4776, 1.5440, 1.4934, 1.6136. As $\bar{z}^{(1)} = 1.889$, $\bar{z}^{(2)} = 1.6201$ only the first 6 compounds according to the given technique are chosen correctly, and the others it is erroneous. The considered example allows to check up also correctness of the technique as the structure of a lattice of compounds of table 3 is beforehand known: the first 6 compounds have structure of a crystal lattice such as Na2SiF6, and all subsequent chemical compounds have structure of a crystal lattice such as Trirutile. Thus, we receive 100 % of right answers that proves a technique while in work [4] 86,4 % of right answers for the same group of chemical compounds are received.

Conclusion

In work complex objects with internal structure are considered. The structural - predicate model of knowledge, which is generalization of structural - attributive model of knowledge is offered. In work the method of reception of new knowledge of structure of complex objects with required properties which is based on measurement of distances between groups of predicates at their some interpretation is developed. The optimum measure for measurement of these distances with the maximal discernability of different groups of predicates is constructed. Stages of the decision of a problem designing of complex objects are considered.

Bibliography

1. V.N. Koval, Yu.V. Kuk. Distances between predicates in by-analogy reasoning systems, “Information Theories and Applications”, International Journal, vol. 10, N 1, p. 15-22, Sofia, 2003.
2. V.N. Koval, Yu.V. Kuk. Finding Unknown Rules of an Environment by Intelligent Goal-Oriented Systems, “Information Theories and Applications”, International Journal, vol. 17, N 3, p. 127-138, Sofia, 2001.
3. Гладун В.П. Партнерство с компьютером. Человеко-машинные целеустремленные системы. – Киев: «Port-Royal», 2000. –128 с.
4. Величко В.Ю. Розв’язання дослідницьких задач в дискретних середовищах методами виведення за аналогією. – Киев: Кандидатская диссертация. – 2003. – 150 с.

Authors' Information

Valeriy Koval – Institute of Cybernetics, Head of Department, address: 03680, Kiev, Prospect Glushkova, 40, Ukraine; e-mail: icdepval@ln.ua

Yuriy Kuk - Institute of Cybernetics, senior scientific researcher, address: 03680, Kiev, Prospect Glushkova, 40, Ukraine; e-mail: vkyk@svitonline.com .

CLUSTER MANAGEMENT PROCESSES ORGANIZATION AND HANDLING

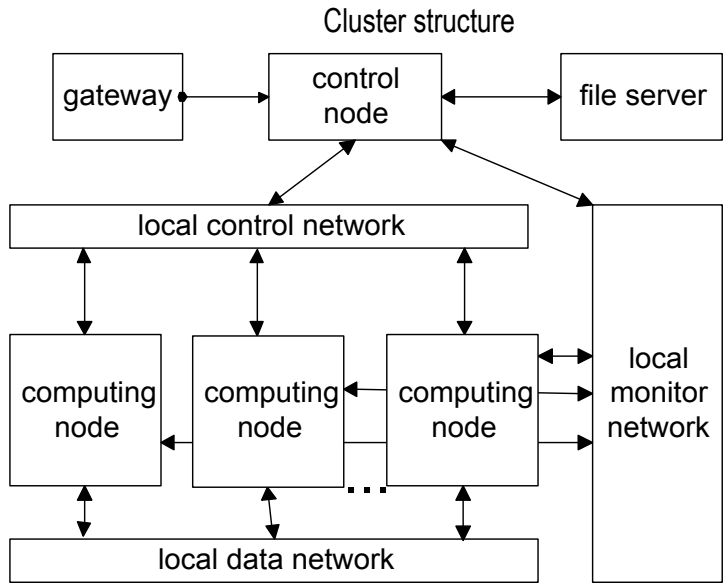
Valeriy Koval, Sergey Ryabchun, Volodymyr Savyak, Anatoliy Yakuba

Abstract*: he paper describes cluster management software and hardware of SCIT supercomputer clusters built in Glushkov Institute of Cybernetics NAS of Ukraine. The paper shows the performance results received on systems that were built and the specific means used to fulfil the goal of performance increase. It should be useful for those scientists and engineers that are practically engaged in a cluster supercomputer systems design, integration and services.*

Keywords*: cluster, computer system management, computer architecture.*

1. Cluster Complex Architecture

Basis cluster architecture is the array of servers (contains computing nodes and the control node), are connected among themselves by several local computer networks - a high-speed network of data exchange between computing nodes, a network of dynamic management of a server array and a network for cluster nodes monitoring. User access to cluster as a whole can cope by the access server - a gateway on which check of the rights of access of users to cluster and preliminary preparation of tasks for execution is realized. File services are given user tasks by a file server through the cluster control node. A file server in a system provides data access on file level protocols, like Network File System (NFS). A file server is connected directly to a local data network via high throughput channel. In some cases, the gateway and/or file server functions may be carried out on the control node.



Cluster computing node is a server, more often dual-processor, for direct execution of one user task in one-program mode. Computing nodes are dynamically united through a network in a resource for a specific task, simultaneously on cluster some problems may be executed, depending on amount of free computing nodes.

The control node of cluster is a server on which are carried out compilation of tasks, assignment of cluster resources (computing modules - cluster nodes, processors) to the user task, global management of processes activated on nodes during task execution, granting to task needed services of a file server.

2. Dynamic Management with Cluster Nodes

The role of the dynamic management is to manage access to computing nodes and to provide a dynamic reconfiguration of a system. Dynamic management of a cluster system is mostly determined by the used logical systems of a parallel programming (LSPP) (i.e. their architecture and communication libraries). But it can also be influenced by nodes interconnect architecture, rather, a data communication network (means to connect the cluster nodes among themselves and with cluster control node).

A basis of a dynamic cluster reconfiguration under a user task is defined by the list of cluster resources allocated to the task (nodes, processors). After the resources are reconfigured, the system provides a corresponding handling of a user task only within the framework of the appointed resources.

The element of this list of cluster resources is assigning to task the name of node and quantity of processors, which are active in the node. A node always is appointed entirely, whereas the request of a task always specifies necessary amount of processors.

The cluster resources handling system estimates real presence of resources and "collects" the number of processors necessary to a task from the pool of really active nodes at the moment of free nodes request. Processors are allocated always in the cluster node staff, i.e. it is impossible to allocate in one node on one processor to the different tasks, processors of node unused in a task always should stand idle.

In the cluster, where the communication network is based on the switch (Gigabit Ethernet, Infiniband), any of nodes accessible to a task can cope irrespective of other nodes in this configuration up to full restart. Mutual influence of cluster nodes upon serviceability of a communication network does not exist as a whole - it is provided with the switch.

For a network on basis SCI cards the opportunity of a direct handling of the cluster node within the framework of allocated cluster resources is sharply limited, as the communication network "rises" entirely and serviceability of separate node can depend on serviceability of connections with the next nodes essentially [1].

Though at application 2D-and 3D-topology, it is possible the dynamic change of routing that supposes detour short, but defective connection due to working, but longer, connections through other nodes. However if several

nodes die, then a general cluster performance is going down up to transition to a disabled condition. On the other hand, when using a central switch (which is not mirrored), the switch causes a death of all the system.

An opportunity of reconfiguration depends also on a usage of local disk memory of the node. For a cluster systems with a distributed storage based on a local node's hard drives there is a problem found with an execution of user tasks in a background batch mode. When a repeated return to a computing process for the task execution is required, it is necessary to receive the same cluster resources for a task that was provided in a previous stage of the task execution (it implicitly demands long reservation of disk resources on all cluster nodes, appointed to a task).

Reduction of negative influence of this restriction is possible only at refusal from the local disk resource for background tasks for the benefit of network file systems (for example NFS) or the general file systems oriented on cluster application (GFS) [2]. This allows do not care about granting the same cluster resources for the task being executed in a background batch mode.

After task is finished, all allocated resources should be returned in a pool of free resources. Rational use of this pool assumes a regular check of resources' state. The system diagnosis and makes a conclusion about an unavailable resources in an emergency configuration to exclude their incorrect usage. This part of a management system is one of the most important parts of all the cluster management software.

3. Management of Cluster Accessibility

There are several approaches known in a field of cluster resources access management. All of them are based on a standard user authentication on a stage of a system user login. After login is made there are following general ways possible:

1. A user receive an access to all cluster nodes, assigned as a resource to one's queued task, i.e. the task is executed on behalf of the user and a user has a full control over the behaviour of nodes, usage of node own resources (main memory, exchanges with a file server and other nodes, employment of the processor) is given to this user.
2. A user receives an access to an interface of a task status control and management of a task execution. Thus, a user has no real access to cluster nodes allocated.

At the first approach the list of users is exported to all cluster parts or the real system user is dynamically created for the period of a task execution. The control over access to the system variables, data and command files of cluster management, nodes essentially becomes complicated, as for communication network SCI this control should be more rigid, than for cluster on the basis of the switch. On the other hand, granting to the user the full access to node allows going to the manual management of task execution up to loading into local disk memory of a node. One of the examples of the mentioned approach implementation is MBC-1000M (Moscow) system [3].

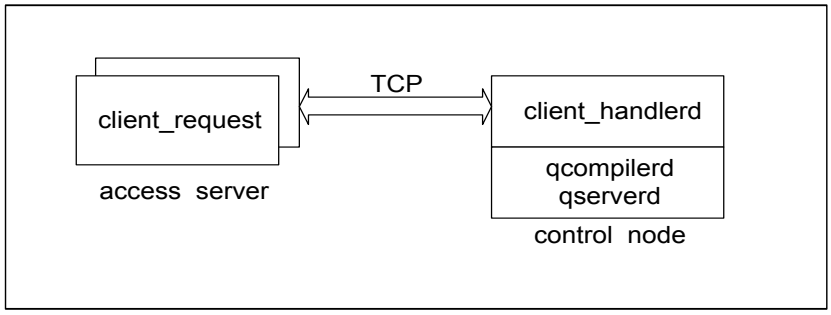
In our opinion, the second approach, despite of considerably big system costs on the organization and support of user work, is represented to more reliable in preservation of integrity of the system software, its functioning and cluster security from the non-authorized access. In this case all works on task execution on nodes are carried out by the specialized pseudo-users existed only on cluster nodes. On behalf of these pseudo-users, the task is executed. For integrity of the approach, an every LSPP has the unique specialized pseudo-user; i.e. the policy of safety does not permit a real user, except for repair managers, to log in into cluster nodes. Such a system provides greater security and reliability of a cluster.

Absence of direct access of the user to cluster nodes is compensated by presence of a specific user interface. An interface allows users to operate a task execution, task queues, to load the data for a task, to supervise a condition of the nodes, which are included in a resource of a task, etc. A program of the user interface cooperates with a demon started on the control node and carrying out all necessary user work. The cluster administrator has the possibility to execute any of these functions.

4. Task Processes Handling

Users, as with the remote access as taking place in a corporate local network, get access only to a gateway - access server, the last holds all user catalogues exported from a cluster file server and support user preparations of the tasks for execution. The subsystem of service of users and their tasks has client-server architecture: the

client part settles down on a gateway, the server part - on the control node, connection between these parts is organized under TCP- protocol.

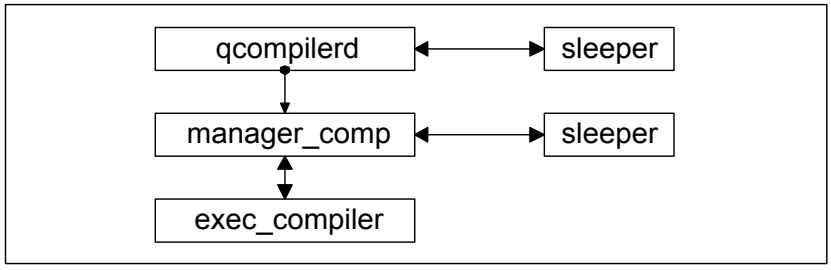


Requests from the user are transferred to the control node and executed by a demon **client_handlerd,** at this one control node can serve a little clusters with identical architecture. The demon **client_handlerd** carries out the requested action and returns result of performance to the user.

One of such actions is the definition of necessity to compile task and queue it up for compilation with the subsequent placing (at the absence of compiling mistakes) in the execution queue. Each of these queues is served by the demon, correspondingly, **qcompilerd** and **qserverd,** their activities on the control node; their Status may be change only by the cluster administrator. In the same way the user receives data about queued tasks, on cluster congestion, presence of free resources, etc.

The **qcompilerd** functions are:

- Search of a task (without the control of parameters of task execution);
- Creation of working structure where this task is compiled;
- Start of the compilation manager, monitoring the specified task, and return to search of other task to compile.

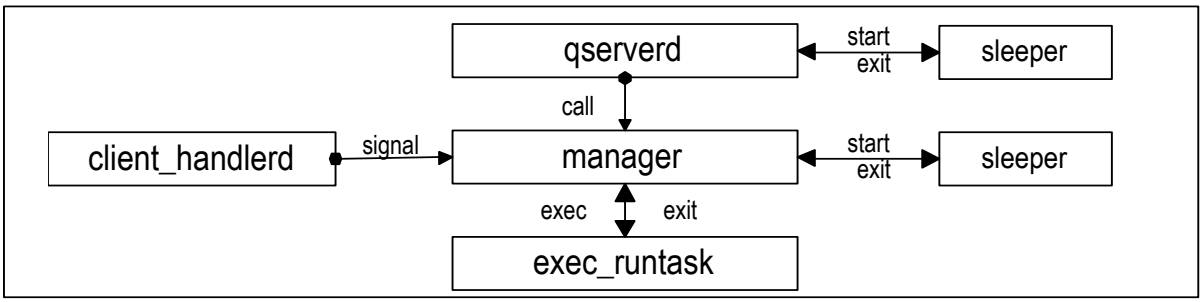


The manager of compilation, in turn, starts as independent process a command file of compilation in a mode *chroot*, expects the end of compilation and returns after that results of compilation in the user individual catalogue.

The **qserverd** functions are:

- Search of a task (with the control of parameters of task execution);
- Updating or creation of working structure of a task for execution;
- Assignment of resources to a task;
- Export of an environment, start of the manager of execution of a specific target and return to search another task for execution.

The manager of execution **(manager)**, in turn, starts as independent process the command file of execution **(exec_runtask)** in a mode *chroot*, expects the end of execution **exec_runtask** or the user signal about task execution stopping - through a demon **client_handlerd -** and returns after that results of execution to the user individual catalogue.



5. Cluster Management System (Base Functions)

Management system – cluster control facilities, used both the system administrator, and various software systems over the operating system, having for an object "continuous" monitoring of computing process, the equipment and the software. It contains, at least, three obligatory parts:

- A direct control of computing process and functioning of the cluster equipment;
- Management of service means of a task stream processing and user works with cluster;
- Monitoring cluster infrastructure (system of power supplies and cooling, a cluster configuration and availability of the cluster components through its communication networks).

The management system may be resident on one of control nodes with an opportunity to change this place to another, and may be distributed among them is depends only on the rules of functioning of managing means.

Obligatory functions of a management system are:

- Management of start, stop and restart all cluster equipment, and its separate nodes and also active means of the cluster system software, in particular, means of a task stream processing;
- Monitor service of the system administrator needs with results of the analysis of a cluster status, its configuration and availability of its nodes;
- Management reconfiguration of node connections if it allows the accepted circuit of a configuration;
- User authentication at its local or remote login to the cluster, support of its functioning during task preparation, granting of help services both online and offline;
- Support of service means of the user interface at compilation, assembly and task execution, under the control of intermediate results over long task execution, on preservation of results of the task running, maintenance of user tasks with services of a file server and DBMS on it;
- Support of a message exchange between the system administrator and users;
- Remote user maintenance with means **upload/download** to transfer the data between its local client computer and the cluster client individual catalogue.

6. Support of the User Computing Process Means

Cluster oriented tasks should use the communication libraries, more often implementing the MPI interface. In this interface the task will start on the zero allocated node with the indication of necessary processors quantity, names of a task code file and some other parameters. For example, **mpirun-np 16 /test/test2**, where **mpirun** - standard command for task start, **np** - required number of processors, **/test/test2** - a path to the task code file.

Implicitly in this start rights of the owner of the catalogue from which start is carried out, and rights of the owner of a code file are taken into account also. The coordination of these rights and maintenance of start correctness, and also a correctness of access to the data, dissipated upon cluster file system, are assigned to service means. Compilation of a task is made on behalf of the pseudo-user, representing chosen LSPP, on the control node without attraction of cluster resources with the subsequent transferring the compiled task to queue for execution on cluster nodes under the control of the same pseudo-user determined as the only thing for ordered LSPP.

Client-server means of user's interaction are included into means of support of the user computing process with control facilities tasks also. The accepted principle is the user alienation from executed tasks, that is client, placed in cluster environment, get access only to the gateway – access server physically separated by network addresses from the control node and other cluster nodes, and working areas of the tasks started on execution are placed on the control node. Functioning service means, **client-request** on the access server and **client_handlerd** on the control node, having established connection among them, support it activity till the moment of the termination of concrete user request.

The direct task start is connected to significant inconveniences by the rights of access. More effectively to add additional interfacing means to start the task on allocated cluster resources. These interfacing means should coordinate correctly rights of access during start, estimate and prepare for real use the list of cluster resources, check their sufficiency and, maybe, real availability. As unification of LSPP is absent, these means are individually adjusted on each type of LSPP through environment variables of execution PATH, LD_LIBRARY_PATH and specific ones for concrete LSPP.

Cluster tasks, as a matter of fact, are tasks with great volumes of calculations and consequently, the period of the maximal uninterrupted execution cannot be uncertain, that is why the monopolization of cluster as a whole or only some its parts under one task is incorrect, long on time of the task running should represent a chain of consecutive starts and breaks of the execution (i.e. a set of quantums to run the task), alternated by the idle periods waiting the reception of quantum. Service that means to support the execution of such tasks should provide a correctness of the termination of concrete quantum, preservation of the intermediate data and renewal the execution in the other quantum.

One more service means, facilitated work of users, may be the debugger of cluster tasks, it allows with cluster resources limited from above receiving reports as task executions on the concrete processor, as characteristics of data exchanges between cooperating processors. The attitude to such debuggers dual, rough debugging on them goes conveniently enough and naturally, exact debugging is usually connected to searches of opportunities of increase of task productivity, searches of memory "leakage" and adjustment of a task for the big number of processors, that just and cannot really be supported by noncommercial cluster debuggers.

7. System Means for Increasing the Cluster Performance

Among many means to improve the quality of cluster functioning, it is possible to discuss the basic:

- To carry out hardware improvements in a communication network of nodes, in particular, using network adapters SCI-technology instead of switch oriented Gigabit Ethernet, making up the connections on the basis of 2D-topology (or 3D-topology) and choosing the optimal variant of node switching (i.e., for 16-node cluster with processors Xeon only transition from the network based on switch with Gigabit Ethernet to a network based on SCI gives almost 30 % a gain of performance in Linpack test, and replacement of switching 2x8 nodes on switching 4x4 nodes gives a gain on 4-6 %).
- To maximize the using of node main memory due to exact selection of the used software. So, use only a necessary minimum of demons on node allows to achieve employment of all 12-16MB on the unloaded node.
- To use architecturally – optimized libraries and the compilers giving the most effective codes, in particular, Intel compilers for languages C and Fortran or family compilers GCC, use library MKL (Intel Math Kernel Labs) instead of library ATLAS.

Total results of consecutive changes for 16-node cluster with processors Xeon 2.66 GHz at 2 processors and main memory 1 GByte on node (that gives peak performance in 166 Gflops) are resulted in table 1.

The analysis of table 1 shows, that obligatory elements of cluster adjustment, needed for the maximal productivity, should be - "thin" adjustment of a node main memory for system using, installation, adjustment and use of the richest noncommercial libraries, even for rather weak communication network on Gigabit Ethernet. In case of replacement of switch oriented weak network by more powerful (in particular, by SCI as with Infiniband [4] we did not have experiments) yet it is necessary to choose rational configuration of data connections, recommended the vendor firms, and to use communication library Scali, instead of MPICH-SCI.

Table 1

Changes in structure and the system software	*The measured maximal performance in Linpack test (Gflops)*	*Ratio max/peak performance (%%)*
Initial configuration: Communication network =Gigabit Ethernet, Accessible MM = 0.83 GByte, Compiler = GNU, Library = ATLAS	**71**	**43**
Communication network =SCI, Switching = 2x8, Communication library = MPICH-SCI	**94**	**57**
Accessible MM = 0.99 Gbyte	**99**	**60**
Switching = 4x4	**104**	**63**
Library = MKL, Communication library = SCALI	**112**	**67**

One more factor influencing the common cluster performance is a rational choice of structure of file system. Generally, when installation of commercial OS Red Hat Cluster Suite which contains cluster oriented file system Global File System is not supposed, and there is a local system of a data storage based on a RAID-array in the structure of control node entering or served by the specialized server, and local disk memory on cluster nodes is absent, the most effective means may appear export of references to contents of a RAID-array to all points of the cluster where work with files is supposed. Thus even for the user individual catalogues which formally should be on a gateway – access server, their physical accommodation in disk memory of the gateway is not supposed, they only there are exported from a file server by the references. Similar by results of the decision can be offered for access to files in an executing task - despite of accommodation of the big data files in the individual catalogue of the user, direct access to which to the absolute address from node is impossible, and copying of data files in working structure of task execution is comprehensible only to the small sizes of files (for example, tens Mbytes), indirect addressing through tables of address transformation will provide access to the data of great volume without their moving to working task structures.

The reference to databases, which are stored in the same RAID-array, actually does not differ from described. Unfortunately, experiments in this direction just begin, as well as authentic results are absent.

Bibliography

1. http://www.scali.com
2. http://www.redhat.com/software/rha/gfs
3. http://parallel.ru/computers/reviews/MVS1000M.html (In Russian)
4. http://www.mellanox.com

Authors' Information

Valeriy N. Koval – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; email: icdepval@ln.ua

Sergey G. Ryabchun – Ustar Corp., office 1, 16A, Dovzhenko str., Kiev, 03057, Ukraine; email: sr@ustar.ua

Volodymyr V. Savyak – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; email: Volodymyr.Savyak@ustar.kiev.ua

Anatoliy A. Yakuba – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; email: ayacuba@voliacable.com

MULTI-AGENT USER BEHAVIOR MONITORING SYSTEM BASED ON AGLETS SDK[1]

Alexander Lobunets

***Abstract**: The paper describes an experience that was obtained during development of multi-agent system using Java and Aglets SDK. The user behavior monitoring system described in this paper utilizes a neural network classifier for user processes analysis. The overview of the neural classifier is out of the scope of this article. The main issues pointed in this paper include software technology evaluation, agent oriented patterns, usage of UML for software design and brief Aglet API overview. The monitoring system prototype is installed in local area network of IT Department of Space Research Institute NASU-NSAU, Ukraine.*

***Keywords**: neural network, multi-agent system, network security system, user behavior model, intrusion detection system, aglet development.*

Introduction

Nowadays computer user behavior monitoring is one of the important issues. As a result of agile development in the field of informational technologies, a computer's user becomes one of the centric objects for observing and monitoring. The analysis of user activity monitoring can be used in different application areas such as automatic user's environment adaptation, computer and network security, personnel observing, e-commerce, etc. Taking into account the urgency of information security issue a number of scientific efforts are focused on intrusion detection systems (IDS).

A number of innovative approaches and new models for network security assurance system have been proposed recently [Gorod, 2001]. The basic ideas are to make the IDE more intellectual in terms of attack detection and data processing by means of rule based networks, neural networks [CannMah], genetic algorithms, human like immunology systems, variable sized Markov chains [Sokol], etc and make usage of the particular distributed components of IDS cooperative. Thus, new previously unknown types of attacks compromising computer network will be detected by IDS. For such an idea, the multi-agent system model proves to be very promising [BGISZ].

It should mention there are two key elements during the process of designing a monitoring system: user behavior model and implementation technologies evaluation. The article [SKL, 2004] describes user model evaluation in details. In this paper, implementation issues are pointed.

Despite of existent extension for agent modelling [AUML] a row UML diagrams where used for notation during system analysis and design. This choice is based mostly on ASDK particularity and its incompatibility with world accepted standards such as FIPA [FIPA].

Agent-based User Behavior Monitoring System Architecture

The user behavior monitoring system architecture is based on a complex user's model [SK, 2004]. The mentioned model consists of two parts and considers both dynamic and static properties of user's behavior. Both parts of the model make use of neural networks for abnormality detection.

It is known that integrated intrusion detection system (IDS) should detect different known attacks and unknown as well. Thus, IDS should contain various autonomous interactive modules. To meet these requirements the architecture of such system is designed using agent approach (Fig. 1).

According to [SKL, 2004] the proposed system contains the following types of agents:

[1] The work is particially supported by the grant of President of Ukraine for the support of scientific researches by young scientists № Ф8/323, "Prototype of intelligent multiagent security system".

User agent. This is agent is used to detect anomalies in user activity which is carried out on the basis of neural network.

Host agent. Performs system calls processing and detects anomalies and known types of attacks. For example, it allows detecting „Trojan horse“ attacks.

Network agent. Operates at the firewall and analyze the network traffic. The information extracted from packets is used to detect known attacks and anomalies in the network. It is expected to be implemented with a help of neural networks and probability approaches.

Server agents. These agents are responsible for the server security.

Controller agent. This agent is responsible for anomalies' analysis and detection of distributed attacks in the scale of whole system.

Database. Contains data for different types of agents.

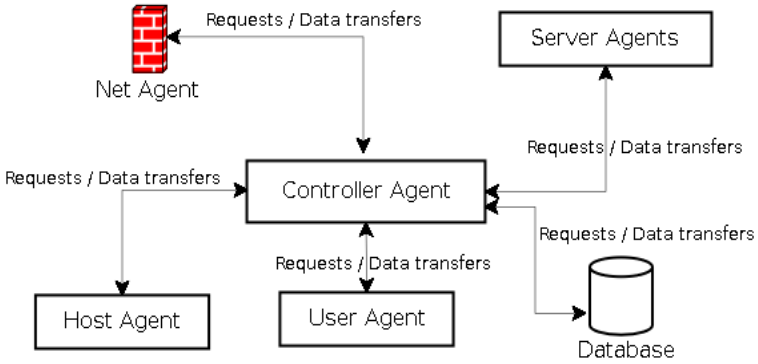


Fig. 1 – General system architecture

As a user logs on, Controller Agent instantiates corresponding User Agent. At the same time, User Agent obtains data about user behavior model. During user’s session agent performs monitoring of user's activity on the basis of neural network behavior model. At the same time, it aggregates data for further behavior model correction. When the session is finished, User Agent sends data to database. In case of anomaly detection, User Agent informs Controller Agent about suspicious activity.

Host Agents and Server Agents detect system anomalies and known attacks.

Implementation Technology Evaluation

Java and Aglets Software Development Kit (ASDK) were chosen for implementation of the monitoring system. Java as a programming language for agents makes it easier than ever for programmers to build complex agents and offers the set of unique features for developing multi-agent systems. It should mention such features of Java as platform independence, secure code execution, dynamic class loading, multi threading and object serialization.

ASDK is open source software, which was developed at the IBM Tokyo Research Laboratory and distributed later under the IBM Public License. Aglets is a Java mobile agent platform and library that eases the development of agent based applications. An aglet is a Java agent able to autonomously and spontaneously move from one host to another [ASDK]. ASDK includes both a complete agent platform with a standalone aglets server Tahiti and a Java library that allows development of mobile agents. Using ASDK developers can embed the aglets technology in their applications as well.

An aglet runs as a thread or multiple threads inside the context of a hosting Java application. When aglets travel across a network, they migrate from one computer running a hosting platform to another. Each aglet host owns a security manager that enforces restrictions on the activities of the untrusted aglets [Venner, 1997]. The migration is performed via uploading aglet's code through class loading mechanism. Basic Aglets API classes and interfaces are shown at the Fig. 1 below:

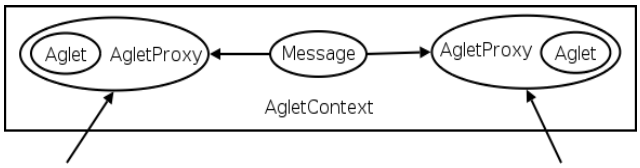


Fig. 1 – Basic classes and interfaces of the Aglets API

Referring to Fig.1 agent is represented by a Java class Aglet that interacts with environment via AgletProxy class for security reasons. AgletContext plays a role of a sandbox and a runtime environment for the aglet code.

An aglet has defined lifestyle (Fig. 2) and can experience the following events in its life [Oshima, 1998]: creation, cloning, dispatching, retraction, activation and deactivation, disposal and much more. Most of the activities involve either duplication, transmission across a network or persistent storage of aglet's state, which is carried out by one the mention above Java features – serialization.

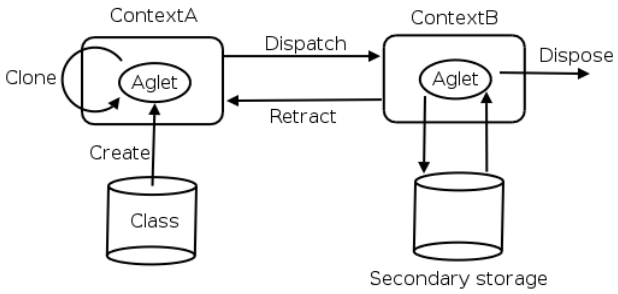


Fig. 2 – Aglet object life cycle

The callback-programming model allows developers to make an agent response to corresponding events in its lifestyle. Thus, it is required to override a few methods that will be called from an external entity (the aglets runtime environment) during the agent life [Ferrari, 2004]. A detailed documentation, related to aglet programming, is available at [ASDK].

Requirements and Use Cases

According to general architecture concepts proposed in [SKL, 2004] the system can be divided into two components. These components communicate with each other via local area network (LAN) with the help of mobile agents and messages (Fig. 3). The first component provides agent-hosting facilities for Client Node. It should be installed on every user's workstation in LAN. The second component represents a Server Node. It serves queries from client agents and is used as an agent code repository.

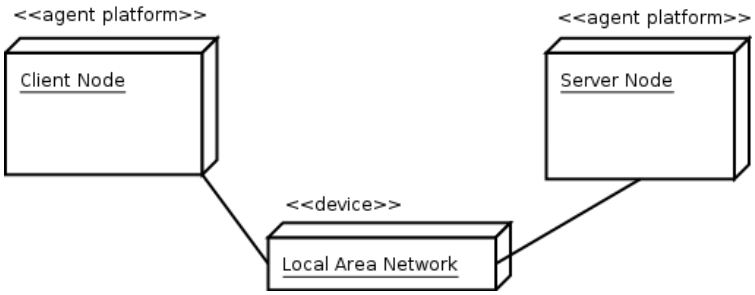


Fig. 3 – System components diagram

Due to the paradigm, which was used for the development of such a system, agents are the structural functional elements. Each agent operates in a context of a hosting platform such as Client or Server Nodes. Depending on concrete set of functions executed by an agent the following types of agents were defined in [SKL, 2004]: a user behavior agent (UserAgent), a coordinating agent (ControllerAgent).

The ControllerAgent is responsible for user registration, managing user related data storage such as process logs and neural network configuration, creating the Server Node agents and further communication with them. This agent is a Server Node resident (a static agent).

The UserAgent encapsulates a neural network model for a concrete user. The functions of this kind of agent are aggregation and transfer of process logs from user workstation to Server Node, user's processes analysis in a real time with the help of neural network. This agent is a mobile one and migrates to Client Node after its creation at Server Node.

Aglets development model constates certain limitations related to agent communication. Thus there were introduced additional types of agents such as: Watcher agent and Host agent (has a similar name as mentioned in architecture description but performs different operations).

The Watcher agent serves as intermediate agent in communication between Client and Server Nodes. In couple with UserAgent and HostAgent it implements a well-known Master-Slave pattern [LO, 1998]. In accordance with mentioned pattern, Watcher agent plays a role of master and creates its subordinates (slave agents). This agent is a Server Node resident (a static agent).

As for the HostAgent its function is to provide connection reliability facilities via system heart-beat messages. This agent instantiated by Watcher agent and dispatched to user's workstation. On arrival, HostAgent replies to a special type of message (if-alive) from its master. Thus, a Server Node knows that client workstation is reachable and current user is logged in. In case of three failed attempts to query HostAgent, Watcher agent considers this situation as user's logout, sends sign off message to ControllerAgent and disposes itself.

TrainAgent encapsulates a neural network training function. The agent activates on a schedule defined by ControllerAgent. After its initialization TrainAgent walks through user profiles and perform neural model correction for every user, operating system and workstation. The next time user signs in, an updated neural model will be loaded and used for analysis. This agent is a Server Node resident (a static agent).

One of the key components of the system is a logging application. For the cross-platform purposes, a custom process logging application was used. The log format is described below:

TIME | PROC_ID | PROC_NAME | STATUS, where

TIME – process registration time;

PROC_ID – process unique identifier (assigned by operating system);

PROC_NAME – name of a registered process;

STATUS – process status, accepts one of the following values STARTED or FINISHED.

The following packages were defined for system use cases in terms of UML :

- Controller package use cases
- Create user data storage
- Register user sign in/sign out
- Instantiation of Watcher agent
- Instantiation of TrainAgent
- Watcher package use cases
- UserAgent instantiation
- HostAgent instantiation
- Heart-beats generation
- Process logs aggregation and storage
- HostAgent packages use cases
- Heart-beats handling
- User sign out event notification
- UserAgent package use cases
- Launching logging application
- Transferring process logs to Server Node
- User activity analysis

- TrainAgent package use cases
- Get user processes list
- Perform neural network training

Client and Server Nodes represent separate packages with their own use cases. User Node use cases are defined as following:

- Create agent hosting environment
- User sign in event notification
- User sign out notification

Server Node use cases are listed below:

- ControllerAgent instantiation
- TrainAgent schedule set up

Programming Agents and Hosting Platform

After the detailed analysis of the use cases defined above the following class diagram can be drawn:

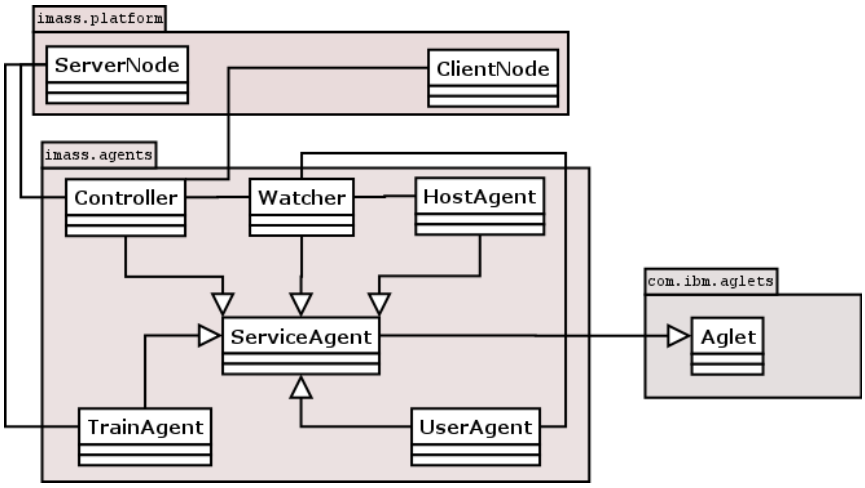


Fig. 4 – Monitoring system class diagram

Let's make a brief review of each class and package presented on Fig. 4 (Note that diagram was simplified for better understanding).

ServiceAgent is a derived class of Aglet class from Aglets package (com.ibm.aglets). It inherits all required methods for agent's life-cycle handling and defines additional properties and methods to be used by monitoring system agents. For example, a master-slave pattern was used to simplify the communication scenario. A slave agent needs to store master's AgletProxy object for sending messages. Thus, Service agent defines a new property masterProxy of AgletProxy data type.

As it was mentioned above, all system agents extend ServiceAgents functionality. Controller and Watcher agents implement master-slave pattern in the following way: a Controller creates Watcher agent, assigns masterProxy property the value pointing to its AgletProxy object and stores Watcher's AgletProxy object for further communication. The same pattern is implemented by Watcher agent and HostAgent in couple with UserAgent. In this relation Watcher plays a role of master – creates both agents, assigns masterProxy property and stores their AgentProxy objects.

The package imass.platform contains classes that implement agent-hosting platform since aglets exist only within AgletContext. The hosting platform performs the following operations [Oshima, 1998]:

- platform parameters setting
- AgletRuntime instance initialisation

- user authentication
- creating MAFAgentSystem_AgletsImpl instance
- factory components installation
- creating AgletContext instance
- creating ContextListener instance and adding it to the created context
- security manager installation
- context start up
- communication layer start up

A detailed instructions regarding hosting platform implementation can be found in [Ferrari, 2004] as well.

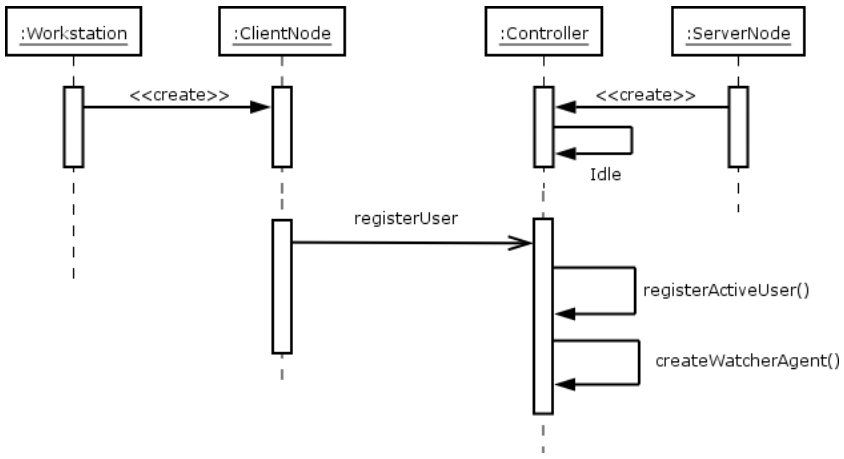


Fig. 5 – System initialization sequence diagram

Fig. 5 and Fig. 6 describe the interaction between system functional elements. A user "sign in" event initiates Client Node application creation. A ClientNode object obtains remote AgeltProxy object of the Controller aglet using its getRemoteProxy() method:

```
AgletProxy massControllerProxy = this.getRemoteProxy(massURL, massController);
if(massControllerProxy == null){
   print("failed to register Client Node at " + massURL);
   shutdown();
}
```

When Controller's AgletProxy object was obtained successfully the registration message is sent to Controller agent as shown bellow:

```
Message msg = new Message("Register");
msg.setArg("senderhostURL", factory.getHostingURL());
msg.setArg("username", username);
msg.setArg("os", os);
Object[] reply = null;
try {
        reply = (Object[])massControllerProxy.sendMessage(msg);
} catch(Exception e){
        print("failed to send register message");
        shutdown();
}
```

As shown at Fig. 5, at the Server Node side a Watcher agent is created and corresponding HostAgent and UserAgent aglets are disposed to Client Node (Fig. 6).

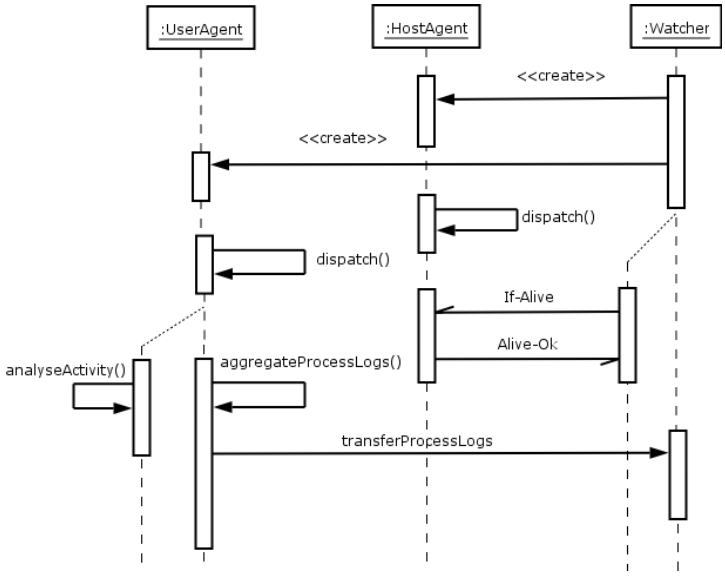


Fig. 6 – Monitoring system agents interaction sequence diagram

As it was mentioned before, one of the UserAgent functions is process logs aggregation. This operation is implemented using Aglets messaging. The UserAgent composes a message class if the “store-process-log” type and attaches latest process logs portion:

```
Message msg = new Message("store-process-log");
msg.setArg("content", DataPacket);
watcherProxy.sendAsyncMessage(msg);
```

The composed message is sent to corresponded Watcher agent. The Watcher agent handles all incoming messages using a common callback method defined in com.ibm.aglets.Aglet class. The signature of the mentioned method is the following:

public boolean handleMessage(Message msg) {}

The Watcher agent fetches attachment from Message object and appends process logs to user’s data storage.

