

УДК 007:519.816

**РЕАЛИЗАЦИЯ ПРЕЦЕДЕНТНОГО МОДУЛЯ
ДЛЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ¹**

Р.В. Алехин (*r.alekhin@gmail.com*)

П.Р. Варшавский (*VarshavskyPR@mpei.ru*)

ФГБОУ ВПО «Национальный исследовательский университет
«МЭИ», Москва

В работе затрагиваются вопросы реализации прецедентного модуля для интеллектуальной системы поддержки принятия решений (ИСППР). Предложен метод поиска решения на основе прецедентов с использованием теории структурного отображения и онтологического подхода. Рассмотрена возможность решения задачи экспертной диагностики состояний сложного объекта с использованием прецедентного модуля в прототипе ИСППР для одной из подсистем энергоблока.

Введение

Большой интерес к моделям и методам, базирующимся на прецедентном подходе, обусловлен тем, что на сегодняшний день весьма актуальна проблема моделирования человеческих рассуждений (рассуждений так называемого «здорового смысла») для интеллектуальных систем поддержки принятия решений (ИСППР). ИСППР принадлежат к классу интегрированных систем, сочетаю-

¹ Работа выполнена при финансовой поддержке РФФИ (проекты № 12-07-00508, № 14-01-00427).

ших строгие математические методы и модели поиска решения с нестрогими, эвристическими моделями и методами, базирующимися на экспертных знаниях, моделях человеческих рассуждений и накопленном системой опыте (прецедентах) [Вагин и др., 2008]. ИСППР предназначены для помощи лицам, принимающим решения (ЛПР), при управлении сложными объектами и процессами различной природы в реальном масштабе времени при наличии в имеющейся информации (данных и знаниях) различного рода неопределенностей, называемых для краткости НЕ-факторами.

Методы поиска решения на основе прецедентов (СВР – Case-Based Reasoning) и аналогий (АВР – Analogy-Based Reasoning) успешно используются в различных областях человеческой деятельности (медицина, техника, юриспруденция и др.), а также в системах экспертного диагностирования, машинного обучения и ИСППР. Применение подобных методов в ИСППР позволяет ЛПР в реальном масштабе времени при наличии разного рода НЕ-факторов как в исходной информации, получаемой от объекта и среды, так и в экспертных знаниях корректно диагностировать ситуацию на объекте и принимать адекватные управляющие воздействия в различных нештатных (аномальных) ситуациях.

В работе основное внимание уделяется разработке прецедентного модуля (СВР-модуля) с использованием механизмов структурного отображения [Falkenhainer et al., 1989] и онтологического подхода [Гаврилова и др., 2000], а также применению СВР-модуля в составе прототипа ИСППР для решения задач диагностики состояний сложного объекта и обнаружения управляющих воздействий на примере системы автоматического охлаждения зоны реактора (САОЗ) в контуре водо-водяного энергетического реактора (ВВЭР) АЭС [Башлыков и др., 2013].

1. Прецедентный подход и особенности представления прецедентов

Прецедентный подход основывается на накоплении опыта и последующей адаптации решения известной задачи к решению новой. Прецедент можно определить как случай, имевший место ранее и служащий примером или оправданием для последующих случаев подобного рода.

Как правило, CBR-методы включают в себя четыре основных этапа, образующие так называемый CBR-цикл [Варшавский, 2009]:

- *retrieve* – извлечение наиболее соответствующего (подобного) прецедента (или прецедентов) для сложившейся ситуации из библиотеки прецедентов (БП);
- *reuse* – повторное использование извлеченного прецедента для попытки решения текущей проблемы (задачи);
- *revise* – пересмотр и адаптация в случае необходимости полученного решения в соответствии с текущей проблемой (задачей);
- *retention* – сохранение вновь принятого решения как части нового прецедента.

В общем случае модель представления прецедента включает описание ситуации, решение для данной ситуации и результат применения решения [Еремеев и др., 2012]: *CASE* = (*Situation, Solution, Result*), где *Situation* – ситуация, описывающая данный прецедент, а *Solution* – решение (например, диагноз и рекомендации ЛПР), *Result* – результат применения решения, который может включать список выполненных действий, дополнительные комментарии и ссылки на другие прецеденты, а также в некоторых случаях могут приводиться обоснование выбора данного решения и возможные альтернативы. В различных моделях представления прецедентов используются разные способы описания указанных компонент. Прецеденты могут быть представлены в виде списка параметров, концептуальных графов, семантической сети, древовидных структур, предикатов, фреймов, рисунков и мультимедийной информации.

В большинстве случаев для представления прецедентов достаточно простого параметрического представления, т.е. представления прецедента в виде набора параметров с конкретными значениями и решения (диагноз по проблемной ситуации и рекомендации ЛПР) [Варшавский, 2009]. Но наиболее перспективной возможностью является представление прецедентов с использованием онтологического подхода [Гаврилова и др., 2000]. Выбор онтологии для представления прецедентов обусловлен рядом важных достоинств, отличающих их от других моделей представления знаний. Использование онтологического подхода позволяет задать сложную структуру прецедента, включающую данные разных типов, естественность представления структурированных знаний и достаточно простое их обновление в относительно однородной среде.

Последнее свойство особенно важно для ИСППР, ориентированных на открытые и динамические предметные области.

Таким образом, онтология представляет собой базу знаний (БЗ) по предметной области, которая используется для поддержки СВР-цикла, а также задает структуру прецедента и обеспечивает его хранение.

Знания о предметной области и модель представления прецедентов описываются в виде иерархии концептов онтологии, а каждый прецедент из БП – в виде иерархии экземпляров концептов, связанных отношениями языка описания онтологий для семантической паутины (OWL) [Варшавский и др., 2012].

2. Теория структурного отображения

Теория структурного отображения (SMT – structure mapping theory) позволяет формализовать некоторый набор неявных ограничений, которыми пользуется человек, оперируя такими понятиями, как сходство, аналогия и подобие [Falkenhainer et al., 1989]. Согласно SMT предполагается, что аналогия является отображением знаний одной области (базы) в другую область (цель), базирующимся на системе отношений, которые имеются между объектами базовой области и объектами целевой области, а также, что человек (ЛПР) предпочитает оперировать некоторой целостной системой взаимосвязанных глубинных отношений, а не простым набором поверхностных и слабо связанных фактов.

Процесс поиска решения на основе аналогий согласно SMT включает следующие основные этапы:

1. Определение потенциальных аналогов. Имея целевую ситуацию (цель), определить другую ситуацию (базу), которая является аналогичной или подобной целевой ситуации;

2. Отображение и вывод. Построить отображение, состоящее из соответствий между базой и целью. Это отображение может включать дополнительные знания (факты) о базе, которые могут быть перенесены в цель.

3. Оценка «качества» соответствия. Оценить полученное соответствие, используя такие структурные критерии, как число подобий и различий, степень структурного соответствия, количество и тип новых знаний, полученных по аналогии из кандидатов заключений.

Рассмотрим механизм структурного отображения (SME – structure mapping engine), базирующийся на SMT. Этот механизм пред-

назначен для моделирования поиска решений на основе аналогий и прецедентов и позволяет сформировать наиболее общие соответствия (Gmaps) для структурированных представлений базовой и целевой областей, а также обеспечивает оценку полученных соответствий.

Входными данными для алгоритма SME являются структурные представления базовой и целевой областей.

Алгоритм SME реализует следующие четыре этапа:

1. Построение локальных соответствий (гипотез соответствия МНs). Определить соответствия между элементами в базовой и целевой областях с помощью правил:

- если два отношения имеют одинаковое имя, тогда создается гипотеза соответствия;
- для каждой гипотезы соответствия между отношениями проверяются их соответствующие аргументы: если их количество и тип совпадают, тогда создается гипотеза соответствия между ними.

Далее определить оценку правдоподобия локальных соответствий, используя следующие правила и коэффициенты, задаваемые экспертом (CF1 – величина, на которую необходимо увеличить оценку правдоподобия МН, если имена базового и целевого элементов совпадают;

CF2 – величина, на которую необходимо увеличить оценку правдоподобия МН, если хотя бы базовый элемент имеет родительское отношение – отношение более высокого уровня):

- увеличить оценку правдоподобия для соответствия на CF1, если базовый и целевой элементы имеют одинаковые имена;
- увеличить оценку правдоподобия для соответствия на CF2, если хотя бы у базового элемента имеется родительское отношение.

2. Построение глобальных соответствий (Gmaps). Сформировать системы соответствий, которые используют совместимые пары объектов (1:1), называемые Emaps.

3. Построение кандидатов заключения (Inferences). С каждым Gmap связать множество (возможно пустое) кандидатов заключения Inferences – факты, которые присутствуют в базовой области, но не присутствуют изначально в соответствующей целевой;

4. Оценка глобальных соответствий Gmaps (SES). Получить оценку для Gmaps, которая зависит от оценок правдоподобия локальных соответствий МНs.

Таким образом, в результате выполнения алгоритма формиру-

ются наиболее общие соответствия G_{map} , включающие следующие составляющие:

- соответствия – множество парных соответствий между базовой и целевой областями;
- кандидаты заключения – множество новых фактов, которые предположительно могут содержаться в целевой области;
- оценка глобальных соответствий – числовая оценка качества G_{map} .

3. Метод извлечения прецедентов на основе теории структурного отображения

Существует целый ряд методов поиска решений на основе прецедентов (извлечения прецедентов) и их модификаций [Варшавский и др., 2009], например: метод ближайшего соседа, метод извлечения прецедентов на основе деревьев решений, метод извлечения прецедентов на основе знаний, метод извлечения с учетом применимости прецедента.

Предлагается осуществлять извлечение и определение сходства прецедента C и текущей ситуации Q в два этапа:

- определение сходства прецедента с текущей ситуацией на основе онтологии предметной области и формирование парных соответствий с помощью алгоритма на базе SMT;
- определение сходства прецедента и текущей ситуации по методу ближайшего соседа [Варшавский П.П. и др., 2013] с учетом полученных парных соответствий.

В результате применения данного метода получим множество прецедентов, каждому из которых сопоставлены две оценки сходства с текущей ситуацией:

– оценка на основе онтологии предметной области: $S_{struct} = \sum_{i=1}^k LS_i / SES_{MAX}$, где k – количество соответствий, LS_i – оценка правдоподобия для i соответствия, SES_{MAX} – оценка для случая, когда каждый элемент в базовой области имеет родительское отношение и в качестве базовой области выбирается целевая;

– оценка по методу ближайшего соседа: $Sim(C, Q) = 1 - d_{CQ} / d_{MAX}$, где d_{CQ} – расстояние между текущей ситуацией и прецедентом, d_{MAX} – максимальное расстояние в выбранной метрике.

Исходя из этих данных, ЛПП может выбрать наиболее подходящий прецедент в текущей ситуации и получить решение прецедента в качестве рекомендации к текущей ситуации.

4. Реализация CBR-модуля для ИСППР

Предлагаемый подход реализован в прототипе CBR-модуля. На рис. 1 представлена архитектура разработанного прототипа CBR-модуля.



Рис. 1. Архитектура прототипа CBR-модуля

Работа реализованного прототипа CBR-модуля рассмотрена на примере решения задач диагностики состояний сложного объекта и обнаружения управляющих воздействий на примере подсистемы САОЗ ВВЭР АЭС [Башлыков и др., 2013]. На основе технологического регламента и оперативных инструкций были сформированы десять начальных прецедентов. Рассмотрим один из них более подробно (рис. 2). В данном прецеденте представлена следующая информация: текущее значение давления в емкости САОЗ 1 (Т517В01) соответствует 63, температура корпуса САОЗ 1 (ТН11Т500) равна 20 и необходимо выдать рекомендацию подпитать первый насос аварийного расхолаживания (ТН11D01) борным концентратом 40 г/кг по причине отключения САОЗ 1 из-за закрытия задвижек ТН11S24 и ТН11S25. В данном примере текущая ситуация имеет схожую структуру с рассмотренным прецедентом, но в ней отсутствует решение в виде рекомендации.

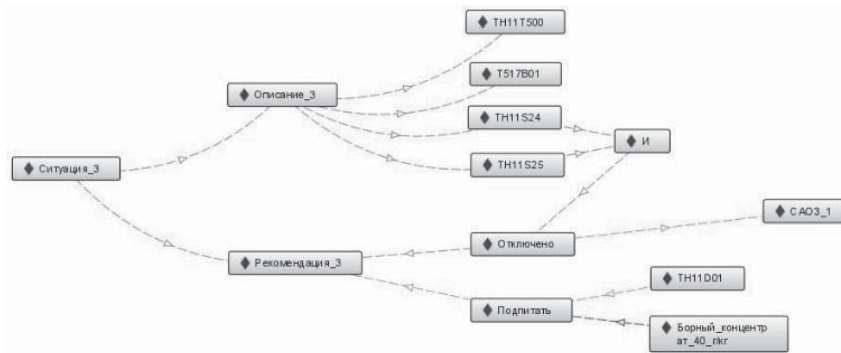


Рис. 2. Описание прецедента – общий вид

Результаты работы CBR-модуля представлены на рис. 3. ЛПР имеет возможность выбрать наиболее подходящий прецедент исходя из двух оценок сходства:

- структурная оценка на основе теории структурного отображения;
- оценка по методу ближайшего соседа.

Прецедент	Структурная оценка	Оценка по МБС	
Прецедент_3	97%	80%	Выбрать...
Прецедент_1	97%	66.67%	Выбрать...
Прецедент_2	96%	80%	Выбрать...
Прецедент_4	93%	66.67%	Выбрать...
Прецедент_5	93%	66.67%	Выбрать...

Рис. 3. Результат работы прототипа CBR-модуля

Заключение

В работе рассмотрены общие понятия и особенности прецедентного подхода. Для организации представления прецедентов рассмотрена возможность использования онтологического подхода. Предложен метод поиска решения на основе прецедентов с использованием структурной аналогии на базе SMT. Разработана

архитектура и выполнена программная реализация СВР-модуля в среде MS Visual Studio 2010 на языке C#. Реализованный СВР-модуль применен в составе прототипа ИСППР для решения задач диагностики состояний сложного объекта и обнаружения управляющих воздействий на примере подсистемы САОЗ ВВЭР АЭС.

СПИСОК ЛИТЕРАТУРЫ

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах // Под ред. В.Н. Вагина, Д.А. Поспелова. – М.: Физматлит, 2004. – 704 с.

[Еремеев и др., 2012] Еремеев А.П., Варшавский П.Р., Куриленко И.Е. Моделирование временных зависимостей в интеллектуальных системах поддержки принятия решений на основе прецедентов // International Journal «Information technologies and knowledge», Vol. 6, № 3, 2012, с. 227–239.

[Гаврилова и др., 2000] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.

[Варшавский и др., 2012] Варшавский П.Р., Алехин Р.В., Зо Лин Кхаинг Применение онтологического подхода для реализации поиска решения на основе прецедентов в интеллектуальных системах поддержки принятия решений // Труды 13-й национальной конференции по ИИ с международным участием (КИИ-2012). Т. 3. – Белгород: Издательство БГТУ, 2012. – С. 72–79.

[Falkenhainer et al., 1989] Falkenhainer B., Forbus K., Gentner D. The Structure-Mapping Engine: Algorithm and examples // Artificial Intelligence, 41, 1989, – P. 1–63.

[Варшавский и др., 2009] Варшавский П.Р., Еремеев А.П. Моделирование рассуждений на основе прецедентов в интеллектуальных системах поддержки принятия решений // Искусственный интеллект и принятие решений. 2009. №2. – С. 45–57.

[Башлыков и др., 2013] Башлыков А.А., Еремеев А.П. Реализация системы интеллектуальной поддержки принятия решений для операторов кризисного центра концерна РОСЭНЕРГОАТОМ. Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII Международной научно-технической конференции (Коломна, 20–22 мая 2013 г.). В 3-х томах. Т. 3. – М.: Физматлит, 2013, с. 1056–1066.

[Варшавский П.Р. и др., 2013] Варшавский П.Р., Алехин Р.В. Метод поиска решений в интеллектуальных системах поддержки принятия решений на основе прецедентов // International Journal «Information models and analyses», Vol. 2, Number 4, 2013 – ITHEA – Bulgaria, Sofia – 2013. – ISSN 1314-6416. pp. 385–392.

УДК 004.415

СИТУАЦИОННЫЙ ЧЕЛОВЕКО-МАШИННЫЙ ИНТЕРФЕЙС ДЛЯ ИНФОРМАЦИОННЫХ СИСТЕМ ИНТЕЛЛЕКТУАЛЬНОЙ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

А.А. Башлыков (*aleks-ltd@mail.ru*)

Национальный исследовательский университет «МЭИ»,
ООО «ТАСМО-БИТ», Москва

В статье приводятся результаты анализа операторских интерфейсов для интеллектуальной поддержки принятия управляющих решений в человеко-машинных системах управления сложными объектами. Вводится понятие ситуационного человеко-машинного интерфейса, основанного на методах когнитивной графики, и описываются принципы его построения. Приводится пример организации такого ситуационного интерфейса для диспетчеров трубопроводных систем.

Введение

Анализ аварий, произошедших на сложных экологически опасных объектах и технологиях, показывает, что в большом числе случаев основным виновником такого положения дел является человек-оператор ([Котик и др., 1985], [Чачко, 1986], [Дьяков, 1991]). Немаловажную роль при этом играет несовершенство человеко-машинного интерфейса в части оценки складывающейся ситуации и интеллектуальной поддержки принятия управляющих решений. В работах [Rasmussen, 1980], [Vicente et al., 1992], [Burns et al., 2004], [Башлыков, 2008] рассматриваются существующие и перспективные типы человеко-машинного операторского интерфейса, включая: *приборный операторский интерфейс*, где информация представляется на экранах аналоговых электромеханических и цифровых электронных приборов; *технологический операторский интерфейс*, где информация представляется наглядно в виде множества электронно-управляемых технологических схем, представленных на щитах управления, которые интегрируются в базовый образ объекта управления (ОУ); *экологический операторский*

интерфейс – это способ представления, где информация отображается в виде видеок кадров. В качестве элементов в данном интерфейсе могут использоваться образы, отражающие физические закономерности, присущие ОУ. Примером таких образов могут служить термодинамические циклы, гидравлические закономерности при пуске нефтепровода; *функциональный операторский интерфейс*, где представление информации основано на функционально-ориентированной философии управления. Например, на философии обеспечения наивысшей безопасности ОУ; *иммерсивный операторский интерфейс*, где представление информации основано на использовании метода виртуальной реальности; *когнитивный операторский интерфейс*, где представление информации осуществляется в виде динамических образов состояния технологических систем ОУ или всего ОУ, описанных методами когнитивной графики [Башлыков, 2012].

Рассмотренные выше множества подходов к построению операторских интерфейсов при всей их прогрессивности и реализуемости не позволяют решить главную задачу человека-оператора как лица, принимающего решения (ЛПР) – снижение объема перерабатываемой информации.

На взгляд автора, и основываясь на гипотезах Д.А. Поспелова, [9] основным подходом, позволяющим решить главную задачу ЛПР, является построение *ситуационных операторских интерфейсов* (СиОИ). Суть СиОИ заключается в том, что при организации такого интерфейса управление реализуется ЛПР путем обработки только той информации, которая относится к текущей ситуации, требующей принятия управляющих решений.

1. Принципы организации процессов человеко-машинного управления по ситуациям

Будем считать что: управление – это, прежде всего, искусство ЛПР правильно определить и оценить ситуацию на ОУ и выбрать наиболее эффективные методы управления, наилучшим образом отвечающие возникшей ситуации. Различные возникающие проблемные ситуации требуют различных методов их разрешения; для каждого из методов разрешения проблемных ситуаций существуют свои ограничения.

Ситуационный тип процесса управления представлен на рисунке 1.

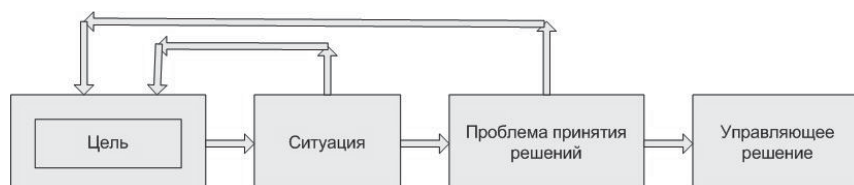


Рис.1. Ситуационный тип процесса управления

В настоящее время во всех человеко-машинных системах управления для построения пользовательского интерфейса используется технологический принцип представления информации. Такое решение увеличивает информационную нагрузку на ЛПР, т.к. решение всех задач управления в реальном времени возлагается на человека (рис. 2).

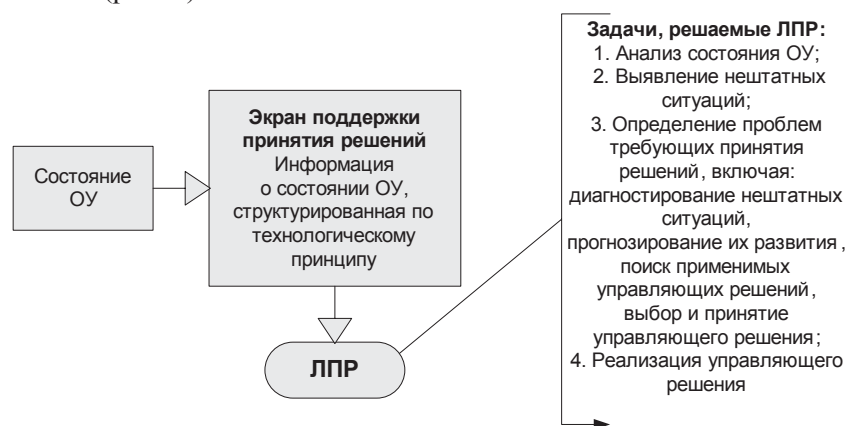


Рис.2. Задачи, решаемые ЛПР при структуризации информации для построения пользовательского интерфейса по технологическому принципу

Структуризация информации для управления ЛПР по ситуационному принципу позволяет резко снизить информационную нагрузку на человека за счет снижения числа решаемых задач при управлении (рис. 3).

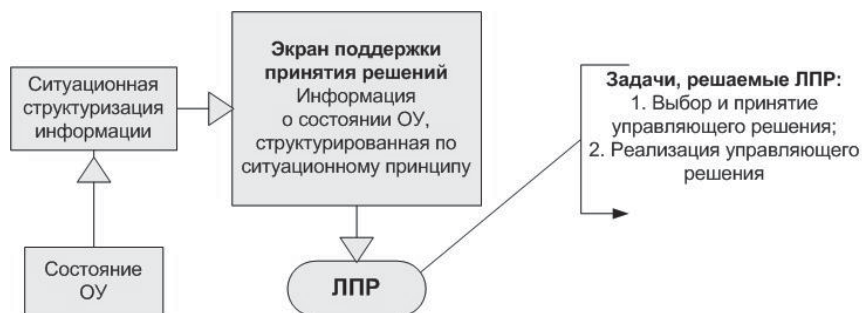


Рис.3. Задачи, решаемые ЛПР при структуризации информации для построения пользовательского интерфейса по ситуационному принципу

Ниже сформулируем основные принципы построения операторского интерфейса путем структуризации и представления информации ЛПР для управления по ситуационному принципу.

Принцип 1. Управление по ситуациям будем рассматривать как новый подход к построению операторских интерфейсов, который позволяет представлять ЛПР из всего множества данных о состоянии ОУ информацию, относящуюся только к конкретным нештатным ситуациям и характеризующим их условиям, и принимать решения, относящиеся только к этим конкретным ситуациям и характеризующим их условиям [Башлыков, 2014а]. При этом из всей имеющейся информации о состоянии ОУ для ЛПР является только информация о конкретной ситуации принятия решений (СПРР), под которой понимается конкретный набор обстоятельств, максимально влияющих на ОУ в данное время.

Управление по ситуации не всегда является простым набором навыков, предписаний и руководств. Это, скорее, способ мышления о возникших проблемах и способах их решения в рамках ограниченного, но достаточного для принятия решений объема информации. Причем особый акцент ставится на то, что хотя общий процесс управления всегда одинаков, специфические приемы, которые должен использовать ЛПР для эффективного достижения целей управления, могут значительно варьироваться в зависимости от реальной ситуации.

Принцип 2. Будем считать, что различные типы ситуаций требуют выбора различных принципов выработки управляющих воздей-

ствий на ОУ. В соответствии с выводами, изложенными в работах [Башлыков, 2014а], [Башлыков, 2014b], будем применять следующие принципы выработки управляющих воздействий (управления): *основанные на навыках; основанные на правилах; основанные на знаниях.*

Принцип 3. Условиями выбора информации для представления ЛПР СПрР будем считать: информация о СПрР должна содержать минимальное и конечное количество факторов и описывать их состояние и взаимосвязи; информация о СПрР должна содержать только те факторы, которые существенным образом влияют на состояние ОУ; информация о СПрР должна включать в себя только те факторы, которые влияют на управление в настоящий момент времени (а не в прошлом или будущем); при описании информации о СПрР необходимо учитывать место и время возникновения отклонения от режима функционирования ОУ, степень отклонения, причины и последствия возникновения отклонения.

Принцип 4. Будем считать, что логика обработки ситуационной информации ЛПР или компьютерными средствами интеллектуальной поддержки принятия решений (СИППР) включает последовательность задач, представленных на рис. 2.

Принцип 5. Для обеспечения требуемого качества управления используемый информационный процесс представления информации должен обладать следующими характеристиками: полнотой и качеством информации; ценностью и полезностью информации; достоверностью и точностью информации; актуальностью информации, зависящей от времени ее получения; оперативностью и доступностью информации; частотой использования определенных видов информации.

Принцип 6. Информация для управления по ситуациям и ее представление формируются СИППР и ЛПР параллельно [Башлыков, 2014а], [Башлыков, 2014b].

2. Информационный базис для распознавания, оценки и диагностики нештатных ситуаций при интеллектуальной поддержке принятия управляющих решений

Выбор информации для распознавания и оценки наличия СПрР включает в себя: поиск характеристик сложившегося состояния ОУ, которое, с точки зрения ЛПР и СИППР, может быть удов-

летворительным или неудовлетворительным. В последнем случае ситуация становится проблемной, она характеризует реальные несоответствия желаемого и действительного уровней удовлетворения процессом управления со стороны ЛПР; при выявлении признаков возникающих ситуаций главным элементом должна стать целевая направленность в описании деятельности ОУ.

Все ситуации, требующие принятия решений, связаны, в первую очередь, либо с реализацией ранее установленной цели, либо с формированием новых целей; после выявления признаков СПрР следующим шагом является ее корректное определение и описание. В этой связи необходимо найти приемлемую для этой цели классификацию информационных потоков.

Учитывая иерархическую природу возникновения и развития СПрР и в целях адекватного отображения функционирования иерархического ОУ, необходимо формирование набора иерархически связанных определений и описаний СПрР. Для этого можно использовать один из трех видов стратегии: стратегии «Снизу вверх» – анализ процессов возникновения ситуаций, начиная с нижнего уровня иерархии ОУ; стратегии «Сверху вниз» – анализ процессов возникновения ситуаций с верхнего уровня иерархии, а затем их разукрупнение и детализации; комбинированная стратегия – анализ процессов возникновения ситуаций как сверху, так и снизу. Для выбора адекватной стратегии анализа процессов возникновения СПрР необходимо четко представлять, к какому типу они относятся. Для этого необходимо провести классификацию ситуаций; чтобы определить СПрР, необходимо «увидеть» возникшее противоречие в процессе управления.

Определение – это сложная процедура, состоящая из нескольких фраз, основными являются: выявление проблемы, т. е. определение ее симптомов; диагноз проблемы. Решение выявленной проблемы предполагает установление причинно-следственной цепи, иерархию причин и следствий, которая ведет «назад» от следствия к причине до той точки, в которой можно предпринять действие, помогающее устранить исходные причины.

Важным этапом исследования проблемы является ее распознавание. Для изучения проблемы, выяснения ее особенностей необходимо реализовать процедуры в следующей последовательности: разделение проблемы на части; выделение основных и вто-

ростепенных характеристик проблемы; установление причинно-следственных характеристик проблемы; прогнозирование и анализ требуемых действий; разработка рекомендаций к действиям. Следующей характеристикой методологии исследования проблемы является ее постановка. Будем различать несколько уровней постановки проблемы: низший – характеризуется интуитивным восприятием ЛППР о противоречиях в системе управления; средний – описание в соответствии с принятыми правилами; высший – заключается в ее отработке в соответствии с целью и стратегией ОУ, поиске новых форм.

3. Пример ситуационного интерфейса

В качестве примера применения ситуационного подхода к управлению приведем интерфейс СИППР диспетчерского контроля и управления трубопроводными системами, представленный на рис. 4.

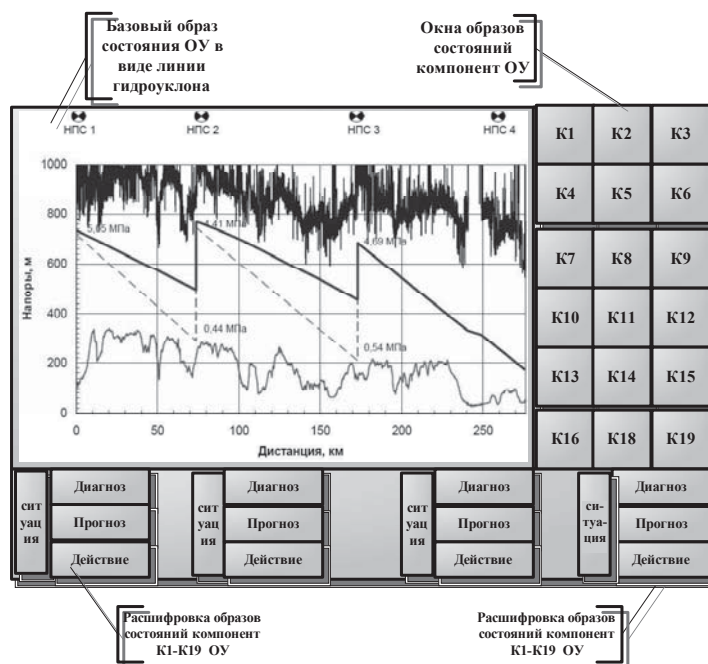


Рис.4. Пример ситуационного интерфейса для диспетчера системы транспорта нефти

Заключение

Разработанный подход и его реализация позволили перейти к технологии управления по ситуациям.

СПИСОК ЛИТЕРАТУРЫ

[**Башлыков, 2008**] Башлыков А.А. Когнитивная графика как средство образного представления информации в человеко-машинных системах управления сложными объектами // Автоматизация, телемеханизация и связь в нефтяной промышленности. 2008. № 12.

[**Башлыков, 2012**] Башлыков А.А. Образное представление состояния сложных технологических объектов управления методами когнитивной графики // Искусственный интеллект и принятие решений. 2012. № 3. 9–18 с.

[**Башлыков, 2014а**] Башлыков А.А. Человек в системе оперативно-диспетчерского управления и проблемы автоматизации процессов для интеллектуальной поддержки принятия решений. – М: Автоматизация, телемеханизация и связь в нефтяной промышленности. ВНИОЭНГ. № 1/2014, стр. 11–19.

[**Башлыков, 2014б**] Башлыков А.А. Принципы формирования требований к человеко-машинному интерфейсу для систем диспетчерского контроля и управления трубопроводными системами. – М: Автоматизация, телемеханизация и связь в нефтяной промышленности. ВНИОЭНГ. № 3/2014, стр. 11–19.

[**Дьяков, 1991**] Дьяков А.Ф. Надежная работа персонала в энергетике. М.: Изд. МЭИ, 1991. – 224 с.

[**Котик и др., 1985**] Котик М.А., Емельянов А.М. Ошибки управления. Психологические причины, метод автоматизированного анализа. Таллин: Валгус, 1985. – 391 с.

[**Поспелов, 1986**] Поспелов Д.А. Ситуационное управление. Теория и практика. – М.: Наука, 1986.

[**Чачко, 1986**] Чачко А.Г. Подготовка операторов энергоблоков: Алгоритмический подход. М.: Энергоатомиздат, 1986. – 232 с.

[**Burns et al., 2004**] Burns С.М., Haidukiewicz J.R. Ecological interface design. – Boca Raton: CRC Press, 2004.

[**Rasmussen, 1980**] Rasmussen J. The human as a system component. – New York: Academic Press, 1980.

[**Vicente et al., 1992**] Vicente K.J., Rasmussen J. Ecological interface design: theoretical foundations // IEEE Transactions on systems, man, and cybernetics. – 1992. – Vol. 22, №4. P. 589–606.

УДК 007:519.816

ВЕРИФИКАЦИЯ МОДЕЛЕЙ ПРОЦЕССОВ НА ОСНОВЕ ТЕМПОРАЛЬНЫХ СЕТЕЙ ПЕТРИ¹

А.П. Еремеев (*eremeev@appmat.ru*)

Ю.И. Королев (*korolevyu@gmail.com*)

Национальный исследовательский университет «Московский энергетический институт», Москва

Рассматриваются возможности верификации темпорального подкласса сетей Петри, предлагаемого для использования в динамических интеллектуальных системах типа интеллектуальных систем поддержки принятия решений реального времени (ИСППР РВ) для моделирования процессов в сложных объектах. Описываются графовые инструменты анализа, отражающие смену состояний в таких сетях, и возможность применения для их верификации метода model checking (модельного контроля).

Введение

Обеспечение корректной работы программных и аппаратных компонентов систем управления приобретает сегодня первостепенное значение. Надежность и предсказуемость поведения таких систем, в том числе в нештатных ситуациях, зачастую являются более важными свойствами, чем производительность, модифицируемость и т.п. Большинство имеющихся объектов управления обладают рядом нетипичных для традиционного управления свойств, таких как уникальность, отсутствие формализуемой цели существования и количественного критерия оптимальности, высокая динамичность, неполнота описания объекта, наконец, субъективность поведения лица, принимающего решения (ЛПР) при управлении данным объектом. Поэтому в последние годы технология решения подобных задач значительно изменилась. Сегодня назначение ИСППР РВ – это не только своевременное представление необходимой информации ЛПР для принятия им эффективных решений по управлению, но и предварительный анализ этой инфор-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-01-00427) и Фонда содействия инновациям.

мации. Такой подход позволяет переложить значительную часть операций (прогнозирование развития ситуации, выбор наиболее перспективных решений из множества альтернатив, диагностика, мониторинг и др.) на ЭВМ. Для решения подобных задач активно используются интеллектуальные технологии и методы искусственного интеллекта. Сложность интеллектуальных систем типа ИСППР РВ зачастую существенно превышает сложность классических систем управления. На кафедре прикладной математики Национального исследовательского университета «Московский энергетический институт» (НИУ «МЭИ») уже более тридцати лет проводятся исследования по разработке математического и программного обеспечения интеллектуальных систем поддержки принятия решений (ИСППР) [Вагин, 1988], в том числе ИСППР РВ [Вагин и др., 1988]. Одним из основных элементов ИСППР РВ является блок моделирования. В качестве эффективного средства моделирования и анализа процессов для ИСППР РВ предлагается использовать аппарат раскрашенных (цветных) сетей Петри реального времени с поддержкой темпоральной логики Аллена (РСП РВ ТЛА). Данный формализм позволяет адекватно моделировать как количественные, так и качественные темпоральные (временные) зависимости [Еремеев и др., 2013], что крайне важно при работе с динамическими объектами и системами. Отметим, что использование сетей Петри подразумевает достаточно высокий уровень параллелизма, что, как и учет темпоральных зависимостей, требует наличия средств анализа и верификации моделей, созданных на его основе данного аппарата.

1. Средства анализа моделей

Параллельные, распределенные и многопоточные программы, характерные для многих систем управления, в том числе, реального времени типа ИСППР РВ, крайне подвержены ошибкам. Хорошо известно, что даже в тех случаях, когда функционирование каждой из параллельных взаимодействующих компонент системы абсолютно ясно, человеку трудно понять работу всей системы в целом. Такие системы, которые работают правильно «почти всегда», годами могут сохранять «тонкие» ошибки, проявляющиеся в исключительных ситуациях. Наиболее очевидным и широко распространенным методом проверки правильности программных систем

является тестирование – проверка работы построенной системы в различных ситуациях, при различных исходных данных. Однако в случае с параллельными системами обычно невозможно заранее определить все возможные траектории функционирования. Поэтому в качестве основного метода повышения качества разработки применяется верификация – формальная проверка того, что система (модель) удовлетворяет сформулированным заранее требованиям [Карпов, 2010]. Методы верификации различаются в зависимости от того, какой аппарат лежит в основе проверяемой системы. Рассматриваемый подкласс РСП РВ ТЛА представляет собой визуальный язык программирования [Еремеев и др., 2013] с формально определенным синтаксисом. Модели, разработанные с помощью этого аппарата, кажутся полностью формализованными. Однако с точки зрения семантики это не так: из самой модели не следует непосредственно полное формальное описание ее поведения. Параллелизм, присущий сетям Петри в целом, и учет темпоральных зависимостей, введенный для упрощения разработки, позволяют «элегантно» моделировать процессы, но зачастую делают целостное восприятие процесса более сложным. Поэтому для анализа поведения и верификации моделей, построенных с помощью РСП РВ ТЛА, необходимо использовать дополнительные инструменты. Известны три основные группы методов анализа сетей Петри [Мурата, 1989]: основанные на построении *графов изменения состояний*; *матричные методы*, использующие уравнения сети и инварианты; *методы редукции*. При работе с раскрашенными сетями Петри последние две группы методов используются редко из-за высокой сложности формальных определений подобных подклассов. Чаще в качестве основного инструмента анализа рассматриваются *графы достижимости и покрытия* [Szuruga, 2008]. Для работы с инструментальными средствами анализа РСП РВ ТЛА определим формально ключевые понятия состояния и перехода между состояниями. В качестве иллюстративного примера рассмотрим модель системы управления экстренным торможением поезда [Еремеев и др., 2013] (рисунок 1). Состояние сети представляет собой пару (M, S) , где M – маркировка, – функция на множестве мест P , а S – темпоральный вектор, ставящий в соответствие каждому месту сети число – темпоральную метку. Для сети, изображенной на рисунке 1, множество мест зададим

следующим образом: $P = \{Timer1, LightSig, SoundSig, Brake, Timer2, Driver\}$. Тогда начальное состояние сети (M_0, S_0) будет выглядеть так:

$$M_0 = (on, off, off, off, on, active),$$

$$S_0 = (0, 0, 0, 0, 0, 0).$$

Переход от одного состояния сети к другому может быть обусловлен двумя причинами:

- срабатыванием перехода $t \in T$ с подстановкой b (подстановка – функция, которая замещает каждую переменную в защитной функции $G(t)$ и функциях весовых и временных значений дуг E_M, E_S , влияющих на переход t , значением соответствующего типа);
- течением времени – постепенное уменьшение каждой временной метки на фиксированную величину, пока не появится переход, который может сработать.

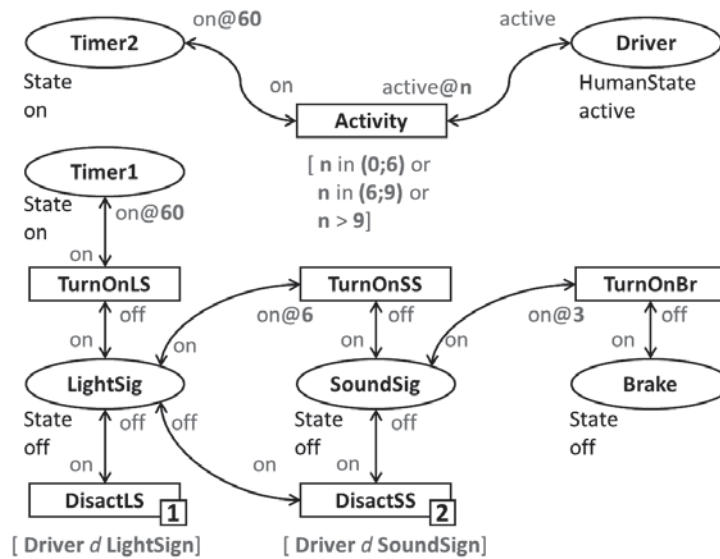


Рис. 1. Пример РСП РВ с поддержкой логики Аллена

Следует отметить, что безусловный приоритет при смене состояний сети имеет событие срабатывания перехода. Течение времени позволяет только дожидаться момента, когда может сработать очередной переход. Будем считать, что состояние (M', S') достижимо

из состояния (M,S) , если существует конечная последовательность переходов, начинающаяся с состояния (M,S) и оканчивающаяся состоянием (M',S') . Обозначим за $R(M,S)$ множество всех состояний, достижимых из состояния (M,S) . Анализ РСП РВ ТЛА можно проводить, используя *граф достижимости* (ГД), вершины которого помечены элементами множества $R(M_0,S_0)$, а каждая дуга отображает изменение состояния (M_i,S_i) на (M_j,S_j) по истечении времени $\tau \geq 0$ и срабатывания перехода t в подстановке b . Фрагмент графа достижимости, построенного по этим правилам, представлен на рисунке 2. Подробное описание построения приведено в [Еремеев и др., 2014].

Анализ свойств сети может осуществляться с помощью маркировки вершин ГД и меток дуг. Каждая метка дуги представляет собой тройку, состоящую из перехода, его подстановки и значения временного промежутка перед его срабатыванием. Последний параметр, таким образом, позволяет определить время, затраченное на переход от одного состояния к другому. Используя стандартные алгоритмы поиска кратчайшего или длиннейшего пути между двумя узлами графа, можно найти минимальное и максимальное время перехода из одного состояния в другое.



Рис. 2. Фрагмент графа достижимости

Если продолжить строить ГД для анализируемой сети, можно убедиться, что из-за непрерывного уменьшения значений темпоральных меток ГД оказывается бесконечным. Подобная ситуация возникает при анализе практически любой РСП РВ ТЛА. Очевидно, что в этом случае данный формализм неудобен для анализа сетей. Введение дополнительных условий на темпоральные метки позволяет трансформировать бесконечный граф в конечную структуру.

Одним из главных преимуществ РСП РВ ТЛА является возможность представления множества достижимых состояний с помощью конечного *графа покрытия* (ГП). Бинарное отношение покрытия позволяет определять эквивалентные по своим характеристикам

состояния сети Петри. Будем считать, что два состояния покрывают друг друга, если их маркировки совпадают, а временные метки либо совпадают, либо не превышают максимального возраста доступа места $p \in P$, то есть такого значения темпоральной метки, когда фишки-токены становятся доступными для всех выходных переходов места p . При этом удовлетворяются условия рефлексивности, симметричности и транзитивности, следовательно, отношение покрытия есть отношение эквивалентности (\sim) на множестве $R(M_0, S_0)$.

Графы ГД и ГП строятся одинаково. Различие заключается только в способе добавления новой вершины в графы. Для ГП после определения нового состояния сети необходимо проверить, есть ли в графе вершина, которая отображает состояние, покрывающее новое. Если есть, то необходимо добавить только новую дугу, которая идет к найденной вершине. В противном случае вершина нового состояния добавляется в ГП вместе с соответствующей дугой. Таким образом, каждая вершина ГП помечена элементом фактор-множества $R(M_0, S_0)/\sim$, причем не существует двух или более вершин, помеченных одним и тем же элементом, а количество вершин ГП совпадает с числом элементов $R(M_0, S_0)/\sim$. Конечность множества $R(M_0, S_0)/\sim$ обеспечивается двумя факторами:

- множество всех возможных маркировок M на конечном множестве мест сети P конечно;
- максимальный возраст доступа каждого места $p \in P$ задается в общем случае вещественным числом.

Таким образом, ГП для РСР РВ ТЛА всегда конечен. Пример ГП, построенного для сети на рисунке 1, приведен в [Еремеев и др., 2014].

ГП для РСР РВ ТЛА предоставляет такие же возможности анализа сетевых свойств, как и полный ГД, так как содержит все достижимые маркировки и позволяет увидеть все состояния с точностью до темпоральных меток.

2. Верификация моделей по методу **model checking**

Для верификации технических систем свойства их поведения должны быть выражены формально логическими утверждениями, которые обеспечат их простую, лаконичную и недвусмысленную запись. Для спецификации таких свойств необходимы логические

утверждения, истинность которых зависит от времени, например «Посланный запрос когда-нибудь позже будет обработан». Обычная логика высказываний является плохо пригодной для формулировки утверждений о поведении технических систем при изменении их состояний во времени. Формализация даже простейшего примера «Любой посланный запрос когда-нибудь позже будет обслужен» с помощью, например, логики предикатов первого порядка приводит к громоздкому утверждению [Карпов, 2010]:

$$(\forall t \geq 0)(\text{Послан}(\text{Запрос}, t) \rightarrow (\exists t' > t)(\text{Обслужен}(\text{Запрос}, t'))))$$

Поэтому при верификации темпоральных конструкций используются выражения темпоральных логик. Если нет необходимости подробно описывать закономерности поведения системы и взаимодействие ее объектов, целесообразнее применять не сложные интервальные логики, а простые расширения обычной логики высказывания. Традиционно при верификации используются темпоральные логики *линейного времени* (Linear Temporal Logic) LTL [Pnueli, 1977] и *ветвящегося времени* (Computational Tree Logic) CTL [Clarke et al., 1986].

Перспективным методом верификации РСП РВ ТЛА является model checking (проверка модели) МС [Clarke et al., 1981]. Другие методы – дедуктивная верификация [Floyd, 1967] и проверка эквивалентности [Milner, 1980] – в общем случае не могут быть полностью автоматизированы, что негативно сказывается на возможности их применения при создании систем управления и ИСППР РВ. С другой стороны, исследования в области МС привели в последнее время к разработке очень эффективных алгоритмов верификации, позволяющих проверять реальные, разрабатываемые промышленностью программно-аппаратные системы. В частности, в работе [Карпов, 2010] приводятся эффективные алгоритмы МС, позволяющие проверить, что формула темпоральной логики LTL или CTL, выражающая некоторое свойство поведения динамической системы во времени, выполняется (является истинной) на модели системы с конечным числом состояний. В качестве модели при этом используется структура Крипке [Kripke, 1963], которая формально задается как пятерка $K = (C, C_0, H, AP, L)$, где:

- C – конечное непустое множество состояний;
- $C_0 \subseteq C$ – непустое множество начальных состояний;

- $H \subseteq C \times C$ – множество переходов, удовлетворяющее требованию: $(\forall c \in C)(\exists c' \in C)((c, c') \in H)$;

- AP – конечное множество атомарных предикатов;

- $L: C \rightarrow 2^{AP}$ – функция пометок, сопоставляющая каждому состоянию множество истинных в нем атомарных предикатов.

Нетрудно убедиться, что ГП для РСП РВ ТЛА уже является такой структурой:

- C соответствует множество вершин ГП;

- S_0 соответствует вершина начального состояния (M_0, S_0) ;

- H соответствует множество переходов между вершинами ГП, требование существования перехода из любой вершины обеспечивается правилом построения ГП;

- AP соответствует конечное фактор-множество $R(M_0, S_0)/\sim$, содержащее по одному элементу для каждого класса эквивалентности отношения покрытия состояний РСП РВ ТЛА;

- L соответствует функция пометок вершин ГП элементами $R(M_0, S_0)/\sim$.

Таким образом, верификация РСП РВ ТЛА с помощью метода МС является естественным расширением начального анализа сетей с помощью графов состояний.

Заключение

Общепризнано, что как тестирование, так и верификация по отдельности не могут гарантировать достаточного уровня правильности разрабатываемых систем. Существует большое число примеров, когда в тщательно проверенных и оттестированных реализациях с помощью верификации впоследствии обнаруживались тонкие ошибки. С другой стороны, нельзя надеяться только на верификацию. Часто причина ошибок кроется в том, что и при разработке, и при доказательстве алгоритмов неявно выдвигаются неправильные предположения о характере работы, т.е. используются неадекватные формальные модели. Таким образом, как тестирование, так и верификация обладают своими преимуществами и недостатками, поэтому эти подходы можно считать взаимодополняющими. Для повышения надежности реализаций при разработке систем управления, программных и аппаратных систем должны применяться оба подхода. Для проведения подобных исследований применительно к ИСППР РВ, в том числе с исполь-

зованием РСР РВ ТЛА, разрабатывается базовый инструментарий [Еремеев и др., 2013].

СПИСОК ЛИТЕРАТУРЫ

[Вагин и др., 1988] Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Известия РАН. Теория и системы управления, 2001. – № 6.

[Вагин, 1988] Вагин В.Н. Дедукция и обобщение в системах принятия решений – М.: Наука, 1988.

[Еремеев и др., 2013] Еремеев А.П., Королев Ю.И. Реализация интеллектуальных систем реального времени на основе сетей Петри с поддержкой темпоральных зависимостей // Программные продукты и системы. – 2013. – №3.

[Еремеев и др., 2014] Еремеев А.П., Королев Ю.И. Анализ и верификация раскрашенных сетей Петри реального времени с поддержкой логики Аллена // Открытые семантические технологии проектирования интеллектуальных систем: материалы IV Междунар. научн.-техн. конф. (Минск, февраль 2014г.)/ редкол.: В.В. Голенков (отв. ред.) [и др.] – Минск: БГУИР, 2014.

[Карпов, 2010] Карпов Ю.Г. Model Cheking. Верификация параллельных и распределенных программных систем – СПб.: БХВ-Петербург, 2010.

[Мурата, 1989] Мурата Т. Сети Петри: Свойства, анализ, приложения // ТИИЭР. – 1989. – Т. 77, №4.

[Clarke et al., 1981] Clarke E.M., Emerson E.A. Design and synthesis of synchronization skeletons using branching-time temporal logic // Logic of Programs, 1981.

[Clarke et al., 1986] Clarke E.M., Emerson E.A., Sistla A.P. Automatic verification of finite-state concurrent systems using temporal logic specifications // ACM Trans. Program. Lang. Syst., 8(2), 1986.

[Floyd, 1967] Floyd R.W. Assigning meaning to programs. // Proc. Symposium on Applied Mathematics, v. 9, 1967.

[Kripke, 1963] Kripke S.A. Semantical consideration on modal logic // Acta Philosophica Fennica, v.16, 1963.

[Milner, 1980] Milner R. A Calculus of Communicating Systems. Lecture Notes in Computer Science, Vol 92, Springer-Verlag 1980.

[Pnueli, 1977] Pnueli A. The temporal logic of program // Proc. of the 18th Anny. Symp. on Foundation of Computer Science, 1977.

[Szpyrka, 2008] Szpyrka M. Modelling and Analysis of Real-Time Systems with RTCP-Nets // Petri Net, Theory and Applications. I-Tech Education and Publishing. – 2008.

УДК 517.11

**ОПЕРАТИВНОЕ ЦЕЛЕПОЛАГАНИЕ
В АНТРОПОЦЕНТРИЧЕСКИХ ОБЪЕКТАХ
С ПОЗИЦИИ МОДЕЛИ «ЭТАП»**

С.Ю. Желтов (*zhl@gosniias.ru*)

Б.Е. Федун (*boris_fed@gosniias.ru*)

ФГУП Национальный научный центр Государственный
научно-исследовательский институт авиационных систем
«ФГУП ГосНИИАС», Москва, Россия.

Современные сложные антропоцентрические объекты (Антр/объекты) с позиций управления ими удобно описывать концептуальной математической моделью «Этап». Модель формализует как сферу функционирования Антр/объекта, так и сферу задач, которые решаются на объекте. Оперативное целеполагание – «первичная фаза управления» в Антр/объекте. В терминах модели «Этап» – это оперативная смена типовой ситуации (ТС) текущего сеанса функционирования объекта. Структура алгоритма оперативного целеполагания: а) априорно заданная последовательность ТС (множество «А»), обеспечивающая решение генеральной задачи предстоящего сеанса функционирования Антр/объекта, б) априорно заданное множество ТС (множество «Б»), которые могут возникнуть в процессе сеанса функционирования, в) матрица знаний с терминами лингвистических переменных для каждой пары ТС (из множества «А») – ТС (из множества «Б»), позволяющая рассчитать приоритеты возникшей ТС (из множества «Б») и текущей ТС (из множества А). При описании оперативного целеполагания используются понятия осмотрительность экипажа Антр/объекта, его ситуационная осведомленность, ситуационная уверенность экипажа.

Введение

Антропоцентрический объект (Антр/объект) – это некоторая физическая оболочка, содержащая совокупность: бортовых измерительных систем, бортовых исполнительных систем, развитую бортовую цифровую вычислительную систему (БЦВС), экипаж (операторы) и его кабину с современным информационно-управляющим полем (ИУП).

Антропоцентрический объект готовится к каждому сеансу функционирования, получая на борт генеральную задачу сеанса функционирования (ГЗФ), последовательность планируемых этапов выполнения сеанса, ожидаемые угрозы [[Ефимов, 2003].

Антр/объекты с позиций проектирования алгоритмов бортового алгоритмического и индикационного обеспечения (АиИО) удобно описывать концептуальной математической моделью «Этап» [Федунов, 2006].

Модель «Этап» для Антр/объекта формализует сферу деятельности Антр/объекта и классифицирует задачи его системообразующего ядра (рис.1).

Формализация сферы деятельности Антр/объекта включает в себя выделение множества ГЗФ, представление каждой из них через семантическую сеть типовых ситуаций (ТС), представление каждой ТС через семантическую сеть проблемных субситуаций (ПрС/С).

Классификация задач системообразующего ядра Антр/объекта включает в себя выделение двух глобальных уровней управления (ГЛУУ): уровень оперативного целеполагания (ИГЛУУ), на котором решается задача назначения текущей ТС (задача оперативного целеполагания); уровень конструирования рационального способа достижения цели активизированной ТС (ИГЛУУ – на нем решается задача оперативного построения конкретного фрагмента семантической сети ПрС/С активизированной ТС и решаются задачи текущей ПрС/С этой ТС).

Модель «Этап» позволяет классифицировать бортовые интеллектуальные системы:

– интеллектуальная информационная система «Ситуационная осведомленность экипажа» (ИИС СОЭ), предоставляющая экипажу на ИУП кабины Антр/объекта информационную модель внешней и внутрибортовой обстановки с выделением в ней непосредственно угрожающих объектов [3]. Экипаж использует информацию с ИИС СОЭ для решения задачи оперативного целеполагания в соответствии с выполняемой ГЗФ и рангом Антр/объекта в группе (при выполнении ГЗФ группой Антр/объектов);

– интеллектуальные бортовые системы каждого этапа (типовой ситуации (ТС)) функционирования Антр/объекта (из класса бортовых оперативно советующих экспертных систем – БОСЭС ТС), рекомендуемых экипажу способ достижения цели этого этапа, с глубиной проработки рекомендаций, достаточных для автоматической их реализации при согласии на это экипажа.

Задачи I ГЛУУ и II ГЛУУ решаются в системообразующем ядре Антр/объекта.

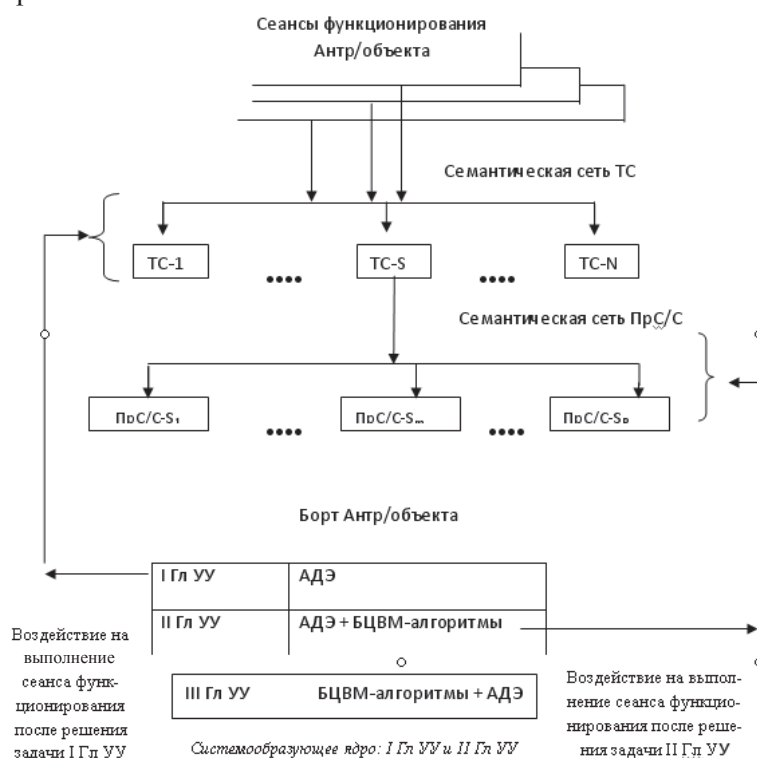


Рис. 1. Модель Антр/объекта «Этап» для проектирования бортового АйИО для системообразующего ядра Антр/объекта

Модель процесса функционирования Антр/объекта (рис.1) содержит набор назначенных (в техническом задании на разработку Антр/объекта) к алгоритмизации сеансов функционирования (с описанием генеральной задачи каждого сеанса); представление каждого сеанса через дерево (семантическую сеть) типовых ситуаций (ТС) и представление каждой ТС в свою очередь через дерево (семантическую сеть) проблемных субситуаций (ПрС/С). Семантические сети строятся по причинно-следственному отношению.

Множество ТС конечно и строится по всей совокупности заданных сеансов функционирования.

На современном уровне развития теории и практики создания бортового АиИО решение задач ИГЛУУ доступно только экипажу. Инженеры проектировщики АиИО для обеспечения решения экипажем этой задачи должны создать на ИУП легко воспринимаемую экипажем информационную модель внешней и внутрибортовой обстановки, обеспечивающую экипажу ситуационную осведомленность для назначения текущей адекватной ТС (см. модель «Этап»). Эта информационная модель реализуется бортовыми БЦВМ-алгоритмами (названная выше ИИС СОЭ) и предъявляется экипажу на ИУП кабины.

Задачи ИГЛУУ могут решать создаваемые в настоящее время интеллектуальные системы класса бортовых оперативно советующих экспертных систем типовых ситуаций (БОСЭС ТС) сеансов функционирования Антр/объекта [Федунов, 2002]. Выработываемые ими решения предъявляются на ИУП экипажу в качестве рекомендаций. Экипаж анализирует и санкционирует их исполнение.

1. Структура алгоритмов оперативного целеполагания

В модели «Этап» задачей ИГЛУУ является оперативное назначение экипажем текущей ТС. Мотивация такого назначения складывается из [Ефимов, 2003]:

- поставленной генеральной задачи сеанса функционирования, на выполнение которой нацелен экипаж;
- состояния материальной части Антр/объекта;
- возникшей или ожидаемой внешней естественной (для ЛА гроза, потеря ориентировки и т.д.) или искусственной (тер/акт, воздушный противник и т.д.) угроз;
- состояния экипажа;
- появления новой цели сеанса функционирования (в том числе и обеспечения физического и психологического гомеостаза Антр/объекта).

Часть из этих мотивов слабо структурирована или даже вербально не обозначена. В силу этого на Антр/объектах задачи ИГЛУУ решаются только экипажем. Для бортового АиИО остается только создание на ИУП информационной модели, адекватной сложившейся внешней и внутрибортовой обстановки.

Оперативное целеполагание, осуществляемое экипажем Антр/объекта в процессе выполнения подготовленного сеанса функционирования, включает в себя следующие моменты [Ефимов, 2003]:

а) осмотрительность экипажа, предусматривающая обнаружения им потенциальных угроз, которые могут помешать выполнению текущего этапа (текущей ТС) сеанса функционирования Антр/объекта;

б) ситуационную осведомленность экипажа, предусматривающую выделение среди наблюдаемых сущностей (во внебортовом и во внутрибортовом мире Антр/объекта) тех сущностей, которые непосредственно угрожают выполнению текущего этапа (текущей ТС) сеанса функционирования Антр/объекта или его гомеостазу;

в) ситуационную уверенность, позволяющую экипажу вовремя обнаружить непосредственную угрозу и выбрать способ эффективного противодействия ей;

г) переход к реализации выбранного способа. В работе не рассматривается осмотрительность экипажа, которая на борту Антр/объекта обеспечивается:

– размещением мощных бортовых измерительных систем, контролирующих внешний и внутрибортовой мир и предъявляющих экипажу обнаруженные сущности с выделением среди них тех сущностей, которые могут представлять угрозу;

– конструированием кабины экипажа с достаточным обзором из нее окружающего внешнего мира.

Наше внимание будет сосредоточено на позициях б) и в), для которых будут вскрыты структуры алгоритмов, решающих имеющиеся там проблемы.

1.1. Априорная информация для назначения экипажем текущей цели функционирования

Априорная информация для оперативного целеполагания, «загружаемая» на борт Антр/объекта перед предстоящим сеансом функционирования:

– задание последовательности ТС предстоящего сеанса (обозначим через «А» это множество ТС).

Наглядный пример №1. Антр/ объект «Пассажирский самолет». ГЗВ «Перевозка пассажиров».

Последовательность ТС для пассажирского самолета:

ТС «Посадка пассажиров» → ТС «Рулежка» → ТС «Взлет» → ТС «Выход на маршрут» → ТС «Маршрут» → ТС «Заход на посадку» → ТС «Посадка» → ТС «Рулежка» → ТС «Высадка пассажиров». Все названные ТС составляют множество «А»;

– множество возможных угроз (множество «У-Б»), которые могут появиться в процессе выполнения Антр/объектом сеанса функционирования;

– таблица соответствия «угроза (из множества «У-Б») – ТС, в которой вырабатывается метод эффективного противодействия угрозе. Множество таких ТС обозначим «Б»».

Наглядный пример №2. Антр/ объект «Пассажирский самолет». ГЗВ «Перевозка пассажиров».

Множество «У-Б» для гражданского пассажирского самолета состоит из следующих типов возможных угроз: «Значимые отказы бортовой аппаратуры», «Проблемы в пассажирском салоне», «Недостаток топлива», «Внешние угрозы (гроза, встречные самолеты)»;

– для каждой пары «ТС из множества «А» – ТС из множества «Б»» разрабатывается матрица знаний, позволяющая рассчитывать приоритеты этих ТС [6].

Наглядный пример №3. Антр/ объект «Пассажирский самолет». ГЗФ «Перевозка пассажиров».

Матрица знаний для пары ТС (рис. 2).

Используемый опыт	Координаты ситуационного вектора SV(ТС «Маршрут» - ТС «Возврат на аэродром»)				Конкурирующая ТС
	X1: ЛП «Расстояние до аэропорта»	X2: ЛП «Возможность блокировки дебошира»	X3: ЛП «Реакция пассажиров»	X4: ЛП «Влияние на престиж авиакомпании»	
Множество термов лингвистической переменной (ЛП)	Большое (bg) Среднее (md) Малое (sm)	Большая (eff) Средняя (smef) Небольшая (insg)	Возмущение (indg) Недовольство (dspl) Неодобрение (dspr)	Большое (cns) Среднее (mtk) Малое (ngr)	
1.1	md	eff	indg	mtk	ТС «Маршрут»
1.2	bg	smef	dspl	cns	
2.1	sm	insg	dspl	cns	ТС «Возврат на аэродром»
2.2	md	smef	indg	mtk	

Рис. 2. Матрица знаний для пары «ТС «Маршрут» (из множества «А») – ТС «Возврат на аэродром» (из множества «Б») для угрозы «Дебошир на борту пассажирского самолета»)

Представление термов лингвистических переменных X1-X4, используемых в матрице знаний (рис. 2) через ансамбли нечетких множеств, показано на рис. 3–6.

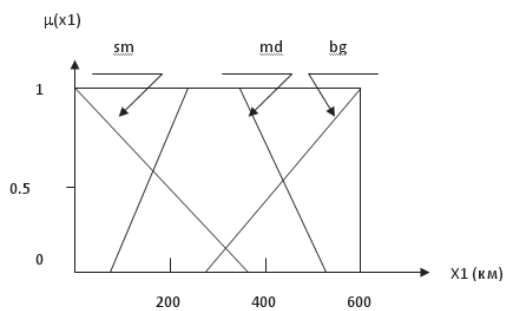


Рис. 3. Представление термов лингвистической переменной X1 через ансамбль нечетких множеств, определяемых на универсальном множестве [0 - 600]

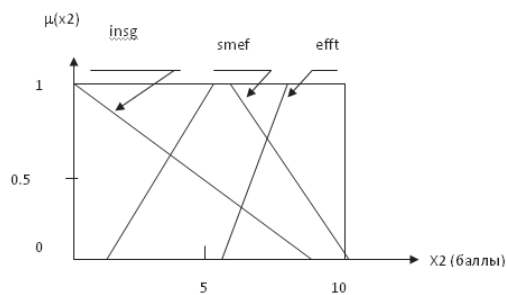


Рис. 4. Представление термов лингвистической переменной X2 через ансамбль нечетких множеств, определяемых на универсальном множестве [0 - 10]

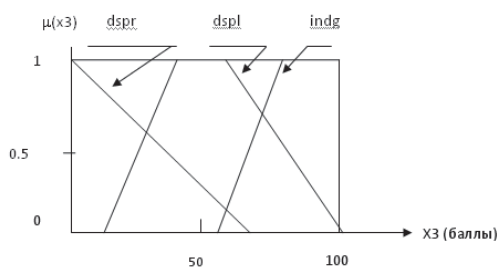


Рис. 5. Представление термов лингвистической переменной X3 через ансамбль нечетких множеств, определяемых на универсальном множестве [0 - 100]

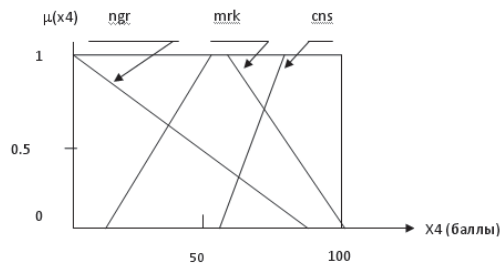


Рис. 6. Представление термов лингвистической переменной X3 через ансамбль нечетких множеств, определяемых на универсальном множестве [0 - 100]

1.2. Интеллектуальная система, обеспечивающая экипажу ситуационную осведомленность

Для Антр/объектов типа самолет истребитель такую информационную модель создает специальная бортовая интеллектуальная информационная система «Ситуационная осведомленность экипажа» (ИИС СОЭ). Она относится к классу систем, не предполагающих использования диалоговых процедур с пользователем. Ее база знаний имеет одноуровневую семантическую структуру с набором продукционных правил, подключающих к выполнению в зависимости от текущей обстановки определенную группу БЦВМ-алгоритмы [Васильев, 2010].

В БЦВС антропоцентрического объекта реализована интеллектуальная информационная система ситуационной осведомленности экипажа (ИИС СОЭ) в составе функциональных блоков: внешние угрозы, внутрибортовые угрозы, оценка достаточности бортовых расходимых средств. В каждом блоке ИИС СОЭ выделяются непосредственные угрозы, при игнорировании которых почти сразу срывается выполнение текущего этапа функционирования Антр/объекта.

Выходная информация системы предъявляется оператору на индикаторах ИУП и через бортовые речевые информаторы.

Размещение на борту Антр/объекта ИИС СОЭ является только подспорьем экипажа в решении им задачи оперативного целеполагания – назначения ТС. Развернутая на ИУП информационная модель внешней и внутрибортовой обстановки решает только про-

блему выделения угроз, непосредственно мешающих выполнению текущей ТС.

1.3. БЦВМ-алгоритм «Оперативное целеполагание»

С позиций ММ «Этап» составляющие БЦВМ-алгоритма оперативного целеполагания следующие (рис.7):

- сопоставление возникшей угрозы (из множества «У-Б») с ТС, в которой вырабатывается способ эффективного противодействия возникшей угрозе (ТС из «Б»);
- сопоставление выполняемой/текущей ТС (ТС из множества «А») с ТС из множества «Б», в которой дается способ эффективного противодействия этой угрозе (в блоке «Таблица соответствия «Угроза (из множества У-Б) – «ТС (из множества «Б»)»);
- расчет приоритетов [Федунов, 2002] этих двух ТС, используя вывод по прецеденту (в блоке База матриц знаний «ТС (из множества «А») – ТС (из множества «Б»)» с числовыми X^*1, \dots, X^*N);
- выбор приоритетной ТС и предъявление ее экипажу.

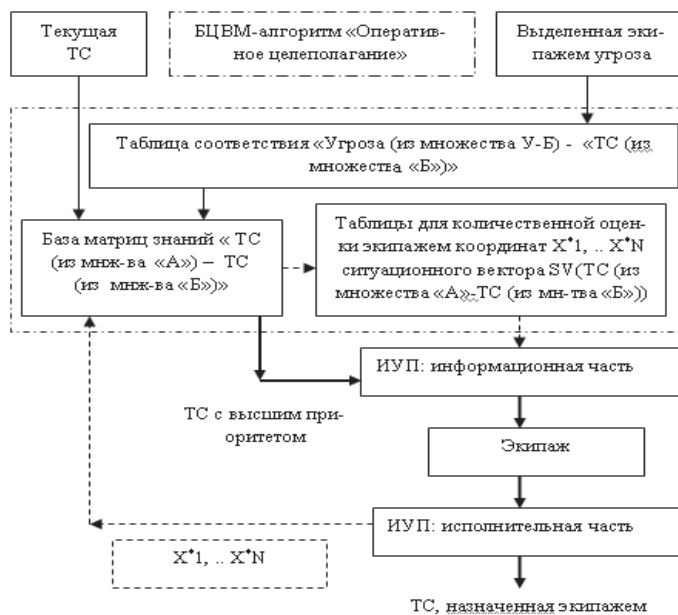


Рис. 7. Функциональная схема БЦВМ-алгоритм «Оперативное целеполагание»

Входная информация алгоритма: текущая ТС; выделенная экипажем угроза; количественная оценка экипажем сложившейся ситуации (см. пример в разделе 2).

Выходная информация алгоритма: таблица для количественной оценки экипажем координат $X^*1, .. X^*N$ ситуационного вектора SV(ТС (из множества «А» – ТС (из множества «Б»)), выбираемая в блоке «Таблицы для количественной оценки ...» ; ТС с высшим приоритетом

1.4. Оперативное целеполагание экипажем

Задача оперативного целеполагания в процессе выполнения Антр/объектом подготовленного сеанса функционирования возникает перед экипажем при обнаружении им непосредственной угрозы текущему этапу (текущей ТС) выполняемого сеанса функционирования (рис. 8).

Априорная информация и наличие на борту Антр/объекта ИИС «Ситуационная осведомленность экипажа» являются, конечно, только информационным подспорьем экипажу в решении им задачи оперативного целеполагания. Предъявляемая ими на ИУП информационная модель внешней и внутрибортовой обстановки решает только проблему выделения угроз, непосредственно мешающих выполнению текущей ТС. Причем предъявленную ИИС СОЭ непосредственную угрозу экипаж, обладающий дополнительной неформализованной информацией, хочет еще подтвердить [Васильев, 2010].

При возникновении для выполняемой ТС угрозы, которую предъявляет ИИС СОЭ или которую обнаруживает сам экипаж (эвристический алгоритм), экипаж фиксирует эту угрозу (сплошные линии связи на рис.4). По этому сигналу в базе матриц алгоритма «Оперативное целеполагание» активизируется матрица «текущая ТС (из множества «А») – ТС (из множества «Б»), соответствующая выделенной экипажем угрозе». Для количественной оценки экипажем возникшей ситуации ему предъявляется на ИУП соответствующая таблица (пунктирные линии связи на рис. 8).

Столбцы этой таблицы (слева направо):

- название координат ситуационного вектора этой матрицы;
- универсальные множества нечетких множеств, соответствующих лингвистическим переменным, и размерность их точек;
- столбец оценки ситуации экипажем (пример см. в разделе 3).

Экипаж оценивает по координатам сложившуюся ситуацию и передает эту информацию в алгоритм «Оперативное целеполагание» (пунктирные линии связи на рис. 4), где рассчитываются приоритеты текущей ТС и ТС, выделенной из множества «Б».

В БЦВМ-алгоритме «Оперативное целеполагание» определяется приоритетная ТС, которая предъявляется экипажу на ИУП в качестве рекомендации как новая цель функционирования (жирные линии связи на рис. 4). Решение о принятии этой рекомендации к исполнению принимает экипаж.

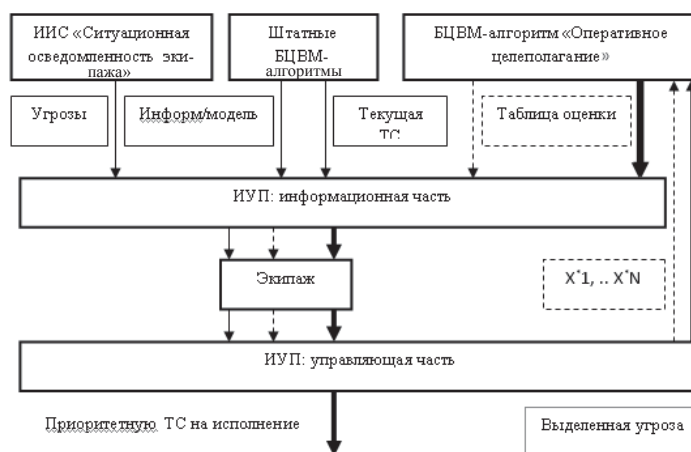


Рис. 8. Решение экипажем задачи оперативного целеполагания (задача ПЛУУ)

Работа алгоритма показана на примере, приведенном ниже в разделе 2.

2. Пример назначения оперативной цели полета экипажем пассажирского самолета

Антропоцентрический объект – пассажирский самолет. *Сеанс функционирования*: перевозка пассажиров из одного аэропорта в другой.

Априорная информация:

- заданная последовательность ТС (множество «А») предстоящего сеанса функционирования (см. наглядный пример №1);
- множество угроз, которые могут появиться в процессе вы-

полнения Антр/объектом предстоящего сеанса функционирования (множество «У-Б»). См. наглядный пример №2;

- априорная информация для оперативного целеполагания, усвоенная экипажем:

- множество ТС (множество «Б»), в которых обеспечивается эффективное противодействие угрозам из множества «У-Б».

Пусть в текущей ТС «Маршрут» возникла непосредственная угроза из раздела угроз «Проблемы в пассажирском салоне»: «Дебошир в салоне самолета». Экипаж выделил эту угрозу и передал ее в БЦВМ-алгоритм «Оперативное целеполагание». В алгоритме в блоке «Таблица соответствия «Угроза (из множества У-Б) – ТС (из множества «Б»)» этой угрозе сопоставлена ТС ««Возврат на аэродром (из множества «Б»)». Из блока «Базы матриц знаний «ТС (из множества «А») – ТС (из множества «Б»)» выбирается матрица «ТС «Маршрут» (текущая ТС) – ТС «Возврат на аэродром (из множества «Б»)» (рис. 6).

Экипаж, выполняющий ТС «Маршрут», выделив возникшую угрозу, активизирует связь «Угроза – ТС из множества «Б»», в которой эффективно может решаться возникшая проблема. В приведенном примере: угроза ««Дебошир на борту», соответствующая ТС из множества «Б» – ТС «Возврат на аэродром». Вслед за этим из БЦВМ-алгоритма «Оперативное целеполагание» экипажу предъявляется таблица ситуационного вектора для количественной оценки его координат (рис. 9). Пусть сложившуюся ситуацию экипаж оценил точкой $X^*1 = 350$, $X^*2 = 4$, $X^*3 = 60$, $X^*4 = 40$ из декартового произведения универсальных множеств $[0-600] \times [0-10] \times [0-100] \times [0-100]$.

Координаты ситуационного вектора SV(ТС "Маршрут" - ТС «Возврат на аэродром»)	Универсальное множество $[...]$ Размерность оценки	Ваша оценка ситуации (оценка должна браться из соответствующего универсального множества)
X1 = «Расстояние до аэропорта»	[0, 600], км	$X^*1 = 350$,
X2 = «Возможности блокировки дебошира»	[0, 10], баллы	$X^*2 = 4$
X3 = «Реакция пассажиров»	[0, 100], баллы	$X^*3 = 6$
X4 = «Влияние на престиж авиакомпании»	[0, 100], баллы	$X^*4 = 40$

Рис. 9. Таблица количественной оценки экипажем координат ситуационного вектора SV(ТС «Маршрут» – ТС «Возврат на аэродром»)

В соответствии с [Федунов, 2002] определяются приоритеты ТС «Маршрут» и ТС «Возврат на аэродром» (рис.10).

Используемый опыт	Оценка степени принадлежности координат точки $X^*1=350$, $X^*2=4$, $X^*3=60$, $X^*4=40$ соответствующему нечеткому множеству [6]				Вычисление приоритетов ТС		
	X_1	X_2	X_3	X_4	min	Приоритет ТС (max)	Конкурирующие ТС
1.1	0.10	0.00	0.15	0.65	0.00		ТС «Маршрут»
1.2	0.20	0.90	0.95	0.00	0.00	0.00	
2.1	0.10	0.60	0.95	0.00	0.00		ТС «Возврат на аэродром»
2.2	1.00	0.90	0.15	0.65	0.15	0.15	

Рис.10. Числовая матрица знаний для точки $X^*1=350$, $X^*2=4$, $X^*3=60$, $X^*4=40$

В результате вычислений определяется ТС с высшим приоритетом. Это ТС «Возврат на аэродром», которая и рекомендуется экипажу в качестве оперативной цели функционирования.

Заключение

1. В докладе проф. Groumpos Peter P. [Groumpos, 2013] представлена классификация интеллектуальных систем поддержки принятия решений, включающая только системы, поддерживающие решение задач II ГЛУУ и III ГЛУУ. Систем, поддерживающих решение задач I ГЛУУ, в обзоре нет. Это слабо формализованная предметная область и наша работа является, по-видимому, первой, в которой конструктивно описывается алгоритм оперативного целеполагания на Антр/объектах (см., например, [Осипов и др., 2014]).

2. Задача оперативного целеполагания в процессе выполнения Антр/объектом подготовленного сеанса функционирования воз-

никает при появлении непосредственной угрозы текущему этапу (ТС). С позиций ММ «Этап» суть этой задачи – сопоставление текущей ТС с ТС, в которой эффективно решается возникшая перед экипажем проблема, и выбор для реализации приоритетной ТС.

3. Структура алгоритма оперативного целеполагания, используемого экипажем Антр/объекта, строится на базе априорной информации:

- последовательность ТС, обеспечивающая выполнения генеральной задачи предстоящего сеанса функционирования (множество ТС «А»);

- множество возможных угроз (множества «У-Б») и сопоставленных им ТС (из множества «Б»), в которых эффективно решается возникшая перед экипажем проблема:

- матрицы знаний для каждой пары ТС (из множества «А») – ТС (из множества «Б»), соответствующей возникшей угрозе.

При возникновении угрозы выполняемой ТС экипаж количественно оценивает возникшую ситуацию. По этой оценке в матрице знаний определяется приоритетная ТС, которая предъявляется экипажу на ИУП в качестве оперативной цели функционирования Антр/объекта.

СПИСОК ЛИТЕРАТУРЫ

[Васильев, 2010] Интеллектуальные системы управления. Под редакцией акад. РАН С.Н. Васильева. Изд. Машиностроение. 2010.

[Ефимов, 2003] Ефимов А.Н. Штурмовики идут на цель. Изд. Патриот. М., 2003. 479 с.

[Осипов и др., 2014] Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания: I. Картина мира и целеполагание. М. ТиСУ. №4. 2014.

[Федунов, 2002] Федунов Б.Е. Механизмы вывода в базе знаний бортовых оперативно советующих экспертных систем // Изв. РАН. ТиСУ. 2002. №4.

[Федунов, 2006] Федунов Б.Е. Макромодель пилотируемых летательных аппаратов для разработки алгоритмов бортового интеллекта. Журнал «Мехатроника, автоматизация, управление». №3. 2006. Приложение «Управление и информатика в авиакосмических системах». Стр. 13–16.

[Groumpos, 2013] Peter P. A critical overview of decision support systems (DSS): theories, approaches and challenges. CSIT'2013.

УДК 004.89

ГИБРИДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ: КООРДИНАЦИЯ И СОГЛАСОВАННОСТЬ

И.А. Кириков (*baltbipiran@mail.ru*)

Калининградский филиал Института проблем информатики РАН,
Калининград

А.В. Колесников (*avkolesnikov@yandex.ru*)

Балтийский федеральный университет имени Иммануила Канта,
Калининградский филиал
Института проблем информатики РАН, Калининград

С.В. Листопад (*ser-list-post@yandex.ru*)

Калининградский филиал Института проблем информатики РАН,
Калининград

С.А. Солдатов (*baltbipiran@mail.ru*)

ООО «ГИМАС +», Калининград

В работе рассматриваются вопросы моделирования гетерогенности систем поддержки принятия решений на основе координации и согласованности. Предлагаемые модели апробированы на сложных задачах планирования производства и транспортной логистики.

Введение

Двадцатый век принес в искусственный интеллект (ИИ) «свежий ветер» перемен социальной в смысле М.Минского и системной в смысле В.Б.Тарасова парадигм, исходящих из факта, что отдельно взятая технология ИИ обладает как преимуществами, так и недостатками, и только в продуманной комбинации методов и моделей может возникнуть эмерджентность и синергетический эффект, когда дополнительность и сотрудничество сведут к минимуму частные недостатки и позволят автоматизированно решать сложные задачи практики принятия решений. Краткая история гибридизации изложена в [Колесников, 2001].

На рубеже XX и XXI веков знания изучены, обобщены и опубликованы в работах А.Н. Борисова [Борисов и др., 1986], А.В. Гаврилова [Гаврилов, 2003], А.В. Колесникова [Колесников, 2001; Колесников и др., 2011а], Д.А. Пospelова [Гаазе-Раппопорт и др., 1987], Г.В. Рыбиной [Рыбина, 1998], В.Б. Тарасова [Тарасов, 2002],

Н.Г. Ярушкиной [Ярушкина, 2004], S. Goonatilake и S. Khebbal [Goonatilake et al., 1992], L. Medsker [Medsker, 1995]. В России сложились несколько школ в этой области междисциплинарных знаний: Д.А. Поспелова – В.Б. Тарасова; В.Н. Вагина – А.П. Еремеева; Г.В. Рыбиной в Москве; Н.Г. Ярушкиной в Ульяновске; В.Ф. Пономарева – А.В. Колесникова в Калининграде.

Цель настоящей работы – отобразить в моделях гетерогенность для релевантности информационных интеллектуальных систем оригиналу, а их консультации по качеству сделать не хуже, чем предлагают коллективы, принимающие решения в сложных задачах.

1. Естественные и искусственные гетерогенные системы

Гетерогенная система (от греческого ἕτερος – разный; γένω – рождать) – неоднородная система, состоящая из однородных частей, разделенных поверхностью раздела, разрывами в непрерывности физических свойств. Однородные части могут отличаться друг от друга по составу и свойствам.

На рис. 1 показана малая часть примеров естественных гетерогенных систем. Актуальность исследований и полученных для приведенных на рис. 1 объектов-оригиналов не вызывает сомнений. Калининградская школа сконцентрировала свое внимание на системах поддержки принятия решений (СППР) – коллективах экспертов под руководством лица, принимающего решения [Колесников, 2001; Колесников и др., 2011а].

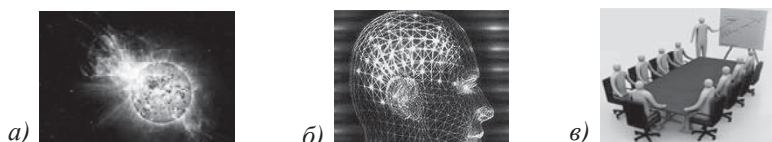


Рис. 1. Примеры естественных гетерогенных систем: а) звезда «Солнце»; б) головной мозг человека; в) СППР

Искусственные гетерогенные системы создаются человеком в разных сферах деятельности с использованием междисциплинарных знаний для получения объектов с полезными для людей свойствами. На рис. 2 приведены примеры искусственных гетерогенных систем.

2. Математические модели искусственных гетерогенных систем в информатике

К искусственным гетерогенным системам отнесем: гибридные системы, агрегаты и агрегативные системы, многомодельные системы, интегрированные экспертные и синергетические системы, гибридные интеллектуальные системы (ГИИС) и эмерджентные модели. Рассмотрим ГИИС и синергетические системы Калининградской школы.

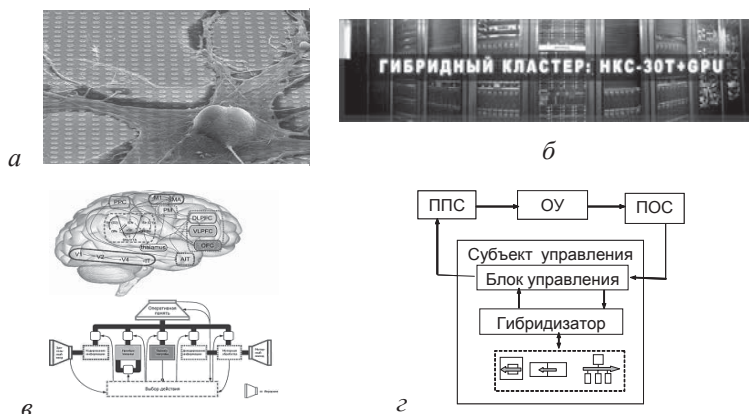


Рис. 2. Примеры искусственных гетерогенных систем: а) гибрид микросхемы и нейрона крысы; б) гетерогенный кластер Сибирского отделения РАН; в) искусственный мозг человека Sprain канадских специалистов; г) гибридная адаптивная система управления А.В. Колесникова

Функциональная гибридная интеллектуальная система (ФГиИС) – записанная в аксиоматической теории схем ролевых концептуальных моделей система, использующая для решения задачи несколько методов имитации интеллектуальной деятельности человека. Крупнозернистые ФГиИС для решения гетерогенной задачи pr^b^u специфицируются следующей схемой ролевых концептуальных моделей:

$$\begin{aligned}
 res_A^u &= R_1^{res\ met}(res_A^u, met^u) \circ R_1^{res\ pr}(res_A^u, pr^{ui}) \circ R_1^{res\ pr}(res_A^u, pr^{uo}) \circ \\
 &\circ R_1^{res\ st}(res_A^u, st^u) \circ R_1^{st\ st}(st^u(t), st^u(t+1)) \circ R_1^{pr\ st}(pr^{ui}(t), st^u(t+1)) \circ \\
 &\circ R_1^{st\ pr}(st^{up}(t), pr^{uo}(t)) \circ R_1^{res\ res}(RES^o, RES^o) \circ R_1^{pr\ pr}(pr^{ui}, PR^{zi}) \circ \\
 &\circ R_2^{pr\ pr}(PR^{zo}, pr^{uo}),
 \end{aligned} \quad (2.1)$$

где t – модельное время, $t = 0, 1, 2, 3, \dots$; \circ – конкатенация; res_A^u – ГиИС, ресурс решения неоднородной задачи; met^u – интегрированный метод решения prb^u ; pr^{ui} – исходные данные DAT^u [Колесников и др., 2011a] задачи prb^u , передаваемые на вход одного или нескольких элементов res^3 из декомпозиции \widehat{prb}^u задачи prb^u ; pr^{uo} – выход одного или нескольких элементов res^3 из декомпозиции \widehat{prb}^u , как цель GL^u решения задачи prb^u ; $st^u(t)$ – состояние ГиИС; RES^3 – непустое множество, как минимум из двух элементов res^3 ; PR^{3i}, PR^{3o} – множества свойств «вход» и «выход» элементов из RES^3 соответственно; $R_1^{st\ st}, R_1^{pr\ st}, R_1^{st\ pr}$ – отношения функционирования ГиИС; $R_1^{res\ res}$ – отношения интеграции [Колесников и др., 2011a] элементов; $R_1^{pr\ pr}$ – отношения между входами ГиИС и входами элементов; $R_2^{pr\ pr}$ – отношения между выходами элементов и выходами ГиИС. Элемент res^3 решает подзадачи или выполняет вспомогательные операции методом met^3 и имеет свойства $PR^3 \subseteq PR$, наиболее важные из которых – «вход» pr^{3i} , «выход» pr^{3o} , а также «состояние» st^3 .

Схема элемента ГиИС выглядит следующим образом:

$$\begin{aligned}
 res^3 = & R_1^{res\ met}(res^3, met^3) \circ R_1^{res\ pr}(res^3, pr^{3i}) \circ R_1^{res\ pr}(res^3, pr^{3o}) \circ \\
 & \circ R_1^{res\ st}(res^3, st^3) \circ R_1^{st\ st}(st^3(t), st^3(t+1)) \circ R_1^{pr\ st}(pr^{3i}(t), st^3(t+1)) \circ \\
 & \circ R_1^{st\ pr}(st^3(t), pr^{3o}(t)),
 \end{aligned} \quad (2.2)$$

где $R_1^{st\ st}, R_1^{pr\ st}, R_1^{st\ pr}$ – отношения «состояние – состояние», «вход – состояние», «состояние – выход». На множестве автономных методов $MET^3 = \{met_y^3 \mid y = 1, \dots, N_{met}\}$ различаются met_1^3 – аналитические, – met_2^3 нейронные, met_3^3 – нечеткие, met_4^3 – рассуждения на основе опыта, met_5^3 – эволюционные, met_6^3 – статистические, met_7^3 – логические.

Функциональная гибридная интеллектуальная система с координацией – новый класс искусственных гетерогенных систем [Колесников и др., 2011b]. Его описание получено расширением модели (2.1), (2.2). Координация контролирует промежуточные состояния решения подзадач prb^h [Колесников и др., 2011b], т.е. результаты элементов res^3 , и изменяет свойства «вход» pr^{3i} элементов res^3 . Для этого в (2.1), (2.2) введено отношение $R_1^{st\ pr}(st^u(t), pr^{ui}(t+1))$, т.е. по состоянию ФГиИС $st^u(t)$ меняются исходные данные $pr^{ui}(t+1)$ для элементов в момент времени $t+1$. Множество $R_1^{st\ pr}$ устанавливает отношения между текущим состоянием $st^u(t)$ гибрида res_A^u

(2.1) и состоянием входов элементов res^3 на следующем шаге. Для изменения входов pr^{3i} элементов res^3 в (2.2) введено отношение $R_1^{st\ act}(st^u, act^{3k})$, где $ACT^{3k} = \{act_1^{3k\ \alpha}, \dots, act_{N_{pr}}^{3k\ \alpha}\}$ – координирующие действия в базе знаний B_{ext} . Множество $R_1^{st\ act}$ – отношения состояния st^u ФГиИС res_A^u в момент времени t и необходимых координирующих действий ACT^{3k} .

Ниже приведена схема крупнозернистых ФГиИС с координацией:

$$res_A^u = res_A^u \circ R_1^{st\ pr}(st^u(t), pr^{ui}(t+1)), \quad (2.3)$$

и модифицированная схема элемента ФГиИС:

$$res^3 = res^3 \circ R_1^{st\ act}(st^u, act^{3k}). \quad (2.4)$$

Отношения $R_1^{st\ pr}$ и $R_1^{st\ act}$ не задаются априори, а, как и $R_1^{st\ st}$, $R_1^{pr\ st}$, $R_1^{st\ pr}$, фиксируются при работе ФГиИС и решения k -задачи prb^k [Колесников и др., 2011b].

Гибридная интеллектуальная многоагентная система (ГиИ-МАС) с самоорганизацией на основе анализа согласованности целей агентов совмещает преимущества многоагентных систем и ГиИС, моделирует самоорганизацию СППР и определяется выражениями:

$$himas = \langle AG^*, env, INT^*, ORG, \{so^{goa}\} \rangle, \quad (2.5)$$

$$AG^* = \{ag_1, \dots, ag_n, ag^{dm}\}, \quad (2.6)$$

$$INT^* = \{prot, lang, ont, rcl\}, \quad (2.7)$$

$$ORG = ORG_{coop} \cup ORG_{neut} \cup ORG_{comp}, \quad ORG_{coop} \cap ORG_{neut} = \emptyset, \quad (2.8)$$

$$ORG_{coop} \cap ORG_{comp} = \emptyset, \quad ORG_{comp} \cap ORG_{neut} = \emptyset,$$

$$act_{himas} = \left(\bigcup_{ag \in AG^*} act_{ag} \right) \cup act_{ia} \cup act_{ac} \cup act_{col}, \quad (2.9)$$

$$act_{ag} = (MET_{ag}, IT_{ag}), \quad ag \in AG^*, \quad \left| \bigcup_{ag \in AG^*} IT_{ag} \right| \geq 2, \quad (2.10)$$

$$ag = ag \vee himas, \quad (2.11)$$

где AG^* – множество агентов ag , моделей экспертов, включая агента, принимающего решения (АПР) – ag^{dm} ; n – число агентов-экспертов; env – концептуальная модель внешней среды ГиИМАС; INT^* – элементы структурирования взаимодействий агентов: $prot$ – протокол взаимодействия; $lang$ – язык сообщений; ont – модель предметной области; rcl – классификатор отношений агентов:

нейтралитет, конкуренция и сотрудничество; ORG – множество архитектур ГиИМАС (ORG_{coop} – с сотрудничающими; ORG_{neut} – с нейтральными и ORG_{comp} – с конкурирующими агентами); $\{so^{goa}\}$ – множество моделей макроуровневых процессов в ГиИМАС; so^{goa} – модель самоорганизации на основе анализа согласованности целей агентов [Колесников и др., 2011a]; act_{himas} – функция ГиИМАС в целом; act_{ag} – функция агента из множества AG^* ; act_{ia} – функция «анализ взаимодействий» АПР ag^{dm} ; act_{ac} – функция «выбор архитектуры» АПР ag^{dm} ; act_{col} – коллективная функция ГиИМАС, определяемая архитектурой org , конструируемая динамически; met_{ag} – метод решения задачи; it_{ag} – интеллектуальная технология метода met_{ag} .

3. Основные результаты по гетерогенным искусственным системам

Для подтверждения эффективности моделей (2.1) – (2.4) исследована задача оперативно-производственного планирования (ОПП) в ООО завод «Калининградгазавтоматика» и создана автоматизированная система, моделирующая ежедневные планерки управленческого персонала по составлению производственного плана. Автоматизированное решение задачи ОПП моделировалось с учетом координации и без нее (рис. 3). По оси X отложены месяц и год, для которых проводится моделирование решения задачи ОПП, по оси Y – суммарная стоимость заказов.

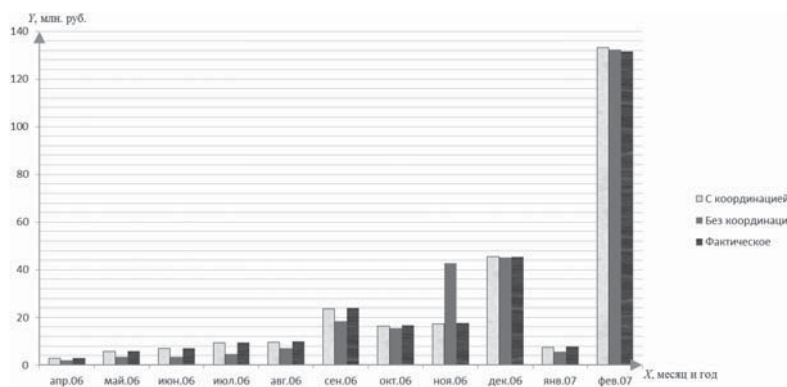


Рис. 3. Результаты решения задачи оперативно-производственного планирования

На рис. 3 видно, что результаты моделирования решения задачи ОПП без учета координации не соответствуют фактическим. Относительная погрешность результатов решения задачи ОПП без учета координации в среднем достигает 36%. Полученные в ходе проведения экспериментов по решению сложной задачи ОПП с учетом координации рекомендации, отображенные в протоколе планерок, получили высокую оценку экспертов. В результате теоретически обоснована необходимость учета координации при решении сложной задачи ОПП и подтверждена релевантность разработанных моделей сложной задаче ОПП.

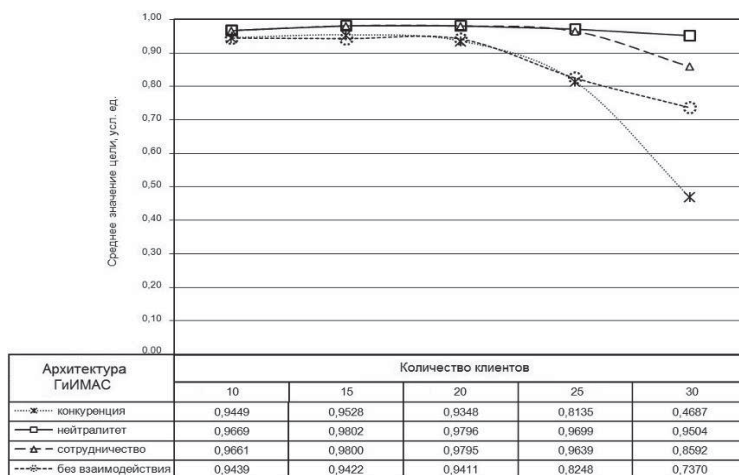


Рис. 4. Среднее значение сводного критерия качества маршрута

Для оценки влияния согласованности целей агентов на качество решений ГИИМАС проведены серии экспериментов, в которых требовалось решить сложную транспортно-логистическую задачу (СТЛЗ), т.е. найти для нескольких транспортных средств совокупность маршрутов, оптимальную по четырем критериям: суммарная стоимость; общая длительность поездок для всех транспортных средств; вероятность опоздания хотя бы к одному клиенту; надежность. Для пяти задач и каждой архитектуры ГИИМАС проведено по 100 вычислительных экспериментов. По всем задачам и архитектурам ГИИМАС, а также для архитектуры без взаимодействия построена графическая зависимость среднего значения сводного

критерия качества маршрута (рис. 4) от числа клиентов. На рис. 4 видно, что качество принимаемых решений ГиИМАС с нейтральными агентами выше, чем ГиИМАС других архитектур. Однако чем меньше размерность задачи, тем меньше влияние согласованности на качество решения.

По итогам практического использования программного продукта, реализующего ГиИМАС, на двух объектах средняя суммарная себестоимость доставки грузов в день сократилась на 7,2%, средняя суммарная длительность доставки в день – на 12,13%, среднее время построения маршрутов в день – на 23,14%.

Заключение

Многолетний научный опыт, накопленный человеческим обществом, со все большей убедительностью показывает, что исследования естественных гетерогенных систем с присущими им разнообразием, дополнительностью, сотрудничеством и относительностью позволяют получать ценные знания о космосе, природе, мироздании, человеке и его роли. Особенно интересны междисциплинарные исследования с целью разработки искусственных гетерогенных систем с полезными для человека свойствами. Во-первых, это согласуется с новой картиной мира, к которой общество постепенно эволюционирует. Во-вторых, это качественный фундамент будущих инноваций. Информатика и ИИ накопили опыт реализации проектов компьютерных СППР, консультации которых позволяют человеку решать более сложные задачи в естественных гетерогенных системах.

СПИСОК ЛИТЕРАТУРЫ

[Борисов и др., 1986] Борисов А.Н. и др. Диалоговые системы поддержки принятия решений на базе мини-ЭВМ: Информационное, математическое и программное обеспечение. – Рига: Зинате, 1986.

[Гаазе-Раппопорт и др., 1987] Гаазе-Раппопорт М.Г., Поспелов Д.А. От амебы до робота: модели поведения. – М.: Наука. Гл. ред физ.-мат. лит., 1987.

[Гаврилов, 2003] Гаврилов А.В. Гибридные интеллектуальные системы. – Новосибирск: Изд-во НГТУ, 2003.

[Колесников, 2001] Колесников А.В. Гибридные интеллектуальные системы. Теория и технология разработки. – СПб.: Изд-во СПбГТУ, 2001.

[Колесников и др., 2011a] Колесников А.В., Кириков И.А., Листопад С.В., Румовская С.Б., Доманицкий А.А. Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем. – М.: ИПИ РАН, 2011.

[Колесников и др., 2011b] Колесников А.В., Солдатов С.А. Моделирование коллективного интеллекта, решающего сложную задачу планирования на машиностроительном предприятии // В мире научных открытий. Серия «Математика. Механика. Информатика». 2011. №12.

[Рыбина, 1998] Рыбина Г.В. Автоматизированное построение баз знаний для интегрированных экспертных систем // Изв. РАН. Теория и системы управления. 1998. №5.

[Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. – М.: Эдиториал УРСС, 2002.

[Ярушкина, 2004] Ярушкина Н.Г. Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004.

[Goonatilake et al., 1992] Goonatilake S., Khebbal S. Intelligent Hybrid Systems // The First Singapore Int. Conference on Intelligent Systems: proceedings. 1992.

[Medsker, 1995] Medsker L.R. Hybrid Intelligent Systems. – Boston: Kluwer Academic Publ., 1995.

УДК 004.81

**КОГНИТИВНАЯ АРХИТЕКТУРА СИСТЕМ ПОДДЕРЖКИ
ПРИНЯТИЯ РЕШЕНИЙ:
СЕМИОТИЧЕСКИЙ ПОДХОД**

А.А. Кулинич (*kulinich@ipu.ru*)
Институт проблем управления РАН, Москва

Рассмотрена когнитивная архитектура СППР, построенная на основе модели интеллекта как формы организации ментального опыта человека и по принципам семиотической системы.

Введение

В системах поддержки принятия решений (СППР) считается, что для сложной и плохо определенной предметной области невозможно создать ее точную модель, которая могла бы предложить единственное и оптимальное решение. В этом случае используются грубые модели, не учитывающие детали и упрощающие реальные процессы предметной области. В результате получаем решение (множество решений), оставляя за пользователем системы возможность выбора решения с учетом его личных предпочтений, не учтенных в моделях предметной области.

Таким образом, СППР включают два компонента: компьютер с программным обеспечением, решающим определенный класс задач, и лицо, принимающее решение (ЛПР) на основании рекомендаций, предложенных компьютером. При этом качество решений, принимаемых таким тандемом, зависит как от качества модели предметной области, так и от интеллектуальных способностей (знаний, опыта принятия решений и т.д.) ЛПР. Особую трудность при принятии решений представляют плохо определенные предметные области, модели которых определяются только экспертным образом. Здесь ответственность за качество принимаемого решения ложится на ЛПР и определяется уровнем его знаний и опыта. В этих случаях в качестве резерва для повышения качества принимаемых решений становится актуальным исследование подходов, направленных на моделирование и стимуляцию когнитивных способностей ЛПР.

В этом направлении развиваются и исследуются когнитивные программные архитектуры СППР, направленные на моделирование когнитивных процессов человека для имитации его способностей решать сложные задачи в условиях неполноты информации. В настоящее время когнитивные архитектуры СППР в динамических плохо определенных ситуациях исследованы недостаточно. Этот пробел устраняет настоящая работа.

1. Когнитивные архитектуры

В настоящее время понятие когнитивная архитектура связывают с архитектурой (программной и аппаратной) интеллектуальных систем, являющихся составной частью интеллектуальных агентов (роботов).

Когнитивная архитектура интеллектуальной системы строится на основе исследований когнитивными психологами интеллекта человека. Считается, что искусственная интеллектуальная система, реализующая когнитивные функции естественного интеллекта, будет способна решать множество трудно решаемых задач, с качеством, свойственным естественному интеллекту человека. Элементами естественной интеллектуальной системы являются следующие когнитивные функции: восприятия и представления в памяти окружающего мира; убеждений; целеполагания; предпочтений; элементы, организующие функционирование всех элементов когнитивной системы человека в процессах познания и решения сложных задач.

Среди реализованных программных когнитивных архитектур искусственного интеллекта наиболее известные архитектуры [Langley et al., 2009]: ACT-R, SOAR, CLARION, COGENT и др.

1.1. Когнитивная архитектура СППР

Рассматриваемая в этой статье когнитивная архитектура СППР основывается на исследованиях психологов, в которых феномен человеческого интеллекта объясняется формой организации его ментального опыта [Холодная, 1997]. Согласно этой модели ментальный опыт неоднороден и состоит из трех взаимодействующих слоев опыта: когнитивный; метакогнитивный; интенциональный.

Когнитивный слой ментального опыта обеспечивает познавательный контакт субъекта с действительностью. Здесь осуществ-

включает распознавание объектов, установление отношений между объектами и представление их в виде модели действительности в ментальном пространстве, которую называют репрезентацией ситуации. Метакогнитивный слой ментального опыта регулирует познавательную деятельность субъекта и управляет процессами разворачивания репрезентации ситуации в ментальном пространстве индивидуального интеллекта. Интенциональный опыт определяет сферу деятельности субъекта, его целеполагание, убеждения и предпочтения. Т.е. структуры метакогнитивного опыта, регулирующие процессы распознавания и разворачивания действительности в ментальном пространстве, в конечном итоге находятся под влиянием структур интенционального опыта.

Считается, что состояние окружающего мира с помощью структур кодирования, когнитивных схем (фреймов, сценариев, понятий, когнитивных карт и т.д.) представляется в виде репрезентации в когнитивном опыте субъекта в семантических и понятийных структурах. При принятии решений интенциональный опыт формирует предпочтения в виде цели, активируя тем самым структуры метакогнитивного опыта, управляющего перестройкой текущей репрезентации к виду, удовлетворяющему поставленной цели.

Такой не вычислительный механизм работы интеллекта исследовался в рамках феноменологического подхода к исследованию интеллекта: гештальт-психологической теории интеллекта [Вергеймер, 1987] и теории интеллектуальной компетентности субъекта [Glaser, 1984], основанной на особенностях организации индивидуальной базы знаний субъекта. Здесь интеллектуальная продуктивность объясняется не столько объемом усвоенных знаний, сколько способами их хранения и воспроизведения. В теории компетенции считается, что переструктуризация семантических структур способна привести к качественным изменениям в понимании проблемной ситуации, что приводит к ее решению, которое называется «инсайтом». Формальных критериев, изменения репрезентации к виду, соответствующему решению проблемы, в рассматриваемых психологических теориях нет.

Предложенная когнитивная архитектура СППР, основанная на семиотическом подходе, позволяет указать направление поиска структуры соответствующей «инсайту» в диалоге с аналитиком.

2. Семиотическая модель когнитивного опыта

Семиотическая модель когнитивного опыта основана на модели знака немецкого логика Г.Фреге. Его модель знака известна как «треугольник Фреге» и включает три связанные составляющие: имя знака, смысл знака и значение знака. Имя знака – это символ, слово, обозначающее в сознании человека объект реального мира. Под смыслом знака Фреге понимает описание объекта реального мира, отличающего его от других объектов. И, наконец, под значением знака Фреге понимает реальный объект, который определяется смыслом знака [Бирюков, 1960].

Особенность представления знания в виде знака заключается в том, что знак связывает ментальные процессы, в которых принимает участие имя, смысл знака, определяемые субъектом и зависящие от уровня его знаний, с объектами реального мира (значением), обозначаемые именем и определяемые его смыслом. Это означает, что между реальным миром и знаками существуют отношения моделирования, позволяющие осуществлять процесс мышления со знаками, а не реальными объектами. Так, основатель семиотики (науки о знаках и знаковых системах) Ч. Пирс считает: «У нас нет способности мыслить без посредства знаков».

В области искусственного интеллекта развивается прикладная семиотика, ориентированная на использование знаков и знаковых систем в моделях представления знаний, принятия решений, реализованных в компьютерных системах [Поспелов, 1999; Осипов, 2002].



Рис. 1

В когнитивную архитектуру СППР в качестве ее элементов включены элементы естественного интеллекта: интенциональный и метакогнитивный слои опыта ЛППР. Для поддержки функционирования этих слоев ментального опыта предложены методики: поддержки естественных способностей эксперта при формулировке предпочтений, цели, оценок достижения цели и т.д. (интенциональный опыт); поддержки аналитического познавательного стиля субъекта – его способностей структуризации репрезентации текущей ситуации (метакогнитивный опыт). Кроме того, строится формальная семиотическая модель когнитивного опыта для моделирования динамики изменения репрезентации ситуации. Архитектура СППР приведена на рисунке 1.

Модель когнитивного опыта включает: схемы кодирования информации, обеспечивающие ее представление; семантические структуры, задающие отношения между понятиями в понятийных структурах; понятийные структуры, обеспечивающие понятийное представление наблюдаемой действительности; механизмы взаимодействия элементов структуры когнитивного опыта.

2.1. Схемы кодирования информации (модель объекта)

В качестве модели представления реальности применяется расширенная модель знака Г. Фреге. В нашей модели объект реального мира v представлен тройкой: имя объекта – d ; множество признаков объекта $\{f_i\}$ и их значения $z_{iq} \in Z_i$, где Z_i – возможные значения признака; представление объекта – это объект реального мира v , имеющий признаки $\{f_i\}$ и для обозначения которого общепринято использовать имя d . Значения признака определены в виде упорядоченного множества лингвистических значений, $Z_i = \{z_{i1}, \dots, z_{iq}\}$, $z_{iq+1} \succ z_{iq}$, $q=0 \dots n-1$, и называются качественным доменом.

Множество объектов предметной области представляется в семантическом пространстве $SS(v)$, определяемом прямым произведением качественных доменов всех его признаков, т.е. $SS(v^0) = \times Z_i$. Окрестность $z_i \pm \varepsilon_i$ значений i -го признака объекта $v^0 = (z_p, \dots, z_m)$, $z_i \in \pm \varepsilon_i$, в котором не меняется имя класса, к которому он принадлежит, будем называть интервалом толерантности класса по этому признаку $\Delta_i = [z_{ik} + \varepsilon_{ik}, z_{ik} - \varepsilon_{ik}]$.

Подпространство семантического пространства $SS(d^0) \subseteq SS(v^0)$, полученное прямым произведением интервалов толерантности

всех признаков объекта v^0 , будем называть базовым понятием или классом d^0 объекта v^0 : $SS(d^0) = \times_i [z_{ik} + \varepsilon_{ik}, z_{ik} - \varepsilon_{ik}] = \times_i \Delta_i$, $SS(d^0) \subseteq SS(v^0)$.

2.2. Закономерности предметной области

В этой работе под закономерностями предметной области в широком смысле подразумеваются функциональные зависимости значений признаков объектов этой предметной области. Эти зависимости задаются экспертом с помощью когнитивной карты (F, W) . Причинно-следственные отношения W для значений признаков в общем случае задаются отображением: $W: \times_i Z_i \rightarrow \times_i Z_i$, где W – множество правил отображения значений факторов. Уравнение динамики значений признаков, позволяющее моделировать исторический процесс изменения их значений – эволюцию, задано системой логико-лингвистических уравнений, которое представлено как конечно-разностное уравнение вида: $Z(t+1) = W^{\circ} Z(t)$, где \circ – правило вывода, $Z(t+1)$, $Z(t) = (z_p, \dots, z_q)$ – состояние ситуации – это векторы значений признаков в последовательные моменты времени рассуждений или моделирования ситуации.

2.3. Семантические и понятийные структуры

Обычными типами связи (отношениями) между классами являются отношения часть-целое, класс-подкласс или связи наследования. Задание связей наследования знаков позволяет определить их место в понятийной иерархии предметной области. В рассматриваемой модели связи наследования представляются в виде так называемого концептуального каркаса [Кулинич, 2013], определенного в семантическом пространстве предметной области. В работе [Кулинич, 2013] были определены: качественное положительное, отрицательное обобщение по одному признаку или по нескольким признакам базового понятия.

Рассмотрим множество всех подпространств $\{SS(d^H)\}$, $\forall H$, полученных в результате качественного обобщения базового понятия по одному, двум и т.д. признакам в семантическом пространстве $SS(v)$, $SS(d^H) \subset SS(v)$.

Множество подпространств $\{SS(d^H)\}$ удовлетворяет свойствам рефлексивности, антисимметричности и транзитивности, для любой пары которых определены верхняя и нижняя границы. В этом случае это множество образует решетку $(\{SS(d^H)\}, \cap, \cup)$, структурирующую семантическое пространство. Решетку подпространств

$(\{SS(d^H)\}, \cap, \cup)$ будем называть качественным концептуальным каркасом этого семантического пространства. Концептуальный каркас отражает возможную идеальную структуру знаний о предметной области и строится на основе качественного индуктивного обобщения базового понятия [Кулинич, 2013].

2.4. Репрезентация ситуации в модели когнитивного опыта

Начальная репрезентация ситуации в модели когнитивного опыта строится для наблюдаемого объекта v . Начальная репрезентация строится в семантическом пространстве объекта и включает: $(d^0, SS(d^0), V(d^0), W(d^0))$, d^0 – имя объекта (класса объектов), $SS(d^0)$ – область толерантности базового понятия, $W(d^0)$ – зависимости значений признаков объекта – его когнитивная карта. Изменение начальной репрезентации возможно либо в результате рассуждений ЛППР или эволюционного процесса ситуации, приводящего к обобщению: имени базового понятия ($d^0 \rightarrow d^H$), области толерантности ($SS(d^0) \rightarrow SS(d^H)$), когнитивной карты ($W(d^0) \rightarrow W(d^H)$).

2.5. Оценка состояния репрезентации

В когнитивной архитектуре СППР для поддержки принятия решений о качестве репрезентации вводится оценка ее состояния. Оценка основана на определении относительной важности всех признаков относительно экспертно заданной цели G_i , $i=1, \dots, n$. Формально оценка состояний может быть представлена в виде следующего отображения:

$$G_i: Z(t) \rightarrow (o_{ij}, \succ),$$

где G_i – отображение текущих значений всех факторов для i -ой цели; $Z(t)$ – текущие значения всех факторов; (o_{ij}, \succ) – упорядоченные оценки важности всех признаков относительно i -ой цели.

Интегральная оценка состояния $O_i(t)$ представляется линейной сверткой, значение которой отражает степень достижения цели:

$$O_i(t) = \sum_j o_{ij} x_j(t), j=1, \dots, n,$$

где $x_j(t) \in [0, 1]$ – элементы вектора состояния $X(t)$ получены в результате отображения φ лингвистических значений признаков $z_j(t) \in Z(t)$ на отрезок числовой оси $[0, 1]$, т.е. $\varphi: z_j(t) \rightarrow x_j(t) \in [0, 1]$.

При поддержке построения репрезентации решения важны не столько оценки достижения цели, а относительное изменение этой оценки в состоянии $Z(t+1)$ и в текущем состоянии $Z(t)$.

Относительное изменение оценки γ_i определится из следующего соотношения:

$$\gamma_i = \frac{O_i(t+1) - O_i(t)}{O_i(t)}.$$

2.6. Поддержка построения репрезентации решения

Выбор новой репрезентации основан на анализе относительно изменения оценки γ_i при ее изменении от начального к обобщенному состоянию $d^0 \rightarrow d^H$ и $SS(d^0) \rightarrow SS(d^H)$.

Если оценка относительного изменения больше нуля, т.е. $\gamma_i > 0$, то динамика изменения (эволюция) системы происходит в направлении к цели. В этом случае никаких действий можно и не принимать. Если оценка относительного изменения меньше нуля, т.е. $\gamma_i < 0$, то динамика изменения (эволюция) системы происходит в направлении от цели. В этом случае формулируются задачи управления и(или) структурного (параметрического) управления.

Управление сводится к поиску управляющих воздействий $U(t)$, направленных на изменение значений признаков, способных изменить траекторию изменения значений признаков и, следовательно, изменить знак оценки относительного изменения состояния, т.е. $\gamma_i > 0$. Формально поиск управляющих воздействий для заданной цели G_i сводится к решению уравнения $G_i = W^o U$ относительно управляющих воздействий U , т.е. к решению обратной задачи. Решение запишется в следующем виде: $U^* = G_i^o W^{-1}$, где U^* – новые значения признаков, позволяющие достичь цели G_i , o – процедура обратного вывода.

Обратные задачи – это некорректно поставленные задачи и поэтому не имеют единственного решения – это множество решений $U^* = \{u_i^*\}$. Если полученные решения отобразить на концептуальный каркас предметной области, то эти решения будут представляться в виде множества репрезентаций возможных решений, $\{(d^{ui}, SS(d^{ui}), V(d^{ui}), W(d^{ui}))\}$. Множество репрезентаций решений образует частично упорядоченное множество по включению содержаний и может быть представлено в виде концептуального каркаса решений $(\{SS(d^{ui})\}, \cap, \cup)$.

Если в этом концептуальном каркасе существует класс решений d^{ui} с непустым объемом – $V(d^{ui}) \neq \emptyset$, т.е. $\exists v \in V(d^{ui})$, то зависимости признаков этого объекта v и будут новой когнитивной картой $W(d^{ui})$ в новой репрезентации, которая является репрезентацией решения.

Заключение

Рассмотрена программная когнитивная архитектура СППР, построенная на основе модели интеллекта как формы организации ментального опыта человека. Отличительной особенностью этой архитектуры является то, что она построена по принципам семиотической системы.

СПИСОК ЛИТЕРАТУРЫ

- [Вертгеймер, 1987] Вертгеймер М. Продуктивное мышление. М.: Прогресс, 1987.
- [Glaser, 1984] Glaser R. Education and Thinking: The role of knowledge/ Amer. Psychologist. V.39 (2). (1984). P. 93–104.
- [Бирюков, 1960] Бирюков Б. В. Теория смысла Готлоба Фреге. В кн.: Применение логики в науке и технике. М., 1960, с. 502–555.
- [Осипов, 2002] Осипов Г.С. От ситуационного управления к прикладной семиотике. Новости искусственного интеллекта. 2002, № 6, с. 3–7.
- [Поспелов, 1999] Поспелов Д.А., Осипов Г.С. Прикладная семиотика // Новости ИИ. – 1999. – №1. С. 9–35.
- [Саати, 1993] Саати Т. Принятие решений. Метод анализа иерархий. Пер. с англ. – М.: Радио и связь, 1993. – 320 с.
- [Бирюков, 1960] Бирюков Б. В. Теория смысла Готлоба Фреге. В кн.: Применение логики в науке и технике. М., 1960, с. 502–555.
- [Кулинич, 2013] Кулинич А.А. Концептуальные «каркасы» плохо определенных предметных областей. Открытые семантические технологии проектирования интеллектуальных систем: материалы III Международной научно-технической конференции (Минск, 21–23 февраля 2013 г.)/ под ред. Голенкова В.В. – Минск: БГУИР, 2013, с. 135–142.
- [Холодная, 1997] М.А. Холодная. Психология интеллекта: парадоксы исследования. – Томск: Изд-во Томского университета: Изд-во «Барс». 1997. – 392 с.
- [Langley et al., 2009] Langley P., Laird J.E., Rogers S. Cognitive architectures: Research issues and challenges// Cognitive Systems Research, Volume 10, Issue 2, June 2009, Pages 141–160.

УДК 519.816

ОСОБЕННОСТИ ПОСТРОЕНИЯ ИЕРАРХИЧЕСКОЙ СИСТЕМЫ КРИТЕРИЕВ ДЛЯ ОЦЕНКИ СЛОЖНОГО ТЕХНИЧЕСКОГО КОМПЛЕКСА ¹

В.Н. Лобанов (*Fisher_1980@mail.ru*)
ОАО «НИИВК им. М.А. Карцева», Москва

А.Б. Петровский (*pab@isa.ru*)
Институт системного анализа РАН, Москва

В работе рассмотрены основные особенности агрегирования исходных показателей, описывающих сложный технический комплекс, при построении иерархической системы составных критериев и интегрального показателя для многокритериальной оценки и выбора комплекса.

Введение

Сравнение и выбор сложных комплексов и систем, в частности, технических, представляют собой слабо структурируемую и плохо формализованную задачу. Это обусловлено тем, что такие комплексы характеризуются большим числом исходных показателей, а выбор осуществляется по многим количественным и качественным критериям. Вместе с тем, как правило, вариантов конфигураций бывает немного, и такие варианты оказываются несравнимыми друг с другом по своим показателям, что не позволяет применять для выбора лучшего варианта известные методы принятия решений [Ларичев, 2006; Ногин, 2005; Саати, 1993; Петровский, 2009; Hwang et al., 1987; Roy, 1996].

Одним из подходов к решению задач многокритериального выбора большой размерности служит сокращение пространства признаков, характеризующих объекты выбора. В этом случае исходные признаки агрегируются в небольшое число комплексных критериев или в единственный интегральный показатель, характеризующий качество объекта. Тогда лицу, принимающему решение (ЛПР), гораздо легче провести сравнение разных вариантов, осуществить их

¹ Работа частично поддержана Российским фондом фундаментальных исследований (проект 14-07-00916).

ранжирование и выбрать наиболее предпочтительный. Однако сам процесс построения иерархической системы составных критериев и интегрального показателя достаточно трудоемок и содержит немало подводных камней.

Рассмотрим основные из встречающихся здесь трудностей на примере многокритериальной оценки и выбора малогабаритного вычислительного комплекса персонального уровня (ВКПУ). В настоящее время такие комплексы находят все большее применение при решении различных фундаментальных и прикладных задач и выступают альтернативой дорогостоящих суперкомпьютеров и высокопроизводительных кластеров.

Для выбора вычислительного комплекса была применена технология ПАКС-М, которая является модернизированной версией многоэтапной технологии ПАКС (последовательное агрегирование классифицируемых состояний) [Петровский и др., 2012]. Технология ПАКС-М [Petrovsky et al., 2013], ориентированная на решение задачи выбора несколькими методами, предоставляет ЛПР возможность избежать возникновения противоречий на разных этапах процедуры агрегирования критериев. Сопоставление результатов, полученных несколькими методами для разных систем критериев, позволяет провести анализ итоговых результатов, выбрать наиболее предпочтительную систему критериев и оценить качество сделанного выбора.

1. Выбор исходных показателей

Перечень заданных альтернатив в задаче выбора вычислительного комплекса персонального уровня состоял из трех вариантов: ВК1, ВК2 и ВК3. При постановке задачи выбора желательно включить в число исходных признаков такие показатели, которые характеризуют основные свойства объекта. Первоначально в качестве исходных характеристик ВКПУ были выбраны следующие группы показателей [Петровский и др., 2013], [Лобанов и др., 2013]:

ХМ. Технические характеристики модуля (частота ядра процессора; разрядность ядра процессора; количество потоков; количество ядер процессора; объем поддерживаемой процессором оперативной памяти; количество процессоров в модуле; объем оперативной памяти модуля; наличие ускорителя универсальных

вычислений; дисковая память модуля; наличие в модуле оптического накопителя данных).

ВХ. Вычислительные характеристики комплекса (число модулей в комплексе; скорость обмена между модулями; наличие встроенных средств ввода-вывода; наличие бесперебойного питания; программные характеристики комплекса; возможность модернизации технических и программных средств комплекса).

КХ. Конструкционные характеристики комплекса (размеры комплекса (высота, глубина, ширина); масса комплекса; защищенность от помех).

ЭХ. Эксплуатационные характеристики комплекса (энергопотребление; уровень шума; тепловыделение; условия эксплуатации (температура, влажность); наработка на отказ).

ПК. Производительность комплекса.

СИ. Стоимость изготовления комплекса.

Однако достаточно трудно заранее сказать, какие из характеристик окажутся существенными для выбора объекта. Это становится ясным только после решения задачи. В таком случае ряд исходных признаков можно исключить из рассмотрения, что даст упрощение задачи выбора.

2. Агрегирование исходных показателей

Процедура агрегирования характеристик включает несколько этапов:

- разбиение исходных показателей на группы, в которых критерии близки друг другу по смыслу или являются неотъемлемыми характеристиками какого-либо составного критерия;
- построение иерархического дерева агрегирования показателей на основе сформированных групп критериев;
- формирование порядковых шкал оценок для составных критериев с указанием диапазона оценок исходных показателей для каждой градации на шкале.

Когда объект имеет достаточно много характеристик, которые исчисляются десятками, как в данном случае, объединять сразу все исходные показатели в несколько итоговых критериев или в единственный интегральный критерий достаточно сложно, а порой и просто невозможно. Заранее трудно себе представить, как будет определяться ценность того или иного варианта выбора, который

одновременно характеризуется показателями, отражающими технические, конструкционные, эксплуатационные, экономические и другие свойства объекта. Во-первых, это очень трудоемко, а во-вторых, результаты выбора (если они вообще будут получены) будет сложно объяснить.

При построении дерева агрегирования показателей рекомендуется предварительно определить, какие из показателей могут рассматриваться в качестве итоговых критериев. В нашем случае такими показателями стали производительность комплекса и стоимость изготовления комплекса. Использование СИ и ПК как итоговых критериев связано с их особой ролью. СИ является единственным показателем, характеризующим материальные затраты на приобретение вычислительного комплекса, а ПК обеспечивает сравнение скорости вычислений, выполняемых комплексом.

Остальные исходные показатели комплекса были объединены в три составных критерия ВХ, КХ, ЭХ. Группа показателей «Технические характеристики модуля» была включена в качестве составной части в вычислительные характеристики комплекса. Поэтому, в принципе, можно в исходных показателях вместо всей группы признаков, характеризующих свойства модуля, задать один-единственный показатель или вообще исключить свойства модуля из перечня исходных показателей.

3. Число итоговых критериев

Как известно, человеку, в силу особенностей его физической памяти, легче оперировать небольшой по объему информацией. На практике для этого бывает достаточно от трех до семи показателей.

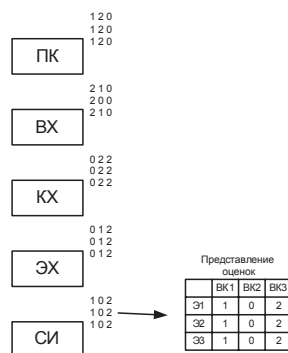
Итоговые критерии в компактной форме передают содержательный смысл исходных характеристик и дают возможность обосновать выбор наиболее предпочтительного варианта. Число итоговых критериев, по которым выполняются сравнение и выбор вариантов, обычно определяется степенью удовлетворенности и пониманием ЛПР полученного результата. Сокращение числа критериев оценки позволяет ЛПР не только решить задачу выбора, но и упрощает анализ полученных результатов. Как правило, число итоговых критериев не должно превышать 3–7. Вместе с тем при

слишком малом числе итоговых критериев и недостаточно продуманном агрегировании показателей на предыдущих шагах могут возникать ситуации, когда все рассматриваемые варианты при их очевидном различии получают одинаковые оценки. Осуществить выбор в таких случаях невозможно.

Многометодная технология ПАКС-М, ориентированная на решение задачи выбора несколькими методами, предоставляет ЛПР возможность избежать возникновения противоречий на разных этапах процедуры агрегирования критериев. Такая возможность обеспечивается благодаря входящему в технологию ПАКС-М алгоритму выявления агрегированного группового предпочтения, реализуемого с помощью нескольких методов группового многокритериального выбора и процедуры голосования. Для этого строятся несколько иерархических систем с разной степенью агрегирования итоговых критериев и с разными схемами их объединения. В рассматриваемом примере все исходные характеристики были сведены в один, три и пять итоговых критериев.

В качестве итоговых были определены следующие пять критериев: ПК. Производительность комплекса; СИ. Стоимость изготовления комплекса; ВХ. Вычислительные характеристики комплекса; КХ. Конструкционные характеристики комплекса; ЭХ. Эксплуатационные характеристики комплекса. Результаты многокритериальной оценки комплексов по этим критериям даны на рисунке 1. Справа на рисунке приведен формат представления оценок по каждому критерию: столбец – оценки соответствующего комплекса, данные тремя экспертами, строка – оценки трех комплексов, данные соответствующим экспертом.

Рис. 1. Оценка комплексов по пяти критериям



4. Иерархические системы агрегирования критериев

Для сравнения и выбора вычислительного комплекса было построено несколько иерархических систем, последовательно агрегирующих близкие друг другу по смыслу исходные характеристики в составные критерии, которые формировались разными способами. Критерии ПК, СИ, ВХ, КХ, ЭХ выступали как итоговые критерии и как промежуточные критерии при построении систем с тремя итоговыми критериями и с единственным интегральным показателем агрегирования.

По первому способу агрегирования критерии ПК и ВХ объединяются в составной критерий «Вычислительный потенциал комплекса» ВП=(ПК, ВХ). Критерии КХ и ЭХ объединяются в составной критерий «Затраты на обслуживание комплекса» ЗО=(КХ, ЭХ). Далее критерии ВП, ЗО и СИ объединяются в единственный интегральный показатель «Категория комплекса» КК=(ВП, ЗО, СИ). Система составных критериев представлена на рисунке 2а.

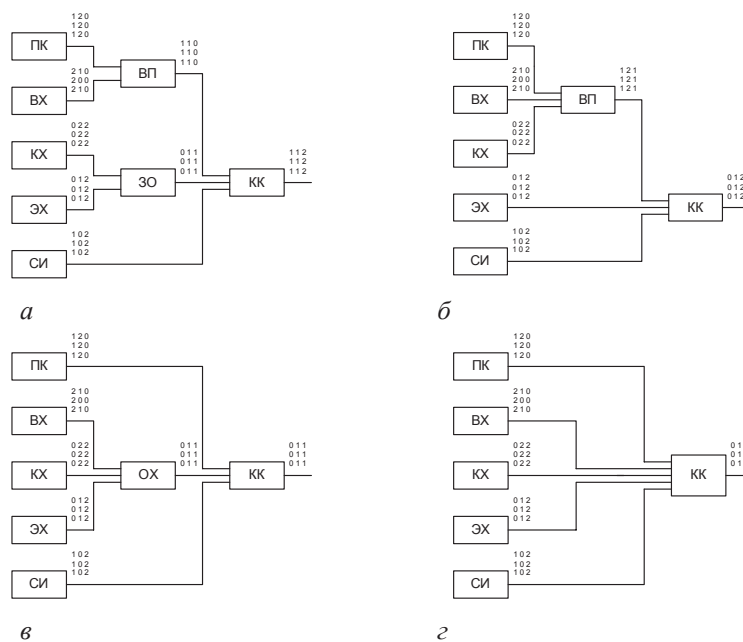


Рис. 2. Способы агрегирования критериев: первый (а), второй (б), третий (в), четвертый (г).

По второму способу критерии ПК, ВХ и КХ объединяются в составной критерий «Вычислительный потенциал комплекса» ВП=(ПК, ВХ, КХ). Далее критерии ВП, ЭХ и СИ объединяются в единственный интегральный показатель «Категория комплекса» КК=(ВП, ЭХ, СИ). Система составных критериев представлена на рисунке 2б.

По третьему способу критерии ВХ, КХ и ЭХ объединяются в составной критерий «Обобщенные характеристики комплекса» ОХ=(ВХ, КХ, ЭХ). Затем критерии ПК, СИ и ОХ объединяются в один интегральный показатель «Категория комплекса» КК=(ПК, СИ, ОХ). Система составных критериев представлена на рисунке 2в.

По четвертому способу все пять критериев сразу объединяются в единый интегральный показатель «Категория комплекса» КК=(ПК, ВХ, КХ, ЭХ, СИ). Система составных критериев представлена на рисунке 2г.

5. Формирование шкал критериев

Для каждого исходного показателя, составного критерия и интегрального показателя формируется своя вербальная шкала оценок с двумя или тремя градациями. Например, производительность комплекса оценивалась как ПК⁰ – высокая (>2000 Гфлопс), ПК¹ – средняя (2000-500 Гфлопс), ПК² – низкая (<500 Гфлопс); стоимость изготовления комплекса как СИ⁰ – высокая, СИ¹ – средняя, СИ² – низкая.

При агрегировании показателей в составные критерии формирование вербальных градаций шкал составных критериев выполняется на основе комбинаций градаций оценок исходных показателей. Поэтому очень важно правильно именовать составной критерий, чтобы его название сочетало в себе свойства объединяемых характеристик и было понятным для ЛПР.

Конструировать каждую градацию шкалы составного критерия можно различными способами, используя разные методы для построения самой шкалы составного критерия, выбирая разные числа градаций и диапазоны изменения переменных на градациях шкал критериев предыдущего или данного уровня иерархии исходя из предпочтений ЛПР или знаний эксперта. Для формирования шкалы оценок каждого составного критерия в технологии ПАКС

применяют различные методы вербального анализа решений, например, ЗАПРОС, ОРКЛАСС, стратификации кортежей, в котором используются однотипные наборы порядковых вербальных шкал исходных показателей [Ларичев, 2006], [Петровский, 2009]. Построение шкалы каждого составного критерия рассматривается как задача порядковой классификации, где в качестве классифицируемых объектов выступают комбинации градаций оценок исходных показателей, а классами являются градации оценок составного критерия. Тем самым каждая комбинация градаций оценок будет соответствовать некоторой градации оценок на шкале комплексного критерия.

В данной работе градация шкалы составного критерия строилась методом стратификации кортежей тремя способами с назначением разных диапазонов изменения градации на шкале. Каждую построенную систему критериев удобно рассматривать как выражение точки зрения некоторого ЛПР/эксперта. Иными словами, можно считать, что при использовании любой системы критериев каждый вариант комплекса оценивался тремя независимыми экспертами.

Единый интегральный показатель, объединяющий в нашем случае все свойства сравниваемых вариантов выбора, называется КК. Категория комплекса и имеет три вербальные градации оценок: $КК^0$ – перспективный комплекс, $КК^1$ – современный комплекс, $КК^2$ – устаревающий комплекс. При агрегировании пяти критериев шкала интегрального показателя формировалась следующим образом. Комплекс считался перспективным ($КК^0$), если сумма градаций оценок критериев, входящих в показатель КК, не превышала значение 4. Комплекс считался современным ($КК^1$), если сумма градаций оценок критериев, входящих в показатель КК, равнялась 5 или 6. Комплекс считался устаревающим ($КК^2$), если сумма градаций оценок критериев, входящих в показатель КК, превышала значение 6.

При агрегировании критериев и построении их шкал рекомендуется объединять небольшое число критериев. В противном случае достаточно сложно сформировать порядковую шкалу составного критерия, поскольку число получаемых сочетаний оценок равно x^n , где x – число исходных градаций, n – число исходных критериев. Так, при четвертом способе агрегирования, где 5 крите-

риев объединяются в единый интегральный показатель категории комплекса, число различных сочетаний исходных градаций оценок равно $3^5=243$ и имеет громоздкое представление.

6. Анализ и объяснение полученных результатов

Выбор наиболее предпочтительного варианта комплекса проводился с помощью трех методов группового многокритериального выбора: метода АРАМИС (агрегирование и ранжирование альтернатив около многопризнаковых идеальных ситуаций), метода лексикографического упорядочивания по градациям критериальных оценок и метода взвешенных сумм рангов. Итоговое обобщенное упорядочение вариантов комплексов строилось при помощи процедуры Борда [Петровский, 2009].

Иерархическая система с итоговыми критериями ПК, СИ, ВХ, КХ, ЭХ дала возможность выбрать наиболее предпочтительный вариант комплекса и объяснить результаты выбора. Итоговое упорядочение вариантов имело вид: $ВК1 > ВК2 \approx ВК3$. Комплекс ВК1 оказался предпочтительнее комплексов ВК2 и ВК3, которые можно считать примерно равноценными.

Полученные результаты можно объяснить следующим образом. Сопоставляя оценки комплексов по пяти критериям, можно отметить, что комплекс ВК1, по мнению экспертов, имеет две высокие (0 градация на шкале) оценки по критериям КХ и ЭХ, две средние (1 градация на шкале) оценки по критериям ПК и СИ, а также одну низкую (2 градация на шкале) оценку по критерию ВХ. Суммарное значение градаций оценок, присвоенных каждым из экспертов, равняется 4. Комплексы ВК2 и ВК3 по совокупности оценок уступают комплексу ВК1 и примерно равноценны.

Сравнение комплексов по интегральному показателю КК, полученному первым способом агрегирования критериев, показало, что $ВК1 \approx ВК2 > ВК3$. Это не вполне согласуется с результатами сравнения комплексов по пяти критериям и связано с тем, что введенный промежуточный критерий ВП уравнил между собой комплексы ВК1 и ВК2, что привело к иному результату. Кроме того, критерии ВП и ЗО усложняют объяснение полученного результата. Система агрегирования критериев, построенная первым способом, не позволяет внятно объяснить результаты.

При сравнении комплексов по показателю КК, полученному вторым способом агрегирования критериев, оказалось, что $ВК1 > ВК2 > ВК3$. Результаты оценки комплексов по показателю КК близки к полученным при оценке комплексов по пяти критериям. По шкале показателя КК комплекс считается перспективным, если общая сумма градаций оценок по промежуточным критериям не превышает значения 2 даже при наличии двух средних оценок или одной низкой оценки по промежуточным критериям. Полученные результаты в общем согласуются с результатами сравнения комплексов по пяти критериям. Однако система агрегирования критериев, построенная вторым способом, также не позволяет внятно объяснить результаты.

При сравнении комплексов по показателю КК, полученному третьим способом агрегирования критериев, оказалось, что $ВК1 > ВК2 \approx ВК3$. Сравнение по критерию ОХ показало, что комплекс ВК1 ненамного опережает комплексы ВК2 и ВК3, а те, в свою очередь, считаются равноценными. Сравнение комплексов по показателю КК дало результаты, согласующиеся с результатами сравнения комплексов по пяти критериям. Одновременно была построена иерархическая система с тремя итоговыми критериями ПК, ОХ, СИ. Результаты сравнения комплексов по этой системе критериев также согласуются с оценками по пяти критериям.

Полученные результаты можно объяснить следующим образом. По составному критерию ОХ комплекс ВК1 получает высокую оценку, так как, несмотря на низкую оценку по вычислительным характеристикам, он имеет лучшие оценки по конструкционным и эксплуатационным характеристикам, в то время как комплексы ВК2 и ВК3 заметно уступают ему по этим параметрам. При агрегировании критериев ПК, СИ и ОХ в единый показатель агрегирования КК общая предпочтительность комплексов не меняется. Система агрегирования критериев, построенная третьим способом, не содержит противоречий и дает понятное объяснение полученных результатов.

При сравнении комплексов по показателю КК, полученному четвертым способом агрегирования критериев, оказалось, что $ВК1 > ВК2 \approx ВК3$, что согласуется с результатами сравнения комплексов по пяти критериям и сравнением по третьей системе агрегирования критериев.

Итак, результаты выбора наиболее предпочтительного варианта комплекса, которые получены по двум из четырех иерархических систем критериев, построенных третьим и четвертым способом, согласуются между собой и с оценками комплексов по пяти критериям и имеют понятные объяснения. Напротив, две иерархические системы критериев с единственным показателем, построенные первым и вторым способом, дают результаты, которые различаются как между собой, так и отличаются от результатов оценки комплексов по пяти критериям. В целом по всем системам агрегирования критериев вычислительный комплекс ВК1 оказался наиболее предпочтительным.

При агрегировании критериев немаловажную роль играет установление семантических связей между исходными показателями и составными критериями. На пути к удовлетворительному решению ЛПР часто приходится сталкиваться с несогласованностью и противоречивостью получаемых результатов. Такие ситуации могут быть вызваны как неудачным объединением составных критериев, так и неудачным назначением градаций шкал составных критериев и интегрального показателя.

Заключение

В работе рассмотрены основные особенности и трудности, связанные с построением интегрального показателя качества сложного технического комплекса. Сформированы несколько иерархических систем с различной степенью агрегирования критериев, результаты выбора по которым согласуются между собой и понятным образом объясняются для ЛПР.

Совпадение результатов для иерархических систем с различной степенью агрегирования критериев дает возможность ЛПР выбрать наиболее удобную систему критериев, либо совместно применять построенные системы критериев с целью повышения обоснованности получаемого решения при выборе наиболее предпочтительного варианта вычислительного комплекса.

Отметим также, что вычислительный комплекс ВК1 оказался наиболее предпочтительным по совокупности своих характеристик по всем построенным системам агрегирования критериев. Это свидетельствует о достаточно высокой надежности предложенного подхода к многокритериальному выбору сложной системы.

СПИСОК ЛИТЕРАТУРЫ

- [**Ларичев, 2006**] Ларичев О.И. Вербальный анализ решений / О.И. Ларичев, под ред. А.Б. Петровского // М.: Наука, 2006.
- [**Лобанов и др., 2013**] Лобанов В.Н., Петровский А.Б. Выбор вычислительного кластера, основанный на агрегировании многих критериев // Вопросы радиоэлектроники, сер. ЭВТ, 2013. – Вып.2. – С. 39–54.
- [**Ногин, 2005**] Ногин В.Д. Принятие решений в многокритериальной среде: количественный подход. – М.: Физматлит, 2005.
- [**Петровский, 2009**] Петровский А.Б. Теория принятия решений // М.: Издательский центр «Академия», 2009.
- [**Петровский и др., 2012**] Петровский А.Б., Ройзензон Г.В. Многокритериальный выбор с уменьшением размерности пространства признаков: многоэтапная технология ПАКС // Искусственный интеллект и принятие решений, 2012. – № 4. – С.88–103.
- [**Петровский и др., 2013**] Петровский А.Б., Лобанов В.Н. Многокритериальный выбор сложной технической системы по агрегированным показателям // Вестник Ростовского государственного университета путей сообщения, 2013. – № 3. – С. 79–85.
- [**Саати, 1993**] Саати Т. Принятие решений. Метод анализа иерархий. – М.: Радио и связь, 1993.
- [**Hwang et al., 1987**] Hwang C.L., Lin M.J. Group Decision Making under Multiple Criteria. – Berlin: Springer-Verlag, 1987.
- [**Petrovsky et al., 2013**] Petrovsky A.B., Lobanov V.N., Zaboleeva-Zotova A.V. Selection of Computing Cluster by Aggregated Complex Criteria // Advances in Decision Technology and Intelligent Information Systems / Ed. by K.J. Engemann, G.E. Lasker. – Tecumseh: The International Institute for Advanced Studies in Systems Research and Cybernetics, 2013. – Vol. XIV. – Pp. 27–31.
- [**Roy, 1996**] Roy B. Multicriteria Methodology for Decision Aiding. – Dordrecht: Kluwer Academic Publishers, 1996.