The architecture of the developed system and its components is presented on Fig. 7:

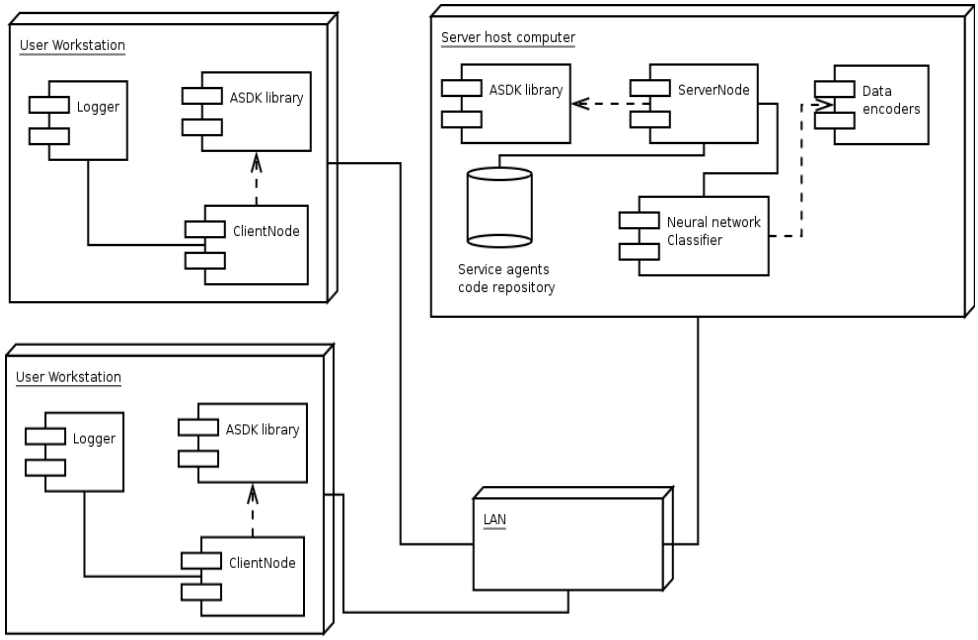


Fig. 7 – Deployment diagram for multi-agent monitoring system

According To Fig.7, user's workstation only needs to have installed aglets hosting platform. All service agents are stored on Server Node and migrate to Client Nodes on certain events described above. In such a way, it is easily to maintain the monitoring system without additional installations on workstations: update agent functions, add new types of agents.

Conclusion

In this paper the most suitable in our case technologies were evaluated – Java programming language and Aglets Software Development Kit – for implementation of a user behavior monitoring system using agent approach and neural network.

The system has a scalable architecture and minimal requirements for foregoing installation on workstations.

The further development is possible in the following directions: development of new types of agents (for example, network traffic analysis), implementation of decision making components (for instance, a fuzzy logic controller), administrative facilities enhancement (web based access, extended GUI).

Bibliography

[ASDK] Official ASDK website // http://aglets.sourceforge.net/

[AUML] Agent UML website // http://www.auml.org

[BGISZ] Jai Sundar Balasubramaniyan, Jose Omar Garcia-Fernandez, David Isacoff, Eugene Spafford, Diego Zamboni. *An Architecture for Intrusion Detection using Autonomous Agents.* http://citeseer.nj.nec.com/balasubramaniyan98architecture.html

[CannMah] James Cannady, James Mahaffey. *The Application of Artificial Neural Networks to Misuse Detection: Initial Results.*

[FIPA] Foundation for Intelligent Physical Agents website // http://www.fipa.org

[Ferrari, 2004] L. Ferrari. The Aglets 2.0.2 User's Manual. October, 2004. http://puzzle.dl.sourceforge.net/sourceforge/aglets/manual.pdf

[Gorod, 2001] V.Gorodetski, O.Karsaev, A.Khabalov, I.Kotenko, L.Popyack, V.Skormin. Agent-based model of Computer Network Security System: A Case Study. *Proceedings of the International Workshop "Mathematical Methods, Models and Architectures for Computer Network Security".* Lecture Notes in Computer Science, vol. 2052, Springer Verlag, 2001, pp.39-50.

[LO, 1998] Danny Lange, Mitsuru Oshima. Programming and deploying Java Mobile Agents with Aglets. ISBN: 0201325829; Published: Aug 20, 1998; Copyright 1998;

[Oshima, 1998] Mitsuru Oshima, Guenter Karjoth, Kouichi Ono. Aglets Specification 1.1 Draft. September, 1998. http://www.trl.ibm.com/aglets/spec11.htm

[SK, 2004] Скакун С.В., Куссуль Н.Н. Нейросетевая модель пользователя компьютерных систем // Кибернетика и вычислительная техника. 2004 Выпуск 143. С.55-68.

[SKL, 2004] С.В. Скакун, Н.Н. Куссуль, А.Г. Лобунец. Реализация нейросетевой модели пользователей компьютерных систем на основе агентной технологии.

[Sokol] Sokolov A.M. Computer System Intrusion Detection utilizing second-order Markoff chain. *Artificial Intelligence*. Vol. 1, pp. 376-380. (in Russian)

[Venners, 1997] B. Venners. The architecture of Aglets. Java World Magazine. April, 1997. http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood.html

Author's Information

Alexander G. Lobunets – Space Research Institute NASU-NSAU, system developer; 40 Glushkov Ave 03187, Kyiv, Ukraine; e-mail: alexander.lobunets@gmail.com

2.3. Ontologies

DEVELOPMENT OF EDUCATIONAL ONTOLOGY FOR C-PROGRAMMING

Sergey Sosnovsky, Tatiana Gavrilova

Abstract: *Development of educational ontologies is a step towards creation of sharable and reusable adaptive educational systems. Ontology as a conceptual courseware structure may work as a mind tool for effective teaching and as a visual navigation interface to the learning objects. The paper discusses an approach to the practical ontology development and presents the designed ontology for teaching/learning C programming.*

Keywords: *Ontology Design, Knowledge, Educational Ontology, C Programming, Ontology Visualization.*

Introduction

The intensity of modern technology development makes exceptional demands of the process of education. The speed of the knowledge deterioration increases steadily. According to the experts' reports the "half-value period" of a modern specialist is from 3 to 5 years. The number and the diversity of students grow up. Programs for life-long and distance education appear. Students differ in the learning goals, background, cultural aspects, which increase not only the volume of knowledge but also the ways, how it is taught. Different subjective views on the same knowledge for different groups of students may exist.

In these conditions a teacher as the main knowledge provider in the framework of modern education is overloaded. It becomes impossible for him/her alone to preserve the high quality of the knowledge taught. The solution is now obvious, knowledge should be created in the reusable and sharable form, in a way that once developed it could be used by anyone as a whole or partially.

Even greater need in making knowledge shareable and reusable is declared in the field of educational systems development. The knowledge base of a modern computer-based educational system should support the import and export of the knowledge in a standard format using standard protocols. Even for the domains where knowledge is pretty stable, like C Programming, such a perspective lead to the exceptional opportunity of using different systems from different developers in a common framework. The first step in this way is making the process of engineering of educational knowledge ontology-based.

The term of ontology emerged and became popular (even fashionable) during the last one and half decades. Though very yang it is yet a quite mature area of research. Ontological engineering inherits the practical and theoretical results of knowledge engineering, which has about forty yeas of history. According to one of the definitions "ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base" [Swartout et al., 1997]. It "defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary" [Neches et al., 1991].

In this paper we present the stepwise approach to ontology engineering and describe the experience of ontology developing for C-Programming. Developed knowledge structure is not just the hierarchy of the C language standard. It represents the application ontology designed for the purpose of education and accumulates the authors' experience of teaching several C-based programming courses. Next section gives details of the proposed algorithm for ontology development as well as a set of recommendations, which may be helpful in building a "beautiful ontology". Then in the section 3 we describe our domain, the motivation for the work presented here and, finally, the developed educational ontology for C programming. The summary and future work discussion conclude the paper in the section 4.

Stepwise Ontology Design

Generalizing our experience in developing different teaching ontologies for e-learning in the field of artificial intelligence and neurolinguistics [Gavrilova et al., 2004(a); Gavrilova, Voinov, 1998; Gavrilova et al., 1999;

Gavrilova, 2003; Gavrilova, 2004(b)] we propose a 5-step algorithm that may be helpful for visual ontology design.

We put stress on visual representation as a powerful mind tool [Jonassen, 1996] in structuring process. Visual form influences both analyzing and synthesizing procedures in ontology development process.

Concise Algorithm for Ontology Design:

1. Glossary development: gather all the information relevant to described domain, select and verbalize all essential objects and concepts.
2. Laddering: define main levels of abstraction and define type of ontology (taxonomy, partonomy, genealogy, etc.). Reveal hierarchies among these concepts and represent them visually on defined levels.
3. Disintegration: try to detail "big" concepts into a set of "smaller" ones via top-down strategy.
4. Categorization: group similar concepts and create meta-concepts to generalize the groups via bottom-up structuring strategy.
5. Refinement: update the visual structure and exclude excessiveness, synonymy, and contradictions. Try to create beautiful ontology.

Some Precepts to Create Beautiful Ontology:

Conceptual balance (Harmony). It is a challenge to formulate the idea of well-balanced tree, but some tips may be helpful:

- One-level concepts should be linked with a "parent" concept by one type of relationship (is-a, has part, etc).
- The depth of the branches should be more or less equal (±2 nodes).
- The general outlay should be symmetrical.
- Try to avoid cross-links.

Clarity:

- Minimal number of concepts is the best tip according Ockham's razor principle proposed by William of Ockham in the fourteenth century: "Pluralitas non est ponenda sine neccesitate", which translates as "entities should not be multiplied unnecessarily". The maximal number of branches and the number of levels should follow Miller's number (7±2) [Miller, 1956].
- The type of relationship should be understandable if the name of relationship is missing.

C programming Ontology

Domain Description

During a number of years, we have been teaching C-based programming courses to undergraduate students of the School of Information Sciences at the University of Pittsburgh and artificial intelligence disciplines in Saint-Petersburg State Polytechnic University. Several adaptive computer-based systems have been developed for serving such learning activities as parameterized quizzes, interactive examples and social navigation [Brusilovsky et al., 2004(a); Brusilovsky et al., 2004(b); Brusilovsky et al., 2004(c)].

The natural development of such tools is an evolvement towards the distributed web-based architecture where applications share the common students' profiles (student model) and the ontology of the domain (domain model). Some progress in this direction has been made [Brusilovsky, 2004]. Ontology of the domain as a framework for common knowledge base would allow our applications to "speak the same language". Moreover applications from side developers can share our knowledge base and become the part of the architecture.

Another motivation to build the ontology of C programming is connected with the attempts to create more meaningful and effective teaching strategies as there is no predefined way to teach C. Different textbooks and different instructors (even when using the same textbook) may introduce C concepts, combine them into lectures and explain them one on the basis of another in very different orders. One teacher may believe that it is better to teach "while" before "if-else", when another thinks visa versa. Not only the order of teaching concepts, but also the emphasis instructors' place on the different parts of the course and didactic paradigms they use could be different. Students may be required to learn first the structure of C program in details, or may be given "Hello World" example and immediately asked to code the similar program; the programming patterns for some courses

(like algorithm design or data structures) might have much higher importance then for the introductory C course etc.

The advantage of the ontology is that it attempts to unify different views on the domain. Selected parts of the ontology could be used for different sections of the course. The order, in which a teacher presents the material, is up to him/her while the basic hierarchical link structure is not violated.

Development of Educational C Ontology

We used the algorithm described above to create the ontology for teaching/learning C programming. Figure 1 demonstrates four top levels of the developed ontology. Lower levels trivially expand the hierarchy therefore we have hidden them. The main type of relationships is “has part”. That is why this is partonomy.

Naturally, the upper level central node is the C programming; second level represents the abstract meta-concepts, which combine more concrete entities. The major difficulties were to compose and to name these intermediate concepts. Figure 1 presents the fourth “release” of the design drafts.

The concepts of the third level are the main parts of the material that students study. They combine very separate areas of C programming knowledge, where an emphasis needs to be placed. The entities of the third level in their turn are subdivided into programming topics as sub-concepts. These topics for some branches are already concrete enough to be the theme of the lecture or the section in a syllabus. However, as we mentioned before, this level is far from being the last one.

As we told already, the purpose of this ontology is use in education; therefore it attempts to reflect not simply the standard of the language, but all necessary knowledge that students need to learn, including helpful programming techniques and compiler usage. It does not mean that we necessary provide a system, which teaches students for example to work with compilers. However, this branch in the ontology let us to use it, say, for navigating them in entering the online compiler tutorials.

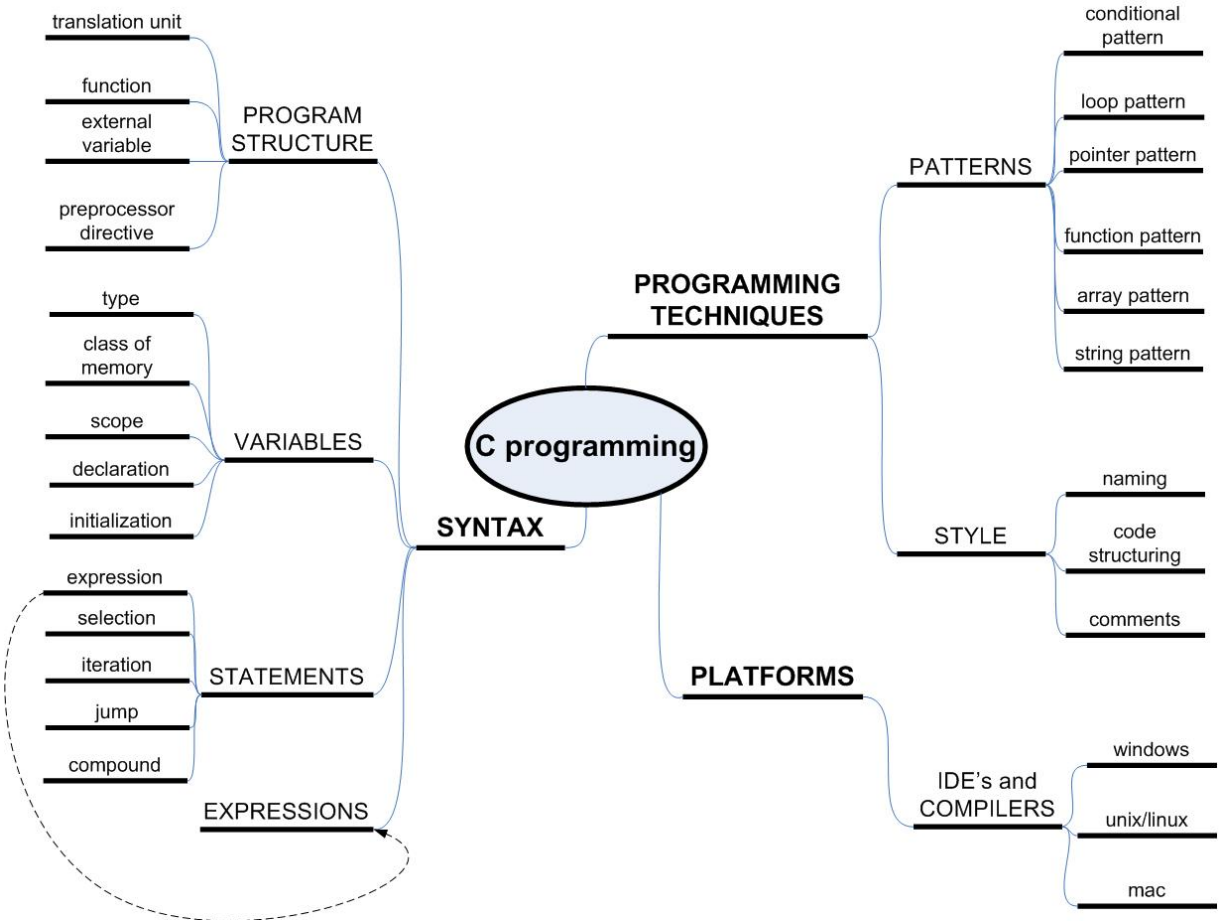


Figure 1. Top Levels of the Educaional Ontology of C Programming

The association link between expression sub kind of statements and expressions as a section of the third level though adds some irregularity to our ontology, is needed because of the educational purpose. In C standard expression statement is a kind of statement. That is why expression is a sub concept of statements. However, from the point of view of teaching/learning C, expressions are totally different area then any other.

The expressive power of the ontology allows us to encode different relations between concepts (by concepts we mean here entities in the hierarchy, but not the knowledge elements of the lowest level). Besides the link topology representing the whole-part hierarchical relationships, the order of concepts in a group represent the interconnection between them and preferred sequence of their study, though the last one is rather a recommendation then a directive.

As the main source for knowledge elicitation on the stage of glossary development we used [Kernighan, Ritchie, 1988].

Summary and Future Work

The paper has proposed the stepwise algorithm for ontology development and implementation of this algorithm for creation of the educational ontology for C programming. Created ontology does not simply replicate the hierarchical structure of the C language standard, but reflects the authors' vision on what is important in studying C and accumulates their experience of teaching C-related programming courses. Following three subsections discuss the directions of the future research.

Ontology-based Common Domain Model

The developed ontology is going to be used for several computer-based educational systems as a domain knowledge representation model. The C programming as a domain for adaptive educational systems is "lucky" to be formal enough for its concepts possess grammatical structure. This is especially true for the sub kinds of the SYNTAX meta-concept (see figure 1). Traditionally, the extraction of grammatically meaningful structures from textual content and the determination of concepts on that basis is a task for the special class of programs called parsers. In our case, we have developed the parsing component with the help of the well-known UNIX utilities: lex and yacc. This component processed source code of a C program and generates a list of concepts used in the program [Sosnovsky et al., 2004]. This tool will help us to automatically index the content of our adaptive systems in terms of the concepts of the developed ontology. This leads us to the exceptional opportunities of implementing mutual adaptation across different educational application. As a result, the possible set of instructional strategies increases, since on every step instructional treatments from more applications are available.

Ontology Visualization

As we already mentioned above the ontology is not just a technical instrument but a powerful mind tool also. Ontology visualization and creation of a student interface for an educational system is one of the authors' primary goals. The hierarchical structure of the ontology makes it natural to create a navigable hypermedia interface on its basis. In 1995 Gaines and Shaw created the WebMap – system integrating concept maps (which can be to some extend considered as an ontology visualization technique) with WWW, making a first step in this direction [Gaines, Shaw, 1995]. It seems very natural to use hypermedia as an implementation framework for ontology; hence we can use different methods of hypermedia adaptation, which are well developed now [Brusilovsky, 2001].

Ontology Evaluation

One more direction of future research is the evaluation of the developed ontology, from both perspectives: as a knowledge base framework and as an interface framework. From the first point of view, we can evaluate its structural consistency as a domain knowledge representation mechanism. Also, the quality of defined concepts as assessment units might be evaluated.

From the second point of view, the quality of ontology-based interface is to be evaluated on the subjective and objective levels. Subjective evaluation could be done on the basis of questionnaires filled by students at the end of the course. To evaluate it objectively we are going to perform the statistical analysis of logs of students' work with the system to find: first, how does work with the system correlates with course performance, second, how reasonable student use the interface, i.e. do they follow our hints and suggestions, and third, if they do, how do they benefit from it, how reasonably the system adapt its behavior to the specific student.

Acknowledgements

The work reported in this paper is supported by NSF grant # 7525 *Individualized Exercises for Assessment and Self-Assessment of Programming Knowledge* and by grant 04-01-00466 RFBR.

Bibliography

[1] [Swartout et al., 1997] Swartout, B., Patil, R., Knight, K., Russ, T. Toward Distributed Use of Large-Scale Ontologies, Ontological Engineering. AAAI-97 Spring Symposium Series, 1997, 138-148.

[2] [Neches et al., 1991] Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., Swartout, W. Enabling Technology for Knowledge Sharing. AI Magazine. Winter 1991, 36-56.

[3] [Gavrilova et al., 2004(a)] Gavrilova, T., Guian, F., Koshy, M. Ontological Tower of Babel. In proceedings of Second International Conference on Knowledge Economy and Development of Science, KEST 2004, Beijing, Tsinghua. University Press, 2004, 101-106.

[4] [Gavrilova, Voinov, 1998] Gavrilova, T., Voinov, A. Work in Progress: Visual Specification of Knowledge Bases. In A.P. del Pobil, J. Mira, M. Ali (Eds.) Lecture Notes in Artificial Intelligence 1416 "Tasks and Methods in Applied Artificial Intelligence", Springer, 1998, 717-726.

[5] [Gavrilova et al., 1999] Gavrilova, T., Voinov, A., Vasilyeva, E. Visual Knowledge Engineering as a Cognitive Tool. In Proceedings of International Conference on Artificial and Natural Networks IWANN'99, Benicassim, Spain, 1999, 123-128.

[6] [Gavrilova, 2003] Gavrilova, T. Teaching via Using Ontological Engineering. In Proceedings of XI International Conference "Powerful ICT for Teaching and Learning" PEG-2003, St.Petersburg, Russia, 2003, 23-26.

[7] [Gavrilova et al., 2004](b) Gavrilova, T., Kurochkin, M., Veremiev, V. Teaching Strategies and Ontologies for E-learning Information Theories and Applications, vol.11, N1, 2004, 61-65.

[8] [Jonassen, 1996] Jonassen, D. *Computers in the Classroom: Mindtools for Critical Thinking*, Englewood Cliffs, NJ: Prentice Hall, 1996.

[9] [Miller, 1956] Miller, G. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review, 1956, vol. 63, pp. 81-97.

[10] [Brusilovsky et al., 2004(a)] Brusilovsky P., Sosnovsky S., Shcherbinina O. QuizGuide: Increasing the Educational Value of Individualized Self-Assessment Quizzes with Adaptive Navigation Support. In Janice Nall and Robby Robson (eds.) Proceedings of E-Learn 2004. Washington, DC, USA: AACE, 2004, 1806-1813.

[11] [Brusilovsky et al., 2004(b)] Brusilovsky P., Sosnovsky S., Yudelson M., An Adaptive E-Learning Service for Accessing Interactive Examples. In Janice Nall & Robby Robson (eds.) Proceedings of E-Learn 2004. Washington, DC, USA: AACE, 2004, 2556-2561.

[12] [Brusilovsky et al., 2004(c)] Brusilovsky, P., Chavan, G., Farzan, R. Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In: P.De Bra (ed.) Proceedings of the Third International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH'2004), Eindhoven, The Netherlands, 2004.

[13] [Brusilovsky, 2004] Brusilovsky, P. A component-based distributed architecture for adaptive Web-based education. In: U. Hoppe, F. Vardejo and J. Kay (eds.) Proceedings of International Conference on Artificial Intelligence in Education: Shaping the Future of Learning through Intelligent Technologies (AI-ED'2004), Sydney, Australia, July 20-24, 2004. Amsterdam: OIS Press, 2004, 386-388.

[14] [Kernighan, Ritchie, 1988] Kernighan, B., Ritchie, D. C Programming Language (2nd Edition). Prentice Hall PTR; 2 edition: 1988.

[15] [Sosnovsky et al., 2004] Sosnovsky S., Brusilovsky P., Yudelson M. Supporting Adaptive Hypermedia Authors with Automated Content Indexing. In Lora Aroyo and Carlo Tasso (eds.) Proceedings of AH2004 Workshops: Part II /Second Workshop on Authoring Adaptive and Adaptable Educational Hypermedia/, Eindhoven, The Netherlands, 2004, 380-389.

[16] [Gaines, Shaw, 1995] Gaines, B. R. and Shaw, M. L. G. Concept maps as hypermedia components. International Journal of HumanComputer Studies, 43(3), 1995, pp. 323--361.

[17] [Brusilovsky, 2001] Brusilovsky, P. Adaptive hypermedia. User Modelling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 2001, 87-110.

Author's Information

Sergey Sosnovsky – University of Pittsburgh; 135, North Bellefield Street; Pittsburgh, PA, 15217, USA phone: +1 (412) 624-5513; e-mail: sas15@pitt.edu

Tatiana Gavrilova – Saint-Petersburg State Polytechnic University; Intelligent Computer Technologies Dept.; Politechnicheskaya, 29; Saint Petersburg - 195251, Russia; phone: + 7 (812) 550-40-73; e-mail: gavr@csa.ru

HOW CAN DOMAIN ONTOLOGIES RELATE TO ONE ANOTHER?[1]

Alexander S. Kleshchev, Irene L. Artemjeva

Abstract: *Building domain ontologies and applying them to different objectives, researchers faced the fact that many ontologies are associated with one another by one or another relations. Therefore, the problem arose to study relations among different ontologies of the same domains as well as of different ones. A formalization of a relation among domain ontologies is the analogous mathematical relation among mathematical models of these ontologies. The article considers the case when domain ontology model is represented by logical relationship system. Relations among domain ontologies give a possibility to reuse one ontology model when another ontology models are worked out and when new intellectual computer system for same or different domain is worked out.*

Keywords*: Mathematical model of domain ontology, ontologies representing the same conceptualisation, resemblance between ontologies, simplification of ontologies, composition of ontologies, intellectual task solver.*

Introduction

Building domain ontologies and applying them to different objectives, researchers faced the fact that many ontologies are associated with one another by one or another relations. Therefore, the problem arose to study relations among different ontologies of the same domains as well as of different ones. Although, as noted in [van Heijst et al, 1996], the field is still in its infancy and many questions are unsolved or even unaddressed (for example, how can ontologies be compared and integrated?), by now there has been some information in professional literature related to this problem. Many works studying this problem considered relations among ontologies within the context of ontology integration.

In [Gangemi et al, 1999] ontology integration is defined as the construction of an ontology C that formally specifies the union of the vocabularies of two other ontologies A and B. Three aspects of an ontology are taken into account: (a) the intended models of the conceptualisations of its vocabulary, (b) the domain of interest of such models, i.e. the topic of the ontology, and (c) the namespace of the ontology. The most interesting case is when A and B are supposed to commit to the conceptualization of the same domain of interest or of two overlapping domains. In particular, A and B may be:

Alternative ontologies: `The intended models of the conceptualizations of A and B are different (they partially overlap or are completely disjoint) while the domain of interest is (mostly) the same. This is a typical case that requires integration: different descriptions of the same topic are to be integrated.

Truly overlapping ontologies: Both the intended models of the conceptualisations of A and B and their domains of interest have a substantial overlap. This is another frequent case of required integration: descriptions of strongly related topics are to be integrated.

Equivalent ontologies with vocabulary mismatches: The intended models of the conceptualisations of A and B are the same, as well as the domain of interest, but the namespaces of A and B are overlapping or disjoint. This is the case of equivalent theories with alternative vocabularies.

Overlapping ontologies with disjoint domains: The intended models of the conceptualizations of A and B overlap while the domain of interest are disjoint. This concerns overlapping theories with different extensions. Actually, it is often the case that some fragments from an ontology A can be reused as components in another ontology B that models a different topic.

1 This paper was made according to the program of fundamental scientific research of the Presidium of the Russian Academy of Sciences «Mathematical simulation and intellectual systems», the project "Theoretical foundation of the intellectual systems based on ontologies for intellectual support of scientific researches".

Homonymically overlapping ontologies: The intended models of the conceptualizations of A and B do not overlap, but A and B overlap. This is the case of two unrelated ontologies with a vocabulary intersection that – if presented – generates polysemy: this is one of the reasons to maintain ontology modules.

To be sure that A and B can be integrated at some level, C has to commit to both A's and B's conceptualizations. In other words, the intention of the concepts in A and B should be mapped to the intention of C's concepts. The authors call this approach principled conceptual integration.

As noted in [Gangemi et al, 1996], the ontological integration envisaged is at a deeper level than representational integration. In fact, the representational integration concerns heterogeneity of formal languages, or heterogeneity of data base schemata. Ontological integration concerns the heterogeneity among conceptualizations.

In [Guarino, 1998] it is noted that information integration is a major application area for ontologies. As well known, even if two systems adopt the same vocabulary, there is no guarantee that they can agree on a certain piece of information unless they commit to the same conceptualization. Assuming that each system has its own conceptualization, a necessary condition to make an agreement possible is that the intended models of the original conceptualizations overlap. Supposing now that these two sets of intended models are approximated by two different ontologies, it may be the case that the two ontologies overlap while the intended models do not. Hence, it seems more convenient to agree on a single top-level ontology rather than relying on agreements based on the intersection of different ontologies.

In [Sowa] ontology integration is defined as the process of finding commonalities between two different ontologies A and B and deriving a new ontology C that facilitates interoperability between computer systems that are based on the A and B ontologies. The new ontology C may replace A or B, or it may be used only as an intermediary between a system based on A and a system based on B. Depending on the amount of change necessary to derive C from A and B, different levels of integration can be distinguished: alignment, partial compatibility, and unification.

Alignment is a mapping of concepts and relations between two ontologies A and B that preserves the partial ordering by subtypes in both A and B. If an alignment maps a concept or relation x in ontology A to a concept or relation y in ontology B, then x and y are said to be equivalent. The mapping may be partial: there could be many concepts in A or B that have no equivalents in the other ontology. Before two ontologies A and B can be aligned, it may be necessary to introduce new subtypes or supertypes of concepts or relations in either A or B in order to provide suitable targets for alignment. No other changes to the axioms, definitions, proofs, or computations in either A or B are made during the process of alignment. Alignment does not depend on the choice of names in either ontology. For example, an alignment of a Japanese ontology to an English ontology might map the Japanese concept Go to the English concept Five. Meanwhile, the English concept for the verb go would not have any association with the Japanese concept Go.

Partial compatibility is an alignment of two ontologies A and B that supports equivalent inferences and computations on all equivalent concepts and relations. If A and B are partially compatible, then any inference or computation that can be expressed in one ontology using only the aligned concepts and relations can be translated to an equivalent inference or computation in the other ontology.

Refinement is an alignment of every category of an ontology A to some category of another ontology B, which is called a refinement of A. Every category in A must correspond to an equivalent category in B, but some primitives of A might be equivalent to non-primitives in B. Refinement defines a partial ordering of ontologies: if B is a refinement of A, and C is a refinement of B, then C is a refinement of A; if two ontologies are refinements of each other, then they must be isomorphic.

Unification is a one-to-one alignment of all concepts and relations in two ontologies that allows any inference or computation expressed in the one to be mapped to an equivalent inference or computation in the other. The usual way of unifying two ontologies is to refine each of them to more detailed ontologies whose categories are one-to-one equivalent.

Alignment is the weakest form of integration: it requires minimal change, but it can only support limited kinds of interoperability. It is useful for classification and information retrieval, but it does not support deep inferences and computations. Partial compatibility requires more changes in order to support more extensive interoperability, even though there may be some concepts or relations in one system or the other that could create obstacles to full interoperability. Unification or total compatibility may require extensive changes or major reorganizations of A

and B, but it can result in the most complete interoperability: everything that can be done with one can be done in an exactly equivalent way with the other.

In [Wielinga et at, 1994] more general and more special ontologies are considered. Ontologies can have a recursive structure, meaning that ontology expresses a viewpoint on another ontology. Such a viewpoint entails a reformulation and/or reinterpretation on other ontology. This multi-level organization raises research questions such as the required expressiveness of the mapping formalisms for expressing viewpoints between ontologies. At least two different mapping operations can be identified. The first one is the mapping of terminology in one formalism onto the terminology of another formalism. The other one is the adding of supplementary commitments to one ontology by the mapping of the terms of the ontology onto the terms of the other ontology that takes additional commitments. The first terminology mapping will occur frequently. Since the ontology describes the meaning of the domain theory, for which it is a meta-model, without commitment to the language, in which this meaning is expressed, it will be confronted with meta-models, which partially convey the same meaning, but with different terminology. In this case merging of the two ontologies, or translation of the one ontology into the other is simply a mapping of terminology (e.g. boat in one ontology can be mapped on ship in another ontology if they refer to the same type of object in the universe of discourse (note that the knowledge bases described by these ontologies, even when they describe the same object in the real world, may be totally different!)). The second type of mapping occurs when it is necessary to provide an interpretation of underlying ontology or to provide a more specific interpretation that takes additional commitments. If the more restrictive ontology is already available (such as, sometimes, the ontology of a task or of a method) than it is necessary to map this ontology on the more general one. An example of this type of mapping occurs when there exists a model of the problem-solving task, that should be accomplished, and an existing ontology of the domain of the application. In this case, it is necessary to map terminology from the task (e.g. hypothesis) on terminology of the domain ontology. A simple mapping will not always be possible. Sometimes the ontology - introducing the additional commitments - needs to be constructed. This will often be the case with domain-model oriented ontologies.

In [Laresgoiti et al] and [Schreiber et al] a combination of ontologies is introduced. An example of some artifact such as a ship is considered. One can define multiple viewpoints on a ship. Well-known examples of such viewpoints are the physical structures (what are the parts of a ship?) and the functional structure (how can a ship be decomposed in terms of functional properties?). Although these two viewpoints often partially overlap, they constitute two distinct ways of “looking” at a ship. The purpose of ontology is to make those viewpoints explicit. For a design application such as CAD application, one would typically need a combined physical/functional viewpoint: a combination of two ontologies. For a simulation application (e.g. modelling the behavior of a ship), one would need an additional behavioral viewpoint. Many other viewpoints exist such as the process type in the artifact (heat, flow, energy, ...). Each ontology introduces a number of specific conceptualizations, that allow an application developer to describe, for example, a heat exchange process.

In [Studer et al, 1998] constructing ontologies from reusable ontologies is considered. Assuming that the world is full of well-designed modular ontologies, constructing a new ontology is a matter of assembling existing ones. There are several ways to combine ontologies. In [Studer et al, 1998] the most frequently occurring ones are only given. The simplest way to combine ontologies is through inclusion. Inclusion of one ontology into another has the effect that the composed ontology consists of the union of the two ontologies (their classes, relations, axioms). In other words, the starting ontology is extended with the included ontology. Conflicts between names have to be resolved. Another way to combine ontologies is by restriction. This means that the added ontology only is applied on a restricted subset of what it was originally designed for. The last way to assemble ontologies that is discussed in [Studer et al, 1998] is polymorphic refinement, known from object-oriented approaches.

It is possible to make some conclusions from this overview.

Many authors consider supporting interoperability as a main objective of ontology integration. But if this objective is reached, then it is not clear, what properties integrated ontologies and the result of their integration will have. Before studying these relations and building their formal models, it seems necessary to declare the fundamental properties, that all the relations among ontologies will have.

Consideration of overlapping but different conceptualizations as a necessary condition for possibility of ontology integration seems slightly speculative. If a conceptualization is adequate [Kleshchev et al, 2000a], then it must include the domain reality. In this case, the reality must be a subset of the intersection of these

conceptualizations. But the conceptualization that is their intersection is adequate, too. And any top-level conceptualization is worse (wider) than initial ones and especially than their intersection.

Vocabularies (concept systems) are only external structures, by which sets of intended situations, sets of intended knowledge systems and correspondences between them are expressed. Thus, it is unlikely that the union of the vocabularies can be considered as a principal property of ontology integration.

In the same way a mapping of concepts between two ontologies can be but one of ways to determine relations between ontologies. This way cannot be always applied to do this. If there is a mapping between concepts of two ontologies, then this fact alone does not allow us yet to call corresponding concepts as equivalent. The notion of equivalence is defined in mathematics as reflexive, symmetric and transitive relation.

When defining relations among ontologies, any references to properties of inferences or computations cannot be considered as admissible because they darken rather than clarify the meaning of introduced relations. The condition that all the inferences or computations are equivalent cannot be verified.

Properties of Relations among Domain Ontologies

Any domain is characterized by its reality, i.e. by the set of all the possible situations that have ever taken place in the past, are taking place now and will take place in the future [Kleshchev et al, 2000a]. Since the reality is known only partially, the domain knowledge system gives a more comprehensive idea of it. The knowledge system determines the set of situations admitted by the system, i.e. of such situations that are considered as possible in the reality by this knowledge system. So an observer comes across only situations of the reality, but a person possessing a knowledge system is able to imagine situations admitted by the knowledge system. Where does he or she take these imaginary situations from? They are determined by a conceptualization, that can be imagined as the implicitly given set of all the intended situations, i.e. all the situations which can be imagined within the framework of this conceptualization. In this case, the set of the situations admitted by a knowledge system is a subset of the set of all the intended situations.

An investigation of a domain, i.e. of its reality, is aimed at obtaining such a knowledge system that admits the set of situations being as near to the reality as possible. So the set of the situations admitted by a knowledge system is considered as an approximation of the reality, and the investigation of the domain is aimed at obtaining the best (the most adequate) approximation of its reality. This investigation perpetually gives birth to new knowledge systems instead of outdated ones. Where does these knowledge systems come from? They are determined by a conceptualization, too. So a conceptualization can be imagined also as the implicitly given set of all the intended knowledge systems, i.e. of such knowledge systems that can be formed within the framework of the concept system introduced by the conceptualization.

Ontology of a domain is an explicit representation of a conceptualization of the domain. Since the ontology can represent the conceptualization imprecisely, it determines two external approximations both for the set of all the intended situations and for the set of all the intended knowledge systems.

A relation among knowledge systems of the same or different domains is a relation defined on the sets of the situations admitted by these knowledge systems. If this relation takes place among these knowledge systems, and another, more adequate, knowledge system is found instead of one of them, then, in the general case, this relation does not have to take place among the renewed collection of knowledge systems. But from practical needs, it is quite desirable to have a possibility to determine with what other knowledge systems the new knowledge system is in the same relation.

A relation among ontologies of the same or different domains is a relation defined on the sets of all the intended knowledge systems of these ontologies (i.e. a subset of the Cartesian product of these sets) possessing the property that only the tuples consisting of knowledge systems belong to the relation that are in the analogous relation. Thus, if relations among ontologies are determined, then it determines the analogous relation among all the intended knowledge systems of these ontologies. In this article the relations possessing this property are considered only.

A formalization of a relation among domain ontologies is the analogous mathematical relation among mathematical models of these ontologies. The article considers the case when domain ontology model is represented by logical relationship system [Kleshchev, 2000a, 200b].

Ontologies Representing the Same Conceptualization

Domain ontology is a collection of agreements. It defines domain terms, determines their interpretations, contains statements that restrict the meaning of these terms and also gives interpretations for these statements. These agreements are the result of understanding among some members of the community working in this domain [Kleshchev et al, 2000a]. Different members of this community can advance different ontologies of this domain. The question arises: do these ontologies represent the same conceptualization or different ones? Let us discuss this question on the assumption that the models of these ontologies have the form of unenriched logical relationship systems [Kleshchev et al, 2000b].

If a conceptualization is considered as a set of all the intended situations, then two ontologies can represent the same conceptualization only when the sets of terms for situation description in these ontologies are the same. If a conceptualization is considered as a set of all the intended knowledge systems, then two ontologies can represent the same conceptualization only when the sets of terms for knowledge description in these ontologies are the same (they can be empty sets).

Let us consider the case when two different ontologies have the same sets of terms for situation description as well as the same sets of terms for knowledge description. In this case, to be different, these ontologies must have different sets of ontological agreements. Two points of view are possible on the condition under that these ontologies represent the same conceptualization: (1) when both the sets of intended situations and the sets of intended knowledge systems determined by these ontologies are the same; (2) when, following the definitions of the previous section, the sets of intended knowledge systems determined by these ontologies are the same, and for any knowledge system the sets of situations admitted by this knowledge system in these two ontologies are also the same.

The models of these ontologies have the same sets of unknowns and the same sets of parameters but different sets of logical relationships. Formalization of the conditions above means that:

1. the sets of logical relationships for the models of these ontologies are equivalent as applied logical theories (two applied logical theories are equivalent, if they have the same set of models [Kleshchev et al, 2000b]);
2. the models of this domain determined by the models of these ontologies for the same knowledge model have the same models of the reality, i.e. the models of these ontologies are equivalent as unenriched logical relationship systems [Kleshchev et al, 2000b].

It is easily seen that both these conditions are equivalent. Thus, equivalent transformations of the logical relationship set for a domain ontology model (as an applied logical theory) lead to a model of another ontology representing the same conceptualization. These transformations can be, for example, transformation of an applied logical theory to a disjunctive normal form, a conjunctive normal form and so on.

Now let us consider the case when two ontologies of the same domain have the same sets of terms for situation description but different sets of terms for knowledge description. In this case, following the previous section, we can consider these ontologies as representing the same conceptualization, if there is a one-to-one correspondence between their knowledge system sets, and for any corresponding knowledge systems the sets of the situations admitted by these knowledge systems are the same. When passing to models, it means that the models of these ontologies are equivalent [Kleshchev et al, 2000b].

Now let us consider the case when two ontologies of the same domain have different sets of terms for situation description but the same sets of terms for knowledge description. In this case, following the previous section, we can consider these ontologies as representing the same conceptualization if for any knowledge system there is a one-to-one correspondence between the sets of the situations admitted by this knowledge system in both these ontologies. When passing to models, it means that the models of these ontologies have the same sets of all possible enrichments and are isomorphic [Kleshchev et al, 2000b].

Resemblance between Ontologies

In the case when both terms for situation description and terms for knowledge description are different in two ontologies, it is possible to speak of resemblance between these ontologies only (of the same or different domains).

Two knowledge systems related to different ontologies (of the same or different domains) can be considered as resembled if there is a one-to-one correspondence between the sets of situations admitted by these knowledge systems. So two ontologies of the same or different domains can be considered as resembled if there is such a one-to-one correspondence between their sets of intended knowledge systems that any corresponding knowledge systems are resembled. It means that the models of these ontologies are isomorphic [Kleshchev et al, 2000b].

If terms of an ontology are substituted by different terms (by abstract designations), then, as a result, a resembled ontology will be obtained. The resemblance between ontologies is a relation of equivalence. It is reflexive, symmetric and transitive.

Simplification (Coarsening) of Ontologies

Comparing different ontologies of the same domain, one can sometimes say that one of these ontologies is a simplification (coarsening) of another. In the same way considering ontologies of different domains, one can sometimes say that an ontology of one of these domains resembles a simplified ontology of another domain. The availability of more simple and more complex ontologies of the same domain can be important to develop knowledge based systems for specialists of different qualifications (for example, medical systems for physicians of high qualification and for doctor's assistants).

One can say that a knowledge system related to an ontology is more simple than a knowledge system related to another ontology (of the same or different domains) if for every situation admitted by the second knowledge system (of the more complex ontology) the only situation admitted by the first knowledge system (of the more simple ontology) can be set as corresponding. Then one can consider an ontology as more simple than another ontology (of the same or different domains) if for every knowledge system of the second ontology the only more simple knowledge system of the first ontology can be set as corresponding. It means that a model of the first ontology is a homomorphic image of the second ontology [Kleshchev et al, 2000b].

A domain model <O1, k2> is a simplification (coarsening) of a domain model <O1, k1> if the enriched logical relationship system <O1, k2> is a homomorphic image of the system <O1, k1>. A coarsened model of medical diagnostics can be obtained, for example, by elimination of a few signs.

The simplification determines a partial order of ontologies. If B is more simple than A, and C is more simple than B, then C is more simple than A. If one ontology is simpler than another, and the second ontology is simpler than the first ontology, then they resemble one another.

Composition of Ontologies

When we speak about complex domains, we must usually bear in mind that these domains include knowledge from other different domains. Thus, when knowledge and reality of complex domains are described, concepts related to other domains are used. These other domains are components of the complex domain. Ontologies of complex domains are built from components, which are ontologies of other domains.

We can consider that a (starting) knowledge system related to a complex domain consists of knowledge systems (components) related to other domains if every component is more simple than the starting knowledge system, and the transfer from any situation admitted by the starting knowledge system to corresponding situations admitted by components takes place without the loss of information. The latter statement means that for any two different situations admitted by the starting knowledge system the two sets consisting of the situations corresponding to these two situations and admitted by all the components are different. In this case a starting ontology of a complex domain can be considered as consisting of components which are ontologies of other domains if every component is more simple than the starting ontology, every knowledge system of the starting ontology consists of knowledge systems of components, and the transfer from any knowledge system of the starting ontology to corresponding knowledge systems of components takes place without loss of information. The latter statement means that for any two different knowledge systems of the starting ontology the two sets consisting of the knowledge systems of components corresponding to these two knowledge systems are different. It follows from these definitions that every model of a starting ontology for a complex domain is the product of ontology models that are components [Kleshchev et al, 2000b].

Using Relations among Domain Ontologies for Working out Intellectual Solvers for Applied tasks

At present time, a demand arises to develop program systems for different domains having means for adaptation of problem solving methods to alteration of knowledge in these domains. Such program systems are called the intellectual solvers for applied problems. The base of developing an intellectual solver is domain ontology. Intellectual problem solvers on domain should permit experts and specialists to form and edit ontology and knowledge on domain and to get the programs for solving applied problems in this domain.

If there are alternative points of view on the same domain then we can speak about equivalence or resemblance between different ontologies of the domain. Recognition of the equivalence between alternative points of view on the same domain can give a possibility to solve tasks arising within the framework of a point of view using methods worked out within the framework of another point of view. Recognition of a resemblance between ontologies of the same domain can give a possibility to solve the tasks described within the framework of one concept system by methods developed within the framework of the other concept system. Recognition of a resemblance between ontologies of different domains can give a possibility to solve the tasks of one domain by reasoning using analogy in the case when methods for solving analogous tasks of the other domain have been developed.

A mathematical specification of an applied task can contain a domain model, input and output data of the task, task conditions (a set of formulas), and also criterion of selecting solutions. All the components of the applied task specification are represented in terms of the domain model. If every value of input data is replaced by a variable (different variables correspond to different values) in the task specification then the mathematical specification of the task will be transformed into a mathematical specification of a class of applied tasks. These variables will be called variables of the class of applied tasks. There is a one-to-one correspondence between the set of tasks belonging to the class and the set of all the admissible substitutions of values instead of these variables. To get the mathematical specification of an applied task belonging to a class it is necessary to replace all the variables of the class by values of input data.

If the domain model is replaced by the domain ontology model and knowledge base of the domain are considered as another set of input data of all the tasks of the class then the mathematical specification of the class of applied tasks will be transformed into the mathematical specification of the class of applied tasks corresponding to the domain ontology. There is a one-to-one correspondence between the set of tasks belonging to the class and the Cartesian product of the set of all the admissible substitutions of values instead of variables of the class of the tasks by the set of all the possible knowledge bases for the domain ontology model. To get the mathematical specification of an applied task belonging to a class of tasks corresponding the domain ontology it is necessary to replace all the variables of the class by values of input data and to enrich the domain ontology model by an appropriate knowledge base.

Finally, if domain terms in the mathematical specification of the class of applied tasks corresponding to a domain ontology are replaced by abstract designations then this mathematical specification of the class will be transformed into a mathematical task. The transformation of a mathematical specification of a class of applied tasks corresponding to a domain ontology into a mathematical task is important because different classes of applied tasks corresponding to ontologies of different domains, generally speaking, can be reduced to the same mathematical task.

If intellectual solver can solve mathematical tasks then it can be used for any domain which ontology model is isomorphic or equivalent to ontology model from mathematical task specification.

Let's consider a set of mathematical specifications of applied tasks such that every specification contains the same domain model. Such a set will be called an applied multitask. Just as an applied task was transformed into a class of applied tasks, the latter was transformed into a class of applied tasks corresponding to a domain ontology, and the latter was transformed into a mathematical task, so an applied multitask can be transformed into a class of applied multitasks, the latter can be transformed into a class of applied multitasks corresponding to a domain ontology, and the latter can be transformed into a mathematical multitask. An intellectual solver is intended for solving applied multitasks of a class of applied multitasks or for solving applied multitasks of a class of applied multitasks corresponding to a domain ontology.

The availability of simpler and more complex ontologies of the same domain can be important to develop intellectual solvers for specialists of different qualifications. As this takes place, working out methods for solving

tasks based on a more simple ontology can be a simplification of methods for solving the corresponding tasks based on a more complex ontology. The same can also take place for ontologies of different domains.

The same methods often can be used for solving a few tasks and subtasks. Abstraction of applied tasks to mathematical ones gives a possibility of reusing methods for their solving. If different applied tasks can be reduced to the same mathematical task then a method for solving the mathematical task can be used for solving these applied tasks too. A decomposition of a mathematical task into mathematical subtasks in working out a method for solving the mathematical task gives an additional possibility for reusing methods. In this case, the same mathematical subtasks can be components of decompositions of different mathematical tasks and methods for solving these subtasks can be components of methods for solving different mathematical tasks.

Ontologies of complex domains are built from components, which are ontologies of other domains. The fact that an ontology of a complex domain is a composition of other domain ontologies can be used to work out methods for solving tasks in the complex domain. These tasks can be divided into subtasks corresponding to tasks for components of the ontology. If methods for solving these tasks have been already known, working out a method for solving the whole task may be considerably simplified.

Conclusions

In this article, general properties of relations among domain ontologies have been considered. Examples of these relations can be the relation between ontologies representing the same domain conceptualization, the relation of resemblance between ontologies, the relation “to be more simple or more complex” and the relation among an ontology consisting of components, which are other ontologies, and these components. A formalization of these relations has been suggested. This formalization preserves the properties above. These results show that the definitions of an ontology and its model given in [Kleshchev et al, 2000a] allow us to recognize these relations among ontologies. Relations among domain ontologies give a possibility to reuse one ontology model when another ontology models are worked out and when new intellectual computer system for same or different domain is worked out.

References

[van Heijst et al, 1996] van Heijst G., Schreiber A.T., and Wielinga B.J. Using Explicit Ontologies in KBS Development. In International Journal of Human and Computer Studies, 1996, 46 (2-3): 183-292.

[Gangemi et al, 1999] Gangemi A., Pisanelli D.M. and Steve G. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies // In Data & knowledge Engineering, Vol. 31, N 2, 1999, pp. 183–220

[Gangemi et al, 1996] A. Gangemi, G.Steve, F. Giacomelli. ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration. In P. van der Vet (ed.) Proceedings of the Workshop on Ontological Engineering, ECAI96, 1996.

[Guarino, 1998] Guarino N. Formal Ontology and Information systems. In Proceeding of International Conference on Formal Ontology in Information Systems (FOIS’98), N. Guarino (ed.), Trento, Italy, June 6-8, 1998. Amsterdam, IOS Press, pp. 3-15.

[Kleshchev et al, 2000a] Kleshchev A.S., Artemjeva I.L. Mathematical Models of Domain Ontologies. Technical Report 18-2000. Vladivostok, 2000. 43 p. (available in http://iacp.dvo.ru/es/)

[Kleshchev et al, 2000b] Kleshchev A.S., Artemjeva I.L. Unenriched Logical Relationship Systems. Technical Report 1-2000. Vladivostok, 2000. 43 p. (available in http://iacp.dvo.ru/es/)

[Laresgoiti et al] L. Laresgoiti, A. Anjewierden, A. Bernaras, J. Corera, A.Th.Schreiber, B.J.Wielinga. Ontologies as Vehicles for Reuse: a Mini–expirement. Available from http://ksi.cpsc.ucalgary.ca/KAW/KAW96/laresgoiti/k.html

[Schreiber et al] G.Schreiber, W.Jansweijer, B.Wielinga. Framework & Formalism for expressing Ontologies (version 2). Technical Report, University of Amsterdam, DO1b2. http://www.swi.psy.uva.nl/projects/NewKACTUS/Reports.html

[Sowa] Sowa J., Knowledge Representation: Logical, Philosophical and Computational Foundations. In http://www.bestweb.net/sowa/ontology/gloss.htm

[Studer et al, 1998] R Studer, V.R. Benjamins, D. Fensel. Knowledge Engineering: Principles and methods. In Data & Knowledge Engineering 25, 1998, p 161–197

[Wielinga et at, 1994] Wielinga, B., Schreiber A.T., Jansweijer W., Anjewierden A. and van Harmelen F. Framework and Formalism for Expressing Ontologies (version 1). ESPRIT Project 8145 KACTUS, Free University of Amsterdam deliverable, DO1b.1, 1994. http://www.swi.psy.uva.nl/projects/NewKACTUS/Reports.html

Author's Information

Alexander S. Kleshchev – kleschev@iacp.dvo.ru

Irene L. Artemjeva – artemeva@iacp.dvo.ru

Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences
5 Radio Street, Vladivostok, Russia

DEVELOPMENT OF PROCEDURES OF RECOGNITION OF OBJECTS WITH USAGE MULTISENSOR ONTOLOGY CONTROLLED INSTRUMENTAL COMPLEX

Alexander Palagin, Victor Peretyatko

Abstract*: the ontological approach to structuring knowledge and the description of data domain of knowledge is considered. It is described tool ontology-controlled complex for research and developments of sensor systems. Some approaches to solution most frequently meeting tasks are considered for creation of the recognition procedures.*

Keywords*: the tool complex, methods of recognition, ontology.*

Introduction

One of the ambitious purposes of the world civilization is construction of the knowledge-oriented society. In computer science, a main priority direction thus is intellectualization of computer resources and technologies, in particular creation knowledge-oriented ontology controlled intelligence systems for various assignments. Information technologies on their basis are composing components of all high technologies. Except for usage in spheres of socioeconomic activity (the most difficult) spheres of research and development activity which result are objects of new knowledge, engineering, high technologies are rather important. The majority of these applications of intelligent systems is related to the problem solving, recognition (identifications), the diversification of their settings and implementation which are extremely various. The present material is devoted to usage of the instrumental ontology-controlled complex for development of sensor systems, and in particular new (and refinement of already known) methods of recognition.

Productivity

Productivity is the most important parameter for the tasks related to recognition of signals data acquisition from external sources and their processing. The logical approach to the execution of these conditions is creation of tool complexes. Instrumental complex (IC) should unite in itself the block of interaction with an environment, the block of digital signal processing, and the block of interaction with the user. (Fig. 1)

The block of digital signal processing (DSP) contains the ontological component. Recently ontology's designing and knowledge processing become the object of steadfast notice of contributors of the various domains mainly ones operating in the knowledge engineering area. Among others, it is possible to mark such important directions of their interests:

- Knowledge management. To this directions such sections may be relevant as (intelligent) search, an automatic information accumulation from various sources (channels of news operating RSS), extract of knowledge from texts (Text mining) or sets of others - unstructured documents (text, databases, HTML, XML, etc.). The result of such analysis should become the generated document, which briefly formulates the major positions of the document, or groups of documents.

Attempt to create advanced (system like ontology for WEB) [1]. Two key standards, which subsequently were used as a basis of the project named Semantic Web are completed. It is possible, that we are on a

threshold of significant events' variations comparable with those, which have brought the Internet, World Wide Web and HTML. The given development - attempt to correct an ancient disadvantage of the Internet - its weak structure ability. New standards are Resource Definition Framework (RDF) and Web Ontology Language (OWL). They are the part of Semantic Web project and the main idea consists in making the information transmitted on the Network more clear, having provided possibility of identification, sorting and processing. Till now Web has been mainly oriented to operation of the person, but Web of the following generation, by opinion of developers of the project will be designed on the computer processing of the formation [1]. As a basis of the future WEB is assumed to be not only to search and read, but also to understand a contents of the Internet - information, and to reach it not through the creation of programs of the artificial intelligence, simulating activity of the person, but through usage of resources of expression of semantics of the data and their links [2].

Ontologies designing on the basis of available program systems which are reduced to filling special forms by the description of this or that data domain. Such developments are carried on more often in the research (educational) projects, for example [7].

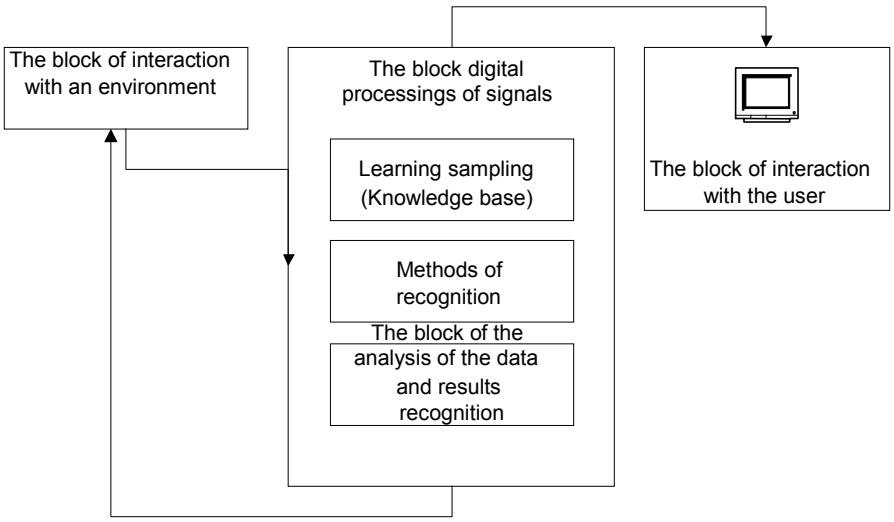


Fig. 1 Block-scheme of the tool complex.

Ontology

Ontology (O) as the formal description of the defined data domain can be represented as:

$$O = \{X, R, F\},$$

Where:

O - a finite set concepts (terms, quantum's of knowledge);

R - a finite set of the relations between concepts;

F - set of functions of interpretations of concepts and/or relations.

The set X is frequently bunched as subsets $X = \{X1, X2, \ldots Xq\}$, $q=1,2, \ldots, Q$ on subsumption to tags, making tree-like network structure. In a case $R \subset \varnothing$, $F \subset \varnothing$, ontology $O = \{X\}$ represents the usual dictionary - glossary.

Relation R - are served for association concepts in orderly structure (semantic model of a field of knowledge). Link is always unidirectional - one of the concepts is a grandparent, and another - the descendant. Link can define the ratio between quantity of grandparents and descendants 1:1, 1:N, N:1, N:M. Link can be hierarchical i.e. if A it is coupled with B, and B - with C and so on.

For effective scientific operation in this or that field of knowledge - it is necessary to fulfill the following standard procedures: to structure knowledge which concern both to a researched theme, and to adjacent areas, to study

existing methods of researches, putting forward hypotheses and trying to check them on concrete examples to develop new methods.

Usually, the contributor fulfils all this, drawing up all logical chains as speak, in mind. In this case, designed in process of research operations ontology can help him in solution of a lot of the important tasks essentially. It is correctly and full constructed model of the field of knowledge can become the power factor for research and development designing operations. Even preliminary constructed variants of ontology give more complete "imbedding" in a theme researches (general domain). The description of objects and links connecting them will allow presenting more precisely processes, which occur in this or that system (a fragment of the system).

Ontology in the application to methods of recognition fulfils such functions:

Formalistic - structuring of knowledges;

Information- retrieval - carries out relevant navigation;

Creative- generation of applications;

Transforming- perfecting of the base methods of recognition;

Extension- support of extended model computer ontology.

Computer Ontology

In [7] computer ontology is described which have been created on Lotus Notes platform (though at the given stage editors of ontology are developed, such as OntoEdit, OILed, Protege which give a graphic interface and create an output file according to standards of representation for Semantic Web). The program fulfilled on Lotus-Domino platform, is non-relational database which has the defined advantages: the convenient system of replication, the power 128 bit system of encoding, a built-in system of navigation, broadcasting of contents of base - programs in Web, etc.

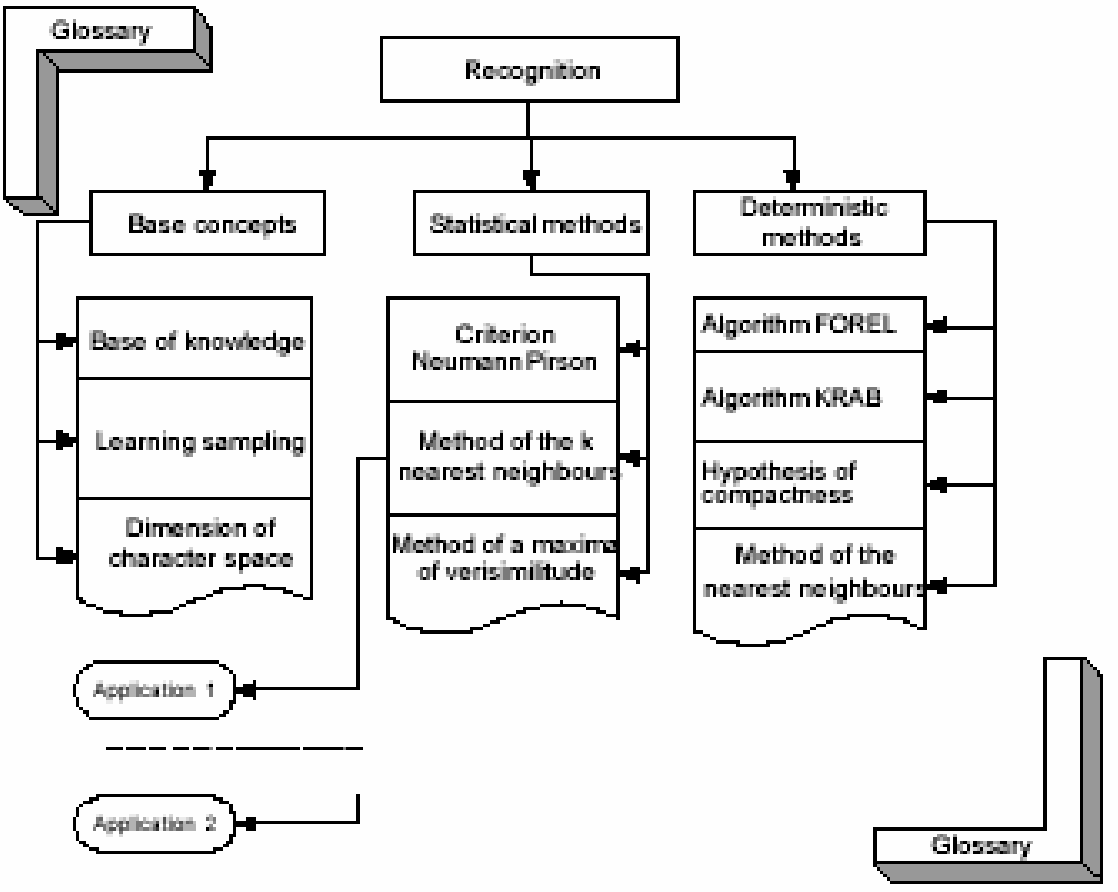


Fig. 2 Scheme ontology

The system is suitable for filling by the information from arbitrary fields of knowledges. In process of creation, the area of methods of recognition, which can be used, for construction of sensor systems has been selected.

The program will consist of the following blocks:

Categories - the description of categories which concern to a given theme,

Applications – examples of usages (demonstrations) of operations of different kind,

Navigation - relevant navigation by category (both through headers, and internal content),

Glossary – description of general scientific terms of ontology,

The library - storage of necessary files,

Diary - clone of an organizer. Allows to plan the operations and to bring various arbitrary records,

Help for the user.

Ontology represents an outline (see fig. 2). The theme ontology "Recognition" has three main categories: base concepts, statistical methods and deterministic methods. Inside these subsections also are concepts - quantum's of knowledge. It is necessary to mark that each of concepts can be referred to more than one subsection. For example, some methods of recognition are simultaneously statistical and deterministic ones. It is possible to connect the derived object of the application in which practical application of some concept is described in parent concepts to each quantum of knowledge. Besides, from anyone concept the reference to arbitrary others concepts, units of the application or a glossary can be created.

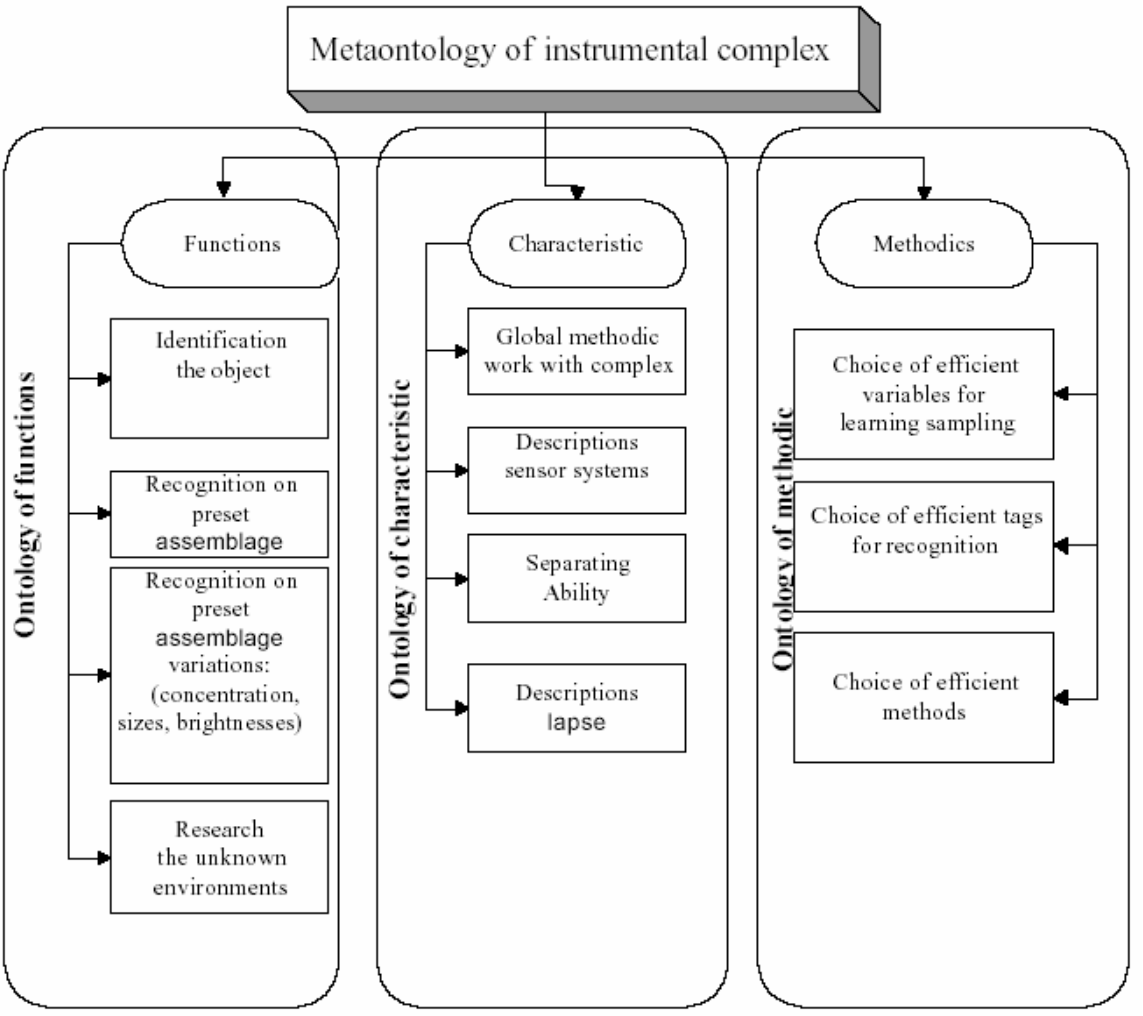


Fig. 3 Functional diagram instrumental complex

For each of concepts, (for each quantum of knowledge) it is underlined:

Theme;

The type of the message (definition, an explanation, an example, etc.)

Category (in a case with methods of recognition - the class of a methods);

Keywords;

Weight coefficient of concept (on a five-point scale);

The language of the message (Russian, English);

Source.

More detailed description of ontological components of the complex is resulted in [7].

The modern level of development of the methods of a multivariate statistical analysis allows to carry out classification of objects on a wide and objective basis, in view of all essential structural - typological tags and characters of object allocation in the preset system of tags.

At logical level IC should contain such descriptions:

Common procedures of operation with the complex for various conditions;

Sets of characteristics of the block of interaction with an environment (a set of sensor controls, procedures of operation with them, errors, separating ability for various sorts of defined objects);

A method (or several methods) of recognition;

Procedures of choice of effective variables, for samplings;

Procedures of choice (automatic or hand-held) of effective methods for operation with each concrete object or group of objects (fig. 3)

This tool complex is intended, in particular, for research of the gaseous environments, and coupled to a set of the sensor controls included in the projected instrument (system). Sampling procedure consists in removal of metrics from sensor controls at heating. The scheme of the procedure of removal of the data with the subsequent processing is represented on fig. 4.

Classification of Multivariate Objects

There are many methods of classification of multivariate objects with the help of a computer. Methods of the first group are connected with task of "recognition", identifications of objects and have received the name of methods of pattern recognition. The sense of recognition consists in that any object showed to the computer with the least probability of an error has been referred to one of beforehand-generated classes. Here to the computer all over again show a taught sequence of objects (about each it is known to what class or "image" ' it belongs), and then, "having trained", the computer should recognize to what classes from the investigated collection the proposed object is belonging.

More common approach to the classification includes not only reference of objects to one of classes, but also simultaneous creation of "images", the number of which can be beforehand unknown. At absence of a taught sequence such classification is made on the basis of tendency to collect in one group somewhat similar objects moreover so that objects from different groups (classes) were whenever not similar. Such methods have received the name of methods of automatic classification.

Now tens and hundreds of the various algorithms realizing multivariate classification automatically are developed. They are based on various hypotheses about character of allocation of objects in multivariate space of tags, on various mathematical procedures. Browses of these methods widely represented in the literature.

Various requirements are characteristic for various types and procedures of recognition of the objects to the measured data. In general, *choice given* (variables for recognition) is the most challenge in all chain of the operations coupled to recognition. Thus, it is possible to speak about two types of choice:

About choice of the measured data: the most sensitive for these objects a range, periodicity, etc., that is those aspects which are defined *by a procedure of samplings* (and, accordingly, can be modified at this stage);

About choice of variables – tags for *recognition.*

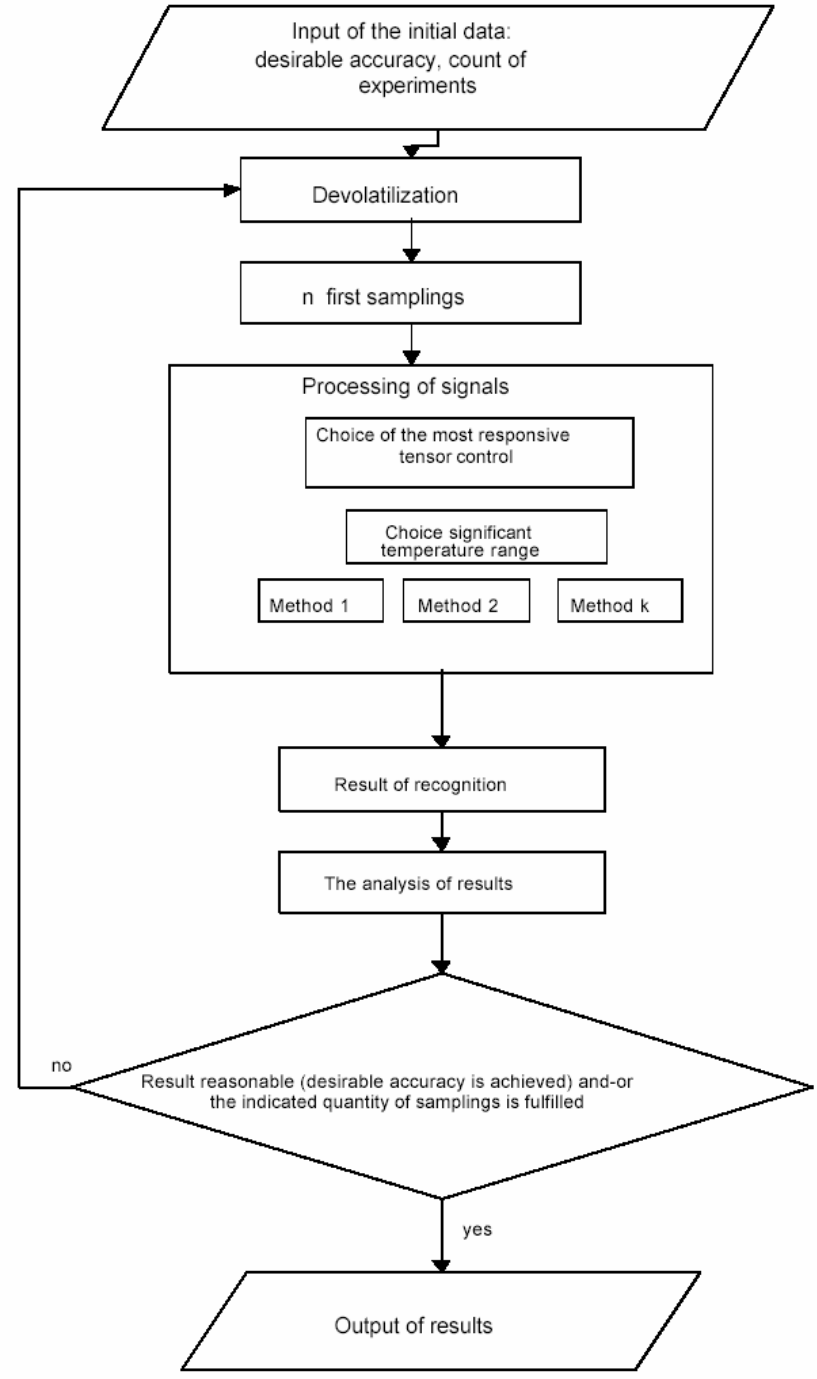


Fig. 4 Procedure of sampling

The Requirement of Independence

The requirement of independence of tags is typical of the majority of methods of recognition. The requirement is logically proved: if the data are dependent, the information contained in one tag, already is presented at the learning sampling, and in other measured variables the method of recognition can "tangle" its repetitions only. For example, for method Bayes (posterior probability) this requirement is extremely strict and mandatory.

For other methods, this rule can be neglected. To such ones the methods based on clusterizations, "cognizance" objects (a method to the nearest units, a method of measurement standards) are concerned.

However, the independence of tags can be frequently guaranteed by the very sensor subsystem. For support of independence of tags pertinently takes advantage the check of correlation tags.

For example, for obtaining an independent data set from 3 samplings the same process (object) it is possible to divide the data into intervals, and to check up on correlation each of intervals. Then the most correlated intervals are deleted from learning sampling (fig. 5.). Thus, it is possible to investigate, for definition of optimal quantity of parts.

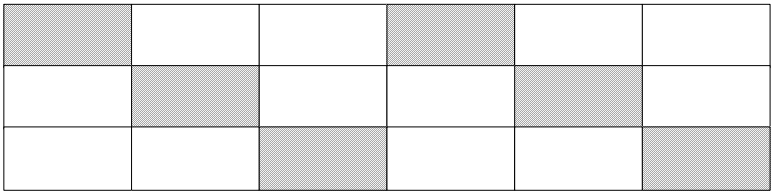
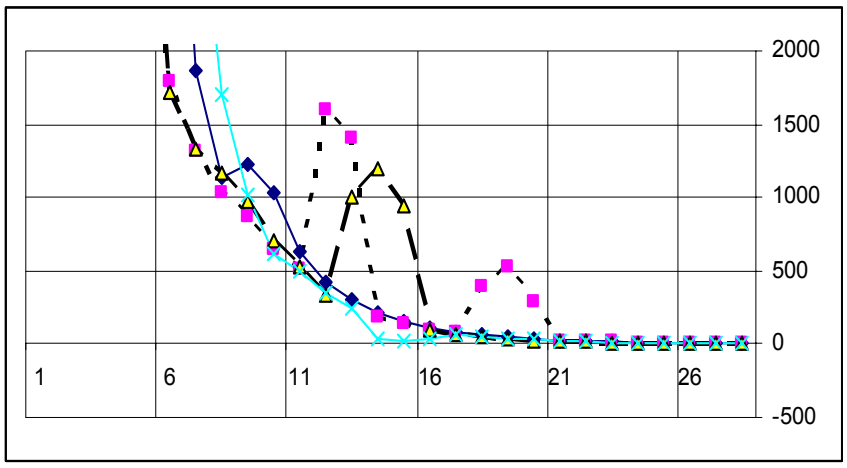


Fig. 5 Scheme of division of an interval of sampling on 6 parts, for definition of minimum correlation

It is possible also, for the analysis of a digital data to make the various sorts of the intermediate conversions, for more visual data representation, and, more convenient extract of the information from these data. For example, at presence of a plenty of the schedules constructed in one range, it is possible to construct a matrix in which to point that quantity of schedules which passes through each of coordinate squares. This information (fig. 6) can be used for development and modification of existing methods of recognition. In particular, it carries the information on potential possibilities of definition of measured objects on various intervals of a scale of argument.



4	1	0	0	0
4	2	0	0	0
4	4	1	0	0
0	4	4	4	4

Fig. 6 Schedules of the data, and a matrix of quantities

Conclusion

Considered tool ontology- controlled complex is the tool for creation and learning of procedures of identification of objects. Some approaches are resulted in creation of new procedures of the analysis and direct recognition.

Bibliography

1. [T. Berners-Lee, 1999]. T. Berners-Lee Weaving the Web, Harper, San Francisco, 1999 P. 2-13
2. [T. Berners-Lee, 2001]. T. Berners-Lee. -"Semantic Web Road map ", http://www.w3.org/DesignIssues/Semantic.html
3. [Stin Deker, 2001] Stin Deker, Sergey Melnik, Franc van Hermelen, etc. - " Semantic Web: roles XML and RDF ", "Open systems", #9 2001 of page 41-70;
4. [Fu K.S., 1977] Fu K.S. Structural methods in pattern recognition. - M.: the World, 1977, page 12-15
5. [Voloshin G.Ya., 2000] Voloshin G.Ya. Method of pattern recognition (the abstract of lectures) Page 9, 36-40, 42-43
6. [Gorelik A.L., 1977] Gorelik A.L., Skripkin V.A. Method of recognition. - M.: Высш. шк., 1977, page 37-45
7. [Palagin A.V.,2005] Palagin A.V., Peretyatko V.J. Tool ontology controlled complex for research and developments of touch systems. УСиМ, №2, 2005.

Authors' Information

Alexander Palagin - Deputy director, professor. Institute of Cybernetics, NASU, Kiev, Ukraine e-mail: palagin_a@ukr.net

Victor Peretyatko - postgraduate student of the Institute of the Cybernetics. Timoshenko st., apt 3-a, 12, e-mail: victor@iapm.edu.ua

A CONCEPT OF THE KNOWLEDGE BANK ON COMPUTER PROGRAM TRANSFORMATIONS

Margarita A. Knyazeva, Alexander S. Kleshchev

Abstract: *The paper presents basic notions and scientific achievements in the field of program transformations, describes usage of these achievements both in the professional activity (when developing optimizing and unparallelizing compilers) and in the higher education. It also analyzes main problems in this area. The concept of control of program transformation information is introduced in the form of specialized knowledge bank on computer program transformations to support the scientific research, education and professional activity in the field. The tasks that are solved by the knowledge bank are formulated. The paper is intended for experts in the artificial intelligence, optimizing compilation, postgraduates and senior students of corresponding specialties; it may be also interesting for university lecturers and instructors.*

Keywords: *Knowledge bank; Ontology; Knowledge base; Ontology editor; Database editor; Knowledge processing*

Introduction

The science, higher education and professional activity are closely linked in knowledge-intensive domains. Scientific achievements – the foundation of higher education content – are used in the professional activity and contribute to its progress. Higher education institutions train specialists both for science and professional activity. The more university graduates are aware of the latest scientific breakthroughs, the more they are in demand both in science and professional activity which is a validation and utility criterion of scientific knowledge in the last resort. It also formulates tasks to be solved by science. Finally, it sets up requirements for the training level of university graduates; the most important of which is a skill to apply obtained knowledge in practice. Progress in such domains is usually based upon ideas that help to solve problems in all the three areas – science, education and professional activity. One of the science-intensive domains where those areas are interlinked is computer program transformation.

At present the task of increasing processing power is still of current interest. The modern development of computer science is connected with new parallel architectures. With processors getting more powerful, the requirements to them are growing. The relevant software is necessary for achieving high efficiency of parallel machines. Computer architecture getting more complicated, the programming languages are also becoming more complicated. That leads to poor quality both of source programs and target ones. Therefore, to keep the gain obtained at the expense of the possibilities of new computer architectures it is necessary to optimize source programs and improve programming language compilers. A lot of problems connected with systems of program transformations still remain unsolved (for example, problems of taking context into account and choosing an order (strategy) of using transformations as any transformation used in wrong environment and at an inappropriate step may result in the deterioration of program instead of its improvement). Intensive scientific research on program transformations is to facilitate progress in high-technology professional activity – developing optimizing and unparallelizing compilers. Universities must train up-to-date specialists to ensure both scientific and professional activities in this field.

The main achievements in the three spheres linked with program transformations, major problems impeding progress in the filed and possible solutions are considered in the present paper.

Hereinafter classical transformations, restructuring transformations and transformations for parallel machines are regarded as program transformations. Classical transformations are those developed for sequential machines but also useful for parallel ones. Restructuring transformations are transformations that increase the degree of parallelism in the program at the expense of changes in the program structure. Transformations for parallel machines follow exclusively from the parallelism of the machine architecture [Evstigneev, 1996].

Some basic notions and scientific achievements in the field of program transformations are considered in the following part.

This paper was made according to the program of fundamental scientific research of the Department of Power Engineering, Mechanical Engineering, Mechanics, Control Processes, the Russian Academy of Sciences, within the framework of the project “Synthesis of intellectual systems of control over knowledge bases and databases”; according to the program of the Far Eastern Branch of the Russian Academy of Sciences (the project code is 04-3-ZH-01-003).

Modern State on Program Transformations

The necessity of program optimization was first realized almost together with the creation of the first translators of programming languages. The practice showed that raised level of a source programming language told negatively on object program efficiency obtained as a result of translating. In this connection at the end of 1950s the task to increase efficiency with the help of transformations during the process of translating was taken on. This task will be called *program optimizing transformation task*. Transformations that increase the program efficiency are called *optimizing transformations* (OTs). *The system of optimizing transformations* is a set of OTs together with the strategy of their utilization [Kasyanov, 1988].

Program models provide theoretical base for studying and substantiating algorithms of program transformations. Thus, at present there are models of both sequential and parallel programs in the field of program transformations. Program transformations are introduced in terms of program models, i.e. over schemas not programs. The transformation description consists of two parts – the description of *contextual conditions* of the transformation, i.e. conditions under which the application of the transformation is possible and preferable, and the description of the *transformation* itself that assigns what should be changed in the schema. The same transformation under various contexts can determine transformations different in their contents. Each transformation is described in terms of one model and its application, as a rule, does not exceed its limits. It makes it possible to consider the transformed schema as a source one for further transformations. The transformation context is described in terms of program model and flow analysis, which is a means of getting reliable and accurate information about program performance without its execution. There are various classifications of transformation that are reflected in catalogues and information systems of transformations. There are means of formalizing knowledge about program transformations. One can state that program transformations are a developed field of the applied science that has already accumulated/acquired/gained extensive knowledge.

The knowledge about program transformations is necessary for using it in the professional activity while developing optimizing compilers. At present, there are compilers with wide sets of transformations for most widespread programming languages and types of computers. Thus, optimizing compiler manufacturing is a practical application of program transformations.

Scientific and practical activities connected with program transformations require training relevant specialists. Some of these specialists perform practical tasks of developing optimizing compilers; others conduct research on program transformations. Thus, specialists in the sphere of optimizing compilers are still needed both in science and practice. They are trained by leading universities of Russia, the Commonwealth of Independent States (Former Soviet Union), and other countries. Knowledge of modern scientific achievements and ability to develop optimizing compilers are the essential requirements for the level of their training that is supported theoretically, methodologically, and instrumentally.

However, there are a few interwoven problems in science, practice and education in the field of program optimization that are analyzed in the following part.