УДК 004.8

ИЗВЛЕЧЕНИЕ ЗНАНИЙ О ГРУППАХ ВРЕМЕННЫХ РЯДОВ ДЛЯ ЗАДАЧ ПРОГНОЗИРОВАНИЯ¹

И.Г. Перфильева (*irina.perfilieva@osu.cz*)

Университет г. Острава, Чешская Республика

А.А. Романов (*romanov73@gmail.com*)

Ульяновский государственный технический университет, Ульяновск

В статье рассматривается подход к прогнозированию групп временных рядов. Применяется метод F-преобразования высшего порядка для анализа скорости изменения кусочной тенденции временного ряда. Предлагается группировать временные ряды по степени сходства таких изменений. Тогда построение прогноза нейросетевыми методами только одного представителя из группы позволит спрогнозировать поведение остальных похожих временных рядов.

Введение

Ранее примененный подход, базирующийся на методе F-преобразования, для декомпозиции временного ряда позволяет строить адекватные модели временных рядов [Perfilieva, 2013]. На практике примером такой ситуации может служить зависимость группы показателей в экономике от общего фактора.

Если брать во внимание приложение временных рядов к экономике, часто имеются показатели, изменения которых характеризуются большой степенью сходства. Это говорит и о схожем характере поведений локальных тенденций временных рядов за определенный период.

Проводя анализ с целью выявления групп похожих временных рядов показателей, можно выявлять информацию об отношении каждого показателя к значимому фактору. Положительными эффектами от такого анализа являются:

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 13-01-00324). Автор Ирина Перфильева выражает благодарность проекту М.МТ-7026/2012-36 (KОНТАКТ II, LH 12229 «Research and development of methods and means of intelligent analysis of time series for the strategic planing problems») за частичную поддержку при проведении исследований, связанных с тематикой данной статьи.

- Сокращение времени прогнозирования и упрощение прогнозных операций в целом за счет упрощения вычислительных операций.

- Выявление групп похожих процессов в общем массиве с целью получения информации о факторах, группирующих эти процессы.

В работе рассматривается два массива временных рядов. Первый массив – временные ряды статистики, собранной в период с 1970-х по 2000-е годы по РСФСР и России в областях: экономика, производство, социальная сфера, здравоохранение, культура. Вторым массивом служат временные ряды соревнования NN3 (http://www.neural-forecasting-competition.com/downloads/NN3/datasets/NN3_REDUCED.xls).

Определим понятия, которые будут встречаться в тексте. Похожие временные ряды – ряды, имеющие коэффициент корреляции в рамках заданного порога. Группа временных рядов – множество временных рядов, временные ряды которого имеют значение коэффициента совместной корреляции, лежащее в указанных пределах.

1. Построение F-компонент высшего порядка

Для каждого временного ряда задаются функции нечеткого разбиения, строится F-преобразование нулевого порядка (F^0 -преобразование). Для построения F-преобразования первого порядка [Perfilieva, 2011] (F^1 -преобразование): $F_k^1 = c_{k,0}^1 + c_{k,1}^1(x - x_k)$, необходимо определить коэффициенты $c_{k,0}^1, c_{k,1}^1$. В формуле $k=1, \dots, n$, n – количество компонент для ряда. Коэффициенты $c_{k,0}^1$ являются компонентами F^0 -преобразования. Коэффициенты $c_{k,1}^1$ определяются как:

$$c_{k,1}^1 = \frac{\sum_{t=x_{k-1}}^{x_{k+1}} f(t)(t - x_k)A_k(t)}{\sum_{t=x_{k-1}}^{x_{k+1}} (t - x_k)^2 A_k(t)},$$

где $f(t)$ – значения временного ряда в момент времени t , $A_k(t)$ – базисные функции нечеткого разбиения для F^1 -преобразования в момент времени t_k . Коэффициенты $c_{k,1}^1$ равны средневзвешенному значению тангенса угла наклона кусочно-линейного трен-

да временного ряда. Именно они будут служить основой анализа групп временных рядов. Сами компоненты F^1 – преобразования представляют собой векторы и описывают локальные тенденции временного ряда. В качестве примера построения одной из таких компонент приводим график временного ряда №104 из NN3 и компоненту последнего отрезка разбиения F_n^1 :

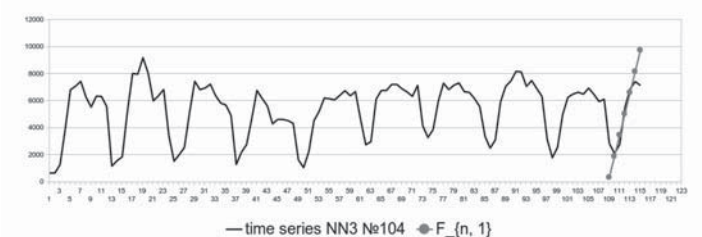


Рис. 1. ВР №104 из NN3 и компонента последнего отрезка разбиения F_n^1

2. Формирование групп временных рядов

Группирование временных рядов будем производить, анализируя коэффициенты $C_{k,1}^1$. Для каждой пары временных рядов вычисляется коэффициент корреляции Пирсона (линейный коэффициент корреляции):

$$r_{f_1(x) f_2(x)} = \frac{\sum_{i=1}^m (f_1(x_i) - \overline{f_1(x)}) (f_2(x_i) - \overline{f_2(x)})}{\sqrt{\sum_{i=1}^m (f_1(x_i) - \overline{f_1(x)})^2 \sum_{i=1}^m (f_2(x_i) - \overline{f_2(x)})^2}}$$

где $f_1(x), f_2(x)$ – значения коэффициентов $C_{k,1}^1$ для первого и второго временного ряда из пары соответственно.

Важным условием для проведения анализа является равная длина временных рядов. Перед построением компонент F-преобразования различного порядка производится нормировка значений временного ряда к интервалу $[0, \dots, 1]$ по модулю. Нормировка необходима для обеспечения одинаковой области значений коэффициентов $C_{k,1}^1$. При формировании групп временных рядов возникают следующие проблемы.

Необходимо правильно выбрать количество точек, которое покрывает базисная функция – данная величина непосредственно

влияет на последующую корреляцию временных рядов по коэффициенту $C_{k,l}^1$. Чем большее количество точек будет покрывать базисная функция, тем более сглаженным будет ряд, и корреляция будет рассчитываться по общим тенденциям рядов.

Задание порога влияет на количество похожих по коэффициенту корреляции временных рядов. Важно определить, что более важно: включить в группу больше рядов, или отобрать только самые близкие по корреляции ряды.

3. Вычислительные эксперименты

Рассмотрим два эксперимента. Первый эксперимент выполнен на множестве временных рядов статистики, обозначенном выше. Были отобраны временные ряды, имеющие одинаковую длину (в конкретном случае было выбрано 14 рядов длиной 37 точек). Заранее известно, что ряды в этом множестве обладают схожим характером поведения. Второй эксперимент был проведен на временных рядах соревнования NN3 (количество рядов равно 8, длина ряда – 126 точек). Количество точек, которое покрывает базисная функция, было выбрано равным 7, чтобы компоненты F-преобразования сглаживали резкие выпадения временных рядов.

Если выделить группы временных рядов при коэффициенте корреляции > 0.9 , то получим следующий результат:

Таблица 1

Формирование групп из рядов статистики

Ряд	Количество похожих рядов	Ряд	Количество похожих рядов
1	10	8	8
2	10	9	9
3	7	10	8
4	2	11	9
5	4	12	8
6	1	13	10
7	10	14	8

Один из вариантов формирования группы по данным результатам – выделить ряды, имеющие максимальное количество похожих и включить в список все участвующие ряды. Например, можно выделить следующую группу рядов: 1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14.

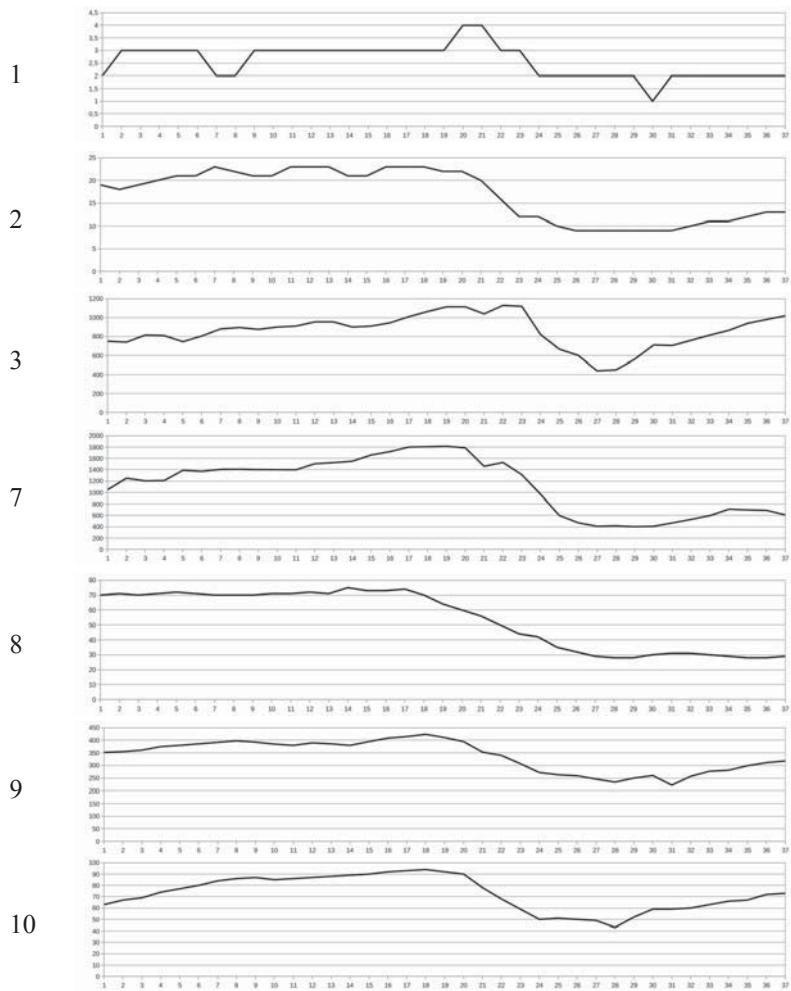
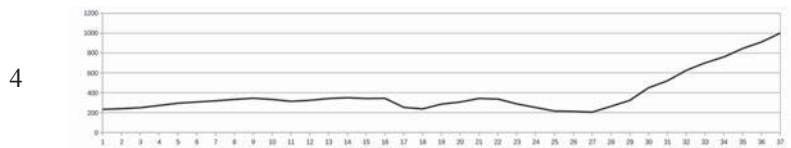


Рис. 2. Представители рядов статистики из сформированной группы

Не вошли в группу похожих рядов:



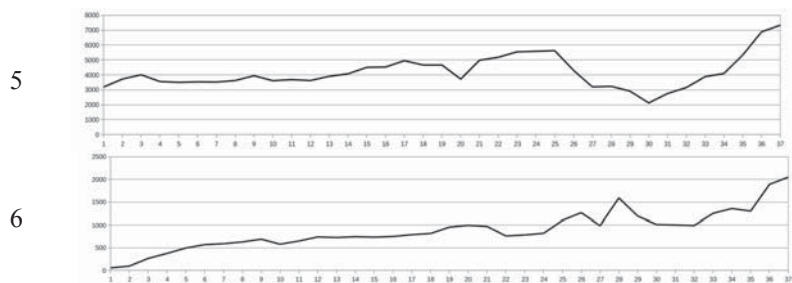


Рис. 3. Представители рядов статистики, не вошедшие в группу

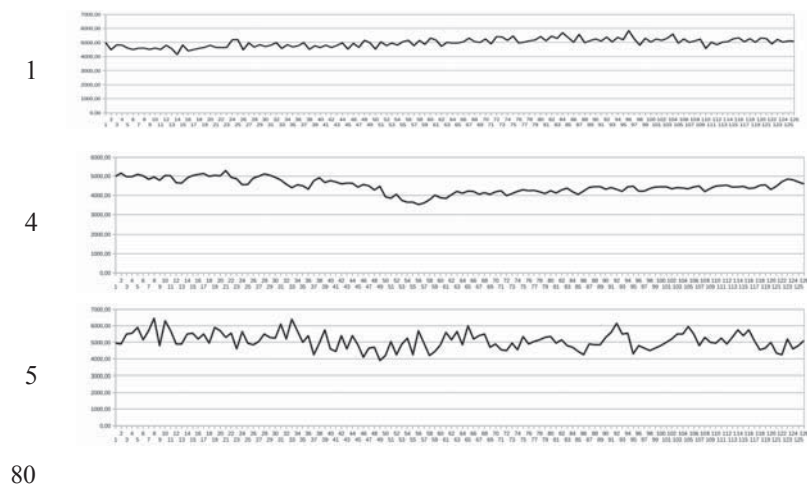
Если выделить группы временных рядов из банка данных NN3 при коэффициенте корреляции > 0.9 , то получим следующий результат:

Таблица 2

Формирование групп из рядов NN3

Ряд	Количество похожих рядов
1	3
4	3
5	3
6	3

В группу похожих временных рядов входят: 1, 4, 5, 6 ряды.



6

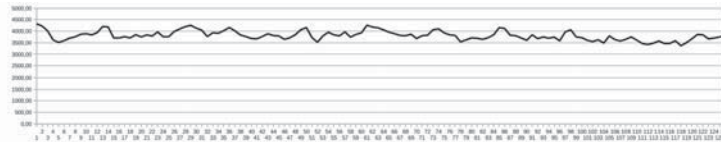
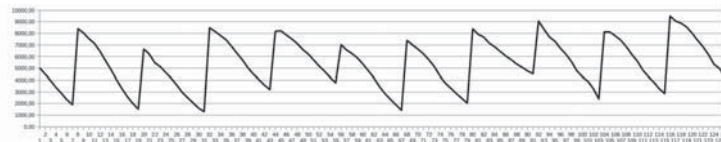


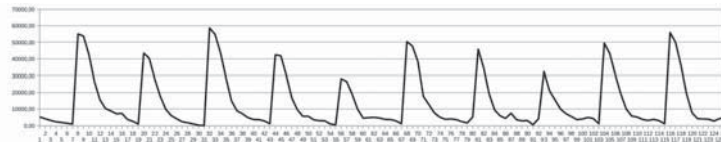
Рис. 4. Представители рядов NN3 из сформированной группы

Не похожие временные ряды:

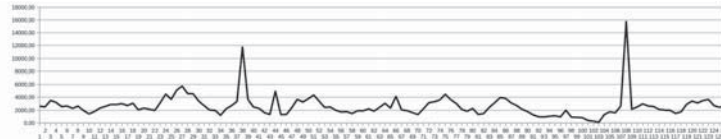
2



3



7



8

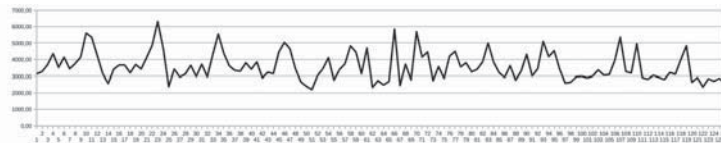


Рис. 5. Представители рядов NN3, не вошедшие в группу

Затем для одного из представителей группы строится прогноз коэффициента $C_{k,1}^1$. Для примера возьмем группу рядов из статистических рядов. Спрогнозируем коэффициент $C_{k,1}^1$ для ряда №1 из выделенной группы. Теперь можно построить прогноз для других временных рядов группы. В качестве примера построим прогноз для одной компоненты F_n^1 второго и третьего ряда:

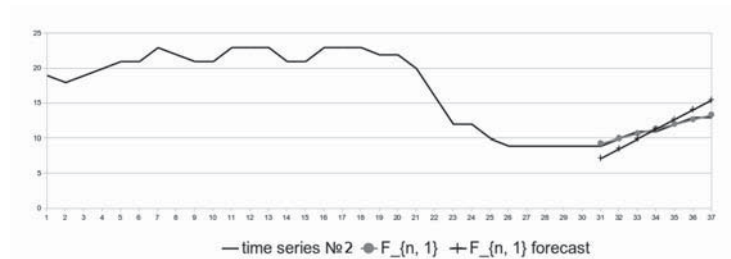


Рис. 6. Пример прогнозирования локальной тенденции ряда статистики

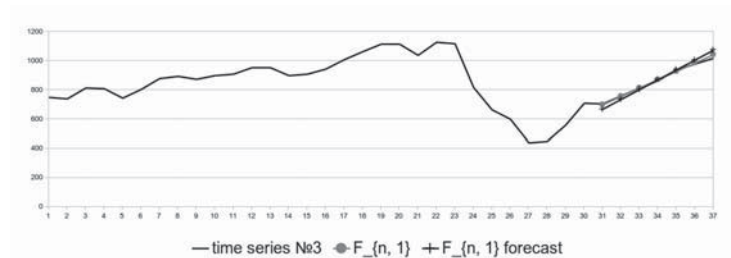


Рис. 7. Пример прогнозирования локальной тенденции ряда статистики

В этом примере использовались существующие коэффициенты $C_{k,0}^1$ и прогнозное значение $C_{k,1}^1$ первого ряда группы. Но даже в том случае, если прогнозировать коэффициент $C_{k,0}^1$, то локальная тенденция сохраняет свое направление.

Заключение

Был рассмотрен подход использования компонент F-преобразования высших порядков, а именно нулевого и первого, для решения задачи группировки и прогнозирования временных рядов. Эксперименты показали, что использование коэффициентов F¹-преобразования позволяет объединять временные ряды, обладающие схожими локальными тенденциями. В то же время сформированные группы временных рядов позволяют упростить построение прогноза за счет сокращения вычислительных операций и идеи использования спрогнозированных локальных тенденций одного ряда для построения прогноза других рядов группы.

СПИСОК ЛИТЕРАТУРЫ

[Perfilieva, 2011] I. Perfilieva, M. Dankova, and B. Bede. Towards a Higher Degree F-Transform // Fuzzy Sets and Systems. 2011. Vol. 180. Pp. 319.

[Perfilieva, 2013] I. Perfilieva, N. Yarushkina, T. Afanasieva, and A. Romanov, «Time series analysis by soft computing methods,» Int. Journ. of General Systems, vol. 42, pp. 687–705, 2013.

[Афанасьева, 2011] Афанасьева Т.В., Ярушкина Н. Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения. – 2011. – №3. – С. 7–16.

[Ярушкина и др., 2007а] Ярушкина Н. Г., Вельмисов А. П., Стецко А. А. Средства Data Mining для нечетких реляционных серверов данных // Информационные технологии. – 2007. – №6. – С. 20–29.

[Ярушкина и др, 2007б] Ярушкина Н. Г., Афанасьева Т.В. Нечеткие временные ряды как инструмент для оценки и измерения динамики процессов // Датчики и системы. – 2007. – №12. – С. 46–50.

УДК 351.854

МНОГОКРИТЕРИАЛЬНЫЙ ПОДХОД В ЗАДАЧЕ РАСПРЕДЕЛЕНИЯ РЕСУРСОВ

Е.Я. Ройтман (*reja41@yandex.ru*)

ОАО «Геопром», Москва

Г.В. Ройзензон (*rgv@isa.ru*)

Институт системного анализа РАН, Москва

В работе рассматривается решение задачи оптимального распределения ресурсов (схемы производства) для предприятий цветной металлургии. В представленном подходе задействованы методы многокритериального линейного программирования и методы группового вербального анализа решений. Предлагается постановка задачи многокритериальной оптимизации для формирования парето-оптимального набора альтернатив и применение метода группового ранжирования для выбора наилучшей многокритериальной альтернативы. Сформирован набор критериев оценки для выбора схемы производства. Представлен пример решения практической задачи оптимального распределения ресурсов между предприятиями цветной металлургии. Предложены направления использования представленного подхода для решения целого ряда актуальных задач в цветной металлургии.

Введение

Производство в цветной металлургии характеризуется многопередельностью и многономенклатурностью. Этот аспект предполагает возможность изготовления одной и той же продукции по альтернативным технологическим схемам, характеризующимся использованием разного оборудования и ресурсов (сырья, топлива, энергии). Очевидно, что при этом будут разные издержки производства. Это предопределяет эффективность использования экономико-математических методов (ЭММ). В 40-80-е годы прошлого века был накоплен достаточно большой опыт использования ЭММ в основных отраслях, в том числе и в цветной металлургии [Канторович, 1942], [Рытиков и др., 1984], [Шевакин и др., 1977], [Ларичев и др., 1986], [Канторович, 1972], [Бурков и др., 1970], [Проценко и др., 1979]. Наиболее широко они использовались на прокатном переделе [Рытиков и др., 1984], [Шевакин и др., 1977].

После приватизации как в черной, так и в цветной металлургии возникло достаточно большое количество крупных акционерных обществ. Многие из них включают в себя несколько предприятий, в том числе выпускающих аналогичную продукцию, например ТМК, ЕВРАЗ групп в черной металлургии, УГМК, РУСАЛ в цветной металлургии. В условиях рыночной конкуренции минимизация издержек является одной из актуальных задач эффективного управления, в том числе крупными холдингами. Важность задачи возрастает в связи с вступлением России в ВТО и ростом конкуренции на мировом рынке. Вступление России в ВТО диктует необходимость внедрения новых методов планирования и управления, в которых будут использованы современные методы принятия решений [Ларичев, 2006], [Петровский, 2009], [Ройзензон, 2012], [Ройтман и др., 2012], [Лившиц и др., 2011], [Власов и др., 2011] и более гибкие подходы к ценообразованию [Ройтман, 2006].

Наиболее широкое применение при решении задач планирования и управления нашел метод линейного программирования, разработанный Л.В. Канторовичем [Канторович и др., 1972], получивший самое широкое применение как в отечественной, так и в зарубежной практике, в том числе и при решении различных задач в металлургии [Канторович, 1942], [Канторович, 1972]. Опыт применения метода линейного программирования показывает, что его использование эффективно только при недостатке того или иного ресурса, именно тогда вступает в действие механизм поиска оптимума из числа возможных вариантов.

В связи с резким падением объемов производства в 1990-е годы на многих предприятиях цветной металлургии имеется избыток производственных мощностей и решение классической задачи на полную мощность производства может оказаться неактуальным. Но многие предприятия переходят на двух- и односменный режим работы и в этом случае она может представлять интерес. Среди актуальных задач для холдинга, управляющего несколькими предприятиями, могут быть:

- сокращение номенклатуры и определение рационального ассортимента предприятий и / или цехов;
- определение оптимального ассортимента сохраняемых предприятий при ликвидации одного из предприятий холдинга;
- перераспределение ассортимента при вводе новых мощностей или реконструкции цехов, заводов;

- оценка возможности увеличения выпуска продукции, пользующейся повышенным спросом за счет использования более рациональных технологий.

Опыт решения подобных задач показывает, что получаемые при этом результаты требуют постоянного экспертного анализа с целью оценки реализуемости предлагаемых решений. Это связано с тем, что не всегда альтернативные технологии, закладываемые в расчет, являются полностью взаимозаменяемыми на том или ином производстве. Требуется проведение многократных вариантных расчетов с корректировкой тех или иных исходных данных для получения удовлетворительного результата. Это подтверждает рассматриваемый далее пример.

Необходимо отметить, что как в советское время, так и сейчас при решении задачи рационального распределения ресурсов в цветной металлургии недостаточное внимание уделяется вопросам экологии [Ларичев и др., 1986], а также учету различных социальных процессов (например, при сокращении рабочих мест на предприятиях или ликвидации системообразующих для того или иного региона (города) предприятий).

Таким образом, при выборе той или иной схемы производства в цветной металлургии необходимо одновременно рассматривать наряду с хорошо «рассчитываемыми» факторами, такими как «трудоемкость производства проката» и «себестоимость производства проката», также экологические аспекты и вопросы социальной значимости того или иного предприятия. Последние два аспекта достаточно сложно оценивать количественно, т.к. экологические проблемы того или иного региона могут быть связаны не только с производством цветных металлов. Аналогичная ситуация складывается и при оценке социальной напряженности в том или ином регионе. Подобные трудноформализуемые аспекты могут быть учтены путем экспертного оценивания.

С учетом вышеизложенного задача оптимального распределения ресурсов в цветной металлургии может быть решена в несколько этапов. На первом этапе решается оптимизационная задача, в результате которой формируется небольшой (около 10) набор вариантов. На втором этапе эти варианты необходимо оценить с точки зрения уровня качества продукции, потенциального экологического ущерба и социальной значимости. И, наконец, на последнем

этапе осуществляется окончательный многокритериальный (групповой [Петровский, 2009], [Ройзензон, 2012] или индивидуальный [Ларичев, 2006]) выбор наилучшего варианта с активным участием в этом процессе экспертного сообщества.

1. Постановка задачи

Формулировка задачи: необходимо изготовить продукцию различных типоразмеров j ($j = 1, \dots, n$) на заводе k ($k = 1, \dots, f$) холдинга в количестве каждого размера Q_{jk} . При этом на каждом заводе можно применять технологические схемы i ($i = 1, \dots, m$) и оборудование – агрегаты r ($r = 1, \dots, l$) [Рытиков и др., 1984].

Интенсивность использования того или иного варианта изготовления продукции типа j на заводе k определяется количеством продукции x_{jki} , производимой по данному варианту. Известно, что затраты на изготовление единицы продукции j по варианту i на заводе k характеризуются показателем C_{jki} (это может быть себестоимость продукции, трудоемкость в станко-часах, человеко-часах и т.п.). Требуется, чтобы сумма затрат отвечала условию

$$\sum_{j=1}^n \sum_{k=1}^f \sum_{i=1}^m C_{jki} x_{jki} = C(\min).$$

Задача решается при следующих уравнениях-ограничениях:

1. Общий выпуск каждого профилеразмера должен соответствовать плановому заданию по холдингу Q_j :

$$\sum_{k=1}^f \sum_{i=1}^m x_{jki} = Q_j.$$

2. Оборудование каждого вида может быть использовано только в пределах имеющегося фонда станко-часов:

$$\sum_{i=1}^m \sum_{j=1}^n t_{jki} x_{jki} \leq T_{rk},$$

где t_{jki} – норма времени изготовления продукции вида j на заводе k по варианту i на агрегате r с учетом коэффициента выполнения норм на данном агрегате; T_{rk} – фонд времени работы агрегата r , установленного на заводе k .

3. Выпуск отдельных профилеразмеров j на конкретных заводах k должен быть не менее установленных плановых заданий:

$$\sum_{i=1}^m x'_{jki} \geq Q'_{jk}; \text{ при этом } \sum_{k=1}^f Q'_{jk} \leq Q_j,$$

т.е. распределенные по заводам плановые задания Q'_{jk} не должны превосходить общего планового задания по продукции j в холдинге.

4. Потребность в рабочей силе для выполнения плана производства не должна превосходить планового значения:

$$\frac{1}{T} \sum_{j=1}^n \sum_{i=1}^m x_{jki} \tau_{jki} \leq C_k,$$

где τ_{jki} – суммарные затраты человеко-часов на изготовление единицы продукции вида j на заводе k по технологической схеме i ; C_k – плановая потребность в рабочей силе на заводе k ; T – время работы одного рабочего в планируемом периоде.

5. План производства в натуральном и стоимостном выражении должен отвечать условиям

$$\sum_{j=1}^n \sum_{i=1}^m x_{jki} \geq Q_k,$$

где Q_k – план производства завода k в натуральном выражении:

$$\sum_{j=1}^n \sum_{i=1}^m x_{jki} C_j \geq ТП_k$$

где C_j – цена изделия j ; $ТП_k$ – план производства завода k в стоимостном выражении (товарная продукция).

6. Неотрицательность переменных x_{jki} : $x_{jki} \geq 0$.

Задача сводится к отысканию значений x_{jki} , при которых функция цели имеет минимальное значение. Задача относится к классу задач линейного программирования и решается симплекс-методом. В результате решения задачи для заданной функции цели будет определено: на каких заводах, в каком количестве и по каким схемам целесообразно изготавливать продукцию. Полученные результаты используются для расчета загрузки оборудования, определения потребности в рабочей силе, объема товарной продукции, размера прибыли, потребности в заготовке.

В табл. 1 приведены расчеты для холдинга, в состав которого входит три завода.

Таблица 1

Показатели	Базовый вариант			Минимум себестоим.			Минимум чел.-часов		
	Завод 1	Завод 2	Завод 3	Завод 1	Завод 2	Завод 3	Завод 1	Завод 2	Завод 3
План в тоннах	18000	8000	10000	18754,45	5700	11545,55	16338,37	9651,63	10010
Тов. продукц., тыс. руб.	4916500	2465300	2749000	5025571	1788300	3495586	4406895	3004505	2698400
Прибыль, тыс. руб.	899617,9	217013,1	310538	917340	154821	602060	732021	239763	234588
Рентаб., %	22,4	9,65	12,73	22,32	9,48	20,80	20	8,67	9,52
Погреб. в раб. силе	113	75	113	102	79	140	62	69	65
Производ., т/чел.	159,29	106,67	88,50	183,87	72,15	82,47	263,52	139,88	154,00
Производ., тыс. руб./чел.	43508,85	32870,67	24327,43	49270,3	22636,71	24968,47	71078,95	43543,55	41513,85
Число групп размер.	8	8	8	8	6	6	7	6	4
Трудоёмкость, чел-час/т	10,33	15,47	18,72	8,95	22,88	20,03	6,23	11,87	10,76

2. Формирование системы критериев

С учетом вышеизложенного схемы производства можно охарактеризовать четырьмя группами составных критериев [Ройзензон, 2005]: экономические аспекты (план в тоннах; товарная продукция (тыс. руб.); прибыль (тыс. руб.); рентабельность (%); потребность в рабочей силе и др.), качество продукции, социальные аспекты (рост безработицы, уровень социальной напряженности и т.п.) и экологические аспекты. Таким образом, выбор (индивидуальный или групповой) [Ларичев, 2006], [Петровский, 2009], [Ройзензон, 2012] по четырем составным критериям той или иной схемы производства может, например, быть весьма привлекательным с чисто экономической точки зрения, но при этом создавать различные социальные проблемы и ухудшать экологическую обстановку в том или ином регионе, где базируются заводы холдинга.

Предлагаемый подход многокритериального выбора с использованием составных критериев ранее был успешно применен для ряда практических задач: выбора сложной технической системы [Ройзензон, 2004], оценки результативности научных проектов [Петровский и др., 2008], [Петровский и др., 2010], [Ройзензон, 2012] и анализа деятельности научных организаций [Петровский и др., 2012].

Заключение

В работе предложено решение задачи оптимального распределения ресурсов (схемы производства) для предприятий цветной металлургии. В представленном подходе используются методы многокритериального линейного программирования и методы группового вербального анализа решений. Предложены постановка задачи многокритериальной оптимизации для формирования парето-оптимального набора альтернатив и применение метода группового ранжирования для выбора наилучшей многокритериальной альтернативы. Сформирован набор критериев оценки для выбора схемы производства. Представлен пример решения практической задачи оптимального распределения ресурсов между предприятиями цветной металлургии. Предложены направления использования предлагаемого подхода для решения целого ряда актуальных задач в цветной металлургии.

СПИСОК ЛИТЕРАТУРЫ

[Бурков и др., 1970] Бурков В.Н., Кацнельсон М.Б. Оперативное управление поставками металлопродукции. Решение некоторых проблем оптимизации // Автоматика и телемеханика. – 1970. – № 1. – С. 148–158.

[Власов и др., 2011] Власов С.А., Генкин А.Л., Никулина И.В., Кузнецов И.А., Кравцов С.В. Имитационные приложения для систем управления металлургическими предприятиями // Четвертая Международная конференция «Системный анализ и информационные технологии» (САИТ-2011). Труды конференции в 2 т. – Т. 2. – Челябинск: Издательство ЧГУ, 2011. – С. 9–13.

[Ларичев, 2006] Ларичев О.И. Вербальный анализ решений / Под ред. А.Б. Петровского. – М.: Наука, 2006. – 181 с.

[Ларичев и др., 1986] Ларичев О.И., Бебчук Б.Ц. Человеко-машинные процедуры решения многокритериальных задач линейного программирования с простыми иерархическими моделями // Системы и методы поддержки принятия решений. Сборник трудов / Под ред. С. В. Емельянова, О. И. Ларичева. – № 12. М.: ВНИИСИ, 1986. – С. 100–108.

[Лившиц и др., 2011] Лившиц В.Н., Лившиц С.В. Системный анализ нестационарной экономики России (1992–2010): рыночные реформы, кризис, инвестиционная политика. – М.: Маросейка, 2011. – 478 с.

[Канторович, 1942] Канторович Л.В. Рациональные методы раскроя металла – Произв.-техн. бюлл. НК боеприпасов СССР, 1942. – № 7–8. – С. 21–29.

[Канторович и др., 1972] Канторович Л.В., Горстко А.Б. Оптимальные решения в экономике. – М.: Наука, 1972. – 232 с.

[Канторович, 1972] Канторович Л.В. О математическом обеспечении АСУ «Металл» // Приборы и системы управления. – 1972. – № 12. – С. 8–10.

[Петровский и др., 2008] Петровский А.Б., Ройзензон Г.В. Снижение размерности признакового пространства в задачах многокритериальной классификации: стратификация кортежей // Одиннадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2008). Труды конференции. – Т. 2. – М.: Ленанд, 2008. – С. 262–270.

[Петровский, 2009] Петровский А.Б. Теория принятия решений. – М.: Издательский центр «Академия», 2009. – 400 с.

[Петровский и др., 2010] Групповое упорядочивание научных проектов по несогласованным многокритериальным оценкам / А.Б. Петровский, Г.В. Ройзензон, И.П. Тихонов, А.В. Балышев // Двенадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2010). Труды конференции. – Т. 3. – М.: Физматлит, 2010. – С. 201–207.

[Петровский и др., 2012] Многокритериальный анализ деятельности научных организаций / А.Б. Петровский, Г.В. Ройзензон, И.П. Тихонов и др. // Тринадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2012). Труды конференции. – Т. 3. – Белгород: Издательство БГТУ, 2012. – С. 64–71.

[Проценко и др., 1979] Проценко О.Д., Соловейчик Д.И., Мольт Л.И., Тихонравов А.Л. Формирование планов производства предприятий цветной металлургии при многоцелевой оптимизации // Цветные металлы. – 1979. № 7. – С. 73–77.

[Ройзензон, 2004] Ройзензон Г.В. Формирование критериев оценки для выбора сложной технической системы // Труды девятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2004). – Т. 2. – М.: Физматлит, 2004. – С. 861–868.

[Ройзензон, 2005] Ройзензон Г. В. Способы снижения размерности признакового пространства для описания сложных систем в задачах принятия решений // Новости искусственного интеллекта. – 2005. – № 1. – С. 18–28.

[Ройзензон, 2012] Ройзензон Г.В. Синергетический эффект в принятии решений // Системные исследования. Методологические проблемы. Ежегодник / Под ред. Ю.С. Попкова, В.Н. Садовского, В.И. Тищенко. – № 36. 2011–2012. М.: УРСС, 2012. – С. 248–272.

[Ройтман, 2006] Ройтман Е.Я. Проблемы ценообразования на прокат тяжелых цветных металлов // Цветные металлы. – 2006. – № 6. – С. 5–7.

[Ройтман и др., 2012] Ройтман Е.Я., Ройзензон Г.В. Принятие оптимальных решений в цветной металлургии // IX Всероссийская школа-семинар «Прикладные проблемы управления макросистемами». Материалы докладов. – Апатиты: КНЦ РАН, 2012. – С. 56–58.

[Рытиков и др., 1984] Рытиков А.М., Ройтман Е.Я., Тулинова Н.Д. Применение экономико-математических методов для решения задачи специализации заводов ОЦМ // АСУ, экономика, планирование и оптимизация производства полуфабрикатов из цветных металлов и сплавов. Сборник научных трудов. – М.: Металлургия, 1984. – С. 4–8.

[Шевакин и др., 1977] Шевакин Ю.Ф., Рытиков А.М. Вычислительные машины в производстве труб. М.: Металлургия, 1977. – 240 с.

УДК 007.52:681.518.2;681.327.12

ИНТЕЛЛЕКТУАЛЬНАЯ ПОДСИСТЕМА ПОСТРОЕНИЯ ОТКАЗОУСТОЙЧИВЫХ БЕЗУСЛОВНЫХ ДИАГНОСТИЧЕСКИХ ТЕСТОВ И ПРИНЯТИЯ РЕШЕНИЙ НА ИХ ОСНОВЕ¹

А.Е. Янковская (*ayyankov@gmail.com*)

Томский государственный архитектурно-строительный университет,
Национальный исследовательский Томский государственный
университет, Томский государственный университет
систем управления и радиоэлектроники, Томск

А.И. Гедике (*gai@tsuab.ru*)

Томский государственный архитектурно-строительный университет,
Томск

Приводятся краткое изложение модели представления данных и знаний, математические основы построения отказоустойчивых безусловных диагностических тестов (ОУБДТ), архитектура и принципы программной реализации интеллектуальной подсистемы построения ОУБДТ и принятия решений на их основе.

Введение

Учет ошибок измерения и/или занесения значений признаков, задающих описания исследуемых объектов, является весьма актуальной задачей при разработке интеллектуальных систем (ИС) для различных проблемных и междисциплинарных областей, поскольку такие ошибки существенно влияют на качество и достоверность принимаемых решений.

Предпосылками для повышения надежности при принятии решений в интеллектуальных распознающих системах послужил математический аппарат, основанный на теории разделяющих систем, используемый при синтезе и диагностике асинхронных автоматов и наиболее полно представленный в монографии [Сагалович, 2012], а также подход к построению отказоустойчивых решающих правил [Янковская, 2009].

¹ Работа выполнена при финансовой поддержке РФФИ (проекты № 13-07-00373, № 14-07-00673 и частично № 13-07-98037) и РГНФ (частично проект № 13-06-00709).

Следует отметить, что задача выбора оптимального количества признаков при наличии ошибок измерений применительно к медицинской диагностике впервые была поставлена А.Е. Янковской в 1983 г.

Существуют различные подходы к созданию ИС, например, алгебраический подход, предложенный Ю.И. Журавлёвым [Журавлёв, 1990], включающий тестовые методы распознавания образов и успешно развиваемый как его учениками, так и другими учеными. К сожалению, нам неизвестны публикации по методам, посвященным построению и применению отказоустойчивых диагностических тестов (ОУБДТ).

Предлагаемый нами подход к созданию прикладных ИС для конкретных проблемных и междисциплинарных областей базируется на матричной модели представления данных и знаний, выявлении по обучающей выборке различного рода закономерностей, построении решающих правил на основе выявленных закономерностей, применении логико-комбинаторных методов распознавания образов для принятия решений относительно исследуемых объектов [Янковская, 2011].

В рамках предлагаемого подхода поиск подмножества признаков, устойчивых к задаваемому пользователем количеству ошибок, сводится к построению множества ОУБДТ [Янковская, 2009; Гедике и др., 2011; Yankovskaya et al., 2012].

Создание подсистемы построения ОУБДТ и принятия решений на их основе предлагается осуществлять на базе интеллектуального инструментального средства (ИИС) ИМСЛОГ, являющегося интегрированной средой для конструирования прикладных интеллектуальных систем [Yankovskaya et al., 2003].

1. Модель представления данных и знаний

Будем считать, что все объекты рассматриваемой проблемной области можно разбить на конечное число образов, которым соответствуют различные итоговые решения, и построение ОУБДТ будем осуществлять на основе анализа обучающей выборки, реализованной с использованием матричной модели представления данных и знаний, включающей матрицы описаний (\mathbf{Q}) и различий (\mathbf{R}) [Янковская, 2011].

Строки матрицы \mathbf{Q} сопоставляются обучающим объектам s_i ($i = \overline{1, N}$, где N – количество объектов), столбцы – характеристиче-

ским признакам z_j ($j = \overline{1, M}$, где M – количество признаков, задающих описание каждого объекта). Элемент $q_{i,j}$ матрицы \mathbf{Q} задает значение j -го признака для i -го объекта из множества значений, равного $\{1, 0, -\}$. Единица означает, что j -й признак присущ i -му объекту, ноль – не присущ, прочерк – значение признака безразлично.

Строки матрицы \mathbf{R} сопоставляются строкам матрицы \mathbf{Q} , столбцы – целочисленным классификационным признакам k_j ($j = \overline{1, L}$, где L – число механизмов классификации, разбивающих объекты из рассматриваемой проблемной области на классы эквивалентности). Элемент $r_{i,j}$ матрицы \mathbf{R} задает принадлежность i -го объекта некоторому классу по j -му механизму классификации путем указания номера класса.

Считается, что объекты, имеющие одинаковую комбинацию значений признаков k_j , соответствующую некоторому решению, принадлежат одному и тому же образу. Это означает, что число неповторяющихся строк матрицы \mathbf{R} равно числу образов и каждое подмножество строк матрицы \mathbf{Q} , которым соответствуют одинаковые строки, задает описание некоторого образа. Можно каждому образу сопоставить его условный номер и задавать принадлежность объектов образам (грубее) одним столбцом.

Предлагаемая модель позволяет представлять не только данные, но и знания экспертов, поскольку одной строкой матрицы \mathbf{Q} можно задавать в интервальной форме (используя значения «прочерк») подмножество объектов, для которых характерно одно и то же решение, задаваемое соответствующей строкой матрицы \mathbf{R} .

Будем считать, что описания обучающих объектов, представленные строками матриц \mathbf{Q} и \mathbf{R} , не содержат ошибочных значений.

2. Построение ОУБДТ и принятие решений на их основе

Одним из подходов к построению ОУБДТ является построение по матрицам \mathbf{Q} и \mathbf{R} безызбыточной булевой матрицы импликаций \mathbf{U}' , задающей условия различимости обучающих объектов, принадлежащих разным образам [Янковская, 2011].

Столбцы матрицы \mathbf{U}' сопоставляются характеристическим признакам, строки – результатам попарного сравнения объектов, принадлежащих разным образам. Элемент строки матрицы \mathbf{U}' принимает единичное значение, если соответствующий признак является различающим для данной пары, иначе – нулевое значение. Матрица

U' задает достаточные условия различимости любой пары обучающих объектов из разных образов, описания которых представлены матрицами Q и R .

Для построения ОУБДТ и принятия решений, устойчивых к t ошибочным значениям признаков в описании исследуемого объекта, необходимо и достаточно, чтобы каждая строка матрицы U' содержала не менее h ($h = 2t + 1$) единичных значений [Янковская, 2009; Yankovskaya et al., 2012]. Если данное условие выполняется, то задача построения минимальных (содержащих минимальное число признаков) ОУБДТ сводится к задаче нахождения кратчайших h -кратных столбцовых покрытий матрицы U' . Отметим, что построение безыбыточных (в терминологии Ю.И. Журавлёва – тупиковых) ОУБДТ сводится к поиску безыбыточных h -кратных столбцовых покрытий матрицы U' .

С целью сокращения перебора целесообразно реализовать поиск кратчайших h -кратных столбцовых покрытий в процессе построения дерева поиска с использованием ряда правил, определяющих очередность выбора строк матрицы U' и выбора признаков в строках, а также позволяющих отсекалть ветви, не приводящие к искомому покрытию.

Алгоритм построения минимальных ОУБДТ (включая построение матрицы U' и поиск ее кратчайших h -кратных столбцовых покрытий) приведен в публикации [Гедике и др., 2011].

Управляющие и информационные взаимосвязи между блоками создаваемой подсистемы при построении ОУБДТ представлены на рис. 1.

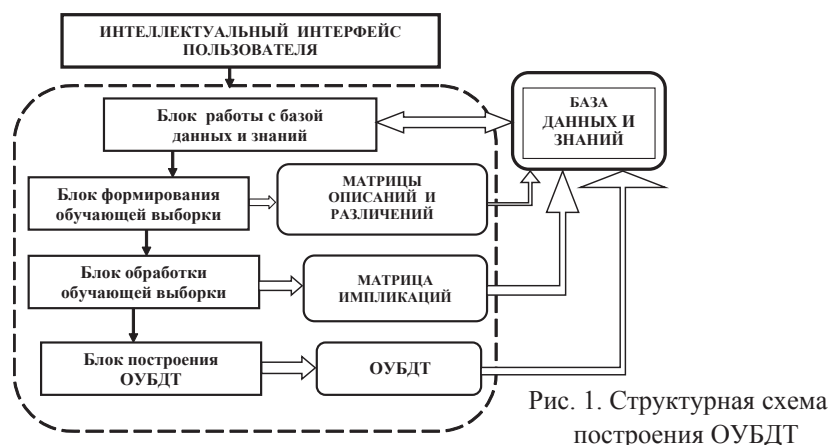


Рис. 1. Структурная схема построения ОУБДТ

Множество построенных ОУБДТ является основой для построения равномогного множества решающих правил.

Построение решающих правил сводится к построению по матрицам \mathbf{Q} и \mathbf{R} сжатых матриц описаний – \mathbf{Q}_i и различений – \mathbf{R}_i ($i = \overline{1, K}$, где K – количество построенных с использованием операции «проекция» ОУБДТ). В результате выполнения операции «проекция» в матрицу \mathbf{Q}_i включаются столбцы из матрицы \mathbf{Q} , сопоставленные признакам, входящим в i -й минимальный ОУБДТ, а также удаляются одинаковые строки из матрицы \mathbf{Q}_i и соответствующие им строки из матрицы \mathbf{R}_i .

Принятие решения относительно исследуемого объекта на множестве всех построенных ОУБДТ заключается в определении, к какому образу относит этот объект каждый тест [Янковская, 2009]. Итоговое решение о принадлежности исследуемого объекта тому или иному образу принимается с использованием процедуры голосования по всем тестам.

Управляющие и информационные взаимосвязи между блоками создаваемой подсистемы при принятии решений представлены на рис. 2.



Рис. 2. Структурная схема принятия решений

Отметим, что принимаемые итоговые решения могут носить классификационный, прогностический или организационно-управленческий характер.

3. Программная реализация

Создание интеллектуальной подсистемы построения ОУБДТ и принятия решений на их основе предлагается осуществлять на базе интеллектуального инструментального средства (ИИС) ИМСЛОГ [Yankovskaya et al., 2003], являющегося интегрированной средой для конструирования прикладных ИС.

Архитектура ИИС ИМСЛОГ является открытой и представляет собой иерархическую систему программных модулей. Один модуль реализован как резидентный, имеет встроенную систему команд, выполняет функции координирующего центра и называется ядром. Все остальные модули являются динамически подключаемыми, называются плагинами и подразделяются на функциональные модули, модули системных данных и базовый модуль интеллектуального пользовательского интерфейса.

Ядро обеспечивает автоматическое подключение и регистрацию всех плагинов, их динамическую загрузку и выгрузку в процессе работы, передачу управления от одного плагина к другому и информационную связь между плагинами, контроль корректности выполнения плагинов.

Модули системных данных хранят информационные структуры, используемые при работе. Базовый модуль интеллектуального интерфейса обеспечивает интерактивное взаимодействие пользователя с ядром ИИС ИМСЛОГ посредством графической оболочки либо (при необходимости) консольного приложения. Функциональные модули реализуют системные функции и встроенные средства ИИС ИМСЛОГ, а также функциональные блоки разрабатываемой интеллектуальной системы (подсистемы).

Каждый подключаемый модуль (плагин) должен содержать блок регистрации, необходимый для его автоматического подключения к ядру и динамического вызова на выполнение, а также программный шлюз, обеспечивающий корректные вызовы зарегистрированных системных команд, динамическое выделение требуемого объема оперативной памяти для хранения системных структур данных, передачу результатов выполнения команд и возврат управления ядру после завершения выполнения функции, соответствующей данному плагину.

Следует отметить, что в ИИС ИМСЛОГ встроено интерактивное средство, позволяющее в графическом виде формировать ар-

хитектуру и сценарий работы создаваемой системы (подсистемы) путем компоновки функциональных блоков в требуемой для решения поставленных задач последовательности и соединения входов и выходов этих блоков.

Сформированный сценарий работы сохраняется в виде шаблона и может исполняться как в автоматическом, так и в интерактивном режиме. При этом запуск плагинов и передача параметров между ними выполняется ядром ИИС ИМСЛОГ.

Кроме того, в ИИС ИМСЛОГ встроены средства создания, пополнения и редактирования баз данных и знаний, предназначенных для хранения информации об обучающих объектах и результатов их обработки.

Из вышесказанного следует, что каждому функциональному блоку создаваемой прикладной интеллектуальной системы нужно сопоставить программный компонент в виде динамически загружаемого модуля (плагина) и визуальный компонент, отображающий все входные и выходные параметры данного блока.

В соответствии с вышеизложенным математическим подходом к построению ОУБДТ и принятию решений на их основе в состав разрабатываемой интеллектуальной подсистемы включены следующие функциональные блоки, программно реализуемые как плагины:

1. Интеллектуальный интерфейс пользователя.
2. Блок работы с базой данных и знаний (выбор обучающих объектов).
3. Блок формирования обучающей выборки (построение матриц \mathbf{Q} , \mathbf{R}).
4. Блок обработки обучающей выборки (построение матрицы \mathbf{U}').
5. Блок построения минимальных ОУБДТ с учетом заданного количества ошибочных значений признаков.
6. Блок формирования решающих правил (построение сжатых матриц \mathbf{Q}_i и \mathbf{R}_i).
7. Блок формирования описания исследуемого объекта.
8. Блок принятия решений.

С точки зрения пользователя создаваемая подсистема предназначена для выполнения двух основных функций – построение

ОУБДТ и принятие на их основе решений относительно исследуемых объектов.

При этом функцию построения ОУБДТ требуется выполнять повторно только в случае изменения обучающей выборки, а функцию принятия решений можно выполнять многократно с использованием различных обучающих выборок, хранящихся в базе данных и знаний, для которых ранее была выполнена функция построения ОУБДТ, а также с различными исследуемыми объектами.

Заключение

Учет наличия ошибок измерения и/или занесения значений признаков, задающих описания исследуемых объектов, несомненно, повысит достоверность и качество принимаемых решений.

Кратко описаны модель представления данных и знаний об изучаемых объектах, математические основы построения ОУБДТ, архитектура и принципы программной реализации интеллектуальной подсистемы построения ОУБДТ принятия решений на их основе.

Реализация на базе ИИС ИМСЛОГ алгоритмов построения ОУБДТ и принятия решений на их основе позволит сконструировать интеллектуальную подсистему поддержки принятия решений, устойчивую к задаваемому пользователем количеству ошибок в описаниях исследуемых объектов, что расширит сферу ее практического применения.

В перспективе развитие подхода к принятию решений с применением ОУБДТ видится в изучении вопроса, дают ли такие тесты возможность не только принимать достоверные и качественные решения при наличии ошибок в описании исследуемого объекта, но и определять, какие именно признаки являются ошибочными.

Дальнейшие исследования будут направлены на построение ОУБДТ и принятие решений, основанные на конвергенции нескольких наук и научных направлений [Янковская, 2010], а также на повышение эффективности алгоритмов. Первые шаги в этом плане представлены в публикациях [Yankovskaya et al., 2013a; Yankovskaya et al., 2013b].

СПИСОК ЛИТЕРАТУРЫ

[Гедике и др., 2011] Гедике А.И., Янковская А.Е. Построение отказоустойчивых безусловных диагностических тестов для интеллектуальных

систем поддержки принятия решений // Системный анализ и информационные технологии (САИТ-2011). Труды четвертой Междунар. конф. – Челябинск: Изд-во Челябинского государственного университета, 2011. – Т. 1. – С. 128–133.

[Журавлёв, 1990] Журавлёв Ю.И. Распознавание образов и анализ изображений // Искусственный интеллект. Кн. 2. Модели и методы. – М.: Радио и связь. – 1990. – С. 149–191.

[Сагалович, 2012] Сагалович Ю.Л. Разделяющие системы. – М.: ИППИ РАН, 2012.

[Янковская, 2009] Янковская А.Е. Принятие решений, устойчивых к ошибкам измерения значений признаков в интеллектуальных системах // Искусственный интеллект. Интеллектуальные системы (ИИ-2009). Материалы X Междунар. научно-технич. конф. – Таганрог: Изд-во ТТИ ЮФУ, 2009. – С. 127–130.

[Янковская, 2010] Янковская А.Е. Анализ данных и знаний на основе конвергенции нескольких наук и научных направлений // Интеллектуализация обработки информации. Сб. докладов 8-й Междунар. конф. – М.: МАКС Пресс, 2010. – С. 196–199.

[Янковская, 2011] Янковская А.Е. Логические тесты и средства когнитивной графики. – Saarbrücken, Germany: LAP Lambert Academic Publishing GmbH & Co. KG, 2011. – 92 с.

[Yankovskaya et al., 2003] Yankovskaya A.E., Gedike A.I., Ametov R.V., Bleikher A.M. IMSLOG-2002 Software Tool for Supporting Information Technologies of Test Pattern Recognition // Pattern Recognition and Image Analysis. – 2003. – Vol. 13, No. 4. – P. 650–657.

[Yankovskaya et al., 2012] Yankovskaya A.E., Kitler S.V. Parallel Algorithm for Constructing k-Valued Fault-Tolerant Diagnostic Tests in Intelligent Systems // Pattern Recognition and Image Analysis. – 2012. – Vol. 22, No 3. – P. 473–482.

[Yankovskaya et al., 2013a] Yankovskaya A.E. and Kitler S.V. Intelligent System for Parallel Fault-Tolerant Diagnostic Tests Construction // Journal of Software Engineering and Applications. – 2013. – Vol. 6, No. 4A. – P. 54–61.

[Yankovskaya et al., 2013b] Yankovskaya A.E., Ametov R.V. Construction of fault-tolerance signal features subsets // Proceedings of 11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-11–2013). Vol. 1. – Samara: IPSI RAS, 2013. – P. 356–359.

СЕКЦИЯ 7 | **ПРИКЛАДНЫЕ
ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ**

УДК 004.8

**СОЗДАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СЕРВИСОВ
С ВИРТУАЛЬНОЙ РЕАЛЬНОСТЬЮ¹**

В.В. Грибова (*gribova@iacp.dvo.ru*)

К.А. Федорищев (*fleo@mail.ru*)

Институт автоматки и процессов управления
ДВО РАН, Владивосток

Работа посвящена описанию инструментария, платформы, а также технологии разработки нового класса программного обеспечения – интеллектуальных систем с виртуальной реальностью.

Введение

В настоящее время интеллектуальные системы активно используются для решения различных профессиональных задач. Типичная архитектура интеллектуальной системы состоит из базы знаний; решателя задач, предназначенного для обработки символьной информации, представленной в базе знаний при построении логического вывода; пользовательского интерфейса, отображающего пользователю результаты работы, как правило, в форме таблиц и текстов некоторой структуры.

В то же время очень активно развивается класс систем, предназначенный для создания симуляторов, компьютерных обучающих тренажеров, которые позволяют на основе виртуальной реальности (реализованной средствами компьютерной графики, физики, аппаратными технологиями) проводить отработку профессиональ-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 13-07-00024-а и № 12-07-00179-а) и КПФИ «ДВ».

ных навыков и знаний, ставить безопасные виртуальные эксперименты и т.п. Реализация данного класса систем осуществляется с использованием различных специализированных и универсальных инструментальных средств, пакетов прикладных программ, библиотек для создания виртуальных сред, поддерживающих традиционные подходы и технологии программирования [Грухин, 2008], [Craig et al., 2009], несмотря на то, что в основе таких систем лежат экспертные знания специалистов предметных областей.

Целью данной работы является описание инструментария для создания нового класса интеллектуальных систем, объединяющего технологию создания и архитектуру интеллектуальных систем с системами виртуальной реальности.

1. Постановка задачи

Анализ литературы, а также практический опыт разработки компьютерных тренажеров показал, что в общем случае процесс создания виртуальной среды состоит из нескольких основных этапов: разрабатываются 3D-модели объектов виртуального мира, для большинства моделей объектов описываются сценарии их поведения и возможного изменения отображения в виртуальной среде; затем из разработанных объектов формируется виртуальная среда (виртуальное окружение) – определяется положение объектов относительно друг друга, их размер, повороты и др. необходимые атрибуты; отдельной трудоемкой задачей является описание возможных сценариев как влияния объектов друг на друга, так и изменения виртуального мира при воздействии на него пользователя; функционал некоторых интерактивных сред требует включения оценки действий пользователя при его взаимодействии с виртуальным миром.

Все имеющиеся средства создания систем данного типа ориентированы на их использование программистами, иногда совместно с дизайнерами, включение в процесс разработки экспертов предметной области возможен только в качестве консультантов, а не полноправных его участников, вместе с тем доля экспертных знаний в таких системах составляет большую часть системы, но эти экспертные знания встраиваются в программный код системы. Любое изменение требует трудоемкого репрограммирования, последующей сборки и компиляции виртуальной среды [Craig et al., 2009].

Таким образом, актуальной задачей является разработка специализированного инструментария и технологии разработки интеллектуальных систем с виртуальной реальностью, во-первых, ориентированных на специалистов различных типов, принимающих участие в разработке, – экспертов предметной области, дизайнеров, программистов, во-вторых, разделяющих работу между ними, в-третьих, имеющих средства интеграции и проверки целостности компонентов в единую работоспособную систему, в-четвертых, обеспечивающих широкую доступность через Интернет как средств разработки и сопровождения таких виртуальных сред, так и готовых реализаций.

2. Основные принципы создания инструментария

Для реализации указанных выше требований, а также для упрощения процесса разработки программирование системы на языке программирования заменяется разработкой ее декларативной модели с последующей интерпретацией, а также возможностью подключения дополнительных императивных модулей обработки, что позволяет свести к минимуму кодирование на языке программирования, а также расширить потенциальный функционал инструментария в процессе его жизненного цикла.

Для включения в процесс разработки дизайнеров и экспертов предметной области предлагается онтология виртуальной среды, в терминах которой эксперты предметной области и дизайнеры разрабатывают и сопровождают модель системы через структурный и графический редактор соответственно. Использование онтологии [Валькман, 2012], [Гаврилова, 2001], [Клещев, 2008] виртуальной среды, описывающей систему понятий виртуального мира, связей между ними и ограничения целостности, направлено на упрощение создания модели, поскольку разработчикам не надо изучать какой-либо формализм для описания модели (особенно это важно для экспертов предметной области, не знакомых с языками программирования и технологией разработки программных систем). В онтологии выделены логический и презентационный уровни. Логический уровень предназначен для экспертов предметной области, презентационный – для дизайнеров. Выделение в онтологии двух уровней – логического и презентационного направлено на

разделение работ между разработчиками – экспертами предметной области и дизайнерами.

Для обеспечения широкой доступности средства разработки и готовое приложение предлагаются как интернет-сервисы. Использование Интернета и технологии облачных вычислений в общем случае, как отмечено в ряде источников, например, [Концер, 2008] [Клещев и др., 2010], дает новый уровень гибкости для управления и сопровождения программных средств, а также значительно расширяет аудиторию пользователей.

В соответствии с представленными принципами создания разработана архитектура системы, подробно представленная в [Грибова и др., 2012]. Программный комплекс состоит из нескольких компонентов, которые можно разделить на два класса: информационные и программные. Информационными компонентами комплекса являются: онтология виртуальной среды, декларативная модель, сформированная по онтологии, мультимедиа-данные, внешние функции. Программными компонентами комплекса являются: структурный редактор декларативной модели; графический 3D-редактор; редактор внешних функций; интерпретатор.

Онтология виртуальной среды состоит из трех основных компонентов: объекты, действия и сценарий [Грибова и др., 2012].

Объекты = {Простой объект, Изменяемый объект, Составной объект, Таблица, Источник света, Камера}; $0 \leq i \leq$ количество объектов

Простой объект = <Имя объекта, Описание, Логические атрибуты, Презентационные атрибуты >

Логические атрибуты = {Логический атрибут}; $0 \leq i \leq$ количество логических атрибутов

Логический атрибут = <Имя \in Строка, Значение \in Типы данных >

Презентационные атрибуты = <Модель, Текстуры, Координаты, Повороты, Масштабы, Анимация >

<...>

Действия = {Действие_i | Действие_i \in Действие}; $0 \leq i \leq$ количество действий

Действие = <Имя действия, Описание, Способы выполнения, Входные параметры, Изменение состояния объектов, Получение оценки, Параметры обработки, Сообщение >

<...>

Сценарий = <Имя сценария, Этапы, Последовательность этапов>

Этапы = { Этап }_i, 0 ≤ i ≤ количество этапов

Этап = <Входные параметры, Вершины, Дуги, Начальная вершина, Конечная вершина>

<...>

На основе этой онтологии строится декларативная модель, состоящая из множества объектов виртуальной среды, множества действий, которые можно осуществлять с этими объектами, и сценария.

3. Платформа для реализации интеллектуальных систем с виртуальной реальностью

Реализация программного комплекса выполнена на платформе IASaaS [Грибова и др. 2011], которая позволила решить ряд поставленных к программному комплексу требований. Информационные и программные ресурсы комплекса, включая онтологию, декларативные модели, медиа-данные, агенты, редакторы, располагаются в фонде платформы IASaaS. Вся информация – онтология, декларативные модели, данные (включая временные) – имеет унифицированное представление в форме семантических сетей. Программные компоненты комплекса, включая редакторы и интерпретатор, реализованы как сервисы платформы IASaaS на основе мультиагентного подхода на языке программирования Java. Агенты реализованных сервисов обмениваются информацией друг с другом с помощью сообщений.

Веб-клиенты графического редактора и интерпретатора реализуются как Flash-модули, встраиваемые в веб-страницу и отображаемые пользователю в браузере. Генерация трехмерной виртуальной сцены происходит с применением Flash-библиотеки Alternativa3D, использующей технологию Molehill. Взаимодействие между веб-клиентом и сервисом платформы происходит путем передачи асинхронных сообщений серверному агенту, ответственному за обработку событий, и фоновое ожидание от него результирующих сообщений.

4. Разработка модели системы

Разработка модели интеллектуальной системы с виртуальной реальностью включает следующие этапы [39]: разработка логического представления декларативной модели экспертами предметной области; разработка презентационного (графического) представления модели дизайнерами; разработка агентов программами (данный этап не является обязательным, однако, если предоставляемого функционала оказывается недостаточно, его функционал может быть расширен дополнительными функциями).

Технология разработки декларативной модели состоит из следующих шагов: создание информационного ресурса декларативной модели; формирование объектов, действий и сценария. Для создания информационного ресурса декларативной модели эксперт через административную систему платформы IASaaS подает заявку на создание нового информационного ресурса, после создания которого приступает к формированию трех составляющих декларативной модели: объектов, действий, сценария. Формирование модели и ее последующая модификация осуществляются с помощью редактора ИРУО, который является составным компонентом платформы IASaaS. Онтология виртуальной среды поступает на вход редактору ИРУО, пользователь (эксперт), конкретизируя термины онтологии, получает логическое представление конкретной модели.

Разработка презентационного представления модели происходит после того, как эксперт выполнит формирование ее логического представления. Презентационное представление сцены формирует дизайнер с помощью сервиса «Графический редактор трехмерных сцен». В данный редактор загружается сформированное экспертом логическое представление модели. С помощью редактора в визуальном интерактивном режиме дизайнер определяет презентационные атрибуты объектов сцены. Для задания презентационных атрибутов дизайнеру необходимо определить трехмерную модель объекта, текстуру, координаты, повороты, коэффициенты масштабирования, анимации.

Таким образом, технология проектирования предполагает сначала разработку логического представления модели экспертом предметной области с использованием структурного редактора,

затем разработку презентационного представления дизайнером с использованием графического редактора на основе загруженной в него логической модели. При этом два представления неразрывно связаны: презентационное представление является расширением логического представления и определяет, как будут представлены на виртуальной сцене объекты, определенные в логическом представлении экспертом предметной области.

В описании декларативной модели виртуальной среды могут понадобиться параметры, которые должны вычисляться специфическим (нестандартным) способом. В этом случае такой параметр помечается как «параметр обработки агентом», для него в модели указываются имя, описание и возвращаемое значение. В общем случае программная реализация специфической обработки параметра является независимой задачей и может быть выполнена отдельным внешним независимым модулем. Такие модули в системе формируются как агенты (платформа IACPaaS реализована как мультиагентная).

Написание агента для программиста, по сути, представляет собой написание некоторой независимой функции с известными входными и выходными данными. Входными данными для агента являются динамическая версия декларативной модели виртуальной среды и сообщение, переданное от веб-клиента. Требуемые параметры среды и параметры сообщения веб-клиента считаются известными программисту (он должен их получить из описания вычисляемого параметра).

Проверка целостности разработанной декларативной модели профессиональной виртуальной среды состоит из двух уровней. Первый уровень заключается в том, что декларативная модель строится на основе онтологии, содержащей набор ограничений целостности, который проверяется автоматически при создании декларативной модели, используя структурный редактор. Таким образом, первый уровень проверки целостности обеспечивает синтаксическую проверку целостности модели. Второй уровень проверки целостности осуществляется на программном уровне в графическом редакторе и интерпретаторе, в каждом из которых проверяется наличие всех необходимых на данный момент сущностей декларативной модели для воспроизведения, а также их корректность, т.е. наличие необходимых ресурсов, ссылок, отсутствие

перекрестных ссылок и т.п. Таким образом, второй уровень проверки целостности обеспечивает частичную семантическую проверку целостности декларативной модели.

5. Прикладные виртуальные среды

Используя предложенный инструментарий, созданы следующие виртуальные среды: компьютерный обучающий тренажер для офтальмологии, виртуальная химическая лаборатория, проект городского района.

Компьютерный обучающий тренажер включает обучающие задания по классическим методам исследования в офтальмологии: определение клинической рефракции (скиаскопии), исследование поля зрения методом кампиметрии, определение остроты зрения по таблицам Сивцева, Головина, определению остроты зрения по оптотипам Б.Л. Поляка, и другие.

Данный тренажер включает следующие экспертные знания: характеристики остроты зрения для каждой строки таблицы в соответствии с вариантами ответов пациентов, характеристики для разных оптотипов и других медицинских приспособлений, сценарии действий и поведения студентов в соответствии с различными ответами пациентов, объяснения правильных и неправильных действий обучающихся. Все эти знания являются основой данного тренажера и были внесены экспертом предметной области в декларативную модель с помощью структурного редактора. Затем дизайнером было сформировано презентационное представление заданных экспертом элементов сцены с помощью создания трехмерных моделей и внесения их через графический редактор в декларативную модель виртуальной среды. Программистская задача включала программирование одного небольшого модуля-агента, определяющего номер строки таблицы при взаимодействии с ним пользователя.

Реализация аналогичного компьютерного тренажера с помощью иных средств создания виртуальных сред потребовала бы значительно больших усилий и средств, т.к. ту работу, которую выполнил эксперт предметной области, пришлось бы выполнять программисту вместе с экспертом, создавая множество блоков кода, реализующих все соответствия и соотношения между большими

наборами знаний. Программисту также пришлось бы создавать программное окружение виртуальной среды с помощью различных движков, библиотек, технологий, вручную реализуя множество необходимых интерактивных возможностей тренажера для пользователей: перемещение в пространстве, взаимодействие с объектами, генерация и обработка событий, условий и ответов на них, получение и отображение результатов и объяснений. Дизайнеру же кроме создания необходимых трехмерных моделей и текстур пришлось бы вместе с программистом много времени уделять правильному формированию сцены с помощью применения и настройки каких-то дополнительных инструментальных средств графического моделирования. Кроме того, программисту пришлось бы отдельно программировать все облачные задачи для обеспечения работы тренажера в Интернете.

Заключение

В работе описаны основные принципы создания и компоненты инструментария для разработки нового класса приложений – интеллектуальных систем с виртуальной реальностью. Особенностью инструментария является то, что кодирование интеллектуальной системы заменяется проектированием ее декларативной модели с последующей интерпретацией. При этом в процесс проектирования включаются дизайнеры и эксперты предметной области, для которых разработаны онтология и редакторы, управляемые онтологией, для упрощения процесса разработки модели и ее последующего сопровождения. Инструментарий и готовое приложение, создаваемое на его основе, реализованы как облачные сервисы.

СПИСОК ЛИТЕРАТУРЫ

[Валькман, 2012] Валькман Ю.Р., Валькман Р.Ю. Онтологии в моделировании образного мышления. – Труды КИИ-2012. Т.1-Белгород: Изд-во БГТУ, 2012. С. 310–317.

[Гаврилова, 2001] Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. – СПб.: Питер, 2001. 384 с.

[Грибова и др., 2011] Грибова В.В., Клещев А.С., Крылов Д.А., Москаленко Ф.М., Смагин С.В., Тимченко В.А., Тютюнник М.Б., Шалфеева Е.А. Проект IASaaS. Комплекс для интеллектуальных систем на основе

облачных вычислений // Искусственный интеллект и принятие решений. 2011. № 1. С. 27–35.

[Грибова и др., 2012] Грибова В.В., Федорищев Л.А. Интернет-комплекс для создания обучающих систем с виртуальной реальностью // Дистанционное и виртуальное обучение. – 2012. – № 7. – С. 4–12.

[Клещев, 2008] Клещев А.С. Роль онтологии в программировании. Ч. 2. Интерактивное проектирование информационных объектов//Информационные технологии, 2008. №11, с. 28–33.

[Клещев и др., 2010] Клещев А.С., Грибова В.В. Шалфеева Е.А. Управление интеллектуальными системами //Известия РАН. Теории и системы управления. 2010. № 6. С. 122–137.

[Концер, 2008] Концер Т. «Облачные» вычисления: все как сервис // PC Week/RE. №32 (638). 2008. <http://www.pcweek.ru/themes/detail.php?ID=112879>.

[Трухин, 2008] Трухин А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. – 2008. – №1. – С. 32–39.

[Craig et al., 2009] Craig A., Sherman W, Will J., Developing Virtual Reality Applications: Foundations of Effective Design. – New York: Morgan Kaufmann, 2009. – 448 pp.

УДК 004.855.5, 004.853

МОДЕЛЬ ПРОЯВЛЕНИЯ ЭМОЦИЙ В ЕСТЕСТВЕННОЙ РЕЧИ¹

И.А. Ребрун (*igar_rebrun@mail.ru*)

К.В. Сидоров (*bmisidorov@mail.ru*)

Н.Н. Филатова (*nfilatova99@mail.ru*)

Д.М. Ханеев (*t_ran@mail.ru*)

Тверской государственной технической университет, Тверь

Рассматривается задача формализованного описания влияния изменения эмоционального состояния (зарегистрированного на основе объективных данных) на процесс формирования речевых сообщений на основе анализа коррелированных данных от двух источников сигналов (головного мозга и речевого аппарата).

Введение

На данном этапе проблема распознавания эмоционального состояния человека путем анализа его речи решается довольно успешно в основном для случаев сильных проявлений эмоций. Число вычисляемых для этого признаков обычно составляет несколько десятков. Необходима разработка более простой модели, описывающей взаимосвязи между основными параметрами эмоций (знака, силы, направления развития и др.) и объективными оценками фазово-пространственных характеристик наблюдаемого акустического (речевого) сигнала. Решение этой задачи позволит создать такую модель интерпретатора эмоций по естественной речи, которая обеспечит достаточную чувствительность к малым изменениям эмоционального состояния и будет учитывать гендерные и возрастные особенности человека.