Analysis of Problems and Possible Ways of Their Solving

The main problem in the sphere of program optimization science is that it is impossible to carry out computer experiments opportunely. Their goal is to define how often transformations can be applied in real programs and what effect can be achieved. The only means of conducting such experiments is optimizing compilers. However, the period between the moment of the publication of the new transformation description and the moment of the ending of realization of optimizing compiler containing this transformation is so long that the results of computer experiments with this transformation appear to be out-of-date. Besides, an optimizing compiler usually contains a wide set of transformations and built-in strategy of their application so it is quite problematic to get results of computer experiments related to one transformation (not to the whole set).

The heterogeneity of notions and models in the sphere of program transformations, impossibility to prototype optimizing compilers and their quick moral aging are important problems.

The object of transformation is modeled by a lot of various schemas that are described in different terms, which makes it difficult to select one program model when designing each optimizing compiler. Describing transformations authors use different systems of notions, which hampers reading scientific literature by optimizing compiler developers.

Optimizing compiler is an extremely complicated program system the development of which is quite labor-intensive and time-consuming. Real characteristics of such a compiler are not quite know until it is used. Usually when developing it they prototype it to determine its characteristics. However, no prototyping facilities for optimizing compilers have been proposed yet which makes the risk higher.

Finally, each optimizing compiler is a program that is difficult to modify, i.e. it is virtually impossible to change a set of transformations in it. Because of a long period of its development, the compiler becomes somewhat obsolete by the moment of its putting into operation in comparison to the current level of program optimization science and gets more obsolete during its utilization.

The above-mentioned heterogeneity of notions and models in the sphere of program transformations and impossibility to use active teaching methods in education are major problems.

This heterogeneity creates difficulties for teachers when they systematically give information on the results achieved in this sphere and for students when they apprehend it.

Training students needs acquiring logically complicated theory and gaining practical skills of using its results. Special tools are necessary to let students do practical tasks and carry out research sufficiently within the time limits set by the curriculum. However, such tools have not been designed yet. Therefore, as a rule, students try to assimilate program transformations as a theoretical discipline gaining practical skills after their graduation from the university.

Thus, in spite of significant scientific, practical and educational achievements in program transformations there are a few problems hampering their development. To solve the above-mentioned problems is a topical task. The multipurpose computer knowledge bank is used as the general concept within the framework of which these problems can be solved [Orlov, 2003a].

The following section considers Specialized Knowledge Bank on Program Transformations (SKB_PT) as the concept of program transformation information control to solve the problems [Orlov, 2003a].

Concept of Specialized Knowledge Bank on Program Transformations

The general tasks of SKB_PT are centralization of knowledge on program transformations, coordination of their processing and collective development in order to achieve the most quality and up-to-date knowledge in this sphere and facilitate its using in science, education and professional activity.

The definition of Multipurpose knowledge bank (MKB) is given in the work [Orlov, 2003a]. According to it, MKB is a set of specialized knowledge banks (SKB). SKB for support of scientific research, educational and professional activities in a domain is a resource integrating the relevant information, providing specialists and computer programs with the access to this information, and containing tools for performing those tasks of information processing the effective methods of solving them have been already developed.

SKB_PT consists of Information Content (IC), Shell of IC, Program Content (PC) and Administration Block (AB). IC contains the relevant information. Shell of IC provides computer programs with the access to the information. Editing tools are necessary to form and develop IC. The work [Orlov, 2003b] proposes the concept of universal editor of information of different integration levels (Editor of IDIL). Other tools of information processing can be added to PC as effective methods of solving the corresponding tasks are developed. AB of SKB_PT manages users and controls the life cycle of SKB_PT. A special user of SKB_PT – Administrator – performs functions of managing other users and information resources. Shell of IC can be built on; new programs can be developed for PC. A special user of SKB_PT – Supporter – is responsible for building on Shell of IC and adding new programs to PC of SKB_PT [Orlov, 2003a]. The work [Orlov, 2003c] describes general methods of realization of MKB and specialized knowledge banks.

The information contained in IC of SKB_PT includes the ontology of knowledge on program transformations, ontologies of programming languages, the ontology of Structural Program Models (SPM), the program base storing source programs represented in the ontology of SPM, the knowledge base storing knowledge about program transformations, the archive of program transformation histories.

The editing tools are the editor of the ontology of knowledge on program transformations, ontologies of programming languages and the ontology SPM, the specialized editor of database and editors of programming languages that are developed using the editor of IDIL.

The users of the editing tools are carriers of the following information: ontologies of programming languages (linguists), the ontology of SPM, the ontology of knowledge on PT (knowledge engineers), knowledge on a domain (experts) and programs (programmers).

Besides information carriers, other users solving other IC tasks (connected with program transformations) can work with SKB_PT: scientists, optimizing compiler developers, teachers, students (users solving training IC tasks), and guests (users that are allowed only to view IC).

PC of SKB_PT (i.e. service programs realized through the shell) includes editing tools of SKB_PT, Program transformer of SKB_PT, Tools visualizing program transformation histories and Code generators on different platforms. The prototyping tool for optimizing compilers (PT_OC) is also a part of PC. When entering, Program transformer gets the structural program model (the object of transformations), the data about necessary transformations from the database; when the processing is over it gives the transformed model of structural program and the information about the applied transformations – the program transformation history. The visualizing tools make it possible to view the history. Code generators on different platforms let generate object programs according to the transformed models of structural programs. The prototyping tool for optimizing compilers integrates four subsystems into one system (the optimizing compiler prototype): Program transformer, editors of programming languages, visualizing tools for the transformation history and Code generators on different platforms.

The following section considers ways of using SKB_PT for scientific, industrial and educational purposes in the field of program transformations.

Possible Usage of Specialized Knowledge Bank on Program Transformations for Scientific, Industrial and Educational Purposes

The proposed concept of specialized knowledge bank allows to support the collective development of information resources (first of all, databases) and process them with the help of computer programs.

The specialized knowledge bank can be used to support scientific research. It allows to minimize labor costs when writing scientific reviews, to make it possible to classify optimizing transformations, including new ones, to form and develop notion systems in this area, to include new transformations in the knowledge system, to promptly introduce specialists to new results, to conduct computer experiments, to present scientific results in a form convenient to use in the professional activity, to compare new results with the ones archived before.

Having decided to participate in the activity of SKB_PT, the scientist applies to Administrator of the bank for a registration. In the application he\she explains which class of users he/she wants to belong to, what tasks he/she would like to solve, informs of his/her qualification in the sphere of program transformations. After screening the application (and, possibly, consulting the scientific society in order to get approval of his/her qualification and whether granting his/her application is desirable), Administrator makes a decision to register the applicant as a bank user and informs him/her of it via e-mail. Administrator gives Editing tools to scientists and creates theories for editing in the experimental domain of Information content. Administrator ensures that the information being edited by one user will be unavailable to other users both for editing and using. After describing and modifying the given theories, scientists submit an application to store them in the effective domain of IC. Administrator analyses the theories and makes a decision whether to provide free access to them or send them to be improved.

Together with the application to open the edited theory in IC for free access, the scientist can give his/her articles (monographs, textbooks, etc.) describing it. That helps Administrator and other scientists to analyze the theory. When opening the modified theory for free access, Administrator makes a decision if it should substitute the old one (in case of its existence) or be left as an alternative. The base theories in the knowledge bank are the ontology and program transformation database [Artemjeva, 2002a] [Artemjeva, 2003b] [Artemjeva, 2003c]. Specialists can develop their own theories using the base ones as a foundation or propose their variants for storage and collective exchange.

Program transformation experiments can be conducted in the following way. When entering, Program transformer of the knowledge bank gets SPM. SPM can be designed with the help of high-level language editors and the program transforming the results of the work of the editors into the structure determined by the SPM ontology. SKB_PT Program transformer makes a control flow analysis and data flow analysis in SPM. The obtained information is a source for transformation modelling. First, Program transformer defines saving sectors of SPM on the base of contextual conditions; then it transforms SPM. Transformations are made on the base of the transformation knowledge given in the knowledge database. The transformed SPM is a result of work of SKB_PT Program transformer. Special code generators transform SPM into a representation necessary for further processing (for example, imperative one). Measuring systems on different platforms allow knowing experimental results of the transformation efficiency. Visualizing tools provide information on histories of applied program transformations.

SKB_PT can be used for prototyping of optimizing compilers. The specialist applies to Administrator of the bank for a registration. Administrator provides him/her with prototyping Tool, with the help of which the specialist integrates three subsystems of the bank: Programming language editors, Program transformer of the knowledge bank and Code generators on different platforms. The prototype developer selects a language of a number of languages having editors in the bank, a code generator on the necessary platform of a number of code generators in the bank, a set of optimizing transformations from the database and assigns the strategy of applying these transformations. If an empty set of optimizing transformations is assigned, the compiler prototype will be non-optimizing. Optimizing compiler prototyping makes it possible to research strategies and transformation sets in such compilers.

The specialized knowledge bank can be used for self-developing optimizing compilers: a set of optimizing transformations contains all the transformations from the database both in its current state and in all the future ones. Since the database is constantly modified – new transformations are added and morally aged ones are

excluded, characteristics of the compiler prototype are automatically changed according to the changes in the database.

SKB_PT can also be used for training students. Teachers can use Information content of the bank in their preparation for lectures on program transformations. Laboratory tutorials on program transformations can be given on the base of the specialized knowledge bank. Students are to prototype optimizing compilers and conduct experiments with them, to replenish the bank database. They are supposed to find new optimizing transformations in the scientific literature, to include them in the database, carry out research. Students are to use scientific publications on program transformations to find necessary knowledge to fulfill the task, to formalize the knowledge, and to put it into the database by means of the knowledge editor. To conduct the experiment the student has to select a lot of structural programs in the source programming language as an experimental material, to put these programs into the bank Archive by means of language editors and get optimized versions of the programs by means of Program transformer. Students can get transformation histories of these programs and study them using the bank visualizing tools.

Students will better understand the subject – program transformations – by solving tasks and carrying out research. The bank mission is to provide thorough feedback in training and give an opportunity to acquire practical skills in knowledge formalizing and using; the teacher's role is to estimate the final result.

Conclusion and Acknowledgements

This paper reviews the present of scientific research, professional activity and education and analyses the problems in the area of program transformations. The information resource concept based on the modern paradigm of information computer processing is proposed as an approach to the problems. The introduced resource is called Specialized Knowledge Bank on Program Transformations. The classes of users, the structure of its information and program content are described. The paper also describes the possible usage of knowledge banks in scientific research in the industry and education.

Bibliography

[Artemjeva, 2002a] Artemjeva I.L., Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Part 1. The terms for optimization object description. In The Scientific and Technical Information, 2002. (In Russian).

[Artemjeva, 2003b] Artemjeva I.L., Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Part 2. The terms for optimization process description. In The Scientific and Technical Information, 2003. (In Russian).

[Artemjeva, 2003c] Artemjeva I.L., Knyazeva M.A., Kupnevich O.A. Domain ontology model for the domain "Sequential program optimization". Part 3. Examples of optimizations transformation description. In The Scientific and Technical Information, 2003. (In Russian).

[Evstigneev, 1996] Evstigneev V.A., Kasyanov V. N. Optimizing transformations in unparallelizing compilers. Programming, 1996, № 6, pp. 12-26. (In Russian).

[Kasyanov, 1988] Kasyanov V. N. Optimizing transformations of the programs. Moskow: Nauka, 1988. (In Russian).

[Orlov, 2003a] Orlov V.A., Kleshchev A.S. Multi-purpose bank of knowledge. Technical report. Part 1. Notion and policy. Vladivostok: IACP FEBRAS, 2003. 40 p. (http://www.iacp.dvo.ru/es/) (In Russian).

[Orlov, 2003b] Orlov V.A., Kleshchev A.S. Multi-purpose bank of knowledge. Part 3. A notion of a unified idea editor. Vladivostok: IACP FEBRAS, 2003. 28 p. (http://www.iacp.dvo.ru/es/) (In Russian).

[Orlov, 2003c] Orlov V.A. Multi-purpose bank of knowledge. Part 6. Implementation details. Vladivostok: IACP FEBRAS, 2003. 28 p. (http://www.iacp.dvo.ru/es/) (In Russian).

Authors' Information:

Margarita A. Knyazeva, Alexander S. Kleshchev, – Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of Sciences; 5 Radio Street, Vladivostok, Russia

e-mail: mak@nt.pin.dvgu.ru, kleschev@iacp.dvo.ru

IMPLEMENTATION OF VARIOUS DIALOG TYPES USING AN ONTOLOGY-BASED APPROACH TO USER INTERFACE DEVELOPMENT

Valeriya Gribova

Abstract. *A new method to implementation of various dialog types using an ontology-based approach to user interface development is proposed. The main idea of the approach is to form necessary to the user interface development and implementation information using ontologies and then based on this high-level specification to generate the user interface. To combine various types of dialog (verbal and graphical) in the framework of the same interface two ontologies are suggested, the ontology of the graphical user interface and the ontology of graphical static scenes on a plane.*

Keywords*: Ontology, interface model, user interface development*

Introduction

The user interface is an integral part of most software systems. Experts note that complexity and functionality of software systems are increasing every year; at the same time the number of users with a wide range of expertise and, accordingly, requirements to software is rapidly growing. The competition at the software market is increasing, too. All these factors demand a tool capable of realizing dialog between the user and software in accordance with his or her requirements, which are subject to changes during the software life cycle.

Modern tools for user interface development – Interface Builders, User Interface Management Systems, Model-Based Interface Development Environment – are, on the one hand, only oriented to implementation of the GUI (Graphical User Interface) based on using different interface elements – menus, windows, buttons, lists, etc. On the other hand, they do not support design of all user interface components.

To solve the problems mentioned above a new ontology-based approach to user interface development is proposed. The main idea of the approach is to form information necessary for the user interface development and implementation using ontologies and then, based on this information, to generate the user interface. For implementation of different types of dialog (verbal and graphical), ontologies of the graphical user interface and of graphical static scenes on a plane have been developed. They manage design of the presentation component of the user interface and allow us to implement various types of the dialog.

The aim of the present study is to describe implementation methods of various types of the dialog - verbal and graphical - within the limits of the ontology-based approach to user interface development.

The Basic Conception of the Ontology-based Approach

Rapid software progress demands that the cost of interface development need to be decreased, and its maintenance need to be simplified, which is even more important. According to experts, for example, [1] software maintenance exceeds the cost of its development in 3 or 4 times. These requirements in full measure relate to user interface development. The user interface has an additional requirement, namely, adaptability for users with a wide range of expertise.

Taking into account the requirements mentioned above, a new approach to user interface development based on ontologies is suggested [2]. The approach is a modification of the model-based approach to user interface development [3].

The main ideas of the ontology-based approach are as follows: aggregation of uniform information in components of an interface model, formation of information for every component on the basis of the appropriate ontology model and automated the code generation according to this information.

The interface model consists of a domain model, a presentation model, an application program model, a model of a dialog scenario and a relation model.

The domain model determines domain terms, their properties and relations between them. In this system of concepts, output and input data of the application program and information on the intellectual support of the user are expressed.

The presentation model determines a visual component of the interface. It provides support for various types of the dialog.

The application program model determines variables, types of their values shared by the interface and the application program, protocols for communication between the application program and the interface, addresses of servers and methods of message transfer.

The model of a dialog scenario determines abstract terms used to describe the response to events (sets of actions, executed when an event is occurs, sources of events, modes of transfer between windows, methods of the window sample selection and so on).

The relation model determines relations between components of the interface model.

Fig. 1 shows the basic architecture of the user interface development tool based on ontologies. We show only the basic one because (the architecture) as a whole involves additional components such as design critics, advisors, automated design tools, etc. These components are not included in the basic architecture.

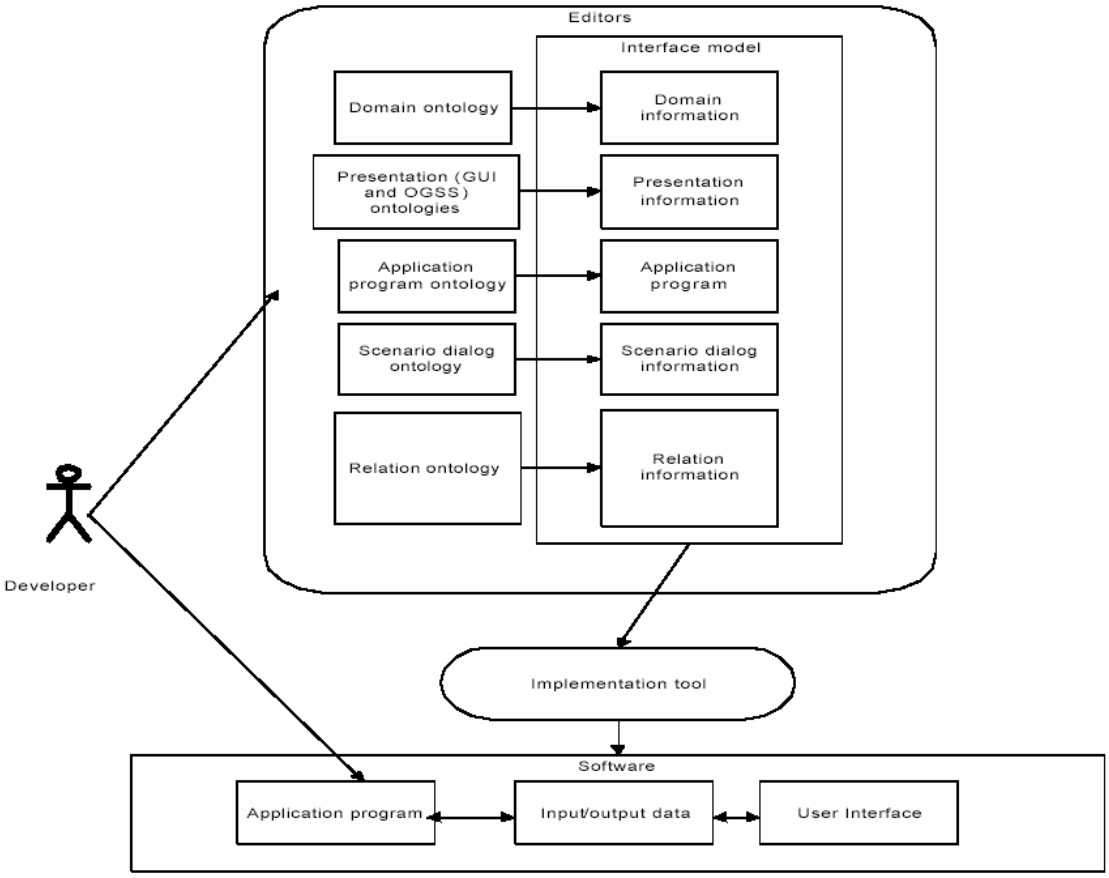


Fig. 1. The basic architecture of the user interface development tool based on ontologies

An Approach to Implementation of Various Dialog Types

When the user interface is developed, it is necessary that information representing input and output data of an application program be presented in accordance to user requirements and in forms accepted in the domain for which the software is developed.

In this case time various representation forms of information and types of dialog, for example, verbal and graphical, are often required in the framework of the same interface. Fig. 2 shows how the information can be

presented to users in various forms. According to our conception, an ontology model for creating and managing every component of the model interface is suggested.

In this way, the interface developer creates domain information. It is a verbal description of domain terms, their properties and relations with other terms. This domain information can be presented in the interface verbally or graphically in various forms depending on user's requirements to representation of information, expertise of users and on their preferences.

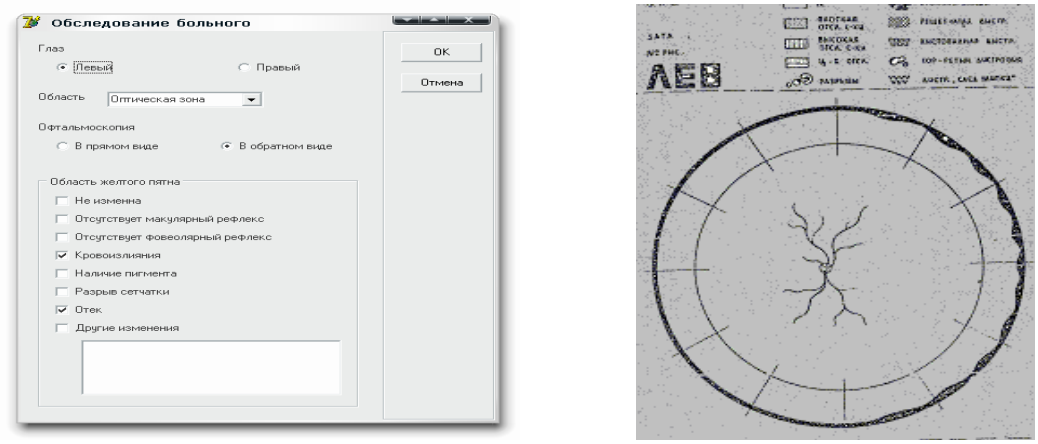


Fig. 2 Representation the same information in various forms

The presentation model is responsible for representation of a domain model. The former is the basis for visual representation of the interface.

To combine various dialog types in the framework of the same interface two ontologies are suggested, the ontology of the graphical user interface (OGUI) and the ontology of graphical static scenes on a plane (OGSS).

The OGUI is intended for presentation of information in the verbal form using interface elements – windows, menus, lists, buttons, etc. The OGUI describes interface elements, their properties and relation to one another. Interface elements permit the user to choose and install values, to start operation and also to move within the program. At present, the OGUI in the OIL language is available in the Internet [4]. The process of design information in the verbal form consists in correlating domain fragments with interface elements and specifying properties of interface elements (dimension, color, location and so on).

The design process in the form of graphical scenes is accomplished by the OGSS. For this purpose terms of the domain are associated with graphical images and their properties peculiar for a particular interface are specified. The generation of graphical scenes, their interpretation (input data for an application program) and automated construction of graphical scenes (output data of a application program are accomplished by the OGSS. Thus, the OGSS is the managing structure for organization of dialog based on graphical scenes.

To provide flexibility and simplify modification of the user interface the correlation between input and output data and term values is established on the basis of the domain model, as the output and input data are independent of the dialog type.

The Ontology of Graphical Static Scenes on a Plane

To implement dialog based on graphical scenes a domain independent the OGSS has been developed. A graphical static scene S on a plane is defined as: S = <B, F, P>, where B is the base graphic representation, F - filler, P - primitive. Fig. 3 shows the OGSS model in the URL language [5].

The base graphic representation B, further for brevity named the base, is any graphic figure, scheme, sketch, etc., being a basis for drawing various images on it. The base B consists of the following elements: B = (NmB, ImB, Db, Db′), where NmB is the base name, ImB is the base image, Db is the description of the main elements of the base, and Db′ is the alternative description of the base elements.

The description of the main elements of the base is Db={(b1,nb1), (b2,nb2),…,(bn,nbn)}. Here b1,…,bn are simple elements of the base image, such as B=b1∪b2∪…∪bn, i.e. merging simple elements forms the base

image; nb1,…,nbn - names of simple elements. The alternative description of elements of the base is Db′={(b′1,nb′1), (b′2,nb′2),…,(b′f,nb′f)}. Here b′1,b′2,…,b′f are compound elements of the base image, nb′1,…,nb′f are names of compound elements of the base image. Each compound element is some merging of simple elements of the base image, i.e.

b′1=bi1 ∪bi2∪…∪ bik, where 1≤k≤n, bi1,bi2,…,bik, are simple elements of the base image;

b′2=bj1 ∪bj2∪…∪ bjh, where 1≤h≤n, bj1,bj2,…,bjk are simple elements of the base image;

b′f=bv1 ∪bv2∪…∪ bvd, where 1≤d≤n, bv1,bv2,…,bvk are simple elements of the base image.

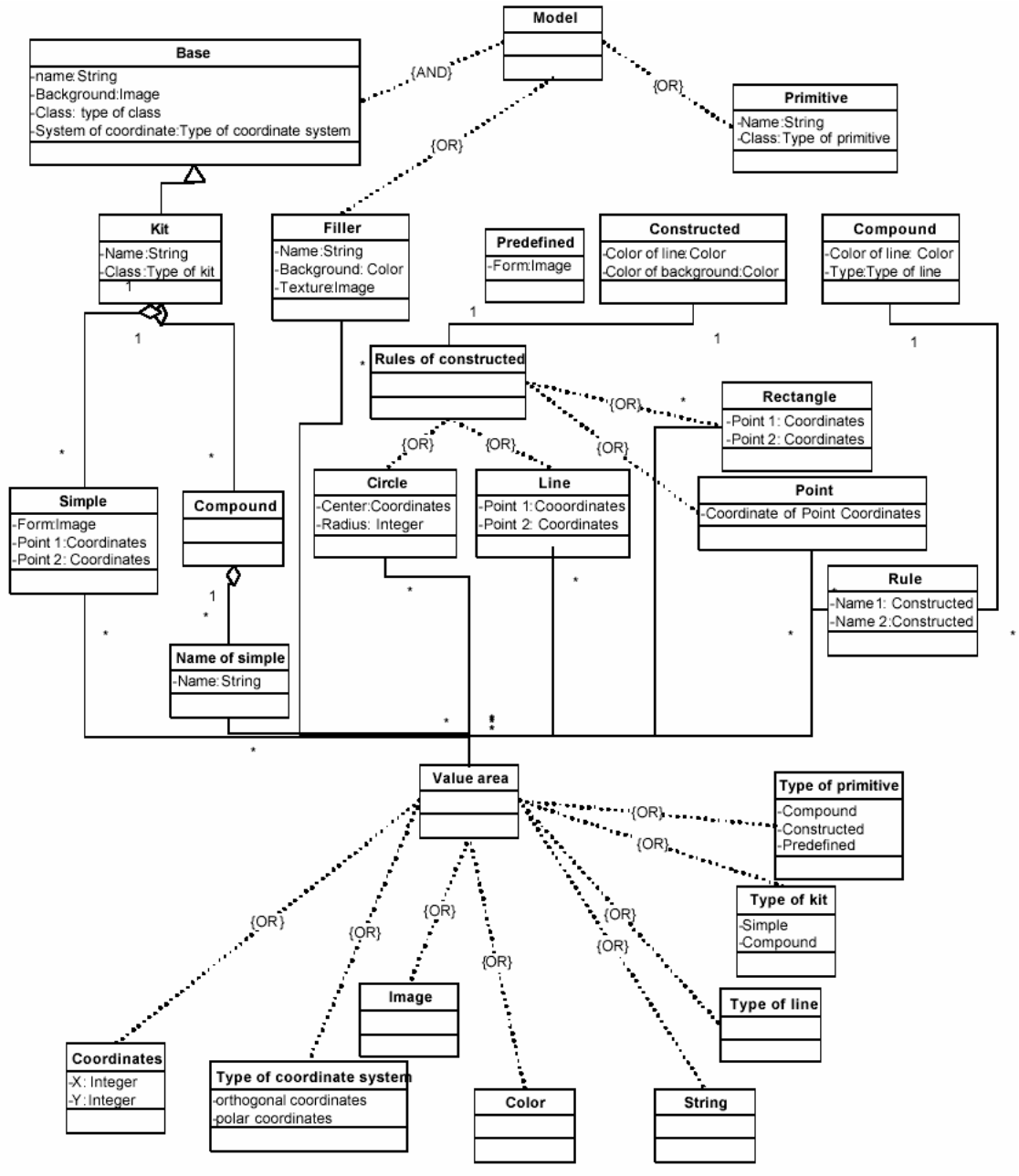


Fig. 3. The model of the ontology of graphical static scenes on a plane

The filler determines possible color and texture options for elements of the base image. A set of possible fillers can be defined for the base F={f1,f2,…,fn}. The filler is defined as fi=<Nfi, Col, Tex>, where Nfi is the name of the filler, Col is the color of the filler, Tex is the texture of the filler.

The primitive determines possible images applied on the base image. A set of possible primitives can be defined for the base: P=p1,p2,…,pn. The primitive pi is defined as: pi=<Npi, Tp, Dp(Tp)>, where Npi is the name of the

primitive, Tp is the type of the primitive, Dp(Tp) is the description of the primitive. The type of any primitive can be defined as Tr is predefined, Tb is constructed, Tc is compound.

By the predefined primitive is meant a primitive whose image is known. The description Dp(Tr) of the predefined primitive is Dp(Tr)= < Ip>, where Ip is the image of the primitive.

The constructed primitive it defined by a form, a color of a line and a background. Hence it has the following description: Dp(Tb)=< F, R(F), Cl, Cb>, where F is the form of the primitive, R(F) is the rule for construction of the primitive of a specified form, Cl is the color of the line, Cb is the color of the background. The following forms of a primitive are defined: a circle, a point, a line and a rectangle. For each form, it is necessary to define rules of its construction. The rule of a circle construction is defined by the circle center coordinates and radius: R(circle) =<(x,y), r>; the point is defined by the coordinates: R(point) =<(x,y) >; the line is defined by coordinates of two points: R(line) =<(x1,y1), (x2,y2)>; a rectangle is defined by coordinates of two points, its top left and lower right vertexes: R(rectangle) =<(x1,y1), (x2,y2)> .

The compound primitive is a set of constructed primitives, connected by themselves by lines of a certain color and type: Dp(Tc)=< {(Npi, Npj), (Npj Npk),…, (Npn Npv)}, Cl, Lt>. To describe the compound primitive it is necessary to determine a set of pairs of constructed primitive names (Npi, Npj), that must be connected by lines of a certain color Cl and type Lt.

The Design Process of Dialog in the Form of Graphical Static Scenes

The design process of dialog in the form of graphical static scenes is carried out in two phases.

At the first phase, it is necessary for the interface developer to correlate elements the OGSS with the information, specific for a certain domain. In this way, the developer defines a base, i.e. its image, name (a term of the domain), as well as names and images of base components. Further, with the same editor the developer determines fillers and/or primitives by specifying their possible properties.

At the second phase the developer forms the design of the interface, namely, specifies location of the base, primitives, fillers and additional elements of the graphical user interface determined by the OGUI.

The input data dialog of the user with the applied program consists in composing the graphic static scenes. Fig. 4 shows examples of graphical static scenes. According to the specification of the OGSS for a certain domain, the interface recognizes a graphic scene and transfers values of the output data to the applied program in the format assigned by the developer. Then the interface generates graphical scenes based on computation results of the applied program in conformity with the same description.

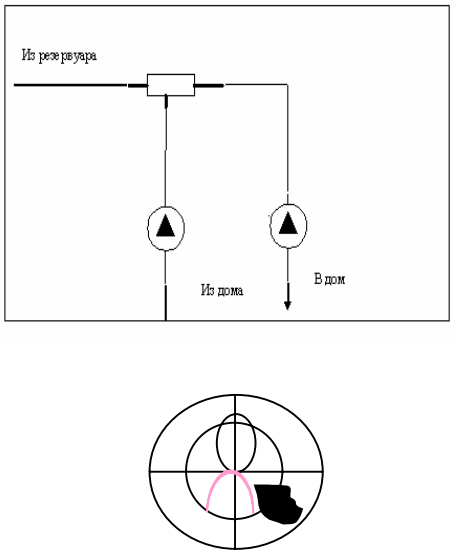


Fig. 4 Examples of graphical static scenes for heat supply and medicine domains.

Discussion

Development of the user interface combining various types of dialog (verbal and graphical) is a topical problem. No less important, as noted above, is the problem of reducing the cost of development and simplifying maintenance of the user interface. In the paper, we have considered how the proposed technology of development allows the specified problems to be solved.

First, the interface is automatically generated based on the declarative description of its model.

Second, information for each component of the model is formed on the basis of the ontology model offered to the developer.

Third, the interface developer can generate various types of dialogues according to requirements of users (verbal and/or graphical on the basis of the OGUI and the OGSS.

Fourth, information transmitted to the applied program and back (the input/output data) does not depend on the form of its representation to the user and is formed based on the domain model. Thus, modification of the dialog does not require any modification of other interface components.

Acknowledgements

The research was supported by the Far Eastern Branch of Russian Academy of Science, the grants № 05-02-01-027, 05-01-01-119.

Bibliography

1. Sommerville I. Software engineering. Addison-Wesley Publishing company,1997.
2. Gribova V., Kleshchev A. From an ontology-oriented approach conception to user interface development International //Journal Information theories & applications. 2003. vol. 10, num.1, p. 87-94
3. Da Silva P.P., Griffiths T. and Paton N.W., Generating User Interface Code in a Model-Based User Interface Development Environment, Proc. Advanced Visual Interfaces, V. di Gesu, et al. (eds), ACM Press, 2000.
4. http://interface.es.dvo.ru/ontology.htm
5. The unified modelling language. http://www.uml.org/

Author's Information

Gribova Valeriya –Ph.D. Senior Researcher of the Expert System Department, Institute for Automation & Control Processes, Far Eastern Branch of the Russian Academy of the Sciences: Vladivostok, +7 (4323) 314001 gribova@iacp.dvo.ru, http://www.iacp.dvo.ru/es.

ОНТОЛОГИИ КАК ПЕРСПЕКТИВНОЕ НАПРАВЛЕНИЕ ИНТЕЛЛЕКТУАЛИЗАЦИИ ПОИСКА ИНФОРМАЦИИ В МУЛЬТИАГЕНТНЫХ СИСТЕМАХ Е-КОММЕРЦИИ

Анатолий Я.Гладун, Юлия В.Рогушина

Аннотация. *Предлагается использовать онтологическое представление знаний об интересах покупателя в процессе е-коммерции. Это повышает эффективность поиска наиболее продавцов, наиболее подходящих покупателям, мультиагентными системами. Реализован алгоритм сравнения онтологий покупателей с онтологиями электронных магазинов (таксономий) и мультиагентная система электронной коммерции, использующая онтологии для поиска информации в распределенной среде.*

Ключевые слова: *онтология, е-коммерция, мультиагентная система.*

Введение

Сегодня мы являемся свидетелями и участниками эволюции постиндустриального общества в общество, называемое информационным. В информационном обществе приоритетным направлением является создание и эффективное использование знаний и информационных ресурсов.

Колоссальные перспективы развития рынка товаров и услуг в сети Интернет впечатляют даже специалистов - только за последние месяцы прошлого года торговый оборот сети возрос в несколько раз. Стремительный технический прогресс в этой области дает мощный импульс глобализации мировой экономики и делает все более привлекательным объектом инвестиций новые информационные технологии, направленные на развитие электронной коммерции (е-коммерции).

Однако происходящие изменения приводят к возникновению ряда новых проблем. Поистине огромный объем предложений, широкое разнообразие товаров и услуг, высокая динамика изменений рынка – все это приводит к резкому возрастанию сложности и трудоемкости работы как продавцов, так и покупателей в сети, отнимает их время и тем самым повышает стоимость данных услуг.

Нужна кардинальная смена самой концепции обработки информации в сети Интернет, которая бы позволила более содержательно отвечать запросам клиентов, более оперативно реагировать на изменяющиеся требования и гибко адаптироваться к условиям рынка.

Вхождение Украины в мировое информационное пространство требует решения многоаспектной проблемы автоматизации современных бизнес–приложений. Под электронным бизнесом понимают все формы электронной бизнес-деятельности, такие как е-коммерция, е-консалтинг, е-издательство и т. п. Е-коммерция является частным случаем электронного бизнеса. Под е-коммерцией понимают различные формы торговли товарами и услугами посредством использования электронных средств, в том числе и Интернета. При этом заказ товаров осуществляется через телекоммуникации, а расчеты между покупателем и продавцом - при помощи электронных средств платежа [1].

Улучшение эффективности выполнения задач е-бизнеса требуют дальнейшего развития методов автоматизации бизнес-процессов. Системы е-коммерции должны обеспечивать потребителю доступ к информации о товарах, представленной в электронной форме, и ее быстрый поиск в сетевой среде. Сложность транзакций очень велика из-за динамичности и огромного количества информации, доступной пользователям через Интернет. Индустриальная разработка программного обеспечения для е-коммерции требует создания и использования соответствующих моделей, стандартов, языков и форматов, ориентированных на обработку знаний. Для решения этих задач с успехом применяются агентно-ориентированные технологии, базирующиеся на использовании интеллектуальных программных агентов (ПА).

Системы поддержки е-коммерции

Сейчас для поддержки электронной коммерции разработано много разнообразных программных продуктов разного уровня сложности. Моделирование реальных приложений должно отображать сложность бизнес-процессов реального мира. Представляется очевидным, что необходима интеллектуализация таких средств, чтобы избавить пользователя от необходимости самостоятельно выполнять повторяющиеся действия. В частности, наиболее популярным на сегодняшний день является использование в е-коммерции ПА и мультиагентных систем (МАС). Одна из главных трудностей практического подхода связана с объединением сложных интеллектуальных способностей в мобильных ПА.

ПА - это программное обеспечение, обладающее рядом характерных свойств, предназначено для упрощения диалога пользователя со сложным и динамичным информационным окружением. Многие исследователи определяют ПА как программную сущность, которая функционирует продолжительно и автономно в конкретном окружении, часто вместе с другими процессами и ПА [2]. Продолжительность и автономность позволяют ПА гибко и интеллектуально действовать в соответствии с изменениям среды без постоянных указаний или вмешательства пользователя. В идеальном варианте агент, который долгое время функционирует в среде, должен быть способен обучаться на своем опыте. Кроме того, ПА, который сосуществует в среде с другими агентами и процессами, должен быть способен общаться и кооперироваться с ними [3,4].

Мы проанализировали ряд агентов ЭК. PersonaLogic [5] отфильтровывает список товаров, которые удовлетворяют ограничениям пользователя. Firefly выбирает товары через оценки других потребителей. BargainFinder [6] - виртуальный агент покупок, способный оценить цены и их пригодность для пользователя. Он представляет вопрос потребителя параллельно группе on-line продавцов, заполняя форму на каждом сайте. Kasbah [7] - on-line МАС для транзакций типа "потребитель-потребителю". Пользователь, желающий продавать или покупать товары, создает агента, задает этому некоторое стратегическое направление и отсылает его к централизованному агентному рынку. Агенты Kasbah проактивно разыскивают потенциальных покупателей или продавцов и ведут переговоры с ними от имени их создателя. Tete-a-Tete [8] используется для посредничества и переговоров, обеспечивая потребностям и продавцов, и покупателей.

При поиске продукта покупатель выбирает набор характеристик товара, которые представляют для него интерес. Если не найден товар, полностью удовлетворяющий всем требованиям, тогда либо появляется сообщение, что результат не найден, либо список товаров, частично удовлетворяющих запросу. При этом принципы ранжирования результатов поиска часто непонятны пользователю. Заказчик не получает ни четкого обзора результатов, ни предложений для дальнейшего исследования. Идеальной была бы ситуация, в которой покупателю предлагались бы альтернативы, наиболее близкие к его потребностям. Но для этого система должна обрабатывать знания о конкретном заказчике.

К сожалению, большинства существующих систем е-коммерции не обеспечивают общий язык взаимодействия, стандартизированное описание домена, адаптируемость, способность к обучению, персонализацию. Агенты е-коммерции, созданные различными разработчиками, не способны взаимодействовать друг с другом. Кроме того, использование многих терминов и выражений крайне неоднозначно и в значительной мере зависит от той предметной области (ПрО), которая интересует пользователя.

Постановка задачи

ПА, которые ищут информацию только по ключевым словам, не имеют прикладных знаний о ПрО, которая интересует пользователя, а самостоятельно извлекать эти знания могут только после продолжительной работы и поэтому дают крайне нерелевантные результаты. Поэтому необходимо найти формализованные средства представления таких знаний и разработать алгоритмы, позволяющие агентам эффективно использовать их в поиске товаров и услуг мультиагентными системами е-коммерции.

В связи с этим мы предлагаем использовать описания ПрО, которые интересуют конкретного пользователя, представленные в виде онтологий - средство построения распределенных и неоднородных систем баз знаний. Это позволяет избежать разногласий в использовании терминологии и помочь агентам установить правильные соответствия между предложениями продавцов и потребностями покупателей. При этом предполагается, что продавцы также предоставляют пользователям знания о предлагаемых товарах, представленные в виде онтологий.

Онтологическое представление знаний

Онтология - это знания, формально представленные на базы концептуализации, которая предполагает описание множества объектов и понятий, связей между ними. Формально онтология состоит из терминов, организованных в таксономию, их определений и атрибутов, а также связанных с ними аксиом и правил вывода.

Часто набор предположений, которые составляют онтологию, имеет форму логической теории первого порядка, где термины словаря являются именами унарных и бинарных предикатов, которые называют соответственно концептами и отношениями. В простейшем случае онтология описывает только иерархию концептов, связанных отношениями категоризации. В более сложных случаях к ней добавляются подходящие аксиомы для отображения других отношений между концептами и для того, чтобы ограничить их интерпретацию. Онтология - это база знаний, описывающая факты, которые предполагаются всегда истинными в рамках определенного сообщества на основе общепринятого значения словаря, который используется.

Формальная модель онтологии O - упорядоченная тройка вида: O=<X, R, F>, где X - конечное множество концептов предметной области, которое представляет онтология O; R - конечное множество отношений между концептами заданной предметной области; F - конечное множество функций интерпретации, заданных на концептах и/или отношениях онтологии O [9].

Создание онтологий покупателей

На сегодняшний день разработано довольно много языков представления онтологий (например, DAML-OIL, OWL [10]) и свободно распространяемых инструментальных средств их создания (например, Protégé [11], OntoEdit [12]).

Большие онтологии, такие как CYC, создаются на основе абстрактного и очень общего описания понятий предметной области и связей между ними. Основная цель проекта CYC - раз и навсегда построить базу

знаний всех общих понятий, включающую семантическую структуру терминов, связей между ними, правил, которая будет доступна разнообразным программным средствам. Но на практике для каждого пользователя возможен свой контекст для представления терминов в зависимости от ситуации и модели мира пользователя. Поэтому часто пользователю не нужна огромная онтология, содержащая описание всего мира.

Очевидно, что создание онтологий является достаточно сложным и трудоемким процессом. Оно требует от пользователя достаточно четкого и структурированного представления пользователя об интересующей его области и умения работать с соответствующим программным обеспечением. Кроме того, это целесообразно только в том случае, если покупатель выполняет более-менее однотипные закупки на протяжении длительного периода времени (специалисты по закупкам в малом и среднем бизнесе - B2C, B2B) и в больших объемах (например, торговые отделы и отделы снабжения крупных организаций - B2B, государственные закупки - B2G).

Продвинутый пользователь создает онтологию той области, к которой относится его заказ, и затем использует ее при поиске наиболее подходящих продавцов. Для повышения релевантности поиска пользователю необходимо описать свои знания и представления об объектах ПрО, связях между ними и правилах их преобразования, используя при этом стандартные средства создания и представления онтологий. Это обеспечивает независимость пользователя от применяемого программного обеспечения, т.к. одна и та же онтология может быть использована при работе с различными системами поддержки электронной коммерции.

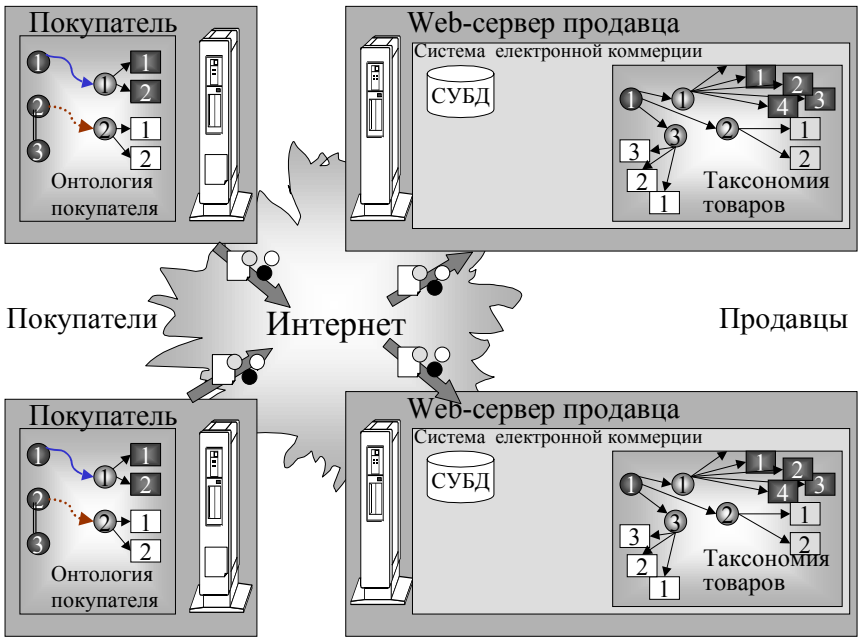


Рис.1. Использование онтологической информации при поиске покупателями продавцов

В данном подходе мы ориентируемся на то, что созданные пользователем онтологии относительно просты и компактны. Множество функций интерпретации пусто - F=∅, а R - множество отношений между концептами ПрО содержит только несколько базовых отношений ("иметь элемент", "иметь цену", "иметь свойство", "синоним", и т.п.).

Онтологии продавцов также довольно просты, т.е. даже если они содержат много концептов, то их структура достаточно стандартна. Обычно онтологии, описывающие товары, которые предлагаются в электронном магазине, - это простые таксономии - иерархические системы понятий, связанных между собой отношением “быть элементом класса”, т.е.: $\mathbf{O = T^0 = \langle X, \{"бытьIэлементомIкласса"\}, \{\} \rangle}$.

Онтологии позволяют устанавливать общую терминологию для коммуникации между пользователями (как людьми, так и программными сущностями). Запрос пользователя дополняется онтологической информацией о той ПрО, к которой он относится (рис.1).

Сравнение онтологий покупателей и продавцов

Мы использовали онтологическое представление знаний в разрабатываемой МАС е-коммерции для персонификации агентов покупателей и агентов продавцов: преимущества предоставлялись тем продавцам, в онтологиях которых было больше терминов из онтологий покупателей.

Сопоставление онтологий требует разнообразных методов, методологий и технологий, которые необходимы для эффективного использования в выполнении различных заданий онтологий, полученных из различных источников. Каждая из онтологий ПрО может охватывать определенные аспекты знания и может использовать различную терминологию. Должны быть созданы специальные онтологии отображения для связи различных терминологий и стилей моделирования, использованных в этой специфической для домена онтологии.

Сегодня существуют программные системы, предназначенные для этого. Например, в проекте Sesame обеспечивается сравнение версий онтологий, представленных в формате RDF, и анализ этих изменений. Но такое программное обеспечение слишком сложно для пользователей, не специализирующихся в ИТ.

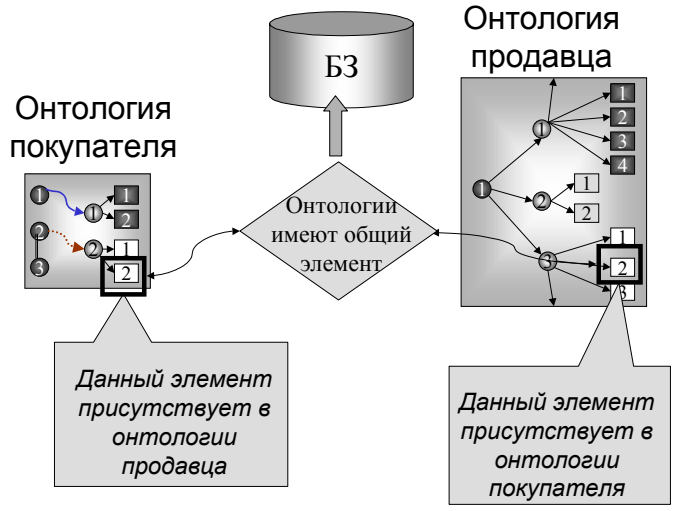


Рис.2. Предварительный шаг сравнения онтологий – поиск одинаковых элементов.

Мы предлагаем сравнивать онтологии следующим образом: для пары онтологий O1 и O2 строится оценочная функция f(O1, O2), которая определяет их меру близости. При этом учитываются следующие факторы:

- вхождение одного и того же термина в обе онтологии;
- то, что два термина находятся в разных онтологиях в одном и том же отношении;
- то, что два термина находятся в разных онтологиях в отношениях одного типа или разных (например, в иерархическом отношении и отношении синонимии);
- существуют ли вообще любые отношения (прямые или опосредствованные) между одними и теми же терминами.

На предварительном этапе сравнения онтологий (рис.2) формируется множество элементов – концептов ПрО , которые входят в обе онтологии.

$$Y = X_{O_1} \cap X_{O_2}$$

Затем проверяются попарно все отношения, в которых состоят эти элементы.

Пользователь должен определить для каждого используемого им отношения, к какой группе оно относится. В онтологии продавца такая проблема не возникает в связи с тем, что используется единственное иерархическое отношение.

Коэффициенты, определяющие вес различных факторов, зависят от специфики ПрО и определяются пользователем. В наиболее простом случае используется только первый фактор.

Программная реализация

В системе е-коммерции выделяются три взаимосвязанные подсистемы: торговля, управление диалогом и поиск товаров на основе онтологий. В торговой подсистеме могут быть агенты товаров и заказов, а также агенты продавцов и покупателей, склада, поставщиков и т.д. Агенты товаров и заказов ведут переговоры со стратегиями скидок для постоянных покупателей, скидок за оптовую покупку, скидок по состоянию конкурентов, скидок по затовариванию склада и др. При этом несколько агентов покупателей (потенциальных конкурентов) могут объединить свои заказы для получения большей скидки, т.е. перейти от конкуренции к кооперации. В подсистеме же управления диалога нужно создать систему выдачи результатов переговоров агентов. Подсистема онтологии обеспечивает быстрый поиск в распределенной среде релевантного товара для покупателя.

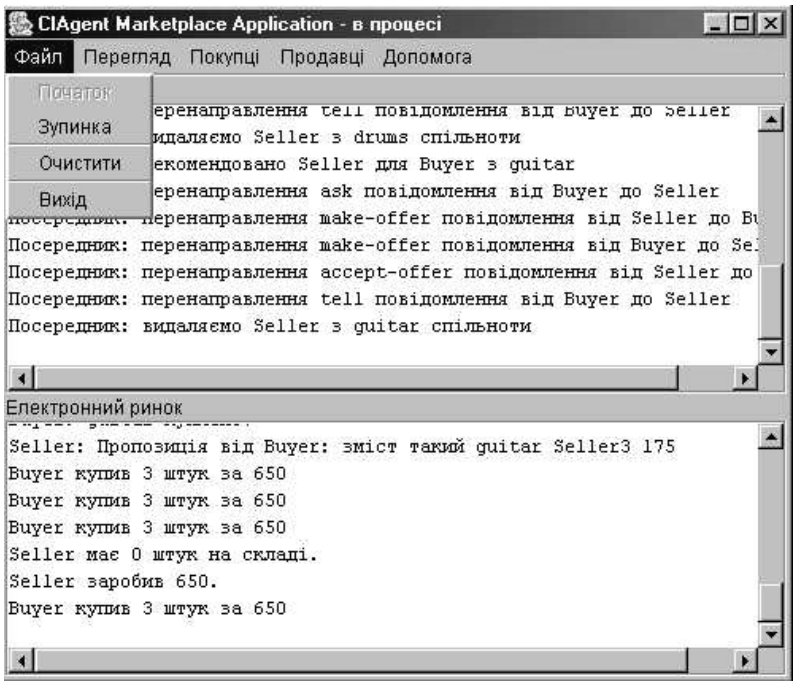


Рис. 3 Окно ведения переговоров между торговцем и покупателем через посредника

На основе анализа существующих МАС е-коммерции [5-9] мы сформулировали требования к программной реализации:

1. Обеспечение переносимости кода на различные платформы (UNIX, LINUX, Windows).
2. Доступность других платформ в сети. Это требование является продолжением предыдущего. Мобильные агенты должны осуществлять свою работу в гетерогенной компьютерной среде.
3. Поддержка сетевого взаимодействия. Помимо операций, непосредственно связанных с перемещением между агентскими серверами, агент должен обладать средствами для коммуникации с другими агентами и доступа к удаленным ресурсам. Поэтому поддержка сетевых услуг должна включать в себя широкий спектр возможностей (служба имен, RPC, OLE, CORBA, RMI и т.д.).
4. Многопоточная обработка. Для реализации одновременного выполнения нескольких действий агентская система должна включать в себя поддержку параллельного выполнения функций агента и поддержку средств синхронизации.
5. Безопасность. Мобильные агенты, приходящие из сети, могут содержать потенциально опасный, вредоносный код. Поэтому система должна поддерживать средства безопасности, достаточные для ее нормальной работы.

Для разработки логической модели МАС использовался язык UML. Структура МАС включает: модуль интерфейса; модуль функций для взаимодействия с пользователем (обработчик событий); главный модуль МАС – модуль координации и управления (в соответствии с поставленной задачей) и модуль

онтологий для работы с данными (сортировка, фильтрация, поиск и т.д.); модуль возврата результатов пользователю (в виде log-файла - сообщений на интерфейс пользователя).

На основе Java разработан FacilitatorAgent (агент–посредник), который управляет рынком, а также агенты BuyerAgents (агенты–покупатели) и SellerAgents (агенты–продавцы), используемые для взаимодействия внутри этого рынка. Все эти интеллектуальные агенты получены из базового класса CIAgent, который детально описан в [14]. Агенты клиента и продавца различаются, прежде всего, сложностью их стратегий переговоров. Переговоры начинаются с простой логики (в терминах if-then-else), а затем переходят к методам формирования правил, которые базируются на приобретенных фактах.

Язык KQML конкретизирует формат и содержание взаимодействий между продавцом и покупателем. Процедура BuySellMessages описывает переговоры между продавцом и покупателем в процессе сделки на рынке. Покупатель и продавец никогда не общаются непосредственно, а используют для этого FacilitatorAgent (которого иногда называют агентом брокера) как посредника в переговорах о купле-продаже. Менеджер коммуникаций (BaySelMessage) содержит в себе сообщения, которые должны быть посланы другим агентам, представленные на языке коммуникаций с примитивами на языке KQML: *обратиться с просьбой*, *принять*, *отвергнуть*, *изменить*, *предложить*, *проинформировать*, *запросить данные*, *отказаться* и *подтвердить*. На рис.3 представлено окно МАС, предназначенное для ведения переговоров между торговцем и покупателем в процессе процедуры купли-продажи.

Заключение

Представляется целесообразным использовать онтологическое представление знаний в электронной коммерции для автоматизации установления общего словаря для покупателей и продавцов. Применение стандартных форматов для представления онтологий обеспечивает их интероперабельность и возможность их повторного использования для решения других задач.

Предложенный в статье подход значительно повысит релевантность поиска и позволит найти наиболее выгодные и соответствующие потребностям пользователя предложения продавцов. Кроме того, это стимулирует у продавцов создание описаний их товаров на семантическом уровне, что само по себе - еще один шаг по преобразованию Интернет в глобальную распределенную БЗ.

Реализован алгоритм сравнения онтологий покупателей с онтологиями электронных магазинов (таксономий). Разработана МАС для е-коммерции, использующая онтологии для поиска информации в распределенной среде. Модуль принятия решения в МАС построен с использованием теории нечетких множеств (сочетание числового и лингвистического подходов). Алгоритм принятия решения позволил выделить три группы агентов в системе по уровню их “интеллектуальности”.

Описан протокол переговоров сбыта, который предоставляет более широкие возможности контроля над процессом продажи. Как и в других областях, в е-коммерции повышение эффективности непосредственно связано с использованием знаний и их интероперабельностью.

Представленная МАС может использоваться как для моделирования ситуаций, связанных с рынком, так и для разработки готового программного продукта не только для е-коммерции, но и для других бизнес-приложений, например, для электронного документооборота в корпоративных системах.

Перспективы дальнейших исследований

Упорядочение онтологий очень важно в контексте Semantic Web. Semantic Web предоставит нам большое количество свободно доступных онтологий, специфических для разных доменов. Чтобы сформировать реальную семантическую сеть, которая позволит компьютерам комбинировать и выводить неявное знание, эти отдельные онтологии должны быть упорядочены и связаны.

Формирование онтологии - трудная задача, которая требует углубленного знания домена и, в большинстве случаев, специальных навыков из области инженененерии знаний. Для того, чтобы облегчить процесс конструирования онтологии, необходимо разрабатывать методологии, которые позволят автоматизировать извлечение структурированного знаний пользователей об области их интересов.

Еще одни важным направлением для дальнейших исследований представляется разработка более выразительных средств представления онтологий – как языков, так и программного обеспечения.

Список литературы

1. Gladun A., Rogushina J. Multiagent Ontology-Based Intelligent System Of E-Commerce // Proceedings of Int.Conf. TPSD`2004, Kiev. - P.55-58.
2. Rao A.S., Georgeff M.P. Modelling rational agents within a BDI-architecture. In R. Pikes and E. Sandewall, eds.. Proc. of Knowledge Representation and Reasoning (KR&R-91), Morgan Kaufmann Publishers: San Mateo, CA, April 1991. - P. 473-484.
3. Bratman M. E., Pollack M. E. Toward an architecture for resource-bounded agents. Technical Report CSLI-87-104, Center for the Studv of Language and Information, SRI and Stanford University, August 1987.
4. Cohen P.R., Levesque H.J. Intention is choice with commirment. Artificial Intelligence. N 42. 1990. - P.213-261.
5. PersonaLogic. - http://www.personalogic.com.
6. Firefly.- http://www.firefly.com.
7. BargainFinder. - http://bf.cstar.ac.com/bf.
8. Kasbah. - https://kasbah.media.mit.edu.
9. Tete-a-Tete. - (http://ecommerce.media.mit.edu/tete-atete.
10. OWL. Web Ontology Language. W3C. - http://www.w3c.org/TR/owl-features/.
11. Protégé. - http://protege.stanford.edu/ontologies/ ontologyOfScience.
12. OntoEditTM Datasheet. - http://www.ontoprise.de/customercenter/index.html.
13. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. - Спб.: Питер, 2001.
14. Gladun A., Gritsenko V. Multi-Agent System Model For E-Business And Its Computer Software Implementation Technology // Problems of Programming, 2004, № 2-3, P. 510-520.

Информация об авторах

Гладун Анатолий Ясонович – к.т.н., с.н.с. Международного научно-учебного центра информационных технологий и систем НАНУ и МОН Украины

Адрес для переписки – 01033, Киев, пр. Ак.Глушкова, 40, тел.(044)266-63-44, E-mail: glad@irtc.kiev.ua

Рогушина Юлия Витальевна, к. ф.-м.н., с.н.с. Института программных систем НАНУ

Адрес для переписки – 01033, Киев, пр. Ак.Глушкова, 40, тел.(044)268-46-98, E-mail: _jjj_@ukr.net

IMPLEMENTING SIMULATION MODULES AS GENERIC COMPONENTS

Anton Kolotaev

***Abstract**: In this paper generic programming techniques applicability to build simulation models library is discussed. Policy-based design is proposed for implementation simple simulation modules.*

***Keywords**: generic programming, library design, discrete-event simulation, policy-based design.*

Introduction

Modular decomposition of a simulation model into a set of interacting components is generally accepted method and has a long history. Advantages of modular approach are well known. First of all, it allows reflecting the structure of a model being simulated into the simulation program code that undoubtedly helps in the program comprehension and makes its maintenance easier. Secondly, the approach gives opportunity for using a module as a building block for many simulations that is for module reuse. It greatly reduces new simulation models development and debugging costs. The more complex a model constructed the more evident modular approach advantages become.

One of the most widely used languages to develop simulations of large and complex systems (such as telecommunication systems) is C++. This is due to flexibility and efficiency of the language.

One who develops a simulation program in C++ can use all power of the general-purpose programming language with its highly developed machinery to build abstractions that proves to be very useful when dealing with complex data structures and non-trivial algorithms that arise in complex systems simulations.

C++ programming language allows creating highly expressive programs which executables are efficient as analogous programs written in lower level languages. When using C++ one can minimize penalty due to introducing abstractions and writing the program on a higher level that is to eliminate *abstraction penalty*. Execution slow down is acceptable in case it takes several seconds and efficiency could be sacrificed to easier model development. Simulations of large telecommunication systems may take hours of execution and a tool that introduces no abstraction penalty is very relevant to simulating large systems.

General-purpose languages (including C++) don't offer any ready components to build simulation programs. A platform providing simulation modules coordination have to implemented as well as auxiliary components that are often used when building simulations (for instance, various random number generators with certain distribution laws).

Simulation libraries provide frameworks and auxiliary components to build simulations. Some libraries are delivered together with simulation model libraries for certain domains, e.g. ns-2 simulator [1] is delivered with large number components that simulate network routers and links. Several libraries come with tools that help in conducting simulation experiments. For example, OMNeT++ [2] contains graphical module editor GNED, running the simulation environment Tkenv, Plove for analyzing simulation results etc.

To design a good library is more challenging task than to design a single application. One of the hardest tasks is to design a library so that it could evolve without making to modify a code that uses the library.

A simulation model library should possess provide interoperable, customizable components. Invalid component parameter combinations as well as composing components which interaction is not permitted should be detected as soon as possible (it is advisable to detect and localize such situations before simulation execution).

Simulation model library should be open for interaction with libraries in other domains (e.g. one may need graph library to deal with network topology; modern random number generators are made as standalone libraries, e.g. Boost.Random [3]).

Extensibility is an important feature of a simulation library. The library user has to be given an opportunity to create own simulation model components easily. Such components should be interoperable as native for the library ones.

The main method to achieve these qualities is abstracting a component that could describe in simplified form as a factoring any non-relevant points as the component's parameters. Binding with the parameters various values, we can achieve different behaviors of the component. In object-oriented languages, the main way of such parameterization is applying dynamic polymorphism, which is often presented by virtual functions language feature.

Unfortunately, such parameterization by OOP methods can be performed only to a certain extent, when virtual function usage overhead becomes unacceptable high. The main impact on performance lies in compilers' inability to inline the virtual function body into the invocation point that prohibits many optimizations around the invocation point. In addition to the loss of performance type safety may be damaged (non-typed containers in Java and C# is illustrating example) and many checks that could have been performed at compile-time cannot be performed until run-time. Bounded dynamic polymorphism requires classes which to be used as parameters to be inherited from some base classes (so called intrusive, or invasive polymorphism) that reduces the components' adaptability (we may mention primitive types boxing/unboxing in Java and C#).

The issues stated could be solved if parameters' binding is made at compile-time, so called static polymorphism, which is supported in C++ language by very powerful template machinery.

Since language means providing static polymorphism differ from ones providing dynamic polymorphism by abstractions construction properties, design with static polymorphism requires following to principles that differ from OOP principles. Such principles are studied in the field of generic programming.

According to [4], Generic programming is a sub-discipline of computer science that deals with finding abstract representations of efficient algorithms, data structures, and other software concepts, and with their systematic organization. The goal of generic programming is to express algorithms and data structures in a broadly adaptable, interoperable form that allows their direct use in software construction.

There are some domains for which generic programming libraries are developed and widely used: linear data structures (dynamic array, list, deque, balanced tree) and algorithms on them – STL, algorithms and data

structures for computational geometry – CGAL (Computational Geometry Algorithm Library) [5], graphs – BGL (Boost Graph Library), matrices – MTL, GMCL (Generative Matrix Computational Library), arrays for numerical computing – blitz++, iterative numeric methods – ITL++ etc.

OMNeT++ library design makes clear-cut distinction between simple and compound modules in simulation models library. Simple modules contain the algorithms in the model. All model behaviors are a composition of simple modules' behavior. Compound modules govern such composition. They are composed of other modules (it doesn't matter whether they are simple or compound) and are to bind aggregated modules parameters and input and output ports. The valuable idea is that all behavioral aspects of a model are contained in simple modules while all combinatorial aspects are contained in compound modules.

In this paper generic programming techniques to implement, simple modules are discussed. For this purpose, we will consider applicability of policy-based design described in [6] in depth.

We are aimed to achieve the following design goals:

1. Simple module should be as stable to changes in simulation library architecture as possible.
2. Models composed from library modules should not be less effective in terms of execution time and memory consumed than monolithic models.
3. Simulation modules should be very customizable and allow every reasonable customization.

The techniques to be described are used telecommunication systems simulation library Tksym developed by the author for comparative analysis of various routing algorithms.

Language Features to Capture Variability

Firstly, we should select appropriate language mechanism to capture variability of simple modules. For throughout discussion of concepts about variability in software the reader is referred to [4].

Object-oriented programming offers two design patterns to capture behavioral variability namely Strategy and Template Method design patterns [7].

Let's assume that we have component C with variable features $f_{1, ..., } f_N$. When using Strategy design pattern each feature f_i will be presented in C by a pointer to abstract base class F_1. During run-time, the user binds the pointer with concrete strategy object. When N is quite big there is memory overhead to store N pointers. On the other hand, the user needs not to define new classes he/she only configures at run-time the component.

When using Template Method class C has virtual functions $F_{1,} F_N$ for each variable feature. The users bind the component with certain set of concrete features by defining a class that inherits from C and implements member functions $F_{1,} F_N$. In this case, memory overhead is fixed – storing virtual functions table pointer per class object. User has to implement a new class each time he/she needs to instantiate C with new feature combination. Without templates, it is very painful task.

When we move from virtual functions to class templates more variability schemes arise.

We may parameterize C by N template parameters $F_{1,} F_N$. There are several strategies to access from C to concrete features. Firstly, concrete features may be implemented as static member functions in F_i. There is no time or memory overhead but this approach doesn't allow associating some state with concrete features. Alternative approach is to aggregate in C concrete feature objects of type F_i. Features may be accessed as ordinary member functions. Such approach has the following disadvantage: some memory is needed to store each object even it has zero size. The disadvantage can be avoided by inheriting class C from features $F_{1,} F_N$. If the compiler used supports Empty Base Class Optimization (EBCO), empty base classes will contribute no memory overhead to class C. This resembles Strategy design pattern except it is done at compile-time.

Instead of parametrizing C by N parameters we may aggregate them as nested types to a single type which will be the parameter of C.

We may parameterize C by a class derived from it. It is called *curiously recurring template*. Features are accessed from C by casting this pointer to derived class. It is very helpful technique in many cases. Suppose there are set of elementary functionalities $E = \{ e_{1, ..., } e_N \}$ and there are set of concrete classes $C = \{ C_{1, ..., } C_M \}$ and a concrete class F_i has functionality obtained by combining some subset of elementary features. An elementary feature embedded in concrete component might want to have access to other elementary features that constitutes the component. Curiously, recurring template trick is elegant solution to this design problem.

Unfortunately, it is not advisable to apply it to simulation module construction (it was the main method for combining elementary functionality in earlier versions of Tksym). Elementary functionality assume existence certain member functions in derived class that restricts the method scalability (Function member name collisions may appear as the concrete component is getting more complex and consisting more and more base classes. To solve such collisions the user has to introduce complicated class hierarchy that greatly reduces the program understanding).

Policy-based design is chosen to implement simple modules in Tksym.

Implementing Simple Server Module in Policy-based Design

Let's consider policy-based implementation of simple server *s* that behaves according the following algorithm.

When an entity *e* arrives into *s*, check that *s* is free is performed. If *s* is busy then *e* is stored in a queue *q* associated with *s*. If *s* is in idle state, *s* starts processing *e* for some time *t*(*e*). After the processing has complete, *e* is to be sent further and *s* asks *q* is there are any entities to process. If *q* contains such an entity it is extracted from *q* and is sent for processing to *s*. If *q* is empty *s* returns to idle state.

```
template
<
     class Base,
     class World,
     class Queue,
     class ProcessingTime,
     class Sink,
     class Events
>
struct Server : Base, World, Queue, ProcessingTime, Sink, Events
{
     typedef Base::Entity   Entity;

     using Queue::queue;
     using Events::events;

     // input port of the module where it receives messages from other modules
     void process (Entity e)
     {
         if (being_sent_)              // if the server in busy state
             queue()->push(e);         // store e in queue
         else                          // else
             startProcessing(e);       // start processing it
     }

     // external clients may want to know state of the server
     Entity const & beingSent() const { return beingSent_; }

private:
     void startProcessing(Entity e)
     {
         // evaluate time t to process e
         // and schedule to call "release" after t seconds
         World::schedule(ProcessingTime::get(e),
             boost::bind(&ResourceEx::release,this));

         // go to busy state
         being_sent_ = e;

         // notify event listeners processing of e has started
         events()->OnStartProcessing(e);
     }

     void release()
     {
         // place entity has been processed to output port
         Sink::process(being_sent_);

         // notify event listeners that processing of e has finished
         events()->OnStopProcessing(being_sent_);

         // go to idle state
         being_sent_ = 0;

         // if there are any entities to process
         if (!queue()->empty())
         {
```

```
                // start processing of the first one
                startProcessing(queue()->top());
                // remove it from the queue
                queue()->pop();
            }
        }

    private:
        Entity being_sent_;
    };
```

At the first place, It is to be noted that template version of Server requires much less from the component's parameters than if it were designed with virtual functions. Virtual function usage requires exact matching of the functions' signatures. Template requires from parameters to have members that have parameters list that can be matched to call arguments (that is broader since it includes, for instance, type conversions).

For example, the requirement on parameter Base is to contain inner type Entity. Type Entity must have default constructor, copy constructor and cast operator to a type that can be condition in is-statement. An entity constructed by default designates empty entity and serves to indicate idle state of the server.

There are syntactic and semantic requirements. If syntactic requirements are violated a compile-time error is reported. Semantic requirement violation may be detected only at run-time by assertions, pre- and post-conditions checking.

Let's discuss some features of the policy-based design.

A component is derived from a set of policies that are template parameters. If policy has several member functions they are composed into single class (see, for example Queue and Events policies). It helps to implement policy classes, which delegate their functionality to other classes.

Each of the class parameters can be implemented in different ways.

For example, class Server can operate with different queues. They may differ by queuing discipline: simple queues with principles first come – first served (FCFS_QueueHolder) or first come – last served (FCLS_QueueHolder), priority queues for which comparison criteria is to be provided (PriorityQueueHolder).

```
template <class QueueStorage>
    struct QueueHolder
{
    typedef QueueStorage    queue_type;

    queue_type  const & queue() const { return queue_; }
    queue_type        & queue()       { return queue_; }

private:
    queue_type      queue_;
};

template <class Entity>
     struct FCFS_QueueHolder : QueueHolder <std::queue<Entity> >
{};

template <class Enity>
     struct FCLS_QueueHolder : QueueHolder <std::stack<Entity> >
{};

template <class Entity, class ComparisionCriteria>
     struct PriorityQueueHolder :
        QueueHolder<std::priority_queue<Entity, ComparisionCriteria > >
{};
```

Server may use exclusive queue (classes that make use of QueueHolder) or shared queue (QueueInDerived or QueueIndirect).

```
// We assume that Derived has member function QueueType & getQueue()
// and QueueInDerived is base (perhaps, indirect) for Derived
```

```
template <class QueueType, class Derived>
    struct QueueInDerived
{
    typedef QueueType  queue_type;
    queue_type const & queue() const
       { return static_cast<Derived const *>(this)->getQueue(); }
    queue_type       & queue()
       { return static_cast<Derived       *>(this)->getQueue(); }
};
// accessing a queue placed somewhere else
template <class QueueType, class Holder>
    struct QueueIndirect
{
    typedef QueueType  queue_type;
    void setHolder(Holder * h) { holder_ = h; }
    queue_type const & queue() const { return holder_->queue(); }
    queue_type       & queue()       { return holder_->queue(); }
private:
    Holder     * holder_;
};
```

Note, since Queue should expose only queue() method there is no need to implement 4 queue_type members (push, pop, top, empty) in every delegating queue.

A queue may have infinite or limited capacity. In case of limited capacity, several strategies are possible when an entity is being inserted into a full queue: to ignore the entity, to replace another entity, to move the entity to special handler. Each of the options can be easily implemented as a strategy class.

Conclusion

We have discussed simple module implementation using policy-based design. We have made server class highly customizable without sacrifying its efficiency. We have minimized the module's dependency from simulation platform (The only call to the simulation kernel is World::schedule with nullary callable entity as second argument). A question that remains open is how to compose together different modules, or in OMNeT++ classification how to create compound modules. It requires some kind of metaprogramming: static metaprogramming in C++ or writing special generator or even dedicated CASE tool that supports generic components description. We suggest that compound modules should be binders – metafunctions that bind several parameters with concrete values leaving the rest ones free. Currently in Tksym is being used manual component configuration that looks very awkward and ways to implement compound elegantly are being searched.

Bibliography

[1] The Network Simulator - ns-2. Home page. http://www.isi.edu/nsnam/ns/
[2] OMNeT++ Home page. http://www.omnetpp.org/
[3] Boost Random Number Library Home Page. http://www.boost.org/libs/random/
[4] K. Czarnecki and U. Eisenecker. Generative Programming: Methods, Techniques, and Applications. Addison-Wesley, 1999.
[5] Computational Geometry Algorithm Library Home Page. http://www.cgal.org
[6] Alexandrescu, A. Modern C++ Design: Generic Programming and Design Patterns Applied, Addison-Wesley Professional, 2001.
[7] Gamma E., Helm R., Johnson R., Vlissides J. Design Pattern. Addison-Wesley Professional; 1995.

Author's Information

Kolotaev Anton – SPIIRAS, Ph.D student, Saint-Peterburg, V.O. 13th line, 39, Russia;
e-mail: Anton.Kolotaev@transas.com

ИСПОЛЬЗОВАНИЕ SEMANTIC WEB ТЕХНОЛОГИЙ ПРИ АННОТИРОВАНИИ ПРОГРАММНЫХ КОМПОНЕНТОВ

Михаил Рощин, Алла Заболеева-Зотова, Валерий Камаев

***Abstract**: В данной статье описывается принципиально новый подход при аннотировании компонентов с использованием логического формализма.*

***Keywords:** Semantic Web, компоненты, моделирование, семантика, логический формализм.*

Вступление

Цель нашей исследовательской работы заключается в содействии создания программного обеспечения с использованием заранее созданных компонентов, предоставляя семантически полное описание этих компонентов, совместно с методами и техническими приемами для управления и манипуляции ими, внедряя Semantic Web технологии в так называемую структуру аннотирования (которая является универсальной программной системой, предоставляющей семантическое описание компонентов по открытым гибким и централизованным принципам, адаптированным к стандартам системных решений). При этом структура аннотирования приобретет новые свойства и технические возможности для лучшего решения задач логического вывода (например, проверка на непротиворечивость информации), для автоматизации процессов извлечения знаний на основе содержания описания, для обеспечения автоматизированного поиска и конструировании сложных запросов, а также для реализации механизма получения информации и ее интерпретации с различных точек зрения.

Основная часть

В настоящее время, компоненты и их использование являются ключевой проблемой в области создания программного обеспечения. Системы, основанные на компонентах, легко подаются пониманию, построению, системной разборке, в отличии от монолитных систем. Связующие технологии (middleware), ассоциирующиеся с компонентами (CORBA, COM, JavaBeans, и т.д.), обеспечивают стандартизированные независимые решения для взаимосвязи компонентов, способствуя приближению программного обеспечения к стандартам plug-and-play и лучшему повторному переиспользованию относительно системных решений. Кроме того, эти технологии предлагают такие модели компонентов, которые четко соответствуют решению проблемы развития системы. В результате, переиспользование компонентов приобретает все большую популярность в различных проектах.

На ранних этапах переиспользованию компонентов придавалось чрезмерное внимание, но подход был слишком упрощенным. Сейчас, сфера практического применения и схема повторного использования, принятые в области технологии системных решений (с соответствующими средствами и методами поддержки создания ПО), делают повторное использование компонентов и развитие системных решений достаточно эффективным. Проведенный анализ подтверждает, что основой этого успеха является соответствующее описание семантической информации о компоненте (например, функциональные и нефункциональные свойства компонента, его поведение, временные требования и ограничения, необходимое обслуживание и т.п.).

Стандартная структура аннотирования является универсальной программной системой, которая позволяет описывать компоненты (создавать аннотации, аннотировать) и интерфейсы, как отдельных компонентов, так и их групп. Аннотации могут быть представлены на протяжении всех фаз жизненных циклов компонентов – от предъявления требований к дизайну и его разработке до маркетинговых шагов и внедрения. Для того, чтобы упростить процесс аннотации и обеспечить адекватность ее содержания, аннотация предоставляется на основе модели. Модель аннотации определяются соответствующей проектной группой. Когда тип описания определен, необходимо определить форму, содержание и семантику понятий, используемых в аннотации. Аннотации используются в развитии области системных

решений, чтобы выделить наиболее подходящих кандидатов среди компонентов для включения их в разрабатываемую системы, а они в свою очередь содержат информацию необходимую для их интегрирования. Аннотации могут также служить средством достижения автоматизированного конфигурирования системы.

Параллельно с дискуссиями о повторном использовании программного кода и развитии компонентно-ориентированного подхода, демонстрирующие необходимость семантического описания компонентов, Semantic Web технологии получили свое развитие в Internet. Semantic Web означает ассоциирование семантики с web информацией, которую в свою очередь можно использовать для развития интеллектуального программного обеспечения, собирающего информацию из различных независимых источников в Internet и автоматически интегрирующего эту информацию, несмотря на то, что значение этой информации в зависимости от месторасположения и групп пользователей может быть представлено по разному, что заранее не оговаривается. Основная цель Semantic Web заключается в описании различных специфических областей знаний с помощью онтологий, другими словами получение знаний и характеристика области специфических отношений и правил. Последним стандартом для описания онтологий, предложенным W3C, является язык описания web онтологий OWL (Web Ontology Language), который основан на синтаксисе XML. OWL базируется на логическом формализме Description Logics (Логика Описаний), который предназначен для описания правил, применяемых для различных задач логического вывода. Идея Semantic Web обеспечивает наилучшие результаты, когда речь идет о сравнении и соотнесении информации из различных источников, в отличии от простого поиска по ключевым словам, который теряет всякий смысл там, где объем информации растет экспоненциально. По Semantic Web технологиям доступно большое количество результатов исследований, но до сих пор никто не обращал внимания на особенности и специфические требования по семантическому описанию компонентов для повторного использования.

Очевидно, что подходы Semantic Web и желаемые механизмы расширения структуры аннотирования имеют одинаковые задачи. При соответствующем подходе задача заключается в решении проблемы манипуляции аннотациями в структуре аннотирования путем добавления логического формализма для представления знаний (используя синтаксис OWL вместо XML) и механизма онтологий в качестве основы моделирования области знаний. Этот подход обеспечит возможность автоматизированного суждения о свойствах компонентов (анализ, проверка на непротиворечивость, извлечение информации и ее представление в различных формах). В дальнейшем, это обеспечит поиск компонентов (что очень важно для переиспользования компонентов), основанном на механизмах логического вывода, путем сравнения и наложения предварительных и окончательных условий на семантическое описание компонентов.

До сих пор отсутствовали какие-либо исследования в области использования технологий Semantic Web для аннотирования компонентов в области систем программного обеспечения.

Стандартная структура аннотирования предлагает описание компонентов, основанное на моделях и интерфейсе, с точки зрения сложно описываемых свойств, таких как поведение или нефункциональные свойства, требования. Введение технологии Semantic Web позволит усовершенствовать все характеристики, благодаря механизмам, которые содействуют появлению ряда важных новых свойств. Они включают в себя следующее:

а) Систематизация описания области знаний:

- Это позволит разработать онтологии специфических областей знаний, написанных с помощью OWL, предоставляя формальное определение классов, ролей и отношений между ними, и соответствующие аннотации. Это обеспечивает создание иерархической структуры (определение классов с помощью других классов и ролей) и определение свойств ролей (например, симметрия, транзитивность и т.д.).
- Онтологии облегчают определение типов аннотаций и, соответственно, предоставляют лучшую структуризацию содержаний аннотаций и более четкое выражение когнитивных понятий.

б) Автоматизированные процедуры:

- Автоматизированная мотивация на основе знаний извлеченных из аннотаций (т.е. свойства компонентов) для многочисленных целей, в частности для получения адекватного компонента в зависимости от желаемых свойств, которые не поддаются точной спецификации. Это обеспечит отбор компонентов и их наложение или сравнение с выбранной схемой моделирования ПО.
- Автоматизированный поиск компонентов с помощью аннотаций. Подобный поиск необязательно должен быть ограничен одной областью знаний (благодаря способности сравнивать семантику).
- Использование и адаптация уже известных семантических описаний. Это используется для извлечения и реорганизации содержания аннотации, в частности при использовании различных языковых групп (например, язык маркетологов и разработчиков).

в) Логический формализм:

- Включение дополнительной информации (например, формальное описание дизайна ПО с помощью UML) в процесс поиска, основанного на аннотациях.
- Обеспечение автоматизированной конфигурации компонентов.
- Проверка аннотаций на непротиворечивость.
- Трансформация аннотаций из одной области описания в другую.

Все эти новые качества будут содействовать уменьшению стоимости разработки и внедрения программного обеспечения основанного на компонентах.

Заключение

Предлагаемый подход при аннотировании программных компонентов предполагает использование логического формализма, что само собой подразумевает нетривиальные решения проблем логического вывода и выразительности языка описания. В данный момент мы работаем над созданием метамодели на верхнем уровне абстракции и проводим опыты для подтверждения предложенных решений. В частности мы пришли к выводу о необходимости ввода механизма так называемого «проблемно-ориентированного логического вывода», подразумевающего разбиение семантического описания на ролевые группы и использование соответствующих механизмов логического вывода, в зависимости от используемых логических операторов.

Информация об авторах

Рощин Михаил Александрович – аспирант Волгоградского государственного технического университета, roshchin@gmail.com

Заболеева-Зотова Алла Викторовна – доцент кафедры САПР и ПК, Волгоградского государственного технического университета, доктор технических наук, zabzot@vstu.ru

Камаев Валерий Анатольевич – заведующий кафедрой САПР и ПК, профессор Волгоградского государственного технического университета, доктор технических наук, cad@vstu.ru

2.4. Computer Models of Common Sense Reasoning

DIAGARA: AN INCREMENTAL ALGORITHM FOR INFERRING IMPLICATIVE RULES FROM EXAMPLES (PART 1)

Xenia Naidenova

Abstract: *An approach is proposed for inferring implicative logical rules from examples. The concept of a good diagnostic test for a given set of positive examples lies in the basis of this approach. The process of inferring good diagnostic tests is considered as a process of inductive common sense reasoning. The incremental approach to learning algorithms is implemented in an algorithm DIAGaRa for inferring implicative rules from examples.*

Keywords: *Incremental and non-incremental learning, learning from examples, machine learning, common sense reasoning, inductive inference, good diagnostic test, lattice theory.*

Introduction

Our approach to machine learning problems is based on the concept of a good diagnostic (classification) test. This concept has been advanced firstly in the framework of inferring functional and implicative dependencies from relations [Naidenova and Polegaeva, 1986]. But later the fact has been revealed that the task of inferring all good diagnostic tests for a given set of positive and negative examples can be formulated as the search of the best approximation of a given classification on a given set of examples and that it is this task that all well known machine learning problems can be reduced to [Naidenova, 1996].