Повышение чувствительности интерпретаторов эмоций необходимо для развития моделей принятия согласованных решений в слабо структурированных средах. Учитывая способности человека контролировать и управлять своей мимикой и жестами, наиболее важным источником сведений об его эмоциональном состоянии становится речь. Обычно человек использует эмоции как дополни-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-01-00719).

тельный источник информации (не вербальной), которую он хочет передать собеседнику. Следует отметить, что этот процесс может протекать не вполне осознанно, т.е. человек может пытаться скрыть какую-то информацию от собеседника, но эмоциональный окрас сообщения его выдаст. Таким образом, модель интерпретации эмоций, слабо выраженных в речи, позволит дополнить модель принятия согласованных решений новой объективной информацией, что должно привести в итоге к повышению качества решений.

1. Эмоции и их проявление в естественной речи

Принятие решений, как любые проявления активности человека, сопровождается эмоциональными переживаниями. Эмоции являются результатом деятельности мозга, это своеобразное проявление осознанных реакций организма на внешние стимулы.

При формализованном описании эмоций разработчики интеллектуальных систем опираются в основном на модели OCC, KARO, EMA, Affective Computing, созданные психологами. Модель OCC (Ortony, Clore, Collins) описывает качественную и количественную сторону 22 видов эмоций. Качественно OCC характеризует условия возникновения каждой эмоции, количественно указывает, как интенсивность эмоции определяется условиями ее возникновения и какие переменные влияют на интенсивность. [Ortony, 1988]. KARO («Knowledge, Actions, Results and Opportunities») является описанием модели OCC на основе формальной логики [Steunebrink, 2008]. Модель EMA также основана на развитии модели OCC. Для описания эмоций используется несколько переменных: полезность, желательность (D), вероятность события (p) и т.п. [Gratch, 2004]. Представители направления AC (Affective Computing) и ASP декларируют возможность распознавания эмоций человека системой по мимике, жестам и сигналам от ограниченного числа биосенсоров. Переход из одного эмоционального состояния в другое описывается с помощью цепей Маркова [Picard, 1995].

В работе [Фоминых, 2006] эмоция определяется как числовая функция F (сила эмоций) от некоторого набора параметров. Для каждого вида эмоций предложены свой набор параметров и своя функция F . Такие общие представления близки к OCC. Различие заключается в наборе параметров, соответствующих каждой эмоции. Основным параметром для утилитарных эмоций является количество полученного (потерянного) ресурса R или уровень дости-

жений. Модель носит определительный характер. Имеется еще ряд отечественных разработок, в которых предлагаются подходы к математическому описанию процесса эмоционального возбуждения ([Розалиев, 2009], [Перервенко, 2009] и др.), однако для построения числовых оценок их использование затруднительно.

Большинство предложенных решений относится к т.н. системам распознавания речи и базируется на оценках спектрально-временных, кепстральных и амплитудно-частотных признаков, определяемых в ходе анализа акустической волны. Для распознавания знака эмоций в речи обычно используются несколько десятков таких признаков. Причем высокую точность (порядка 80–90%) эти системы обеспечивают для речи с ясно выраженным эмоциональным окрасом (в основном для экстремальных и близких к ним ситуациям). В докладе рассматривается задача уменьшения признакового пространства и распознавания речевых образцов с невысокой интенсивностью проявления эмоций.

2. Биотехническая система для исследования эмоций в естественной речи при слабом эмоциональном возбуждении

При разработке моделей проявления эмоций в речи используется экспериментально-статистический подход. В качестве объектов анализа выступает выборка из паттернов речевых сообщений. Для их формирования могут использоваться образцы искусственной (сценической) или естественной речи. В первом случае паттерны получают из фраз дикторов (обычно профессиональных актеров или оперных певцов), имитирующих определенный вид эмоции. При втором подходе в качестве примеров используют записи речи операторов, или других специалистов, профессионально участвующих в коммуникационном процессе [Хроматиди, 2005]. Как в первом, так и во втором случае у нас отсутствует объективная информация, подтверждающая изменение эмоционального состояния испытуемого. Это приводит к сомнениям по поводу состава выборок. Очевидно, что при экспертной оценке эмоционального состояния испытуемого в выборку надежно можно включать только образцы речи с ярко выраженными эмоциями. Границы кластеров, отображающих объемы понятий об эмоционально окрашенной речи (ЭОР) с разными знаками и уровнями проявления эмоций, будут представлены очень неточно.

В работе [Сидоров и др., 2014] рассматривался способ регистрации образцов речи, основанный на объективных данных, характеризующих состояние источника речевых сообщений. Для контроля эмоционального состояния испытуемого применялась регистрация потенциалов, характеризующих электрическую активность мозга.

Развитием этого предложения является биотехническая система (БТС) для исследования эмоций в естественной речи при слабом эмоциональном возбуждении. К особенностям этой БТС относятся два независимых канала подачи стимулов (визуальных V и акустических A) и два канала регистрации откликов испытуемого: E, канала регистрации электроэнцефалограммы (ЭЭГ) и R, канала регистрации речи испытуемого.

Информация, получаемая при обработке сигналов канала E, отображает эмоциональные реакции испытуемого, которые он сознательно не корректирует. Оценки эмоционального состояния испытуемого на основе канала R могут быть искажены в силу субъективных причин, например, сознательным желанием испытуемого не демонстрировать явно своих оценок предъявляемого стимула.

Главное преимущество двухканальной БТС связано с возможностью регистрации речевого сообщения только после регистрации изменений в ЭЭГ, вызванных восприятием соответствующего стимула. Такой подход повышает доверие к выборкам образцов речи, которые впоследствии могут использоваться для формирования правил распознавания видов эмоций. С помощью указанной БТС нами сформирована мультимодальная БД, включающая 210 фраз различных дикторов и соответствующие им записи ЭЭГ. В создании этой коллекции приняли участие мужчины и женщины в возрасте от 18 до 27 лет, родным языком которых является русский. Часть записей сделана с участием студентов-иностранцев, родным языком которых является французский. Образцы речи, продолжительностью от 3 до 10 секунд, регистрировались с частотой дискретизации 22050 Гц и разрешением 16 бит. Образцы ЭЭГ длительностью по 1 минуте регистрировались с частотой дискретизации 250 Гц. В качестве стимульного материала использовались видео- и музыкальные клипы с тематикой, ориентированной на разные вкусы. Предварительно каждый испытуемый знакомился с образцами стимулов, и для него формировался индивидуальный сценарий исследования. Следует отметить, что в оценке видеостимулов, как уже отмечалось ранее [Лапшина, 2007], все испытуемые

оценивали положительно кадры с детенышами животных и давали отрицательные оценки на сцены насилия, а также стоматологические или хирургические операции. В оценке музыкальных стимулов расхождения были максимальными и, в зависимости от субъективных предпочтений, сценарии для канала A настраивались для каждого испытуемого в отдельности.

3. Признаки для описания проявления эмоций в речи

Предположим, что воздействие некоторого внешнего стимула (S) вызывает у испытуемого определенную эмоциональную реакцию. Примем допущение, что после снятия S испытуемый останется в достигнутом эмоциональном состоянии в течение всего времени формирования им речевого сообщения (T). Тогда образцы речи ($A(t, t < T)$) можно рассматривать как паттерны временных рядов, содержащих информацию о некоторой эмоциональной реакции (т.е. оценке испытуемым S). Для анализа $\{A(t)_j\}$ используются методы нелинейной динамики, эффективность которых уже иллюстрировалась рядом исследователей. Эта группа методов базируется на фундаментальной математической теории, в основе которой лежит теорема Ф. Такенса [Takens, 1981], которая доказывает возможность реконструкции (построения) аттрактора по временному ряду. Методика построения аттрактора для образцов ЭЭГ и ЭОР описана в работе [Филатова и др., 2014]. Получены двухмерные проекции аттракторов для речевых образцов и для соответствующих образцов ЭЭГ (рис.1).

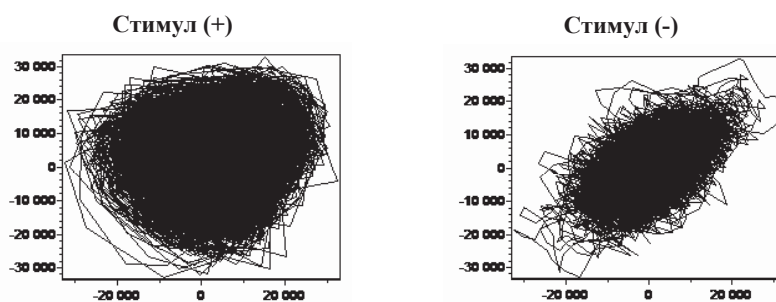


Рис.1. Двухмерные проекции аттракторов, построенные по одному отведению ЭЭГ при вариации внешнего видеостимула (S_j)

Для количественной меры, характеризующей вариации аттракторов при изменении эмоционального состояния говорящего, предлагается использовать два признака: длину вектора, описывающего контур аттрактора (R_{max}^j) [Сидоров и др., 2012], и новый признак – плотность аттрактора (γ). Правило оценки первой характеристики (R_{max}^j) иллюстрирует рис. 2.

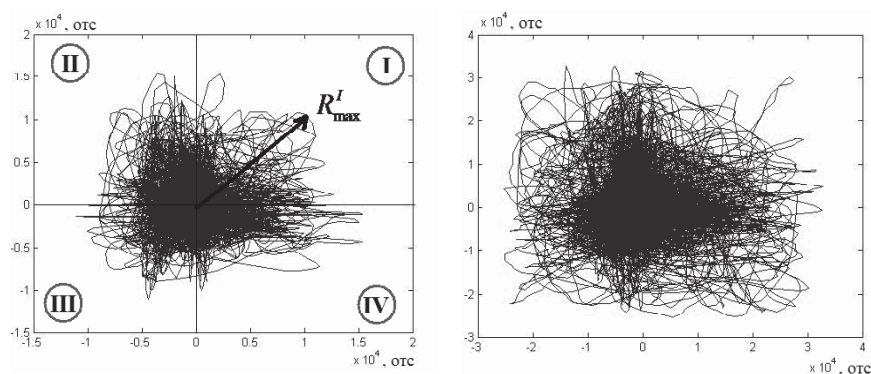


Рис.2. Двухмерные проекции аттракторов, построенные по образцам естественной речи при отрицательных и положительных эмоциях

Максимальный вектор контура аттрактора зависит от интенсивности акустического сигнала и уменьшается при переходе от речевых паттернов, зарегистрированных при положительных эмоциях, к образцам речи при отрицательных эмоциях. Для оценки плотности траекторий аттрактора его двухмерная проекция покрывается регулярной сеткой. В наших экспериментах использовалась сетка с шагом 100 отсчетов, с общим числом ячеек 256, с площадью каждой ячейки (s_j).

Для каждой j -ой ячейки определяется число точек аттрактора, попавших внутрь (m_j). Количество точек (r_j), оказавшихся на границе ячеек j -ой и $(j+1)$ -ой, делится поровну между граничными клетками. Отношение числа точек аттрактора, связанных с ячейкой ($N_j = m_j + r_j/2$), к ее площади условно определяется как плотность аттрактора на j -ом участке проекции: $\gamma_j = N_j/s_j$. Очевидно, что для любого аттрактора $Y(x(t), x(t+\tau))$ можно получить график распределения характеристики (γ_j) в зависимости от номера ячеек,

покрывающих плоскость проекции. На рисунке 3 показаны примеры сравнения аттракторов речевых сигналов при отрицательных и положительных эмоциях испытуемого.

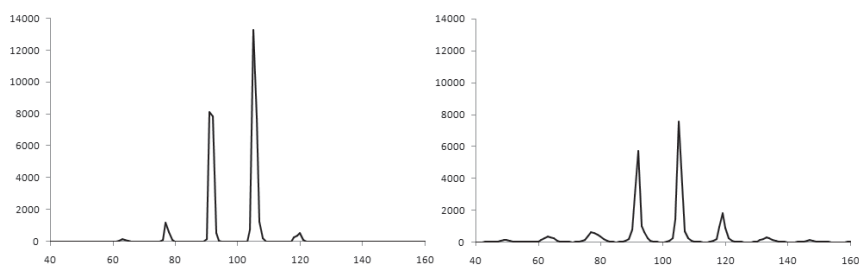


Рис.3. Плотность аттракторов образцов речи при отрицательных и положительных эмоциях

Проведенные эксперименты со слабовыраженными эмоциями показали, что различия в нейтральных образцах и паттернах, зарегистрированных при восприятии отрицательных стимулов, обычно плохо проявляются. Эта особенность приводит в других признаковых пространствах к большим пересечениям указанных кластеров [Филатова и др., 2014], и именно поэтому результаты, полученные с помощью нового признака, имеют большую ценность. Учитывая существенную вариабельность оценок (R_{max}^i и γ) в связи с индивидуальными различиями испытуемых и нестационарностью речевого сигнала, целесообразно от количественных оценок признаков перейти к лингвистическим шкалам. Для фаззификации признаков вводятся два терм-множества (TR и $T\gamma$), включающие 3 терма для оценки размеров вектора, описывающего контур, и 5 термов для оценки плотности аттрактора.

4. Алгоритмическая модель проявления эмоций в речи

Для создания алгоритмической модели, интерпретирующей проявление эмоций в естественной речи, использована гибридная система, интегрирующая стратегии нейросетевых моделей и методы обработки нечетких данных для построения классификационных правил [Филатова и др., 2014]. Система проходит два

цикла обучения: сначала на фазифицированных описаниях сигналов ЭЭГ (ОВ1), а потом на аналогичных выборках векторов, созданных на основе речевых образцов (ОВ2). В процессе обучения система должна для каждой обучающей выборки построить правила распознавания эмоциональных состояний d такие, что классификация испытуемого с помощью классификатора $K(ОВ1)$ не будет противоречить результатам его классификации с помощью классификатора $K(ОВ2)$. Алгоритм генерации нейроподобной иерархической структуры (НИС) подробно описан в работе [Филатова и др., 2014]. Для построения алгоритма использованы идеи растущих пирамидальных сетей [Гладун, 2001], которые позволяют создавать хорошо интерпретируемые описания классов. Классификатор реализован на языке C# 3.0 для среды исполнения NET Framework 3.5. Ядро программы (модули фазификации, генерации сетевой модели, выделения описаний классов, нечеткого логического вывода) инвариантно к предметной области. Исследование работы классификатора показало довольно высокую точность работы как на образцах ЭЭГ, так и на речевых паттернах. Применение набора признаков, рассчитанных на основе реконструкции аттрактора, позволяет снизить ошибку классификации по сравнению с традиционным использованием для описания речевых объектов характеристик спектров мощности (табл. 1).

Таблица 1

Выборка	ОВ	ТВ
Число объектов из {Class+Э} → {Class нет Э}	0	0
Число объектов из {Class +Э } → {Class -Э}	0	0
Успешность классификации Class +Э, %	100	100
Число объектов из {Class нет Э } → {Class +Э }	0	1
Число объектов из {Class нет Э } → {Class -Э }	0	1
Успешность классификации Class нет Э, %	100	86
Число объектов из {Class -Э } → {Class +Э }	0	0
Число объектов из {Class -Э } → {Class нет Э }	0	2
Успешность классификации Class -Э, %	100	80

Заключение

Биотехническая система для исследования эмоций в естественной речи при слабом эмоциональном возбуждении и алгоритмическая модель, интерпретации проявления эмоций в естественной речи прошли испытания на образцах русской, французской и немецкой речи. Испытания подтвердили хорошие разделительные свойства предложенных признаков, отображающих изменения свойств временных рядов на основе оценки морфологии аттракторов. Новые признаки инвариантны к виду первичного сигнала. Это позволило использовать их не только для анализа речевых аттракторов, но и для построения аттракторов ЭЭГ. Сходимость результатов интерпретации эмоций на множествах образцов ЭЭГ и речевого сигнала подтверждает (в 100% случаев) достоверность предложенной модели.

СПИСОК ЛИТЕРАТУРЫ

- [Ortony, 1988] Ortony A., Clore G. L., Collins A. The Cognitive Structure of Emotions // Cambridge. UK: Cambridge University Press. 1988.
- [Steunebrink, 2008] Steunebrink B.R., Dastani, M.M. & Meyer, J-J.Ch., N. Fakotakis & N. Avouris (Eds.) A Formal Model of Emotions: Integrating Qualitative and Quantitative Aspects // Proc. 18th European Conference on Artificial Intelligence. 2008.
- [Gratch, 2004] Gratch J., Marsella S. A domain-independent framework for modeling emotions // J. of Cognitive Systems Research. 2004. № 5(4).
- [Picard, 1995] Picard R.W. Affective Computing // M.I.T Media Laboratory Perceptual Computing Section Technical Report. № 321.
- [Фоминых, 2006] Фоминых И.Б. Эмоции как аппарат оценок поведения интеллектуальных систем // Труды 10-й Национальной конференции по искусственному интеллекту КИИ-2006. Т.2. – М.: Физматлит, 2006.
- [Розалиев, 2009] Розалиев В.Л. Моделирование эмоциональных реакций пользователя при речевом взаимодействии с автоматизированной системой: дис. ...канд. тех. наук. – Волгоград: ВГТУ, 2009.
- [Перервенко, 2009] Перервенко Ю.С. Исследование инвариантов нелинейной динамики речи и принципы построения системы анализа психофизиологического состояния: дис. канд. тех. наук. – Таганрог: ТГРТУ, 2009.
- [Хроматиди, 2005] Хроматиди А.Ф. Исследование психофизиологического состояния человека на основе эмоциональных признаков речи: дис. ...канд. тех. наук. – Таганрог. ТГРТУ, 2005.

[Сидоров, 2014] Сидоров К.В., Филатова Н.Н., Ханеев Д.М. Применение нейрореподобной иерархической структуры для классификации знака эмоций // XVI Всероссийская НТК «НЕЙРОИНФОРМАТИКА-2014»: Сб. научных трудов. Ч. 2. – М.: НИЯУ МИФИ, 2014.

[Лапшина, 2007] Лапшина Т.Н. Психофизиологическая диагностика эмоций человека по показателям ЭЭГ: дис. ...канд. тех. наук. – М.: МГУ, 2007.

[Takens, 1981] Takens F. Detecting strange attractors in turbulence // Dynamical Systems and Turbulence. Heidelberg: Springer-Verlag. 1981.

[Филатова, 2014] Филатова Н.Н., Ханеев Д.М., Сидоров К.В. Интерпретатор сигналов на основе нейрореподобной иерархической структуры // Программные продукты и системы. 2014. №1(105).

[Сидоров, 2012] Сидоров К.В., Филатова Н.Н. Автоматическое распознавание эмоций человека на основе реконструкций аттракторов образцов речи // Программные системы и вычислительные методы. 2012. №1(1).

[Гладун, 2004] Гладун В.П. Растущие пирамидальные сети // Новости искусственного интеллекта. 2004. №1.

УДК 004.89

**НЕКОТОРЫЕ АСПЕКТЫ УПРАВЛЕНИЯ ПРОЦЕССАМИ
ПОСТРОЕНИЯ ИНТЕГРИРОВАННЫХ ЭКСПЕРТНЫХ
СИСТЕМ НА ОСНОВЕ ИНТЕЛЛЕКТУАЛЬНОГО
ПЛАНИРОВАНИЯ¹**

Г.В. Рыбина (*galina@ailab.mephi.ru*)

Ю.М. Блохин (*galina@ailab.mephi.ru*)

Национальный исследовательский ядерный университет «МИФИ»,
Москва

В данной работе рассматриваются проблемы интеллектуализации процессов разработки интегрированных экспертных систем на основе задачно-ориентированной методологии и инструментального программного комплекса АТ-ТЕХНОЛОГИЯ. Описывается опыт применения методов интеллектуального планирования для синтеза макетов архитектуры прототипов интегрированных экспертных систем на основе использования интеллектуального планировщика.

Введение

В современных условиях бурно развивающегося рынка информационных технологий и смещения акцентов в сторону так называемой «открытости» систем как для пользователей, так и для разработчиков важными требованиями к методологиям и инструментальным средствам построения интеллектуальных систем становятся факторы снижения стоимостных показателей разработки и интеллектуальной нагрузки на инженеров по знаниям. Особую значимость и актуальность эти проблемы приобретают для поддержки процессов построения интегрированных экспертных систем (ИЭС), обладающих мощной функциональностью и масштабируемой архитектурой.

Опыт, накопленный в ходе создания целого ряда статических и динамических ИЭС на основе использования задачно-ориентированной методологии [Рыбина, 2008] и поддерживающего эту методологию инструментального комплекса АТ-ТЕХНОЛОГИЯ, показал, что значительная часть проблем связана с высокой слож-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-01-00467).

ностью этапов проектирования и реализации ИЭС, причем существенное влияние оказывают специфика конкретной проблемной области и человеческий фактор [Рыбина, 2011]. Поэтому одним из важных направлений развития задачно-ориентированной методологии и комплекса АТ-ТЕХНОЛОГИЯ стала разработка *интеллектуальной программной среды* и ее базового компонента – интеллектуального планировщика (ИП) с целью расширения степени автоматизации планирования и управления проектами по разработке ИЭС. К настоящему времени для комплекса АТ-ТЕХНОЛОГИЯ создано несколько версий ИП, разработанных на основе объединения моделей и методов интеллектуального планирования с методами, применяемыми в ИЭС [Рыбина и др., 2011]. В фокусе внимания данной работы находится новый этап исследований, связанный с развитием ИП и других компонентов интеллектуальной среды.

1. Некоторые аспекты современных исследований в области интеллектуального планирования

Под *интеллектуальным планированием* в настоящее время чаще всего подразумевается автоматическая генерация планов некоторой программно-аппаратной системой, однако сам термин «интеллектуальное планирование» не имеет однозначного определения. Например, в русскоязычной литературе наиболее употребляемым является термин «интеллектуальное планирование» [Осипов, 2011], а в англоязычной – «автоматическое планирование» (automated planning) [Nau, 2007], причем в обоих случаях подразумевается собственно процесс генерации плана вычислительной машиной. Соответственно, в контексте интеллектуального планирования *план* – это некоторое представление будущего поведения, в частности, план обычно представляет собой набор действий с некоторыми ограничениями (например, темпоральными) для исполнения некоторым агентом или агентами [Nau, 2007].

Под *планировщиком*, практически, однозначно понимается программная система, осуществляющая генерацию планов на основе формального описания среды, начального состояния среды и поставленных перед планировщиком целей. В некоторых работах планировщик называется агентом [Осипов, 2011], в других случаях в состав планировщика входит также *проектировщик*, осуществляющий распределение ресурсов для выполнения построенных планов.

В области интеллектуального планирования к настоящему времени разработано значительное число методов, подходов, формализмов и т.д., среди которых следует выделить: планирование с помощью пропозициональной логики; планирование в пространстве планов; планирование в пространстве состояний; планирование как задача удовлетворения ограничений; планирование на основе прецедентов; трансляция в другую проблему; темпоральное планирование; планирование в недетерминированных и вероятностных областях; иерархическое планирование (HTN-формализм) и др. Детальные обзоры можно найти в работах [Осипов, 2011; Nau, 2007; Rintanen, 2005] и др., а ниже выделены только некоторые особенности наиболее востребованных подходов.

Планирование как задача удовлетворения ограничений. Задача планирования формулируется как задача удовлетворения ограничений (CSP-задача) [Осипов, 2011], на основе чего был разработан известный алгоритм GraphPlan, использующий технику прямого распространения ограничений и обеспечивающий перевод описания задачи планирования в специальную структуру – *граф планирования*, из которого впоследствии извлекается частично-упорядоченный план.

Планирование в пространстве состояний. Данный подход, основанный на эвристическом поиске, был успешно реализован в известной системе STRIPS, однако наибольший интерес возник вследствие разработки эффективных эвристических функций, например $h(LM)$, $h(FF)$, $h(CG)$ [Helmert, 2004], а также Reachability Heuristic и Planning Graph Heuristic [Bryce et al., 2007], основанных на GraphPlan. В настоящее время поиск в пространстве состояний является наиболее популярным подходом в интеллектуальном планировании, в частности, на его основе реализован планировщик TFD / LAMA [Richter et al., 2011].

Преобразование в другие задачи. Здесь осуществляется преобразование исходной задачи планирования в другие задачи, имеющие эффективные решения, например, целочисленное программирование, CSP-задача, системы линейных уравнений и др. Этот подход реализован в целом ряде планировщиков, в том числе COLIN (COntinuous LINear) и OPTIC [Benton et al., 2012].

Темпоральное планирование. В отличие от классического планирования, где используется достаточно тривиальная модель времени, в темпоральном планировании [Benton et al., 2012] за счет исполь-

зования более сложных моделей времени учитываются такие параметры, как продолжительность действий; параллельность планов; ресурсы; динамика изменения среды; привязка целей ко времени и др.

Планирование в недетерминированных и вероятностных областях. Наибольший интерес к исследованиям в данных областях связан с разработкой языка PPDDL (Probabilistic Planning Domain Definition Language) и пришедшего ему на смену языка Relational Dynamic Influence Diagram Language (RDDL).

Иерархическое планирование (HTN-формализм). Исследования в области HTN-планирования имеют более прикладной характер, в частности, планировщики OPlan, SIP2, SHOP2 [Nau et al., 2003] и др. сегодня широко используются в различных предметно-ориентированных HTN-системах планирования. HTN-формализм также хорошо комбинируется с другими методами интеллектуального планирования.

Кроме вышеперечисленных подходов достаточно известными являются исследования в области планирования на основе прецедентов, с помощью пропозициональной логики, в пространстве планов и др. [Осипов, 2011]. Следует также отметить, что своеобразным языковым стандартом в области интеллектуального планирования является язык PDDL [McDermott, 1998], для которого уже созданы три основные версии.

Методы интеллектуального планирования находят широкое применение в целом ряде прикладных задач. Не претендуя на полноту обзора, выделим лишь наиболее востребованные области применения интеллектуального планирования: управление автономными роботами [Nau, 2008]; композиции веб-сервисов [Zou et al., 2012]; автоматизированное обучение (в частности, для построения индивидуальных планов обучения [Garrido et al., 2012]); калибровка оборудования [Parkinson et al., 2012]; ресурсно-календарное планирование [Бурков и др., 2009]; оптимальное использование ресурсов во времени [Coles, 2012] и др.

Однако в целом применение методов интеллектуального планирования для автоматизированной поддержки процессов построения интеллектуальных систем является слабоисследованной областью, и здесь можно сослаться в основном на опыт, накопленный в ходе создания прикладных ИЭС, в частности, разработки и использования обучающих ИЭС и динамических ИЭС [Рыбина, 2011].

2. Особенности интеллектуальной программной среды для поддержки построения интегрированных экспертных систем

Полное формальное описание модели интеллектуальной программной среды и методов реализации отдельных компонентов представлено в [Рыбина, 2008], поэтому здесь приведем краткое описание этой модели в виде четверки $M_{AT} = \langle KB, K, P, TI \rangle$, где KB – технологическая база знаний (БЗ) о составе проекта и типовых проектных решениях, используемых при разработке ИЭС; K – множество текущих контекстов K_r , состоящих из множества объектов из KB , редактируемых или выполняющихся на текущем шаге управления; P – интеллектуальный планировщик, управляющий процессами разработки и тестирования ИЭС; TI – множество инструментов, применяющихся на различных этапах жизненного цикла (ЖЦ) разработки ИЭС.

Декларативной основой процесса интеллектуальной поддержки разработки ИЭС является компонент KB , выступающий в качестве информационного хранилища в данной среде и определяемый как $KB = \langle WKB, SKB, PKB \rangle$, где WKB – это БЗ, содержащая знания о типовых проектных процедурах (ТПП), описывающих последовательности и способы применения инструментальных средств, а также последовательности этапов создания ИЭС; SKB – это БЗ, включающая знания об использовании ТПП и повторно-используемых компонентов (ПИК), включая фрагменты созданных ранее прототипов ИЭС; PKB (опционально) – представляет собой БЗ, содержащую специфические знания для решения задач, требующих нестандартного подхода.

Текущий контекст K_i представляется в виде совокупности $K_i = \langle KD, KP \rangle$, где KD – декларативный контекст, предназначенный для хранения общей информации о структуре проекта, инженере по знаниям и текущем пользователе; KP – процедурный контекст, включающий в себя текущие объекты, явно влияющие на дальнейшие шаги ИП (этап ЖЦ, редактируемый объект, цель, исполнитель, план разработки и т.д.).

Основной процедурный (операционный) компонент P – в общем виде описывается как $P = \langle SK, AF, Pa, Pb, I, GP \rangle$, где SK – состояние текущего контекста, при котором активизируется ИП; AF – множество функциональных модулей, входящих в состав ИП;

Pa – процедура выбора текущей цели на основании плана разработки; *Pb* – процедура выбора наилучшего функционального модуля-исполнителя из списка возможных кандидатов; *I* – процедуры, обеспечивающие интерфейс с соответствующими компонентами комплекса АТ-ТЕХНОЛОГИЯ; *GP* – процедуры работы с планами разработки ИЭС.

Перечислим основные цели, для достижения которых необходимо использовать знания различных типов в процессе разработки прототипа ИЭС: проверка ссылочной целостности проекта по разработке ИЭС; автоматизированное построение диаграммы компонентов; синтез макета архитектуры прототипа ИЭС; планирование последовательности шагов по созданию прототипа ИЭС с учетом специфики и особенностей решаемых задач (генерация планов разработки, автоматизированный синтез макета архитектуры, контроль исполнения планов); определение множества наиболее актуальных подзадач для каждого из этапов в ходе разработки прототипа ИЭС и др.

Задача построения планов разработки прототипа прикладной ИЭС с помощью ИП требует привлечения самых разных знаний о моделях и методах решения типовых задач [Рыбина, 2008], о технологии проектирования и разработки ИЭС, о способах интеграции с внешними базами данных и пакетами прикладных программ и др. В качестве *проекта* по разработке ИЭС выступает хранящаяся в некотором формате совокупность знаний и данных о решаемой задаче, на основе которых под управлением ИП осуществляется процесс прототипирования прикладной ИЭС, а под *макетом архитектуры* прототипа ИЭС понимается множество взаимосвязанных ПИК и иных подсистем, обеспечивающих решение данной задачи.

Важное место в контексте управления процессами построения прототипов ИЭС занимают вопросы, связанные с методами, алгоритмами и программными средствами реализации текущей версии ИП, обеспечивающего поддержку следующей функциональности: генерация *обобщенного* и *детального* планов разработки как решение задачи планирования в пространстве состояний (представленной на языке PDDL); синтез макета архитектуры прикладной ИЭС (на основе знаний, описанных на языке представления знаний [Рыбина, 2008]); визуализация планов разработки; исполнение планов разработки в автоматизированном режиме. Обобщенный и

детальный планы разработки прототипа ИЭС связаны между собой на основе HTN-формализма. Поскольку построение детального плана разработки ИЭС является задачей высокой вычислительной сложности (аналогично любой сложной задаче интеллектуального планирования), то генерация детального и обобщенного планов выполняются отдельно.

Ниже описываются некоторые особенности реализации вышеперечисленных функциональных возможностей ИП, который интегрируется с внешними PDDL-планировщиками с целью решения сформулированной в терминах языка PDDL 2.2 [McDermott, 1998] задачи генерации планов разработки ИЭС.

3. Общее описание архитектуры интеллектуального планировщика

В текущей версии ИП реализованы следующие основные компоненты (рис.1): компонент синтеза макета архитектуры, визуализатор плана, компонент взаимодействия с инженером по знаниям, компонент генерации планов разработки ИЭС и др. Рассмотрим особенности реализации наиболее важных компонентов.

Визуализатор плана. Предназначен для инженера по знаниям с целью визуализации текущих планов и инициирования выполнения определенной задачи плана. Визуализация осуществляется в текстовом и графическом виде, причем с помощью графической визуализации для максимальной наглядности план отображается в виде двумерной графической сцены (с использованием библиотеки QT).

Компонент синтеза макета архитектуры. Производится синтез начального макета архитектуры на основе модели архитектуры прототипа ИЭС (представленной в виде иерархии расширенных диаграмм потоков данных), синтез фрагментов архитектуры на основе выполнения задач плана, а также слияние нового фрагмента с текущим макетом.

Компонент генерации плана. Обеспечиваются процессы генерации обобщенных планов (на основе модели архитектуры ИЭС) и детальных планов (с использованием текущего макета архитектуры ИЭС и обобщенного плана). Реализация данных процессов осуществляется с помощью взаимодействия с внешними PDDL-планировщиками через компонент интеграции.

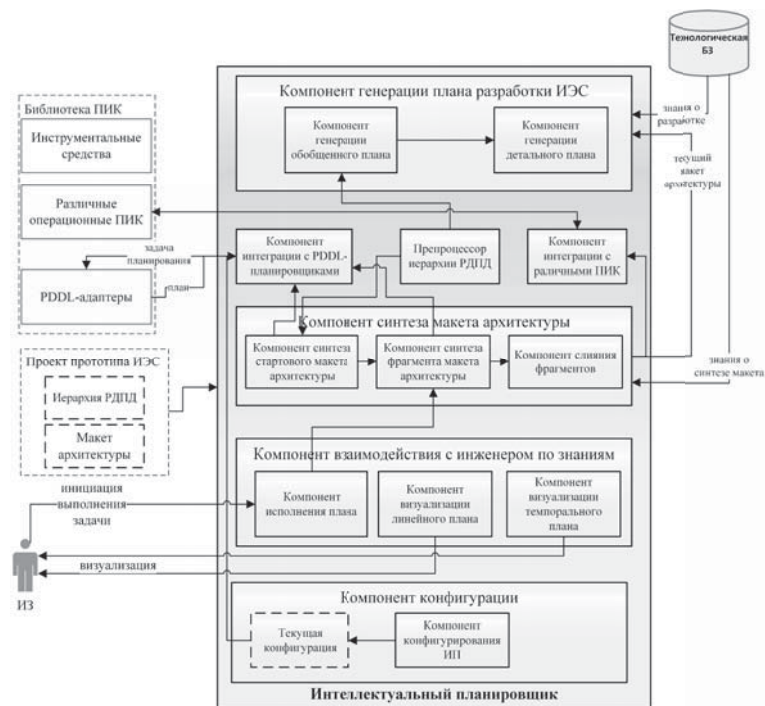


Рис. 1. Архитектура интеллектуального планировщика комплекса АТ-ТЕХНОЛОГИЯ

Заключение

Проведено экспериментальное программное исследование текущей версии ИП при прототипировании обучающих ИЭС по различным курсам/дисциплинам. Кроме того, осуществлялась интеграция ИП с несколькими PDDL-планировщиками, в частности LAMA/TFD. В перспективе планируется расширение возможностей ИП для использования в условиях коллективной разработки прототипов ИЭС при ограниченных ресурсах.

СПИСОК ЛИТЕРАТУРЫ

[Бурков и др., 2009] Бурков В.Н., Коргин Н.А., Новиков Д.А. Введение в теорию управления организационными системами / Под ред. чл.-корр. РАН Д.А. Новикова. – М.: Либроком, 2009.

[Осипов, 2011] Осипов Г.С. Методы искусственного интеллекта. – М.: ФИЗМАТЛИТ, 2011.

[Рыбина, 2008] Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. – М.: Научтехлитиздат, 2008.

[Рыбина, 2011] Рыбина Г.В. Практическое применение задачно-ориентированной методологии построения интегрированных экспертных систем (обзор приложений в статических и динамических проблемных областях) // Приборы и системы. Управление, контроль, диагностика. 2011. №12.

[Рыбина и др., 2011] Рыбина Г.В., Иващенко М.Г., Блохин Ю.М. Интеллектуальная технология построения интегрированных экспертных систем // Искусственный интеллект и принятие решений. 2011. №3.

[Benton et al., 2012] Benton J., Coles A., Temporal Planning with Preferences and Time-Dependent Continuous Costs. // Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, 2012.

[Bryce et al., 2007] Bryce D., Kambhampati S. A Tutorial on Planning Graph-Based Reachability Heuristics // AI Magazine. 2007.

[Coles, 2012] Coles A. Opportunistic Branched Plans to Maximise Utility in the Presence of Resource Uncertainty // Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012). – N.:IOS Press, 2012.

[Garrido et al., 2012] Garrido A., Morales L., Serina I. Using AI Planning to Enhance E-Learning Processes // Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling. – Canada.: AAAI Press, 2012.

[Helmert, 2004] Helmert M. A planning heuristic based on causal graph analysis. // Proceedings of the 14th ICAPS. 2004.

[McDermott, 1998] PDDL – the planning domain definition language. – <http://www.informatik.uni-ulm.de/ki/Edu/Vorlesungen/GdKI/WS0203/pddl.pdf>.

[Nau, 2007] Nau D.S. Current trends in automated planning // AI Magazine. 2007. Vol. 28. No 4.

[Nau et al., 2003] Nau D.S., Au T., Ilghami O., et al. SHOP2: An HTN Planning System // Journal of Artificial Intelligence Research №20. 2003.

[Parkinson et al., 2012] Parkinson S., Longstaff A., Crampton A., Gregory. The Application of Automated Planning to Machine Tool Calibration. // Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling. – Canada.: AAAI Press, 2012.

[Richter et al., 2011] Richter S., Westphal., Helmert. LAMA 2008 and 2011. // Short paper for the International Planning Competition 2011 – 2011.

[Rintanen, 2005] Rintanen J. Introduction to Automated Planning (draft). – Germany, 2005.

[Zou et al., 2012] Zou G., Chen Y., Xu Y., Huang R., Xiang Y. Towards Automated Choreographing of Web Services Using Planning // Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence. – Canada.: AAAI Press, 2012.

УДК 004.89

**ПОДХОД К РЕАЛИЗАЦИИ
ТЕМПОРАЛЬНОГО ВЫВОДА
В ДИНАМИЧЕСКИХ ИНТЕГРИРОВАННЫХ
ЭКСПЕРТНЫХ СИСТЕМАХ¹**

Г.В. Рыбина (*galina@ailab.mephi.ru*)

А.В. Мозгачев (*avmozgachev@mephi.ru*)

Национальный исследовательский ядерный университет «МИФИ»,
Москва

В работе обсуждаются научные и технологические проблемы построения динамических интегрированных экспертных систем и подходы к их решению. Рассматривается развитие задачно-ориентированной методологии построения интегрированных экспертных систем в контексте построения динамических интегрированных экспертных систем. Особое внимание уделяется вопросам представления и обработки темпоральных знаний.

Введение

В последние годы возрастает потребность в использовании динамических интеллектуальных систем (ДИС), в частности, динамических интегрированных экспертных систем (ИЭС) во всех значимых областях человеческой деятельности. При этом величина эффекта от создания и применения систем данного класса во многом зависит от наличия мощных современных инструментальных средств поддержки разработки [Рыбина, 2008]. К настоящему времени сформировался круг общих научных и технологических проблем, препятствующих широкому распространению приложений динамических ИЭС в стратегически важных проблемных областях (ПрО), где возможен наиболее высокий эффект от их применения [Рыбина, 2008], [Рыбина и др., 2014]:

1. Трудности получения темпоральных знаний (т.е. знаний, рассматривающих время в качестве сущности предметной области) из различных источников знаний (эксперты, тексты, базы данных и др.) для динамического представления предметной области.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-01-00467).

2. Сложность разработки формализмов для динамического представления предметной области, что определяется переменным составом сущностей предметной области, изменением во времени входных данных, поступающих от внешних источников (модели внешнего мира), и необходимостью структурирования, хранения и анализа изменяющихся во времени данных.

3. Сложность процессов решения динамических задач, что связано с выполнением одновременных временных (темпоральных) рассуждений о нескольких различных асинхронных процессах (задачах), наличием ограниченных ресурсов (время, память), изменением состава знаний и данных в процессах решения задач.

4. Проблемы моделирования внешнего мира (внешнего окружения) и различных его состояний в реальном времени на всех этапах проектирования и разработки системы до внедрения.

5. Высокая стоимость зарубежных инструментальных программных средств поддержки разработки и сопровождения ДИС различных классов и практическое отсутствие отечественного инструментария для этих целей.

6. Необходимость наличия специальных программно-аппаратных средств сопряжения с внешним миром (наличие датчиков, контроллеров и др.).

Обозначенные выше проблемы в значительной степени определяют большую сложность разработки динамических ИЭС. Кроме того, в настоящее время не предложено универсального подхода, позволяющего комплексно решать описанные проблемы (или часть из них), что подразумевает разработку единой целостной методологии и технологии создания систем подобного уровня сложности на всех этапах жизненного цикла. Современные коммерческие инструментальные средства для поддержки построения большинства классов ДИС (*G2*, *RTworks*, *RTXPS* и др.) при всей своей мощности и универсальности не способны комплексно решать вышеперечисленные задачи в рамках единой методологии [Попов и др., 1996], [Рыбина и др., 2010].

Значительным шагом к построению необходимой методологии стал новый этап в развитии теории и технологии построения ИЭС на основе задачно-ориентированной методологии (ЗОМ), основные положения которой были предложены и экспериментально исследованы Г.В. Рыбиной в 90-х годах прошлого века [Рыбина, 2008].

Несколько лет назад был начат цикл работ, связанных с развитием ЗОМ и поддерживающего ее комплекса АТ-ТЕХНОЛОГИЯ с целью обеспечения возможности построения динамических ИЭС, использующих динамические представления предметной области и решающих динамические задачи [Рыбина и др., 2010], [Рыбина и др., 2013], [Рыбина и др., 2014].

Анализ приведенных проблем показывает, что наименее исследованными из них с точки зрения системного подхода и создания целостной методологии построения динамических ИЭС являются тесно взаимосвязанные подходы к получению, представлению и обработке временных (темпоральных) знаний [Рыбина, 2008].

1. Модели и методы темпорального вывода в динамических интегрированных экспертных системах

Представление временных зависимостей и анализ различных моделей и методов представления времени занимают значительное место в современных исследованиях в области ДИС, в частности, этому в той или иной степени посвящены работы [Shoham, 1987], [Shoham et al., 1988], [Allen, 1991], [Spranger, 2002], [Еремеев и др., 2004], [Стефанюк, 2007], а также работы по управлению во времени [Moore, 1993], [Осипов, 2008], [Осипов, 2011] точечной линейной моделью времени [Еремеев и др., 2004], [Еремеев и др., 2009], интервальным временным логикам [Allen, 1983], [Плесневич, 1999], [Еремеев и др., 2012], ветвящемуся времени [Ladkin et al., 1990], [Еремеев, 2006], а также работы [Рыбина и др., 2010], [Рыбина и др., 2013], в которых авторами были приведены аргументы об оправданности использования интервальной логики Аллена [Allen, 1983] и управления во времени [Осипов, 2008] в динамической версии инструментального комплекса АТ-ТЕХНОЛОГИЯ.

Рассмотрим более детально предложенную модель темпорального вывода на продукционных правилах для динамических ИЭС (рис. 1), предусматривающую обработку знаний, содержащих темпоральные зависимости, совместно с базовыми знаниями о ПрО на основе универсальных средств вывода (АТ-РЕШАТЕЛЬ), реализованных в задачно-ориентированной методологии [Рыбина и др., 2007], [Рыбина и др., 2013].

Темпоральный вывод – процесс рассуждений, учитывающий временные и причинно-следственные соотношения между объек-

тами предметной области, а также возможность изменения в процессе работы состава знаний и данных. Особенности темпорального вывода на продукционных правилах в динамических ИЭС являются работа по тактам, что означает получение новых данных на каждом такте работы, и необходимость выполнения вывода также на каждом такте.

Модель, соответствующая представленному на рис. 2 механизму вывода, является обобщением базовой модели [Рыбина и др., 2007], [Рыбина, 2008] для случая темпорального вывода, т.е. формально данная модель темпорального вывода может быть представлена следующим образом: $I' = \langle A, S'', K, W', D \rangle$, где A – процесс выбора активных правил из базы знаний (БЗ), необходимых для решения задачи в очередном цикле работы интерпретатора, т.е. подмножества БЗ, правила из которого могут быть применены, если пройдут этап сопоставления; S'' – процесс сопоставления посылок активных правил с состоянием рабочей памяти, включающий построение интерпретации модели развития событий, сопоставление темпоральных и нетемпоральных частей правил, а также различные преобразования НЕ-факторов знаний (неопределенность, неточность, нечеткость), в том числе фаззификацию [Рыбина и др., 2007], [Рыбина, 2008]; K – процесс разрешения конфликтов, определяющий выбор применяемого на текущем шаге вывода правила из списка успешно прошедших сопоставление; W' – процесс выполнения следствия выбранного при сопоставлении правила с вычислением истинности следствия, учитывая НЕ-факторы; D – процесс дефаззификации фаззифицированных в ходе вывода значений параметров.

Следовательно, в отличие от развития классической модели вывода в контексте вывода на недоверенных знаниях [Рыбина и др., 2007], [Рыбина, 2008], этап сопоставления дополняется процессами, обеспечивающими модификацию интерпретации модели развития событий и сопоставление темпоральных частей посылок активных правил. Следует также отметить, что в случае динамических ИЭС вывод осуществляется независимо на каждом такте работы, причем общей частью является интерпретация модели развития событий, модифицирующаяся в рамках каждого такта. При интеграции механизмов базового и темпорального выводов особое внимание уделяется таким аспектам вывода, как реализация опе-

вала); логические связки: \sim (отрицание), $\&$ (конъюнкция), \vee (дизъюнкция); связки Аллена: b (*before* – раньше), a (*after* – позднее), m (*meets* – встречается), o (*overlaps* – перекрывает), s (*starts* – начинает), d (*during* – во время), e (*is equal* – равно, точнее, синхронно с), f (*finishes* – заканчивает); связки сравнения: $>$ (больше), $<$ (меньше), $=$ (равно); целые числа; левая и правая скобки: (и). Элементарными формулами модифицированной логики Аллена являются: выражения $X r Y$, где X, Y – интервалы и r – любая связка Аллена; выражения $X r Y$, где X, Y – события и r – связка Аллена из множества $\{b, a, e\}$; выражения $X r Y$, где X – событие, Y – интервал и r – связка Аллена из множества $\{b, a, s, d, f\}$; выражения $XN r N$, где X – переменная, N – характеристика, r – связка сравнения, N – целое число. В качестве формул модифицированной логики Аллена выступают: переменные; элементарные формулы; если f и g – формулы, то формулами также являются: $\sim f$, $(f \& g)$, $(f \vee g)$.

Таким образом, модель развития событий задается совокупностью темпоральных объектов (событий и интервалов), а локальная модель развития событий в правиле – формулами логики Аллена. Интерпретация модели развития событий задается временами возникновения событий и временами начал и окончаний интервалов. Применение модифицированной логики Аллена предполагает использование дискретного времени, т.е. изменения в ПрО происходят по тактам.

Использование механизмов управления во времени, т.е. применение определенных действий, исходя из изменения состояний ПрО без ввода сущности времени, опирается на [Moore, 1993], [Осипов, 2008], [Осипов, 2011] и подразумевает реализацию различных способов активации правил, позволяющих реагировать на конкретные наблюдаемые события и интервалы.

Для представления темпоральных знаний на основе базового языка представления знания (ЯПЗ) комплекса АТ-ТЕХНОЛОГИЯ [Рыбина, 2008] был разработан обобщенный ЯПЗ для динамических ИЭС, позволяющий представлять темпоральные знания, основанные на модифицированной интервальной логике Аллена и логике управления во времени, совместно с базовыми знаниями, в том числе содержащими знания с неопределенностью, неточностью и нечеткостью. Для этого были введены: новые типы объектов (событие и интервал), новые типы атрибутов (логические выражения для условий возникновения), изменение структуры посылки пра-

вил (добавление требований по локальной модели развития событий) и новые типы правил (обычное, периодическое, реакция).

Опишем кратко суть предложенного метода обработки темпоральных знаний, позволяющего производить вывод на основе знаний, содержащих временные (темпоральные) зависимости, путем проверки соответствия локальных моделей развития событий интерпретации глобальной модели развития событий в ПрО. Основными изменениями в рамках модели вывода выступают необходимость на этапе сопоставления выполнять формирование интерпретации развития событий и обработку темпоральной части посылок продукционных правил. При формировании интерпретации развития событий происходит привязка событий и интервалов, определенных в проблемной области, к временной оси путем идентификации фактов их возникновения на основании имеющихся в рабочей памяти данных и рассмотрения истории возникновения в прошлом. Отдельного внимания требуют случаи повторных возникновений и нестандартного расположения событий на временной оси (например, наблюдения признаков окончания интервала до его открытия). Обработка темпоральной части посылок продукционных правил опирается на результат формирования интерпретации развития событий в ПрО. Для активных правил осуществляется решение задачи проверки соответствия локальных моделей развития событий, описанных с помощью модифицированной логики Аллена и логики управления во времени, с построенной интерпретацией.

2. Программные средства темпорального вывода в динамических интегрированных экспертных системах

Результатом исследований стали разработка программных средств темпорального вывода (темпорального решателя), обеспечивающих глубинную интеграцию с универсальным АТ-РЕШАТЕЛЕМ [Рыбина, 2008], что позволяет осуществлять поддержку решения задач на основе динамических ИЭС как в статических, так и в динамических ПрО.

Темпоральный решатель, являющийся одним из новых компонентов динамической версии инструментального комплекса АТ-ТЕХНОЛОГИЯ [Рыбина, 2008] в процессе функционирования в соответствии с постановкой задачи темпорального вывода на продукционных правилах [Рыбина и др., 2014] и предъявляемыми к нему функциональными требованиями, выполняет решение двух

задач: построение интерпретации модели развития событий в ПрО и означивание темпоральной части продукционных правил. Архитектура темпорального решателя приведена на рис. 2.

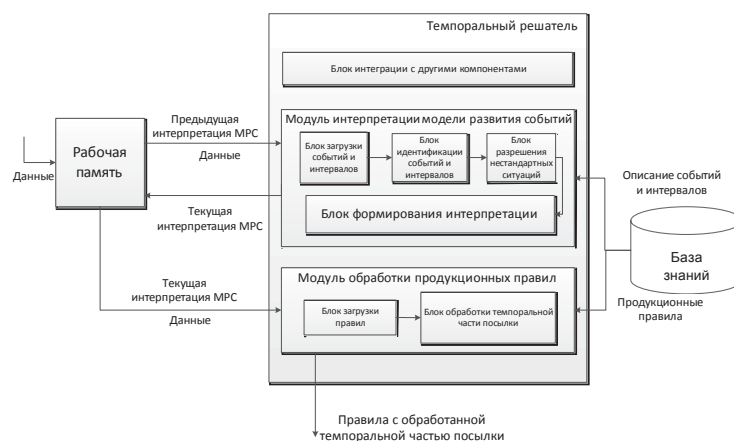


Рис. 2. Архитектура темпорального решателя

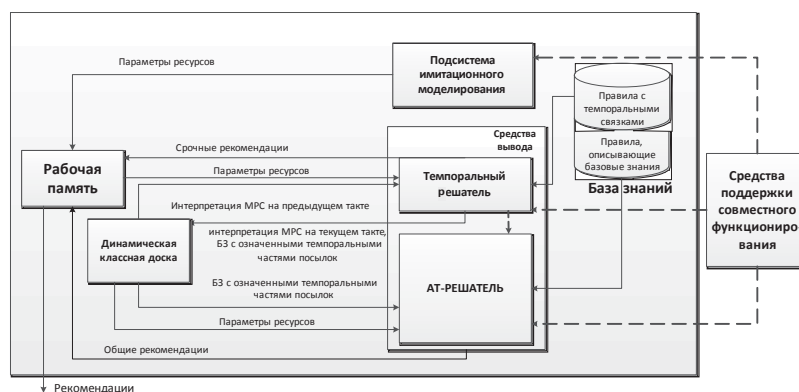


Рис. 3. Схема взаимодействия темпорального решателя, АТ-РЕШАТЕЛЯ и подсистемы имитационного моделирования

Кроме того, для динамической версии комплекса АТ-ТЕХНОЛОГИЯ было выполнено создание новых программных средств, обеспечивающих глубинную интеграцию методов и средств экспертных систем и имитационного моделирования в рамках единой архитектуры динамических ИЭС. На рис. 3 представлена

схема взаимодействия темпорального решателя, универсального АТ-РЕШАТЕЛЯ и подсистемы имитационного моделирования.

Взаимодействие темпорального решателя с АТ-РЕШАТЕЛЕМ и подсистемой имитационного моделирования обеспечивают средства поддержки совместного функционирования. Также компоненты взаимодействуют посредством общей рабочей памяти.

Заключение

Экспериментальные исследования показали преимущества разработанных инструментальных программных средств над аналогами по следующим критериям: мощность ЯПЗ, скорость работы, сокращение сроков разработки динамических ИЭС. В настоящее время выполняется разработка серии прототипов динамических ИЭС для конкретных задач с большой практической значимостью.

СПИСОК ЛИТЕРАТУРЫ

[Еремеев, 2006] Еремеев А.П. Логика ветвящегося времени и ее применение в интеллектуальных системах поддержки принятия решений // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006: Труды конференции в 3-х томах.– М.: Физматлит, 2006. Т. 3.

[Еремеев и др., 2004] Еремеев А.П., Троицкий В.В. Концепции и модели представления времени и их применение в интеллектуальных системах // Новости искусственного интеллекта. 2004. №1.

[Еремеев и др., 2009] Еремеев А.П., Куриленко И.Е. Компонента временных рассуждений для интеллектуальных систем поддержки принятия решения реального времени // Искусственный интеллект и принятие решений. 2009. №1.

[Еремеев и др., 2012] Еремеев А.П., Королев Ю.И. Сети Петри как инструмент для разработки интеллектуальных систем поддержки принятия решений реального времени // Гибридные и синергетические интеллектуальные системы: теория и практика. Материалы 1-го международного симпозиума – Калининград: Изд-во БФУ им. И. Канта, 2012, ч. 2.

[Осипов, 2008] Осипов Г.С. Динамические интеллектуальные системы // Искусственный интеллект и принятие решений. 2008. №1.

[Осипов, 2011] Осипов Г.С. Методы искусственного интеллекта. М.: ФИЗМАТЛИТ, 2011.

[Плесневич, 1999] Плесневич Г.С. Метод аналитических таблиц для логики событий // Труды Международной конференции «Интеллектуаль-

ное управление: новые интеллектуальные технологии в задачах управления (ICIT'99)». – Переславль-Залесский: ИПС, 1999.

[Попов и др., 1996] Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статические и динамические экспертные системы. Учебное пособие. – М.: Финансы и статистика, 1996.

[Рыбина, 2008] Рыбина Г.В. Теория и технология построения интегрированных экспертных систем. – М.: Научтехлитиздат, 2008.

[Рыбина и др., 2007] Рыбина Г.В., Демидов Д.В. Модели, методы и программные средства вывода в интегрированных экспертных системах // Инженерная физика. 2007. №2.

[Рыбина и др., 2010] Рыбина Г.В., Паронджанов С.С., Шанцер Д.И., Мозгачев А.В. Тенденции развития и применения современного программного инструментария для поддержки построения динамических интеллектуальных систем // Приборы и системы. Управление, контроль, диагностика. 2010. №11.

[Рыбина и др., 2013] Рыбина Г.В., Мозгачев А.В., Паронджанов С.С., Со Тиха Аунг. Динамические интегрированные экспертные системы: представление и обработка темпоральных знаний // Приборы и системы. Управление, контроль, диагностика. 2013. №6.

[Рыбина и др., 2014] Рыбина Г.В., Мозгачев А.В. Реализация темпорального вывода в динамических интегрированных экспертных системах // Искусственный интеллект и принятие решений. 2014. №1.

[Стефанюк, 2007] Стефанюк В.Л. Динамическая экспертная система и логическая проблема фрейма // Международная конференция по искусственному интеллекту (AIS'07/CAD-2007). – М.: Физматлит. 2007. Т.2.

[Allen, 1983] Allen J. Maintaining knowledge about temporal intervals // Communications of the ACM. 1983. N.26 (11).

[Allen, 1991] Allen. J. Time and time again: The many ways to represent time // Journal of Intelligent Information Systems. 1991. N. 6(4).

[Ladkin et al., 1990] Ladkin, P., Anger F., Rodriguez R. Temporal reasoning with intervals in branching time. TR-90-028, International Computer Science Institute, 1990.

[Moore, 1993] Moore B. Questions and Answers about G2 // Copyright, Gensym Corporation. 1993.

[Shoham, 1987] Shoham Y. Temporal logics in AI: Semantical and ontological considerations // Artificial Intelligence. 1987. N. 33(1).

[Shoham et al., 1988] Shoham Y. Goyal N. Representing time and action in AI. revised version of: Problems in formal temporal reasoning // Artificial Intelligence. 1988. N. 36(1).

[Spranger, 2002] Spranger S. Representation of Temporal Knowledge for Web-based Applications. – Munchen, 2002.

УДК 519.767.6

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ФОРМИРОВАНИЯ ПРОЕКТНЫХ КОМАНД НА ОСНОВЕ КОМПЕТЕНЦИЙ¹

Д.Ю. Тихомирова (*darya.tikhomirova@gmail.com*)

В.В. Шеркунов (*v.sherkunov@gmail.com*)

Ульяновский государственный технический университет, Ульяновск

В работе описываются онтологические методы поиска экспертов в научных организациях на основе их компетенций. Представлен метод формирования концептуальной модели квалификационного справочника путем семантического анализа текстового документа, в котором он содержится. Рассмотрен способ преобразования данных об образовании кандидата в список его компетенций. Приведен генетический алгоритм, использованный для получения наилучшего распределения экспертов. Дан краткий обзор разрабатываемой интеллектуальной системы.

Введение

В настоящее время во многих проектных организациях используются системы управления проектами, которые позволяют контролировать процесс реализации проекта. Но такие системы практически не охватывают начальные этапы проектирования. Одним из важных моментов на таких этапах является формирование проектной команды, от которой во многом зависит успешное выполнение проекта. Данная область не охватывается существующими системами управления проектами. Поэтому актуальной является разработка методов и средств, позволяющих формировать наиболее подходящую проектную команду.

Целью данной работы является создание алгоритмов и программного продукта, позволяющих автоматизировать процесс формирования проектных команд исходя из описания проекта, задач в рамках данного проекта, штатного расписания, должностных инструкций, квалификационного справочника, описания компетенций и информации об экспертах.

¹Работа выполнена при финансовой поддержке РФФИ (проекты № 14-01-31086 мол_а и № 14-01-31080 мол_а).

1. Методы формирования проектных команд

В данной работе понятие команда определяется как группа людей, взаимодополняющих и при необходимости взаимозаменяющих друг друга в ходе достижения поставленных целей. Обычно команда создается на период осуществления всего проекта либо одной из стадий его жизненного цикла [Лачинина, 2009].

Под проектной командой в соответствии с ГОСТ Р 54869–2011 понимается совокупность лиц, групп и организаций, объединенных во временную организационную структуру для выполнения работ проекта. Данная команда состоит из представителей различных групп, обладающих знаниями в конкретной предметной области или набором конкретных навыков и выполняющих работу по проекту [Руководство РМВоК, 2008].

В настоящее время существует несколько подходов к поиску экспертов для решения возникших задач с учетом их компетенции. Данные методы основываются на анализе текстовых документов, авторами которых являются потенциальные кандидаты в эксперты по определенному вопросу. Так, в работе [Крюков и др., 2013] описывается подход в определении компетенций с использованием онтологии и заключается в следующем. Предметная область представляется в виде онтологии с двумя типами вершин: вершины-темы и вершины-термина. Вершины-темы связаны между собой отношением тема-подтема, а вершина-термин связывается ровно с одной темой отношением тема-термин. Каждой теме соответствует набор терминов, которые характеризуют ее. Релевантность документа теме определяется использованием основных терминов этой темы, а компетентность сотрудника в теме определяется путем наличия публикаций, релевантных теме. Таким образом, определяется уровень компетентности сотрудника заданной теме, на основе которого может приниматься решение о выборе данного сотрудника в качестве эксперта.

В работе [Панкова и др., 2011] также предлагается онтологический подход при определении уровня компетенции сотрудников. Данный подход заключается в формировании профиля документа и профиля специалиста. Профиль документа определяется как множество концептуальных моделей документа в некоторой области знаний. Профиль специалиста в некоторой области знаний опре-

деляется по множеству профилей текстовых документов в данной области знаний. Тематическая близость специалиста к документу определяется семантической близостью профиля специалиста к документу в некоторой области знаний. Для вычисления близости профилей документа и специалиста используется трехэтапная модель:

1. Вычисляются близости между каждой парой понятий из сравниваемых профилей;
2. Определяются близости между каждым понятием из профиля документа и профилем специалиста – частные близости;
3. Частные близости агрегируются для вычисления близости между двумя профилями.

Данный способ позволяет подбирать специалистов заданной тематике на основе их метаописаний текстовых документов.

2. Онтологическое представление квалификационного справочника

В данной работе основным источником данных для определения компетенции является квалификационный справочник должностей руководителей, специалистов и других служащих. Данный справочник содержит основные квалификационные характеристики, утвержденные постановлением Минтруда России от 21.08.1998 №37. Квалификационные характеристики являются нормативными документами, предназначенными для обоснования рационального разделения и организации труда, правильного подбора, расстановки и использования кадров, обеспечения единства при определении должностных обязанностей работников и предъявляемых к ним квалификационных требований, а также принимаемых решений о соответствии занимаемым должностям при проведении аттестации руководителей и специалистов.

Данный справочник имеет определенную структуру в описании квалификационных характеристик должностей и разделяет их на три группы:

- должностные обязанности;
- требования к навыкам и знаниям;
- требования к квалификации.

Помимо этого, свидетельствами компетенции могут служить должностные инструкции, которые определяют основные навыки и знания определенных должностей в организации. Список сотрудников, доступных для формирования команды, определяется штатным расписанием организации.

Для определения соответствия сотрудника на занимаемую им должность, а также дальнейшего определения необходимых компетенций необходимо сформировать онтологию квалификационного справочника.

Концептуальная модель квалификационного справочника формируется путем синтаксического анализа текстового документа данного справочника. Для синтаксического анализа используется квалификационный справочник в формате PDF, из которого извлекаются должности, путем перебора и фильтрации закладок документа, не принимая во внимание закладки общего значения. Далее, учитывая структуру справочника, для каждой должности извлекаются соответствующие разделы описания квалификационных характеристик. Для этого выбирается текст между двумя должностями, расположенными в порядке следования в документе, и делится на группы, соответствующие формату справочника. Для выделения группы должностных обязанностей из полученного текста извлекается фрагмент между разделами «Должностные обязанности» и «Требования к навыкам и знаниям» (в справочнике указателем на раздел служит словосочетание «Должен знать»), соответственно, для выделения требований к навыкам и знаниям извлекаем фрагмент между разделом «Требования к навыкам и знаниям» и «Требования к квалификации», а для требований к квалификации – фрагмент между разделом «Требования к квалификации» и окончанием текста. После чего каждый фрагмент делится на предложения и записывается в файл OWL в виде утверждений соответствующих концептуальной схеме. Фрагмент полученных утверждений, записанных в файл, представлен ниже:

```
<Declaration>
  <ClassIRI=»#Должность»/>
</Declaration>
<Declaration>
  <ClassIRI=»#Должностные_обязанности»/>
</Declaration>
```

```

<Declaration>
  <Class IRI=»#Знания»/>
</Declaration>
<Declaration>
  <Class IRI=" #Требования_к_квалификации" />
</Declaration>
<Declaration>
  <ObjectProperty IRI=»#имеет»/>
</Declaration>
<Declaration>
  <DataProperty IRI=»#включают»/>
</Declaration>
....
<ObjectPropertyAssertion>
  <ObjectProperty IRI=»#имеет»/>
  <NamedIndividual IRI=" #Главный_конструктор_
    проекта" />
<NamedIndividual
  IRI=" #Должностные_обязанности_главного_конструктора_
    проекта" />
</ObjectPropertyAssertion>
<DataProperty IRI=" #включают" />
  <NamedIndividual
  IRI=" #Должностные_обязанности_главного_конструктора_
    проекта" />
  <Literal
    datatype IRI=" http://www.w3.org/2001/
    XMLSchema#string">Обеспечивает соблюдение требований
    и нормативов по организации труда при проектировании
    новых и реконструкции действующих предприятий,
    разработке технологических процессов и оборудования,
    охраны окружающей среды</Literal>
</DataPropertyAssertion>

```

3. Разрабатываемая система формирования проектных команд

В настоящее время ведется разработка по созданию автоматизированной системы подбора и составления оптимального состава проектной команды на основе компетенций. Данная система подразумевает использование методов онтологического анализа

и генетического алгоритма, для оптимизации проектной команды на каждом этапе жизненного цикла проекта, путем анализа данных квалификационного справочника, должностных инструкций и штатного расписания, представленных в виде онтологий и RDF-файлов (рис. 1).

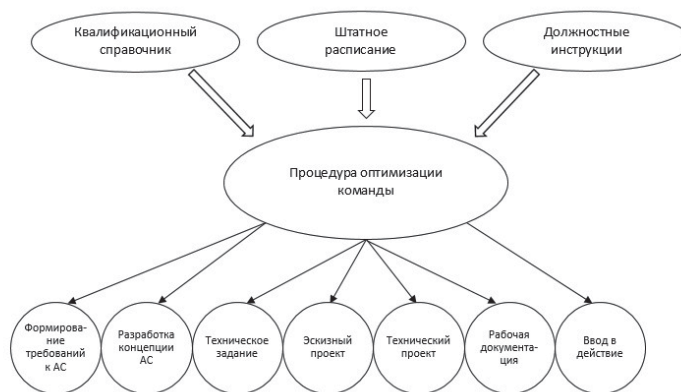


Рис 1. Общая модель разрабатываемой системы

Также в настоящее время ведутся исследования по выявлению задач, необходимых для решения в рамках проекта, для которых и необходим подбор соответствующих специалистов. Данные задачи планируется выделять из технического задания. Методика выделения основных функций из технического задания, реализуемых в проекте, описана в работе [Орлова, 2010].

Первым этапом работы системы является преобразование исходных данных к требуемому формату: создание онтологического представления квалификационного справочника, представление штатного расписания и списка кандидатов в RDF.

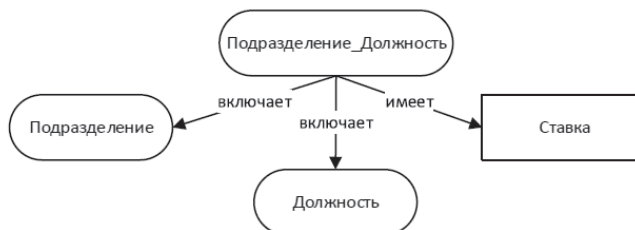


Рис 2. RDF-схема штатного расписания организации

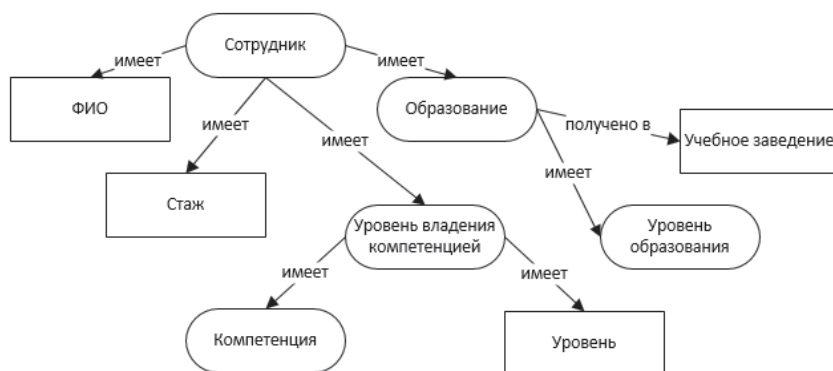


Рис 3. RDF-схема карточки эксперта

Для выполнения анализа соответствия кандидата выделенным позициям проектной команды необходимо преобразовать информацию об уровне владения эксперта различными компетенциями в конкретные знания и умения. На данном шаге система собирает необходимые данные из Excel-файлов описания компетенций и вычисляет степень владения навыками как среднее арифметическое степеней обладания навыками, полученными из описания всех компетенций кандидата.

Результатами выполнения данных этапов является список кандидатов и позиций, на которые нужно распределить экспертов. Данная процедура выполняется с помощью генетического алгоритма. Настраиваемыми параметрами алгоритма являются:

- Размер популяции p_size ;
- Массив весов $\{w_{st}, w_{kn}, w_{act}\}$;
- Количество элитных хромосом $elit$;
- Вероятность кроссинговера $cros$;
- Вероятность мутации mut ;
- Пороговое изменение функции приспособленности $delta$;
- Максимальное количество итераций алгоритма max_step .

Ограничением при распределении кандидатов является требуемый стаж. Также принято допущение, что распределение выполняется только целыми ставками и один сотрудник может занимать одновременно только одну должность.

В качестве функции приспособленности используется аддитивная функция, содержащая показатели стажа, умений и знаний кандидата:

$$f = w_{st} \cdot \frac{st - st_v}{st_v} + w_{kn} \sum (1 - kn) + w_{act} \sum (1 - act),$$

где w_{st} , w_{kn} , w_{act} – это веса коэффициентов стажа, знаний и навыков;

st , st_v – это стаж кандидата и требуемый стаж (стаж вакансии);

kn , act – это степени обладания требуемыми знаниями и навыками соответственно.

На первом шаге алгоритма формируется популяция указанного размера, хромосомами которой являются случайные распределения кандидатов. Для каждого распределения рассчитывается значение функции приспособленности, после чего рассчитывается «приспособленность» всего поколения как среднее арифметическое полученных результатов.

Во второе поколение выбирается указанное количество элитных хромосом. Оставшиеся особи формируются из особей первого поколения с помощью алгоритмов «скрещивания» и «мутации». После этого выполняется их проверка на соответствие ограничениям распределения. Если распределение корректно, то для него рассчитывается функция приспособленности. В обратном случае продолжаются поиск и обработка особей первого поколения.

Этот набор действий повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько циклов, пока не будет выполнен один из критериев остановки алгоритма:

- Изменение значения функции приспособленности меньше указанного порогового изменения;
- Пройдено максимальное количество итераций.

После выполнения одного из критериев остановки из последнего поколения выбирается распределение с минимальной функцией принадлежности. Оно считается наилучшим возможным распределением при заданных параметрах работы алгоритма.

По результатам работы алгоритма выводится график значений функции принадлежности в каждом поколении. Это позволяет проанализировать и скорректировать выбранные значения параметров генетического алгоритма.

Заключение

Представление квалификационного справочника в виде концептуальной модели позволит в дальнейшем применять к нему методы онтологического анализа и использовать методы интеллектуального анализа данных для решения задачи оптимизации проектной команды. В данной работе также представлен вектор дальнейших исследований в данной области по разработке методов и средств выделения задач на начальных этапах проектирования, которые необходимо решить в рамках проекта, и подбора соответствующих специалистов для их решения.