We have chosen the lattice theory as a model for inferring good diagnostic tests from examples from the very beginning of our work in this direction. We believe that it is the lattice theory that must be the mathematical theory of common sense reasoning. One can come to this conclusion by analyzing both the fundamental work in the psychological theory of intelligence [Piaget, 1959], and the experience of modelling thinking processes in the framework of artificial intelligence. The process of objects' classification has been considered in [Shreider, 1974] as an algebraic idempotent semi group with the unit element. An algebraic model of classification and pattern recognition based on the lattice theory has been advanced in [Boldyrev, 1974]. A lot of experience has been obtained on the application of algebraic lattices in machine learning: the works of Finn and his disciples [Finn, 1984], [Kuznetsov, 1993], the model of conceptual knowledge of Wille [1992], the works of the French group [Ganascia, 1989]. The following works are devoted to the application of algebraic lattices for extracting classifications, functional dependencies and implications from data: [Demetrovics and Vu, 1993], [Mannila and Räihä, 1992], [Mannila and Räihä, 1994], [Huntala, et al., 1999], [Cosmadakis, et al., 1986], [Naidenova and Polegaeva, 1986], [Megretskaya, 1989], [Naidenova, et al., 1995a], [Naidenova, et al., 1995b], and [Naidenova, 1992].

An advantage of the algebraic lattices approach is based on the fact that an algebraic lattice can be defined both as an algebraic structure that is declarative and as a system of dual operations with the use of which the elements of this lattice can be generated. This approach allows us to investigate the processes of inferring good classification tests as inductive reasoning processes. In the following part of this chapter, we shall describe our decomposition of the inductive inferring process into subtasks and operations that conform to the operations and subtasks of the natural human reasoning process.

This paper is organized as follows. The concept of a good diagnostic test is introduced and the problem of inferring all good diagnostic tests for a given classification on a given set of examples is formulated. The next section contains the description of a mathematical model underlying algorithms of learning reasoning. We propose a decomposition of learning algorithms into operations and subtasks that are in accordance with human reasoning operations. In the second part of this paper, the concepts of an essential value and an essential example are also introduced and an incremental learning algorithm DIAGaRa is described. The paper ends with a brief summary section.

The Concept of a Good Classification Test

Our approach for inferring implicative rules from examples is based on the concept of a good classification test. A good classification test can be understood as an approximation of a given classification on a given set of examples [Naidenova, 1996]. On the other hand, the process of inferring good tests realizes one of the known canons of induction formulated by J. S. Mille, namely, the joint method of similarity-distinction [Mille, 1900].

A good diagnostic test for a given set of examples is defined as follows. Let R be a table of examples and S be the set of indices of examples belonging to R. Let $R(k)$ and $S(k)$ be the set of examples and the set of indices of examples from a given class k, respectively.

Denote by $FM = R/R(k)$ the examples of the classes different from class k. Let U be the set of attributes and T be the set of attributes values (values, for short) each of which appears at least in one of the examples of R. Let n be the number of examples of R. We denote the domain of values for an attribute Atr by $dom(Atr)$, where $Atr \in U$.

By $s(a)$, $a \in T$, we denote the subset $\{i \in S$: a appears in t_i, $t_i \in R\}$, where $S = \{1, 2, .., n\}$.

Following [Cosmadakis, et al., 1986], we call $s(a)$ the interpretation of $a \in T$ in R. It is possible to say that $s(a)$ is the set of indices of all the examples in R which are covered by the value a.

Since for all a, $b \in dom(Atr)$, $a \neq b$ implies that the intersection $s(a) \cap s(b)$ is empty, the interpretation of any attribute in R is a partition of S into a family of mutually disjoint blocks. By $P(Atr)$, we denote the partition of S induced by the values of an attribute Atr. The definition of $s(a)$ can be extended to the definition of $s(t)$ for any collection t of values as follows: for t, $t \subseteq T$, if $t = a_1\, a_2\, ...\, a_m$, then $s(t) = s(a_1) \cap s(a_2) \cap ... \cap s(a_m)$.

Definition 1. A collection $t \subseteq T$ ($s(t) \neq \varnothing$) of values, is a diagnostic test for the set $R(k)$ of examples if and only if the following condition is satisfied: $t \not\subset t^*$, $\forall\ t^*$, $t^* \in FM$ (the equivalent condition is $s(t) \subseteq S(k)$).

To say that a collection t of values is a diagnostic test for the set $R(k)$ is equivalent to say that it does not cover any example belonging to the classes different from k. At the same time, the condition $s(t) \subseteq S(k)$ implies that the following implicative dependency is true: 'if t, then k.

It is clear that the set of all diagnostic tests for a given set $R(k)$ of examples (call it '$DT(k)$') is the set of all the collections t of values for which the condition $s(t) \subseteq S(k)$ is true. For any pair of diagnostic tests t_i, t_j from $DT(k)$, only one of the following relations is true: $s(t_i) \subseteq s(t_j)$, $s(t_i) \supseteq s(t_j)$, $s(t_i) \approx s(t_j)$, where the last relation means that $s(t_i)$ and $s(t_j)$ are incomparable, i.e. $s(t_i) \not\subset s(t_j)$ and $s(t_j) \not\subset s(t_i)$. This consideration leads to the concept of a good diagnostic test.

Definition 2. A collection $t \subseteq T$ ($s(t) \neq \varnothing$) of values is a good test for the set $R(k)$ of examples if and only if the following condition is satisfied: $s(t) \subseteq S(k)$ and simultaneously the condition $s(t) \subset s(t^*) \subseteq S(k)$ is not satisfied for any t^*, $t^* \subseteq T$, such that $t^* \neq t$.

Good diagnostic tests possess the greatest generalization power and give a possibility to obtain the smallest number of implicative rules for describing examples of a given class k.

The Characterization of Classification Tests

Any collection of values can be irredundant, redundant or maximally redundant.

Definition 3. A collection t of values is irredundant if for any value $v \in t$ the following condition is satisfied: $s(t) \subset s(t/v)$.

If a collection t of values is a good test for $R(k)$ and, simultaneously, it is an irredundant collection of values, then any proper subset of t is not a test for $R(k)$.

Definition 4. Let $X \rightarrow v$ be an implicative dependency which is satisfied in R between a collection $X \subseteq T$ of values and the value v, $v \in T$. Suppose that a collection $t \subseteq T$ of values contains X. Then the collection t is said to be redundant if it contains also the value v.

If t contains the left and the right sides of some implicative dependency $X \rightarrow v$, then the following condition is satisfied: $s(t) = s(t/v)$. In other words, a redundant collection t and the collection t/v of values cover the same set of examples.

If a good test for $R(k)$ is a redundant collection of values, then some values can be deleted from it and thus obtain an equivalent good test with a smaller number of values.

Definition 5. A collection $t \subseteq T$ of values is maximally redundant if for any implicative dependency $X \rightarrow v$, which is satisfied in R, the fact that t contains X implies that t also contains v.

If t is a maximally redundant collection of values, then for any value $v \notin t$, $v \in T$ the following condition is satisfied: $s(t) \supset s(t \cup v)$. In other words, a maximally redundant collection t of values covers the number of examples greater than the collection $(t \cup v)$ of values.

Any example t in R is a maximally redundant collection of values because for any value $v \notin t$, $v \in T$ $s(t \cup v)$ is equal to $\varnothing$.

If a diagnostic test for a given set $R(k)$ of examples is a good one and it is a maximally redundant collection of values, then by adding to it any value not belonging to it we get a collection of values which is not a good test for $R(k)$.

For example, in Table 1 the collection '*Blond Blue*' is a good irredundant test for class 1 and simultaneously it is maximally redundant collection of values. The collection '*Blond Embrown*' is a test for class 2 but it is not good test and simultaneously it is maximally redundant collection of values.

The collection '*Embrown*' is a good irredundant test for class 2. The collection '*Red*' is a good irredundant test and the collection '*Tall Red Blue*' is a maximally redundant and good test for class 1.

It is clear that the best tests for pattern recognition problems must be good irredundant tests. These tests allow construction of the shortest rules of the first type with the highest degree of generalization.

Table - 1. Example 1 of Data Classification. (This example is adopted from [Ganascia, 1989]).

Index of Example	Height	Color of Hair	Color of Eyes	Class
1	Short	Blond	Blue	1
2	Short	Brown	Blue	2
3	Tall	Brown	Embrown	2
4	Tall	Blond	Embrown	2
5	Tall	Brown	Blue	2
6	Short	Blond	Embrown	2
7	Tall	Red	Blue	1
8	Tall	Blond	Blue	1

An Approach for Constructing Good Irredundant Tests

Let R, T, s(t), $t \subseteq T$ be as defined earlier. We give the following propositions the proof of which can be found in [Naidenova, 1999].

PROPOSITION 1.

The intersection of maximally redundant collections of values is a maximally redundant collection.

PROPOSITION 2.

Every collection of values is contained in one and only one maximally redundant collection with the same interpretation.

PROPOSITION 3.

A good maximal redundant test for $R(k)$ either belongs to the set $R(k)$ or it is equal to the intersection of q examples from $R(k)$ for some q, $2 \leq q \leq nt$, where nt is the number of examples in $R(k)$.

One of the possible ways for searching for good irredundant tests for a given class of examples is the following: first, find all good maximally redundant tests; second, for each good maximally redundant test, find all good irredundant tests contained in it. This is a convenient strategy as each good irredundant test belongs to one and only one good maximally redundant test with the same interpretation.

It should be more convenient in the following considerations to denote the set $R(k)$ as $R(+)$ (the set of positive examples) and the set $R/R(k)$ as $R(-)$ (the set of negative examples). We will also denote the set $S(k)$ as $s(+)$.

The following Algorithm 1 solves the task of inferring all good maximally redundant tests for a given set of positive examples. The idea of this algorithm has been advanced in [Naidenova and Polegaeva, 1991].

By $s_q = (i_1, i_2, \ldots, i_q)$, we denote a subset of S, containing q indices from S. Let S(test-q) be the set of elements $s = \{i_1, i_2, ..., i_q\}$, q = 1,2, ..., nt, satisfying the condition that $t(s)$ is a test for $R(+)$. Here nt denotes the number of positive examples.

We will use an inductive rule for constructing $\{i_1, i_2, ..., i_{q+1}\}$ from $\{i_1, i_2, ..., i_q\}$, q = 1, 2, ..., nt-1. This rule relies on the following consideration: if the set $\{i_1, i_2, ..., i_{q+1}\}$ corresponds to a test for $R(+)$, then all its proper subsets must correspond to tests too and, consequently, they must be in S(test-q). Thus the set $\{i_1, i_2, ..., i_{q+1}\}$ can be constructed if and only if S(test-q) contains all its proper subsets. Having constructed the set $s_{q+1} = \{i_1, i_2, ..., i_{q+1}\}$, we have to determine whether it corresponds to the test or not. If $t(s_{q+1})$ is not a test, then s_{q+1} is deleted, otherwise s_{q+1} is inserted in S(test-(q+1)). The algorithm is over when it is impossible to construct any element for S(test-(q+1)).

We use in Algorithm 1 the function to_be_test(t): if $s(t) \cap s(+) = s(t)$ ($s(t) \subseteq s(+)$) then *true* else *false*.

Algorithm 1. Inferring all Good Maximally Redundant Tests (GMRTs) for a set $R(+)$ of positive examples.

1. Input: q = 1, $R(+)$, $s(+)$= {1,2,…, nt}, S(test-q) = {{1}, {2}, ..., {nt}}.
Output: the set *TGOOD* of all GMRTs for $R(+)$.
2. S_q ::= S(test-q);
3. While $\left|\left| S_q \right|\right| \geq q + 1$ do
3.1 Generating $S(q + 1) = \{s = \{i_1, ..., i_{(q+1)}\}: (\forall j)\ (1 \leq j \leq q + 1)\ (i_1, ..., i_{(j-1)}, i_{(j+1)}, ..., i_{(q+1)}\} \in S_q\}$;
3.2 Generating S(test-(q + 1)) = $\{s = \{i_1, ..., i_{(q+1)}\}: (s \in S(q + 1))$ & (to_be_test($t(s)$) = *true*)};
3.3 S(test-q) ::= $\{s = \{i_1, ..., i_q\}: (s \in S(\text{test-}q))$ & $((\forall s')(s' \in S(\text{test-}(q + 1))\ s \not\subset s')\}$;
3.4. q ::= q + 1;
3.5. *max* ::= q;
end while
4. *TGOOD*::= $\varnothing$;
5. While $q \leq max$ do *TGOOD*::= *TGOOD* $\cup$ $\{t(s): s = \{i_1, ..., i_s\} \in S(\text{test-}q)\}$;
5.1 q::= q + 1;
end while
end

The following Table 2 gives an illustration of inferring GMRTs for the examples of class 2 (see, please, Table 1).

The set S_q, q = 2 consists of 10 elements {{2,3}, {2,4}, {2,5}, {2,6}, {3,4}, {3,5}, {3,6}, {4,5}, {4,6}, {5,6}}. But t({2,4}), t({2,6}), t({4,5}), and t({5,6}) are not tests for class2, hence we can construct only two elements of the next level for q = 3: S_3 = S(test-3) = {{2,3,5}, {3,4,6}}.

As a result, the tests obtained correspond to the following implicative rules: “if COLOR of HAIR = *Brown*, then Class = 2” and “if COLOR of EYES = *Embrown*, then Class = 2”.

Algorithm 1 is also used for inferring all good irredundant tests (GIRTs) contained in a good maximally redundant test.

Now let $t = \{a_1, a_2, \ldots, a_m\} \subseteq T$ be a collection of values that is a GMRT for $R(+)$.

We will use a rule of inductive transition from an element $t_q = (A_1, A_2, \ldots, A_q)$ to another element $t_{q+1} = (A_1, A_2, \ldots, A_{q+1})$, t_q, $t_{q+1} \subseteq t$. But now we are interested in obtaining irredundant collections of values. If $t_{q+1} = (A_1, A_2, \ldots, A_{q+1})$ is irredundant, then all its proper subsets must be irredundant too.

Table - 2. Example of inferring logical rules for Class 2 (Table 1) with the use of Algorithm 1.

S(test-1)	$t(s)$, $s \in S$(test-1)	S(test-2)	$t(s)$, $s \in S$(test-2)	S(test-3)	$t(s)$, $s \in S$(test-3)
{2}	‘*Short Brown Blue*’	{2,3}	‘*Brown*’	{2,3,5}	‘*Brown*’
{3}	‘*Tall Brown Embrown*’	{2,5}	‘*Brown Blue*’		
{4}	‘*Tall Blond Embrown*’	{3,4}	‘*Tall Embrown*’	{3,4,6}	‘*Embrown*’
{5}	‘*Tall Brown Blue*’	{3,5}	‘*Tall Brown*’		
{6}	‘*Short Blond Embrown*’	{3,6}	‘*Embrown*’		
		{4,6}	‘*Blond Embrown*’		

Having constructed the set $t_{q+1} = (A_1, A_2, \ldots, A_{q+1})$, we have to determine whether it is an irredundant collection of values or not. If t_{q+1} is redundant, then it is deleted, if t_{q+1} is a test, then t_{q+1} is inserted in the set *TGOOD* of all good irredundant tests contained in *t*. If t_{q+1} is irredundant but not a test, then it is a candidate for extension.

The following Algorithm 2 solves the task of inferring all GIRTs contained in a maximally redundant test for a given set of positive examples.

We use in Algorithm 2 the function to_be_irredundant(*t*)::= if for $(\forall\ a_i)$ $(a_i \in t)$ $s(t) \neq s(t/\ a_i)$ then *true* else *false*.

Algorithm 2. Inferring all GIRTs contained in a given GMRT for *R*(+)**.**

Input: $q = 1$, *R*, *R*(+), *S*, $t = \{a_1, a_2, \ldots, a_m\}$ – a collection of values – a GMRT, *F*(irredundant – *q*) = $\{\{a_1\}, \{a_2\}, \ldots, \{a_m\}\}$ – the family of irredundant subsets of values with *q* equal to 1.

Output: the set *TGOOD* of all the GIRTs for *R*(+) contained in *t*.

1. F_q::= *F*(irredundant – *q*);
1.1 Generating *F*(test-*q*) ={$t = \{a_{i1}, \ldots, a_{iq}\}$: $(t \in F_q)$ & (to_be_test(*t*) = *true*)};
1.2 F_q ::= $F_q \setminus$ *F*(test-*q*) ;
2. While $\left|\left|F_q\right|\right| \geq q + 1$ do
2.1. Generating $F(q + 1)$ =
= $\{t = \{a_{i1}, \ldots, a_{i(q+1)}\}$: $(\forall\ j)$ $(1 \leq j \leq q + 1)$ $(a_{i1}, \ldots, a_{i(j-1)}, a_{i(j+1)}, \ldots, a_{i(q+1)}\} \in F_q\}$;
2.2. Generating *F*(irredundant – (*q* +1)) :
F(irredundant – (*q*+1)) ::= $\{t \in F(q + 1)$: to_be_irredundant(*t*) = *true* };
2.3. q ::= $q + 1$;
2.4. max ::= *q*;
end while
3. *TGOOD* ::= $\varnothing$;
4.While $q \leq$ max do
4.1. *TGOOD* ::= *TGOOD* $\cup$ {*t*: $t \in$ *F*(test-*q*) };
4.2. *q*::= $q + 1$;
end while
end

The Duality of Good Diagnostic Tests

In Algorithms 1 and 2, we used (without explicit definition) correspondences of Galois *G* on $S \times T$ and two relations $S \to T$, $T \to S$ [Ore, 1944], [Riguet, 1948]. Let $s \subseteq S$, $t \subseteq T$. We define the relations as follows:

$S \to T$: $t(s)$ = {intersection of all t_i: $t_i \subseteq T$, $i \in s$} and $T \to S$: $s(t) = \{i: i \in S, t \subseteq t_i\}$.

Extending *s* by an index *j** of some new example leads to receiving a more general feature of examples:

$(s \cup j^*) \supseteq s$ implies $t(s \cup j^*) \subseteq t(s)$.

Extending *t* by a new value *A* leads to decreasing the number of examples possessing the general feature '*tA*' in comparison with the number of examples possessing the general feature '*t*':

$(t \cup A) \supseteq t$ implies $s(t \cup A) \subseteq s(t)$.

We introduce the following generalization operations (functions):

generalization_of(t) = $t' = t(s(t))$; generalization_of(s) = $s' = s(t(s))$.

As a result of the generalization of *s*, the sequence of operations $s \to t(s) \to s(t(s))$ gives that $s(t(s)) \supseteq s$. This generalization operation gives all the examples possessing the feature $t(s)$.

As a result of the generalization of *t*, the sequence of operations $t \to s(t) \to t(s(t))$ gives that $t(s(t)) \supseteq t$. This generalization operation gives the maximal general feature for examples the indices of which are in $s(t)$.

These generalization operations are not artificially constructed operations. One can perform mentally a lot of such operations during a short period of time. We give some examples of these operations. Suppose that somebody has seen two films (s) with the participation of Gerard Depardieu ($t(s)$). After that, he tries to know all the films

with his participation ($s(t(s))$). One can know that Gerard Depardieu acts with Pierre Richard (t) in several films ($s(t)$). After that, he can discover that these films are the films of the same producer Francis Veber $t(s(t))$.

Namely, these generalization operations will be used in the algorithm DIAGaRa.

The Definition of Good Diagnostic Tests as Dual Objects

We implicitly used two generalization operations in all the considerations of diagnostic tests. Now we define a diagnostic test as a dual object, i.e. as a pair (SL, TA), $SL \subseteq S$, $TA \subseteq T$, $SL = s(TA)$ and $TA = t(SL)$.

The task of inferring tests is a dual task. It must be formulated both on the set of all subsets of S, and on the set of all subsets of T.

Definition 6. Let $PM = \{s_1, s_2, \ldots, s_m\}$ be a family of subsets of some set M. Then PM is a Sperner system [Sperner, 1928] if the following condition is satisfied: $s_i \not\subset s_j$ and $s_j \not\subset s_i$, $\forall(i,j)$, $i \neq j$, $i, j = 1, \ldots, m$.

Definition 7. To find all *Good Maximally Redundant Tests* (GMRTs) for a given class $R(k)$ of examples means to construct a family PS of subsets $s_1, s_2, \ldots, s_{np}$ of the set S such that:

1) $s_j \subseteq S(k)$, $\forall j = 1,..., np$;

2) PS is a Sperner system;

3) each s_j is a maximal set in the sense that adding to it the index i of the example t_i such that $i \notin s_j$, $i \in S$ implies $s(t(s_j \cup i)) \not\subset S(k)$. Putting it in another way, $t(s_j \cup i)$ is not a test for the class k, so there exists such example t^*, $t^* \in R(-)$ that $t(s_j \cup i) \subseteq t^*$.

The set of all GMRTs is determined as follows:

$\{t: t(s_j), s_j \in PS, \forall j, j = 1,..., np\}$.

Definition 8. To find all *Good Irredundant Tests* (GIRTs) for a given class $R(k)$ of examples means to find a family PRT of subsets $t_1, t_2,..., t_{nq}$ of the set T such that:

1) $t_j \not\subset t$ $\forall j, j = 1,..., nq$, $\forall t, t \in R(+)$ and, simultaneously, $\forall t_j, j = 1,..., nq$, $s(t_j) \neq \varnothing$ there does not exist such a collection $s^* \neq s(t_j)$, $s^* \subseteq S$ of indices for which the following condition is satisfied $s(t_j) \subset s^* \subseteq S(k)$;

2) PRT is a Sperner system;

3) each t_j – a minimal set in the sense that removing from it any value A belonging to it implies $s(t_j$ without $A) \not\subset S(k)$.

Decomposition of Good Classification Tests Inferring into Subtasks

The Algorithms 1 and 2 find all the GMRTs and GIRTs for a given set of positive examples but the number of tests can be exponentially large. In this case, these algorithms will be not realistic. Now we consider some decompositions of the problem that provide the possibility to restrict the domain of searching, to predict, in some degree, the number of tests, and to choose tests with the use of essential values and/or examples. This decomposition gives an approach to constructing incremental algorithms of inferring all good classification tests for a given set of examples.

We consider two kinds of subtasks (please, see also [Naidenova, 2001]:

for a given set of positive examples

1) given a positive example t, find all GMRTs contained in t;

2) given a non-empty collection of values X (maybe only one value) such that it is not a test, find all GMRTs containing X.

Each example contains only some subset of values from T, hence each subtask of the first kind is simpler than the initial one. Each subset X of T appears only in a part of all examples, hence each subtask of the second kind is simpler than the initial one.

Forming the Subtasks

The subtask of the first kind. We introduce the concept of an example's projection proj(R)[t] of a given positive example t on a given set R(+) of positive examples. The proj(R)[t] is the set Z = {z: (z is non-empty intersection of t and t') & ($t' \in R$(+)) & (z is a test for a given class of positive examples)}.

If the proj(R)[t] is not empty and contains more than one element, then it is a subtask for inferring all GMRTs that are in t. If the projection contains one and only one element equal to t, then t is a GMRT.

To make the operation of forming a projection perfectly clear we construct the projection of t_2 = '*Short Brown Blue*' on the examples of the second class (Table 1). This projection includes t_2 and the intersections of t_2 with the other positive examples of the second class, i.e. with the examples t_3, t_4, t_5, t_6 (Table 3).

Table - 3. The Intersections of Example t_2 with the Examples of Class 2.

Index of Example	*Height*	*Color of Hair*	*Color of Eyes*	*Test?*
2	*Short*	*Brown*	*Blue*	*Yes*
3		*Brown*		*Yes*
4				*No*
5		*Brown*	*Blue*	*Yes*
6	*Short*			*No*

In order to check whether an element of the projection is a test or not we use the function to_be_test(t) in the following form: to_be_test(t) = if $s(t) \subseteq s(+)$ then *true* else *false*, where s(+) is the set of indices of positive examples, $s(t)$ is the set of indices of all positive and negative examples containing t. If s(-) is the set of indices of negative examples, then $S = s(+) \cup s(-)$ and $s(t) = \{i: t \subseteq t_i, i \in S\}$.

The intersection $t_2 \cap t_4$ is the empty set. Hence, the row of the projection with the number 4 is empty. The intersection $t_2 \cap t_6$ is not a test for Class 2 because $s(Short) = \{1,2,6\} \not\subset s(+)$, where s(+) is equal to {2,3,4,5,6}.

Finally, we have the projection of t_2 on the examples of the second class in Table 4.

The subtask turns out to be very simple because the intersection of all the rows of the projection is a test for the second class: $t(\{2,3,5\})$ = '*Brown*', s(*Brown*) = {2,3,5} and {2,3,5} $\subseteq s(+)$.

The subtask of the second kind. We introduce the concept of an attributive projection proj(R)[A] of a given value A on a given set R(+) of positive examples.

The projection proj(R)[A] = {t: ($t \in R$(+)) & (A appears in t)}. Another way to define this projection is: proj(R)[A] = {t_i: $i \in (s(A) \cap s(+))$}. If the attributive projection is not empty and contains more than one element, then it is a subtask of inferring all GMRTs containing a given value A. If A appears in one and only one example, then A does not belong to any GMRT different from this example.

Forming the projection of A makes sense if A is not a test and the intersection of all positive examples in which A appears is not a test too, i.e. $s(A) \not\subset s(+)$ and $t' = t(s(A) \cap s(+))$ is not a test for a given set of positive examples.

Denote the set {s(A) ∩ s(+)} by splus(A). In Table 1, we have:

s(+) = {2,3,4,5,6}, splus(Short) → {2,6}, splus(Brown) → {2,3,5}, splus(Blue) → {2,5}, splus(Tall) → {3,4,5}, splus(Embrown) → {3,4,6}, and splus(Blond) →{4,6}.

Table - 4. The Projection of the Example t_2 on the Examples of Class 2.

Index of Example	Height	Color of Hair	Color of Eyes	Test?
2	Short	Brown	*Blue*	Yes
3		Brown		Yes
5		Brown	Blue	Yes

For the value 'Brown' we have: s(Brown) = {2,3,5} and s(Brown) = splus(Brown), i.e. s(Brown) ⊆ s(+).

*Analogously for the value '*Embrown*' we have:* s(Embrown*) = {3,4,6} and* s(Embrown*) =* splus*(*Embrown*), i.e.* s*(*Embrown*) ⊆* s*(+).*

Table - 5. The Result of Reducing the Projection after Deleting the Values '*Brown'* and '*Embrown'*

Index of Example	Height	Color of Hair	Color of Eyes	Test?
2	Short		Blue	No
3	Tall			No
4	Tall	Blond		No
5	Tall		Blue	No
6	Short	Blond		No

These values are irredundant and simultaneously maximally redundant tests because t({2,3,5}) = 'Brown' and t({3,4,6}) = 'Embrown'. It is clear that these values cannot belong to any test different from them. We delete 'Brown' and 'Embrown' from further consideration with the following result as shown in Table 5.

Now none of the remaining rows of the second class is a test because s(Short, Blue) = {1,2}, s(Tall) = {3,4,5,7,8}, s(Tall, Blond) = {4,8}, s(Tall, Blue) ={5,7,8}, s(Short, Blond) = {1,6} $\not\subset$ s(+). The values 'Brown' and 'Embrown' exhaust the set of the GMRTs for this class of positive examples.

Bibliography

[Boldyrev, 1974] N. G. Boldyrev, "Minimization of Boolean Partial Functions with a Large Number of "Don't Care" Conditions and the Problem of Feature Extraction", Proceedings of International Symposium "Discrete Systems", Riga, Latvia, pp.101-109, 1974.

[Cosmadakis et al., 1986] S. Cosmadakis, P. C. Kanellakis, N. Spyratos, "Partition Semantics for Relations", Journal of Computer and System Sciences, Vol. 33, No. 2, pp.203-233, 1986.

[Demetrovics and Vu, 1993] J. Demetrovics and D. T. Vu, "Generating Armstrong Relation Schemes and Inferring Functional Dependencies from Relations", International Journal on Information Theory & Applications, Vol. 1, No. 4, pp.3-12, 1993.

[Finn, 1984] V. K. Finn, "Inductive Models of Knowledge Representation in Man-Machine and Robotics Systems", Proceedings of VINITI, Vol. A, pp.58-76, 1984.

[Ganascia, 1989] J.- Gabriel. Ganascia, "EKAW - 89 Tutorial Notes: Machine Learning", Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, Paris, France, pp. 287-296, 1989.

[Huntala et al., 1999] Y. Huntala, J. Karkkainen, P. Porkka, and H. Toivonen, "TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies", The Computer Journal, Vol. 42, No. 2, pp. 100-111, 1999.

[Kuznetsov, 1993] S. O. Kuznetsov, "Fast Algorithm of Constructing All the Intersections of Finite Semi-Lattice Objects", Proceedings of VINITI, Series 2, No. 1, pp. 17-20, 1993.

[Mannila and Räihä, 1992] H. Mannila, and K. – J. Räihä, "On the Complexity of Inferring Functional Dependencies", Discrete Applied Mathematics, Vol. 40, pp. 237-243, 1992.

[Mannila and Räihä, 1994] H. Mannila, and K. – J. Räihä, "Algorithm for Inferring Functional Dependencies". Data & Knowledge Engineering, Vol. 12, pp. 83-99, 1994.

[Megretskaya, 1989] I. A. Megretskaya, "Construction of Natural Classification Tests for Knowledge Base Generation", in: The Problem of the Expert System Application in the National Economy, Kishinev, Moldavia, pp. 89-93, 1988.

[Mille, 1900] J. S. Mille, The System of Logic, Russian Publishing Company "Book Affair": Moscow, Russia, 1900.

[Naidenova and Polegaeva, 1986] X. A. Naidenova, J. G. Polegaeva, "An Algorithm of Finding the Best Diagnostic Tests", The 4-th All Union Conference "Application of Mathematical Logic Methods", Theses of Papers, Mintz, G; E, Lorents, P. P. (Eds), Institute of Cybernetics, National Acad. of Sciences of Estonia, Tallinn, Estonia, pp. 63-67, 1986.

[Naidenova and Polegaeva, 1991] X. A. Naidenova, J. G. Polegaeva, "The System of Knowledge Acquisition from Experimental Facts", in: "Industrial Applications of Artificial Intelligence", James L. Alty and Leonid I. Mikulich (Eds), Elsevier Science Publishers B.V., Amsterdam, The Netherlands, pp. 87-92, 1991.

[Naidenova, 1992] X. A. Naidenova, "Machine Learning As a Diagnostic Task", in: "Knowledge-Dialogue-Solution", Materials of the Short-Term Scientific Seminar, Saint-Petersburg, Russia, editor Arefiev, I., pp.26-36, 1992.

[Naidenova et al., 1995a] X. A. Naidenova, J. G. Polegaeva, J. E. Iserlis, "The System of Knowledge Acquisition Based on Constructing the Best Diagnostic Classification Tests", Proceedings of International Conference "Knowledge-Dialog-Solution", Jalta, Ukraine, Vol. 1, pp. 85-95, 1995a.

[Naidenova et al., 1995b] X. A. Naidenova, M. V. Plaksin, V. L. Shagalov, "Inductive Inferring All Good Classification Tests", Proceedings of International Conference "Knowledge-Dialog-Solution", Jalta, Ukraine, Vol. 1, pp.79-84, 1995b.

[Naidenova, 1996] X. A. Naidenova, "Reducing Machine Learning Tasks to the Approximation of a Given Classification on a Given Set of Examples", Proceedings of the 5-th National Conference at Artificial Intelligence, Kazan, Tatarstan, Vol. 1, pp. 275-279, 1996.

[Naidenova, 1999] X. A. Naidenova, "The Data-Knowledge Transformation", in: "Text Procesing and Cognitive Technologies", Paper Collection, editor Solovyev, V. D., - Pushchino, Russia, Vol. 3, pp. 130-151, 1999.

[Naidenova and Ermakov, 2001] X. A. Naidenova, A. E. Ermakov, "The Decomposition of Algorithms of Inferring Good Diagnostic Tests", Proceedings of the 4-th International Conference "Computer – Aided Design of Discrete Devices" (CAD DD'2001), Institute of Engineering Cybernetics, National Academy of Sciences of Belarus, editor A. Zakrevskij, Minsk, Belarus, Vol. 3, pp. 61-69, 2001.

[Naidenova, 2001] X. A. Naidenova, "Inferring Good Diagnostic Tests as a Model of Common Sense Reasoning", Proceedings of the International Conference "Knowledge-Dialog-Solution" (KDS'2001), State North-West Technical University, Publishing House « Lan », Saint-Petersburg, Russia, Vol. II, pp. 501-506, 2001.

[Ore, 1944] O. Ore, "Galois Connexions", Trans. Amer. Math. Society, Vol. 55, No. 1, pp. 493-513, 1944.

[Piaget, 1959] J.Piaget, La genèse des Structures Logiques Elémentaires, Neuchâtel, 1959.

[Riguet, 1948] J. Riguet, "Relations Binaires, Fermetures, Correspondences de Galois", Bull. Soc. Math., France, Vol. 76., No 3, pp.114-155, 1948.

[Shreider, 1974] J. Shreider, "Algebra of Classification", Proceedings of VINITI, Series 2, No. 9, pp. 3-6, 1974.

[Sperner, 1928] E. Sperner, "Eine satz uber Untermengen einer Endlichen Menge". Mat. Z., Vol. 27, No. 11, pp. 544-548, 1928.

[Wille, 1992] R. Wille, "Concept Lattices and Conceptual Knowledge System", Computer Math. Appl., Vol. 23, No. 6-9, pp. 493-515, 1992.

Author's Information

Naidenova Xenia Alexandrovna - Military medical academy, Saint-Petersburg, Stoikosty street, 26-1-248, naidenova@mail.spbnit.ru.

DIAGARA: AN INCREMENTAL ALGORITHM FOR INFERRING IMPLICATIVE RULES FROM EXAMPLES (PART 2)

Xenia Naidenova

Abstract: *An approach is proposed for inferring implicative logical rules from examples. The concept of a good diagnostic test for a given set of positive examples lies in the basis of this approach. The process of inferring good diagnostic tests is considered as a process of inductive common sense reasoning. The incremental approach to learning algorithms is implemented in an algorithm DIAGaRa for inferring implicative rules from examples.*

Keywords*: Incremental and non-incremental learning, learning from examples, machine learning, common sense reasoning, inductive inference, good diagnostic test, lattice theory.*

Introduction

In the part 1 of this paper, we considered the decompositions of inferring all good classification tests for a given set of examples into the subtasks of the first kind and of the second kind. We also considered the operations of forming the subtasks of both kinds. Now we continue by introducing the rules of reducing the subtasks.

Reducing the Subtasks

The following theorem gives the foundation for reducing projections both of the first and the second kind. The proof of this theorem can be found in [Naidenova et al., 1995b].

THEOREM 1.

Let A be a value from T, X be a maximally redundant test for a given set R(+) of positive examples and s(A) ⊆ s(X). Then A does not belong to any maximally redundant good test for R(+) different from X.

To illustrate the way of reducing projections, we consider another partition of the rows of Table 1 (see, please Part 1 of this paper) into the sets of positive and negative examples as shown in Table 6.

Table - 6. The Example 2 of a Data Classification.

Index of Example	Height	Color of Hair	Color of Eyes	Class
1	Short	Blond	Blue	1
2	Short	Brown	Blue	1
3	Tall	Brown	Embrown	1
4	Tall	Blond	Embrown	2
5	Tall	Brown	Blue	2
6	Short	Blond	Embrown	2
7	Tall	Red	Blue	2
8	Tall	Blond	Blue	2

Let s(+) be equal to {4,5,6,7,8}. The value 'Red' is a test for positive examples because s(Red) = splus(Red) ={7}. Delete 'Red' from the projection. The value 'Tall' is not a test because s(Tall) = {3,4,5,7,8} and it is not equal to splus(Tall) = {4,5,7,8}. Also t(splus(Tall)) = 'Tall' is not a test. The attributive projection of the value 'Tall' on the set of positive examples is in Table 7.

In this projection, splus(Blue) = {5,7,8}, t(splus(Blue)) = 'Tall Blue', s(Tall Blue) = {5,7,8} = splus(Tall Blue) hence 'Tall Blue' is a test for the second class. We have also that splus(Brown) = {5}, but {5} ⊆ {5,7,8} and, consequently, there does not exist any good test which contains simultaneously the values 'Tall' and 'Brown'. Delete 'Blue' and 'Brown' from the projection as shown in Table 8.

However, now the rows t_5 and t_7 are not tests for the second class and they can be deleted as shown in Table 9. The intersection of the remaining rows of the projection is 'Tall Blond'. We have that s(Tall Blond) = {4,8} ⊆ s(+) and this collection of values is a test for the second class.

Table - 7. The Projection of the Value *'Tall'* on the Set *R(+).*

Index of Example	Height	Color of Hair	Color of Eyes	Test?
4	Tall	Blond	Embrown	Yes
5	Tall	Brown	Blue	Yes
7	Tall		Blue	Yes
8	Tall	Blond	Blue	Yes

Table - 8. The Projection of the Value *'Tall'* on *R(+)* without the Values *'Blue'* and *'Brown'.*

Index of Example	Height	Color of Hair	Color of Eyes	Test?
4	Tall	Blond	Embrown	Yes
5	Tall			No
7	Tall			No
8	Tall	Blond		Yes

Table - 9. The Projection of the Value *'Tall'* on *R(+)* without the Examples t_5 and t_7.

Index of Example	Height	Color of Hair	Color of Eyes	Test?
4	Tall	Blond	Embrown	Yes
8	Tall	Blond		Yes

As we have found all the tests for the second class containing *'Tall'* we can delete *'Tall'* from the examples of the second class as shown in Table 10.

Table - 10. The Result of Deleting the Value *'Tall'* from the Set *R(+).*

Index of Example	Height	Color of Hair	Color of Eyes	Test?	Class
1	Short	Blond	Blue	Yes	1
2	Short	Brown	Blue	Yes	1
3	Tall	Brown	Embrown	Yes	1
4		Blond	Embrown	Yes	2
5		Brown	Blue	No	2
6	Short	Blond	Embrown	Yes	2
7			Blue	No	2
8		Blond	Blue	No	2

Next we can delete the rows t_5, t_7, and t_8. The result is in Table 11.

The intersection of the remaining examples of the second class gives a test 'Blond Embrown' because *s(Blond Embrown) = splus(Blond Embrown) = {4,6}* $\subseteq$ *s(+).*

Table - 11. The Result of Deleting t_5, t_7, and t_8 from the Set *R(+).*

Index of Example	Height	Color of Hair	Color of Eyes	Class
1	Short	Blond	Blue	1
2	Short	Brown	Blue	1
3	Tall	Brown	Embrown	1
4		Blond	Embrown	2
6	Short	Blond	Embrown	2

The choice of values or examples for forming a projection requires special consideration.

In contrast to incremental learning, where the problem is considered of how to choose relevant knowledge to be best modified, here we come across the opposite goal to eliminate irrelevant knowledge not to be processed.

Choosing Values and Examples for the Formation of Subtasks

Next, it is shown that it is convenient to choose essential values in an example and essential examples in a projection for the decomposition of the problem of inferring GMRTs into the subtasks of the first or second kind.

An Approach for Searching for Essential Values

Let t be a test for positive examples. Construct the set of intersections $\{t \cap t': t' \in R(-)\}$. It is clear that these intersections are not tests for positive examples. Take one of the intersections with the maximal number of values in it. The values complementing the maximal intersection in t is the minimal set of essential values in t.

Return to Table 6. Exclude the value 'Red' (we know that 'Red' is a test for the second class) and find the essential values for the examples t_4, t_5, t_6, t_7, and t_8. The result is in Table 12.

Consider the value 'Embrown' in t_6: splus(Embrown) = {4,6}, t({4,6}) = 'Blond Embrown' is a test.

The value 'Embrown' can be deleted. But this value is only one essential value in t_6 and, therefore, t_6 can be deleted too. After that splus(Blond) is modified to the set {4,8}.

We observe that t({4,8}) = 'Tall Blond' is a test. Hence, the value 'Blond' can be deleted from further consideration together with the row t_4. Now the intersection of the rows t_5, t_7, and t_8 produces the test 'Tall Blue'.

Table - 12. The Essential Values for the Examples *t_4, t_5, t6, t_7, and t_8.*

Index of Example	Height	Color of Hair	Color of Eyes	Essential Values	Class
1	Short	Blond	Blue		1
2	Short	Brown	Blue		1
3	Tall	Brown	Embrown		1
4	Tall	Blond	Embrown	Blond	2
5	Tall	Brown	Blue	*Blue, Tall*	2
6	Short	Blond	Embrown	Embrown	2
7	Tall		Blue	Tall, Blue	2
8	Tall	Blond	Blue	Tall	2

An Approach for Searching for Essential Examples

Let *STGOOD* be the partially ordered set of elements *s* satisfying the condition that *t*(*s*) is a GMRT for *R*(+). We can use the set *STGOOD* to find indices of essential examples in some subset *s** of indices for which *t*(*s**) is not a test. Let $s^* = \{i_1, i_2, \dots, i_q\}$. Construct the set of intersections $\{s^* \cap s' : s' \in STGOOD\}$. Any obtained intersection $s^* \cap s'$ corresponds to a test for positive examples. Take one of the intersections with the maximal number of indices. The subset of *s** complementing in *s** the maximal intersection is the minimal set of indices of essential examples in *s**. For instance, $s^* = \{2,3,4,7,8\}$, $s' = \{2,3,4,7\}$, $s' \in STGOOD$, hence 8 is the index of essential example t_8 in *s**.

In the beginning of inferring GMRTs, the set *STGOOD* is empty. Next we describe the procedure with the use of which a quasi-maximal subset of *s** that corresponds to a test is obtained.

We begin with the first index i_1 of *s**, then we take the next index i_2 of *s** and evaluate the function to_be_test ($t(\{i_1, i_2\})$). If the value of the function is *true*, then we take the next index i_3 of *s** and evaluate the function to_be_test ($t(\{i_1, i_2, i_3\})$). If the value of the function is *false*, then the index i_2 of *s** is skipped and the function to_be_test ($t(\{i_1, i_3\})$) is evaluated. We continue this process until we achieve the last index of *s**.