СПИСОК ЛИТЕРАТУРЫ

[Крюков и др., 2013] Крюков К.В., Кузнецов О.П., Суховеров В.С. О понятии формальной компетентности научных сотрудников // Материалы III Международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» (OSTIS-2013). Минск, 2013.

[Лачинина, 2009] Лачинина Т. Формирование команды в управлении проектами // Финансовая газета, региональный выпуск 17/2009.

[Орлова, 2010] Орлова Ю.А. Алгоритмическое обеспечение анализа текста технического задания и построения моделей программного обеспечения // Межвузовский сборник научных статей «Известия ВолГТУ». Серия «Актуальные проблемы управления, вычислительной техники и информатики в технических системах». 2010. №6.

[Панкова и др., 2011] Панкова Л.А., Пронина В.А., Крюков К.В. Онтологические модели поиска экспертов в системах управления знаниями научных организаций // Проблемы управления. 2011. №6.

[Руководство РМВоК, 2008] Руководство к своду знаний по управлению проектом, 4-е издание. – PMI, USA. 2008.

УДК 007.52:681.518.2; 004.056.5

ОСНОВЫ СОЗДАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОИСКА ВЕРОЯТНОГО НАРУШИТЕЛЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ¹

А.Е. Янковская (*ayuankov@gmail.com*)

Томский государственный архитектурно-строительный университет,
Национальный исследовательский Томский государственный
университет, Томский государственный университет
систем управления и радиоэлектроники, Томск

В.Г. Миронова (*mvg@security.tomsk.ru*)

Томский государственный университет систем управления
и радиоэлектроники, Томск

А.А. Шелупанов (*saa@keva.tusur.ru*)

Томский государственный университет систем управления
и радиоэлектроники, Томск

Приводятся основы создания интеллектуальной системы поиска вероятного нарушителя информационной безопасности. Интеллектуальная система предназначена для выявления субъектов, со стороны которых возможно нанесение негативного воздействия на информацию, и основана на матричном способе представления данных и знаний и тестовых методах распознавания образов.

Введение

Для обеспечения безопасности информации, обрабатываемой в информационных системах, используется система защиты информации (СЗИ). СЗИ должна быть спроектирована и внедрена на основе данных, полученных при анализе функционирования информационной системы, уязвимостей, нарушителей и угроз безопасности. Одним из этапов построения адекватной СЗИ является выявление вероятных нарушителей информационной безопасности (ИБ) и их классификация.

В основе обеспечения безопасности конфиденциальной информации (КИ) в информационной системе, используемой в той или

¹ Работа выполнена при финансовой поддержке РФФИ (проекты № 13-07-00373, № 14-07-00673 и № 13-07-98037) и РГНФ (проект № 13-06-00709).

иной организации, лежит СЗИ, основные стадии создания которой перечислены в [ISO/IEC 15408-1:2009], [ISO/IEC 17799:2000].

Первой стадией создания СЗИ является предпроектное обследование, то есть аудит ИБ – исследование текущего состояния ИБ информационных систем организации, описание которой представлено в вышеупомянутых нормативных документах и публикациях [Миронова, 2011а]. Аудит ИБ позволяет оценить текущий уровень защиты информации в информационных системах, выявить ее уязвимые места, определить нарушителей и сформировать рекомендации к созданию новой или модернизации старой СЗИ.

Выявление вероятных нарушителей безопасности КИ является одним из важных этапов проведения аудита ИБ. Модель вероятного нарушителя безопасности необходима для систематизации информации о типах и возможностях субъектов, целях несанкционированных воздействий и выработки адекватных организационных и технических мер противодействия субъектам, а также для количественной оценки вероятности угроз безопасности.

В статье предлагаются основы создания интеллектуальной системы поиска вероятного нарушителя (ИС ПВН) ИБ.

1. Поиск вероятного нарушителя в информационной системе. Основные понятия и определения

Поиск вероятного нарушителя предлагается осуществлять на основе тестового распознавания образов [Журавлёв и др., 1990] с использованием тестовых методов распознавания образов [Янковская, 2011а], реализованных в разрабатываемой нами ИС ПВН ИБ, основанной на матричном способе представления знаний, выявлении различного рода закономерностей, включая отказоустойчивые безызбыточные безусловные диагностические тесты (ОУББДТ) [Yankovskaya et al., 2012], [Янковская, 2009а], принятии итогового решения на основе процедуры голосования на множестве правил, построенных на основе ОУББДТ и обоснования результатов принятия решения с применением графических, включая когнитивные, средств [Янковская, 1998], [Янковская А.Е. 2007], [Янковская, 2009а], [Янковская, 2011а].

Для дальнейшего изложения введем используемые далее основные понятия и определения.

Под объектом будем понимать вероятного нарушителя ИБ, описываемого в пространстве характеристических признаков (ХП), в качестве которых выступают признаки, используемые специалистами по защите информации в ходе выявления возможных нарушителей ИБ, например, месторасположение нарушителя, наличие прав доступа к информации, наличие уязвимостей в системах, наличие сведений об информационной системе или СЗИ и др.

Будем считать, что все объекты рассматриваемой проблемной области можно разбить на конечное число образов, которым соответствуют различные итоговые решения, и построение ОУББДТ [Yankovskaya et al., 2012], [Янковская, 2009a] будем осуществлять на основе анализа обучающей выборки, заданной с применением матричного способа представления данных и знаний, включающего матрицы описаний (\mathbf{Q}) и различений (\mathbf{R}) [Янковская, 2011a].

Строки матрицы \mathbf{Q} сопоставляются обучающим объектам s_i ($i = \overline{1, N}$, где N – количество объектов), столбцы – ХП z_j ($j = \overline{1, M}$, где M – количество признаков, задающих описание каждого объекта). Элемент q_{ij} матрицы \mathbf{Q} задает значение j -го целочисленного признака для i -го объекта. Если значение ХП несущественно для объекта, то данный факт отмечается прочерком («–») в соответствующем элементе матрицы \mathbf{Q} .

Строки матрицы \mathbf{R} сопоставляются строкам матрицы \mathbf{Q} , столбцы – целочисленным классификационным признакам k_j ($j = \overline{1, L}$, где L – число механизмов классификации, разбивающих объекты из рассматриваемой проблемной области на классы эквивалентности). Элемент r_{ij} матрицы \mathbf{R} задает принадлежность i -го объекта некоторому классу по j -му механизму классификации путем указания номера класса.

Множество всех неповторяющихся строк матрицы различений сопоставлено множеству выделенных образов, представленных одностробцовой матрицей \mathbf{R}' , элементами которой являются номера образов. Данная модель не допускает пересечения объектов из разных образов. Элементами матрицы \mathbf{Q} служат значения ХП, часть которых перечислена в нормативных документах и публикациях [Базовая модель угроз, 2008], [Миронова, 2011a], [Миронова, 2011b], [Миронова, 2012].

Значения основных ХП, используемых при определении и описании нарушителя ИБ, перечислены ниже (через \mathbf{Z}_i обозначе-

но множество значений ХП): z_1 – территориальное расположение нарушителя ИБ относительно границ контролируемой зоны, $Z_1 = \{1,2\}$; z_2 – наличие у нарушителя доступа в систему, $Z_2 = \{1,2\}$; z_3 – наличие у нарушителя ИБ прав для осуществления легальных действий в ИС, $Z_3 = \{1,5\}$; z_4 – использование для атаки на систему и получения информации уязвимости, $Z_4 = \{1,4\}$; z_5 – наличие у нарушителя сведений о системе и/или СЗИ, $Z_5 = \{1,8\}$; z_6 – наличие документов, регламентирующих порядок действия пользователя в систему в части ИБ, $Z_6 = \{1,6\}$; z_7 – профессиональные качества нарушителя ИБ, $Z_7 = \{1,14\}$; z_8 – моральные качества, $Z_8 = \{1,13\}$; z_9 – депрессия, $Z_9 = \{1,5\}$; z_{10} – осложнение симптоматики депрессии, $Z_{10} = \{1,2\}$; z_{11} – эпизоды депрессии и психических расстройств в семейном анамнезе, $Z_{11} = \{1,2\}$; z_{12} – второй и последующие эпизоды депрессии, $Z_{12} = \{1,2\}$; z_{13} – наличие союзника, $Z_{13} = \{1,2\}$; z_{14} – алкоголизация, $Z_{14} = \{1,2\}$; z_{15} – особенности здоровья человека, $Z_{15} = \{1,8\}$; z_{16} – социальные факторы, $Z_{16} = \{1,11\}$; z_{17} – нравственные качества, $Z_{17} = \{1,10\}$.

На рис. 1 приведем иллюстрирующий пример описания матриц Q , R и R' , представляющих собой частичное описание знаний (используется только часть характеристического признакового пространства) для поиска вероятного нарушителя ИБ. Следует отметить, что рамки статьи не позволяют перечислить все ХП и их значения.

$$Q = \begin{matrix} & z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 & z_9 & z_{10} & z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} & z_{17} \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 1 \\ 3 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 1 \\ 1 \\ 3 \\ 4 \\ 1 \\ 1 \\ 4 \\ 8 \\ 3 \\ 1 \\ 1 \\ 8 \\ 2 \\ 1 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 3 \\ 5 \\ 11 \\ 3 \\ 5 \\ 11 \\ 14 \\ 13 \end{matrix} & \begin{matrix} 7 \\ 13 \\ 6 \\ 7 \\ 6 \\ 10 \\ 4 \\ 12 \\ 5 \\ 11 \\ 3 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 14 \end{matrix} & \begin{matrix} 14 \\ 8 \\ 4 \\ 8 \\ 9 \\ 3 \\ 4 \\ 5 \\ 2 \\ 11 \\ 3 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 13 \end{matrix} & \begin{matrix} 8 \\ 3 \\ 2 \\ 2 \\ 3 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \end{matrix} & \begin{matrix} 2 \\ 1 \\ 6 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \\ 2 \end{matrix} & \begin{matrix} 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 1 \\ 2 \end{matrix} & \begin{matrix} 2 \\ 2 \\ 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 8 \\ 7 \\ 3 \\ 1 \\ 1 \\ 8 \\ 6 \\ 6 \\ 7 \\ 8 \\ 8 \\ 8 \\ 8 \\ 8 \\ 8 \\ 10 \\ 11 \\ 1 \end{matrix} & \begin{matrix} 9 \\ 3 \\ 2 \\ 2 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 3 \\ 4 \\ 6 \\ 9 \\ 5 \\ 6 \\ 7 \\ 8 \\ 4 \\ 3 \\ 4 \\ 8 \\ 4 \\ 3 \\ 3 \\ 10 \\ 11 \\ 1 \end{matrix}
 \end{matrix}$$

$$R = \begin{matrix} & k_1 & k_2 & k_3 \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 1 \\ 1 \\ 2 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{matrix}
 \end{matrix}$$

$$R' = \begin{matrix} & k_1 & k_2 & k_3 \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \end{matrix}
 \end{matrix}$$

Рис. 1. Матрицы Q , R и R'

Для построения матрицы различий R выделим следующие классы и перечислим значения элементов класса: k_1 – возможность совершения деструктивных действий: 1 – не может, 2 – пограничное состояние (возможность совершения действий при условиях), 3 – может; k_2 – наличие условий для совершения деструктивных

действий: 1 – есть, 2 – отсутствуют; k_3 – вероятность совершения деструктивных воздействий: 1 – малая, 2 – низкая, 3 – средняя, 4 – высокая, 5 – очень высокая.

Тестом называется совокупность признаков, различающих любые пары объектов, принадлежащих разным образам. Диагностический тест (ДТ) называется безусловным, если порядок предъявления признаков не имеет значения, и условным, если задание каждого следующего признака зависит от значения предыдущего. ДТ называется безызбыточным, если удаление любого признака ведет к нарушению свойства ДТ быть тестом. Смешанный ДТ (СДТ) представляет собой оптимальное сочетание безусловных и условных составляющих. ОУББДТ представляет собой ББДТ, так же, как и ОУСДТ представляет собой СДТ, каждый из которых устойчив к заданному числу ошибок измерений.

ОУББДТ, так же, как и ББДТ, характеризуется одновременным предъявлением всех входящих в него признаков исследуемого объекта при принятии решений.

Под закономерностями в знаниях [Янковская, 2000] будем понимать следующие подмножества признаков: константные, устойчивые (константные внутри образа), неинформативные (не различающие ни одной пары объектов), альтернативные (в смысле включения в ДТ), зависимые (в смысле включения подмножеств различимых пар объектов), несущественные (не входящие ни в один ББДТ), обязательные (входящие во все ББДТ), псевдообязательные (входящие во множество используемых при распознавании ББДТ и не являющиеся обязательными) признаки, а также все минимальные и все (либо часть – при большом признаковом пространстве) безызбыточные различающие подмножества признаков, являющиеся, по сути, соответственно минимальными и безызбыточными (тупиковыми тестами [Журавлёв и др., 1990]) ДТ. К закономерностям будем относить и ОУББДТ, ОУСДТ, а также сигнальные признаки 1-го и 2-го родов, определения которых приведены в [Yankovskaya et al., 2013].

2. Основы создания интеллектуальной системы поиска вероятного нарушителя информационной безопасности

Вкратце изложим основы создания интеллектуальной системы (ИС) поиска вероятного нарушителя информационной безопасности.

Создаваемая нами ИС ПВН ИБ (ИС ДИВНАР) состоит из 3-х компонент: 1-я из которых (ИС ДИОС) предназначена для диагностики организационного стресса (ОС) обследуемого (субъекта) [Янковская, 2011b], 2-я (ИС ДИАПРОД) — для диагностики и профилактики депрессии [Янковская, 2012], [Янковская, 2014a], а 3-я, создаваемая нами ИС диагностики вероятного нарушителя ИБ (ИС ДИНАРЛОГ), – на основе результатов, полученных из первых 2 компонент и дополнительно сформированных ХП путем обследования субъекта, осуществляет поддержку принятия решения об отнесении субъекта к вероятному нарушителю. В отличие от гибридной ИС экспресс-диагностики ДИВНАР, блок-схема которой представлена в статье [Янковская, 2014b], рассматриваемая в данной статье ИС ДИВНАР ориентирована на выявление различного рода вышеупомянутых закономерностей.

Входными данными для ИС ДИОС и ИС ДИАПРОД являются результаты опроса обследуемого субъекта. Результаты диагностики субъекта по ИС ДИОС и ИС ДИАПРОД, так же, как и результаты, характеризующие субъекта и его способность нанести вред сегменту ИБ организации, поступают на вход ИС ДИНАРЛОГ.

В основу построения ИС ДИОС положена идея трехступенчатой диагностики ОС и математический аппарат, базируемый на пороговой и нечеткой логике, а также графических, включая когнитивные, средствах визуализации информационных структур и результатах обоснования диагностических и интервенционных решений. Подробное описание ИС ДИОС приведено в статье [Янковская, 2011b].

В основу создания ИС ДИАПРОД аналогично ИС ДИОС положен тот же самый математический аппарат. Для выявления депрессии и значений (форм) ее тяжести используются совокупность признаков и степень их выраженности, а основание для принятия диагностических решений по каждой степени тяжести депрессии и блок-схема работы ИС ДИАПРОД приведены в статье [Янковская, 2014a].

Что касается третьей компоненты – ИС ДИНАРЛОГ, то для ее создания целесообразно частично использовать математический аппарат, применяемый в ИИС ИМСЛОГ [Yankovskaya A.E., 2003]. В отличие от матричного способа представления данных и знаний в ИИС ИМСЛОГ, в ИС ДИНАРЛОГ предлагается использовать матричный способ представления знаний о вероятном нарушите-

ле – матрицу описаний Q в пространстве ХП и матрицу различий R , состоящую из 1-го столбца. Матрица Q заполняется только на основе знаний высококвалифицированных экспертов.

Оценивание вероятности совершения деструктивных действий в настоящий момент осуществляется на основе качественной шкалы ХП.

Оригинальные алгоритмы и реализующая их подсистема выявления различного рода закономерностей в знаниях, включая построение ОУББДТ, ОУСДТ и сигнальных признаков, приведены в [Yankovskaya et al., 2012], [Yankovskaya et al., 2013], [Янковская, 1998], [Янковская А.Е., 2010].

Эффективные алгоритмы построения множества решающих правил [Янковская, 2007], [Янковская, 2009а] и итогового решения на основе процедуры голосования, принятия решения на базе сочетания логико-комбинаторного (ЛК) и ЛК-вероятностного подходов к тестовому распознаванию образов, а также их программная реализация приведены в [Янковская А.Е., 2010].

Когнитивные средства: гистограмма специального типа, равно-сторонний треугольник, круговая диаграмма специального типа с отрезком и прямоугольником приведены в статье [Yankovskaya А.Е., 2003] и более поздних публикациях, перечисление которых невозможно в связи с ограничением рамок доклада.

Таким образом, основанием для отнесения обследуемого субъекта к вероятному нарушителю ИБ в ИС ДИВНАР является итоговое диагностическое решение, полученное с применением когнитивных средств и сформированное в виде заключения, содержащего информацию: о психическом состоянии обследуемого; его моральных, нравственных, профессиональных качествах; о способах и методах нанесения вреда. Заметим, что если в формировании заключения принимает участие эксперт в области ИБ, то им может быть сформирован и набор рекомендаций на основе результатов принятия решения для организации с целью предотвращения прецедента в отношении безопасности информации и ресурсов ИС в организации.

Заключение

Предложено использовать матричную модель представления данных и знаний для создания ИС ДИВНАР. Проведена структуризация знаний по проблеме поиска нарушителей ИБ с применением

ИС ПВН ИБ. Предложено использовать для конструирования ИС ПВН ИБ ИИС ИМСЛОГ. Использование знаний о нарушителях ИБ позволит спроектировать и внедрить СЗИ, отражающую возможные деструктивные воздействия на информацию, а также сформировать рекомендации по созданию или модернизации СЗИ.

В дальнейшем предполагается кроме качественных шкал формировать и использовать количественные шкалы для оценки вероятности совершения деструктивных воздействий нарушителем ИБ.

СПИСОК ЛИТЕРАТУРЫ

[ISO/IEC 15408-1:2009] ISO/IEC 15408-1:2009 Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model.

[ISO/IEC 17799:2000] ISO/IEC 17799:2000 Information technology – Code of practice for information security management (IDT). – http://www.klubok.net/gost/pdf/GOST_17799-2005.pdf.

[Yankovskaya A.E., 2003] Yankovskaya A.E., Gedike A.I., Ametov R.V., Bleikher A.M. IMSLOG-2002 Software Tool for Supporting Information Technologies of Test Pattern Recognition// Pattern Recognition and Image Analysis. 2003. Vol. 13, No. 2.

[Yankovskaya et al., 2012] Yankovskaya A.E., Kitler S.V. Parallel Algorithm for Constructing k-Valued Fault-Tolerant Diagnostic Tests in Intelligent Systems // Pattern Recognition and Image Analysis. 2012. Vol. 22, No 3.

[Yankovskaya et al., 2013] Yankovskaya A.E., Ametov R.V. Construction of fault-tolerance signal features subsets // 11th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-11-2013). Samara, September 23–28, 2013. Conference Proc. (Vol. I-II), Vol. I, Samara: IPSI RAS, 2013.

[Базовая модель угроз, 2008] «Базовая модель угроз безопасности персональных данных обрабатываемых в информационных системах персональных данных» утверждена ФСТЭК от 15.02.2008 года.

[Журавлёв и др., 1990] Журавлев Ю.И., Гуревич И.Б. Распознавание образов и анализ изображений // Искусственный интеллект в 3-х кн. Кн 2. Модели и методы: Справочник / Под ред. Д.А.Поспелова. М.: Радио и связь. 1990.

[Миронова, 2011a] Миронова В.Г., Шелупанов А.А. Анализ этапов предпроектного обследования информационной системы персональных данных. // Вестник СибГАУ им. М.Ф. Решетнева. 2011. №2(35).

[Миронова, 2011b] Миронова В.Г., Шелупанов А.А., Югов Т.Н. Реа-

лизация модели Take-Grant как представление систем разграничения прав доступа в помещениях // Доклады ТУСУР. 2011. №2(24). Ч. 3.

[Миронова, 2012] Миронова В.Г., Шелупанов А.А. Модель нарушителя безопасности конфиденциальной информации // Информатика и системы управления. 2012. № 1 (31).

[Янковская, 1998] Янковская А.Е. Построение k-значных диагностических тестов в интеллектуальной системе с матричным представлением знаний // Сб. науч. труд. VI Нац. конф. по ИИ с международным участием. Т. I, Пущино. 1998.

[Янковская А.Е., 2007] Янковская А.Е. Критерии оптимизации выбора безызбыточных диагностических тестов для принятия решений в интеллектуальных диагностических системах // Математические методы распознавания образов: 13-я Всеросс. конф. г. Зеленогорск.: Сб. докладов. 2007.

[Янковская, 2009а] Янковская А.Е. Принятие решений, устойчивых к ошибкам измерения значений признаков в интеллектуальных системах// Искусственный интеллект. Интеллектуальные системы // Материалы X Международной научно-технической конференции. Таганрог. 2009.

[Янковская А.Е., 2010] Янковская А.Е. Анализ данных и знаний на основе конвергенции нескольких наук и научных направлений // Интеллектуализация обработки информации: 8-я Международная конф. Респ. Кипр, г. Пафос, 17–24 октября 2010г.: Сб. докл. – М.: МАКС Пресс, 2010. (ISBN 978-5-317-03409-2).

[Янковская, 2011а] Янковская А.Е. Логические тесты и средства когнитивной графики // Издательский Дом: LAP LAMBERT Academic Publishing. 2011. № 8/9.

[Янковская, 2011б] Янковская А.Е., Китлер С.В. Гибридная интеллектуальная система диагностики и коррекции организационного стресса, основанная на сочетании матричного и критериального подходов // Интегрированные модели и мягкие вычисления в искусственном интеллекте // Сб. научн. тр. Межд. научно-техн. конф. г. Коломна. 2011.

[Янковская, 2014а] Янковская А.Е., Китлер С.В. Нечеткая интеллектуальная система экспресс-диагностики и профилактики депрессии: развитие и апробация // Нечеткие системы и мягкие вычисления // Материалы 6-й Всероссийской научно-практической конференции. Санкт-Петербург. 2014.

[Янковская, 2014б] Янковская А.Е., Шелупанов А.А., Миронова В.Г. Гибридная интеллектуальная система экспресс-диагностики, предназначенная для выявления вероятных нарушителей информационной безопасности// Нечеткие системы и мягкие вычисления // Материалы 6-й Всероссийской научно-практической конференции. Санкт-Петербург. 2014.

УДК 681.5+004.8

**ПРОЦЕДУРА ГОЛОСОВАНИЯ В ОДНОРОДНЫХ
КОЛЛЕКТИВАХ РОБОТОВ¹**

В.Э. Карпов (*vkarpov@hse.ru*)
МИЭМ НИУ ВШЭ, Москва

В работе предложены решения задач определения лидера и распределения ролей в однородной группе роботов. Показано, что, используя исключительно локальное взаимодействие, возможен переход от роя к коллективу роботов с иерархической организацией. В основе процедуры выбора лидера лежит алгоритм локального переголосования, а распределение ролей может быть осуществлено волновым методом.

Введение

Активные исследования в области создания систем взаимодействующих роботов ведутся уже почти четверть века. Такие направления, как коллективная, роевая, стайная и пр. робототехника, заняли прочные позиции в современной робототехнике и теории многоагентных систем. Однако до сих пор подавляющее число исследований в этой области остается на теоретическом, модельном уровне. Это хорошо видно по многочисленным обзорам, например, таким, как [Zhiguo et al., 2012] и [Yogeswaran & Ponnambalam, 2010]. Среди множества задач роевой робототехники, до сих пор остающимися недостаточно полно раскрытыми, выделим две – за-

¹ Работа выполнена при поддержке РФФИ (проект № 14-01-00817).

задачу определения лидера в группе роботов, а также задачу распределения ролей среди членов этой группы.

Лидерство. Одной из принципиальных особенностей роевой робототехники является локальный характер взаимодействия роботов друг с другом, а также роботов со средой [Zhiguo et al., 2012]. Такое взаимодействие называется неявной коммуникацией (implicit communication) [Yogeswaran & Ponnambalam, 2010]. Речь идет о том, что каждый робот группы непосредственно взаимодействует лишь со своими соседями, находящимися в некоторой ограниченной зоне видимости. Отсюда обычно следует, что в такой системе роботы самостоятельно принимают решения о дальнейших действиях, опираясь на некоторые простые правила локального взаимодействия.

Вместе с тем, однако, подавляющее число примеров решения задач в области роевой робототехники касается согласованного движения роя (очевидная, наглядная и достаточно простая задача, которая в [Zhiguo et al., 2012] выделена как метод «Следование за лидером» – «Leader-Follower method»). При этом считается, что в группе имеется априори заданный лидер, который и задает это движение. Правила локального взаимодействия могут быть самыми разными. От сугубо формальных [Павловский и др., 2010] до весьма экзотических. Например, в [Dewi et al., 2012] правила стайного движения основаны на модели «пружин и амортизаторов», определяющей притяжение и отталкивание особей к лидеру. В [Карпов, 2012], напротив, описан сугубо технический прием, позволяющий определить лидера для решения задачи согласованного движения. Иной подход к определению лидера предложен в [Даринцев, 2006], где описывается динамическое выделение агентов-координаторов (лидеров), основанное на «геометрических» построениях – лидерами групп становятся роботы, ближе всех расположенные к месту проведения неких технологических действий.

В любом случае мы имеем дело либо с априори заданным лидером, либо с неким сугубо техническим приемом, определяющим лидера в условиях некоторой конкретной задачи. Нас же интересует проблема выявления лидера в более общем случае, с точки зрения механизмов локального информационного обмена.

Распределение задач. Для решения задачи согласованного движения достаточно наличия лидера. Однако для более сложных

задач, решаемых группой роботов, требуется дифференциация их функций и, вообще говоря, распределение задач между роботами. Пожалуй, это одно из наиболее проблемных мест роевой робототехники. В уже упомянутых обзорах распределение задач в коллективе роботов рассматривается лишь декларативно. В лучшем случае упоминаются некоторые физические модели, методы распределенного планирования, оптимизации и пр. общие механизмы [Каляев и др., 2009]. Причиной этому является то, что дифференциация функций и распределение задач не рассматриваются роевой робототехникой как актуальная проблема, полагая, что рой должен решать лишь простые, массовые задачи типа согласованного движения. Это, безусловно, снижает значимость роевого подхода и, вообще говоря, идет вразрез с основным декларируемым тезисом роевой робототехники (роевая робототехника как способ решения сложных задач совокупностью простых технических устройств – роботами).

Мы же будем полагать, что и задача формирования лидера, и задача распределения ролей крайне важны для развития роевой робототехники. Рассмотрим варианты решения этих задач на примере такой модели организации группы роботов, как статический рой.

1. Статический рой

Статический рой характеризуется отсутствием заданного управляющего центра и представляет собой некую фиксированную в данный момент времени сеть – совокупность агентов [Карпов, 2013]. Основные свойства статического роя – это активность, локальность взаимодействия и функциональная неоднородность.

Важным вопросом представляется организация механизма такой функциональной неоднородности. Рассмотрим следующую задачу. Пусть имеется множество агентов (роботов), способных к локальному информационному обмену между ближайшими соседями. Далее, в некоторый момент времени статический рой должен реализовать некую процедуру распределения ролей: кто-то должен стать управляющим центром, кто-то – выполнять функции обработки информации, кто-то – сбора информации из внешней среды и т.д.

Общие соображения относительно принципа распределения ролей могут базироваться на следующих очевидных рассуждениях:

узел сети (агент), имеющий максимальное количество связей, становится претендентом на роль управляющего центра. Его ближайшее окружение – анализаторы информации, подготавливающие ее для принятия решения. Узлы, расположенные на периферии сети, отвечают за сбор информации. На рис. 1 изображен пример сети.

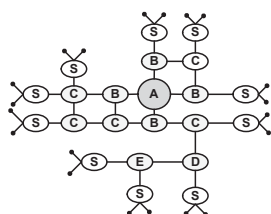


Рис. 1. Пример организации сети

Здесь узел *A* становится управляющим центром, его ближайшие соседи (*B*) – анализаторами, а периферийные узлы (*S*) будут отвечать за внешнюю сенсорику. При этом центральным вопросом является то, каким образом узлы-агенты выберут центральный, главный узел. Итак, рассмотрим далее возможные способы организации такого голосования.

2. Задача голосования

Сформулируем задачу следующим образом. Пусть имеется множество агентов с ограниченными коммуникационными возможностями. Это означает, что агенты способны лишь к непосредственному локальному взаимодействию между соседями. Для этого можно предположить, что агенты имеют фиксированное число коммуникационных портов – точек контакта, позволяющим им устанавливать каналы обмена информацией. Задача состоит в том, чтобы агенты выбрали единственного лидера путем голосования.

Пусть в некоторый момент времени агенты получают глобальный сигнал о начале голосования. В этот момент времени каждый агент устанавливает каналы связи со своими соседями. Таким образом, образуется некий в общем случае направленный граф. Вершинами его являются агенты, а входящие дуги интерпретируются как возможность получения информации от узла-источника – образуется канал связи. Зафиксируем статический рой, т.е. будем полагать, что далее его топология меняться не будет. Каждый агент описывается четверкой

$$A = (\alpha, L, C, W),$$

где α – идентификатор или имя агента, L – список агентов-соседей, от которых агент может получать информацию (входящие дуги), C – идентификатор «кандидата», за которого голосует агент α , W – вес кандидата C , т.е. число голосов, которое, по мнению агента, следует отдать за кандидата.

Суть процедуры голосования заключается в том, что каждый агент определяет, за кого голосуют его соседи. При этом в зависимости от веса кандидата, за которого голосует сосед, агент может поменять свое мнение и проголосовать за того же кандидата, что и его сосед.

На рис. 2 представлен один такт такой схемы голосования. Пометки вершин означают следующее: в «числителе» указывается идентификатор агента α , а величины C_α и W_α означают идентификатор кандидата и его вес соответственно.

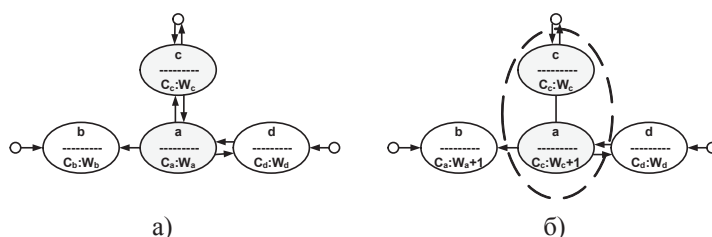


Рис. 2. Один такт голосования: а – начальное, б – конечное распределение голосов

Предположим, что агент a голосует за кандидата C_a , а агент c – за кандидата C_c . Если вес W_a окажется меньше веса W_c , то агент a может поменять свое мнение и переголосовать. При этом к весу нового кандидата будет добавлен еще один голос. Вероятность того, что агент i изменит свое мнение под влиянием мнения агента j (оппонента), может быть определена так:

$$p_{ij} = \frac{W_i}{W_i + W_j}.$$

Т.е. склонность к перемене своего мнения естественным образом зависит от степени убежденности (или веса) кандидата. Распределение голосов кандидатов и их весов в начальный момент времени осуществляется также вполне естественно: каждый агент голосует за себя (объявляет себя кандидатом), а вес этого решения

равен количеству соседей этого агента. Ниже приведен алгоритм поведения агента при голосовании.

Алгоритм G1(α). Принятие решения агентом

α – агент
 C_α – «кандидат», за которого голосует агент α
 W_α – вес кандидата
 L_α – список агентов-соседей
procedure G1 (α)
 1. Выбрать среди своих соседей оппонента A_{op} :
 $A_{op} \in L_\alpha, C_{op} \neq C_\alpha$
 $W_{op} = \max_{i \in L_\alpha} W_i$
 2. Вычислить значение вероятности изменения
 мнения $p_\alpha = \frac{W_{op}}{W_\alpha + W_{op}}$
 3. С вероятностью p_α изменить мнение: $C_\alpha \leftarrow C_{op}$
 $W_\alpha \leftarrow W_{op} + 1$
end procedure

Можно «загрубить» алгоритм принятия решения, заставляя агента безусловно менять свое мнение о кандидате, если в его окружении имеется более сильный «оппонент». Если же наблюдается паритет между весами мнений агента и его самого сильного оппонента, то здесь выбор решения может быть осуществлен уже вероятностным образом. Общая схема голосования может выглядеть так:

Алгоритм V(A). Голосование

A – множество агентов
 α – агент
 C_α – «кандидат», за которого голосует агент α
 W_α – вес кандидата
 L_α – список агентов-соседей
 eoj – флаг завершения процедуры голосования
procedure V (A)
 1. $eoj \leftarrow false$
 2. **for all** $\alpha \in A$ **do** -- Инициализация агентов
 3. $C_\alpha \leftarrow \alpha$
 $W_\alpha \leftarrow \dim(L_\alpha)$
 4. **end for**

```

5. while not eoj do    -- Основной цикл голосо-
вания
6. for all  $\alpha \in A$  do  -- Цикл по всем агентам
7. G1( $\alpha$ )
8. end for
9. «Определение условия завершения процедуры го-
лосования eoj»
10. end while
end procedure

```

В этом алгоритме наиболее проблемным является п.9: «Определение условия завершения процедуры голосования *eoj*». В отсутствие глобальной информации о состоянии сети агент должен сам принимать решение о том, что голосование закончено. Получаемая от ближайшего окружения информация является явно недостаточной для этого, поэтому возможны два варианта поведения агента:

1. Считать, что голосование должно быть завершено максимум через некоторое определенное количество тактов. Для этого необходима верхняя оценка количества шагов алгоритма голосования.

2. Реализовать некоторую процедуру обмена сообщениями, которая могла бы определить, что голосование завершено, и никто из агентов уже не меняет свое решение.

Ниже приведены примеры работы процедуры голосования. На рис. 3а изображены 3 такта голосования для плотной группы роботов.

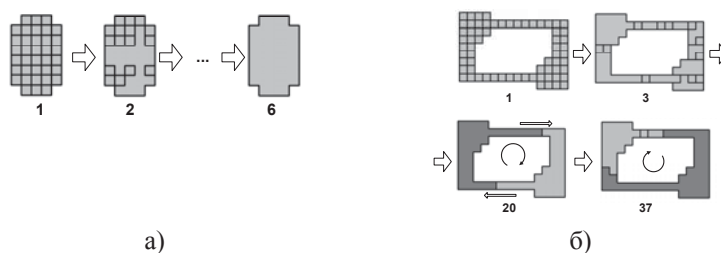


Рис. 3. а – процедура голосования в плотной группе (такты 1, 2 и 6),
б – Циклическая процедура голосования (такты 1, 3, 20 и 37)

На первом такте каждый агент голосует за себя, поэтому количество клеток, обозначающих «границы» распределения голосов

за соответствующих кандидатов, равно числу роботов. На втором такте в результате переголосования происходит укрупнение областей, голосующих за выбранных кандидатов. Наконец, на 6-м такте все голоса отданы за одного кандидата и процедура голосования завершается.

На рис. 3б приведен пример процесса голосования в крайне неблагоприятных условиях. Здесь наблюдаются две явно выраженные зоны, соединенные двумя перешейками. В такой ситуации наблюдается циклический процесс распространения голосов. К 20-му такту голосования образованы две устойчивые области, каждая из которых голосует за своего кандидата. И далее начинается процесс циклического переголосования. Это хорошо видно на рисунке, соответствующему 37 такту, когда предпочтения областей, фактически, поменялись местами. Процесс устанавливается лишь к такту 51, когда в системе прекращаются колебания и остается единый кандидат в лидеры.

Судя по результатам имитационного моделирования, процесс голосования сходится. При этом количество тактов голосования не превышало числа роботов в группе.

Распределение задач. При отсутствии морфологических различий между агентами, распределение ролей в статическом рое определяется исключительно текущей топологией системы. Сам процесс распределения представляет собой известную процедуру распространения волны управления. Инициатором распространения является лидер R_0 . Непосредственные соседи лидера («ближний круг») получают от него иницирующий пакет, согласно которому им назначается роль R_1 , и т.д. Таким образом, роль i -го робота определяется ролями его окружения:

$$R_i = \max_{k \in Z} R_k + 1.$$

Волновое распределение ролей реализуется исключительно локальным взаимодействием.

Заключение

Итак, были предложены простые и эффективные механизмы решения таких важных задач роевой робототехники, как определение лидера и распределение ролей между членами группы. Под эффективностью понимается их приемлемость для роботов с ограничен-

ными когнитивными возможностями (ограниченность сенсорики, вычислительных мощностей, каналов связи и т.д., т.е. всего того, что характерно для роевой робототехники). Несмотря на свою простоту, реализация этих механизмов позволяет говорить о наличии принципиальной возможности образования действительно сложных по своей организации структур в однородных коллективах.

СПИСОК ЛИТЕРАТУРЫ

[Даринцев, 2006] Даринцев О.В. Система управления коллективом микророботов // «Штучный интеллект» 4, 2006, №4, с. 391–399.

[Иванов, 2011] Иванов Д.Я. Использование принципов роевого интеллекта для управления целенаправленным поведением массово применяемых микророботов в экстремальных условиях // Изв. высших учебных заведений. М.: Машиностроение. 2011, №9, с. 70–78.

[Каляев и др., 2009] Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов. М.: Физматлит, 2009. 280 с.

[Карпов, 2012] Карпов В.Э. Частные механизмы лидерства и самосознания в групповой робототехнике // Тринадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2012. Т. 3. Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, 2012, с. 275–283.

[Карпов, 2013] Карпов В.Э. Управление в статических роях. Постановка задачи // Тр. VII Международной научно-практической конференции Интегрированные модели и мягкие вычисления в искусственном интеллекте, Коломна. Т.2. М.: Физматлит, 2013, с. 730–739.

[Павловский и др., 2010] Павловский В.Е., Кирикова Е.П., Павловский В.В. Моделирование поведения больших групп роботов в среде с препятствиями // Тр. научно-технического семинара «Управление в распределенных сетевых и мультиагентных системах». СПб.: ОАО «Концерн ЦНИИ «Электроприбор», 2010, с. 10–13.

[Dewi et al., 2012] Dewi T., Risma P., Oktarina Y. Wedge Formation Control of Swarm Robots. // The 14th Industrial Electronics Seminar, Electronic Engineering Polytechnic Institute of Surabaya (EEPIS), Indonesia, 2012. P. 294–298.

[Yogeswaran & Ponnambalam, 2010] Yogeswaran M. and Ponnambalam S. G. (2010). Swarm Robotics: An Extensive Research Review, Advanced Knowledge Application in Practice, InTech. P. 259–278.

[Zhiguo et al., 2012] Zhiguo Shi, Jun Tu, Qiao Zhang, Lei Liu, Junming Wei, A Survey of Swarm Robotics System // Proc. of the Third Intern. Conf. on Advances in Swarm Intelligence, Shenzhen, China, 2012. V.1, P. 564–572.

МОДЕЛЬ КООПЕРАЦИИ АГЕНТОВ (РОБОТОВ)¹

А.А. Кулинич (*kulinich@ipu.ru*)
Институт проблем управления РАН, Москва

Исследована модель кооперации агентов (роботов) в динамических ситуациях, основанная на критериях полезности агента и когнитивного диссонанса их отношений.

Введение

Исследование вопросов кооперации агентов (роботов) возможно с использованием абстрактных математических моделей абстрактных многоагентных систем. Многоагентная система состоит из множества интеллектуальных, общающихся между собой агентов (роботов), которые могут объединяться в коалицию для решения сложных задач, решение которых невозможно отдельными агентами.

Интересны две группы методов образования коалиций (коалиционных алгоритмов). Первая группа – это алгоритмы образования коалиций, основанные на методах решения кооперативных игр. Задача решается при условии полной информированности всех агентов о целях, ресурсах и стратегиях других агентов, а также при условии возможности дележа выигрыша, полученного коалицией агентов. Методы решения кооперативных игр (С-ядро, МН-решения и др.) [Оуэн, 1971] имеют экспоненциальную сложность вычислений и коммуникаций между агентами относительно числа агентов, что ограничивает возможность их практического применения [Смирнов и др., 2011].

Вторая группа методов – это методы, основанные на моделировании поведения агентов при образовании коалиций на основе теорий социальной зависимости субъектов [Conte, 2001]. Эти методы позволяют решать вопросы образования открытых динамических коалиций, состав которых может меняться в процессе решения за-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-01-00817).

дачи, а агенты могут свободно входить в состав коалиции или выходить из нее [Conte, 2001].

Далее рассматривается модель, основанная на моделях социального поведения, изложенных в работах социальных психологов Д. Хоманса [Хоманс, 1984] и Л. Фестингера [Фестингер, 1999]. В этих работах предложены: модель функционирования малых социальных групп (коалицию можно рассматривать как малую социальную группу) на основе обмена полезностями и концепция когнитивного диссонанса, позволяющая оценить субъективные ощущения агентов о справедливости в коалиции и моделировать ее устойчивость.

1. Базовая модель

Рассмотрим динамическую систему и множество агентов N , которые пытаются изменить состояние системы на выгодное для каждого из них состояние. В качестве математической системы выбрана модель представления экспертных знаний о процессах в динамической системе в виде когнитивной карты [Кулинич, 2012]. В когнитивной карте определено множество факторов ситуации $A = \{a_j\}$, упорядоченное множество лингвистических значений факторов $L_i = \{l_{ij}\}$, $l_{ij} < l_{ij+1}$, $i = 1 \dots n$, $j = 1 \dots m$, причинно-следственные отношения между значениями факторов ситуации $W \subseteq \times L_i$. Динамика изменения факторов ситуации в когнитивной карте задается системой конечно-разностных уравнений:

$$Y(t+1) = W \circ Y(t), \quad (1)$$

где W – система правил «Если, То», заданная на множестве всех возможных значений факторов ситуации $W: \times L_i \rightarrow \times L_i$, $Y(t)$ – состояние ситуации – это вектор лингвистических значений (l_{1j}, \dots, l_{nb}) , т.е. $Y(t) = (l_{1j}, \dots, l_{nb})$, $l_{ij} \in L_i \forall i$.

Каждый агент характеризуется кортежем $\langle g_q, r_q, e_q, \mu_q(Y_q, g_q) \rangle$, где:

1. $g_q = (l_{1j}, \dots, l_{nb})$ – вектор целевых факторов и их лингвистических значений для агента q ;

2. $r_q = (l_{1j}, \dots, l_{nb})$ – стратегия управления агента q , где $r_q \in U_q$, $U_q = \times L_{qi}^r$, $L_{qi}^r \subseteq L_i$ – ресурсы агента q . Считается, что если агент q применяет стратегию r_q для достижения своей цели g_q , то прогноз

развития ситуации на n шагах моделирования определится из соотношения (1) при условии, что $Y_q(0) = r_q$, т.е.

$$Y_q(1) = W^{\circ} r_q, Y_q(2) = W^{\circ} Y_q(1), \dots, Y_q(n) = W^{\circ} Y_q(n-1).$$

3. $e_q: (u(g_q), (v(r_q))) \rightarrow R^+$ – эффективность достижения агентом целевой ситуации, где $u(g_q)$ – полезность целевой ситуации, $u: g_q \rightarrow R^+$; $v(r_q)$ – затраты агента для достижения целевой ситуации, $v: r_q \rightarrow R^+$;

4. $\mu_q(Y_q(n), g_q)$ – возможность достижения агентом q целевой ситуации за счет собственных ресурсов в условиях противодействия агентов противников.

При определении возможности достижения целевой ситуации считается, что в пространстве состояний $(\times L_i)$ определена метрика $\rho(a, b)$, $a, b \in \times L_i$. Тогда возможность достижения цели определяется как близость прогнозной ситуации $Y_q(n)$ и его целевой ситуацией g_q :

$$\mu_q(Y_q(n), g_q) = \rho(Y_q(n), g_q).$$

По сути, этот показатель определяет «силу» каждого агента коалиции без поддержки сторонников, при условии противодействия противников.

1.1. Необходимые и достаточные условия образования коалиции

Сторонниками по коалиции считаются агенты с близкими целями, т.е. $K \subseteq N$, $\rho(g_i, g_j) \leq \varepsilon$, $\forall i, j \in K$, а агенты $N \setminus K$ – это противники. Для оценки устойчивости коалиции в этой работе на основе теории социального поведения субъектов [Хоманс, 1999] анализируется взаимная полезность агентов при объединении их ресурсов для достижения общей цели. Считается, что дисбаланс во взаимной полезности агентов в коалиции создает у них ощущения несправедливости, которые увеличивают их мотивации выхода из коалиции, угрожающие ее устойчивости. Для оценки ощущений агентов о несправедливости в коалиции используется теория когнитивного диссонанса [Фестингер, 1999].

1.2. Критерий взаимной полезности агентов в коалиции

Агенты q и i называются взаимно полезными, если объединение их стратегий $r_i \oplus r_q$ увеличивает возможность достижения собственных целей (g_i, g_q) агентами q и i . Т.е. если $\mu(Y_{i+q}, g_i) > \mu(Y_i, g_i)$,

$\mu(Y_{i+q}, g_q) > \mu(Y_i, g_i)$. Степень полезности $P(i, q)$ агента q для агента i определяется из соотношения:

$$P(i, q) = \frac{\mu(Y_{i+q}, g_i)}{\mu(Y_i, g_i)} - 1,$$

где $\mu(Y_{i+q}, g_i)$, $\mu(Y_i, g_i)$, $\mu(Y_i, g_i) \in [0, 1]$ – возможности достижения агентами q и i своих целей при объединении их стратегий и, соответственно, собственными силами в условиях противодействия противников.

Социальное поведение в социальной группе – это обмен ценностями между агентами (ресурсами, полезностями и т.д.). Целостность группы достигается при балансе «сил» между обменами [Хоманс, 1984]. Т.е. по критерию взаимной полезности коалиция K устойчива, если агенты коалиции одинаково полезны друг для друга $P(i, q) = P(q, i)$, $\forall i, q \in K$ и имеют равную эффективность участия в коалиции, т.е. $e_i = e_q$.

1.3. Критерий когнитивного диссонанса

Наличие дисбаланса взаимной полезности или эффективности агентов, включенных в коалицию, приводит к возникновению латентных конфликтов в коалиции, которые характеризуют ощущения агентов о несправедливости в коалиции, и определяются уровнем когнитивного диссонанса. Когнитивный диссонанс – это противоречие в системе знаний человека, порождающее у него неприятные переживания и побуждающее его к действиям, направленным на устранение этого противоречия [Фестингер, 1999]. Когнитивный диссонанс вызывает у субъекта (агента) стремление уменьшить его или препятствовать его росту. Проявление этого стремления у агентов приводит к изменению его поведения, которое приводит к их выходу из коалиции, т.е. к неустойчивости коалиции.

Степень когнитивного диссонанса по взаимной полезности между агентами i, q определяется из соотношения:

$$D^W(q, i) = \frac{P(i, q) - P(q, i)}{P(i, q) + P(q, i)}, D^W(q, i) = -D^W(i, q), \forall i, q, i \neq q. \quad (2)$$

Степень когнитивного диссонанса по эффективности между агентами i, q определится из соотношения:

$$D^E(i, q) = \frac{e_i - e_q}{e_i + e_q}, D^E(i, q) = -D^E(q, i), \forall i, q, i \neq q. \quad (3)$$

По критерию когнитивного диссонанса устойчивая коалиция определится следующим образом: коалиция K устойчива, если для всех агентов $i, q \in K$ их когнитивные диссонансы по полезности и эффективности близки к нулю $D^W(i, q) \approx 0, D^E(i, q) \approx 0, \forall i, q$.

2. Пример

В качестве примера рассмотрим описание в терминах предложенной модели кооперативного взаимодействия агентов, играющих в футбол.

Игру в футбол будем считать динамической системой. Параметры этой системы суть положение игроков на футбольном поле двух противоборствующих команд $A = \{a_i\}$ и $B = \{b_i\}$. Состояние футбольной игры динамической системы будем характеризовать вектором положения всех игроков – агентов на плоскости в последовательные моменты времени: $(x_{a_i}(t); y_{a_i}(t)), \dots, (x_{b_i}(t); y_{b_i}(t)), i \in N$, где N – число агентов обеих команд, $t = 1, \dots, n$. Кроме этого, заданы координаты мяча $Ball = (x_m(t); y_m(t))$. Координаты ворот противников, заданные интервалами: ворота команды $A - G_A = (x_a; [y_{a1}, y_{a2}])$ и, соответственно, команды $B - G_B = (x_b; [y_{b1}, y_{b2}])$.

Цели игроков команды A – забить мяч в ворота команды B , т.е. $\forall a_i, g_{a_i} = (x_m(t); y_m(t)) | (x_m(t); y_m(t)) \in (x_b; [y_{b1}, y_{b2}])$, и соответственно, игроков команды B забить мяч в ворота команды A , т.е. $\forall b_i, g_{b_i} = (x_m(t); y_m(t)) | (x_m(t); y_m(t)) \in (x_a; [y_{a1}, y_{a2}])$.

Стратегии игроков (агентов) – это их следующие действия на каждом шаге дискретного времени: $r1(a_i(b_j), t)$ – движение к мячу; $r2(a_i(b_j), t)$ – движение с мячом к воротам; $r3(a_i(b_j), t)$ – удар по воротам; $r4(a_i(b_j), t)$ – перемещение в позицию паса; $r5(a_p, a_j(b_p, b_j), t)$ – пас партнеру по команде.

Эффективность игрока – это соотношение его ожиданий вознаграждения за игру $u(a_i(b_j))$ к результативности его игры, которая выражается интегральной оценкой его результативных действий: удар по воротам и пас партнеру по команде, т.е. $v(r3(a_i(b_j)), r5(a_p, a_j(b_p, b_j), t))$.

Возможность достижения цели игроками команд $A(B)$ определяется его ресурсами: близостью агента к воротам противника, т.е. $d(a(b)_p, G_{B(A)}, t) = \rho((x_{a(b)_i}(t); y_{a(b)_i}(t)), (x_{a(b)}; [y_{a(b)1}, y_{a(b)2}]))$; близостью противников $d(a_p, b_j, t) = \rho((x_{a_i}(t); y_{a_i}(t)), (x_{a_i}(t); y_{a_i}(t)))$; индивиду-

альной скоростью агента – $V(a(b)_i)$ и его действиями $r1(a_i(b_j), t), \dots, r5(a_i, a_j(b_j), t)$.

2.1. Кооперация агентов на основе критерия полезности

В работе [Кулинич, 2012] было показано, что полезными агентами для кооперации являются агенты, имеющие большую возможность достижения цели. Определим правило вычисления полезности агента, выразив ее через составляющие их возможности достижения цели, и определим его действия.

Агент a_i полезен своим партнерам по критерию близости к воротам противника (K_G^B), если $d(a_i, G_B) < d(a_q, G_B)$, $i \neq q$, $q=1, \dots, N/2$. Т.е. агент полезен, если он находится ближе всех своих партнеров к воротам противника, ($K_G^B=1$).

Агент a_i полезен своим партнерам по критерию близости противников (K^B), если $d(a_i, b_j) > d(a_q, b_j)$, $i \neq q$, $q=1, \dots, N/2$, $j=N/2+1, \dots, N$. Т.е. агент полезен, если он находится дальше всех своих партнеров от игроков противника, ($K^B=1$).

Агент a_i полезен своим партнерам по критерию скорости (K^V), если $V(a_i) > V(a_q)$, $i \neq q$, $q=1, \dots, N/2$. Т.е. агент полезен, если его скорость больше, чем у всех партнеров, ($K^V=1$).

Интегральная характеристика полезности агента определяется значением функции: $P(a_i) = F(K_G^B, K^B, K^V)$.

В качестве одного из вариантов такой функции рассмотрим линейную свертку с коэффициентами предпочтений полезностей – α, β, γ , полученных с помощью процедуры парных сравнений. В этом случае

$$P(a_i) = \alpha K_G^B + \beta K^B + \gamma K^V, \quad \alpha + \beta + \gamma = 1.$$

Тогда, $P(a_i) = 1$, при $K_G^B = K^B = K^V = 1$, и $P(a_i) = 0$, при $K_G^B = K^B = K^V = 0$.

2.2. Выбор стратегии агента на основе анализа его полезности

Выбор действий агентов будем рассматривать для двух случаев: агент не владеет мячом; агент владеет мячом. Агент, не владеющий мячом, выбирает действия: $r1(a_i(b_j), t)$ – движение к мячу; $r4(a_i(b_j), t)$ – перемещение в позицию паса. Эти действия не кооперативные и их выбор не основан на анализе полезности агента. Для агента, владеющего мячом, выбор действия основан на анализе полезности партнеров по команде. При условии, что полезность

игрока, владеющего мячом, выше, чем полезность остальных партнеров, т.е. $P(a(b)_i) > P(a(b)_q)$, $i \neq q$, $q=1, \dots, N/2$, то агент выбирает действия: $r2(a_i(b)_i, t)$ – движение с мячом к воротам; $r3(a_i(b)_i, t)$ – удар по воротам.

Если полезность игрока, владеющего мячом, ниже, чем полезность одного из партнеров, то выбирается действие: $r5(a_i, a_j(b_i, b_j), t)$ – пас партнеру по команде $(a_j(b_j))$, имеющему большую полезность.

2.3. Устойчивость кооперации агентов

Для анализа устойчивости кооперации агентов необходимы анализ взаимной полезности агентов и анализ когнитивного диссонанса их эффективности на протяжении игры.

Взаимная полезность агентов в игре определяется суммарным числом пасов между парой агентов:

$$P(a_i, a_j) = \frac{\sum_{t=1, \dots, n} r5(a_i, a_j, t)}{\sum_{t=1, \dots, n} r5(a_i, a_j, t) + \sum_{t=1, \dots, n} r5(a_j, a_i, t)}, P(a_j, a_i) = 1 - P(a_i, a_j).$$

Здесь $\sum_{t=1, \dots, n} r5(a_i, a_j, t)$ и $\sum_{t=1, \dots, n} r5(a_j, a_i, t)$ число пасов агента a_i агенту a_j и, соответственно, a_j агенту a_i за промежуток времени игры $t=1, \dots, n$.

Агенты с неравной взаимной полезностью характеризуются когнитивным диссонансом, вычисляемым по формуле (2), и могут изменять свое поведение. Изменение поведения проявляется в изменении правил выбора агента на основе полезности агента партнера.

При анализе когнитивного диссонанса по эффективности агентов считается, что агенты получают вознаграждение пропорционально их результативности (забитые голы). Эффективность агента определится из соотношения:

$$E(a_i) = \frac{\sum_{t=1, \dots, n} r3(a_i, t)}{\sum_{\substack{t=1, \dots, n \\ i=1, \dots, N/2}} r3(a_i, t)}.$$

Агенты с неравной эффективностью характеризуются когнитивным диссонансом по эффективности, вычисляемой по формуле (3), и могут изменять свое поведение, изменения правила выбора агента на основе полезности агента партнера.

2.4. Эксперимент и обсуждение

Предложенная модель кооперации агентов была исследована экспериментально. Для этого в среде моделирования многоагентных систем VisualBots for Excel была разработана программа, моделирующая игру в футбол двух команд, состоящих из четырех игроков. Агенты в каждой команде незначительно отличались по скорости. Считается, что все агенты получают вознаграждение пропорционально количеству забитых голов, независимо от выигрыша или проигрыша команды.

На рисунках 1–4 показаны результаты экспериментов. Здесь по оси абсцисс показаны номера игроков команд, а по оси ординат – количество пасов партнерам (рис. 1 и рис. 3) и забитых голов (рис. 2 и рис. 4) в двух сериях игр: без учета и с учетом диссонанса игроков. Имитационное моделирование поведения агентов на основе анализа полезности агентов (без диссонансов) показало, что агенты с меньшими скоростями характеризуются меньшей результативностью по забитым голам, но не по количеству пасов партнерам (игрок №4 команды *A*, игрок №3 команды *B*). Это может быть интерпретировано как самоорганизация агентов в команде. Т.е. агенты с меньшей скоростью выполняют задачи защитников, а агенты с большей скоростью играют в нападении.

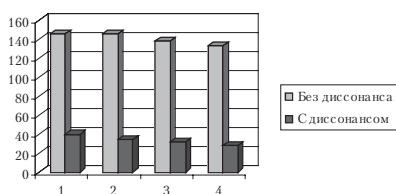


Рис. 1. Пасы игроков команды A

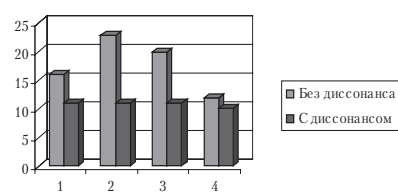


Рис. 2. Голы игроков команды A

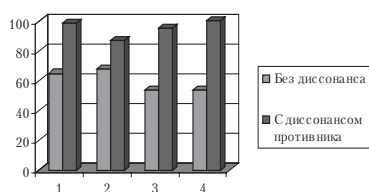


Рис. 3. Пасы игроков команды B

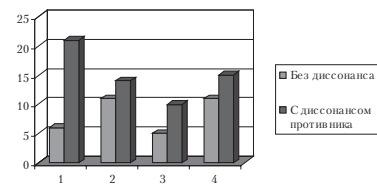


Рис. 4. Голы игроков команды B

При моделировании поведения агентов с учетом диссонансов диссонансы учитывались только для игроков команды *A*. Т.е. все игроки команды *A* при передаче паса партнеру учитывают не только его полезность, но и эффективность. Если эффективность партнера выше, чем у игрока, передающего пас, то пас не передается. Этот игрок, из-за зависти или эгоистических соображений, пытается самостоятельно забить гол. На рис.1 и рис. 2 показано, что результативность распасовки игроков команды *A*, при учете диссонансов агентов, резко снизилась. Снижился и диссонанс по эффективности (забитым голам) – все игроки забивают одинаковое количество голов и, следовательно, получают равное вознаграждение. Игроки же команды *B*, играющие с «эгоистичной» командой *A*, повысили свою результативность (рис. 3, рис. 4).

Заключение

Имитационное моделирование поведения агентов на основе предложенных математических моделей их кооперации на основе критериев полезности агентов и их когнитивных диссонансов показало правдоподобный результат. Такие модели могут служить алгоритмической основой для реализации поведения автономных роботов в задачах кооперации.

СПИСОК ЛИТЕРАТУРЫ

- [Оуэн, 1971] Оуэн Г. Теория игр. М.: Мир, 1971.
- [Смирнов, 2011] Смирнов А.В., Шереметов Л.Б. Модели формирования коалиций кооперативных агентов: состояние и перспективы исследований // Искусственный интеллект и принятие решений. № 1, 2011, с. 36–48.
- [Conte, 2001] Conte R., Edmonds B., Moss S., Sawyer K., *Sociology and Social Theory in Agent Based Social Simulation: A Symposium. Computational and Mathematical Organization Theory*, 2001, v.7, n.3, pp. 183–205.
- [Хоманс, 1984] Хоманс Дж. Социальное поведение как обмен // Современная зарубежная социальная психология. М., 1984. С. 83–91.
- [Фестингер, 1999] Фестингер Л. Теория когнитивного диссонанса. СПб.: Ювента, 1999.
- [Кулинич, 2012] Кулинич А.А. Модель поддержки принятия решений для образования коалиций в условиях неопределенности// Искусственный интеллект и принятие решений. № 2, 2012, с. 95–106.

УДК 004.896 + 007.52

СИТУАЦИОННОЕ УПРАВЛЕНИЕ КАК ОСНОВА АДАПТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ШАГАЮЩИХ РОБОТОВ

В.А. Луцкий (*Lu10001@yandex.ru*)

Московский институт электроники и математики НИУ ВШЭ,
Москва

В работе описывается возможность применения модели ситуационного управления как основы для создания адаптивной системы управления движением шагающего робота. Рассмотрен пример создания системы ситуационного управления для выбора наиболее подходящего в конкретной ситуации алгоритма передвижения шестиногого робота.

Введение

Шагающие роботы являются популярным объектом исследования в наши дни: университеты по всему миру занимаются исследованием шагающих механизмов, робототехнические фирмы выпускают многочисленные реализации шагающих роботов. Популярность шагающих роботов обусловлена во многом их хорошей проходимостью сложных форм рельефа, возможностью устойчивого подъема и спуска по склонам и лестницам [Knight, 2002]. Для наиболее устойчивого прохождения сложных форм рельефа шагающий робот должен быть адаптивным, т.е. обладать способностью адаптации к изменению формы рельефа поверхности.

В настоящее время существует несколько методов стабильного передвижения шагающих роботов (под стабильной походкой понимается походка, не приводящая к падениям и застреваниям, несмотря на все внешние обстоятельства [Bruijn, 2013]). Одним из наиболее известных является метод точки нулевого момента, основанный на расчете точки, где момент равнодействующей всех сил, действующих на робот в горизонтальном направлении, равен нулю. Данный метод применяется во многих человекоподобных двуногих роботах, например Honda ASIMO. Метод позволяет двуногому роботу достаточно стабильно передвигаться по довольно ровной поверхности, но для передвижения по неровным поверхностям он не-

достаточен. Необходимо использовать более сложные адаптивные алгоритмы, учитывающие положение робота в целом и каждой его конкретной конечности в отдельности в процессе передвижения.

Исследование адаптивных алгоритмов передвижения по неровным поверхностям и возможность их применения для разных видов шагающих роботов – основная цель проекта. На первом этапе было решено провести исследование алгоритмов передвижения шестиногого робота как статически устойчивой конструкции, где нет необходимости обеспечивать балансировку при движении и можно сосредоточиться непосредственно на передвижении по неровной поверхности. Система управления передвижением робота должна учитывать множество различных ситуаций, поэтому ей довольно затруднительно сопоставить математическое описание в терминах классической теории автоматического управления. Однако описание можно относительно несложно составить на естественном языке. Одновременно с этим создаваемая система управления уникальна и учитывает особенности конструкции конкретного робота. Учитывая все вышесказанное, было решено разрабатывать систему управления роботом на основании ситуационного управления – подхода к построению подобных систем управления, предложенного Д.А. Поспеловым [Поспелов, 1986].

1. Робот Hexapod MP4

1.1. Конструкция робота

Поскольку создаваемая система управления учитывает особенности конструкции робота, перед описанием непосредственно самой системы приведем кратко описание конструкции робота.

Для проведения исследований шагающих механизмов в 2013-2014 годах был создан шестиногий шагающий робот Hexapod.

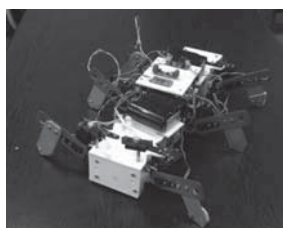


Рис. 1а. Общий вид робота

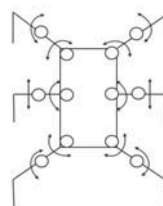


Рис. 1б. Кинематическая схема

Корпус робота собран из трех соединенных модулей, образующих в сборе шестиногую конструкцию с возможностью переделки в четырехногий вариант для дальнейших исследований. Каждый модуль состоит из пластикового корпуса и модуля конечности. Модуль конечности включает в себя 2 сервопривода, обеспечивающих перемещение конечности по двум степеням свободы: горизонтальной и вертикальной и собственно конечности.

Управление роботом осуществляется сервоконтроллером Pololu, расположенном на среднем модуле. Контроллер реализует элементарные команды каждого движения (такие как переместить такой-то сервопривод в такое положение или считать датчики), основная управляющая система робота находится на ПК, связь с которым происходит по специальному протоколу через Bluetooth. Для ориентации в пространстве робот использует датчики: 2 датчика-дальномера для ориентации в пространстве на переднем модуле робота, акселерометр на среднем модуле и датчики касания на каждой ноге робота.

1.2. Режимы передвижения робота

Первоначально робот проектировался для исследований походки шестиногого робота в целом, поэтому в качестве базового алгоритма походки шестиногого робота выбран достаточно простой алгоритм перемещения «3+3 насекомоподобный». Данный алгоритм широко распространен как в природе [Figliolini, 2007], так и механических шагающих устройствах, однако он подходит по большей части для относительно ровных поверхностей и в дальнейшем в качестве базового будет выбран другой алгоритм.

В основу адаптивного передвижения робота положено 3 принципа адаптивности:

- принцип стабилизации (выравнивания положения по показаниям акселерометра);
- принцип останова (нога опускается до тех пор, пока не достигнута поверхность);
- принцип нащупывания поверхности (если после первого опускания ноги поверхность не достигнута, происходит поиск поверхности спереди и сзади относительно изначального положения ноги).

В программе управления передвижением шагающего робота предусмотрено несколько различных способов передвижения, ос-

нованных на базовом алгоритме передвижения и принципах адаптивности [Луцкий, 2013]. Реализовано как полностью неадаптивное передвижение, так и варианты с различными комбинациями методов адаптивности.

Проведенные эксперименты подтвердили, что стабильнее робот проходит неровные участки пути при адаптивном передвижении со стабилизацией, тем не менее такое передвижение получается медленным и использование его на гладкой поверхности будет излишним. Таким образом, необходимо создать систему, которая анализирует положение робота в текущий момент времени и выбирает оптимальный алгоритм для следующего движения. Для реализации этой задачи было решено обратиться к ситуационному управлению.

2. Модель ситуационного управления шестиногого робота

Общая схема ситуационного управления представлена на рисунке 2:



Рис. 2. Схема ситуационного управления

В системе управления шестиногого робота на данном этапе реализована довольно упрощенная модель, состоящая из анализатора, классификатора и блока выбора решения.

2.1. Описание ситуаций

Рассмотрим описание ситуаций, возникающих при передвижении шагающего робота. На самом нижнем уровне описания передвижения шагающего робота нас будет интересовать положение

ног робота в определенный момент времени после совершенного движения. По сути именно положение ног определяет тип передвижения. Пусть n_i – i -ая нога робота. Введем лингвистическую переменную h – «поднятие ноги», имеющую значения «полностью опущена», «сильно опущена», «слабо опущена» и «поднята», также введем отношение r_h – «иметь значение положения ноги». Помимо значения высоты можно определить факт касания поверхности по показаниям датчиков-концевиков. Пусть переменная t – «касание», имеет два значения: «есть касание» и «нет касания», r_t – отношение «наличие касания поверхности».

В этом случае положение для конкретной ноги робота в определенный момент времени принимает вид:

$$s_{ni} = (n_i r_h h) \& (n_i r_t t).$$

Положение робота определяется положением всех шести ног в определенный момент времени:

$$s_n = s_{n1} \& s_{n2} \& s_{n3} \& s_{n4} \& s_{n5} \& s_{n6}.$$

На более высоком уровне описания передвижения робота, рассматривая не отдельные конечности, а робот в целом, можно определить наклон робота на поверхности по показаниям акселерометра. Для описания наклона введем переменную in_x – «наклон вбок», которая имеет значения: «сильно наклонен вправо», «слабо наклонен вправо», «нет наклона», «слабо наклонен влево» и «сильно наклонен вправо». Аналогично можно определить наклон в другой плоскости: переменная t_y – «наклон вперед» имеет значения: «сильно наклонен вперед», «слабо наклонен вперед», «нет наклона», «слабо наклонен назад» и «сильно наклонен назад». Переменной q обозначим робота, r_t – отношение «иметь наклон»:

$$s_t = (q r_t t_x) \& (q r_t t_y).$$

Любой робот, передвигающийся автономно, должен определять препятствия перед собой и выбирать направление передвижения, чтобы их обойти. Очень подробно ориентация робота в пространстве согласно принципам ситуационного управления описана в статье А.С. Ющенко [Ющенко, 2007]. В нашем случае для определения препятствия пока используется довольно простая система, работающая по показаниям двух датчиков-дальномеров. Перемен-

ная d – «расстояние до препятствия» имеет всего 2 значения: «препятствие далеко» и «препятствие близко».

Таким образом, препятствия на пути робота описываются формулой:

$$s_d = (q_r d_r) \& (q_r d_l).$$

Целиком положение робота описывается как совокупность описаний положения ног, наклона и положения роботов относительно препятствий описывается так:

$$s = s_n \& s_t \& s_d.$$

Таким образом, с помощью языка описания ситуаций были описаны все основные ситуации на каждом шаге передвижения робота. Теперь перейдем к следующей составляющей ситуационного управления – обобщению ситуаций.

2.2. Обобщение ситуаций

В модели ситуационного управления существует три типа обобщения ситуаций: обобщение по именам, обобщение по признакам и обобщение по структурам [Раков, 2012]. Поскольку в нашем случае конкретные имена и значения не используются, в качестве основного метода обобщения ситуаций выбран метод обобщения по структурам. Было решено произвести обобщение переменных h – поднятие ноги и t – касание в новую переменную-класс a – тип поверхности под ногой, т.к. эти переменные тесно связаны, и согласно принципам адаптивности отсутствие касания возможно после окончания движения только при полностью опущенной ноге (в случае неадаптивного движения положения ног также соответствуют значению полного опускания). Соотношение переменных приведено в таблице 1.

Таблица 1

h – поднятие ноги	t – касание	a – тип поверхности под ногой
Полностью опущена	есть	глубокая яма
Полностью опущена	нет	яма
Сильно опущена	есть	неглубокая яма
Слабо опущена	есть	нормальный уровень поверхности
Поднята	есть	возвышенность

2.3. Классификация ситуаций и принятие решения

В реализуемой системе для задачи классификации было решено использовать нейронные сети типа LVQ. Для моделирования нейронных сетей использованы утилита Neural Network Toolbox в составе программного комплекса математического моделирования Matlab 2013 и библиотека на языке программирования C++.

На первом этапе исследования решено рассматривать на этапе классификации только переменные описывающие наклон и тип поверхности, решение о направлении движения по датчикам-дальномерам принимается отдельно, вне модели ситуационного управления.

На основании первых экспериментов было принято решение, что анализ поверхности будет проводиться после совершения роботом четырех полных движений. После совершения роботом четырех движений в файле анализа данных появляется набор, достаточный для классификации, состоящий из 28 показателей:

- 6 показаний сенсоров (согласно классификации) за 4 движения;
- показаний акселерометра по осям X и Y за первые и вторые два движения.

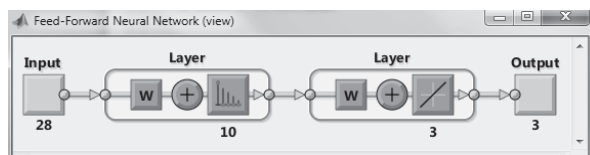


Рис.2. Структурная схема нейронной сети

Для обучения нейронной сети и были созданы 100 обучающих примеров, с заранее известным отношением каждого случая к определенному классу поверхности. Основная задача классификации – определить принадлежность ситуации к одному из 5 классов:

- Робот стоит на ровной поверхности;
- Робот стоит на поверхности с постоянным наклоном (уклон может быть направо, налево, вперед и назад);
- Робот стоит на неровной (бугристой) поверхности без явно выраженного наклона;
- Робот стоит на неровной поверхности с наклоном;

- Робот не имеет опоры на поверхности, зацепился за препятствие (4 и более ноги не касаются поверхности).