For example, in Table 6, s(+) = {4,5,6,7,8}. Find the quasi-minimal subset of indices of essential examples for s(+). Using the procedure described above we get that t({4,6}) = 'Blond Embrown' is a test for the second class and 5,7,8 are the indices of essential examples in s(+). Consider row t_5. We know that 'Blue' is essential in it (see, please, Table 12). We have t(splus{Blue}) = t({5,7,8}) = 'Tall Blue', and 'Tall Blue' is a test for the second class of examples. Delete'Blue' and t_5. Now t_7 is not a test and we delete it. After that splus(Tall) is modified to be the set {4,8}, and t({4,8}) = 'Tall Blond' is a test. Hence, the value 'Tall' together with row t_8 cannot be considered for searching for new tests. Finally s(+) = {4,6} corresponds to the test already known.

An Approach for Incremental Algorithms

The decomposition of the main problem of inferring GMRTs into subtasks of the first or second kind gives the possibility to construct incremental algorithms for this problem. The simplest way to do it consists of the following steps: choose example (value), form subproblem, solve subproblem (with the use of Algorithm 1 or Algorithm 2), delete example (value) after the subproblem is over, reduce *R*(+) and *T* and check the condition of ending the main task.

A recursive procedure for using attributive subproblems for inferring GMRTs has been described in [Naidenova et al., 1995b]. Some complexity evaluations of this algorithm can be found in [Naidenova and Ermakov, 2001]. In the following part of this chapter, we give an algorithm for inferring GMRTs the core of which is the decomposition of the main problem into the subtasks of the first kind combined with searching essential examples.

DIAGaRa: An Algorithm for Inferring All GMRTs with the Decomposition into Subtasks of the First Kind

The algorithm DIAGaRa for inferring all the GMRTs with the decomposition into subproblems of the first kind is briefly described in Figure 1.

$s^* \leftarrow s(+) = \{1, \dots, nt\}$;
$t^* \leftarrow T$;
Do
Begin
1. to find all the GMRTs for a given set of positive examples with the use of the basic algorithm of solving subtask of the first kind;
End

Figure - 1. The Algorithm DIAGaRa.

The Basic Recursive Algorithm for Solving a Subtask of the First Kind

The initial information for the algorithm of finding all the GMRTs contained in a positive example is the projection of this example on the current set $R(+)$. Essentially the projection is simply a subset of examples defined on a certain restricted subset t^* of values. Let s^* be the subset of indices of examples from $R(+)$ which have produced the projection.

It is useful to introduce the characteristic $W(t)$ of any collection t of values named by the weight of t in the projection: $W(t) = ||s^* \cap s(t)||$ is the number of positive examples of the projection containing t. Let *WMIN* be the minimal permissible value of the weight.

Let *STGOOD* be the partially ordered set of elements s satisfying the condition that $t(s)$ is a good test for $R(+)$.

The basic algorithm consists of applying the sequence of the following steps:

Step 1. Check whether the intersection of all the elements of projection is a test and if so, then s^* is stored in *STGOOD* if s^* corresponds to a good test at the current step; in this case the subtask is over. Otherwise the next step is performed (we use the function to_be_test(t): if $s(t) \cap s(+) = s(t)$ ($s(t) \subseteq s(+)$) then *true* else *false*).

Step 2. For each value A in the projection, the set $splus(A) = \{s^* \cap s(A)\}$ and the weight $W(A) = ||splus(A)||$ are determined and if the weight is less than the minimum permissible weight *WMIN*, then the value A is deleted from the projection. We can also delete the value A if $W(A)$ is equal to *WMIN* and $t(splus(A))$ is not a test – in this case A will not appear in a maximally redundant test t with $W(t)$ equal to or greater than *WMIN*.

Step 3. The generalization operation is performed: $t' = t(splus(A))$, $A \in t^*$; if t' is a test, then the value A is deleted from the projection and $splus(A)$ is stored in *STGOOD* if $splus(A)$ corresponds to a good test at the current step.

Step 4. The value A can be deleted from the projection if $splus(A) \subseteq s'$ for some $s' \in STGOOD$.

Step 5. If at least one value has been deleted from the projection, then the reduction of the projection is necessary. The reduction consists of deleting the elements of projection that are not tests (as a result of previous eliminating values). If, under reduction, at least one element has been deleted from the projection, then Step 2, Step 3, Step 4, and Step 5 are repeated.

Step 6. Check whether the subtask is over or not. The subtask is over when either the projection is empty or the intersection of all elements of the projection corresponds to a test (see Step 1). If the subtask is not over, then the choice of an essential example in this projection is performed and the new subtask is formed with the use of this essential example. The new subsets s^* and t^* are constructed and the basic algorithm runs recursively. The important part of the basic algorithm is how to form the set *STGOOD.*

We give in the Appendix an example of the work of the algorithm DIAGaRa.

An Approach for Forming the Set *STGOOD*

Let $L(S)$ be the set of all subsets of the set S. $L(S)$ is the set lattice [Rasiova, 1974]. The ordering determined in the set lattice coincides with the set-theoretical inclusion. It will be said that subset s_1 is absorbed by subset s_2, i.e. $s_1 \leq s_2$, if and only if the inclusion relation is hold between them, i.e. $s_1 \subseteq s_2$. Under formation of *STGOOD*, a collection s of indices is stored in *STGOOD* if and only if it is not absorbed by any collection of this set. It is necessary also to delete from *STGOOD* all the collections of indices that are absorbed by s if s is stored in *STGOOD*. Thus, when the algorithm is over, the set *STGOOD* contains all the collections of indices that correspond to GMRTs and only such collections. Essentially the process of forming *STGOOD* is an incremental procedure of finding all maximal elements of a partially ordered set. The set *TGOOD* of all the GMRTs is obtained as follows: $TGOOD = \{t: t = t(s), (\forall s)\ (s \in STGOOD)\}$.

The Estimation of the Number of Subtasks to Be Solved

The number of subtasks at each level of recursion is determined by the number of essential examples in the projection associated with this level. The depth of recursion for any subtask is determined by the greatest cardinality (call it '*CAR*') of set-theoretical intersections of elements $s \in STGOOD$ corresponding to GMRTs: $CAR = \max (||s_i \cap s_j||, \forall(s_i, s_j)\ s_i, s_j \in STGOOD)$. In the worst case, the number of subtasks to be solved is of order $O(2^{CAR})$.

CASCADE: Inferring all GMRTs of Maximal Weight

The algorithm CASCADE serves for inferring all the GMRTs of maximal weight. At the beginning of the algorithm, the values are arranged in decreasing order of weight such that $W(A_1) \geq W(A_2) \geq \ldots \geq W(A_m)$, where $A_1, A_2, \ldots, A_m$ is a permutation of values. The shortest sequence of values $A_1, A_2, \ldots, A_j$, $j \leq m$ is defined such that it is a test for positive examples and *WMIN* is made equal to $W(A_j)$. The procedure DIAGaRa tries to infer all the GMRTs with weight equal to *WMIN*. If such tests are obtained, then the algorithm stops. If such tests are not found, then *WMIN* is decreased, and the procedure DIAGaRa runs again.

Conclusion

In this paper, we used a unified model for inferring implicative logical rules from examples. The key concept of our approach is the concept of a good diagnostic test. We define a good diagnostic test as the best approximation of a given classification on a given set of examples. In the framework of our approach, we show the equivalence between implicative rules and diagnostic tests for a given set of examples. The task of inferring good diagnostic tests from examples serves as an ideal model of inductive reasoning because this task realizes the canons of induction that has been originally formulated by English logician J.-S. Mille.

We have given the decomposition of inferring all good maximally redundant tests for a given set of examples into operations and subtasks that are in accordance with main human common sense reasoning operations. This decomposition allows, in principle, to transform the process of inferring good tests (and imlicative rules) into a "step by step" reasoning process. Incremental algorithms of inferring good classification tests from examples demonstrate the possibility of this transformation in the best way.

We consider two kinds of subtasks: for a given set of positive examples 1) given a positive example *t*, find all GMRTs contained in *t*; 2) given a non-empty collection of values *X* (maybe only one value) such that it is not a test, find all GMRTs containing *X*. The decomposition of good classification tests inferring into subtasks implies introducing a set of special rules to realize the following operations: choosing examples (values) for subtask, forming subtask, deleting values or examples from subtask and some other rules controlling the process of good test inferring. The concepts of an essential value and an essential example are introduced in order to optimize the choice of subtasks of the first and second kinds.

We have described an inductive algorithm DIAGaRa for inferring all good maximally redundant tests for a given set of positive examples. This algorithm realizes one of the possibilities to transform the searching of diagnostic tests (implicative logical rules) into "step by step" learning procedure.

Our approach is also applicable for inferring functional and associative dependencies from data.

Acknowledgements

The author is very grateful to Professor Evangelos Triantaphyllou (Louisiana StateUniversity) who inspired and supported this paper, and to Dr. Giovanni Felici (IASI – Italian National Research Council), for his invaluable advice concerning all the parts of this work.

Appendix

The data to be processed are in Table 13 (the set of positive examples) and in Table 14 (the set of negative examples).

An Example of Using the Algorithm DIAGaRa

We use the algorithm DIAGaRa for inferring all the GMRTs having a weight equal to or greater than *WMIN* = 4 for the training set of the examples represented in Table 13 (the set of positive examples) and in Table 14 (the set of negative examples).

We begin with

$s^* = S(+) = \{\{1\}, \{2\}, \ldots, \{14\}\}$,

$t^* = T = \{A_1, A_2, \ldots\ldots, A_{26}\}$,

SPLUS = {*splus*(A_i): $A_i \in t^*$} (see *SPLUS* in Table 15).

Please observe that *splus*(A_{12}) = {2,3,4,7} and $t(\{2,3,4,7\})$ is a test, therefore, A_{12} is deleted from t^* and *splus*(A_{12}) is inserted into *STGOOD*. Then $W(A_8)$, $W(A_9)$, $W(A_{13})$, and $W(A_{16})$ are less than *WMIN*, hence we can delete A_8, A_9, A_{13}, and A_{16} from t^*. Now t_{10} is not a test and can be deleted. After modifying *splus*(A) for A_5, A_{18}, A_2, A_3, A_4, A_6 A_{20}, A_{21}, and A_{26} we find that *splus*(A_5) = {1,4,7} and $t(\{1,4,7\})$ is a test, therefore, A_5 is deleted from t^* and *splus*(A_5) is inserted into *STGOOD*. Then $W(A_{18})$ turns out to be less than *WMIN* and we delete A_{18}, which implies deleting t_{13}. Next we modify *splus*(A) for A_1, A_{19}, A_{23}, A_4, A_{26} and find that *splus*(A_4) = {2,3,4,7}. A_4 is deleted from t^*. Finally, $W(A_1)$ turns out to be less than *WMIN* and we delete A_1.

We can delete also the values A_2, A_{19}, and A_6 because $W(A_2)$, $W(A_{19})$, and $W(A_6)$ are equal to 4, t(*splus*(A_2)), t(*splus*(A_{19})), and t(*splus*(A_6)) are not tests and, therefore, these values will not appear in a maximally redundant test t with $W(t)$ equal to or greater than 4. After deleting these values we can delete the examples t_9, t_5 because A_{19} is essential in t_9, and A_2 is essential in t_5. Next we can observe that *splus*(A_{23}) = {1,2,12,14} and $t(\{1,2,12,14\})$ is a test, thus A_{23} is deleted from t^* and *splus*(A_{23}) is inserted into *STGOOD*. Now t_{14} and t_1 are not tests and can be deleted. We can delete the value A_{22} because $W(A_{22})$ is now equal to 4, t(*splus*(A_{22})) is not a test and this value will not appear in a maximally redundant test with weight equal to or greater than 4.

Table - 13. The Set of Positive Examples $R(+)$.

index of example	$R(+)$
1	A_1 A_2 A_5 A_6 A_{21} A_{23} A_{24} A_{26}
2	A_4 A_7 A_8 A_9 A_{12} A_{14} A_{15} A_{22} A_{23} A_{24} A_{26}
3	A_3 A_4 A_7 A_{12} A_{13} A_{14} A_{15} A_{18} A_{19} A_{24} A_{26}
4	A_1 A_4 A_5 A_6 A_7 A_{12} A_{14} A_{15} A_{16} A_{20} A_{21} A_{24} A_{26}
5	A_2 A_6 A_{23} A_{24}
6	A_7 A_{20} A_{21} A_{26}
7	A_3 A_4 A_5 A_6 A_{12} A_{14} A_{15} A_{20} A_{22} A_{24} A_{26}
8	A_3 A_6 A_7 A_8 A_9 A_{13} A_{14} A_{15} A_{19} A_{20} A_{21} A_{22}
9	A_{16} A_{18} A_{19} A_{20} A_{21} A_{22} A_{26}
10	A_2 A_3 A_4 A_5 A_6 A_8 A_9 A_{13} A_{18} A_{20} A_{21} A_{26}
11	A_1 A_2 A_3 A_7 A_{19} A_{20} A_{21} A_{22} A_{26}
12	A_2 A_3 A_{16} A_{20} A_{21} A_{23} A_{24} A_{26}
13	A_1 A_4 A_{18} A_{19} A_{23} A_{26}
14	A_{23} A_{24} A_{26}

Table - 14. The Set of Negative Examples $R(-)$.

index of example	$R(-)$
15	A_3 A_8 A_{16} A_{23} A_{24}
16	A_7 A_8 A_9 A_{16} A_{18}
17	A_1 A_{21} A_{22} A_{24} A_{26}
18	A_1 A_7 A_8 A_9 A_{13} A_{16}
19	A_2 A_6 A_7 A_9 A_{21} A_{23}
20	A_{10} A_{19} A_{20} A_{21} A_{22} A_{24}
21	A_1 A_{10} A_{20} A_{21} A_{22} A_{23} A_{24}
22	A_1 A_3 A_6 A_7 A_9 A_{10} A_{16}
23	A_2 A_6 A_8 A_9 A_{14} A_{15} A_{16}
24	A_1 A_4 A_5 A_6 A_7 A_8 A_{11} A_{16}
25	A_7 A_{10} A_{11} A_{13} A_{19} A_{20} A_{22} A_{26}
26	A_1 A_2 A_3 A_5 A_6 A_7 A_{10} A_{16}
27	A_1 A_2 A_3 A_5 A_6 A_{10} A_{13} A_{16}
28	A_1 A_3 A_7 A_{10} A_{11} A_{13} A_{19} A_{21}
29	A_1 A_4 A_5 A_6 A_7 A_8 A_{13} A_{16}
30	A_1 A_2 A_3 A_6 A_{11} A_{12} A_{14} A_{15} A_{16}
31	A_1 A_2 A_5 A_6 A_{11} A_{14} A_{15} A_{16} A_{26}
32	A_1 A_2 A_3 A_7 A_9 A_{10} A_{11} A_{13} A_{18}
33	A_1 A_5 A_6 A_8 A_9 A_{10} A_{19} A_{20} A_{22}
34	A_2 A_8 A_9 A_{18} A_{20} A_{21} A_{22} A_{23} A_{26}
35	A_1 A_2 A_4 A_5 A_6 A_7 A_9 A_{13} A_{16}
36	A_1 A_2 A_6 A_7 A_8 A_{10} A_{11} A_{13} A_{16} A_{18}
37	A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_{12} A_{14} A_{15} A_{16}
38	A_1 A_2 A_3 A_4 A_5 A_6 A_9 A_{11} A_{12} A_{13} A_{16}
39	A_1 A_2 A_3 A_4 A_5 A_6 A_{14} A_{15} A_{19} A_{20} A_{23} A_{26}
40	A_2 A_3 A_4 A_5 A_6 A_7 A_{11} A_{12} A_{13} A_{14} A_{15} A_{16}
41	A_2 A_4 A_5 A_6 A_7 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{19}
42	A_1 A_2 A_3 A_4 A_5 A_6 A_{12} A_{16} A_{18} A_{19} A_{20} A_{21} A_{26}
43	A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16}
44	A_3 A_4 A_5 A_6 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{18} A_{19}
45	A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15}
46	A_1 A_3 A_4 A_5 A_6 A_7 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{23} A_{24}
47	A_1 A_2 A_3 A_4 A_5 A_6 A_8 A_9 A_{10} A_{11} A_{12} A_{14} A_{16} A_{18} A_{22}
48	A_2 A_8 A_9 A_{10} A_{11} A_{12} A_{14} A_{15} A_{16}

Table - 15. The Set *SPLUS* of the Collections *splus*(*A*) for all *A* in Tables 13 and 14.

SPLUS = {*splus*(A_i): $s(A_i) \cap s(+)$, $A_i \in T$}:	
splus(A_*) → {2,8,10}	*splus*(A_{22}) → {2,7,8,9,11}
splus(A_{13}) → {3,8,10}	*splus*(A_{23}) → {1,2,5,12,13,14}
splus(A_{16}) → {4,9,12}	*splus*(A_3) → {3,7,8,10,11,12}
splus(A_1) → {1,4,11,13}	*splus*(A_4) → {2,3,4,7,10,13}
splus(A_5) → {1,4,7,10}	*splus*(A_6) → {1,4,5,7,8,10}
splus(A_{12}) → {2,3,4,7}	*splus*(A_7) → {2,3,4,6,8,11}
splus(A_{18}) → {3,9,10,13}	*splus*(A_{24}) → {1,2,3,4,5,7,12,14}
splus(A_2) → {1,5,10,11,12}	*splus*(A_{20}) → {4,6,7,8,9,10,11,12}
splus(A_+) → {2,3,4,7,8}	*splus*(A_{21}) → {1,4,6,8,9,10,11,12}
splus(A_{19}) → {3,8,9,11,13}	*splus*(A_{26}) → {1,2,3,4,6,7,9,10,11,12,13,14}

Now choose t_6 as a subtask because this positive example is more difficult to be distinguished from the negative examples. By resolving this subtask, we find that t_6 produces a new test t with $s(t)$ equal to {4,6,8,11}. Delete t_6. We can also delete the value A_{21} because $W(A_{21})$ is now equal to 4, t(*splus*(A_{21})) is not a test and this value will not appear in a maximally redundant test with weight equal to or greater than 4.

Now choose t_8 as a subtask because it is essential in the current projection with respect to the subset {2,3,4,7} that corresponds to one of the GMRTs already obtained. By resolving this subtask, we find that t_8 does not produce any new test. Delete t_8. After that we can delete the values A_+, A_7, A_3, and A_{20} and these deletions imply than all of the remaining rows t_2, t_3, t_4, t_7, t_{11}, and t_{12} are not tests.

The list of all the GMRTs for the training set of positive examples is given in Table 16.

Table - 16. The sets *STGOOD* and *TGOOD* for the Examples of Tables 19 and 20.

№	*STGOOD*	*TGOOD*	№	*STGOOD*	*TGOOD*
1	13	A_1 A_4 A_{18} A_{19} A_{23} A_{26}	9	2,7,8	A_+ A_{22}
2	2,10	A_4 A_* A_{26}	10	1,5,12	A_2 A_{23} A_{24}
3	3,10	A_3 A_4 A_{13} A_{18} A_{26}	11	4,7,12	A_{20} A_{24} A_{26}
4	8,10	A_3 A_6 A_* A_{13} A_{20} A_{21}	12	3,7,12	A_3 A_{24} A_{26}
5	9,11	A_{19} A_{20} A_{21} A_{22} A_{26}	13	7,8,11	A_3 A_{20} A_{22}
6	3,11	A_3 A_7 A_{19} A_{26}	14	2,3,4,7	A_4 A_{12} A_+ A_{24} A_{26}
7	3,8	A_3 A_7 A_{13} A_+ A_{19}	15	4,6,8,11	A_7 A_{20} A_{21}
8	1,4,7	A_5 A_6 A_{24} A_{26}	16	1,2,12,14	A_{23} A_{24} A_{26}

Bibliography

[Boldyrev, 1974]N. G. Boldyrev, “Minimization of Boolean Partial Functions with a Large Number of “Don’t Care” Conditions and the Problem of Feature Extraction”, *Proceedings of International Symposium “Discrete Systems”*, Riga, Latvia, pp.101-109, 1974.

[Cosmadakis et al., 1986] S. Cosmadakis, P. C. Kanellakis, N. Spyratos, “Partition Semantics for Relations”, *Journal of Computer and System Sciences*, Vol. 33, No. 2, pp.203-233, 1986.

[Demetrovics and Vu, 1993] J. Demetrovics and D. T. Vu, “Generating Armstrong Relation Schemes and Inferring Functional Dependencies from Relations”, *International Journal on Information Theory & Applications*, Vol. 1, No. 4, pp.3-12, 1993.

[Finn, 1984] V. K. Finn, “Inductive Models of Knowledge Representation in Man-Machine and Robotics Systems”, *Proceedings of VINITI*, Vol. A, pp.58-76, 1984.

[Ganascia, 1989] J.- Gabriel. Ganascia, “EKAW - 89 Tutorial Notes: Machine Learning”, *Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Paris, France, pp. 287-296, 1989.

[Huntala et al., 1999] Y. Huntala, J. Karkkainen, P. Porkka, and H. Toivonen, “TANE: An Efficient Algorithm for Discovering Functional and Approximate Dependencies”, *The Computer Journal*, Vol. 42, No. 2, pp. 100-111, 1999.

[Kuznetsov, 1993] S. O. Kuznetsov, “Fast Algorithm of Constructing All the Intersections of Finite Semi-Lattice Objects”, *Proceedings of VINITI*, Series 2, No. 1, pp. 17-20, 1993.

[Mannila and Räihä, 1992] H. Mannila, and K. – J. Räihä, “On the Complexity of Inferring Functional Dependencies”, *Discrete Applied Mathematics*, Vol. 40, pp. 237-243, 1992.

[Mannila and Räihä, 1994] H. Mannila, and K. – J. Räihä, “Algorithm for Inferring Functional Dependencies”. *Data & Knowledge Engineering*, Vol. 12, pp. 83-99, 1994.

[Megretskaya, 1989] I. A. Megretskaya, “Construction of Natural Classification Tests for Knowledge Base Generation”, in: *The Problem of the Expert System Application in the National Economy*, Kishinev, Moldavia, pp. 89-93, 1988.

[Mille, 1900] J. S. Mille, *The System of Logic*, Russian Publishing Company "Book Affair": Moscow, Russia, 1900.
[Naidenova and Polegaeva, 1986] X. A. Naidenova, J. G. Polegaeva, "An Algorithm of Finding the Best Diagnostic Tests", *The 4-th All Union Conference "Application of Mathematical Logic Methods"*, Theses of Papers, Mintz, G; E, Lorents, P. P. (Eds), Institute of Cybernetics, National Acad. of Sciences of Estonia, Tallinn, Estonia, pp. 63-67, 1986.
[Naidenova and Polegaeva, 1991] X. A. Naidenova, J. G. Polegaeva, "The System of Knowledge Acquisition from Experimental Facts", in: *"Industrial Applications of Artificial Intelligence*", James L. Alty and Leonid I. Mikulich (Eds), Elsevier Science Publishers B.V., Amsterdam, The Netherlands, pp. 87-92, 1991.
[Naidenova, 1992] X. A. Naidenova, "Machine Learning As a Diagnostic Task", in: *"Knowledge-Dialogue-Solution", Materials of the Short-Term Scientific Seminar*, Saint-Petersburg, Russia, editor Arefiev, I., pp.26-36, 1992.
[Naidenova et al., 1995a] X. A. Naidenova, J. G. Polegaeva, J. E. Iserlis, "The System of Knowledge Acquisition Based on Constructing the Best Diagnostic Classification Tests", *Proceedings of International Conference "Knowledge-Dialog-Solution"*, Jalta, Ukraine, Vol. 1, pp. 85-95, 1995a.
[Naidenova et al., 1995b] X. A. Naidenova, M. V. Plaksin, V. L. Shagalov, "Inductive Inferring All Good Classification Tests", *Proceedings of International Conference "Knowledge-Dialog-Solution*", Jalta, Ukraine, Vol. 1, pp.79-84, 1995b.
[Naidenova, 1996] X. A. Naidenova, "Reducing Machine Learning Tasks to the Approximation of a Given Classification on a Given Set of Examples", *Proceedings of the 5-th National Conference at Artificial Intelligence*, Kazan, Tatarstan, Vol. 1, pp. 275-279, 1996.
[Naidenova, 1999] X. A. Naidenova, "The Data-Knowledge Transformation", in: "*Text Procesing and Cognitive Technologies", Paper Collection*, editor Solovyev, V. D., - Pushchino, Russia, Vol. 3, pp. 130-151, 1999.
[Naidenova and Ermakov, 2001] X. A. Naidenova, A. E. Ermakov, "The Decomposition of Algorithms of Inferring Good Diagnostic Tests", *Proceedings of the 4-th International Conference "Computer – Aided Design of Discrete Devices" (CAD DD'2001)*, Institute of Engineering Cybernetics, National Academy of Sciences of Belarus, editor A. Zakrevskij, Minsk, Belarus, Vol. 3, pp. 61-69, 2001.
[Naidenova, 2001] X. A. Naidenova, "Inferring Good Diagnostic Tests as a Model of Common Sense Reasoning", *Proceedings of the International Conference "Knowledge-Dialog-Solution" (KDS'2001)*, State North-West Technical University, Publishing House « Lan », Saint-Petersburg, Russia, Vol. II, pp. 501-506, 2001.
[Ore, 1944] O. Ore, "*Galois Connexions*", Trans. Amer. Math. Society, Vol. 55, No. 1, pp. 493-513, 1944.
[Piaget, 1959] J.Piaget, *La genèse des Structures Logiques Elémentaires*, Neuchâtel, 1959.
[Riguet, 1948] J. Riguet, "Relations Binaires, Fermetures, Correspondences de Galois", *Bull. Soc. Math*., France, Vol. 76., No 3, pp.114-155, 1948.
[Shreider, 1974] J. Shreider, "Algebra of Classification", *Proceedings of VINITI*, Series 2, No. 9, pp. 3-6, 1974.
[Sperner, 1928] E. Sperner, "Eine satz uber Untermengen einer Endlichen Menge". *Mat.Z*., Vol.27, No.11, pp.544-548, 1928.
[Wille, 1992] R. Wille, "Concept Lattices and Conceptual Knowledge System", *Computer Math. Appl*., Vol. 23, No. 6-9, pp. 493-515, 1992.

Author's Information

Naidenova Xenia Alexandrovna - Military medical academy, Saint-Petersburg, Stoikosty street, 26-1-248, naidenova@mail.spbnit.ru.

ПРОГРАММНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Александр Е. Ермаков, Ксения А. Найденова

Аннотация: *Работа знакомит с несколькими программными средствами, используемыми для решения задач интеллектуального анализа данных. В первом разделе рассматриваются специализированные пакеты программ, предназначенные для решения различных задач анализа данных, опыт применения которых свидетельствует о перспективности их использования в современных условиях (ОТЭКС, ОМИС и др.). Во втором разделе описываются несколько инструментальных программных систем, помогающих пользователю создавать свои собственные технологии извлечения знаний из данных, адаптированные к различным условиям, данным и целям анализа в конкретных проблемных областях исследования. Приводятся примеры применения прикладных систем анализа данных в медицине.*

Ключевые слова: *интеллектуальный анализ данных, извлечение знаний из данных*

1. Пакеты прикладных программ для интеллектуального анализа данных

1.1 Пакет прикладных программ ОТЭКС

ОТЭКС предназначен для решения задач, которые встречаются в практике обработки информации наиболее часто [Загоруйко, 1986; Загоруйко и др., 1999]:

1) ТАКСОНОМИЯ. В пакете имеются разные варианты программ из семейства FOREL и KRAB;

2) ВЫБОР СИСТЕМЫ ИНФОРМАТИВНЫХ ПРИЗНАКОВ. В этом разделе есть программы, реализующие идеи алгоритмов NTPP (направленный таксономический поиск признаков) [Загоруйко, 1999] и алгоритмов поиска логических решающих правил.

3) ВЫДЕЛЕНИЕ ГРУПП СВЯЗАННЫХ ПРИЗНАКОВ.

3) РАСПОЗНАВАНИЕ ОБРАЗОВ. Программы основаны на разных вариантах гипотезы компактности: унимодальной, полимодальной, локальной и проективной унимодальной. Реализованы правила из линейных, логических и таксономических классов правил.

4) ЗАПОЛНЕНИЕ ПРОБЕЛОВ И ОБНАРУЖЕНИЕ ОШИБОК В ТАБЛИЦАХ. В пакете имеются программы из двух семейств: ZET и WANGA [Загоруйко, 1999].

5) ПРОГНОЗИРОВАНИЕ. Продолжение динамических рядов осуществляется программой ZET.

Первые версии этого пакета были реализованы более 20 лет тому назад. По мере появления новых вычислительных машин создавались очередные версии пакета. В него добавлялись новые программы, реализующие более эффективные алгоритмы, совершенствовалось сервисное сопровождение пакета, но основной круг решаемых задач оставался практически неизменным. Количество объектов ограничено объемом памяти компьютера. Число объектов может быть сравнимо с числом признаков. Признаки могут быть разнотипными, допускаются ошибки и пробелы в данных. Достоинства пакета - его ориентированность на пользователя, не являющегося программистом и хорошая документированность. Алгоритмы, реализованные в системе ОТЭКС, описаны в монографии «Прикладные методы анализа данных и знаний» [Загоруйко, 1999]. Пакет ОТЭКС применялся во многих областях – социологии, экономике, геологии, технологии, медицине, биологии. Рассмотрим три примера извлечения знаний в генетике, соответствующих различным уровням организации молекулярно-генетических систем (МРНК, белок, генная сеть).

Пример 1. ПРЕДСКАЗАНИЕ КОЛИЧЕСТВЕННОГО УРОВНЯ ТРАНСЛЯЦИОННОЙ АКТИВНОСТИ МРНК: алгоритм ZET [Загоруйко, 1999; Загоруйко и др., 1986]. Выявлялись значимые контекстные характеристики генов, коррелированных с величиной их трансляционной активности. Алгоритм ZET предназначен для прогнозирования значений пропущенных элементов в таблицах данных типа «объект-свойство». На первом этапе работы алгоритма для заданного пробела выбирается «компетентная» подматрица в виде строк, наиболее похожих на ту строку, в которой локализован заданный пробел, и столбцов, наиболее коррелированных со столбцом, в котором локализован заданный пробел. На втором этапе автоматически определяются параметры формулы, используемой для предсказания пропущенного значения. На третьем этапе выполняется непосредственное прогнозирование. Компетентная матрица имела размеры 3*3. Получены предсказания активности для 171-го гена дрожжей. Точность предсказания различна. Наименьшая ошибка прогноза – 20% была получена для 30% генов.

Пример 2. РАСПОЗНАВАНИЕ САЙТОВ: алгоритм AddDel [Загоруйко, 1999]. Автоматическое распознавание сигнальных пептидов и сайтов разрезания в белках является актуальной задачей как для распознавания их внутриклеточной локализации, так и для решения прикладных задач в медицине и биотехнологии. В исследовании использовались 10 свойств Kidera и 434 структурных и физико-химических свойства аминокислот (более подробно решение задачи изложено в [Загоруйко и др., 2002]). В качестве решающего правила использовалось правило «*k* ближайших соседей». Для выбора признаков использовался алгоритм AddDel, который сочетает идеи последовательного добавления наиболее ценных признаков и последовательного удаления наименее ценных. Тестирование экспериментальных данных показало, что сайты разрезания правильно обнаруживаются и локализуются в 85% случаев.

Пример 3. РАСПОЗНАВАНИЕ ТИПА МУТАЦИОННЫХ НАРУШЕНИЙ В ГЕННЫХ СЕТЯХ. Появившиеся в последние годы новые экспериментальные технологии (Laboratories-on-a-chip) позволяют автоматически получать кинетические характеристики функционирования в клетках сотен и тысяч генов и их продуктов. Актуальной задачей анализа этих данных является распознавание типа мутационных нарушений в генных

сетях. Решение этой задачи позволит разрабатывать методы диагностики заболеваний, связанных с нарушением работы генных сетей, и лекарственные препараты узко специализированного воздействия на заданные молекулярно-генетические и биохимические процессы, протекающие в клетке. В работе [Борисова и др., 2002] исследовалась генная сеть регуляции дифференцировки эритроидной клетки под действием эритропротеина. С помощью модели этой сети [Ratushny et. al., 2002] были получены данные об изменении концентраций различных веществ, участвующих в биохимических реакциях. Было промоделировано 19 мутаций, нарушающих работу определенного звена в генной сети, по 10 вариаций для каждой мутации. Разработана методика распознавания принадлежности некоторого состояния сети к одному из 19 типов мутаций. В основу решения задачи распознавания лег принцип парного сравнения эталонов [Загоруйко, 2000]. В результате работы алгоритма выяснилось, что для построения решающего правила, предназначенного для различения всех типов мутаций, необходима информация о концентрации только 3-х компонентов генной сети: гема, рецепторов, связанных на поверхности с трансферрином и МРНК GATA-1 на интервалах времени длиной 11, 23 и 2 часа соответственно. По выбранным характеристикам все контрольные мутации были распознаны безошибочно.

1.2 Интеллектуальная программа для диагностики больных с заболеваниями предстательной железы

Авторами (Сошников Д.В., Лукьянов И.В., Заведеев И.А.) разработан и внедрен в клиническую практику прототип интеллектуальной учетно-диагностической системы в области урологии, который опирается на практику лечения заболеваний предстательной железы в урологической клинике ГКБ им. С.П.Боткина.

Система содержит базу данных (БД) пациентов (в настоящее время 156 больных), включающую помимо симптомов и результатов обследования больных, учетные данные и сведения о лечении, а также интеллектуальную компоненту, обеспечивающую выработку диагноза и рекомендаций по лечению на основании информации из БД. Наличие средств диагностики, встроенных в систему, выгодно отличает данную систему от существующих аналогов (например, учетная система ПРОСТАТА, разработанная в НИИ урологии МЗ РФ) и позволяет начинающему специалисту использовать на начальных этапах диагностирования знания и методику врача-эксперта высокой квалификации. Диагностика проводится системой в 5 этапов: на основании жалоб больного (симптомов) формируется предварительный диагноз, затем на основании инструментальных тестов формируется окончательный диагноз и вырабатываются рекомендации по лечению с учетом возраста, качества жизни и состояния больного. На каждом этапе работает отдельная экспертная система. Первоначальный вариант системы предназначен для автономного использования на рабочем месте врача. Диагностика может осуществляться как с помощью встроенного модуля с фиксированной базой знаний (БЗ), так и при помощи внешней СУБЗ Diet. Встроенный модуль диагностики реализуется при помощи автоматической генерации кода по исходным правилам базы знаний, которые записаны на языке Object Pascal. Было разработано средство генерации кода для несложного продукционного представления знаний с использованием обратного вывода. Полученный интеллектуальный модуль компилируется вместе с системой и не допускает дальнейшего просмотра и модификации правил. Использование внешней СУБЗ Diet, допускающей распределённое хранение и использование знаний, позволяет организовать централизованную модифицируемую БЗ, которая может поддерживаться несколькими специалистами и использоваться для выработки оптимального диагноза. Такой подход удобен для групп специалистов, работающих с одним учетно-диагностическим комплексом.

1.3 ОМИС – система интеллектуального анализа медицинских данных

ОМИС – современная компьютерная технология системного анализа и обобщения клинико-лабораторных данных [Генкин, 1999]. В 1988 году для решения задач автоматизации научных исследований в области физиологии и медицины было организовано малое предприятие. В конце 1988 года предприятие получило заказ от ЦКДЛ 1 Ленинградского мединститута на разработку системы анализа данных в области гематологии – ГЕМА. В 1990 году ГЕМА начала эксплуатироваться. В системе ГЕМА исследовательский модуль для анализа клинико-лабораторных данных был сопряжен с компьютерной историей болезни и с экспертным модулем. Знания экспертного модуля формировались компьютерной программой на основании результатов исследований, проводимых в диалоге с пользователем. Система ГЕМА создана на базе клиники факультетской терапии 1-го Ленинградского мединститута. В ней реализованы новые возможности анализа средних тенденций, корреляционных связей, методы оценки информативности лабораторных и инструментальных признаков, анализ интервальных и бинарных

структур, обеспечивалась информационная поддержка клинических решений. В 1993 году была разработана первая версия оболочки для создания интеллектуальных медицинских систем (программный комплекс ОМИС). ОМИС предоставляет клиницисту возможности самому генерировать компьютерную историю болезни и использовать ресурсы исследовательского и экспертного модулей не только в гематологии, но и в других предметных областях медицины. С помощью ОМИС проводились исследования в области гематологии, пульмонологии, кардиологии, урологии, онкологии, клинической лабораторной диагностики, результаты исследований отражены в ряде диссертационных работ [Бируля, 1998; Дудина, 1995; Киреенков, 1998; Клименкова, 1997; Крутиков, 1996; Кутузов, 1996; Пань Лю Лан, 1996; Степанова, 1996; Филиппова, 1997; Хирманов, 1994]. Высокое качество анализа клинико-лабораторных данных при помощи программного комплекса ОМИС достигается за счет:

1) Сопряжения исследовательского модуля с компьютерной историей болезни.
2) Выбора традиционных статистических функций, ориентированных на анализ сложно организованных данных.
3) Использования ряда дополнительных функций, специально предназначенных для анализа физиологической и клинической информации.
4) Организации анализа динамики данных, в частности, сравнением информационных образов, приуроченных к разным временным срезам.
5) Выявления отношений между элементами физиологических процессов разных типов (ЭЭГ, ЭКГ и др.) и отношений физиологических процессов с клинико-лабораторными данными.
6) Удобного интерфейса, активного диалога клинициста с компьютером и автоматического построения последовательности функций, приводящих к цели исследования.
7) Автоматизации введения вероятностной меры в пространстве признаков (формирование интервальных и бинарных структур) и формирования БЗ в процессе обучения (приобретение знаний из данных истории болезни и протоколов эксперимента).
8) Автоматизированной разработки консилиума (коллектива) алгоритмов принятия решений (6 различных стратегий).
9) Организации экспертной системы, позволяющей принимать дифференциально-диагностические, прогностические и другие решения.

Исследовательский модуль системы ОМИС отличается простотой интерфейса и набором функций, который позволяет проводить количественный анализ патологий у больных клиницисту, не имеющему специальной математической подготовки. Достигается это специальной организацией меню, при обращении к которому от клинициста требуется выбор не конкретной функции, например, *t*-критерия, а лишь интересующей его задачи. Интеллектуальная система ОМИС осуществит предварительный анализ данных и сама выберет статистические функции (как параметрические, так и непараметрические), которые обеспечат решение выбранной задачи. В системе ОМИС автоматизировано формирование интервальных и бинарных структур – новых понятий медицинской информатики [смотри www.intels.spb.ru/pr01.html]. Интервальные структуры лучше, чем другие статистики представляют информацию о вариабельности признака, а бинарные (матричные) структуры оценивают связи между двумя признаками. Они легко модифицируются в процессе обучения при увеличении эмпирического материала. Полностью автоматизированы в системе и все этапы разработки диагностических алгоритмов: формирование обучающих и контрольных выборок, введение вероятностной меры в пространстве признаков, оценка информативности одномерных и двухмерных признаков, формирование оптимального подмножества информационно-ценных признаков при использовании определенного алгоритма, формирование БЗ, сравнение результативности различных алгоритмов, создание консилиума решающих правил и оценка результатов его работы.

2. Инструментальные программные системы для интеллектуального анализа данных

2.1 Интеллектуальный помощник для извлечения знаний [Bernstein, Provost, 2001]

Data Mining в этой системе рассматривается как процесс анализа данных, в котором на разных этапах применяются различные методы. Процесс анализа данных включает 5 основных подпроцессов: выборка данных, разведочный анализ, обработка и трансформация данных, задание и применение модели или вида анализа, анализ результатов. Анализируя результаты каждого этапа, можно в принципе

перенастроить модель или структуру следующей стадии. Можно вернуться назад и провести некоторые стадии анализа заново. Интеллектуальный помощник для извлечения знаний – система IDA (Intelligent Discovery Assistant) создана для того, чтобы помочь разным пользователям (новичкам и профессионалам) осуществить выбор методов обработки данных и обнаружения знаний наиболее подходящим образом с точки зрения вида исходных данных, целей исследования, тех или иных ограничений и пользовательских предпочтений. Система отвечает на вопросы следующего типа: какой метод выбрать - построение дерева решений, метод Байеса или логистическую регрессию? Нужна ли дискретизация? Каким методом? Процесс обнаружения знаний проходит 3 стадии: предварительная обработка данных, применение индуктивных алгоритмов и обработка результатов. Система получает от пользователя описание его данных, целей и желаемые параметры процессов, такие как скорость и точность. Выбор процедур осуществляется для каждой стадии, определяется порядок их выполнения на основе характеристик входных данных, ограничений пользователя и онтологий, то есть формальных определений каждого процесса (оператора). Система формирует возможные планы процесса и пользователю предоставляется возможность выбора плана. Система настраивает все параметры процедур по сформированному плану, осуществляет настройку процесса и генерирует соответствующий код для его выполнения. Онтология для каждого оператора содержит:

1) Информацию для пользователя о каждом операторе.
2) Спецификацию условий, при которых каждый оператор применяется; эти спецификации содержат предусловия, не только связанные с текущим состоянием процесса обработки данных, но и условия согласования оператора с предыдущим процессом.
3) Спецификацию воздействий данного оператора на состояние процесса и данных (постусловия).
4) Оценки влияния оператора на такие характеристики процесса как скорость, точность, модель понимания процесса и т.д.