На основании отнесения каждой ситуации к конкретному классу система управления принимает решение о выборе алгоритма для дальнейшего передвижения. Если робот не имеет опоры на поверхности, то подается сигнал об остановке движения, если поверхность ровная и нет наклона, следующее движение будет неадаптивным. Если же ситуация относится к остальным трем классам, выбираются определенные комбинации адаптивного передвижения с использованием методов нащупывания поверхности, остановки и стабилизации. После выбора алгоритма система управления на основании показания датчиков-дальномеров определяет направление передвижения и затем начинается передвижение робота. Как показали первые опыты, благодаря такой системе управления робот действительно передвигался быстрее и устойчивее по неровной поверхности.

Заключение

В данной работе рассмотрена возможность применения ситуационного управления для создания адаптивной системы управления движением шестиногого робота. Ситуационное управление в разработанной системе управления роботом применяется для оценки ситуации и выбора одного из алгоритмов передвижения, которые используют различные комбинации трех базовых принципов адаптивности шестиногого шагающего робота. Оценка ситуации производится на основании анализа положения всех конечностей робота, факта касания конечностями поверхности и наклона робота относительно поверхности по показаниям акселерометра. Такой подход позволяет выбирать наиболее подходящий с точки зрения скорости и устойчивости алгоритм передвижения на каждом этапе движения робота. Реализованная система в настоящий момент использует не все принципы ситуационного управления и является упрощенной моделью ситуационного управления. Как показали первые опыты, такую систему можно рассматривать как основу для создания адаптивной системы управления шагающим роботом.

СПИСОК ЛИТЕРАТУРЫ

[Луцкий, 2013] Луцкий В.А. VII – Исследование адаптивных алгоритмов передвижения шестиногого шагающего робота // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов VII Международной конференции (Коломна, 20–22 мая 2013 г.). В 3-х томах. Т.2. – М.: Физматлит, 2013.

[Поспелов, 1986] Поспелов Д.А. Ситуационное управление: теория и практика. М.: Наука. – Гл. ред. физ.-мат. лит., 1986.

[Раков, 2012] Раков В.И. Системный анализ (Начальные понятия). – М.: Академия естествознания, 2012.

[Ющенко, 2007] Ющенко А.С. Ситуационное управление и робототехника // Материалы III Поспеловских чтений «Искусственный интеллект сегодня. Проблемы и перспективы». – М., 2007.

[Bruijn, 2013] Assessing the stability of human locomotion: a review of current measures S. M. Bruijn, O. G. Meijer, P. J. Beek and J. H. van Dieen Journal of the Royal Society. Interface 10, 20120999, 2013.

[Figliolini, 2007] G.Figliolin and P.Rea Mechanics and Simulation of Six-Legged Walking Robots, Climbing and Walking Robots. DiMSAT, University of Cassino, Cassino (FR), Italy, 2007.

[Knight, 2002] R. Knight and U. Nehmzow Walking Robots . A Survey and a Research Proposal Technical Report CSM-375 Department of Computer Science, University of Essex, United Kingdom, 2002.

УДК 004.89:510.64

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ ДЛЯ МАНИПУЛЯЦИОННЫХ ЗАДАЧ

В.Е. Павловский (*vlpavl@keldysh.ru*)

Институт прикладной математики им. М.В. Келдыша РАН, Москва

Е.А. Ефимова (*yefi03@yandex.ru*)

А.П. Алисейчик (*atooxa@gmail.com*)

И.А. Орлов (*orlovbel@gmail.com*)

Е.Д. Елагина (*lizzaelagina@yandex.ru*)

Российский государственный гуманитарный университет, Москва

В работе описываются интеллектуальные функции, разработанные для роботов-манипуляторов. Описан универсальный планировщик, предназначенный для планирования действий манипуляторов в сложных средах, он отработан на компьютерных моделях роботов. Описан интеллектуальный робот для игры в логические настольные игры, описана его логическая программа, разработанная на ПРОЛОГе. Созданные роботы прошли экспериментальную отработку, которая позволила сделать вывод об эффективности созданных программных и аппаратных средств.

Введение

В настоящее время на первый план по актуальности выходят задачи создания интеллектуальных роботов. Среди них активно исследуются задачи создания интеллектуальных роботов-манипуляторов, способных автоматически принимать решения непосредственно во время работы. Отрабатывать такие системы удобно на лабораторных задачах типа различных игр, требующих манипуляции предметами.

В данной работе рассматриваются программные и аппаратные средства для реализации интеллектуальных роботов-манипуляторов. Описываются две задачи и решающие их системы. Первая – универсальный планировщик PlannerM, предназначенный для планирования последовательностей действий роботов. Он демонстрируется на классической задаче-головоломке по раскладыванию кубиков, выполняют эту задачу различные роботы-манипуляторы. Вторая система – робот ManGo для решения игровых интеллектуальных задач для роботов, задач класса настольных логических

игр. Изначально ManGo был создан для задачи игры в «Го» с человеком, однако в качестве первого, более простого, приложения рассматривается игра «Крестики-Нолики» на большом (потенциально неограниченном) поле, на которой отрабатывается управление этим манипулятором.

1. Планировщик PlannerM

При создании планировщика PlannerM (*planner for manipulator*) был реализован модифицированный вариант аналогичного несложного планировщика, описанного в [Братко, 2004].

Для представления задачи планировщика используется схема, называемая пространством состояний. Пространство состояний – это граф, в котором узлы соответствуют проблемным ситуациям, а решение заданной проблемы сводится к поиску пути в этом графе. Ребра графа соответствуют действиям, допустимым в рассматриваемом мире. Если из вершины А можно перейти в вершину В по ребру, значит, этому ребру соответствует действие, переводящее ситуацию А в ситуацию В. На рис.1 показана типичная задача перекладывания кубиков (реструктурирования набора блоков) из начального состояния А в конечное состояние В.



Рис. 1. Задача реструктурирования набора блоков

Задача состоит в том, чтобы найти план перекладывания кубиков, чтобы попасть из ситуации, изображенной на левом рисунке, в ситуацию, изображенную на правом рисунке. Кубики можно перекладывать только по одному. Можно брать только тот кубик, на котором не стоит другой кубик. Ставить кубики можно только на другие кубики или на свободные места. В заданном мире ситуации описываются с помощью двух типов отношений-состояний: $on(a, b)$ (а стоит на b) и $clear(a)$ (на а ничего не стоит). Требуется найти последовательность действий, приводящую к желаемому результату.

тату. Исходная ситуация в данном примере описывается в виде неупорядоченной последовательности: $[on(c, a), on(a, 1), on(b, 3), clear(c), clear(2), clear(b), clear(4)]$. Целевая ситуация также записывается в виде неупорядоченной последовательности. Искомый план представляет из себя последовательность действий. В заданном мире существует только один тип действия: $move(From, Block, To)$, (переложить $Block$ с $From$ на To). На рис.2 ниже дана концептуальная блок-схема планировщика.

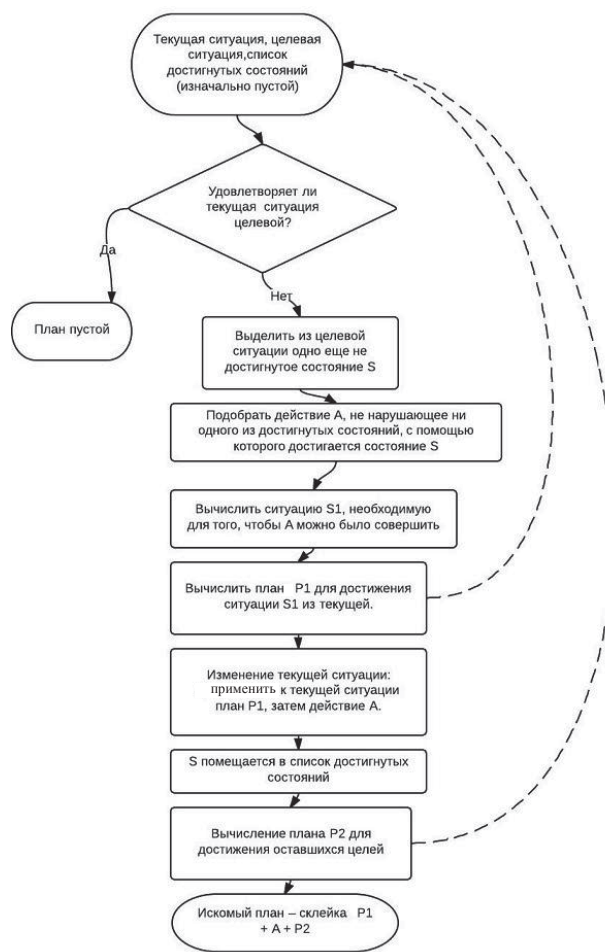


Рис. 2. Блок-схема планировщика

А далее на рис. 3 показана схема вычислений в планировщике при решении указанной конкретной задачи.

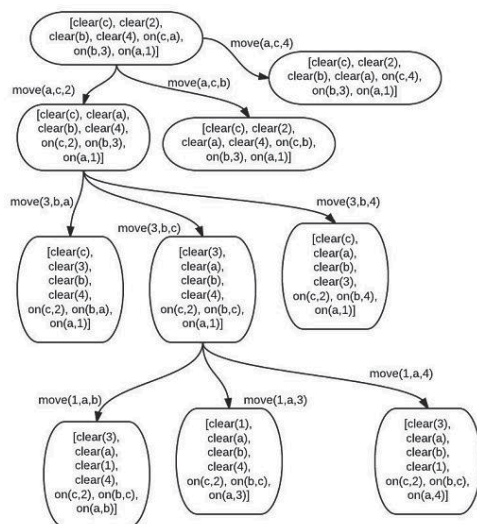


Рис. 3. Вычисления при составлении плана

Приведенная схема решения задачи планирования была отработана в комплексе математического моделирования. На рис.3 показаны различные фазы работы манипулятора при решении описанной задачи. Показан робот с ПУМА-подобной кинематикой, использовались также другие роботы. Моделирование подтвердило эффективность PlannerM.



Рис. 4. Трехмерное моделирование

2. Робот для игры в настольные логические игры. Алгоритм

Древняя китайская игра с японским названием «Гомоку» в России также известна как «Крестики-нолики» или «Пять в ряд». Два игрока по очереди ставят знаки на бесконечную клетчатую доску или на доску размером $m \times n$ до тех пор, пока один из них не выстроит пять своих знаков подряд по вертикали, по горизонтали или по диагонали или пока доска не будет заполнена полностью, если она конечная. В последнем случае признается ничья. В случае бесконечной доски ограничивается количество ходов. Игрок всегда ставит один и тот же знак – «крестик» или «нолик». Поле (клетка) доски может содержать не более одного знака.

Вместо «крестиков» и «ноликов» в оригинальной версии игры используются камни черного и белого цветов. В игре «Гомоку» ход делается не внутрь поля, а в узлы решетки, т. е. на пересечение линий.

Размер доски обычно ограничивается листом клетчатой бумаги. Для игры «Гомоку» используется доска размером 15×15 или 19×19 (как в игре «Го»). В программе, которая реализует ход робота, используется доска 20×20 .

Нетрудно доказать, что при правильной игре первый игрок может сыграть не хуже, чем вничью. Труднее доказать, что он выигрывает, но это также известно. Первый игрок имеет преимущество, поэтому второму игроку дается фора, состоящая в том, что первому игроку запрещаются некоторые ходы. В частности, это происходит в «Рендзю» – одном из вариантов игры «Гомоку». Известно, что профессиональные игроки в «Рендзю», когда ходят первыми, к десятому ходу получают перевес, а к пятнадцатому выигрывают. В [Allis et al., 2012] представлена программа Victoria, которая всегда выигрывает в игре «Гомоку» на доске 15×15 при ходе первой.

В компьютерных приложениях, реализующих игру «Пять в ряд», ход игрока-компьютера часто определяется с помощью оценочной функции. Для каждого пустого поля с индексами (i, j) подсчитывается его оценка (действительное число), учитывающая как выгоду игрока от хода в это поле, так и выгоду противника, которую он мог бы получить, если бы сделал ход в это поле. В качестве оценочной функции используется, например, функция вида

$$ev(i, j) = ev_x(i, j) + a \cdot ev_o(i, j),$$

где $ev_x(i, j)$ – оценка хода игрока в это поле, а $ev_o(i, j)$ – оценка хода его противника. Коэффициент a обратно пропорционален «агрессивности» игрока. При большей его величине стратегия игрока имеет оборонительный характер, при меньшей величине – наступательный.

Программа для робота написана на языке Пролог. В играх на языке Пролог ходы игрока-компьютера обычно реализуются с помощью продукционных правил. В созданной системе используются как правила, так и оценочная функция.

На вход программы подается массив, состоящий из чисел 0, 1 и 2, которые соответствуют пустым и занятым полям «камнями» темного и светлого цвета. Программа возвращает номер ряда и номер столбца поля, в который робот делает ход.

Для отладки стратегии игрока-робота было создано приложение на языке Visual Prolog 7.4, одной из самых развитых реализаций языка Пролог. Быстрый компилятор позволяет не заботиться о деталях алгоритма, а формулировать правила декларативным образом, что увеличивает скорость написания программы. Окно, содержащее игровую доску, представлено на рис. 5. Пользователь может выбрать игрока, который делает ход первым. Если им является компьютер, то окно открывается с уже сделанным им ходом. Первый ход производится случайным образом в поле, отстоящее не более чем на три ряда от центра доски. Ход пользователя выполняется с помощью клика мыши по полю.

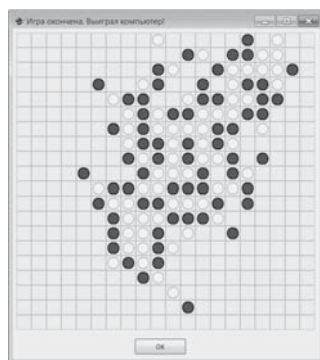


Рис. 5. Окончательная позиция. Компьютер играл «белыми»

Правила хода игрока формулируются следующим образом:

- поставить свой знак пятым в ряд;
- не дать противнику поставить пятый знак в ряд (поставить «блок»);
- не дать противнику поставить четыре знака в ряд;
- поставить свой знак четвертым в ряд;
- поставить «вилку»;
- не дать противнику поставить «вилку»;
- поставить знак в поле, которое находится в максимальном количестве рядов с тремя знаками противника и двумя пустыми полями;
- поставить знак в поле, которое находится в максимальном количестве рядов с тремя знаками противника и хотя бы одним пустым полем;
- поставить знак в поле, оценка которого имеет максимальное значение.

Среди полей с максимальным значением оценочной функции отбираются поля, которые граничат с максимальным количеством клеток, занятых противником. Поле среди отобранных полей выбирается случайным образом.

Для поиска хода используется описанная выше оценочная функция. В качестве оценки ряда взята функция

$$ev(s, k) = 4^{k+1},$$

где s – знак, а k – количество знаков s в ряду; параметр a положен равным единице.

Для хранения позиции в программе используются списки; в целях уменьшения перебора хранятся только занятые клетки. Списки в языке Пролог – это сохраняющий (*persistent*) тип данных, для них возможен откат. В отличие от них, в модифицируемых (*modifiable*) типах данных, например в массивах в языке С, изменения делаются в памяти компьютера, поэтому откат для них невозможен. Использование списков позволяет наиболее просто моделировать ситуацию наперед, например, проверить, не возникнет ли опасная ситуация, если в некоторое поле поставить свой знак или знак противника.

3. Описание манипулятора ManGo и его системы управления

Манипулятор ManGo имеет SCARA-подобную кинематику, которая в наибольшей степени подходит для задач манипулирования предметами в плоскости, в число которых входят настольные игры [Орлов 2013]. Первые шаги по исследованию кинематики были проведены в процессе проектирования робота в программном комплексе САПР, в качестве исполнительных приводов использовалась пневматика итальянской компании «Пневмакс» [Электронный ресурс]. Были рассчитаны оптимальные длины звеньев и точки крепления пневматических приводов для покрытия рабочей зоны размером 500x500 мм, в которой возможно разместить практически любую интеллектуальную логическую настольную игру.

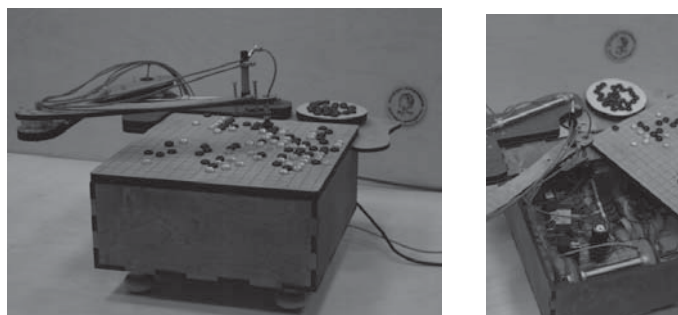


Рис. 6. Общий вид манипулятора ManGo (слева) и системы управления (справа)

В связи с использованием простой двухзвенной кинематической схемы решение обратной задачи кинематики для управления роботом – тривиально. В связи с этим было выбрано наиболее надежное в данном случае траекторное кинематическое управление с обратной связью по углам поворота в шарнирах. Использование в роботе пневматических цилиндров влечет за собой как ряд плюсов (дешевизна, высокие скорости движения, большие усилия и т.п.), так и минусов, основным из которых является сложность точного управления. Для решения данной проблемы было реализовано клапанное ШИМ-управление цилиндрами.

Заключение

Выполненные исследования подтвердили эффективность созданных программно-аппаратных средств и их адекватность поставленным задачам. Развитие этих средств будет выполняться по пути увеличения скорости их работы и усложнения их логических способностей.

СПИСОК ЛИТЕРАТУРЫ

[**Братко, 2004**] Братко Иван. Алгоритмы искусственного интеллекта на языке PROLOG. 3-е издание. М.: Изд-во «Вильямс», 2004. – 640 с.

[**Allis et al., 2012**] L.V. Allis, J. van den Herik, M.P.H. Huntjens. Go-Moku and Threat-Space Search // <http://www.mimuw.edu.pl/~awojna/SID/referaty/Go-Moku.pdf>.

[**Орлов, 2013**] Орлов И.А. Синтез движения манипуляционных систем для пространств со сложными связями и ограничениями : дисс.: 15.10.2013 / И.А. Орлов. – Москва, 2013. – С. 101.

[**Электронный ресурс**] Официальный сайт компании Pneumax S.p.A / <http://www.pneumaxspa.com/>.

СЕКЦИЯ 9 | **МАШИННОЕ ОБУЧЕНИЕ
И ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ
ДАнных**

УДК 007.5:519.816:681.3.016

**ПРОБЛЕМА ОБНАРУЖЕНИЯ АНОМАЛИЙ В НАБОРАХ
ВРЕМЕННЫХ РЯДОВ¹**

С.Г. Антипов (*antisergey@gmail.com*)

В.Н. Вагин (*vagin@appmat.ru*)

Национальный исследовательский университет
«Московский энергетический институт», Москва

В работе рассматривается проблема обнаружения аномалий в наборах временных рядов. Исследуется случай, когда обучающее множество содержит наборы временных рядов, относящихся к нескольким классам. Предлагается алгоритм TS-ADEEP-Multi для случая обучающего множества с несколькими классами. Приводятся результаты программного моделирования.

Введение

В настоящее время системы интеллектуального анализа данных работают со все более сложными техническими объектами и системами. Во многих случаях описание поведения сложных систем содержит сотни независимых атрибутов, которые необходимо анализировать, чтобы наиболее точно смоделировать поведение системы [Вагин и др., 2008]. Важным параметром, отражающим динамику поведения сложной системы, является время. Обобщение информации, отражающей изменение ситуаций во времени, требует использования специальных методов.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-00862а).

Одним из видов динамической информации являются временные ряды. Временные ряды используются в различных областях (экономика, медицина) и описывают длительные процессы, протекающие во времени.

В общем случае временной ряд TS – это упорядоченная последовательность значений $TS = \langle ts_1, ts_2, \dots, ts_n \rangle$, описывающая протекание какого-либо длительного процесса. Значениями ts_i могут быть показания датчиков, цены на какой-либо продукт, курс валюты и т. п. Временной ряд удобно представлять графически, где i -ая точка на графике соответствует значению параметра ts_i , полученному в момент времени i (время t полагается дискретным).

1. Постановка задачи

В данной работе решается задача определения аномалий в наборах временных рядов. Задача определения, или обнаружения, аномалий [Chandola et al., 2009] была поставлена как задача поиска в наборах данных образцов, не удовлетворяющих некоторому предполагаемому типовому поведению. Возможность найти аномалии в некотором наборе данных важна в различных предметных областях – при анализе работы сложных технических систем (например, телеметрии спутников), анализе сетевого трафика, в медицине (анализ снимков МРТ), в банковском деле (анализ транзакций, производимых с помощью кредитных карт) и др.

Аномалия, или «выброс», определяется как элемент, который явно выделяется из набора данных, к которому он принадлежит, и существенно отличается от других элементов выборки. Предположительно, в нашем распоряжении имеется коллекция временных рядов, описывающих некоторые процессы. Эта коллекция используется для описания нормального протекания процессов. Требуется на основании имеющихся данных построить модель, которая является обобщенным описанием нормальных процессов и позволяет различать нормальные и аномальные процессы.

Пусть имеется набор объектов, где каждый объект есть временной ряд: $TSGOOD = \langle TS_{good_1}, TS_{good_2}, \dots, TS_{good_m} \rangle$ – обучающая выборка. Каждый из временных рядов в обучающей выборке является примером «нормального» протекания некоторого процесса. На основании анализа временных рядов из $TSGOOD$ необходимо построить модель, позволяющую относить временные ряды из эк-

экзаменационной выборки $TSTEST = \langle TStest_1, TStest_2, \dots, Tstestm_1 \rangle$ к «нормальным рядам» или «аномалиям» на основании некоторого критерия.

В работе рассматривается задача поиска аномалий для случая, когда обучающее множество состоит из примеров нескольких классов. Задача усложняется тем, что набор исходных данных ограничен и не содержит примеров аномальных процессов; также не задан критерий, по которому можно было бы различать «нормальные» и «аномальные» временные ряды. К тому же многие алгоритмы, хорошо показавшие себя на одних наборах данных, совершенно не подходят для других предметных областей.

2. Алгоритм обнаружения аномалий

Временные ряды, которые представляют данные из разных областей, в различных единицах измерения, требуется привести к некоторым типовым, удобным для дальнейшего анализа формам. Предлагается использовать символическое представление (symbolic aggregate approximation [Lin et al., 2003]). Получение символического представления осуществляется в два этапа. На первом шаге получается нормализованное представление – среднее для временного ряда необходимо сделать равным нулю, а среднеквадратичное отклонение – единице; такое преобразование является необходимым процессом при предварительной обработке данных.

В данной работе предлагается алгоритм TS-ADEEP-Multi, который является обобщением алгоритма TS-ADEEP [Антипов и др., 2012] для случая обучающего множества, содержащего примеры нескольких классов временных рядов. Обобщение является достаточно очевидным: разделив обучающее множество на подмножества, содержащие примеры только одного класса и последовательно применив к ним и каждому из временных рядов экзаменационного множества алгоритм TS-ADEEP, можно определить, является ли рассматриваемый временной ряд аномалией. Для случаев, когда

– временной ряд является аномалией для каждого подмножества;

– временной ряд не является аномалией для единственного подмножества обучающей выборки, ответ очевиден. Однако возможен случай, когда временной ряд не является аномалией для нескольких классов из обучающего множества (но не для всех).

Псевдокод алгоритма TS-ADEEP-Multi приведен далее

Алгоритм **TS-ADEEP-Multi**

(*TSGOOD*: обучающее множество, содержащее примеры нескольких классов;

TSTest: экзаменационное множество)

Результат: *TSAnom_Optimistic* – набор временных рядов-аномалий по «оптимистичной» оценке

TSAnom_Pessimistic – набор временных рядов-аномалий по «пессимистичной» оценке

НАЧАЛО

TSAnom_Optimistic = \emptyset

TSAnom_Pessimistic = \emptyset

Пусть *N* – число классов, содержащихся в обучающем множестве
TSGOOD_C = { *TSGOOD_C*₁, *TSGOOD_C*₂, ..., *TSGOOD_C*_{*N*}} – разбиение множества *TSGOOD* такое, что *TSGOOD_C*_{*k*} содержит только примеры класса *k*, *k*=1..*N*

Для *j* от 1 до |*TSTest*|

нц

выбрать *TSTest*_{*j*} из *TSTest*

Для *k* от 1 до *N*

нц

$I = TSGOOD_C_k \cup TSTest_j$

Найти множество-исключение *I*_{*x*} в *I*

Если *TSTest*_{*j*} ∈ *I*_{*x*}, то *TSTest*_{*j*} является аномалией для класса *k*

(то есть не принадлежит ему)

кц

Если *TSTest*_{*j*} не принадлежит ни одному из классов

*TSGOOD_C*_{*k*}, *k*=1..*N*, то

начало

TSAnom_Optimistic = *TSAnom_Optimistic* ∪ *TSTest*_{*j*}

TSAnom_Pessimistic = *TSAnom_Pessimistic* ∪ *TSTest*_{*j*}

конец

Если *TSTest*_{*j*} принадлежит единственному классу

*TSGOOD_C*_{*k*},

то $TSTest_j$ не является аномалией
Если $TSTest_j$ принадлежит нескольким классам $TSGOOD_{C_k}$,
то $TSAnom_Pessimistic = TSAnom_Pessimistic \cup TSTest_j$

кц

вывести TSAnom

КОНЕЦ

3. Результаты моделирования

Моделирование процесса обнаружения аномалий было проведено как на искусственных, так и на реальных данных. В качестве искусственных данных были взяты классические описания временных рядов, используемые в научной литературе: «cylinder-bell-funnel» [Naoki et al., 2008] и «control chart» (контрольные карты) [Pham et al., 1998]. В качестве реальных данных – «трафик» – использовались данные, собранные с помощью специальных систем анализа трафика при передаче файлов по различным протоколам.

«Cylinder-bell-funnel» [Naoki et al., 2008], как следует из названия, содержит три различных класса – «цилиндр», «колокол», «воронка».

«Control chart» [Pham et al., 1998] (контрольные карты) содержит шесть различных классов, описывающих тренды, которые могут присутствовать в процессах: цикличность, уменьшение значения, резкое падение, увеличение значения, постоянная величина, резкое возрастание.

«Трафик» – данные, полученные на основе анализа трафика при передаче файлов по протоколу ftp в различных условиях (в том числе при одновременной передаче нескольких файлов по нескольким протоколам). В качестве тестовых данных, помимо прочих, использовались специальным образом сгенерированные временные ряды, имитирующие передачу данных.

Для того чтобы определить, насколько хорошо предложенный алгоритм справляется с обнаружением аномалий в наборах временных рядов, было проведено его программное моделирование. Рассмотрим процесс моделирования на примере набора данных «cylinder-bell-funnel». Сначала в качестве обучающего множества TSGOOD генерируется набор временных рядов, принадлежащих двум из трех классов – например, «цилиндр» и «колокол». В качестве экзаменационного множества TSTEST генерируются временные

ряды, принадлежащие всем трем классам – «цилиндр», «колокол», «воронка». Временной ряд TStest_j является «нормальным», если он принадлежит классу «цилиндр» или «колокол», и «аномалией», если не принадлежит им. Соответственно, алгоритм корректно находит аномалии, если он относит временные ряды класса «воронка» из TSTEST к аномалиям, а временные ряды класса «цилиндр» и «колокол» аномалиями не считает. При этом были рассмотрены как численное представление временных рядов, так и символьное с разным размером алфавита. Аналогично моделирование проводилось для других пар классов: «колокол» и «воронка», «цилиндр» и «воронка».

Для временных рядов «цилиндр-колокол-воронка» в качестве обучающего множества использовались возможные комбинации из двух классов. Для временных рядов «контрольные карты» рассматривались все возможные комбинации из двух и трех классов. В результатах показаны худший, лучший и средний результаты определения аномалий.

Результаты моделирования для временных рядов «cylinder-bell-funnel» приведены в таблице 1, для временных рядов «control chart» – в таблице 2. В первом столбце таблиц указано, сколько классов содержит обучающее множество; в следующих двух столбцах указан способ представления временных рядов. Данные, приведенные в последних двух столбцах, дают возможность оценить результаты распознавания.

Таблица 1

Точность определения аномалий для набора данных «cylinder-bell-funnel»

Данные (число классов рядов в обучающем множестве)	Представление временного ряда	Число символов в алфавите	Количество правильно определенных временных рядов (мин/макс/среднее из 300)	Процент правильно определенных временных рядов (мин/макс/среднее)
2	Числовое	–	137/157/144	45.67/52.33/48.00
2	Символьное	20	159/214/185	53.00/71.33/61.67
2	Символьное	35	151/194/170	50.33/64.67/56.67
2	Символьное	50	150/173/157	50.00/57.67/52.33

Таблица 2

**Точность определения аномалий для набора данных
«control chart»**

Данные (число классов в обучающем множестве)	Представление временного ряда	Число символов в алфавите	Количество правильно определенных временных рядов (мин/макс/среднее из 600)	Процент правильно определенных временных рядов (мин/макс/среднее)
2	числовое	–	369/540/463	61.50/90.00/77.17
2	символьное	20	325/585/468	54.17/97.50/78.07
2	символьное	35	326/580/465	54.33/96.67/77.50
2	символьное	50	373/571/473	62.17/95.17/78.83
3	числовое	–	279 /532/428	46.50 /95.17/71.33
3	символьное	20	290 /580/439	48.33 /96.67/73.16
3	символьное	35	277 /574/434	46.17 /95.67/72.33
3	символьное	50	279 /557/436	46.50 /92.83/72.67

Как видно из таблиц 1 и 2, алгоритм определения аномалий может показывать не очень хорошие результаты – лишь чуть более половины временных рядов правильно отнесены или не отнесены к аномалиям. Для исправления такой ситуации предлагается процедура предварительной обработки исходного ряда на этапе нормализации. Цель ее – в сжатии ряда, что позволяет снизить шум [Arning et al., 1996]. Например, одна точка «сжатого» ряда сопоставляется десяти точкам исходного ряда. Такая процедура позволила существенно улучшить результаты. Результаты, приведенные в таблицах 3 и 4, показывают, что при использовании «сжатых» временных рядов алгоритм определения аномалий работает значительно лучше, чем при использовании исходных. Таким образом, дополнительная обработка входных данных позволяет, не изменяя самого алгоритма, значительно улучшить качество его работы.

Таблица 3

Точность определения аномалий для набора данных «cylinder-bell-funnel» для «сжатых» временных рядов

Данные (класс рядов в обучающем множестве)	Представление временного ряда	Число символов в алфавите	Количество правильно определенных временных рядов (мин/макс/среднее из 300)	Процент правильно определенных временных рядов (мин/макс/среднее)
2	Числовое	–	190/267/228	63.33/89.00/76.00
2	Символьное	20	179/254/218	59.67/84.67/72.67
2	Символьное	35	196/259/228	65.33/86.33/76.00
2	Символьное	50	182/264/225	60.67/88.00/75.00

Таблица 4

Точность определения аномалий для набора данных «control chart» для «сжатых» временных рядов

Данные (число классов в обучающем множестве)	Представление временного ряда	Число символов в алфавите	Количество правильно определенных временных рядов (мин/макс/среднее из 600)	Процент правильно определенных временных рядов (мин/макс/среднее)
2	числовое	–	426/570/507	71.00/95.00/84.50
2	символьное	20	441/594/527	73.50/99.00/87.83
2	символьное	35	469/588/538	78.17/98.00/89.67
2	символьное	50	466/576/527	77.67/96.00/87.83
3	числовое	–	396/523/463	66.00/87.17/77.17
3	символьное	20	418/585/510	69.67/97.50/85.00
3	символьное	35	461/575/518	76.87/95.83/86.33
3	символьное	50	441/551/499	73.50/91.83/83.17

Заключение

В работе рассмотрена задача поиска аномалий среди наборов временных рядов. Предложен непараметрический алгоритм TS-ADEEP-Multi для определения аномалий в наборах временных

рядов для случая, когда обучающее множество содержит примеры, принадлежащие нескольким классам. Проведено программное моделирование предложенного алгоритма. Рассмотрен способ повышения точности определения аномалий – «сжатие» рассматриваемых временных рядов с целью избавления от несущественных деталей и шума. В дальнейшем предполагается модифицировать предложенный алгоритм для определения аномалий в наборах временных рядов для случая, когда классы временных рядов априори неизвестны.

СПИСОК ЛИТЕРАТУРЫ

[Антипов и др., 2012] Антипов С.Г., Фомина М.В. Проблема обнаружения аномалий в наборах временных рядов. – Программные продукты и системы, 2012, № 2, с. 79–81.

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. – Изд. 2-е, М.: ФизМатЛит, 2008.

[Arning et al., 1996] Arning A., Agrawal R., Raghavan P. A Linear Method for Deviation Detection in Large Databases.// In Proceedings of KDD'1996, p. 164–169.

[Chandola et al., 2009] Chandola V., Banerjee A. and Kumar V. Anomaly Detection – A Survey, ACM Computing Surveys, Vol. 41(3), Article 15, July 2009, pp. 1–72.

[Lin et al., 2003] Lin J., Keogh E., Lonardi S. and Chiu B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms// In Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003, p. 2–11.

[Naoki et al., 2008] Naoki Saito. Local feature extraction and its application using a library of bases. – PhD thesis, Yale University, December 1994.

[Pham et al., 1998] Pham D.T., Chan A.B. Control Chart Pattern Recognition using a New Type of Self Organizing Neural Network// Proc. Instn, Mech, Engrs. Vol 212, No 1, 1998, pp 115–127.

УДК 007.5:519.816:681.3.016

**ПРОБЛЕМА ФОРМИРОВАНИЯ
ОБОБЩЕННЫХ ПОНЯТИЙ ПРИ НАЛИЧИИ ШУМА
В РЕШАЮЩИХ АТТРИБУТАХ¹**

С.Г. Антипов (*antysergey@mail.ru*)
Л.А. Старостина (*StarostinaLA@mpei.ru*)
М.В. Фомина (*m_fomina2000@mail.ru*)
НИУ МЭИ, Москва

Рассматривается проблема обобщения знаний с учетом потребности обработки неполной и противоречивой информации, хранящейся в реальных массивах данных. Предлагается использовать методы построения продукционных правил и деревьев решений в сочетании с различными алгоритмами обработки «зашумленных» значений. Исследуется влияние шума на работу алгоритмов обобщения при условии внесения шума в атрибут, определяющий класс примера.

Введение

В настоящее время высок интерес многих организаций к средствам интеллектуального анализа данных. Это связано с тем, что подобные средства не только позволяют автоматизировать обработку больших объемов данных, но и повышают качество принимаемых управленческих решений, делая их более обоснованными.

Системы автоматического обнаружения знаний в базах данных имеют возможность перерабатывать и анализировать «сырые» данные, предоставляя извлеченную информацию скорее и успешнее, чем аналитик мог бы найти ее самостоятельно. Средства интеллектуального анализа данных успешно применяются в бизнесе, медицине, в банковской системе (например, для принятия решения о выдаче кредита), для решения различных экономических задач и т.д.

В связи с тем, что информация, хранящаяся в реальных базах данных, не обязательно является полной и точной, встает задача исследовать влияние различных видов шума на работу алгоритмов

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-01-00589а).

обобщения, извлекающих скрытые закономерности из БД, что делает данную работу актуальной.

1. Постановка задачи

Извлечение знаний из баз данных связано с решением задачи индуктивного формирования понятий, или задачи обобщения [Вагин и др., 2008].

Пусть $O = \{o_1, o_2, \dots, o_n\}$ – множество объектов, которые могут быть представлены в интеллектуальной системе S (например, в интеллектуальной системе поддержки принятия решений (ИСППР)). Пусть каждый объект характеризуется r признаками (атрибутами): A_1, A_2, \dots, A_r . Атрибутами могут быть, например, размер, форма, вес, цвет и т.п. Определим $Dom(A_1), Dom(A_2), \dots, Dom(A_r)$ множества допустимых значений атрибутов. Для атрибута A_k $1 \leq k \leq r$, $Dom(A_k) = \{x_{1k}, x_{2k}, \dots, x_{q_k k}\}$, где q_k – число различных значений атрибута A_k . Таким образом, каждый объект $o_i \in O$, $1 \leq i \leq n$ представлен множеством значений атрибутов, т.е. $o_i = \{x_{i1}, x_{i2}, \dots, x_{ir}\}$, где $x_{ik} \in Dom(A_k)$, $1 \leq k \leq r$. Такое описание объекта назовем признаковым описанием. Атрибуты объектов могут быть количественные, качественные или шкалированные [Вагин и др., 2008].

Пусть среди всех объектов O , представленных в ИСППР, имеется множество объектов, V , относящихся к некоторому классу или понятию. Назовем такие объекты положительными объектами для понятия. Все остальные объекты W представляют контрпримеры понятия. Очевидно, $O = V \cup W$ и $V \cap W = \emptyset$. Построим K непустое множество объектов, такое, что $K = K^+ \cup K^-$, где $K^+ \subset V$ и $K^- \subset W$. Будем называть K обучающей выборкой. На основании обучающей выборки надо построить правило, разделяющее положительные и отрицательные объекты обучающей выборки.

Скажем, что понятие сформировано, если удалось построить решающее правило, которое для любого примера из обучающей выборки указывает, принадлежит ли этот пример понятию, или нет. Алгоритмы обобщения, которые мы изучаем, формируют правила типа: «IF условия, THEN сформированное понятие». Решающее правило является корректным, если оно в дальнейшем успешно распознает объекты, не вошедшие первоначально в обучающую выборку.

Присутствие шума в данных изменяет вышеуказанную постановку задачи. Прежде всего, исходная обучающая выборка K заменяется выборкой K' , в которой с некоторой вероятностью встречаются искаженные или отсутствующие значения атрибутов. Работа с такими данными существенно затрудняет работу алгоритмов обобщения, поскольку обучающая выборка, на основе которой формируется обобщенное понятие, будет содержать неточные, а иногда и противоречивые примеры.

2. Алгоритмы обобщения

Наиболее известными методами решения поставленной задачи являются методы, основанные на теории приближенных множеств (алгоритм GIRS) [Pawlak, 2002; Vagin et al., 2008], методы деревьев решений (алгоритмы ID3, C4.5, CART [Quinlan, 1986; Quinlan, 1996; Breiman et al., 1984]) и построение наборов продукционных правил (алгоритм CN2 [Clark et al., 1989]), позволяющие решать задачу обобщения.

Целью данной работы является исследование влияния шума на работу алгоритмов обобщения, которые строят обобщенное понятие в виде деревьев решений. Дерево решений T это взвешенный граф, в котором при движении от корня к листьям в каждом промежуточном узле выполняется проверка некоторого условия по значениям одного из атрибутов, а каждый конечный узел взвешен именем класса.

Для исследования влияния шума на деятельность алгоритма обобщения был использован алгоритм C 4.5, в работу которого были внесены возможности обработки «зашумленных» данных.

Алгоритм C 4.5, так же как и его предшественник ID3, предложенный J.R.Quinlan [Quinlan, 1986], относится к классу алгоритмов, строящих классификационные правила в форме деревьев решений. Однако C4.5 имеет ряд преимуществ по сравнению с ID3, например, допускается использование непрерывных атрибутов.

Алгоритм C 4.5 основан на следующей рекурсивной процедуре:

1. Выбирается атрибут для корня дерева T и формируются ветви для каждого возможного значения этого атрибута.
2. Дерево используется для классификации множества примеров

обучающей выборки. Если все примеры одного листа принадлежат одному классу, этот лист помечается именем этого класса.

3. Если все листы помечены именами классов, то конец алгоритма. Иначе, текущая вершина помечается именем следующего атрибута, создаются ветви для каждого возможного значения этого атрибута и переход к пункту 2.

Критерием выбора очередного атрибута является коэффициент *gain ratio*, основанный на понятии энтропии [Quinlan, 1986].

3. Модели шума

Допустим, что примеры в обучающих выборках K содержат шум, т.е. значения атрибутов могут быть пропущены или искажены. Шум возникает по следующим причинам: некорректное измерение входных параметров, ошибочное описание значений параметров экспертом, использование неисправных измерительных устройств, потери данных при передаче и хранении информации.

Рассмотрим основные требования к данным, на основе которых выполняется обучение. Первым требованием является связь атрибутов, описывающих объекты, с решающим атрибутом. Для различных атрибутов эта связь может быть более или менее тесной. Вторым требованием является слабая взаимосвязь атрибутов между собой. Это позволяет алгоритмам обобщения рассматривать атрибуты как независимые [Zhu et al., 2004]. Следует отметить, что реальные наборы данных далеко не всегда соответствуют этим требованиям.

Шум, внесенный в обучающую выборку K , можно рассматривать как шум в описательных атрибутах, и шум в решающем атрибуте. В работах [Vagin et al., 2011; Vagin et al., 2008; Вагин и др., 2012] подробно рассматривалась проблема влияния шума в описательных атрибутах на качество обобщения для различных алгоритмов. Предлагается оценить влияние шума, внесенного в решающий атрибут, на результаты обобщения. Такое исследование важно, поскольку шум, внесенный в решающий атрибут (атрибут, определяющий, к какому классу относится пример), приводит к появлению *противоречивых* примеров (одинаковые объекты могут быть помечены именами различных классов), а также примеров, ошибочно отнесенных к какому-либо классу.

Одним из основных параметров исследования является уровень шума. Уровень шума – это величина p_0 , которая показывает, что значение атрибута в обучающем или тестовом множестве будет искажено.

Поскольку для экспериментальной проверки влияния шума были использованы наборы данных из репозитория UCI [Merz et al., 1998], изначально не содержащие шума, внесение шума в данные выполнялось искусственно. При этом явно задавались уровень шума и атрибуты, в которые принудительно были внесены искажения.

Для исследования были выбраны две модели шума: «отсутствующие значения», и «искаженные значения». В первом случае для заданного уровня шума с вероятностью p_0 известное значение атрибута в таблице удаляется. Вторым вариантом внесения шума связан с заменой известного значения атрибута на другое, допустимое, но неверное для данного примера. Значения для замены выбираются из областей $Dom(A_k)$, $1 \leq k \leq r$, величина p_0 определяет вероятность такой замены.

Если шум связан с отсутствием в таблице некоторых значений атрибутов, необходимо выбрать способ обработки «отсутствующих значений». Предлагается два пути: пропуск такого примера и восстановление отсутствующих значений, используя метод «ближайшего соседа» [Vagin et al., 2011].

4. Результаты эксперимента

Исследование проводилось в системе формирования обобщенных понятий «Induction of Decision Tree with restoring Unknown Values» (IDTUV3), описанной в [Fomina et al., 2014]. IDTUV3 включает процедуру восстановления неизвестных значений в присутствии примеров, содержащих шум типа «отсутствующие значения». Когда отсутствующие значения атрибутов восстановлены, используется один из алгоритмов формирования обобщенных понятий. Для исследования влияния шума на формирование обобщенных правил, а также на точность классификации тестовых примеров был выбран алгоритм C 4.5.

Система IDTUV3 выполняет следующие основные функции:
– загружает начальные данные из БД;

- вводятся различные типы шума в обучающих и тестовых множествах;
- строится классификационная модель (дерево решений или набор продукционных правил) на базе обучающих выборок;
- проводится классификация для примеров экзаменационных выборок;
- по результатам классификации собирается статистика.

Мы представляем результаты экспериментов, выполненных над наборами данных из известных коллекций множества тестовых данных из California University of Informatics and Computer Engineering «UCI Machine Learning Repository» [Merz et al., 1998].

Рассмотрим результаты эксперимента по внесению шума в решающий атрибут обучающей выборки на этапе построения дерева решений. В таблице 1 представлены результаты, полученные при внесении шума «неизвестные значения» указанного уровня в решающий атрибут обучающей выборки. После построения дерева решения алгоритмом C 4.5 выполнялась классификация примеров, не содержащих шума. При этом в обучающую выборку вносился шум «отсутствующие значения» с последующим восстановлением методом «ближайших соседей».

Таблица 1

Набор данных	Количество ближайших соседей	Точность классификации, %					
		Без шума	Шум 5%	Шум 10%	Шум 20%	Шум 30%	Шум 40%
Glass	3	67.29	66.82	66.26	65.23	65.42	61.28
	5		66.45	65.89	64.02	63.55	63.93
	7		66.82	66.64	67.01	63.08	61.50
	9		66.92	66.36	66.54	64.21	63.64
Heart	3	73.7	72.59	71.26	73.19	70.22	74.44
	5		73.63	72.74	72.89	69.41	70.37
	7		73.04	72.07	72.22	71.70	71.93
	9		73.26	73.04	72.59	73.93	73.26
Australian credit	3	84.78	84.17	83.33	82.52	81.59	81.25
	5		84.67	82.12	83.8	82.38	80.46
	7		84.43	83.62	83.68	81.42	80.81
	9		84.87	83.74	83.19	83.22	80.43

Продолжение таблицы 1

Набор данных	Количество ближайших соседей	Точность классификации, %					
		Без шума	Шум 5%	Шум 10%	Шум 20%	Шум 30%	Шум 40%
Monks-1	3	78.01	78.15	78.01	79.26	77.64	73.89
	5		82.18	77.78	77.96	78.98	74.07
	7		77.82	79.17	75.0	73.1	74.17
	9		78.29	78.7	75.42	71.3	74.03
Monks-2	3	66.67	66.16	65.14	64.72	63.38	63.24
	5		65.14	65.32	64.17	62.78	63.15
	7		64.91	64.21	64.44	62.36	62.18
	9		65.28	65.09	63.43	62.22	63.33
Monks-3	3	93.98	93.61	92.5	88.33	87.18	81.67
	5		92.73	91.67	87.82	83.89	82.41
	7		91.39	91.48	88.66	86.99	81.57
	9		92.45	90.83	90.60	85.37	81.85

Рассмотрим аналогично проблему влияния на работу алгоритма С 4.5 шума «перемешивание значений». Поскольку при использовании такой модели шума нельзя явно указать, где произошло искажение, процедуры коррекции данных после внесения шума не использовались.

Для сравнения в таблице 2 приведены результаты для тех же наборов данных, что и для первой модели шума.

Таблица 2

Набор данных	Точность классификации, %					
	Без шума	Шум 5%	Шум 10%	Шум 20%	Шум 30%	Шум 40%
Glass	67.29	66.17	62.8	61.21	60.09	48.97
Heart	73.7	72.74	63.63	66.37	56.89	53.11
Australian credit	84.78	79.39	75.1	68.9	62.38	55.36
Monks-1	78.01	76.3	74.12	70.65	62.59	53.38
Monks-2	66.67	64.07	62.50	58.19	53.38	49.68
Monks-3	93.98	89.07	81.99	70.74	62.45	53.70

Проведем сравнение полученных результатов. В [Vagin et al., 2011] было показано, что при внесении шума до 30% в информативные атрибуты обучающего множества точность классификации тестовых примеров снижается незначительно (0,5–2%). Для случая внесения шума в решающий атрибут мы получаем гораздо более ярко выраженный эффект. Шум «отсутствующие значения» в решающем атрибуте приводит к снижению точности классификации тестовых примеров на 2–8%. Наиболее сильное влияние оказала модель шума «перемешивание значений»: при внесении такого шума в решающий атрибут обучающей выборки точность классификации снизилась на 8–22%. При этом зависимость снижения точности классификации от уровня шума в выборке близка к линейной. Последнее связано с появлением в обучающей выборке большого количества противоречивых примеров при высоких (свыше 10%) уровнях шума.

Заключение

В работе рассматривалась задача обобщения информации и исследовалось влияние шума в исходных данных на эффективность классификационных моделей, полученных с помощью алгоритмов обобщения. Рассмотрены модели шума в таблицах баз данных, следствием которых является отсутствие значения признака, либо искажение значения признака в обучающей выборке.

При внесении шума в решающий атрибут обучающего множества выявлена высокая чувствительность алгоритма С 4.5 к шуму второго типа. В дальнейшем для снижения такого эффекта предлагается рассмотреть возможность использования алгоритмов обобщения, допускающих работу с противоречивыми данными.

СПИСОК ЛИТЕРАТУРЫ

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. – Изд. 2-е, М.: ФизМатЛит, 2008.

[Вагин и др., 2012] Вагин В.Н., Фомина М.В. Исследование влияния шума на работу алгоритмов обобщения / Тринадцатая национальная конференция по искусственному интеллекту с международным участием, КИИ-2012. Труды конференции. Т.1. – Белгород: Изд-во БГТУ, 2012.

[**Вагин и др., 2013**] Вагин В.Н., Антипов С.Г., Фомина М.В. Моделирование алгоритмов индуктивного формирования понятий в «зашумленных» базах данных. – НТИ, сер. 2. Информ. процессы и системы, № 7, М.: ВИНТИ, 2013.

[**Breiman et al., 1984**] Breiman L., Friedman J. H., Olshen R. A., Stone C. T. Classification and Regression Trees /Wadsworth, Belmont, California, 1984.

[**Clark et al., 1989**] P.Clark, T.Niblett. The CN2 Induction Algorithm, 1989.

[**Fomina et al., 2014**] Fomina M., Morosin O., Vagin V. Argumentation Approach and Learning Methods in Intelligent Decision Support Systems in the Presence of Inconsistent Data/ ICCS 2014. 14th International Conference on Computational Science. Procedia computer science. Vol. 29. Elsevier, 2014.

[**Merz et al., 1998**], Merz C, Murphy P., 1998. UCI Repository of Machine Learning Datasets. Information and Computer Science University of California, Irvine, CA 92697-3425 <http://archive.ics.uci.edu/ml/>.

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

[**Quinlan, 1986**] Quinlan J.R. Induction of Decision Trees// Machine Learning, Vol.1, 1986.

[**Quinlan, 1996**] Quinlan J.R. Improved Use of Continuous Attributes in C 4.5//Journal of Artificial Intelligence Research, Vol. 4, 1996.

[**Vagin et al., 2008**] Vagin V.N., Fomina M.V., Kulikov A.V.: The Problem of Object Recognition in the Presence of Noise in Original Data . In: 10th Scandinavian Conference on Artificial Intelligence SCAI 2008. pp. 60-67(2008).

[**Vagin et al., 2011**] Vagin V., Fomina M. Problem of Knowledge Discovery in Noisy Databases.// International Journal of Machine Learning and Cybernetics. 2011. Vol. 2, Number 3.

[**Zhu et al., 2004**] Zhu, X., Wu, X, Yang, Y. Error Detection and Impact-sensitive Instance Ranking in Noisy Datasets. //Proceedings of 19th National conference on Artificial Intelligence (AAAI-2004), San Jose, CA, 2004.

УДК 004.8

**ИНТЕЛЛЕКТУАЛЬНЫЙ ВЕБ-СЕРВИС
ЭКСПРЕСС-АНАЛИЗА ЭКОНОМИЧЕСКОГО СОСТОЯНИЯ
ПРЕДПРИЯТИЯ¹**

Т.В. Афанасьева (*tv.afanasjeva@gmail.com*)

Г.Ю. Гуськов (*guskovgleb@gmail.com*)

Ульяновский государственный технический университет, Ульяновск

И.Г. Перфильева (*irina.perfilieva@osu.cz*)

Университет г. Острава, Чешская Республика

А.А. Романов (*romanov73@gmail.com*)

И.А. Тимина (*i.timina@ulstu.ru*)

Н.Г. Ярушкина (*jng@ulstu.ru*)

Ульяновский государственный технический университет, Ульяновск

Выбор стратегии развития большого предприятия зависит от оценки его эффективности. В статье описывается программный продукт, который сочетает в себе экспертную систему и веб-сервис. Анализ и лингвистическая интерпретация нелинейных зависимостей в исходных данных (временных рядах) производятся с применением методов искусственного интеллекта и методов нечеткой логики, включая метод F-преобразования и метод нечетких тенденций. Результаты анализа рядов используются для пополнения базы знаний сервиса новыми правилами.

Введение

Авторами был разработан интернет-сервис для экспресс-анализа экономических показателей на основе интегрального метода нечеткого моделирования и прогнозирования нечетких тенденций (<http://tsas.ulstu.ru>). Эта интерактивная веб-система была разработана, чтобы служить в качестве фреймворка для топ-менеджеров, аудиторских фирм, любого эксперта-консультанта.

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 13-01-00324). Автор Ирина Перфильева выражает благодарность проекту М.МТ-7026/2012-36 (KОНТАКТ II, LH 12229 «Research and development of methods and means of intelligent analysis of time series for the strategic planing problems») за частичную поддержку при проведении исследований, связанных с тематикой данной статьи.

При этом перечень решаемых сервисом задач позволяет:

- автоматически собирать информацию о сегменте рынка указанного типа бизнеса;
- получать интегрированные экспресс-оценки деятельности по управлению предприятием;
- сравнивать индексы производительности со средним уровнем таких же индексов для всего множества аналогичных компаний с целью совершенствования навыков управления и увеличения доходов компании;
- накапливать знания о состоянии бизнеса в пределах определенного сегмента рынка и вербализовать их совокупный опыт и сформулировать на этой основе технические процедуры для углубленного анализа эффективности деятельности в целях развития клиентской сети.

Исходными данными для анализа являются публичная финансовая отчетность, объединенная по группам: ликвидность и платежеспособность, показатели рентабельности и показатели активности. Предполагается, что анализируемые отчетности доступны за определенный период и могут быть выражены в виде временных рядов (ВР) – последовательности значений через равные интервалы времени.

Исходные данные представлены четкими значениями и преобразуются в нечеткие заключения, прогнозы и рекомендации. Имеются показатели, изменения которых характеризуются большой степенью сходства. Это говорит и о схожем характере поведения локальных тенденций временных рядов за определенный период. Проводя анализ с целью выявления групп похожих временных рядов показателей, можно выявлять информацию об отношении каждого показателя к значимому фактору, сформировать новые правила и обеспечить самообучение системы.

1. Построение F-компонент высшего порядка

Для каждого временного ряда задаются функции нечеткого разбиения, строится F-преобразование нулевого порядка (F^0 -преобразование). Для построения F-преобразования первого порядка [Perfilieva, 2011] (F^1 -преобразование): $F_k^1 = c_{k,0}^1 + c_{k,1}^1(x - x_k)$, необходимо определить коэффициенты $c_{k,0}^1, c_{k,1}^1$. В формуле $k=1, \dots, n$,

n – количество компонент для ряда. Коэффициенты $C_{k,0}^1$ являются компонентами F^0 -преобразования. Коэффициенты $C_{k,1}^1$ определяются так:

$$C_{k,1}^1 = \frac{\sum_{t=x_{k-1}}^{x_{k+1}} f(t)(t-x_k)A_k(t)}{\sum_{t=x_{k-1}}^{x_{k+1}} (t-x_k)^2 A_k(t)},$$

где $f(t)$ – значения временного ряда в момент времени t , $A_k(t)$ – базисные функции нечеткого разбиения для F^1 -преобразования в момент времени t_k . Коэффициенты $C_{k,1}^1$ равны средневзвешенному значению тангенса угла наклона кусочно-линейного тренда временного ряда. Именно они будут служить основой анализа групп временных рядов. Сами компоненты F^1 -преобразования представляют собой векторы и описывают локальные тенденции временного ряда.

2. Формирование групп временных рядов

Группирование временных рядов будем производить, анализируя коэффициенты $C_{k,1}^1$. Для каждой пары временных рядов вычисляется коэффициент корреляции Пирсона (линейный коэффициент корреляции), при вычислении которого используются значения коэффициентов $C_{k,1}^1$ для первого и второго временного ряда из пары соответственно.

Важным условием для проведения анализа является равная длина временных рядов. Перед построением компонент F -преобразования различного порядка производится нормировка значений временного ряда к интервалу $[0, \dots, 1]$ по модулю. Нормировка необходима для обеспечения одинаковой области значений коэффициентов $C_{k,1}^1$. При формировании групп временных рядов возникают следующие проблемы.

Необходимо правильно выбрать количество точек, которое покрывает базисная функция, – данная величина непосредственно влияет на последующую корреляцию временных рядов по коэффициенту $C_{k,1}^1$. Чем большее количество точек будет покрывать базисная функция, тем более сглаженным будет ряд, и корреляция будет рассчитываться по общим тенденциям рядов.

Задание порога влияет на количество похожих по коэффициенту корреляции временных рядов. Необходимо определить, что более важно: включить в группу больше рядов или отобрать только самые близкие по корреляции ряды.

Степень сходства ВР осуществляется на основе меры ассоциации ВР, построенной по методу скользящего окна [Batyrshin et al, 2005].

3. Моделирование нечетких тенденций

Модель ряда должна быть идентифицирована и формально описана. Основа модели – зависимость, связывающая элементы рядов. Построение зависимости есть процесс извлечения знаний [Zadeh, 1979].

Предлагаемый подход к построению моделей ВР для целей решения задачи прогнозирования подразумевает преобразование исходного дискретного ВР $Y = \{t_i, x_i\}, i \in [1, n]$, где t_i – некоторый момент времени, x_i – уровень ВР, в нечеткий ВР (НВР) $\tilde{Y} = \{t_i, \tilde{x}_i\}, i \in [1, n]$, $X = \{x_i\}, \tilde{x}_i \in \tilde{X}$, где \tilde{x}_i – нечеткая метка. Затем НВР преобразуется в ВР нечеткую элементарную тенденцию (НЭТ), которую обозначим $\tau_i = ETend(\tilde{x}_i, \tilde{x}_{i+1})$. Под *ETend* понимается функционал, который реализуется на основе операций *TTend* – определение типа тенденции *RTend* – определение интенсивности тенденции специальной лингвистической шкалы, построенной для исходного ВР. В результате описанных преобразований получается несколько ВР, четких числовых и нечетких лингвистических, например ВР параметров нечетких тенденций [Ярушкина и др., 2010а].

Данные ряды являются потенциально информативными для построения моделей гипотез прогноза ВР четких x_i и нечетких \tilde{x}_i уровней, типов $\tilde{\nu}_i$ и интенсивностей \tilde{a}_i нечетких тенденций [Ярушкина и др., 2010b], [Ярушкина и др., 2007], [Ярушкина и др., 2010c].

Алгоритм прогноза ВР $Y = \{t_i, x_i\} (i = 1, 2, \dots, n)$ будем рассматривать при условии существования правдоподобности экспертной гипотезы, что нечеткая тенденция (НТ) ВР $Z = \{t_i, z_i\} (i = 1, 2, \dots, k)$ является предиктором временного ряда Y . В основу моделирования поведения ВР в терминах элементарной и локальной НТ положим предположение о развитии системы как результата зависимости НТ от значений НТ в предыдущие моменты времени. Таким образом, структуру модели ВР будем выражать нечетким разностным

уравнением d -го порядка: $\tau_i = \tilde{f}(t, \tau_{i-1}, \dots, \tau_{i-d})$, где d – фиксированное число, параметр модели; $\tau_i, \tau_{i-1}, \dots, \tau_{i-d}$ – последовательность НТ; \tilde{f} – некоторая нечеткая зависимость.

4. Метод и алгоритм прогнозирования ВР на основе НТ

Для прогнозирования числовых значений и НТ состояния организации в данном продукте используем метод моделирования НЭТ, алгоритм которого заключается в следующем:

- Построение F-компонент исследуемых ВР.
- Составление уравнения регрессии Y по Z . Вычисление корреляции.
- На основе полученных значений коэффициента корреляции используются следующие гипотезы прогнозирования:

1. *Гипотеза сохранения тенденции.* Рассматриваем тенденцию за предыдущий месяц. Прогнозируемое значение вычисляется по формуле: $Y(t+1) = Y(t) + Tp$,

где $Y(t+1)$ – прогноз на следующий период времени;

$Y(t)$ – реальное значение в момент времени t ;

Tp – значение тенденции за предыдущий период времени.

2. *Гипотеза устойчивости тенденции.* Используем для прогнозирования скользящее среднее $Y(t+1) = (1/(T+1)) * [Y(t) + Y(t-1) + \dots + Y(t-T)]$,

где T – количество значений по времени.

Рассматриваем тенденции за предыдущий выбранный период, выбираем преобладающий кластер тенденции и вычисляем прогноз по приведенной выше формуле, строим тенденцию. При одинаковых количествах вхождений тенденций используется оптимистичный прогноз, то есть выбирается наибольшее среднее значение тенденций.

Для нечеткой кластеризации тенденций используем FCM-алгоритм, основанный на минимизации целевой функции:

$$J = \sum_{j=1}^C \sum_{i=1}^N \mu_{ij}^m \|x_i - c_j\|,$$

где N – количество показателей;

C – количество кластеров;

μ_{ij} – степень принадлежности объекта i кластеру j ;
 m – любое действительное число, большее 1;
 x_i – i -й объект набора объектов;
 c_j – j -й кластер набора кластеров;
 $\|x_i - c_j\|$ – норма, характеризующая расстояние от центра кластера j до объекта i .

Объектами кластеризации является множество элементов ВР $A = \{a_1, \dots, a_n\}$. Задачей FCM-алгоритма является разбиение этого набора на заданное количество кластеров. Вводится конечное множество признаков (тенденций) $P = \{p_1, \dots, p_q\}$, каждый из которых количественно представляет некоторое свойство или характеристику элементов рассматриваемой проблемной области. При этом n есть общее количество объектов данных, а q – общее количество измеримых признаков.

Далее для каждого из объектов кластеризации должны быть измерены все признаки множества P в некоторой количественной шкале. Тем самым каждому из элементов $a_i \in A$ поставляется в соответствие некоторый вектор $x_i = (x_1^i, x_2^i, \dots, x_q^i)$, где x_j^i – количественное значение признака $p_j \in P$ для объекта $a_i \in A$. Все x_j^i принимают действительные значения. Эти количественные значения признаков тенденций и являются векторами признаков объектов кластеризации. Векторы значений признаков $x_i = (x_1^i, x_2^i, \dots, x_q^i)$ удобно представлять в виде матрицы данных D размерности $n \times q$, каждая строка которой равна значению вектора x_i .

3. Прогнозирование на заданный период на основе НЭТ. Алгоритм прогнозирования на период на основе тенденций работает следующим образом. Мы рассматриваем тенденции за предыдущий период за полгода и запоминаем набор тенденций A . Далее ищем такой же набор A тенденций во всех остальных предыдущих полугодиях. Найдя такой набор B , смотрим, какая тенденция C стоит после этого найденного набора B . Строим прогноз, равный тенденции C . Если не существует такого набора B , который совпадал бы с набором A , мы повторяем поиск набора, но уже ищем не полное его совпадение, а с отклонением в одну тенденцию. Так повторяем до тех пор, пока не найдем подходящий набор тенденций B .

Выбор наилучшей гипотезы осуществляется с учетом наименьшей средней относительной ошибки (MSD, RMSD).

Таблица 1

Оценки качества

Квадрат среднеквадратичного отклонения	$MSD = \frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2$
Среднее квадратичное отклонение	$RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - x'_i)^2}$

5. Формирование правил

В первоначальном варианте системы использовались предопределенные правила и выводы для одной предметной области: экономического экспресс-анализа предприятия по данным бухгалтерской отчетности. Однако такое «жесткое» ограничение не позволяет пользователям самостоятельно производить настройку на различные возникающие ситуации. Необходимо обеспечить их удобным инструментом, руководствуясь, как и прежде, принципом нетребовательности системы к высокому уровню знания математической статистики для пользователя. Описанные в предыдущих пунктах механизмы предлагается заложить в основу инструментария для генерации правил, которые впоследствии будут дополнительно охарактеризованные пользователем. Так, сформированные группы временных рядов предлагается использовать для генерации правил следующих двух типов:

1. При фиксированном уровне сглаживания и уровне корреляции определять, какие из временных рядов были сгруппированы. На основании такого распределения генерировать правила следующего вида: если временные ряды А и Б находятся в одной группе, а временной ряд В в другой (т.е. ряды А и Б имеют схожую динамику в отличие от В), то описать ситуацию как *характеристика_1* и выдать рекомендацию *рекомендация_1*, и т.д. Механизм формирования правил, таким образом, будет осуществлять срез распределения временных рядов по группам.

2. Другим вариантом формирования правил является настройка параметров формирования групп до возникновения нужного (заданного пользователем) распределения показателей. Правила в таком случае будут опираться на значения этих параметров и имеют

следующий вид: Если уровень сглаживания *низкий* и коэффициент корреляции *большой*, то временные ряды имеют *высокую* степень принадлежности к группам, показатели характеризуются как *характеристика_1*, выдать рекомендацию *рекомендация_1*, и т.д.

Еще одним механизмом для генерации правил выступают гипотезы прогнозирования тенденций. Формируются правила, характеризующие в зависимости от характера ряда, основную тенденцию, или наиболее часто возникающую ситуацию движения показателей.

Механизмы формирования правил позволят осуществить гибкий подход к анализу показателей, представленных временными рядами, пользователей. Производя такое формирование, пользователям в абстрактной форме будет предложено охарактеризовать возникающие распределения показателей. Сведения в дальнейшем лягут в основу базы знаний данной предметной области.

6. Структура Интернет-сервиса

Интернет-сервис представляет собой систему, основными компонентами которой являются:

1. Экспертная система, основанная на нечеткой логике, содержащая:

- a) систему нечеткого вывода.
- b) реляционный сервер данных.
- c) нейронную сеть для расчета прогноза остатков временных рядов.
- d) генетический алгоритм для настройки параметров F-преобразования.

2. Система прогнозирования временных рядов на основе интеграции метода F-преобразования и метода нечетких тенденций.

3. Система визуализации временных рядов и лингвистической интерпретации.

Одной из основных особенностей системы является то, что Интернет-сервис является расширяемой системой, способной к самоорганизации и наращиванию его функций. Компонент-ориентированная структура подходит для такой организации работы, при которой будут взаимодействовать несколько независимых компонент и впоследствии могут быть использованы по-разному в ходе процедуры экспертной диагностики.

Нечеткая экспертная система обладает следующими характеристиками: использует статистические данные, интерпретируя их для генерации правил; представляет знания в лингвистическом виде (или функциях принадлежности), нечетких правилах и обученной нейронной сети; работает как система взаимодействующих компонентов; обеспечивает самообучение системы за счет извлечения новых правил из результатов анализа.

СПИСОК ЛИТЕРАТУРЫ

[**Batyrshin et al., 2005**] Association networks in time series data mining / Batyrshin I., Herrera-Avelar R., Sheremetov L., Panova A. // NAFIPS 2005. Soft Computing for Real World applications, Ann Arbor. – Michigan, USA, June 22–25, 2005. – P. 754–759.

[**Namestnikov et al., 2002**] Namestnikov A., Yarushkina N. Efficiency of Genetic algorithms for automated design problems // Известия Российской академии наук. Теория и системы управления. 2002. № 2. С. 127–133.

[**Perfilieva, 2011**] I. Perfilieva, M. Dankova, and B. Bede. Towards a Higher Degree F-Transform // Fuzzy Sets and Systems. 2011. Vol. 180. Pp. 319.

[**Zadeh, 1979**] Zadeh L.A. Fuzzy Sets and Information Granularity // Advances in Fuzzy Sets Theory and Applications / Ed. by M.M.Gupta, R.K.Ragade, R.R.Yager. – Amsterdam : North-Holland Publishing Company, 1979. – pp.3–20.

[**Афанасьева и др., 2011**] Афанасьева Т.В., Ярушкина Н. Г. Нечеткий динамический процесс с нечеткими тенденциями в анализе временных рядов // Вестник Ростовского государственного университета путей сообщения. 2011. №3. С. 7–16.

[**Новак и др., 2008**] Интегральный метод принятия решений и анализа нечетких временных рядов / В. Новак [и др.] // Программные продукты и системы. 2008. № 4. С. 18.

[**Ярушкина и др., 2010а**] Ярушкина Н.Г., Афанасьева Т.В., Перфильева И.Г. Интеллектуальный анализ временных рядов: учебное пособие. – Ульяновск: УлГТУ. 2010. – 320 с.

[**Ярушкина и др., 2010б**] Ярушкина Н.Г., Афанасьева Т.В. Метод нечеткого моделирования и анализа тенденций временных рядов // Интеллектуальные системы управления. Коллективная монография / под ред. акад. РАН С.Н. Васильева. – М.: Машиностроение, 2010. С. 301–305.

[**Ярушкина и др., 2007**] Ярушкина Н.Г., Афанасьева Т.В. Нечеткие временные ряды как инструмент для оценки и измерения динамики процессов // Датчики и системы. 2007. Т. 12. С. 46–50.

[**Ярушкина и др., 2010с**] Ярушкина Н.Г., Перфильева И.Г., Афанасьева Т.В. Интегральный метод нечеткого моделирования и анализа нечетких тенденций // Автоматизация процессов управления. 2010. № 2 (20). С. 59–63.

ОБУЧЕНИЕ МАШИН ОПОРНЫХ ВЕКТОРОВ ДЛЯ РЕШЕНИЯ ЗАДАЧ КАТЕГОРИЗАЦИИ ТЕКСТА¹

Ш.А. Ахмедова (*shahnaz@inbox.ru*),

Е.С. Семенкин (*eugeneseimenkin@yandex.ru*)

Сибирский государственный аэрокосмический университет
им. М. Ф. Решетнева, Красноярск

Т.О. Гасанова (*tatiana.gasanova@uni-ulm.de*)

В. Минкер (*wolfgang.minker@uni-ulm.de*)

Ульмский университет, Ульм

В работе описываются новый алгоритм, основанный на кооперации методов роевого интеллекта, и его модификация, предназначенные для решения задач безусловной и условной оптимизации. Данные эвристики были применены для обучения машин опорных векторов, которыми решались задачи категоризации текста. По полученным результатам разработанные алгоритмы превзошли большинство альтернативных методов, тем самым была доказана их работоспособность и эффективность.

Введение

Категоризация текста – это процесс классификации текстовых документов при наличии фиксированного количества заранее определенных категорий или классов [Joachims, 1998]. Существует множество различных приложений категоризации текста, например, идентификация жанра произведений, определение типа документов, фильтрация спама и так далее.

В настоящее время разработаны различные методы для классификации текстов (например, статистические методы). Одним из наиболее известных и часто используемых подходов, предназначенных для решения задач классификации, является метод опорных векторов (Support Vector Machine, SVM) [Вапник и др., 1974]. Метод опорных векторов – алгоритм типа «обучения с учителем», изначально разработанный для решения задач бинарной категоризации. Идея алгоритма заключается в том, что каждый объект из

¹ Работа выполнена при поддержке Министерства образования и науки РФ, проект 140/14.

набора данных представляется в виде вектора в пространстве, далее полученные векторы необходимо разделить некоторой гиперплоскостью таким образом, чтобы объекты разных классов были как можно дальше друг от друга. То есть сначала алгоритм обучается на некотором наборе данных, а затем категории новых объектов определяются проверкой, по какую сторону построенной гиперплоскости оказались соответствующие этим объектам векторы в пространстве.

Метод опорных векторов входит в группу линейных классификаторов, то есть объекты разных классов линейно разделяются. Но зачастую такое разделение данных невозможно. Поэтому в 1992 году Б. Босер, И. Гийон и В. Вапник предложили способ категоризации данных методом опорных векторов с помощью функций-ядер [Boser et al., 1992].

В данной работе описаны классификаторы, полученные обучением метода опорных векторов с полиномиальным ядром двумя коллективными бионическими алгоритмами, основанными на кооперации известных эвристик стайного типа. Первый алгоритм был разработан для решения задач вещественной безусловной оптимизации, им настраивались параметры ядра, второй же алгоритм является модификацией первого, но для решения вещественных задач условной оптимизации, и он использовался уже непосредственно для обучения SVM-классификаторов с разными ядрами. Далее в работе описано, каким образом обрабатывались данные перед работой классификаторов [Gasanova et al., 2013], а также сформулированы решенные задачи и приведены полученные результаты.

1. Разработанный метод классификации

1.1. Метод опорных векторов

Пусть дан набор данных:

$$X^l = \{ (x_1, y_1), \dots, (x_l, y_l) \}, x_i \in R^m, y_i \in \{-1; 1\},$$

где l – это количество объектов в обучающей выборке с m вещественными атрибутами (таким образом, каждый объект в выборке можно представить как вектор размерности m), y_i – класс i -го объекта из выборки.

В самом общем виде метод опорных векторов [Вапник и др., 1974] заключается в том, что необходимо построить гиперплоскость:

$$\langle w, x \rangle + b = 0,$$

где $w \in R^m, b \in R, \langle _, _ \rangle$ – скалярное произведение, которое разделяло бы объекты класса -1 от объектов класса 1 в пространстве. Тогда классификация нового объекта осуществляется по следующему правилу:

$$f(x) = \begin{cases} 1, & \langle w, x \rangle + b \geq 1 \\ -1, & \langle w, x \rangle + b \leq -1 \end{cases}$$

Помимо прочего, в методе опорных векторов предполагается, что расстояние между построенной гиперплоскостью и ближайшими к ней классифицируемыми объектами должно быть максимально возможным. Гиперплоскость с упомянутым выше свойством может быть получена при решении следующей задачи условной оптимизации:

$$\|w\|^2 \rightarrow \min \\ y_i (\langle w, x_i \rangle + b) \geq 1, i = \overline{1, l}.$$

Выше был описан случай линейной разделимости данных различных классов. Однако в большинстве случаев такое разделение невозможно, и применение метода опорных векторов для линейно неразделимых наборов данных приводит к неудовлетворительным результатам. Одним из решений данной проблемы является перенос данных векторов в пространство большей размерности, и уже для данных в новом пространстве применяется метод опорных векторов. То есть основная идея этого подхода заключается в том, что исходные данные в пространстве большей размерности будут линейно разделимы, поэтому и применение SVM-классификаторов целесообразно. Проще всего реализовать данную идею, применив функцию-ядро [Boser et al., 1992].

Вообще говоря, существует множество различных функций-ядер, но для данной работы была использована полиномиальная функция $K(x, x') = (\alpha \langle x, x' \rangle + \beta)^d$, где α, β, d – параметры функции-ядра $K(x, x')$. Далее было изменено правило классификации:

$$f(x) = \begin{cases} 1, & ((K(w, x) + \beta)^d + b) \geq 1 \\ -1, & ((K(w, x) + \beta)^d + b) \leq -1 \end{cases}$$

Тогда для построения гиперплоскости, наилучшим способом разделяющей объекты разных классов, решалась следующая задача условной оптимизации:

$$\|w\|^2 \rightarrow \min,$$

$$y_i ((\alpha < w, x_i > + \beta)^d + b) \geq 1, i = \overline{1, l}.$$

Таким образом, для решения задачи классификации методом опорных векторов необходимо определить параметры функции-ядра α, β, d , вектор w и параметр смещения b , иначе говоря, для каждого SVM-классификатора со своими параметрами функции-ядра α, β, d решить задачу условной оптимизации для построения разделяющей гиперплоскости, а затем уже из получившихся классификаторов выбрать лучший, то есть с минимальной ошибкой классификации.

1.2. Кооперации эвристик стайного типа

В данной работе для настройки параметров функции-ядра SVM-классификаторов был использован коллективный бионический алгоритм, описанный в [Akhmedova et al., 2013], а его модификация для решения задач условной оптимизации применялась для нахождения вектора w и параметра смещения b .

В настоящее время разработано множество различных эвристик, которые успешно справляются с решением весьма сложных оптимизационных задач. Авторами данной работы ранее [Akhmedova et al., 2013] уже были исследованы пять известных алгоритмов стайного типа: Particle Swarm Optimization (PSO) [Kennedy et al., 1995], Wolf Pack Search (WPS) [Yang, 2007], Firefly Algorithm (FFA) [Yang, 2008], Cuckoo Search Algorithm (CSA) [Yang et al., 2009], Bat Algorithm (BA) [Yang, 2010]. Перечисленные эвристики изначально были разработаны для решения задач вещественной безусловной оптимизации. Идея каждого упомянутого выше алгоритма заключается в имитации некоторого природного процесса или поведения определенных видов животных. Проведенные исследования показали, что невозможно определить заранее, какой алгоритм лучше использовать для решения той или иной задачи. В этой связи был предложен метод, который бы сочетал в себе достоинства перечисленных ранее алгоритмов. Предложенная метаэвристика была названа Co-Operation of Biology Related Algorithms (COBRA) [Akhmedova et al., 2013]. Главная идея разработанного подхода заключается в генерировании пяти популяций (одна популяция для каждого алгоритма, то есть одна популяция для PSO, другая для WPS и т.д.), которые затем параллельно работают, обмениваясь между собой индивидами.

Также был автоматизирован процесс настройки размера популяций, поэтому не было необходимости подбирать его для каждого алгоритма-компонента. Размер каждой популяции может как увеличиваться, так и уменьшаться, в зависимости от того, улучшается ли значение целевой функции или нет за данное число итераций. То есть, если за определенное заранее количество итераций значение целевой функции не улучшается, то увеличивается размер всех популяций, и наоборот. Кроме того, каждая популяция может «пополниться», если к ней добавить индивидов из других популяций (соответственно в этих самых «других популяциях» данных индивидов уже не будет). Такое «пополнение» популяции возможно только, если ее среднее значение целевой функции лучше среднего значения целевой функции любой другой популяции. Таким образом, на каждой итерации разработанной эвристики определяется «алгоритм-победитель». Результатом подобного «соревнования» между популяциями является то, что автоматически определяется, какой алгоритм следует применять для решения заданной задачи. Помимо изменения размера популяций в алгоритме также реализовано «общение» между популяциями, в ходе которого они обмениваются индивидами. Обмен происходит следующим образом: худшие индивиды одной популяции замещаются лучшими индивидами других популяций.

Далее была разработана модификация алгоритма COBRA для решения задач условной оптимизации. Модификация предложенной метаэвристики заключалась в учете ограничений следующими способами: методом динамических штрафов [Eiben et al., 2003], с помощью правил Дэба [Deb, 2000], а также методом, описанным в статье [Liang et al., 2010]. Метод, описанный в работе [Liang et al., 2010], был использован для учета ограничений алгоритмом PSO, в то время как для остальных алгоритмов-компонент метаэвристики COBRA были сначала применены правила Дэба, а затем метод динамических штрафов.

COBRA и ее модификация для решения задач условной оптимизации были протестированы на репрезентативном множестве задач, взятых с конкурсов CEC'2013 и CEC'2009 соответственно. Полученные результаты подтвердили эффективность разработанных алгоритмов, а также целесообразность их применения для решения

оптимизационных задач. Они также показали лучшие результаты, чем все пять их алгоритмов-компонент по отдельности.

Итак, после проведенных исследований было принято решение применить разработанные методы для обучения SVM-классификаторов. COBRA использовалась для настройки параметров функции ядра α, β, d : каждый индивид (всех популяций) описывал набор параметров функции-ядра. Далее для каждого такого индивида с помощью модификации алгоритма COBRA для решения задач условной оптимизации осуществлялся поиск вектора w и параметра смещения b . В конце выбирался индивид с наилучшим показателем эффективности классификации (то есть с минимальной ошибкой классификации).

2. Предобработка данных

В статье [Gasanova et al., 2013] описан новый способ обработки текстовых данных, главная идея которого заключается в том, что каждое слово, которое присутствует в тексте, должно как-то численно соотноситься с некоторым определенным классом. Таким образом, каждому слову ставится в соответствие вещественное число, назовем его C_j для j -го слова, которое зависит от частоты появления этого слова в тексте. Данное число определяется с помощью модифицированной формулы, которая используется для нечетких классификаторов. Функция принадлежности заменяется частотой «встречаемости» слова в текстах определенного класса.

Пусть L – число классов для решаемой задачи; n_i – число объектов из набора данных i -го класса; N_{ji} – количество «появлений» j -го слова в текстах i -го класса; $T_{ji} = \frac{N_{ji}}{n_i}$ – частота «появления» j -го слова в текстах i -го класса; $R_j = \max_i T_{ji}$, $S_j = \arg(\max_i T_{ji})$ – номер класса, который присваивается j -му слову. Тогда C_j вычисляется следующим образом:

$$C_j = \frac{1}{\sum_{i=1}^L T_{ji}} \left(R_j - \frac{1}{L-1} \sum_{\substack{i=1 \\ i \neq S_j}}^L T_{ji} \right).$$

Таким образом, каждый объект-текст из набора данных представляется в виде вектора из $L+1$ чисел, где первое число – идентификатор класса, а остальные являются суммами значений C_j всех слов, которые встречались в данном объекте-тексте.

3. Решение задач категоризации текста

Для исследования эффективности предложенного метода и сравнения его с другими алгоритмами категоризации текста были решены задачи с конкурса DEFT07 («Defi Fouille de Texte») [DEFT'07, 2007]. В конкурсе были представлены три задачи: «A voir a lire» – категоризация отзывов на книги, «Video Games» – категоризация отзывов на видеоигры, «Debates in Parliament» – категоризация отзывов на принятие или отклонение выдвигаемого закона.

Эффективность классификации оценивалась с помощью вычисления параметра *F-score* при $\beta = 1$:

$$F - score = \frac{(\beta^2 + 1) * precision * recall}{\beta^2 * (precision + recall)},$$

$$precision = \frac{|\{relevant_documents\} \cap \{retrieved_documents\}|}{|\{retrieved_documents\}|},$$

$$recall = \frac{|\{relevant_documents\} \cap \{retrieved_documents\}|}{|\{relevant_documents\}|}.$$

Точность классификации «*precision*» для каждого класса вычислялась как число правильно классифицированных объектов из набора данных для данного класса, разделенное на число объектов, которые алгоритм отнес к данному классу. Параметр «*recall*» – это число правильно классифицированных объектов из набора данных, поделенное на число объектов, которое должно было быть отнесено к данному классу.

Таблица 1

Задачи с конкурса DEFT07

Задача	Описание задачи	Итог классификации
«A voir a lire»	3000 отзывов на книги	0:отрицательный отзыв, 1:нейтральный отзыв, 2:положительный отзыв
Video Games	4000 отзывов на видеоигры	0:отрицательный отзыв, 1:нейтральный отзыв, 2:положительный отзыв
Debates in Parliament	28800 голосований за выдвигаемые законы в парламенте Франции	0:против выдвинутого закона, 1:за выдвинутый закон

Таблица 2

Результаты, полученные для задач с конкурса DEFT07

Исследовательская группа	A voir à lire (rank1)	Video Games (rank2)	Debates (rank3)	Rank
J.-M. Torres-Moreno	0.603 (2)	0.784 (1)	0.720 (1)	1
G. Denhiere	0.599 (3)	0.699 (5)	0.681 (5)	3
S. Maurel	0.519 (7)	0.706 (4)	0.697 (3)	4
M. Vernier	0.577 (4)	0.761 (3)	0.673 (6)	5
E. Crestan	0.529 (6)	0.673 (7)	0.703 (2)	6
M. Plantie	0.472 (9)	0.783 (2)	0.671 (8)	7
A.-P. Trinh	0.542 (5)	0.659 (8)	0.676 (7)	8
M. Genereux	0.464 (10)	0.626 (9)	0.569 (11)	10
E. Charton	0.504 (8)	0.619 (10)	0.616 (9)	9
A. Acosta	0.392 (11)	0.536 (11)	0.582 (10)	11
COBRA-SVM	0.619 (1)	0.696 (6)	0.692 (4)	2

В таблице 2 представлены полученные различными методами (включая описанный в данной работе метод COBRA-SVM) результаты для всех трех задач. Причем словом «rank» обозначено, под каким номером в списке находится полученный результат, то есть (1) означает лучший результат для соответствующей задачи, (2) – второй, и т.д. до (11), что означает худший полученный результат для этой задачи. Таким образом, все результаты были ранжированы, а затем вычислялось целое значение от «Rank» следующим образом:

$$Rank = \frac{rank1 + rank2 + rank3}{3}.$$

Заключение

В данной работе описан новый подход к обучению SVM-классификаторов, которыми были решены три задачи с конкурса DEFT07. Настроенные классификаторы демонстрирует меньшую ошибку классификации, судя по значениям, приведенным в таблице 2, чем большинство других известных методов. Полученные резуль-

таты показали эффективность предложенного подхода и целесообразность его применения для решения задач категоризации текста.

СПИСОК ЛИТЕРАТУРЫ

- [**Вапник и др., 1974**] Вапник В.Н., Червоненкис А.Я. Теория распознавания образов. М.: Наука, 1974.
- [**Akhmedova et al., 2013**] Akhmedova Sh., Semenkin E. Co-Operation of Biology Related Algorithms // Proc. of the IEEE Congress on Evolutionary Computation (CEC 2013), Cancún (México). June 20–23, 2013.
- [**Boser et al., 1992**] Boser B., Guyon I., Vapnik V. A training algorithm for optimal margin classifiers // In D. Haussler, editor, proc. of the 5th Annual ACM Workshop on COLT, Pittsburgh. 1992.
- [**Deb, 2000**] Deb K. An efficient constraint handling method for genetic algorithms. // Computer methods in applied mechanics and engineering. 2000. 186(2-4).
- [**Eiben et al., 2003**] Eiben A.E., Smith J.E. Introduction to evolutionary computation. // Berlin: Springer, 2003.
- [**Gasanova et al., 2013**] Gasanova T., Sergienko R., Minker W., Semenkin E., Zhukov E. A Semi-supervised Approach for Natural Language Call Routing // Proc. of the SIGDIAL 2013 Conference. August, 2013.
- [**Joachims, 1998**] Joachims T. Text categorization with Support Vector Machines: Learning with many relevant features // Proc. of the Machine Learning: ECML-98, Tenth European Conference on Machine Learning. 1998.
- [**Kennedy et al., 1995**] Kennedy J., Eberhart R., Particle Swarm Optimization // Proc. of IEEE International Conference on Neural Networks. 1995.
- [**Liang et al., 2010**] Liang J.J., Shang Z., Li Z. Coevolutionary Comprehensive Learning Particle Swarm Optimizer // Proceedings of the Congress on Evolutionary Computation (CEC2010). 2010.
- [**Yang, 2007**] Yang Ch. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search // Proc. of the International Conference on Intelligent Pervasive Computing. 2007.
- [**Yang, 2008**] Yang X.S. Nature-Inspired Metaheuristic Algorithms // Frome: Luniver Press, 2008.
- [**Yang, 2010**] Yang X.S. A New Metaheuristic Bat-Inspired Algorithm. In J. R. Gonzalez, editor, proc. of Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). 2010.
- [**Yang et al., 2009**] Yang X.S., Deb S. Cuckoo search via Levy flights // Proc. of the World Congress on Nature & Biologically Inspired Computing. 2009.
- [**DEFT'07, 2007**] DEFT'07, Plate-forme AFIA 2007. – <http://deft07.limsi.fr/actes.php>.

ПРОБЛЕМЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ ¹

В.И. Городецкий (*gor@iias.spb.su*)
СПИИ РАН, Санкт-Петербург

Мир ИТ вступает в эру Больших Данных, осознавая их огромную ценность и отвечая на экспоненциальный рост данных. Однако в литературе обсуждаются в основном их новые возможности и те новые проблемы, которые не позволяют их реализовать на практике. В работе обсуждаются причины, тормозящие разработки в этой области, и приводятся результаты, направленные на их преодоление.

Введение

Под термином *Большие Данные* (БД) понимают широкий круг проблем обработки и анализа данных, ключевыми чертами которых являются множественность распределенных источников, гетерогенный характер, огромное число атрибутов (тысячи и даже миллионы), сверхбольшой объем данных и высокая скорость их нарастания во времени.

Хотя интеллектуальный анализ данных имеет полувековую историю, понимание специфики БД, а также их огромной научной и коммерческой ценности пришло совсем недавно. В работе кратко анализируется реальное состояние методов и средств анализа БД, описываются теоретические проблемы, тормозящие прогресс в этой области, а также намечаются возможные пути их преодоления. Приводятся типовые примеры БД и задач, решаемых на их основе (разд. 1), кратко описывается уровень современных достижений в области БД на примерах приложений (раз. 2) и анализируются новые проблемы вычислительного характера, свойственные обработке БД. В разд. 4 намечаются пути решения некоторых из проблем. В заключении резюмируются результаты работы.

¹ Работа выполнена при поддержке ОНИТ РАН (проект 1.12 программы «Системы автоматизации, обработки информации и поддержки принятия решений»).

1. Большие данные: примеры и практические достижения

Многообразие данных и разнообразие задач, которые решаются с использованием знаний, извлекаемых из БД, трудно даже очертить, можно лишь выделить несколько типовых примеров. Наиболее распространенными являются задачи типа (поддержки) принятия решений в реальном времени по данным из многих источников гетерогенных данных. Типичными здесь являются задачи обеспечения ситуационной осведомленности при управлении боевыми действиями, в чрезвычайных ситуациях, при оценке и прогнозировании террористических угроз на основе мониторинга тысяч сайтов т.п. Другой пример – это создание персонализированных кросс-доменных контекстно-зависимых рекомендующих систем, в которых обучающая информация черпается из всех источников, где так или иначе отражены «следы» пользователя.

Второй тип задач – это онлайн анализ многомерных временных рядов. Ярким примером здесь является одна из задач, выставленных на конкурс компанией NineSigma¹. В ней рассматривается непрерывное производство, мониторинг которого выполняют 4 млн. сенсоров. Они выдают данные о состоянии оборудования и технологических процессов каждые 10 сек. История таких измерений за 7–8 лет работы содержит порядка 30 млн. записей по 4 млн. чисел в каждой. Задача состоит в том, чтобы по результатам анализа прошлых процессов найти типовые прецеденты их аномального поведения, чтобы затем использовать их для выявления или предсказания аномалий в реальном времени с целью диагностики и восстановления нормальной работы. Самой сложной задачей здесь является поиск аналогов в многомерных временных рядах.

Еще один класс задач инициирован сетевым маркетингом (viral marketing). Он основан на том факте, что люди гораздо больше верят информации, полученной от друзей, чем от анонимных источников типа рекламных рассылок. Отсюда возникла идея использовать для рекламы целевые группы социальных сетей, в которых скорость распространения информации зависит от того, в какие именно узлы сети она введена. Поиск таких узлов социальной сети есть задача обработки БД. Она стала прототипом ряда задач

¹ NineSigma REQUEST #69987, http://www.ninesights.com/servlet/JiveServlet/downloadBody/8380-102-3-12246/NineSigma_REQ_69987.pdf.

управления потоками данных по сети, например, поиска стратегии распространения сетевых вирусов и др. Появились и двойственные задачи, например, поиск стратегии защиты сети от вирусов, размещение пунктов вакцинации при эпидемиях и пр.

Приведенные примеры задач обработки БД показывают, насколько разнообразны и сложны их постановки, насколько ценными могут быть результаты и насколько сложной является задача эффективной обработки БД. Однако анализ соотношения уровня ожиданий и уровня достижений в этой области говорит пока о наличии большой дистанции между ними.

2. Обработка БД: современный уровень

Основная цель анализа БД, как она формулируется экспертами, это предсказание будущих наблюдений. Очевидно, что это далеко не все, что можно получить в результате анализа. Но сейчас и эта задача решается далеко не лучшим образом. Опишем кратко возможности современных методов и средств анализа БД. Они определяются возможностями современных компьютерных инфраструктур для параллельной обработки данных, а также возможностями программных средств. Последние включают в себя программную платформу, управляющую декомпозицией задачи, параллельной обработкой подзадач и «сборкой» результатов, а также средства анализа данных и визуализации результатов.

В области компьютерных инфраструктур для параллельной обработки БД бесспорными лидерами являются средства компании IBM. В области программных инфраструктур наибольшей популярностью пользуется платформа *Hadoop*. Она же, по оценкам экспертов, обладает и наиболее передовыми возможностями, и наибольшим числом удачных результатов.

Apache Hadoop – это программная *Java*-платформа для создания и исполнения параллельных программ и управления БД. Она строится на основе программной модели *MapReduce*, использует библиотеку программ с открытым кодом и собственную файловую систему *Hadoop Distributed File System (HDFS)*. Последняя используется для быстрого доступа к данным большого объема за счет разделения данных на части и хранения их на разных компьютерах с избыточностью, обеспечивая тем самым их сохранность при сбоях и доступность для параллельной обработки. *MapReduce* – это

программная модель параллельной обработки БД. Она реализует традиционный подход – разделение данных, параллельную их обработку на разных процессорах и объединение результатов. Важно, однако, подчеркнуть, что платформы типа *Apache Hadoop* являются инструментами для считывания БД с разнообразной структурой компонент и их подготовки для анализа, но не для самого анализа.

Основными разработчиками программных средств анализа БД традиционно являются такие компании, как IBM, Oracle и другие, стоящие с ними в одном ряду. Например, наиболее распространенные средства IBM *InfoSphere* и *BigInsights* предоставляют аналитические возможности по работе с неструктурированными и структурированными данными в их исходном формате. *BigSheets* и *InfoSphere Streams* – это другие средства IBM для анализа БД с акцентом на различные функции администрирования, повышения производительности, безопасности. Все они реализуются на платформе *Apache Hadoop* и используют аналитические средства IBM, включая сложную текстовую аналитику. Более подробную информацию об этих и других средствах, которые в настоящее время рекламируются как средства поддержки аналитики БД, можно найти, например, в ([InfoSphere], [Streams], [IBM Big Data]).

К настоящему времени имеется ряд примеров успешных применений средств анализа БД для решения практически важных крупномасштабных задач. В основном они построены на базе компьютерных средств IBM, платформы *Apache Hadoop*, а также перечисленных выше программных систем. Одно из таких приложений – управление воздействием среды на реки на основе анализа потоковой информации. Оно выполняет анализ комплексного воздействия человека и среды его обитания на реки. В ней анализируются потоки данных биологического, физического, химического характера, собранные в районе рек штата Нью-Йорк, США. Информация собирается сенсорами, роботами и другими мобильными средствами. Массив собираемых данных используется для пространственного мониторинга вариаций данных среды (температуры, давления, минерализации и мутности воды, и других свойств воды). Эти данные поступают и обрабатываются в реальном времени средством *InfoSphere®Streams*, которое обеспечивает сбор и анализ данных от тысяч источников, визуализацию движения химических компонент, выполняет мониторинг качества воды и защиты рыбы на путях ее миграции.

Среди других успешных приложений средств анализа БД можно отметить систему *онлайн анализа потоков данных для оценки дорожного трафика* (Швеция), систему *TerraEchos@IBM* для разведки и сенсорного наблюдения и защиты критических инфраструктур, обеспечения безопасности по периметру в районе границ объекта; систему *Adelos S4* для анализа структурированного аудиопотока данных, получаемых от акустических сенсоров, используемую в интересах ВМС США, и другие.

Однако все эти и большинство других программных средств анализа БД пока еще весьма ограничены по возможностям и, по сути, реализуют функции технологии OLAP-анализа данных, которая разработана еще в 1990-х годах. Этих возможностей для решения современных задач интеллектуального анализа БД явно недостаточно. Рассмотрим некоторые специфические проблемы, возникающие при обработке именно БД.

3. Проблемы, обусловленные особенностями больших данных

Современные алгоритмические и программные средства интеллектуального анализа БД, несмотря на активную рекламу их достижений, имеют весьма ограниченные возможности. Причина частично в том, что большинство традиционных статистических методов выявления связей между атрибутами данных для работы с БД не может быть использовано. Это обусловлено рядом свойств БД, среди которых самым существенным является большая размерность пространства атрибутов, которая может исчисляться десятками тысяч и более.

Ключевой задачей анализа данных является поиск зависимостей и других видов связей между атрибутами. Среди атрибутов данных особую роль играют так называемые *целевые переменные*. Они задают либо значения оптимизируемых показателей качества, либо являются метками классов, и тогда другие атрибуты рассматриваются как признаки. В общем случае основная задача анализа БД состоит в построении *моделей целевых переменных* в виде функций от других переменных, но самой трудной подзадачей при этом является выбор самих переменных модели.

Значения атрибутов, измеряемых сенсорами, всегда содержат ошибки. Если число таких атрибутов велико, то ошибки вычисле-

ний, как правило, нарастают и начинают доминировать над полезным сигналом. Этот эффект принято называть *накоплением ошибок* (*noise accumulation*) [Fan *et al.*, 2013]. Эти ошибки становятся катастрофическими, когда алгоритмы обработки включают в себя, например, обращение матриц ковариаций, поиск их собственных значений и т.п. При этом воздействие ошибок может оказаться столь специфичным, что известные приемы обеспечения вычислительной устойчивости типа регуляризации не помогут. Обратим внимание, что пользователь старается обычно для гарантии сохранить и использовать как можно больше атрибутов, тем самым способствуя усилению негативного влияния эффекта накопления ошибок.

В [Fan *et al.*, 2013] негативная роль этого эффекта наглядно демонстрируется на примере задачи классификации при большом числе атрибутов, используемых в модели классификации. В примере рассматривается два класса данных с нормальными распределениями $X_1, \dots, X_n \sim N_d(\mu_1, I_d)$ и $Y_1, \dots, Y_n \sim N_d(\mu_2, I_d)$, где I_d – единичная ковариационная матрица размером d , $n=100$ (объем выборки) и $d=1000$ (размерность пространства признаков). Значения всех компонент вектора μ_1 в первом классе полагаются равными нулю, а во втором классе первые 10 элементов вектора μ_2 полагаются равными 3, а остальные – 0. В используемой модели классификации во внимание принимаются m наилучших признаков пространства размерности 1000¹. Классификация выполняется по максимуму значения меры близости, значение которой для заданного входа вычисляется как сумма его проекций на первый и *второй собственные векторы главных компонент* первого и второго классов, вычисляемых по данным выборок классов.

В примере исследуется влияние накопления ошибок на качество линейного разделения данных в зависимости от m – числа учитываемых признаков. Расчеты выполнены для $m = 2, 40, 200$ и 1000 . В каждом таком эксперименте строятся проекции примеров множеств на двумерное пространство первых двух главных компонент. Итоговые графики представлены на рис. 1. Из рисунков видно, что при $m = 2$ примеры разных классов хорошо разделяются линейной границей, хуже обстоит дело при $m=40$. При $m=200$ и

¹ В англоязычной литературе модели анализа данных, в которых большая часть атрибутов не принимается во внимание, принято называть термином *sparse model*, разреженная модель.

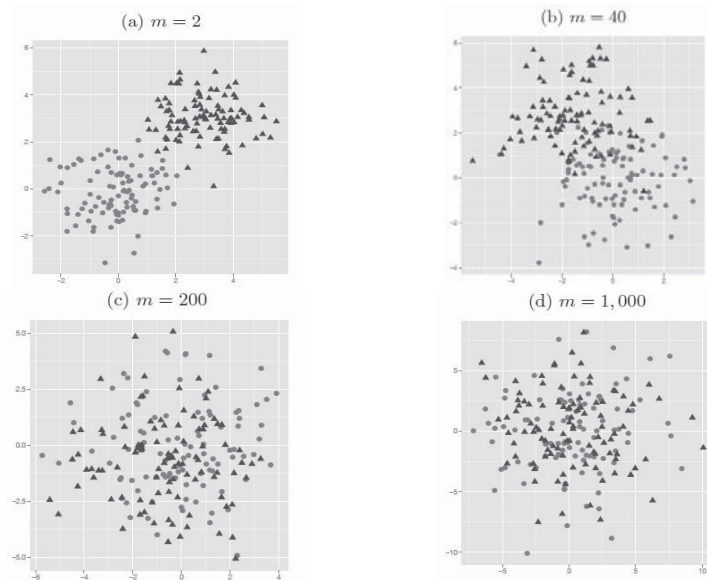
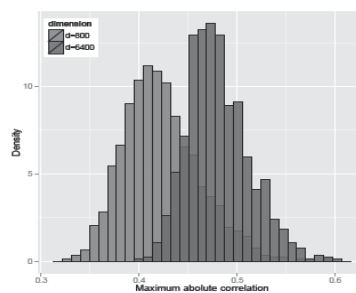


Рис. 1. Графики разброса проекций примеров данных (по 100 в классе) на первые две главные компоненты подпространств размерности m (график заимствован из работы [Fan et al., 2013])

$m=1000$ накопленная ошибка значительно больше полезного сигнала, и классы уже неразделимы. Это характерно для задач обработки БД. Поэтому в работах по анализу БД подчеркивается важность правильного выбора *размерности* пространства и состава атрибутов для построения моделей целевых атрибутов. Другой типичный эффект, возникающий при работе с БД, – это появление *ложных выборочных корреляций* (*spurious correlations*) между независимыми переменными. В [Fan et al., 2013] этот эффект иллюстрирует-

Рис. 2. Появление ложных связей: гистограмма максимума модуля выборочного коэффициента корреляции с другими атрибутами (график взят из [6])



ся результатами расчетов величины *выборочного* коэффициента корреляции для независимых компонент вектора $X=[x_1, \dots, x_d]^T$ с нормальным распределением. И хотя эти переменные независимы, при большой размерности вектора X выборочные корреляции его компонент оказываются значительными. На рис. 2 показана гистограмма для максимального значения выборочного коэффициента корреляции \hat{r} переменной x_1 со всеми другими, теоретически независимыми переменными. Оно построено для $n=60$ при размерностях d вектора X , равных 800 и 6400 при объеме эксперимента 1000 реализаций для каждого случая размерности. Анализ этих графиков показывает, что ложные корреляции действительно возникают, а их число и среднее значение коэффициента корреляции возрастают при росте размерности пространства. Этот эффект может привести к неправильному выбору переменных для модели целевой переменной.

Еще один специфический эффект, возникающий как следствие большой размерности данных, связан с появлением *статистической зависимости между помехой и переменными модели*¹. Обратим внимание на то, что в отличие от ложных корреляций здесь речь идет о реальных зависимостях между помехой и переменными модели. При этом чем больше размерность пространства, тем больше шансов появления такой зависимости. В БД эффект возникает по двум причинам [Fan *et al.*, 2013]:

1. При большом числе переменных высока вероятность появления атрибутов, реально коррелированных с помехой.
2. Данные могут быть получены из разных источников в разное время и измерены с разной точностью и смещением, что при их объединении приводит к появлению зависимостей между переменными и ошибками.

В настоящее время эта проблема пока изучена слабо, однако имеются работы, которые предлагают альтернативные методы статистической обработки БД, работающие при более слабых допущениях.

Существуют и другие проблемы, специфические для БД. Они значительно сужают возможности известных методов для анализа БД и требуют пересмотра методов решения задач статистического анализа.

¹ Этот эффект на английском языке называется *incidental endogeneity*.

4. Некоторые решения проблем анализа больших данных

4.1. Модель данных и роль онтологий

Одни те же БД могут использоваться в нескольких приложениях, которые могут сильно различаться между собой. Для каждого приложения нужно строить свою модель вычисления целевых переменных на разных множествах атрибутов. Решение таких задач облегчается, если использовать мета-модель данных, например, в форме онтологии. Онтология помогает преодолеть еще одну проблему работы с БД. Например, если данные хранятся в реляционных базах данных, то с ними работают с помощью SQL-запросов. Однако большинство пакетов анализа данных, включая пакеты OLAP-анализа, требует иного представления данных. Поэтому аналитик вынужден *экспортировать* данные из SQL-представления в иную форму, а затем результаты вновь *импортировать* в SQL-форму. Это громоздкие процедуры. *Онтологические модели* проще решают проблему несовместимости данных. В настоящее время ведется активная разработка методов и средств автоматизированного создания онтологий, например, с помощью викификации, онтологий повторного использования веба данных (*Linked Data web* [Bizer *et al.*, 2009]) и других технологий, которые значительно упрощают создание онтологий БД.

4.2. Снижение размерности

Типовые технологии работы с БД должны строиться на основе *базовой парадигмы* обработки: в них должны приниматься все меры для того, чтобы *снизить размерность данных*, вовлекаемых в обработку. Существующие методы снижения размерности можно разделить на две группы:

- методы, позволяющие вовлекать в обработку лишь часть примеров данных, сохраняя при этом их представительность и статистическую значимость оценок; они обычно реализуют случайный выбор примеров;

- методы снижения размерности пространства представления данных.

Методы второго типа разнообразны и более важны. Далее дается краткое описание некоторых методов, разработанных при участии автора. Они строятся так, чтобы не нужно было использовать вычислительно неустойчивые алгоритмы типа де-корреляции дан-

ных, перемножения большого числа малых величин, деления на малые величины и т.п.

4.3. Агрегирование атрибутов

Алгоритм выполняет преобразование примеров данных, представленных в терминах значений атрибутов, к множеству утверждений о свойствах подмножеств их значений. Эти утверждения должны задавать подмножества значений атрибутов с общим свойством. Примером является предикат $P_S^{(k)}(x_{i_1} \in \tilde{X}_{i_1}, \dots, x_{i_k} \in \tilde{X}_{i_k})$, который принимает значение «истина» в области, заданной выражением в скобках. В [9] они используются в качестве посылок формул вида:

$$P_S^{(k)}(x_{i_1} \in \tilde{X}_{i_1}, \dots, x_{i_k} \in \tilde{X}_{i_k}) \rightarrow \omega_k, \quad (1)$$

где S есть индекс предиката, k – индекс целевой переменной ω_k , ω_k в правой части утверждения. Переменная ω_k в этом примере может быть указанием на класс принимаемого решения, класс действия, которое должно выполняться системой, диагноз отказа и прочее. Частным случаем является одноместный предикат $P_S^{(k)}(x_{i_1} \in \tilde{X}_{i_1})$. Один из способов построения таких атрибутов предложен в [Gorodetsky *et al.*, 2012].

4.4. Причинный анализ агрегатов

Выражения (1) являются правилами ассоциативной классификации. Доказано [Alferis *et al.*, 2010], что среди них наибольший интерес представляют правила, в которых связь носит причинный характер. Однако, в общем случае, выделение причинных связей из множества ассоциативных является сложной задачей. Традиционные методы используют причинные Байесовские сети [Alferis *et al.*, 2010]. В них определение причинной связи дается в терминах Марковского покрытия узла этой сети, но его поиск является задачей экспоненциальной сложности. Поэтому построить причинную Байесовскую сеть для тысяч правил типа (1) (их может быть и намного больше) невозможно. Поэтому еще с конца 1990-х годов проводились исследования с целью поиска простых метрик для выделения причинных связей из ассоциативных ([Bgin *et al.*, 1997], [Li *et al.*, 2013] и др.). Эти модели названы *причинно-ассоциативными*. Одна из таких мер предложена в [Gorodetsky *et al.*, 2012]. Для пары предикатов A и B она вычисляется по формуле:

$$R(A,B) = p(B/A) - p(B/\bar{A}). \quad (2)$$

Величина $R(A, B)$ для пары случайных событий A и B известна в теории вероятностей как «коэффициент регрессии случайного события A на случайное событие B ». В формуле (2) первый член отвечает вероятности появления события B при условии, что появилось событие A , а второй член – вероятность того же события при условии, что событие A не появилось. Содержательно эта мера удовлетворяет формальным условиям: она некоммутативна, т.е. $R(A, B) \neq R(B, A)$, изменяется в интервале $[-1, 1]$, равна нулю для независимых событий и имеет максимальное по модулю значение при их функциональной зависимости. В обозначениях формулы (1) эта мера вычисляется так:

$$R(P_S^{(k)}, \omega_k) = \frac{p(P_S^{(k)}, \omega_k) - p(P_S^{(k)}) \times p(\omega_k)}{p(P_S^{(k)}) \times (1 - p(P_S^{(k)}))} = \text{для всех } S \text{ и } k. \quad (3)$$

Условие фильтрации агрегатов по мере причинной связи (2) с некоторой целевой переменной будет иметь вид:

$$|R(P_S^{(k)}, \omega_k)| \geq \delta^{(k)}_{\min} \text{ для всех } S \text{ и } k. \quad (4)$$

4.5. Минимизация модели целевой переменной

Специалисты в области интеллектуальной обработки (больших) данных не всегда обращают внимание на то, что послышки причинных связей типа (1) могут быть сильно зависимыми. Эта проблема известна в области коллективного распознавания как проблема разнообразия классификаторов [12], [13]. Учет подобных связей есть серьезный потенциальный источник снижения размерности задачи. В литературе рассматриваются различные эвристические подходы к ее решению. При участии автора разработан математически корректный алгоритм снижения размерности модели целевой переменной, который в настоящее время находится в стадии программной реализации и испытаний.

Заключение

В данной работе описана проблема обработки БД. Анализ состояния исследований и разработок в этой области показал, что в этой области следует гораздо больше говорить о новых проблемах, чем об эффективных решениях. Задачи, которые сейчас оказыва-

ется здесь возможным решать, пока не отвечают ожиданиям, которые стимулировали развитие этой области. В работе на основании анализа работ других авторов описаны причины такого положения дел, а также приведены некоторые результаты, которые могут обеспечить прогресс в аналитике БД.

СПИСОК ЛИТЕРАТУРЫ

[Городецкий и др., 2008] Городецкий В.И., Серебряков С.В. Методы и алгоритмы коллективного распознавания //Автоматика и телемеханика, № 11, 3–40, 2008.

[Aliferis et al., 2010] Aliferis C.F., Statnikov A., Tsamardinos I., Koutsoukos D. Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection. //Journal of Machine Learning Research #11 (2010) 171–234, 2010.

[Bizer et al., 2009] Bizer C., Heath T., and Berners-Lee T. Linked Data -The Story So Far. //International Journal on Semantic Web and Information Systems, 2009.

[Brin et al., 1997] Brin S., Motwani R., Silverstein C. Beyond market baskets: generalizing association rules to correlations.// Proceedings of the ACM SIGMOD International Conference on Management of Data. 1997, 255–264.

[Fan et al., 2013] Fan J. Han F., and Liu H. Challenges of Big Data. Analysis.// Princeton University, August 7, 2013. <http://arxiv.org/pdf/1308.1479.pdf>, 2013.

[Fan et al., 2012] Fan, J., Guo, S., and Hao, N. Variance estimation using re-fitted cross-validation in ultrahigh dimensional regression. //Journal of the Royal Statistical Society: Series B (Statistical Methodology), (2012), 74(1), 37–65, 2012.

[Gorodetsky et al., 2012] Gorodetsky, V., Samoylov, V., Serebryakov, S. Context – driven Data and Information Fusion. //Proceed. of IF Conference, Singapore, 2012.

[IBM Big Data] IBM Big Data Success Stories. <http://public.dhe.ibm.com/software/data/sw-library/big-data/ibm-big-data-success.pdf>.

[InfoSphere] InfoSphere BigInsights Enterprise Edition. <http://www-03.ibm.com/software/products/ru/infobigienteedit/>.

[Kucheva et al., 2003] Kucheva L., Whitaker C. Measures of diversity in Classifier Ensembles // Machine Learning, 2003. № 51, 181–207, 2003.

[Li et al., 2013] Li J., Le T.D., Liu L., Liu J., Jin Zh., and Sun B. Mining Causal Association Rules. //Intern. Workshop on Causal Discovery, Dallas, 2013.

[Streams] InfoSphere Streams Technical Overview–Use Cases Big Data. <http://www.slideshare.net/IBMInfoSphereUGFR/infosphere-streams-technical-overview-use-cases-big-data-jerome-chaillox>.

УДК 004.8

**THEORETICAL BASES OF THE METHODS OF
CONSTRUCTION OF ASSOCIATION MEASURES
IN INTELLIGENT DATA ANALYSIS**

I.Z. Batyrshin (*batyr1@gmail.com*)

Centro de Investigacion en Computacion, Instituto Politecnico Nacional,
Mexico

The general methods of construction of functions satisfying axioms generalizing the properties of correlation coefficient are proposed. These methods give possibility to build non-statistical measures of association on sets with similarity measure and involution operation. The correlation coefficient is obtained as a special case of association measures related with Lukasiewicz t-conorm.

Introduction

An analysis of associations is usually considered as one of the important tasks of data analysis. The Pearson's correlation coefficient [Wilcox, 2010] is the most popular measure of association in statistics and in statistical methods of decision making and pattern recognition. The possibility to measure both positive and negative associations between variables is considered as a basic feature of the correlation coefficient. The negative correlation between variables usually takes place when the positive deviations from the mean value of one variable correspond to the negative deviations from the mean value of another variable. An important task of time series data mining is a similarity search in time series data bases [Agrawal et al., 1993; Батыршин и др., 2007]. For example, one can look for companies with similar patterns of growth or stocks with similar movements in prices [Agrawal et al., 1993]. But it is very interesting also to look for competitive companies with inverse patterns of growth when the rising patterns of one sequence correspond to falling patterns of another one. Such negative associations can be found between prices and sell volumes, between wind velocity and the level of air contamination etc. In [Batyrshin et al., 2012], the axioms of association measures of time series have been introduced. The general methods of construction of association measures satisfying these axi-

oms have been proposed in [Batyrrshin, 2013a]. Generalizations of these methods on sets with involution operation and similarity measure have been proposed in [Batyrrshin, 2013b]. An important task of data mining is the search of association rules discovering frequent patterns in data but it is important also to use the measures that similarly to correlation coefficient can discover positive and negative associations in data [Han et al., 2006]. In [Murthy et al., 1985], it is considered the correlation measure of fuzzy sets different from Pearson's correlation coefficient but with the similar properties.

In this work a modification of the axioms of the association measures considered in [Batyrrshin, 2013a; Batyrrshin, 2013b] and a generalization of the methods of construction of association measures considered in these works are given. Section 1 presents the basic concepts of the theory of fuzzy sets and aggregation functions that will be used in the following sections. Sections 2 and 3 propose the definitions of association measures and methods of their construction. The association methods on the sets of n -tuples are considered. The final section contains conclusions.

1. Basic notations

Consider definitions of some concepts used in the following sections [Аверкин и др., 1988; Батыршин, 2001; Klement et al., 2000; Fodor et al., 1994; Beliakov et al, 2008; Grabisch et al., 2009].

The function $S:[0,1]^2 \rightarrow [0,1]$ is said to be a **t -conorm** if for all $a, b, c \in [0,1]$ the properties of commutativity, associativity, monotonicity are fulfilled together with the boundary condition:

$$S(a,0) = a.$$

From the definition of t -conorm it is fulfilled for all $a \in [0,1]$:

$$S(1,a) = S(a,1) = 1, \quad S(0,a) = a.$$

The element $a \in]0,1[$ is called a **nilpotent** element of S if it exists an element $b \in]0,1[$ such that $S(a,b) = 1$. t -conorm S has no nilpotent elements if and only if on $[0,1]$ from $S(a,b) = 1$ it follows $a = 1$ or $b = 1$.

Consider simplest t -conorms:

$$S_M(a,b) = \max\{a,b\}, \quad (1.1)$$

$$S_P(a,b) = a + b - ab, \quad (1.2)$$

$$S_L(a,b) = \min\{a+b, 1\}. \quad (1.3)$$

Maximum (1.1) and probabilistic sum (1.2) have no nilpotent elements but Lukasiewicz t -conorm (1.3) has.

With each t -conorm S two operations can be associated: the operations of **S-difference** $\stackrel{S}{\ominus}$ and **pseudo-difference** $(-)_S$ defined for all $a, b \in [0, 1]$ as follows:

$$a \stackrel{S}{\ominus} b = \inf\{c \in [0, 1] \mid S(b, c) \geq a\},$$

$$a(-)_S b = \text{sign}(a - b)(\max(a, b) \stackrel{S}{\ominus} \min(a, b)),$$

where $\text{sign}(a) = 1$, if $a > 0$, $\text{sign}(a) = -1$, if $a < 0$, $\text{sign}(a) = 0$, if $a = 0$.

The following pseudo-differences are associated with basic t -conorms S_M , S_P and S_L :

$$\begin{aligned} a(-)_M b &= \text{sign}(a - b)\max(a, b), \\ a(-)_P b &= (a - b)/(1 - \min(a, b)), \\ a(-)_L b &= a - b. \end{aligned} \tag{1.4}$$

An aggregation function of two arguments is a function $M: [0, 1]^2 \rightarrow [0, 1]$ that is non-decreasing in each argument and satisfying the properties:

$$\begin{aligned} M(0, 0) &= 0, \\ M(1, 1) &= 1. \end{aligned}$$

For example, any t -conorm is an aggregation function.

2. Measures of similarity and association

Consider a set X with an **involution** operation $N: X \rightarrow X$, that differs from the identity function, i.e. we consider the function satisfying on X the following conditions:

$$N(N(x)) = x.$$

and $N(x) \neq x$ for some x from X . $N(x)$ will be called a **reflection** of x . Denote FP a set of fixed points of the mapping N , i.e. elements of X such that $N(x) = x$. If FP is not empty denote its elements by x_{FP} . Denote $X_N = X \setminus FP$ a set of elements from X such that $N(x) \neq x$.

Examples:

1) Truth set $X = [0, 1]$, $N(x) = 1 - x$ for all x from X , $x_{FP} = 0.5$.

2) Ordinal scale $X = \{0, 1, \dots, K\}$, $N(j) = K - j$, for all j in X . FP is empty for odd K .

3) X is a fuzzy set $x: U \rightarrow [0, 1]$ with involutive complement $N(x(u)) = 1 - x(u)$ for all u from U . It has one fixed point x_{FP} and $x_{FP}(u) = 0.5$ for all u from U .

4) X is a set of real valued n -tuples $x = (x_1, \dots, x_n)$, $N(x) = -x = (-x_1, \dots, -x_n)$, $x_{FP} = (0, \dots, 0)$, where x can be a vector, time series or variable sample.

5) De Morgan lattice can have more than one fixed point.

Definition 1. Let X be a set with involution N and FP is a set of fixed points on N . A measure of association on $X_N = X \setminus FP$ is a function $A: X_N \times X_N \rightarrow [-1, 1]$ satisfying for all $x, y \in X_N$ the properties:

$$A(x, y) = A(y, x). \quad (2.1)$$

$$A(x, x) = 1. \quad (2.2)$$

$$A(N(x), y) = -A(x, y). \quad (2.3)$$

Note that for fixed points the conditions (2.2) and (2.3) are contradictory for this reason an association measure is defined on a set without fixed points. This definition can be extended on fixed points as follows.

Definition 2. A function $A: X \times X \rightarrow [-1, 1]$ on a set X with involution N will be referred to as:

– a **measure of association of type 1**, if (2.1) and (2.3) are fulfilled for all $x, y \in X$ and (2.2) is fulfilled for all $x \notin FP$;

– a **measure of association of type 2**, if (2.1) and (2.2) are fulfilled for all $x, y \in X$ and (2.3) is fulfilled for all $y \in X$ and all $x \notin FP$.

Further we will consider association measures on $X_N = X \setminus FP$, that can be extended in a suitable way on the set of all points from X .

Proposition 1. An association measure satisfies for all $x \in X_N$ the following properties:

$$A(N(x), x) = -1,$$

$$A(N(x), N(y)) = A(x, y),$$

$$A(N(x), y) = A(x, N(y)).$$

Definition 3. A similarity measure on X is a function $SIM: X \times X \rightarrow [0, 1]$ satisfying for all x, y from X the properties:

$$SIM(x, y) = SIM(y, x),$$

$$SIM(x, x) = 1.$$

A similarity measure SIM will be referred to as **strict** if it is fulfilled:

$$SIM(x, x) > SIM(x, y) \text{ for all } x \neq y.$$

Proposition 2. A similarity measure SIM satisfies on X the property:

$$SIM(N(x),N(y)) = SIM(x,y), \quad (2.4)$$

if and only if it satisfies on X the property:
 $SIM(N(x),y) = SIM(x,N(y))$.

These properties will be referred to as a **cancellation of reflections** (CR for short) and a **permutation of reflections** (PR for short), respectively. Note that in this paper CR property is considered instead of PR property used in [Batyrrshin, 2013a].

Theorem 1a). Suppose SIM is a similarity measure satisfying CR (2.4) and S is a t -conorm, then the function $A_{SIM,S}:X_N \times X_N \rightarrow [-1,1]$, defined for all $x,y \in X_N$ as follows:

$$A_{SIM,S}(x,x) = 1, \\ A_{SIM,S}(x,y) = SIM(x,y)(-)_S SIM(x,N(y)), \text{ if } x \neq y,$$

will be an association measure on X_N .

Theorem 1b). Suppose SIM is a strict similarity measure satisfying CR (2.4) and S is a t -conorm, then the function $A_{SIM,S}:X_N \times X_N \rightarrow [-1,1]$, defined for all $x,y \in X_N$ by:

$$A_{SIM,S}(x,y) = SIM(x,y)(-)_S SIM(x,N(y)),$$

will be an association measure on X_N if one of the following conditions is fulfilled:

- 1) $SIM(N(x),x) = 0$, for all $x \in X_N$ (2.5)
- 2) t -conorm S does not have nilpotent elements.

Proposition 3. Suppose M is an aggregation function of two arguments and SIM is a similarity measure, then the following function defined for all x,y from X by:

$$SIM_M(x,y) = M(SIM(x,y),SIM(N(x),N(y)))$$

will be an association measure satisfying CR (2.4). It will be strict if SIM is strict and M satisfies:

$$M(a,b) < 1 \text{ if } \min(a,b) < 1. \quad (2.6)$$

For example, in (2.6) one can use one of the following aggregation functions:

$$M(a,b) = \min(a,b),$$

$$M(a,b) = (a+b)/2.$$

Definition 4. Non-negative function $D:X \times X \rightarrow R$, satisfying for all x, y from X the properties:

$$D(x,y) = D(y,x),$$

$$D(x,x) = 0,$$

is referred to as a measure of difference. It is called strict if

$$D(x,y) > D(x,x), \text{ for all } x \neq y.$$

Proposition 4. Suppose D is a measure of dissimilarity and $V:R \rightarrow R$ is a strictly decreasing non-negative real function such that $V(0) = 1$, then the function

$$SIM_D(x,y) = V(D(x,y)),$$

will be a similarity measure satisfying CR (2.4), if D for all x,y from X satisfies the property of cancellation of reflections CR:

$$D(N(x),N(y)) = D(x,y). \quad (2.7)$$

SIM_D will be strict if D is strict.

Proposition 5. If it exists a constant H such that $D(x,y) \leq H$ for all $x,y \in X$ and $W:R \rightarrow R$ is a strictly increasing function such that $W(0) = 0$, $W(H) = 1$, then:

$$SIM_D(x,y) = 1 - W(D(x,y)),$$

is a similarity measure satisfying CR (2.4), if D satisfies this property.

SIM_D will be strict if D is strict. SIM_D satisfies (2.5), if

$$D(N(x),x) = H, \text{ for all } x \in X_N$$

3. Association measures of n -tuples

Suppose $X = R^n$ is a set of real n -tuples $x = (x_1, \dots, x_n)$, $n \geq 1$, and $N(x) = -x = (-x_1, \dots, -x_n)$, is a reflection N on X . For any n -tuples $x = (x_1, \dots, x_n)$, $y = (y_1, \dots, y_n)$ from X , $x \neq y$ denotes that $x_i \neq y_i$ for some i . Let $F:R^n \rightarrow R^n$ be a transformation of n -tuples such that from $x \neq y$ it follows $F(x) \neq F(y)$. Consider the following function defined by Minkowski distance with positive parameter r and transformation F [Batyrsin, 2013a]:

$$D_{r,F}(x,y) = \left(\sum_{i=1}^n |F(x)_i - F(y)_i|^r \right)^{1/r}$$

It is clear that D is a strict measure of difference.

Proposition 6. If for all $x \in X$ and all $i = 1, \dots, n$, a transformation F satisfies:

$$F(N(x))_i + F(x)_i = C,$$

for some constant C , then the dissimilarity measure $D(x, y) = D_{r, F}(x, y)$ satisfies CR (2.7).

D satisfies condition $D(N(x), x) = 2 \geq D(x, y)$ for all $x, y \in X$, if $C=0$ and for all $x \in X$ it is fulfilled the property of **r-normality**:

$$\sum_{i=1}^m |F(x)_i|^r = 1$$

In [Batyrrshin, 2013a] the different standardization methods F , satisfying Proposition 6 for $C = 0$ have been considered. Here we generalize this result on more general case of arbitrary constant. This property gives a possibility to use a standardization

$$F(x)_i = (x_i - \min(x)) / (\max(x) - \min(x)),$$

where $\min(x)$ and $\max(x)$ denote minimal and maximal elements of x correspondingly.

Considered here theoretical results give possibility to build association measures for n -tuples (for examples time series) as a result of the procedure containing the following steps:

- a) Definition of transformation F ;
- b) Selection of dissimilarity measure D ;
- c) Selection of the method of transformation of D in SIM ;
- d) Selection of the method of transformation of SIM in an association measure A .

For example, the sample Pearson's correlation coefficient

$$\text{corr}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \cdot \sum_{i=1}^n (y_i - \bar{y})^2}}$$

can be constructed as a result of the selection of the following components of this procedure:

$$F(x)_i = \frac{x_i - \bar{x}}{\sqrt{\sum_{j=1}^n (x_j - \bar{x})^2}}, \quad \bar{x} = \frac{1}{n} \sum_{j=1}^n x_j$$

D is a Euclidean distance, ($r = 2$),

$$\begin{aligned}SIM(x,y) &= 1 - D^2(x,y)/4, \\ A_{SIM}(x,y) &= SIM(x,y) - SIM(x,-y).\end{aligned}\tag{3.1}$$

This method of transformation of a similarity measure in association measure (3.1) is defined by pseudo-difference operation (1.4), associated with Lukasiewicz t-conorm. From this point of view one can consider the Pearson's correlation coefficient as a special case of the class of association measures related with Lukasiewicz t-conorm.

Conclusions

Based on a generalization of results of the previous works of the autor [Batyrrshin, 2013a; Batyrrshin, 2013b], the paper proposed axioms of association measures on sets with involution operation. The new methods of construction of association measures using similarity measures and data transformation are considered. The proposed methods can be used in construction of association measures of objects of different types in intelligent data analysis tasks, for example in time series data mining [Батыршин и др., 2007]. It would be interesting to extend the obtained results on the sets of multisets [Петровский, 2003] and on the sets of soft sets [Молодцов, 2004].

REFERENCES

- [Аверкин и др., 1988] Аверкин А.Н., Батыршин И.З., Блишун А.Ф., Силов В.Б., Тарасов В.Б. Нечеткие множества в моделях управления и искусственного интеллекта. – М.: Наука, Гл. ред. физ.-мат. лит., 1986.
- [Батыршин, 2001] Батыршин И.З. Основные операции нечеткой логики и их обобщения. – Казань: Отечество, 2001.
- [Батыршин и др., 2007] Батыршин И.З., Шереметов Л.Б. Модели и методы перцептивного дата майнинга временных рядов для систем поддержки принятия решений. – Нечеткие системы и мягкие вычисления, т.2, 1, 2007, 29–52.
- [Молодцов, 2004] Молодцов Д.А. Теория мягких множеств. – М.: Едиториал УРСС, 2004.
- [Петровский, 2003] Петровский А.Б. Пространства множеств и мультимножеств. – М.: Едиториал УРСС, 2003.
- [Agrawal et al., 1993] Agrawal R., Faloutsos C., Swami A. Efficient similarity search in sequence databases. // Fourth International Conference on

Foundations of Data Organization, D. Lomet, Ed., Heidelberg: Springer-Verlag, 1993, pp. 69–84.

[Batyrshin et al., 2007] Batyrshin I., Herrera-Avelar R., Sheremetov L., Panova A. Moving approximation transform and local trend associations in time series data bases. // Batyrshin I., Kacprzyk J., Sheremetov L., Zadeh, L.A. (eds). Perception-based Data Mining and Decision Making in Economics and Finance. Studies in Computational Intelligence, vol. 36, pp. 55-83. Berlin: Springer PhysicaVerlag, 2007.

[Batyrshin et al., 2012] Batyrshin I., Sheremetov L., and Velasco-Hernandez J. X. On axiomatic definition of time series shape association measures. // ORADM 2012, Workshop on Operations Research and Data Mining, Cancun, 2012, pp. 117–127.

[Batyrshin, 2013a] Batyrshin I. Constructing time series shape association measures: Minkowski distance and data standardization. // BRICS CCI 2013, Brasil, Porto de Galhinas, 2013. <http://arxiv.org/pdf/1311.1958>.

[Batyrshin, 2013b] Batyrshin I. Association measures and aggregation functions. // Advances in Soft Computing and Its Applications. Lecture Notes in Computer Science, vol. 8266, pp. 194-203. Berlin: Springer, 2013.

[Beliakov et al., 2008] Beliakov G., Pradera A., and Calvo T. Aggregation functions: A guide for practitioners. Springer Publishing Company, Incorporated, 2008.

[Fodor et al., 1994] Fodor J. and Roubens M. Fuzzy preference modelling and multi-criteria decision support. – Kluwer: Dordrecht, 1994.

[Grabisch et al., 2009] Grabisch M., Marichal J.-L., Mesiar R. and Pap E. Aggregation Functions. – Cambridge: Cambridge University Press, 2009.

[Han et al., 2006] Han J. and Kamber M. Data mining: concepts and techniques. 2nd ed. – Amsterdam: Morgan Kaufmann, 2006.

[Klement et al., 2000] Klement E. P., Mesiar R. and Pap E., Triangular norms. – Dordrecht: Kluwer, 2000.

[Murthy et al., 1985] Murthy C. A., Pal S. K., and Majumder D. Dutta. Correlation between two fuzzy membership functions. // Fuzzy Sets and Systems vol. 17, 1, pp. 23–38, 1985.

[Wilcox, 2010] Wilcox R.R. Fundamentals of Modern Statistical Methods. Substantially Improving Power and Accuracy. 2nd ed. – New York: Springer, 2010.

УДК 004.8:519.2

ВКФ-МЕТОД ПОРОЖДЕНИЯ ГИПОТЕЗ: ПРОГРАММНАЯ РЕАЛИЗАЦИЯ¹

Д.В. Виноградов (*vin@viniti.ru*)
ВИНИТИ РАН и РГГУ, Москва

Обсуждаются ВКФ-метод порождения гипотез, его программная реализация и ее применение к массиву SPECT [Kurgan et al., 2001] из UCI MLR [Bache et al., 2013]. Подобно ДСМ-методу наш подход комбинирует индукцию, аналогию и абдукцию. Основные отличия – использование цепи Маркова в процедуре индукции и введение алгоритма уточнения гипотез в процедуре абдукции. ВКФ-система использует простейшую модель и демонстрирует преимущество нашего подхода на массиве SPECT (точность 85.56 % превосходит 84.0% точность алгоритма CLIP3 [Cios et al., 1997] авторов этого массива).

Введение

В 1983 году В.К. Финн предложил формализацию индуктивных методов Д.С.Милля средствами многозначных логик [Финн, 1983] (ДСМ-метод). Этот метод сочетает в себе индуктивное обобщение обучающих примеров в гипотезах о причинах, доопределение свойств неизвестных примеров по аналогии с обучающими и абдуктивное принятие гипотез [Финн, 1998]. Хорошо известно, что основная вычислительная сложность алгоритмов ДСМ-метода сосредоточена на этапе индуктивного порождения гипотез [Кузнецов, 1992]. На практике довольно часто число ДСМ-гипотез на несколько порядков превосходит объем обучающей выборки. Мы думаем, что результаты, полученные в результате излишне затраченных вычислительных ресурсов, не имеют и содержательного смысла. Эксперту сложно отобрать среди множества ДСМ-гипотез, часто отличающихся единственным признаком, те, которые могут им быть объявлены «причинами» исследуемых свойств.

Ранее [Виноградов, 2012] мы предложили подход к порождению гипотез, основанному на использовании цепей Маркова, для по-

¹ Работа выполнена при поддержке программы фундаментальных исследований Президиума РАН «Математическое моделирование и интеллектуальные системы» на 2014 год и гранта РФФИ (проект 14-07-00856а).

лучения случайной подвыборки из множества (экспоненциального числа в худшем случае) всех гипотез. В работе [Vinogradov, 2014] мы расширили этот подход до полной схемы порождения гипотез, соединяющей индукцию, абдукцию и аналогию, который мы называем ВКФ-метод.

1. Необходимые определения

Обучающей выборкой назовем тройку (O, F, E) , где $E \subseteq O \times F$. O называется **множеством объектов**, а F – **множеством признаков**. Каждый объект $o \in O$ описывается подмножеством множества признаков $o' = \{f \in F \mid (o, f) \in E\}$.

Пересечением (или **замыканием**) подмножества $A \subseteq O$ объектов назовем подмножество $A' = \{f \in F : \forall o \in A [(o, f) \in E]\}$. Другими словами, пересечение $A' \subseteq F$ подмножества $A \subseteq O$ объектов является подмножеством всех общих признаков всех объектов из множества A . Заметим, что пересечением пустого подмножества объектов является все множество признаков $\emptyset' = F$.

Замыканием подмножества $B \subseteq F$ признаков назовем подмножество $B' = \{o \in O : \forall f \in B [(o, f) \in E]\}$. Другими словами, замыкание $B' \subseteq O$ множества $B \subseteq F$ признаков является подмножеством всех объектов из множества O , обладающих всеми признаками из множества B . Замыканием пустого подмножества признаков является все множество объектов $\emptyset' = O$.

ВКФ-кандидатом для (O, F, E) называется всякая пара (A, B) , состоящая из подмножества (называемого **списком родителей**) объектов $A \subseteq O$ и подмножества признаков $B \subseteq F$ (отождествляемого с **ВКФ-кандидатом**), удовлетворяющая условиям $A' = B$ и $B' = A$ (последнее условие называется **исчерываемостью**).

Порядок на ВКФ-кандидатах определяется так: $A_1, B_1 \leq A_2, B_2$, если и только если $B_1 \subseteq B_2$ (или, что эквивалентно, как $A_1 \supseteq A_2$).

2. Общая схема ВКФ-метода

Шаг индукции: использует спаривающуюся цепь Маркова для порождения заранее предписанного числа ВКФ-кандидатов. Проверяем каждого из них на мощность числа родителей и наличие контрпримеров (примеров противоположного знака, содержащих в себе все множество признаков исследуемого ВКФ-кандидата).

Увеличивая нижнюю границу на число родителей ВКФ-кандидата, мы увеличиваем «надежность» кандидата, делая процедуру менее чувствительной к ошибкам в данных. Удаляя ВКФ-кандидаты, которые имеют контрпримеры, мы оставляем только «действующие» причины для исследуемого свойства.

Алгоритм 1. Состоянием спаривающейся цепи Маркова будет пара ВКФ-кандидатов $A_1, B_1 \leq A_2, B_2$. В момент времени $t=0$ начинаем с состояния X_0 , соответствующего паре ВКФ-кандидатов $O, O' \leq F', F$. В каждый момент времени t случайно равномерно выбирается один из элементов $O \cup F$. Пусть в этот момент времени t цепь Маркова находилась в состоянии X_t , соответствующем паре ВКФ-кандидатов $A_1, B_1 \leq A_2, B_2$. Если был выбран некий объект $o \in O$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние X_{t+1} , соответствующее паре ВКФ-кандидатов $(A_1 \cup \{o\})'', (A_1 \cup \{o\})' \leq (A_2 \cup \{o\})'', (A_2 \cup \{o\})'$. Если был выбран некий признак $f \in F$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние X_{t+1} , соответствующее паре ВКФ-кандидатов $(B_1 \cup \{f\})', (B_1 \cup \{f\})'' \leq (B_2 \cup \{f\})', (B_2 \cup \{f\})''$. Когда будет достигнуто состояние, соответствующее паре совпадающих ВКФ-кандидатов $A, B \leq A, B$, то проверяются условия $|A| \geq m$ (число родителей не меньше выбранного числа m) и $\neg \exists c [B \subseteq c']$ (**запрет на контрпримеры**). Если обе проверки заканчиваются успешно, то выдается ВКФ-гипотеза A, B и процесс перезапускается с начального состояния $X_0 : O, O' \leq F', F$. Алгоритм останавливается после выдачи требуемого числа ВКФ-гипотез.

Шаг абдукции: предварительно использует алгоритм уточнения порожденных ВКФ-гипотез с помощью обучающих примеров. Для каждой порожденной ВКФ-гипотезы A, B и каждого объекта $o \in O$ вычисляем ВКФ-кандидат $(A \cup \{o\})'', (A \cup \{o\})'$. Снова проверяем условие $\neg \exists c [B \subseteq c']$ (запрет на контрпримеры). Если условие выполняется, то записываем новую ВКФ-гипотезу $(A \cup \{o\})'', (A \cup \{o\})'$, а объект $o \in O$ считаем **объясненным**. Если все объекты будут объясненными, то множество порожденных ВКФ-гипотез **принимается на достаточном основании**. В противном случае, выдаем необъясненные примеры, чтобы пополнить обучающую выборку их аналогами.

Шаг аналогии: предсказывает целевое свойство у объектов, представленных на прогноз. Объект p *доопределяется положительно* (целевое свойство проявляется), если $\exists A, B [B \subseteq p']$. Если $\neg \exists A, B [B \subseteq p']$, то целевое свойство объекта p предсказывается отсутствующим. Это соответствует простейшей стратегии ВКФ-метода. Можно использовать более сложные стратегии, аналогичные таковым в ДСМ-методе.

3. Программная реализация ВКФ-метода

Простейший вариант ВКФ-метода был реализован как программа **ВКФ-система** на C++ с использованием библиотеки boost. Множества признаков представляются объектами класса `boost::dynamic_bitset`. Они сохраняются в контейнерах `std::vector` и `std::list` из стандартной библиотеки C++.

Программа использует `boost::random` классы для генерации случайных признаков и объектов на шаге индукции (внутри спаривающейся цепи Маркова).

Так как индукция соответствует независимым прогонам цепи Маркова, она допускает очевидное распараллеливание. Программа вычисляет и шаг аналогии в несколько потоков. ВКФ-система реализует *многопоточность* через POSIX `pthread`. Библиотека boost содержит классы `boost::thread`, но реализация такого распараллеливания показала меньшую эффективность. Современная реализация POSIX `pthread` под Microsoft Windows использует WinAPI напрямую. Однако программа для ВКФ-метода является *платформенно-независимой*.

ВКФ-система реализована как приложение командной строки (*console*). При ее разработке использовалась среда Code::Blocks (версия 13.12) с открытыми кодами. Компилятором был GNU C++ toolset (версия 4.8.1).

4. Экспериментальная проверка ВКФ-метода

ВКФ-система, описанная ранее, была применена к массиву SPECT [Kurgan et al., 2001] из репозитория для машинного обучения [Vache et al., 2013] Университета Калифорнии в Ирвайне.

Этот массив описывает классификацию изображений сердца при компьютерной томографии излучения единичных протонов

(SPECT). Пациенты классифицировались или как больные, или как здоровые. Признаки были отшкалированы так, чтобы каждое изображение описывалось 22 бинарными признаками. Для улучшения качества прогноза мы добавили отрицания признаков как новые признаки, чтобы отсутствие признака могло быть частью причины. Поэтому каждый объект описывается подмножеством из 44 бинарных признаков.

Массив SPECT состоит из двух таблиц. Одна таблица соответствует обучающей выборке и содержит 40 положительных и 40 отрицательных примеров. Вторая (тестовая) таблица содержит 172 положительных и 15 отрицательных примеров, которые представлены для проверки качества обучения.

Программная реализация ВКФ-системы использует простейшую модель порождения ВКФ-гипотез (только положительные гипотезы, отрицательные примеры используются только для проверки запрета на контрпримеры). Но даже эта модель показала преимущество нашего подхода на массиве SPECT. Из 172 положительных тестовых примеров 151 был доопределен правильно, а из 15 отрицательных правильно предсказано 9.

Таким образом, точность прогноза простейшей ВКФ-стратегии оказалась равной 85.56 %. Авторы массива [Cios et al., 1997] предложили подход машинного обучения CLIP (обучение покрытия, основанное на целочисленном программировании). Третья версия их системы CLIP3 на массиве SPECT достигла точности 84.0 %. Авторы массива утверждают, что ансамбль их систем четвертой версии CLIP4 ensemble достиг точности 90.4 %. Но нам представляется, что взвешенная природа принятия решения таким ансамблем слабых классификаторов имеет существенный недостаток интерпретируемости с точки зрения экспертов в предметных областях.

Заключение

Ранее в статье [Vinogradov, 14] был предложен ВКФ-метод порождения гипотез, основанный на спаривающейся цепи Маркова. Подобно ДСМ-методу [Финн, 09] наш подход комбинирует индукцию, аналогию и абдукцию. Основные отличия – использование спаривающейся цепи Маркова в процедуре индуктивного обобщения и введение дополнительного алгоритма уточнения гипотез в про-

цедуре абдукции. В этом докладе описана его программная реализация и ее применение к массиву SPECT [Kurgan et al., 2001] из UCI репозитория для машинного обучения [Bache et al., 2013].

Программная реализация ВКФ-системы использует простейшую модель порождения гипотез (только положительные гипотезы, отрицательные примеры используются только для проверки запрета на контрпримеры). Но даже эта модель показала преимущество нашего подхода на массиве SPECT (точность 85.56 % превосходит 84.0% точность алгоритма CLIP3 [Cios et al., 1997] авторов этого массива).

Программа ВКФ-система реализована как приложение командной строки на языке C++ с использованием библиотеки boost. При разработке использовались средства с открытыми исходными текстами Code::Blocks (версия 13.12) и GNU C++ toolset (версия 4.8.1).

Из-за независимости запусков спаривающей цепи Маркова, лежащей в основе шага индукции, программа допускает очевидное распараллеливание. ВКФ-система использует многопоточность, основанную на POSIX pthread. Существует также версия, основанная на *boost::thread*, но она демонстрирует большую длительность вычислений.

В работах по применению ДСМ-метода к медицинским задачам [Панкратова и др., 2011] признаки стали иметь разный статус: *необходимые*, *существенные* и *сопутствующие*. Представляется интересным исследовать вопрос о расширении предложенного подхода к этой ситуации.