В дополнение к онтологии все операторы разбиты на логические классы, чтобы уменьшить число рассматриваемых операторов на каждой стадии планирования процесса. В структуре классификации операторов машинного обучения в системе можно выделить три главные группы операторов – предпроцессы, индукция и постпроцессы. Каждая из этих групп подразделяется на подклассы. В листьях классификационного дерева находятся непосредственно исполняемые операторы. Например, индуктивные алгоритмы подразделяются на классификаторы, операторы оценки вероятности класса и блок построения регрессий. Классификаторы далее делятся на решающие деревья и построение правил по примерам. Действующим прототипом системы IDA служит система IDEA. Скорость работы этой системы очень велика. При числе онтологий немногим более дюжины планировщик генерирует все возможные процессы для нескольких сотен проблем с небольшими ограничениями менее чем за секунду. В системе предусмотрены эвристические функции оценки и ранжирования операторов с помощью весовых коэффициентов, формируемых пользователем. Практическая эксплуатация системы IDEA показала, что с её помощью действительно генерируются полезные и интересные для пользователей процессы извлечения знаний. Причем система оказалась неожиданно полезна как для новичков, так и для профессиональных пользователей по той причине, что в ней генерируются сотни планов и она позволяет пользователю уйти от применения только тривиальных, привычных последовательностей обработки и перейти к исследованию новых интересных и раннее не просматриваемых возможностей извлечения знаний. На основе онтологий система предлагает не только прямые пути обработки данных с заданными характеристиками, но и учитывает возможные трансформации данных из одной формы представления в другую. Например, система может преобразовать дерево решений во множество правил и применить к этому множеству оптимизирующую процедуру сокращения правил, которая не применима к решающим деревьям. Байесовский классификатор применим только для категориальных данных, но планировщик включает процесс трансформации данных в необходимую форму представления. Были проведены исследования по оценке эвристических функций ранжирования планов и процессов извлечения знаний в системе IDEA, которые показали очень высокую согласованность системных оценок по ранжированию планов с оценками независимых экспертов. Множество онтологий в IDEA представляет собой мощное средство взаимодействия между специалистами в развитии методов обнаружения знаний в данных. То, что разработано одним исследователем, при включении в онтологию может быть доступно и другим пользователям системы. Это особенно важно при работе многих пользователей в сети. Например, онтологии были пополнены методом двойственного шкалирования [Nishisato, 1994]. Этот метод был

найден одним из специалистов по литературе, опробован для преобразования категориальных данных в числовые и оказался полезен для некоторого класса задач классификации. Неоценимые возможности система IDEA предоставляет для обучения специалистов в области извлечения знаний из данных.

Система IDEA генерирует код для инструментальной системы извлечения знаний WEKA [Witten, Frank, 2000]. WEKA представляет собой коллекцию алгоритмов машинного обучения для решения проблем извлечения знаний (data mining problems) реального мира (задачи с данными большой размерности). Программное обеспечение написано на языке Java и работает почти на всех компьютерных платформах. Алгоритмы можно непосредственно применять к данным или они могут вызываться из программы пользователя. Библиотека программ WEKA также хорошо подходит для развития новых пользовательских методов машинного обучения. Кроме библиотеки программ система WEKA хранит библиотеку наборов данных, полученных из разных источников. Обе библиотеки пополняются пользовательскими программами и наборами данных. Программы и данные WEKA доступны через INTERNET, вместе с программами загружается учебник WEKA и документация для освоения системы. Использование этого ресурса может быть очень полезно, так как в нем сосредоточены программы, реализующие практически все известные методы обработки данных и извлечения знаний.

2.2 Пакет программ Statistica Data Mining

Компанией StatSoft была разработана система Statistica Data Mining. Данная система спроектирована как универсальное средство анализа данных (от взаимодействия с БД до создания готовых отчетов), реализующее графически-ориентированный подход. Statistica Data Mining представляет собой наиболее полный пакет методов Data Mining на рынке программного обеспечения [Большаков, 2003]. Этот пакет обладает большим набором готовых решений, удобным пользовательским интерфейсом, полностью интегрированным с MS Office, мощными средствами разведочного анализа. Statistica Data Mining - оптимальное средство для работы с огромным объемом информации, имеющее гибкий механизм управления. Пакет также обеспечивает многозадачность и имеет открытую архитектуру. Для поддержки пользовательских приложений используется промышленный стандарт VB, Java, C/C++. Основой пакета Statistica Data Mining является браузер, содержащий 300 основных процедур, оптимизированных под задачи Data Mining, и средства логической связи между ними. В пакете предусмотрены средства управления потоками данных, которые позволяют конструировать аналитические методы пользователя. Рабочее пространство пакета разделено на 4 части:

Data Acquisition - Сбор данных. Здесь пользователь идентифицирует источник данных для анализа (файл данных или запрос к БД).

Data Preparation, Cleaning, Transformation – Подготовка, Преобразование и Очистка данных. Здесь данные преобразуются, фильтруются, группируются.

Data Analysis Modelling, Classification, Forecasting – Анализ данных, Моделирование, Классификация, Прогнозирование. Здесь пользователь может при помощи браузера или готовых моделей задать необходимые виды анализа данных.

Reports – Результаты. В данной части пользователь может просмотреть, задать вид и настроить результаты анализа (например, отчет или электронная таблица).

В пакете предлагается широкий набор процедур и методов визуализации данных. Средства анализа, предлагаемые пакетом, можно разделить на две группы – средства анализа преимущественно на основе методов статистики и специализированные средства Data Mining.

Средства анализа на основе методов статистики разделены на 5 основных классов.

Разметка/разбиение и углубленный анализ. Набор процедур этого класса позволяет группировать переменные, вычислять описательные статистики, строить исследовательские графики.

Классификация. Это набор процедур классификации таких, как обобщенные линейные модели, деревья классификации, регрессионные деревья, кластерный анализ.

Обобщенные линейные и нелинейные регрессионные модели. Данный класс процедур содержит обобщенные регрессионные модели и элементы анализа деревьев классификации.

Прогнозирование. Включает модели АРПСС (авторегрессия проинтергированного скользящего среднего), сезонные модели АПРСС, экспоненциальное сглаживание, спектральный анализ Фурье, сезонная декомпозиция, прогнозирование при помощи нейронных сетей и др.

Специализированные процедуры Statistica Data Mining включают:

1) Нейросетевой анализ.
2) Специальную выборку и фильтрацию данных (для больших объемов данных). Модуль может обработать около миллиона входных переменных с целью определения предикторов для регрессии и классификации.
3) Правила ассоциации. Модуль реализует индуктивные методы обнаружения правил ассоциации в данных.
4) Интерактивный углубленный анализ с помощью средств гибкого исследования больших объемов данных.
5) Обобщенный метод максимума среднего и кластеризация методом *K* - средних. Метод ориентирован на кластеризацию больших наборов данных как непрерывной, так и категориальной природы. Обеспечивает предпосылки для распознавания образов.
6) Обобщенные аддитивные модели.
7) Обобщенные классификационные и регрессионные деревья. Модуль реализует методы, разработанные в [Breiman et. al., 1984].
8) Обобщенные CHAID (Chi – square Automatic Interaction Detection) модели (Хи – квадрат автоматическое обнаружение взаимодействия). Модель предназначена для больших объемов данных.
9) Интерактивная классификация и регрессионные деревья.
10) Расширяемые простые деревья (Boosted Trees) - специальный метод построения расширяемых деревьев.
11) Многомерные адаптивные регрессионные сплайны. В пакете эти алгоритмы приспособлены для задач обработки непрерывных и категориальных данных.
12) Критерии согласия как для непрерывных, так и для категориальных данных.
13) Быстрые прогнозирующие модели для большого числа наблюдаемых значений.

Для пользователей, которые слабо разбираются в методах анализа данных, предусмотрены встроенные модули для решения наиболее важных и популярных задач. Более детальное описание пакета можно найти в книгах [Боровиков, 2001; Боровиков, Ивченко, 1999].

2.3 Инструментальное средство для создания и изменения компьютерных систем психофизиологической диагностики

Первая версия инструментального средства (ИС) была разработана в 1996 году для автоматизации процесса создания компьютерных психологических тестов и процедур совместной интерпретации (обработки) результатов применения комплексов (батарей) этих тестов [Naidenova, Ermakov, 1996; Ермаков и др., 1996]. В последующих редакциях в ИС были добавлены возможности автоматизации адаптивных (ветвящихся) диагностических тестов и процедур интерпретации результатов [Ермаков, Найдёнова, 1998]. С самого начала ИС создавалось как элемент технологии, в которую входит также универсальный интерпретатор описаний, сформированных экспертом с помощью ИС. Знания (описания) структурных и функциональных параметров диагностической процедуры (системы) представляются в виде объектно-ориентированной базы, оформленной совокупностью файлов специальной структуры. ИС и интерпретатор описаний содержат систему мета-знаний о правилах работы с описаниями диагностических систем при их создании и применении [Найдёнова и др., 1997]. В ходе дальнейших исследований было установлено, что предложенная модель знаний [Найдёнова и др., 1996] наряду с психодиагностикой может успешно использоваться и для решения ряда задач оценки физиологического состояния человека. При формировании описаний создаваемой диагностической системы в ИС возможны два режима работы эксперта: по сценарию, автоматически формируемому ИС на базе содержащихся в нём мета-знаний и режим выборочного изменения элементов базы описаний по выбору эксперта. С помощью данной технологии был разработан ряд прикладных систем психологической и физиологической диагностики, в том числе в интересах профессионального отбора специалистов и прогнозирования их профессиональной работоспособности в экстремальных условиях деятельности. Одним из объектов базы описаний создаваемых диагностических систем, как правило, являются процедуры перевода значений измеряемых и (или) вычисляемых психологических и физиологических показателей в измерительные шкалы, более удобные для анализа специалистами – процентильную, стенов, станайнов, Т-баллов и ряд других. В ранних версиях ИС для перевода значений первичных шкал в производные (далее –

формирования производных шкал) использовалось явное задание экспертом таблиц пересчета. В явном же виде эксперт должен был задавать и правила продукционного типа, использовавшиеся для управления диагностической процедурой и обработки результатов обследования. Однако в процессе совершенствования технологии в ИС были добавлены возможности интеллектуального анализа данных (ИАД). Во-первых, это автоматизированное формирование ряда производных измерительных шкал на основе обучающих выборок значений диагностических показателей. При этом в ходе формирования стандартизированных шкал (стенов, станайнов, Т-баллов и др.) производится оценка соответствия распределения первичных значений показателя нормальному закону по критерию согласия на основе асимметрии и эксцесса в модификации Фишера [Ллойд, Ледерман, 1989]. Если обучающая выборка согласована с нормальным законом, то по специальным формулам производится непосредственное формирование требуемой стандартизированной шкалы, в противном случае выполняется нормализация выборки за счет нелинейного преобразования её значений – перехода к шкале процентилей, после чего необходимая шкала формируется на базе полученной промежуточной шкалы. Помимо описанной процедуры формирования измерительных шкал в новой версии ИС реализована возможность автоматизированного индуктивного формирования квазиоптимальных комплексов правил – продукций по обучающим выборкам. Для этого используется алгоритм, основанный на поиске наилучших диагностических тестов, аппроксимирующих задаваемые пользователем классификации примеров обучающей выборки [Naidenova, 1992]. Применение перечисленных процедур индуктивного вывода знаний из данных позволило ускорить формирование экспертом баз описаний создаваемых диагностических систем и повысить их адекватность решаемым задачам. В настоящее время ведутся работы по наращиванию описанной компоненты ИАД инструментального средства рядом дополнительных методов, расширяющих его возможности: регрессионного анализа, формирования интегральных показателей на основе взвешенной свертки частных диагностических показателей [Ермаков, Найдёнова, 2001] и рядом других.

Библиография

[Бируля, 1998] Бируля И.В. Лабораторные методы оценки метаболической функции легких и функции состояния почек при бронхиальной астме. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1998.

[Большаков, 2003] Большаков, П. С.. Возможности Statistica Data Mining // Exponenta Pro: математика в приложениях. – 2003. - №1(1). - С. 13 – 16.

[Борисова и др., 2002] Борисова И.А., Загоруйко Н.Г. и др. Диагностика мутации на основе анализа динамики генных сетей // Информационные технологии в генетике. – Новосибирск, 2002. - С. 16 -32.

[Боровиков, Ивченко, 1999] Боровиков В.П., Ивченко Г.И. Прогнозирование в системе Statistica в среде WINDOWS. – М.: Финансы и статистика, 1999. - 382 с.

[Боровиков, 2001] Боровиков В.П. Statistica: искусство анализа данных на компьютере. Для профессионалов. – Санкт-Петербург.: Питер, 2001. - 656 с.

[Виноградов, Галицкий, 2002] Виноградов Д.В., Галицкий Б.А. ИДА для предсказания сдвигов областей на матрице контактов белков // Труды 8-ой национальной конференции по искусственному интеллекту с международным участием (КИИ-2002). - М.: Физматлит, 2002. - Том 1. - С. 94 –102.

[Генкин, 1999] Генкин А. А. Программный комплекс ОМИС как инструмент системного анализа клинико-лабораторных данных (к 10-летию научно-исследовательской фирмы "Интеллектуальные системы") // Клиническая лабораторная диагностика. - 1999. - № 7. - С. 38-48.

[Дудина, 1995] Дудина О.В.. Гомеостаз глюкокортико-стероидных гормонов, магния, кальция, цинка, меди у больных бронхиальной астмой. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1995.

[Ермаков и др., 1996] Ермаков А.Е., Найдёнова К.А., Левич С.Н. Инструментальное средство генерации экспертных психодиагностических систем // Сборник научных трудов 5-ой нац. конфер. с межд. участием «Искусственный интеллект - 96», Казань, 5 -11 октября 1996 г., с.422-425.

[Ермаков, Найдёнова, 1998] Ермаков А.Е., Найдёнова К.А. Концепция автоматизированной разработки адаптивных компьютерных систем психологической и физиологической диагностики // Морской медицинский журнал, № 6, 1998 г., с.29 - 34

[Ермаков, Найдёнова, 2001] Ермаков А.Е., Найденова К.А. Интегральная оценка психологических и физиологических параметров человека // Проблемы реабилитации. - Санкт-Петербург, № 1(4), 2001, с.132-139.

[Загоруйко и др., 1986] Загоруйко Н.Г., Ёлкина В.Н., Емельянов С.В., Лбов Г.С. Пакет прикладных программ ОТЭКС (для анализа данных). - М.: Финансы и статистика, 1986. – 160 с.

[Загоруйко, 1999] Загоруйко Н.Г. Прикладные методы анализа данных и знаний. - Новосибирск: Изд-во института математики, 1999. – 270 с.

[Загоруйко, 2000] Загоруйко Н.Г. Распознавание образов методом попарного сравнения эталонов в компетентных подпространствах признаков // Доклады АН, М.: Наука, 2000. - Том 382. - №1. - С. 24-26.

[Загоруйко и др., 2002] Загоруйко Н.Г., Кутенко О.А., Иванесенко В.А. Распознавание наличия и места локализации сайта разрезания в сигнальных пептидах // Информационные технологии в генетике. – Новосибирск, 2002. - С. 32-46.

[Киреенков, 1998] Киреенков И.С. Артериальная гипертензия у больных бронхиальной астмой с диаритмиями и влияние электрокардиостимуляции на артериальное давление. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1998.

[Клименкова, 1997] Клименкова С.Ф. Легочный фактор переноса и его компоненты у больных бронхиальной астмой. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1997.

[Крутиков, 1996] Крутиков А.Н. Роль гемодинамических и нейро-гормональных факторов в патогенезе сердечной недостаточности при объемной перегрузке сердца. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1996.

[Кутузов, 1996] Кутузов А.Э. Применение проб с изометрической физической нагрузкой у больных инфарктом миокарда на этапах реабилитации. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1996.

[Ллойд, Ледерман, 1989] Справочник по прикладной статистике. В 2-х т. Т. 1: Пер. с англ. / Под ред. Э. Ллойда, У. Ледермана, Ю.Н. Тюрина. – М.: Финансы и статистика, 1989. – 510 с.

[Найдёнова и др., 1996] Найдёнова К.А., Ермаков А.Е., Маклаков А.Г. и др. Модель знаний для автоматизированного проектирования экспертных психодиагностических систем (доклад) // Сб. научных трудов 5-ой национальной конференции с межд. участием «Искусственный интеллект - 96», Казань, 5 -11 октября 1996 г., т.2, с.275-279.

[Найдёнова и др., 1997] Найдёнова К.А., Ермаков А.Е., Левич С.Н. Генерация психодиагностических экспертных систем // Материалы 2-ой международной конференции «Автоматизация проектирования дискретных систем» ноябрь 1997 г., Минск, том 2, с. 214 – 221.

[Пань Лю Лан, 1996] Пань Лю Лан. Динамика глюкокортикоидной активности при лечении больных бронхиальной астмой методами традиционной китайской медицины. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1996.

[Степанова, 1996] Степанова И.А. Обоснование зависимости гемодепрессивного эффекта от различных вариантов комбинированного лечения лимфогранулематоза. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1996.

[Филиппова, 1997] Филиппова Н.А. Иридологические исследования в оценке состояния больных бронхиальной астмой. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1997.

[Хирманов, 1994] Хирманов В.Н. Натрийуретические гормоны и их роль в нарушении мембранного транспорта натрия и патогенезе некоторых форм артериальных гипертензий. - Автореферат диссертации канд. мед. наук. – Санкт-Петербург, 1994.

[Bernstein, Provost, 2001] Bernstein, A. and Provost, F. An Intelligent Assistant for Knowledge Discovery Process. The paper was presented at IJCAI 2001 Workshop on Wrappers for Performance Enhancement in Knowledge Discovery in Databases. Working Paper of the Center for Digital Economy Research, New York, University – Leonard Stern School of Business, CeDER Working Paper # IS-01-01.

[Breiman et. al., 1984] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. Classification and Regression Trees. Wadsworth, Belmont, 1984.

[Naidenova, 1992] Naidenova, X.A. Machine Learning as a Diagnostic Task. // Knowledge-Dialog-Solution. Materials of Scientific and Technical Seminar, Arefiev, I. B. editor. June 16-17, 1992, pp. 26-36.

[Naidenova, Ermakov, 1996] Naidenova K., Ermakov A. CASE-technology for psychodiagnostic // Proceedings of Second Joint Conference on Knowledge - Based Software Engineering, Sozopol, Bulgaria, Sept. 21-22, 1996, p.246-250

[Ratushny et. al., 2002] Ratushny, A.V., Podkolodnaya, O.A. Ananko, E.A., Likhoshvai, V.A. Mathematical Model of Erithroid Cell Differentiation Regulation // Proc. of the 2nd Intern. Conference on Bioinformatics of Genome Regulation and Structure. - Novosibirsk, Russia. - Vol. 1. - P. 203-206.

[Witten, Frank, 2000] I.H.Witten, E. Frank. Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations. San Francisco, Morgan Kaufmann, 2000.

Информация об авторах

Ермаков Александр Евгеньевич – Военно-медицинская академия; Санкт-Петербург, ул. Сикейроса, дом 19, корп. 2, кв. 55; e-mail: alerma@rambler.ru

Найденова Ксения Александровна – Военно-медицинская академия; Санкт-Петербург, ул. Стойкости, дом 26, корпус 1, кв. 248; e-mail: naidenova@mail.spbnit.ru

МОДУЛЬ ФОРМИРОВАНИЯ ТАБЛИЦ СООТВЕТСТВИЯ ИЗМЕРИТЕЛЬНЫХ ШКАЛ В ПОДСИСТЕМЕ ИНДУКТИВНОГО ВЫВОДА ЗНАНИЙ ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ИНСТРУМЕНТАЛЬНОГО СРЕДСТВА

Александр Е. Ермаков, Вадим А. Ниткин

Аннотация: *Описывается разработанный авторами подход к формированию таблиц соответствия значений первичных и стандартизированных измерительных шкал психологических и физиологических показателей с учетом статистического закона распределения этих значений, реализованный в подсистеме индуктивного вывода знаний проблемно-ориентированного инструментального средства, используемого для автоматизированного создания компьютерных систем психологической и физиологической диагностики. Рассмотрен алгоритм и особенности формирования этих таблиц.*

Ключевые слова*: стандартизированная шкала, таблица соответствия значений измерительных шкал, процентиль, диагностический показатель, эмпирический закон распределения значений показателя, обучающая выборка значений показателя, критерий нормальности распределения значений показателя*

Введение

Рассматриваемый модуль формирования таблиц соответствия измерительных шкал входит в подсистему индуктивного вывода знаний из данных проблемно-ориентированного инструментального средства [Ermakov, Naidenova, 1998; Ермаков, Найдёнова, 1999], предназначенного для автоматизированного создания определённого класса компьютерных систем психологической и физиологической диагностики. Эти диагностические системы используются для решения диагностических задач с адаптивной (ветвящейся) структурой диагностической процедуры.

Кроме данного модуля в подсистему индуктивного вывода знаний инструментального средства входит модуль автоматизированного формирования минимальных наборов правил-продукций, описывающих закономерности в обучающей выборке и применяемых при вычислении значений диагностических показателей в процедурах, строящихся на основе комплексов правил.

Инструментальное средство снабжено проблемно-ориентированным пользовательским интерфейсом, обеспечивает эксперту возможность быстрой спецификации создаваемых диагностических систем и автоматизированного формирования баз знаний, исчерпывающе описывающих их структурные и функциональные параметры.

Сформированная база знаний (описаний) затем используется программой - интерпретатором описаний для сбора диагностической информации и её обработки по содержащимся в этой базе правилам и алгоритмам. Инструментальное средство и интерпретатор описаний содержат систему мета-знаний о правилах работы с описаниями диагностических систем при их формировании и практическом применении [Найдёнова и др., 1997]. Используемая модель знаний описывает все структурные и функциональные параметры диагностических задач, решаемых создаваемой диагностической системой.

Одним из них является множество измеряемых и вычисляемых психологических и физиологических показателей, каждый из которых связан с одной или несколькими измерительными шкалами (первичной и дополнительными). Другим объектом модели знаний, используемой при формировании базы описаний диагностической системы, являются процедуры пересчета значений психологических и физиологических показателей из первичных измерительных шкал в шкалы, более удобные для восприятия психологами и врачами – процентилей, станайнов, стенов, Т-баллов и ряд других.

Рассматриваемый модуль позволяет эксперту, формирующему базу описаний, автоматизировать процесс получения новых знаний (таблиц соответствия значений измерительных шкал) из данных обучающих выборок и включить эти таблицы в состав базы знаний создаваемой диагностической системы.

Стандартизированные шкалы

Практика показывает, что первичные результаты различных психологических тестов и процедур оценки физиологических характеристик обычно выражаются в разнотипных измерительных шкалах. Более наглядный вид результаты этих тестов и процедур приобретают при использовании однотипных, сопоставимых шкал, для чего первичные результаты подвергают определённым преобразованиям. Одним из таких преобразований является стандартизация - приведение первичных значений показателей к специальной единой измерительной шкале: Т-баллов, станайнов, стенов или др. [Анастази, 1982; Глушко, 1994; Кулагин, 1984; Осипов, 1976]. Использование стандартизированных измерительных шкал облегчает и формирование различных интегральных оценок психофизиологического состояния обследуемых.

Если закон распределения первичных значений тестовых показателей может считаться нормальным в рамках используемого критерия (критериев) согласия [Ллойд, Ледерман, 1989], то приведение значений этих показателей к стандартизированной шкале осуществляется при помощи линейных преобразований [Анастази, 1982]. При этом обычно переход к стандартизированной шкале осуществляется через шкалу стандартных z-оценок, полученную в результате Z – преобразования, выполняемого по формуле:

$$z = (x - M) / \sigma,$$

где z - стандартная величина, x - первичный результат тестового измерения, M - среднее арифметическое значений первичных результатов, σ - стандартное отклонение этих значений.

Недостатком шкалы z-оценок является то, что в ней приходится оперировать отрицательными и дробными величинами. Поэтому от шкалы z-оценок обычно переходят к более удобной в обращении нормализованной шкале. Для этого используется допустимое на уровне интервальной шкалы линейное преобразование типа

$$s = az + b,$$

где a и b - действительные числа, выбор которых определяется удобством дальнейшей работы со шкалой. При психологическом тестировании обычно используются следующие стандартизированные измерительные шкалы [Анастази, 1982]:

1) шкала Т-баллов (Т-шкала Мак-Колла): $T = 50 + 10 (x - M) / \sigma = 50 + 10z$;
2) шкала стенов Кэттелла: $ST = 5{,}5 + 2 (x - M) / \sigma = 5{,}5 + 2z$;
3) шкала станайнов Гилфорда: $C = 5 + 2 (x - M) / \sigma = 5 + 2z$;
4) шкала структуры интеллекта Амтхауэра: $Z = 100 + 10 (x - M) / \sigma = 100 + 10z$;
5) шкала Векслера: $JQ = 100 + 15 (x - M) / \sigma = 100 + 15z$;
6) шкала оценок Линерта: $SN = 3 + (x - M) / \sigma = 3 + z$.

Для проверки гипотезы о нормальности эмпирического распределения, как известно, может быть использован целый ряд критериев согласия: критерий асимметрии и эксцесса, D_n - критерий Колмогорова, критерий χ^2 Пирсона, критерий Шапиро-Уилка и др. [Ллойд, Ледерман, 1989]. После предварительного сравнительного анализа этих критериев для оценки нормальности эмпирических распределений значений диагностических показателей в модуле формирования таблиц соответствия измерительных шкал мы выбрали критерий, базирующийся на расчете выборочной асимметрии и эксцесса в модификации Фишера [Ллойд, Ледерман, 1989], в силу его алгоритмической простоты и высокой прогностичности.

При несоответствии распределения первичных значений диагностического показателя нормальному закону в рамках используемого критерия согласия, можно идти двумя путями.

1. Изменить само диагностическое средство, например психологический тест или тест оценки знаний таким образом, чтобы добиться соответствия распределения значений показателя нормальному закону.
2. Выполнить принудительную нормализацию исходного, отличного от нормального распределения, с помощью некоторого нелинейного преобразования первичных значений показателя.

Первый путь, по мнению большинства авторов [Анастази, 1982; Глушко, 1994], предпочтителен, но его рассмотрение выходит за рамки данной работы. На практике нередко возникает ситуация, когда тест задан заранее и его по каким-то причинам нельзя заменить другим, более адекватным обследуемому контингенту.

Проанализировав известные нелинейные преобразования диагностических данных, применяемые для их нормализации, мы остановились на процедуре расчета процентилей, являющихся простой и содержательно достаточно понятной числовой характеристикой. Поэтому алгоритм формирования таблиц соответствия первичных и стандартизированных измерительных шкал диагностических показателей, используемый нами в модуле индуктивного вывода знаний из данных, можно описать последовательностью из 5 шагов.

1. Если целевая (требуемая на данном этапе вычислений) шкала - процентильная, то таблица соответствия значений первичной шкалы этой шкале формируется сразу по приводимым далее формулам.
2. Если целевая шкала является одной из выше перечисленных стандартизированных шкал, то по критерию согласия оценивается соответствие эмпирического закона распределения показателя нормальному закону.

Если эмпирический закон распределения можно считать нормальным в рамках используемого критерия согласия, то переходим к шагу 3, иначе – к шагу 4.

3. По ранее приведенным формулам первичные значения оцениваемого показателя преобразуются в z-оценки, а затем - в оценки требуемой стандартизированной шкалы и работа алгоритма завершается.
4. По приведенным далее формулам производится пересчет первичных значений показателя в процентили (значения процентильной шкалы), после чего переходим к шагу 5.
5. Всегда нормально распределенные процентильные значения оцениваемого диагностического показателя преобразуются в значения требуемой стандартизированной шкалы.

Критерий нормальности эмпирического распределения

Для оценки соответствия выборочного распределения значений диагностического показателя нормальному закону, согласно рекомендациям [Ллойд, Ледерман, 1989], необходимо сформировать обучающую выборку и для неё рассчитать следующие характеристики.

1. Среднее арифметическое m_1 значений показателя.
2. Центральные моменты (моменты выборки относительно среднего) 2-го, 3-го и 4-го порядка, соответственно обозначенные m_2, m_3 и m_4.
3. Модифицированные в соответствии с рекомендациями Фишера оценки g_1 и g_2 выборочных коэффициентов асимметрии и эксцесса.
4. Стандартные отклонения σ_1 и σ_2 оценок g_1 и g_2.

Для расчета перечисленных величин используются следующие формулы [Ллойд, Ледерман, 1989]:

$$m_1 = \sum_{i=1}^{k} (f_i \times x_i) / n,$$

где x_i – i-тое значение оцениваемого показателя, f_i – частота появления значения x_i в обучающей выборке, n – объем обучающей выборки, k – количество различных значений x_i в выборке, i – порядковый № значения показателя x_i., $\times$ и / - обозначения операций умножения и деления. Очевидно, что

$$n = \sum_{i=1}^{k} f_i.$$

Обучающей будем называть выборку первичных значений оцениваемого показателя, используемую для формирования требуемой стандартизированной измерительной шкалы. Согласно литературным данным [Осипов, 1976; Ллойд, Ледерман, 1989] каждый критерий согласия имеет границы применения в плане объема обучающей выборки, достаточного для получения удовлетворительных погрешностей при оценивании статистических характеристик генеральной совокупности. Для используемого нами критерия минимальный объем выборки составляет 30 наблюдений. Оценить погрешности, возникающие при формировании стандартизированных шкал, производимом на основе обучающих выборок различного объема, достаточно сложно. Поэтому мы придерживаемся следующей итерационной схемы расчета

таблиц соответствия первичных и стандартизированных измерительных шкал психологических и физиологических показателей, используемых при оценке психофизиологического состояния обследуемых.

1. Производится накопление данных о значениях требуемого показателя, частотах их появления и формируется первоначальная обучающая выборка доступного объема из не менее 30 наблюдений.
2. По данным обучающей выборки производится формирование таблиц соответствия измерительных шкал.
3. Сформированная таблица используется в течение определенного интервала времени и одновременно осуществляется накопление новых данных для формирования расширенной обучающей выборки.
4. Производится переход к п.2 и т.д.

Процесс уточнения таблицы соответствия шкал может быть остановлен после прекращения её изменения в результате очередного расширения обучающей выборки или продолжен при наличии объективных предпосылок к её дальнейшему изменению.

Значение m_2 – несмещенной оценки дисперсии выборки рассчитывается по формуле:

$$m_2 = \sum_{i=1}^{k} f_i \times (x_i - m_1)^2 / (n-1).$$

Значения выборочных моментов m_3 и m_4 вычисляются по формуле:

$$m_s = \sum_{i=1}^{k} f_i \times (x_i - m_1)^s / n, \quad s = 3, 4.$$

Значения оценок g_1 и g_2 рассчитываются по формулам [Ллойд, Ледерман, 1989]:

$$g_1 = k_3 / k_2^{3/2}, \quad g_2 = k_4 / k_2^2, \quad k_2 = m_2 / (1-1/n), \quad k_3 = m_3 / \{(1-1/n)\times(1-2/n)\},$$
$$k_4 = m_4 / \{(1 - 2/[n+1])\times(1 - 2/n)\times(1-3/n)\} - 3\times m_2^2 / \{(1-2/n)\times(1-3/n)\}.$$

Здесь через n, как и ранее, обозначен объем обучающей выборки; k_2, k_3, k_4 – промежуточные величины, формируемые на основе моментов m_2, m_3 и m_4. Для расчета значений стандартных отклонений σ_1 и σ_2 используются формулы [Ллойд, Ледерман, 1989]:

$$\sigma_1 = [6\times n\times(n-1) / \{(n+3)\times(n+1)\times(n-2)\}]^{1/2}, \quad \sigma_2 = [24\times n\times(n-1)^2 / \{(n+5)\times(n+3)\times(n-2)\times(n-3)\}]^{1/2}.$$

Критерием нормальности анализируемой выборки является соотношение абсолютных значений (модулей) оценок g_1 и g_2 с абсолютными величинами стандартных отклонений σ_1 и σ_2 [Ллойд, Ледерман, 1989]. При этом если $|g_1| < |\sigma_1|$ и $|g_2| < |\sigma_2|$, то g_1 и g_2 не являются значимыми и данные обучающей выборки согласованы с гипотезой о нормальности. Если же $|g_1| \geq |\sigma_1|$ или $|g_2| \geq |\sigma_2|$, то данные обучающей выборки считаются не согласованными с гипотезой о нормальности.

Расчет процентилей

Процентиль - процентная доля измерений из обучающей выборки, величина которых ниже или выше данного значения первичного показателя (в зависимости от физического смысла показателя). Процентили указывают на относительное положение данного значения в выборке. По мнению специалистов [Анастази, 1982; Кулагин, 1984] расчет процентилей можно проводить при объеме обучающей выборки не менее 100 человек.

При разработке алгоритма мы исходили из предположения, что обучающая выборка значений диагностического показателя может быть как полной, т.е. содержать все его возможные значения (в случае дискретных величин), появившиеся с некоторыми частотами, так и неполной - содержать лишь некоторое подмножество множества возможных значений. При этом, исходя из практики, мы считали возможными ситуации, когда процентильная шкала, сформированная на основе неполной обучающей выборки, затем используется для пересчета первичных значений показателя, не вошедших в обучающую выборку. Поэтому в реализованном алгоритме нами предусмотрена возможность линейной интерполяции с произвольным шагом для отсутствующих в обучающей выборке первичных значений показателей и формирования таблицы соответствия значений первичной шкалы показателя шкале процентилей, а также

построенным на её основе или напрямую стандартизированным измерительным шкалам даже в случае неполных обучающих выборок. Для этого использовалось известное в аналитической геометрии уравнение прямой, проходящей через две точки $A(x_1, y_1)$ и $B(x_2, y_2)$ [Циркунов, 1966]. В данном случае в качестве точек A и B брались соседние первичные значения показателя x_1, x_2, имеющиеся в обучающей выборке, и соответствующие им уже рассчитанные значения процентилей P_1, P_2. Формула для расчета значения искомого промежуточного процентиля P, соответствующего первичному значению x, отсутствующему в обучающей выборке, имеет вид:

$$P = \text{Round}\ \{(x - x_1)\times(P_2 - P_1) / (x_2 - x_1) + P_1\}.$$

Здесь через Round{C} обозначена функция округления значения C до ближайшего целого (в большую или в меньшую сторону).

Процентиль для первичного значения показателя, имеющегося в обучающей выборке, вычисляется по формуле:

$$P = \text{Round}\ \{100 \times f_{cum} / n\},$$

где f_{cum} – кумулятивная (накопленная) частота, соответствующая первичному значению показателя x, для которого ищется процентиль P. Частота f_{cum} равна сумме частот всех значений показателя, предшествующих данному, плюс частота появления в обучающей выборке данного значения x.

Линейная интерполяция используется нами, если необходимо, и при прямом расчете стандартизированных шкал, когда закон распределения первичных значений - нормальный.

При переходе от первичных значений показателя к процентилям, с последующим их пересчетом в значения стандартизированных шкал или без такового, нами предусмотрена возможность сортировки первичных значений $\{x_i\}$ по возрастанию или убыванию в зависимости от того, должен ли наибольший процентиль соответствовать наибольшему первичному значению показателя или наименьшему. Выбор типа сортировки определяется содержательным смыслом показателя и особенностями решаемой диагностической задачи.

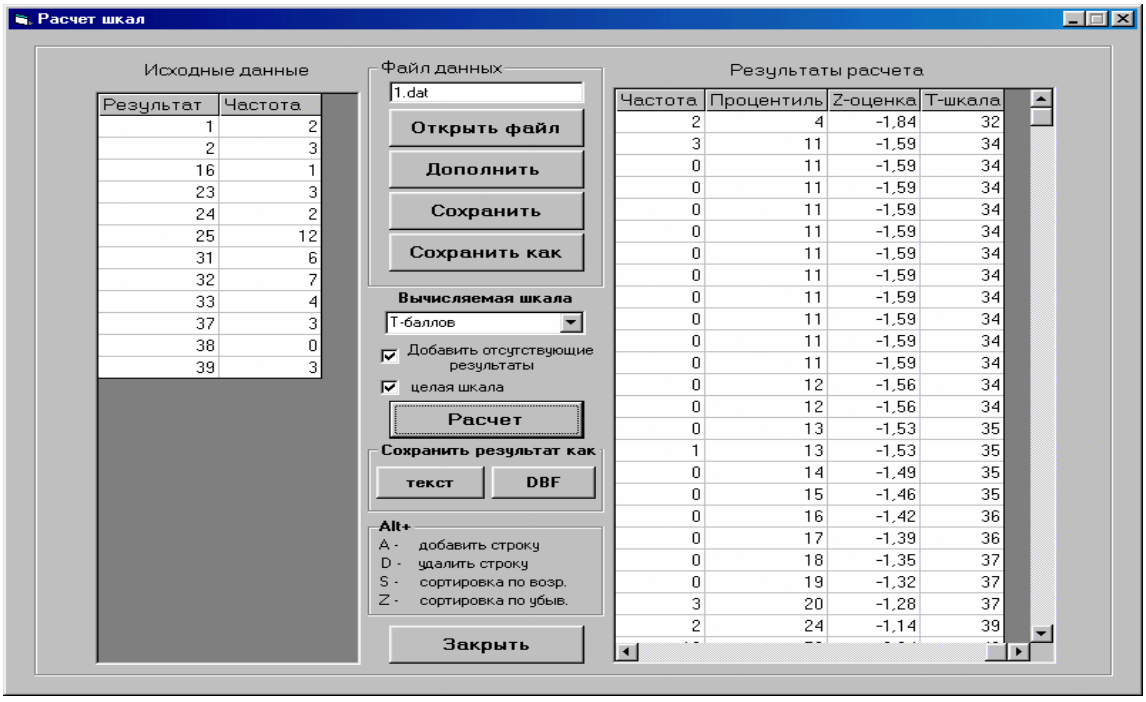


Рис.1 Экранная форма программного модуля индуктивного формирования таблиц соответствия значений измерительных шкал

На рис. 1 приведена экранная форма программного модуля индуктивного формирования таблиц соответствия значений первичных шкал психологических и физиологических показателей стандартизированным на примере формирования Т-шкалы Мак-Колла. Формирование Т-шкалы производится на основе неполной обучающей выборки первичных значений показателя (см. таблицу «Исходные данные»), не подчиняющихся нормальному закону распределения. Как и другие процедуры инструментального средства, модуль запрограммирован на языке Visual Basic 6.0.

Выводы

1. Предложенный подход к автоматизированному формированию таблиц соответствия значений первичных и стандартизированных измерительных шкал психологических и физиологических показателей позволяет обеспечить их корректное формирование даже в случае, если закон распределения первичных значений показателей не соответствует нормальному, за счет перехода к промежуточной шкале процентилей - множеству процентильных значений показателя, всегда распределенных по нормальному закону.

2. Использование метода линейной интерполяции обеспечивает пользователю возможность пересчета первичных значений диагностических показателей в процентильную и стандартизированные измерительные шкалы даже при неполных обучающих выборках.

3. Расчет таблиц соответствия значений первичных и стандартизированных измерительных шкал целесообразно осуществлять на итерационной основе, периодически уточняя их по мере увеличения объема обучающей выборки.

Литература

[Анастази, 1982] Анастази А. Психологическое тестирование: Книга 1, Пер. с англ./ Под ред. К.М.Гуревича, В.И.Лубовского. - М.: Педагогика, 1982. - 320 с.

[Глушко, 1994] Глушко А.Н. Основы психометрии. - М.: МО, 1994. – 100 с.

[Дюк, 1994] Дюк В.А. Компьютерная психодиагностика - СПб.: «Братство», 1994 - 364 с.

[Ермаков и др., 1996] Ермаков А.Е., Найдёнова К.А., Левич С.Н. Инструментальное средство генерации экспертных психодиагностических систем // Сборник научных трудов 5-ой национальной конференции с международным участием «Искусственный интеллект - 96», Казань, 5 -11 октября 1996 г., с.422-425.

[Ермаков, Найдёнова, 1999] Ермаков А.Е., Найдёнова К.А. Технология автоматизированной разработки адаптивных компьютерных систем психологической и физиологической диагностики (доклад) // Материалы 3-ей международной конференции «Автоматизация проектирования дискретных систем», ноябрь 1999 г., Минск, т. 3, с.72-79.

[Ermakov, Naidenova, 1998] A tool for adaptive programming the applied psychodiagnostic systems // International conference «Knowledge - Dialog - Solution» (KDS-98), Szczecin, sept. 1998, p. 91-98.

[Кулагин, 1984] Кулагин Б.В. Основы профессиональной психодиагностики - Л.: Медицина, 1984. – 215 с.

[Ллойд, Ледерман, 1989] Справочник по прикладной статистике. В 2-х т. Т. 1: Пер. с англ. / Под ред. Э. Ллойда, У. Ледермана, Ю.Н. Тюрина. – М.: Финансы и статистика, 1989. – 510 с.

[Найдёнова и др., 1997] Найдёнова К.А., Ермаков А.Е., Левич С.Н. Генерация психодиагностических экспертных систем // Материалы 2-ой международной конференции «Автоматизация проектирования дискретных систем» ноябрь 1997 г., Минск, том 2, с. 214 – 221.

[Осипов, 1976] Рабочая книга социолога / Под ред. Г.В. Осипова. – М.: Наука, 1976. – 511 с.

[Циркунов, 1966] Циркунов А.Е. Сборник математических формул – Минск: «Вышэйшая школа», 1966. – 179 с.

Информация об авторах

Ермаков Александр Евгеньевич – Военно-медицинская академия, 194354, Санкт-Петербург, ул. Сикейроса, дом 19, корп.2, кв.55, e-mail: alerma@rambler.ru

Ниткин Вадим Алексеевич – Военно-медицинская академия, 193275, Санкт-Петербург, ул. Верности, дом 10, кв.271, e-mail: vanit@rambler.ru