Благодарности. Автор благодарит В.К. Финна за поддержку, С.О. Кузнецова и М.А. Михеенкову за полезные обсуждения. В работе над программной реализацией существенную помощь автору оказала Т.А. Волкова, которой он искренне благодарен.

СПИСОК ЛИТЕРАТУРЫ

[Виноградов, 2012] Виноградов Д.В. Вероятностное порождение гипотез с помощью простейших цепей Маркова // Научно-техническая информация. Серия 2. 2012. № 9. С. 21–27.

[Кузнецов, 1992] Кузнецов С.О. О сложности обучения и классификации, основанных на операции сходства // Труды 3-й национальной конференции по искусственному интеллекту (КИИ-92). 1992. Том 1. С. 32–34.

[Панкратова и др., 2011] Панкратова Е.С., Виноградов Д.В. Формальное описание настройки интеллектуальных ДСМ-систем на область клинической и лабораторной диагностики // Научно-техническая информация. Серия 2. 2011. № 9. С. 1–5.

[Финн, 1983] Финн В.К. О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона – Д. С. Милля // Семиотика и информатика. Вып. 20. 1983. С. 35–101.

[Финн, 1998] Финн В.К. Синтез познавательных процедур и проблема индукции // Научно-техническая информация. Серия 2. 1998. № 1–2. С. 3–52.

[Bache et al., 2013] K. Bashe K. and Lichman M. UCI Machine Learning Repository – Irvine (CA): SICS UCI. 2013 – <http://archive.ics.uci.edu/ml>.

[Cios et al., 1997] Cios K.J., Wedding D.K., and Liu N. CLIP3: cover learning using integer programming // Kibernetes, Vol. 26. 1997. № 4–5. 513–536.

[Kurgan et al., 2001] Kurgan L.A., Cios K.J., Tadeusiewicz R., Ogiela M., and Goodenday L.S. Knowledge discovery approach to automated cardiac SPECT diagnosis // Artificial Intelligence in Medicine, Vol. 23. 2001. № 2. 149–169.

[Vinogradov, 2014] Vinogradov D.V. VKF-method of hypotheses generation // Proc. Conf. on Analysis of Images, Social networks and Texts (AIST'14), Berlin: Springer (in press).

УДК 004.8

МЕТОДЫ ОБНАРУЖЕНИЯ ЗНАНИЙ В ЗАДАЧЕ СРАВНЕНИЯ ЭФФЕКТИВНОСТИ СТРАТЕГИЙ ЛЕЧЕНИЯ

А.И. Карачунский (*aikarat@mail.ru*)

Федеральный научно-клинический центр детской гематологии,
онкологии и иммунологии им. Д. Рогачева, Москва

Н.В. Корепанова (*korepanova.natalia@gmail.com*)

С.О. Кузнецов (*skuznetsov@hse.ru*)

Национальный исследовательский университет
«Высшая школа экономики», Москва

В статье предлагается процедура выделения подгрупп пациентов, в которых две стратегии лечения имеют различия в эффективности. Процедура основана на алгоритмах теории паросочетаний и деревьях решений. Представлены результаты применения данной процедуры на данных о детях, больных острым лимфобластным лейкозом.

Введение

Сегодня одним из самых многообещающих способов оптимизации терапии, особенно в детской гематологии, является анализ в подгруппах. Но практически любой экспертный выбор подгруппы для анализа сталкивается с критикой статистиков, связанной с необоснованностью выбора. В рамках данной работы предлагается подход к выявлению подгрупп пациентов с наличием или отсутствием различий в эффективности двух стратегий лечения, позволяя обосновать выбор подгруппы для анализа. В основе предлагаемой процедуры лежит три основных идеи: найти схожих между собой пациентов, попытаться выделить и описать классы пациентов с наличием или отсутствием различий в выживаемости двух стратегий лечения, проверить полученные гипотезы средствами анализа выживаемости.

Статья организована следующим образом: глава 1 посвящена предлагаемой процедуре выделения подгрупп, в главе 2 кратко описаны исходные данные для анализа, в главе 3 представлены результаты применения этой процедуры к исходным данным, и последняя глава представляет собой заключение.

1. Процедура поиска подгрупп

Предлагается процедура поиска подгрупп пациентов, в которых эффективность двух видов лечения не отличается или отличается значительно. В качестве меры эффективности рассматривается общая выживаемость [NCI, 2014]. Входные данные представляют собой множество пациентов, получавших один из двух видов терапии. Каждый пациент описывается значениями его физиологических характеристик, временем жизни от даты постановки диагноза и результатом лечения.

1.1. Расстояние между пациентами

В первую очередь необходимо определить понятие расстояния между двумя пациентами. Но физиологические характеристики пациентов могут принимать как качественные, так и количественные значения, поэтому предлагается ввести определение сравнимости.

Определение 1. Два пациента **сравнимы**, если значения всех качественных признаков одного пациента совпадают со значениями соответствующих свойств другого. Если значения хотя бы одного качественного признака не совпадают, эти пациенты **несравнимы**.

Все количественные признаки центрируются вычитанием среднего и масштабируются делением на среднеквадратическое отклонение. На основании нормализованных значений количественных признаков между сравнимыми пациентами расстояние вычисляется с использованием одной из классических мер расстояния [Deza et al., 2009].

1.2. Поиск схожих пациентов

Вычислив расстояние между всеми парами сравнимых пациентов, получавших разное лечение, применим алгоритм Гейла и Шепли для построения стабильного паросочетания [Alkan et al., 2003; Gale et al., 1962] и получим пары схожих пациентов. В силу того, что в основе данного алгоритма лежит понятие предпочтения, введем соответствующее определение.

Определение 2. Пациент p **предпочитает** пациента q_1 пациенту q_2 , если расстояние между p и q_1 меньше расстояния между p и q_2 . Пациент p **безразличен** в выборе между q_1 и q_2 , если расстояния между p и q_1 , p и q_2 равны.

Так как с большой вероятностью определенные предпочтения являются строгими, будем считать приблизительно полученное паросочетание оптимальным [Alkan et al., 2003].

1.3. Выделение подгрупп и формирование гипотез

Визуализируем паросочетание следующим образом: на координатной плоскости каждой паре паросочетания поставим в соответствие точку так, что координата X соответствует времени выживания пациента, получавшего первый вид лечения, а по оси Y – второй. В зависимости от результатов предлагаемого способа визуализации можно попытаться разделить точки на плоскости, например, на такие классы: время выживания при получении первого лечения больше, чем при получении второго, время выживания при получении и того, и другого лечения мало и т.д. Согласно выбранному разбиению на классы, каждой точке, а значит, и каждому пациенту из соответствующей пары присваивается метка класса. Возникает задача классификации, которую предлагается решать при помощи деревьев решений с различными параметрами [Kotsiantis, 2013, Rokach et al., 2010], так как получаемые при их помощи гипотезы легко интерпретировать для врачей. Среди всех полученных в результате изменения параметров гипотез выбираются наиболее точно классифицируемые (точность классификации оценивается, например, при помощи кросс-валидации).

1.4. Проверка гипотез

Полученные гипотезы предлагается проверять стандартными инструментами анализа выживаемости: кривые Каплан-Майера [Glanz, 2011; Kleinbaum et al., 2012], логранговый критерий [Glanz, 2011; Kleinbaum et al., 2012], критерий эквивалентности [Foody, 2009] и вычисление чувствительности [Glanz, 2011; Daimon, 2008]. Подтвержденная этими методами гипотеза может быть представлена для дальнейшей проверки в рамках клинических исследований.

2. Исходные данные

Всего в исходной базе 1773 пациента в возрасте до 18 лет включительно. Каждый пациент описывается набором значений следующих физиологических характеристик: пол (мужской или женский), статус ЦНС (поражена, не поражена, нет информации),

статус средостения (поражено, не поражено, нет информации), иммунофенотип (8 типов), возраст (в годах), инициальный лейкоз (в млрд./л), пальпируемый размер печени (в см), пальпируемый размер селезенки (в см). Также для каждого пациента определены характеристики, на основании которых рассчитывается время жизни каждого пациента и общая выживаемость в группе: дата постановки диагноза, дата последнего осмотра, исход (жив, мертв, нет информации). Также для каждого пациента определена стратегия лечения (DEXA или MEDROL). Для проведения анализа из исходного набора данных были исключены пациенты, у которых отсутствовало значение хотя бы одной из характеристик. Также исключались пациенты, для которых отсутствует информация о вовлеченности ЦНС или средостения. Всего число полностью описанных пациентов составило 1501: 919 получавших DEXA и 582 получавших MEDROL.

3. Анализ и результаты

Все количественные признаки нормировались и на их основании рассчитывалось Евклидово расстояние [Deza et al., 2009] между всеми парами сравнимых пациентов, получавших разное лечение. Далее было построено и визуализировано паросочетание (рис. 1). Было замечено наличие кластеров точек, границы между которыми были формализованы согласно рис. 1. Для лучшего разделения классов точки между пунктирными линиями исключались из дальнейшего анализа.

В результате компьютерных экспериментов по применению деревьев с разными параметрами для получения описания классов было получено 13 гипотез, 6 из которых получили подтверждение в результате их проверки стандартными методами анализа выживаемости. Вот пример такой гипотезы: *для детей с размером селезенки не меньше 3.5 см и pre-pre-B или pre-B иммунофенотипом в возрасте старше 6.6 года MEDROL эффективнее DEXA*. Всего таких пациентов в базе 47. Кривые выживаемости представлены на рис. 2. Логранг-тест с достоверностью 95% данную гипотезу подтверждает. Чувствительность порядка 84%. Т.е. гипотеза может быть представлена гематологам для дальнейшей проверки.

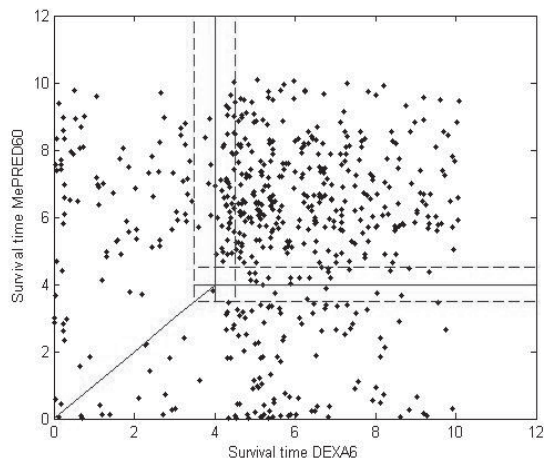


Рис. 1. Визуализация паросочетания с разбиением на классы

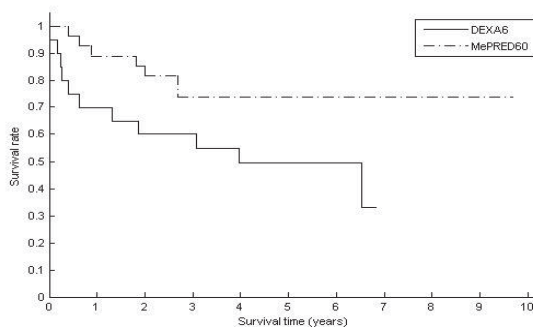


Рис. 2. Кривые Каплан-Майера для представленного примера гипотезы

Заключение

В данной работе представлен новый подход к решению проблемы поиска подгрупп пациентов с различной реакцией на два анализируемых вида лечения. Этот подход основан на построении оптимального стабильного паросочетания, классификации с использованием деревьев решений. Полученные гипотезы предлагается проверять стандартными методами анализа выживаемости. Все

шаги предлагаемой процедуры могут быть легко адаптированы к другой практической проблеме, содержащей анализ в подгруппах и выходящей за рамки гематологии и медицины.

СПИСОК ЛИТЕРАТУРЫ

[**Alkan et al., 2003**] Alkan, A., Gale, D.: Stable schedule matching under revealed preference. *Journal of Economic Theory*. 112, 289–306 (2003).

[**Daimon, 2008**] Daimon, T.: Bayesian sample size calculations for a non-inferiority test of two proportions in clinical trials. *Contemporary Clinical Trials*. 29, 507–516, Elsevier (2008).

[**Deza et al., 2009**] Deza, M.M., Deza, E.: *Encyclopedia of Distances*, pp. 94, 323–324. Springer Berlin Heidelberg (2009).

[**Foody, 2009**] Foody, G.M.: Classification accuracy comparison: Hypothesis tests and the use of confidence intervals in evaluations of difference, equivalence and non-inferiority. *Remote Sensing of Environment*. 13, 1658-1663, Elsevier (2009).

[**Gale et al., 1962**] Gale, D., Shapley, L.S.: College Admissions and the Stability of Marriage. *The American Mathematical Monthly*. Vol. 69, no. 1, 9–15 (1962).

[**Glanz, 2011**] Glanz, S.A.: *Primer of Biostatistics*, 7th ed., pp. 242–243. McGraw-Hill Education (2011).

[**Kleinbaum et al., 2012**] Kleinbaum, D.G., Klein, M.: Kaplan-Meier Survival Curves and the Log-Rank Test. In: *Survival Analysis*, pp. 55–96. Springer New York (2012).

[**Kotsiantis, 2013**] Kotsiantis, S.B.: Decision trees: a recent overview. *Artificial Intelligence Review*. 39, 261–283 (2013).

[**Rokach et al., 2010**] Rokach, L., Maimon, O.: Classification trees. In: *Data Mining and Knowledge Discovery Handbook*, 2nd ed., pp. 149-174. Springer US (2010).

[**NCI, 2014**] NCI Dictionary of Cancer Terms, <http://www.cancer.gov/dictionary?cdrid=655245> (18.05.2014).

ПРЕОБРАЗОВАНИЕ ПРОСТРАНСТВА ПРИЗНАКОВ С УЧЕТОМ ПОВЕДЕНИЯ ОШИБОК ОБОБЩЕНИЯ

М.О. Корлякова (*mkorlyakoa@yandex.ru*)

А.Ю. Пилипенко (*gamerjd@mail.ru*)

Московский государственный технический университет,
Калужский филиал, Калуга

Представлены результаты анализа процедуры формирования обучаемых моделей с точки зрения однородности ошибок. Показано, что построение описаний объектов с малой дисперсией ошибок позволяет значительно повысить качество решения задачи классификации и аппроксимации.

Введение

Значительное число задачи интеллектуальной обработки информации и данных можно свести к поиску взаимосвязи выходных признаков с входными на основании изучения представленного множества примеров. Такие модели принято называть индуктивным машинным обучением [Вагин и др., 2008]. Сравнительно простая форма обучения с учителем для получения решения делает подобные методы эффективными при формировании моделей, ориентированных на распознавание образов, моделирование объектов или явлений по результатам экспериментов и многое другое. Общая стратегия решения задач такого типа может быть представлена в следующем виде:

1. Получение начальной формы описания объектов и формирование целевого свойства из внешних источников.
2. Анализ системы входных признаков и их линейное или нелинейное преобразование, а также селекция наиболее ценных признаков для повышения общей информативности входов.
3. Построение наилучшего с точки зрения ошибки обобщения решателя. При этом проводят поиск не только параметров, но и структуры решателя.

Задача аппроксимации отличается от задачи классификации числом целевых состояний. Каждому примеру соответствует собственное значение целевого свойства. Однако это не исключает не-

обходимости правильно описать примеры, на которых происходит обучение модели, а общая схема обработки идентична задачам классификации. Каждый этап вносит свой вклад в получение качественного или слабого решения. Рассмотрим особенности решения задач формирования аппроксиматоров для задач с большим количеством примеров и признаков.

1. Методы формирования пространства признаков

Основными методами улучшения описания объектов при решении задач обучения по признакам являются технологии селекции и преобразования признаков. Селекция признаков в скалярном или векторном виде позволяет из существующего набора признаков выделить наиболее интересные с точки зрения цели обучения. Пример векторной селекции – метод ADD [Загоруйко, 1999], который обеспечивает существенное уменьшение числа проверок за счет итеративного наращивания списка признаков. Результирующая модель содержит только те признаки, которые были в исходной модели, но позволяет провести обоснованное сокращение их количества.

Более интенсивное воздействие на исходное описание задачи оказывают методы преобразования пространства признаков, среди которых можно выделить два основных типа преобразований: линейные и нелинейные. Пример линейной модели – метод главных компонент (РСА) [Местецкий, 2004], который позволяет развернуть систему координат в пространстве объектов в направлении наибольшей дисперсии. Нелинейные модификации РСА [Gorban et al., 2008; Scholz, 2012] направлены на получение подобного эффекта для примеров, образующих в пространстве признаков скрытые структуры. Для решения задач классификации наиболее эффективные результаты демонстрируют подходы, связанные с выделением групп объектов в исходном пространстве признаков. Такие, как, например, подходы на основе кластерного анализа [Загоруйко, 1999] выделяют группы близких объектов и соотносят с ними новые признаки. Методы такого типа часто используют в качестве модели предварительного анализа пространства признаков, поскольку они не всегда обеспечивают удачное выделение групп. Методы функционального многомерного шкалирования [Авен и др., 1988] позволяют выделять интегральные признаки с максимальным сохранением расстояния между объектами, что по сравнению с вы-

делением кластеров порождает более сложные конгломераты исходных признаков. Подход на основе интерактивной процедуры поиска отображения пространства признаков в модель меньшей размерности (метод ИСКРА) [Петровский и др., 2010] позволяет значительно повысить качество и обоснованность получаемых признаков за счет включения эксперта в процесс отбора. Все перечисленные модели ориентированы на решение задач классификации для небольшого числа классов.

Для задач аппроксимации целевого свойства поиск наилучшей модели описания выходов через входы можно обеспечить за счет МГУА [Ивахненко и др., 1987], который позволяет выбрать наиболее эффективную модель представления экспериментальной информации в форме рекурсивного объединения моделей на основе анализа уровня ошибок. Каждый следующий уровень рекурсии оперирует более общими признаками, чем предшествующий. Теоретически этот подход позволяет описать модель любой сложности. Сложность алгоритма реализации МГУА зависит от числа рядов селекции и вида самой функции. Это приводит к тому, что для неудачных входных признаков сложность МГУА будет значительной.

Подходы, ориентированные на построение решателей, а не на анализ признаков и их преобразование, составляют множество методов формирования решающего правила, которое обеспечит наилучшее возможное описание целевых свойств через существующие входные признаки. В частности, при решении задачи бинарной классификации используют минимизацию структурного риска [Воронцов, 2004; Вапник, 1979], которая позволяет проводить синтез оптимальной модели на основе исследования ошибки обучения и оценки сложности классификатора. При этом следует учитывать, что мультиклассовые классификаторы не всегда оптимально формируются на основе минимизации структурного риска [Daniely, 2014]. Однако выбор наилучшей модели решателя опирается на исследование ошибки обучения и оценку ошибки обобщения.

Поведение ошибок обучения и обобщения может значительно варьироваться в разных группах входных данных. Фактически для любой формы описания входов\выходов всегда есть области, хуже представленные формируемым решателем. Учет разнообразного поведения классификатора для различных частей входного мно-

жества развивается моделями типа усиления (boosting). Эффект достигается за счет организации коллектива решателей на основе анализа ошибок. Например, методы типа AdaBoost строят коллективы решателей относительно подгрупп примеров для обучения в соответствии с остающимися ошибками. Алгоритмы класса AdaBoost и его аналоги могут сформировать решение для выборки объектов любой сложности при неограниченном увеличении числа решателей в коллективе [Хомяков и др., 2010; Fercoq, 2013]. Другой вопрос, что такой экстенсивный подход приводит к быстрому росту объема коллектива решателей и усложнению процедуры построения результатов. При этом можно говорить, что определяющую роль в росте объема группы классификаторов играет сложность пространства объектов. Таким образом, описание объектов является важным для любой модели обучения. Рассмотрим анализ поведения ошибок решателя в зависимости от состава входных признаков.

2. Построение пространства входов при решении задачи определения параметров перемещения стереопары в сцене

Общие ошибки модели решатель-модель необходимо разделять на те, которые зависят от проблем формирования решателя и неспособности модели входов представить выход. Пусть существует модель $T = \{(X_i, d_i)\}, i = 1..N$, тогда, построив множество решателей $F^* = \{F_j(X, W)\}, j = 1..m$, где каждый решатель принадлежит к своему типу функций, получим m независимых решений с комплексами ошибок $E = \{E_j(T_{test} = \{(X, d)\}), T_{test} \neq T\}$. Относительные ошибки E_j получены для наилучшей модели типа $F_j(X, W)$ по тестовой выборке $T_{test} = \{(X, d)\}$ для каждой точки выборки. Будем считать, что $ET = \cap (E_j > \varepsilon), \varepsilon > 0, \varepsilon < 1$ – ошибка, связанная только с дефектами описания объекта. Эта ошибка существует во всех $F_j(X, W)$ и не зависит от их типа. Область, где отмечена такая ошибка, имеет отличную от остальной части выборки дисперсию. Существование такой ошибки свидетельствует о неспособности текущих целевых свойств быть представленными через текущие входные признаки любой моделью. Необходимо пересмотреть исходную модель признаков, а область аномальной

дисперсии ошибки указывает на ту часть модели, которая должна быть изменена.

Таким образом, исследуя неоднородность поведения ошибки разнородных решателей, можно определить направление преобразования модели входов. Рассмотрим процесс изменения модели входных признаков при решении задачи определения параметров перемещения стереопары камер в сцене во время движения мобильной платформы.

Наблюдение объектов сцены с помощью стереопары камер позволяет устанавливать координаты отдельных объектов сцены за счет сопоставления позиций проекций точки в «левом» и «правом» кадре. Рассматривая последовательные во времени стереопары кадров, можно определить перемещение объектов сцены относительно камеры или установить перемещение камер относительно сцены. Формальное решение задачи определения изменения положения системы координат наблюдателя в среде неподвижных объектов по изменению координат этих объектов в моменты времени t_1 и t_2 можно описать следующим образом [Ивашина, 2013]:

$$\begin{aligned} M \cdot V &= V_{new} \\ M &= V_{new} \cdot V^{-1}, \end{aligned} \quad (1)$$

где M – матрица переноса-поворота объекта сцены (или обратная матрица передвижения системы координат), V – начальное положение точек наблюдаемой сцены, V_{new} – новое положение тех же точек сцены. Матрица M содержит 12 неизвестных параметров, однозначно определяющих изменение положения объекта. Таким образом, для решения задачи в идеальном случае достаточно знания пространственных координат 4-х точек сцены в исходном положении и в текущем. Однако, как показано в работе [Ивашина, 2013], точное определение координат объектов невозможно, что значительно усложняет поставленную задачу. Система уравнений (1) имеет единственное решение для каждой пары V, V_{new} . Таким образом, существует функция $M(V, V_{new})$, которая может быть аппроксимирована в нейросетевой модели.

Будем использовать три типа нейросетей: линейную двухслойную, сеть радиальных нейронов и нелинейный двухслойный перцептрон. Прямое решение задачи практически нереализуемо из-за бесконечного разнообразия пар V, V_{new} . Сужение диапазона из-

менчивости положений позволяет улучшить качество решения, но снижает его ценность для практического использования. Однако ошибки решения задачи в значительной мере неоднородны и это позволяет говорить, что необходимо провести преобразование модели входов.

Перемещение объекта V можно представить в виде последовательного переноса-поворота. Перенос-поворот объекта сцены M можно записать в следующем виде: $M = R \cdot T$, R – поворот, T – перенос. Разделим процесс определения M на два этапа. Вычисление поворота системы координат R можно проводить относительно центра масс опорных точек в момент t_2 и t_1 . При переносе не происходит изменения угловой ориентации. Поэтому возможно рассматривать отдельно перенос и поворот. Таким образом, снижается изменчивость аппроксимируемой модели за счет разделения ее на 2 независимые составляющие.

Относительные ошибки определения угла поворота в нейросетевых решателях по истинным значениям поворота приведены на рис.1.а для линейной, радиальной и нелинейной нейросети. Видно, что для всех аппроксиматоров существует область аномальной дисперсии ошибки для диапазона угла $[-0.5, -0.3]$, а уровень ошибок велик и для всех областей.

Следующий шаг снижения сложности аппроксимируемой модели поворота связан с исключением разнообразия, порождаемого произвольным размещением объектов в сцене. Для определения 9-ти параметров поворота R необходимо иметь 3 произвольные точки (не на одном луче и отличные от начала координат). В реальной сцене можно определить значительно большее число наблюдаемых в моменты t_2 и t_1 точек. Однако координаты точек определены с ошибкой, что приведет к искажению R .

Для сокращения изменчивости модели будем рассматривать орты, сформированные как направления наибольшей дисперсии точек сцены, наблюдаемых одновременно. Это позволяет рассматривать не конкретную ошибку 3-х точек, а усредненное положение орта и его ошибку, которая не превышает средней ошибки по всем точкам сцены. Рассматривая поворот сцены относительно системы ортов, считаем Ort_1 начальным положением ортов в момент t_1 для точек V , а Ort_2 конечное положение, определенное по V_{new} . Тогда $R = Ort_1^{-1} \cdot Ort_2$.

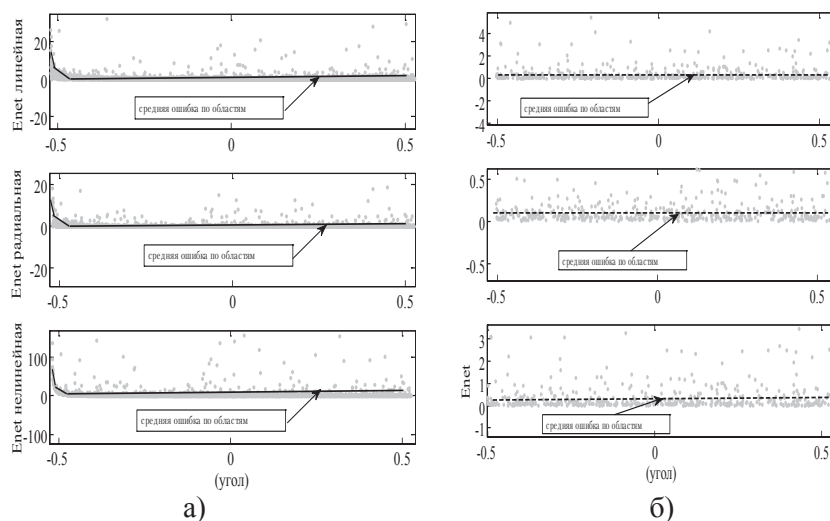


Рис. 1. Относительные ошибки для модели 1 а) и модели 4 б).

Оценка поведения относительных ошибок определения угла показала, что на всем диапазоне изменения угла $[-0.5, 0.5]$ все решатели работают с постоянной дисперсией. Результаты моделирования в виртуальной модели сцены для разных вариантов входов приведены в таблице 1. Из табл. 1 и рис. 1 б) видно, что ошибка модели 4 однородна по всем решателям. Это соответствует более качественной форме входной модели данных и качественной работе аппроксиматоров и лучшей устойчивости к ошибкам координат.

Таблица 1

№	Тип входов модели	Выход	Ошибка обобщения. Для поворота в $^{\circ}$, переноса в мм. Идеальные координаты	Ошибка обобщения. Для поворота в $^{\circ}$. Ошибка в V и V_{new} до 20 мм
1	Координаты точек сцены V и V_{new}	M	$2,94^{\circ}$, 128,38	От 7.3 до 80° для разных типов решателя
2	Координаты точек сцены V и V_{new}	R	1.2° , –	От 3.6 до 31.6° для разных типов решателя

Продолжение таблицы 1

3	Центровка координат точек сцены	R	1,1 °, –	От 3.8 до 5.3 ° для разных типов решателя
4	Орты	R	0.12 °, -	≈3.3 ° для всех решателей

Сложность пространства для всех моделей различна и может быть описана следующим образом:

- Модель 1. 120 входов для 12 выходов (40 точек сцены), поворот сцены $[-8^\circ, +8^\circ]$ по трем осям, перенос $[-500, 500]$ мм.
- Модель 2. 120 входов для 9 выходов (40 точек сцены), поворот $[-8^\circ, +8^\circ]$ по трем осям без построения параметров переноса.
- Модель 3. 120 входов для 12 выходов (40 точек сцены без переноса), поворот сцены $[-8^\circ, +8^\circ]$ по трем осям для централизованной модели координат вида $([V]-V_m, [V_{new}]-V_{m_new})$, где V_m, V_{m_new} – среднее значение V и V_{new}
- Модель 4. 12 входов (координаты концов единичных ортов с поворотом сцены к первому положению в точках V) для 9 выходов (200 точек сцены для построения ортов), угол $[-8^\circ, +8^\circ]$ по трем осям.

Заключение

Анализ поведения ошибок классификаторов позволяет делать вывод о качестве входной информации. Значительная разница в уровнях ошибок для различных областей характерна для всех типов решателей, но объединение информации об ошибках группы различных аппроксиматоров позволяет выделять проблемные области в данных. Построить модель предобработки входной информации можно, опираясь на изучение совместных ошибок группы решателей. Способ преобразования входной информации не определяется поведением ошибки, но позволяет определить наиболее эффективное направление поиска такого преобразования.

СПИСОК ЛИТЕРАТУРЫ

[Авен и др., 1988] Авен П.О., Ослон А.А., Мучник И.Б. Функциональное шкалирование.– М.: Наука, 1988.– 182 с.

[Вагин и др., 2008] Вагин В.Н., Головина Е.Ю., Загорянская А.А., Фомина М.В. Достоверный и правдоподобный вывод в интеллектуальных системах / Под ред. В.Н. Вагина, Д.А. Поспелова. – Изд. 2-е, М.: ФизМатЛит, 2008.

[Вапник, 1979] Вапник В.Н. Восстановление зависимостей по эмпирическим данным. – М.: Наука, 1979.

[Воронцов, 2004] Воронцов К.В. Комбинаторный подход к оценке качества обучаемых алгоритмов // Математические вопросы кибернетики. – 2004. – № 13. – С. 5–36.

[Загоруйко, 1999] Загоруйко Н.Г. Прикладные методы анализа данных и знаний. – Новосибирск: ИМ СО РАН, 1999.

[Ивашина, 2013] Ивашина Е.А., Корлякова М.О., Пилипенко А.Ю. Нейросетевое восстановление координат сцены в стереоскопической системе технического зрения // Гибридные и синергетические интеллектуальные системы: теория и практика: материалы 1-го международного симпозиума / под ред. проф. А.В. Колесникова. – Калининград: Из-во БФУ им. И. Канта, 2012. – С. 163–173.

[Ивахненко и др., 1987] Ивахненко А. Г., Юрачковский Ю. П. Моделирование сложных систем по экспериментальным данным. – М.: Радио и связь, 1987.

[Местецкий, 2004] Местецкий Л.М. Математические методы распознавания образов. Курс лекций – М.: МГУ, 2004. – 85 с.

[Петровский и др., 2010] Петровский А.Б., Ройзензон Г.В. Интерактивная процедура снижения размерности признакового пространства в задачах многокритериальной классификации // Поддержка принятия решений. Труды Института системного анализа Российской академии наук / Под ред. А.Б. Петровского – М: Едиториал УРСС, 2008. – Т.35. – С. 48–60.

[Хомяков и др., 2010] Хомяков М.Ю., Щеголева Н.Л. Сокращение вычислительной сложности классифицирующих алгоритмов семейства ADABOOST // Известия высших учебных заведений России. Радиоэлектроника, №4, 2010, с. 32–39.

[Daniely, 2014] Daniely A., Shalev-Shwartz S. Optimal Learners for Multiclass Problems – <http://arxiv.org/abs/1405.2420v1> (14.05.2014).

[Gorban et al., 2008] Gorban A. N., Kegl B., Wunsch D., Zinovyev A. Y. (Eds.), Principal Manifolds for Data Visualisation and Dimension Reduction, Series: Lecture Notes in Computational Science and Engineering 58, Springer, Berlin – Heidelberg – New York, 2008, XXIV, 340 p.

[Fercoq, 2013] Fercoq O. Parallel coordinate descent for the Adaboost problem // (Submitted on 7 Oct 2013) – <http://arxiv.org/abs/1310.1840v1>.

[Scholz, 2012] Scholz M. Validation of nonlinear PCA // (Submitted on 3 Apr 2012) – <http://arxiv.org/abs/1204.0684> (4.07.2014).

ПОВЫШЕНИЕ БЫСТРОДЕЙСТВИЯ ДСМ-МЕТОДА В ЗАДАЧАХ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

Е.В. Котельников (*kotelnikov.ev@gmail.com*)

Вятский государственный гуманитарный университет, Киров

В работе предлагается способ повышения быстродействия ДСМ-метода в задачах обработки текстов на основе совместной кластеризации терминов и документов с использованием параллельных вычислений. Описываются алгоритм совместной кластеризации и схема распараллеливания. Приводятся результаты экспериментов с текстовыми коллекциями.

Введение

Задачи автоматической обработки текстовой информации, такие как информационный поиск, классификация, кластеризация, аннотирование и т. п., требуют применения эффективных и высокопроизводительных методов и алгоритмов анализа данных. Поэтому их разработка составляет значительную часть содержания исследований в области искусственного интеллекта, компьютерной лингвистики и машинного обучения.

Одним из наиболее мощных и развитых методов анализа данных является ДСМ-метод автоматического порождения гипотез, предложенный В.К. Финном [Финн, 1983]. В ДСМ-методе формализуется взаимодействие трех познавательных процедур – индукции, аналогии и абдукции [Финн, 2010]. Свое название метод получил в честь английского философа и логика Д.С. Милля.

ДСМ-метод успешно применялся во многих областях, в том числе в фармакологии, медицинской диагностике, социологии, истории, криминалистике, робототехнике [Финн, 2009]. Однако примеров использования ДСМ-метода для автоматической обработки текстов немного, можно указать только две кандидатские диссертации [Кожунова, 2009] и [Смирнов, 2008]. Одной из причин такого редкого применения, по-видимому, является высокая вычислительная сложность этапа индукции ДСМ-метода, что ока-

зывается критичным при обработке текстовых корпусов, состоящих из тысяч и десятков тысяч документов.

В работе [Котельников, 2012] ДСМ-метод применялся для анализа тональности текстов, при этом с целью повышения производительности осуществлялась простая кластеризация текстовой коллекции. Однако в указанной работе использовались устаревшие алгоритмы поиска общих фрагментов, а ускорение обработки информации не исследовалось.

В настоящей работе предлагается способ повышения быстродействия ДСМ-метода на основе применения современных и эффективных алгоритмов совместной кластеризации терминов и документов, а также поиска общих фрагментов. Предложенный способ экспериментально оценивается с использованием текстовых коллекций различного объема.

1. ДСМ-метод

ДСМ-метод в данной работе будет представлен на языке теории множеств, следуя терминологии и обозначениям О.М. Аншакова [Аншаков, 2012].

Обозначим: \mathbf{O} – множество объектов (текстовых документов), \mathbf{A} – множество атомов (значимых слов-терминов), \mathbf{P} – множество целевых свойств (метки документов). Каждый текстовый документ представляется подмножеством терминов, принадлежащих множеству \mathbf{A} . Такое подмножество назовем фрагментом. Множество \mathbf{O} включает два типа объектов – те, для которых известно, обладают ли они целевым свойством (положительные и отрицательные примеры), и те, для которых такая информация неизвестна (неопределенные примеры). В терминах машинного обучения первый тип объектов образует обучающий набор данных $\mathbf{O}_{\text{train}}$, а второй – тестовый или распознаваемый \mathbf{O}_{test} [Flach, 2012].

Множество \mathbf{P} включает все интересующие в конкретной задаче целевые свойства текстовых документов. Например, в задаче анализа тональности такими свойствами являются метки, обозначающие принадлежность текстового документа к определенному классу тональности – позитивному или негативному.

ДСМ-метод включает три основных этапа: индукции, аналогии и абдукции. На этапе индукции автоматически порождаются гипотезы – возможные причины наличия или отсутствия целевых

свойств у объектов. Гипотезы формируются отдельно для каждого целевого свойства на основе поиска общих фрагментов у объектов, обладающих (или не обладающих) данным свойством. Таким образом, гипотеза представляет собой пару (фрагмент, свойство). На практике для вычисления степени полезности в гипотезу добавляется еще подмножество родителей – объектов, включающих общий фрагмент гипотезы.

Для каждого целевого свойства генерируются два множества гипотез: H^+ – гипотезы, являющиеся причиной наличия целевого свойства, и H^- – гипотезы, являющиеся причиной его отсутствия. Результатом этапа индукции являются $2 \cdot N$ множеств гипотез, где N – количество целевых свойств.

На этапе аналогии порожденные гипотезы применяются с целью определения наличия или отсутствия целевых свойств у неопределенных примеров. При этом множества положительных и отрицательных объектов пополняются за счет классифицированных неопределенных примеров. В зависимости от выбранной стратегии этапы индукции и аналогии могут повторяться до тех пор, пока множества порождаемых гипотез не перестанут изменяться [Финн, 2010]. После этого выполняется этап абдукции, на котором проверяется условие каузальной полноты – объясняют ли сформированные гипотезы исходные обучающие данные.

2. Способ распараллеливания

Наиболее сложной проблемой ДСМ-метода в вычислительном плане является задача поиска общих фрагментов. В работе [Кузнецов, 1989] показано, что данная задача для случая бинарного представления атомов является $\#P$ -сложной, а количество общих фрагментов может быть в экспоненциальной зависимости от числа объектов и атомов.

Для повышения скорости работы этапа индукции при обработке текстовой информации в статье предлагается способ распараллеливания. Его идея заключается в разделении входных данных на несколько частей на основе совместной кластеризации терминов и документов для дальнейшего распределения полученных кластеров по вычислительным узлам.

Совместная кластеризация терминов и документов представляет собой алгоритм выделения по коллекции документов таких

кластеров, в каждый из которых включаются одновременно документы и термины, сильно связанные между собой. Алгоритм совместной кластеризации был предложен в 2001 году И. Дхиллоном [Dhillon, 2001]. В этом алгоритме на вход поступает матрица «термин-документ» \mathbf{M} , в которой строки представляют термины, а столбцы – документы. Единичное значение в (i, j) -ячейке матрицы означает, что в j -м документе присутствует i -е слово. Входная матрица \mathbf{M} преобразуется в матрицу \mathbf{M}_n следующим образом:

$$\mathbf{M}_n = \mathbf{D}_1^{-1/2} \cdot \mathbf{M} \cdot \mathbf{D}_2^{-1/2},$$

где $D_1(i, i) = \sum_j A_{ij}$, $D_2(j, j) = \sum_i A_{ij}$.

После этого осуществляется сингулярное разложение матрицы \mathbf{M}_n с целью получения l левых и правых сингулярных векторов, начиная со второго [Голуб, 1999]. Количество сингулярных векторов определяется по формуле:

$$l = \lceil \log_2 k \rceil,$$

где k – требуемое количество кластеров.

Полученные левые и правые сингулярные векторы $\mathbf{U}_l = [u_2, \dots, u_{l+1}]$ и $\mathbf{V}_l = [v_2, \dots, v_{l+1}]$ совместно с матрицами \mathbf{D}_1 и \mathbf{D}_2 образуют матрицу \mathbf{Z} :

$$\mathbf{Z} = \begin{bmatrix} \mathbf{D}_1^{-1/2} \times \mathbf{U}_l \\ \mathbf{D}_2^{-1/2} \times \mathbf{V}_l \end{bmatrix}.$$

Строки матрицы \mathbf{Z} представляют собой векторы в l -мерном пространстве. Кластеризация этих векторов традиционными методами (например, с использованием алгоритма KMeans) позволяет получить искомое разбиение терминов и документов на k кластеров.

В предлагаемом способе распараллеливания осуществляется совместная кластеризация не только обучающих документов $\mathbf{O}_{\text{train}}$, но и тестовых \mathbf{O}_{test} . При этом в одном кластере оказываются сильно связанные обучающие документы (положительные и отрицательные примеры), тестовые документы (неопределенные примеры) и термины. Такой подход позволяет, во-первых, распараллелить этап аналогии, во-вторых, усилить критерий отбора гипотез следующим условием: гипотеза подходит для классификации неопределенного примера, если она сформирована из положительных или отрица-

тельных примеров, принадлежащих тому же кластеру, что и данный неопределенный пример.

Вопрос выбора количества кластеров решается в предлагаемом способе в зависимости от выбранного пользователем приоритета: качество классификации или скорость обработки. Если важнее первое, то выбирается такое разбиение, которое дает максимальное число достаточно больших кластеров. Если важнее скорость обработки, количество кластеров выбирается равным количеству узлов многопроцессорной вычислительной системы; при этом надо учитывать, что слишком мелкие кластеры могут не позволить получить удовлетворительное качество классификации.

Заметим, что в работе [Котельников, 2012] также предлагался подход к распараллеливанию на основе кластеризации. Однако совместная кластеризация не использовалась, неопределенные документы при этом не учитывались, а способ определения количества кластеров был тривиален.

Для непосредственного поиска общих фрагментов в ДСМ-системах, как правило, используется алгоритм Норриса [Norris, 1978] (см., например, [Кожунова, 2009]). Однако недавние исследования [Kirchberg, 2012] показали, что наиболее эффективными являются алгоритмы FCbO [Krajca, 2010] и In-Close2 [Andrews, 2011]. В предлагаемом методе использовался алгоритм In-Close2, поскольку он быстрее алгоритма FCbO для разреженных данных, какими являются тексты [Kirchberg, 2012].

На рис. 1 представлена схема распараллеливания ДСМ-метода для этапов индукции и аналогии (без пополнения исходных текстовых коллекций).

На рис. 1 исходные текстовые коллекции разделяются на k кластеров совместно со словарем, сгенерированным по обучающей коллекции. Затем сформированные кластеры распределяются по вычислительным узлам многопроцессорной системы и дальнейшая обработка происходит параллельно. В каждом кластере осуществляется генерация гипотез отдельно для положительных и отрицательных примеров на основе алгоритма In-Close2. Далее порожденные гипотезы на этапе аналогии применяются для классификации неопределенных примеров текстовой коллекции данного кластера. Результатами работы являются решения о принадлежности неопределенных примеров, принятые на этапе аналогии.

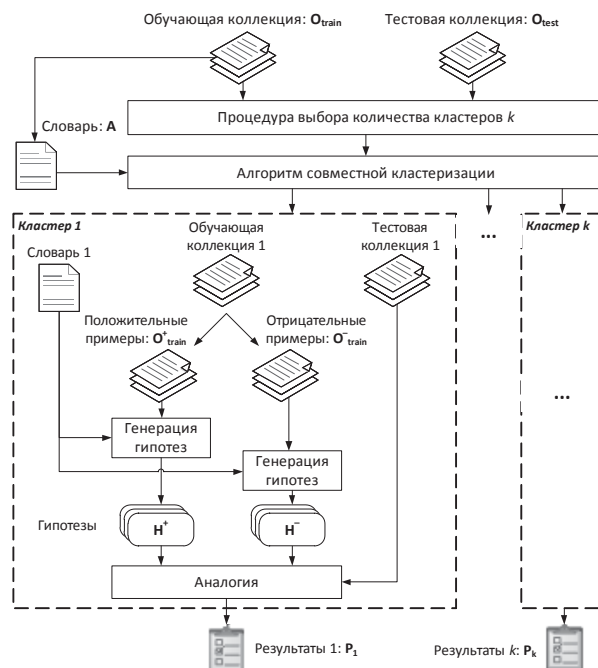


Рис. 1. Схема параллельных вычислений на этапах индукции и аналогии

Эксперименты

Для проведения экспериментов была разработана программа в среде Visual Studio 2010 на языках программирования C++ и C#. При этом C++ использовался для реализации ресурсоемких методов и алгоритмов, а C# – для работы со сложными структурами данных и для ввода-вывода.

Для совместной кластеризации использовалась математическая библиотека Intel Math Kernel Library (MKL, <https://software.intel.com/en-us/intel-mkl>), включающая оптимизированные параллельные алгоритмы линейной алгебры, в том числе для вычисления произведения матриц и сингулярных разложений. Использование MKL позволило распараллелить алгоритм совместной кластеризации (см. рис. 1), что повысило степень распараллеливания ДСМ-метода в целом. Для кластеризации KMeans в рамках алгоритма совместной кластеризации использовался математический пакет Octave (<http://www.gnu.org/software/octave>). Поиск общих

фрагментов был реализован на основе алгоритма In-Close2 (<http://inclose.sourceforge.net>), написанного на C++.

Исследования проводились с использованием текстовых коллекций отзывов о фильмах сайта Кинопоиск (<http://www.kinopoisk.ru>). Были поставлены эксперименты с двумя версиями программы – последовательной и параллельной. Количество кластеров для параллельной версии принималось равным числу ядер исследовательского компьютера (Intel Core i7-3630QM, 4 ядра, 8 Гб ОЗУ, 64-разрядная операционная система Windows 7).

На рис. 2 приведены зависимости времени работы обеих версий программы от суммарного количества документов и терминов. В табл. 1 представлены численные характеристики текстовых коллекций.

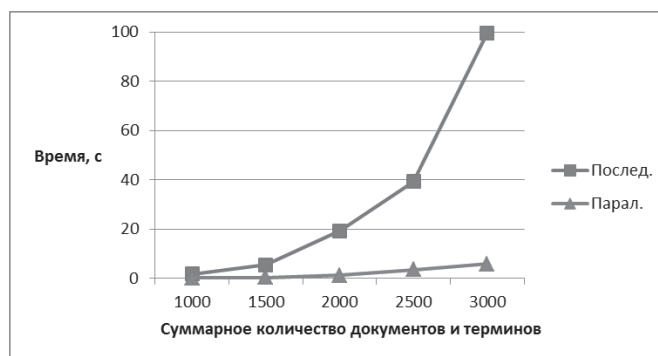


Рис. 2. Время работы последовательной и параллельной версий

Таблица 1

Документы	Термины	Сумма	Версия	Гипотезы	Время, с
150	850	1000	Послед.	220 000	1,6
			Парал.	25 000	0,1
280	1220	1500	Послед.	750 000	5,4
			Парал.	45 000	0,2
350	1650	2000	Послед.	4 600 000	19,2
			Парал.	95 000	1,1
440	2060	2500	Послед.	5 580 000	39,3
			Парал.	260 000	3,4

Продолжение таблицы 1

600	2400	3000	Послед.	13 900 000	99,69
			Парал.	450 000	5,7

Анализ рис. 2 и табл. 1 показывают, что предлагаемый способ позволяет существенно ускорить работу ДСМ-метода. Разделение обучающих данных даже на небольшое количество кластеров в значительной степени нивелирует экспоненциальный рост числа гипотез и времени их генерации. Точность классификации при распараллеливании отличается от последовательной версии на $\pm 3\%$ в зависимости от количества используемых терминов и документов: при суммарном количестве терминов и документов меньшем 2000 точность параллельной версии снижается, при превышении указанного количества точность сохраняется неизменной или даже повышается.

Следует также отметить, что при дальнейшем увеличении размера обучающей коллекции служебные структуры данных алгоритма In-Close2 в последовательной версии не помещались в ОЗУ исследовательского компьютера, в то время как в параллельной версии обработка успешно продолжалась.

Заключение

Таким образом, предложенный способ распараллеливания на основе совместной кластеризации позволяет существенно сократить время обработки текстовых коллекций при сохранении точности классификации за счет выделения в обучающих данных тесно связанных групп документов и терминов.

В дальнейших исследованиях предполагается изучить влияние количества гипотез на точность классификации, а также рассмотреть вопрос о предельных числовых характеристиках текстовых коллекций, которые возможно анализировать на заданной вычислительной платформе.

СПИСОК ЛИТЕРАТУРЫ

- [Аншаков, 2012] Аншаков О.М. ДСМ-метод: теоретико-множественное объяснение // НТИ. Сер. 2. 2012. № 9.
- [Голуб, 1999] Голуб Дж., Ван Лоун Ч. Матричные вычисления. – М.: Мир, 1999.

[Кожунова, 2009] Кожунова О.С. Технология разработки семантического словаря системы информационного мониторинга: автореф. дис. ... канд. техн. наук / ИПИ РАН, 05.13.17. – М.: ИПИ РАН, 2009.

[Котельников, 2012] Котельников Е.В. Опыт применения ДСМ-метода для определения тональности текста // Труды КИИ-2012. Т.1. – Белгород: Изд-во БГТУ, 2012.

[Кузнецов, 1989] Кузнецов С.О. Интерпретация на графах и сложностные характеристики задач поиска закономерностей определенного вида // НТИ. Сер. 2. 1989. № 1.

[Смирнов, 2008] Смирнов И.В. Исследование методов установления значений синтаксических единиц естественных языков на основе интеллектуального анализа данных: автореф. дис. ... канд. физ.-мат. наук / ИСА РАН, 05.13.01. – М.: ИСА РАН, 2008.

[Финн, 1983] Финн В.К. О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона – Д.С. Милля // Семиотика и информатика. 1983. Вып. 20.

[Финн, 2009] Автоматическое порождение гипотез в интеллектуальных системах / Сост. Е.С. Панкратова, В.К. Финн; под общ. ред. В.К. Финна. – М.: «ЛИБРОКОМ», 2009.

[Финн, 2010] Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть I // Искусственный интеллект и принятие решений. 2010. № 3.

[Andrews, 2011] Andrews S. In-Close2, a High Performance Formal Concept Miner // In: Andrews, S., Polovina, S., Hill, R., Akhgar, B. (eds.) ICCS-ConceptStruct 2011. LNCS, vol. 6828. Springer, Heidelberg.

[Dhillon, 2001] Dhillon I.S. Co-clustering documents and words using Bipartite Spectral Graph Partitioning // Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), 2001.

[Flach, 2012] Flach P. Machine Learning. The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, 2012.

[Kirchberg, 2012] Kirchberg M., Leonardi E., Tan Y.S., Link S., Ko R.K.L., Lee B.S. Formal Concept Discovery in Semantic Web Data // Proc. 10th International Conference, ICFCA 2012, Leuven, Belgium, May 7–10, 2012.

[Krajca, 2010] Krajca P., Outrata J., Vychodil V. Advances in algorithms based on CbO // In Proceedings of the 8th International Conference on Concept Lattices and Their Applications (CLA), 2010, vol. 672.

[Norris, 1978] Norris E.M. An Algorithm for Computing the Maximal Rectangles in a Binary Relation // Revue Roumaine de Mathématiques Pures et Appliquées, No. 23(2). 1978.

УДК 519.17

КЛАССИФИКАЦИЯ ЗАДАЧ ОПРЕДЕЛЕНИЯ СХОДСТВА ОРГРАФОВ

В.В. Кохов (*viktor-kokhov@ya.ru*)

Национальный исследовательский университет –
Московский энергетический институт (ТУ), Москва

Сформулированы формализованные постановки обобщенных задач определения максимального общего фрагмента и сходства двух орграфов. Приведена классификация задач определения максимального общего фрагмента двух орграфов и на ее основе предложена классификация задач определения сходства-различия орграфов. Указан универсальный метод решения анализируемых задач.

Введение

Развитие методов искусственного интеллекта актуализировало развитие методов структурного анализа в связи с расширением спектра структурных форм представления знаний и развитием таких прикладных областей, как структурное распознавание образов. При анализе сходства становится важным не только сравнение топологий структур, но и комплексный учет класса структур, весов вершин и ребер/дуг, ограничений на взаиморасположение фрагментов и др. От классических мер структурного сходства переходят к рассмотрению более общих отношений толерантности и подобия. Структурное сходство графовых моделей систем (ГМС) является центральной проблемой в интеллектуальном анализе данных, реализации правдоподобных рассуждений, обработке высказываний на естественных языках и других областях искусственного интеллекта. Определение сходства ГМС необходимо в организации поиска в базах структурных данных (семантический *web*-поиск электронных документов в Интернете) и знаний, представленных семантическими сетями. Это определяет актуальность и значимость разработки моделей, методов и программных средств для определения сходства структурированных нечисловых объектов (графов, орграфов, мультиграфов, семантических сетей и пр.) [Кузнецов и др., 1996]. Определение меры сходства позволяет перейти к наи-

более практически значимым задачам кластеризации множества ГМС, нечеткого поиска в базе ГМС и др. В работе [Kokhov, 2012] выделены два основных подхода к определению структурного сходства ГМС: подструктурно-метрический (ПМ-подход); структурно-характеристический (СХ-подход). В прикладном аспекте характеристизация многообразия видов сходства необходима для решения центральных проблем химической структурной информатики, связанных с установлением корреляций вида «структура-активность» и «структура-свойство» [Johnson et al., 1987], и разработки новых поколений ИПССИ и СИИ с правдоподобными рассуждениями [Финн, 2009]. С целью разработки наиболее эффективных методов определения сходства, ориентированных на учет специфики структур ГМС, ниже предложена классификация задач определения сходства орграфов на основе классификации задач определения максимальных общих фрагментов орграфов. Впервые выделены 72 вида базовых задач определения сходства.

1. Классическая задача определения сходства орграфов и ПМ-подход к ее решению

Пусть задан орграф $G=(V,E)$ с числом вершин $p=|V|$ и дуг $q=|E|$. Порожденный подграф орграфа получается при удалении одной или нескольких вершин и инцидентных к ним дуг. Если из орграфа удалять одну или несколько дуг, то в результате построим подграф (ниже фрагмент) орграфа. Два орграфа $G_1=(V_1,E_1)$ и $G_2=(V_2,E_2)$ изоморфны ($G_1 \approx G_2$), если $\exists \varphi: (V_1 \leftrightarrow V_2) \wedge (v_i, v_j \in V_1 [(v_i, v_j) \in E_1 \leftrightarrow (\varphi(v_i), \varphi(v_j)) \in E_2])$, где $\varphi(v_i), \varphi(v_j) \in V_2$. Орграф $G_1=(V_1,E_1)$ изоморфно вкладывается в орграф $G_2=(V_2,E_2)$ как порожденный подграф $G_1 \subseteq^S G_2$ (фрагмент, $G_1 \subseteq G_2$), если в орграфе G_2 есть порожденный подграф (фрагмент) $G^*=(V^*,E^*)$, для которого выполняется условие $G^* \approx G_1$. Под максимальным общим фрагментом (МСФ) (порожденным подграфом, МС) двух орграфов G_1, G_2 понимаем орграф G^* , для которого справедливы условия: а) $G^* \subseteq G_1$ и $G^* \subseteq G_2$ ($G^* \subseteq^S G_1$ и $G^* \subseteq^S G_2$); б) не существует большего G^* по числу дуг (вершин) фрагмента (подграфа) в орграфе G_1 , для которого выполняется условие а).

Основным инструментом при решении задачи сходства-различия орграфов (расположения фрагментов в орграфе) является использование инвариантов орграфа. Обозначим множество всех

орграфов через \mathfrak{R} . Пусть R – отношение «быть изоморфными» орграфами, а Q – непустое множество с отношением эквивалентности τ (множество чисел, векторов, матриц, орграфов и т.д.). Функция IN , заданная на множестве \mathfrak{R} и принимающая значения в Q , называется *инвариантом орграфа* G , если справедливо условие $\forall G_i, G_j \in \mathfrak{R} [G_i(R)G_j \rightarrow IN(G_i)(\tau)IN(G_j)]$.

Графы G_i, G_j называются IN -эквивалентными $G_i(IN)G_j$, если $IN(G_i)(\tau)IN(G_j)$. Инвариант IN – *полный инвариант орграфа*, если выполняется условие $\forall G_i, G_j \in \mathfrak{R} [IN(G_i)(\tau)IN(G_j) \rightarrow G_i(R)G_j]$.

Будем говорить, что сформулирована *классическая задача определения сходства орграфов*, если заданы следующие параметры:

- $\mathfrak{R} = \{G_1, G_2, \dots, G_p, \dots, G_n\}$ – заданное множество орграфов;
- $D(G_i, G_j)$ – метрика (псевдометрика) или функция для вычисления расстояния между орграфами G_i и G_j ;
- $SI(G_i, G_j)$ – функция для вычисления значения индекса сходства G_i, G_j .

Необходимо построить *матрицу (граф) попарных расстояний между парами орграфов* или матрицу коэффициентов (граф) попарного сходства анализируемых орграфов.

Таким образом, задача определения сходства на множестве орграфов \mathfrak{R} определена, если задана тройка параметров: $SIM = \langle \mathfrak{R}, D, SI \rangle$.

Под результатом решения задачи определения сходства набора из n орграфов будем понимать полный граф с n вершинами, каждое ребро которого $\{v_i, v_j\}$ взвешено значением расстояния $D(G_i, G_j)$ или индекса сходства $SI(G_i, G_j)$. При использовании ПМ-подхода в качестве метрик и индексов сходства используются следующие величины [Bunke et al., 1998, Raymond et al., 2002]:

1. $D_1(G_1, G_2) = p(G_1) + p(G_2) - 2p(MCS(G_1, G_2))$;
2. $D_2(G_1, G_2) = p(G_1) + q(G_1) + p(G_2) + q(G_2) - 2(p(MCF(G_1, G_2)) + q(MCF(G_1, G_2)))$;
3. $SI_1(G_1, G_2) = p(MCS(G_1, G_2))^2 / (p(G_1)p(G_2))$;
4. $DSI_1(G_1, G_2) = 1 - SI_1(G_1, G_2)$;
5. $SI_2(G_1, G_2) = \frac{(p(MCF(G_1, G_2)) + q(MCF(G_1, G_2)))^2}{(p(G_1) + q(G_1))(p(G_2) + q(G_2))}$;
6. $DSI_2(G_1, G_2) = 1 - SI_2(G_1, G_2)$.

Диапазон изменения и разнообразие значений попарных расстояний существенно влияют на возможность решения задачи кластеризации анализируемых орграфов и на результаты кластеризации. Чем шире разнообразие значений, тем больше возможностей для решения задачи кластеризации и получения результатов, которые интересуют исследователя.

Определение. Чувствительностью решения задачи определения сходства пар из N орграфов по инварианту IN относительно MCS по графу попарного сходства называется отношение вида

$$\alpha(IN(SI), MCS) = KL(IN(SI)) / (N(N-1)/2),$$

где $KL(IN(SI))$ обозначает число классов IN -эквивалентности пар орграфов по значениям SI .

IN -эквивалентность определяется совпадением значений расстояний (индексов сходства) для пар анализируемых на сходство-различие орграфов. Аналогично вводятся определения относительно MCF .

На рис. 1 приведены диаграммы орграфов G_1-G_5 , граф значений попарных расстояний (D_2) и граф значений попарного сходства (SI_2). На рис. 2 приведены результаты этапов кластеризации орграфов на основе значений SI_2 при последовательном удалении ребер из графа попарного сходства, вес которых менее границ сходства 68%, 79%, 90%.

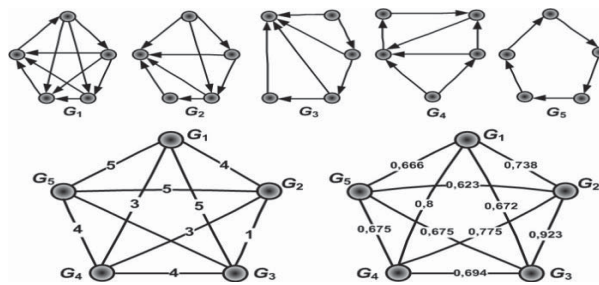


Рис.1



Рис.2

В результате кластеризации получили $\alpha(IN(SI_2), MCF) = 1$, а значит, получили точное решение задачи различения анализируемых орграфов.

2. Классификация задач определения максимальных общих фрагментов

Приведем формализованную постановку *двух базовых задач* изоморфного пересечения орграфов в форме задач распознавания свойств теории вычислительной сложности задач.

Задача 1. Наибольший общий сильносвязный подграф двух орграфов.

УСЛОВИЕ. Заданы два орграфа G_1, G_2 и положительное целое число K .

ВОПРОС. Существуют ли $G_1^* \subseteq^S G_1, G_2^* \subseteq^S G_2$ такие, что они являются *сильносвязными подграфами*, соответственно в G_1, G_2 и

$$|V(G_1^*)| = |V(G_2^*)| > K \text{ и } G_1^* \approx G_2^*?$$

Задача 2. Наибольший общий сильносвязный фрагмент двух орграфов.

УСЛОВИЕ. Заданы два орграфа G_1, G_2 и положительное целое число K .

ВОПРОС. Существуют ли $G_1^* \subseteq^f G_1, G_2^* \subseteq^f G_2$ такие, что они являются *сильносвязными фрагментами*, соответственно в G_1, G_2 и

$$|E(G_1^*)| = |E(G_2^*)| > K \text{ и } G_1^* \approx G_2^*?$$

Обе задачи принадлежат классу NP [Гэри М. и др., 1982], а задача поиска MCF в классе деревьев принадлежит к классу P [Горшков и др., 2004]. На основе анализа многообразия задач определения общего фрагмента, имеющих теоретическую или прикладную значимость, выделим параметры (табл. 1) для постановки обобщенной задачи $CF (CS)$ определения общего фрагмента (подграфа) для орграфов G_1 и G_2 . Задача $CF (CS)$ задана, если заданы параметры

$$\langle G_1, G_2; \sigma, Part, Neig, View, Con \rangle \quad (2.1)$$

и необходимо определить набор (базис) орграфов $B=(b_1, b_2, \dots, b_N)$, относительно которого орграфы G_1, G_2 σ^B -эквивалентны.

Таблица 1

№	Параметр	Смысл параметра	Виды значений параметра
1	σ	Отношение σ -эквивалентности	$\sigma^{B^1}, \sigma^{B^2}, \dots, \sigma^{B^N}, R$
2	<i>Part</i>	Вид соотношения частей орграфов, которыми являются общие фрагменты	<i>S-S; S-Fr; Fr-S; Fr-Fr</i>
3	<i>Neig</i>	Окрестность от числа вершин (дуг) максимального σ^B -эквивалентного подграфа (фрагмента)	0, 1, 2; 0, 1, 2
4	<i>View</i>	Вид результата по количеству общих фрагментов	Одно любое – 1; все неизоморфные – (<i>ALL</i>)
5	<i>Con</i>	Вид результата по виду связности	Слабосвязный (<i>w</i>); одностороннесвязный (<i>o</i>); сильносвязный (<i>s</i>)

В качестве разнообразия критериев Δ рассмотрим: *Cr1* – максимальное число вершин общего подграфа, *Cr2* – максимальное число дуг общего фрагмента. Для вида соотношения частей *S-S; S-Fr; Fr-S* принимаем *Cr1*, а для *Fr-Fr* – *Cr2*.

Часто возникает необходимость поиска решений в заданной окрестности Δ от максимального, то есть $Neig1=Cr1-\Delta$; $Neig2=Cr2-\Delta$, где Δ – целое число. Решение таких задач нашло широкое применение в химической информатике. Ниже в качестве отношения σ -эквивалентности орграфов будем рассматривать отношение *R*.

В соответствии с обобщенной формулировкой задачи (2.1) выделим 4 классификационных разреза: 1) вид соотношения общих фрагментов как частей орграфов G_1, G_2 ; 2) величину окрестности Δ ($\Delta=0$ – точное решение; $\Delta>0$ – решение в заданной окрестности); 3) вид результата по числу решений (одно решение; все неизоморфные решения); 4) вид общего фрагмента по виду связности (слабосвязный, одностороннесвязный, сильносвязный). В результате выделим 72 задачи пересечения орграфов. С целью сокращения изложения в табл. 2 представлены содержание и обозначение задач пересечения только для вида *Fr-Fr*.

Таблица 2

К.Н	Обозначение	Содержание задачи
1.1	$G_1^{Fr} \cap^{Fr}_{01w} G_2$	CF окрестности 0, один, слабосвязный
1.2	$G_1^{Fr} \cap^{Fr}_{11w} G_2$	CF окрестности 1, один, слабосвязный
1.3	$G_1^{Fr} \cap^{Fr}_{21w} G_2$	CF окрестности 2, один, слабосвязный
1.4	$G_1^{Fr} \cap^{Fr}_{0ALLw} G_2$	Все неизом. CF окрестности 0, слабосвязные
1.5	$G_1^{Fr} \cap^{Fr}_{1ALLw} G_2$	Все неизом. CF окрестности 1, слабосвязные
1.6	$G_1^{Fr} \cap^{Fr}_{2ALLw} G_2$	Все неизом. CF окрестности 2, слабосвязные
1.7	$G_1^{Fr} \cap^{Fr}_{01s} G_2$	CF окрестности 1, один, одностороннесвязный
1.8	$G_1^{Fr} \cap^{Fr}_{11s} G_2$	CF окрестности 1, один, одностороннесвязный
1.9	$G_1^{Fr} \cap^{Fr}_{21s} G_2$	CF окрестности 2, один, одностороннесвязный
1.10	$G_1^{Fr} \cap^{Fr}_{0ALLs} G_2$	Все неизом. CF окрестности 0, одностороннесвяз.
1.11	$G_1^{Fr} \cap^{Fr}_{1ALLs} G_2$	Все неизом. CF окрестности 1, одностороннесвяз.
1.12	$G_1^{Fr} \cap^{Fr}_{2ALLs} G_2$	Все неизом. CF окрестности 2, одностороннесвяз.
1.13	$G_1^{Fr} \cap^{Fr}_{01s} G_2$	CF окрестности 0, один, сильносвязный
1.14	$G_1^{Fr} \cap^{Fr}_{11s} G_2$	CF окрестности 1, один, сильносвязный
1.15	$G_1^{Fr} \cap^{Fr}_{21s} G_2$	CF окрестности 2, один, сильносвязный
1.16	$G_1^{Fr} \cap^{Fr}_{0ALLs} G_2$	Все неизом. CF окрестности 0, сильносвязные
1.17	$G_1^{Fr} \cap^{Fr}_{1ALLs} G_2$	Все неизом. CF окрестности 1, сильносвязные
1.18	$G_1^{Fr} \cap^{Fr}_{2ALLs} G_2$	Все неизом. CF окрестности 2, сильносвязные

3. Классификация задач определения сходства двух орграфов при использовании ПМ-подхода

На основе классификации задач определения CF (CS) двух орграфов получаем обобщенную постановку задачи определения сходства двух орграфов. Задача $SIM(G_1, G_2)$ при $\sigma=R$ и $View=1$ задана, если заданы следующие параметры $SIM(G_1, G_2) = \langle G_1, G_2; Part, Neig, Con \rangle$ и необходимо определить расстояние D или коэффициент сходства SI . Всего выделим 72 вида задач определения сходства.

На рис. 3 приведены примеры визуализации с помощью АСНИ «Мастерская граф-моделей» результатов решения трех видов задач поиска MCS для двух орграфов: 1) как слабосвязный подграф (A); 2) как одностороннесвязный подграф (B); 3) как сильносвязный подграф (C).

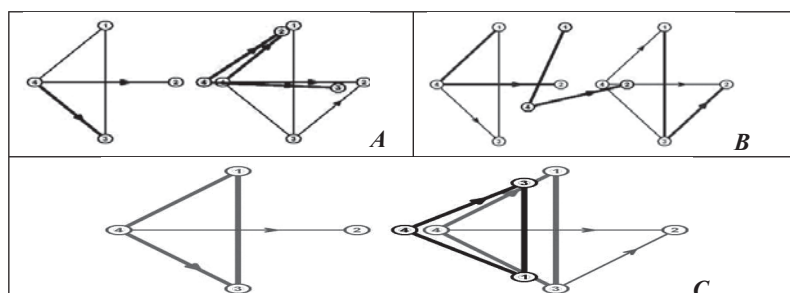


Рис. 3

Примеры содержания и обозначения задач для видов соотношения частей общих фрагментов $Fr-Fr$ и $S-S$ при $Neig=0$ приведены в табл. 3.

Таблица 3

$K.N$	Обозначение	Содержание задач
1.1	$D_2 G_1^{Fr \cap Fr} G_2^{0w}$	Расстояние по слабосвязному CF окрестности 0
1.2	$SI_2 G_1^{Fr \cap Fr} G_2^{0w}$	Коэффициент сходства по слабосвязному CF окрестности 0
1.3	$D_2 G_1^{Fr \cap Fr} G_2^{0o}$	Расстояние по одностороннесвязному CF окрестности 0
1.4	$SI_2 G_1^{Fr \cap Fr} G_2^{0o}$	Коэфф. сх-ва по одностороннесвязному CF окрестности 0
1.5	$D_2 G_1^{Fr \cap Fr} G_2^{0s}$	Расстояние по сильносвязному CF окрестности 0
1.6	$SI_2 G_1^{Fr \cap Fr} G_2^{0s}$	Коэфф. сх-ва по сильносвязному CF окрестности 0
2.1	$D_1 G_1^{S \cap S} G_2^{0w}$	Расстояние по слабосвязному CS окрестности 0
2.2	$SI_1 G_1^{S \cap S} G_2^{0w}$	Коэфф. сх-ва по слабосвязному CS окрестности 0

Продолжение таблицы 3

2.3	$D_1 G_1^s \cap_{00}^s G_2$	Расстояние по одностороннесвязному CS окрестности 0
2.4	$SI_1 G_1^s \cap_{00}^s G_2$	Коэфф. сх-ва по одностороннесвязному CS окрестности 0
2.5	$D_1 G_1^s \cap_{0s}^s G_2$	Расстояние по сильносвязному CS окрестности 0
2.6	$SI_1 G_1^s \cap_{0s}^s G_2$	Коэфф. сх-ва по сильносвязному CS окрестности 0

Заключение

В работе предложена классификация задач определения максимальных общих фрагментов двух орграфов и на ее основе классификация задач определения сходства двух орграфов. Для решения всех выделенных 72 видов задач определения сходства создан универсальный метод, являющийся расширением метода монотонных расширений частичных решений [Нечепуренко и др., 1990] на класс орграфов. Методы решения базовых классов задач определения сходства программно реализованы в АСНИ «Мастерская граф-моделей» [Кохов и др., 2006].

Предложенная классификация задач определения MCF (MCS) поможет создавать базу фактов, учитывая более точные свойства объекта исследования и на основе учета особенностей MCF (MCS) уменьшить количество порождаемых гипотез с повышением эффективности работы ДСМ-метода [Финн, 2009].

СПИСОК ЛИТЕРАТУРЫ

[Горшков и др., 2004] Горшков С.А., Кохов В.А. Эффективность подструктурного подхода к анализу сходства древовидных структур // Труды десятой ежегодной Международной научно-технической конференции студентов и аспирантов «РАДИОЭЛЕКТРОНИКА, ЭЛЕКТРОТЕХНИКА И ЭНЕРГЕТИКА». – М.: МЭИ(ТУ), 2004. Т. 1. С. 315–316.

[Гэри М. и др., 1982] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи: Пер. с англ. М.: Мир. 1982. 416 с.

[Ибрахим, 2009] Ибрахим А.Р. Разработка методов и программных средств для анализа сходства ациклических структур: Дис. ... канд. техн. наук. – М.: МЭИ, 2009.

[Кохов и др., 2006] Кохов В.А., Незнанов А.А., Ткаченко С.В. Программный комплекс для формирования и исследования отношений эквивалентности и толерантности на структурах. Десятая Национальная конференция по искусственному интеллекту с международным участием. КИИ-2006: Труды конференции. В 3-х т. Том 1. – М.: Физматлит, 2006. С. 199–207.

[Кузнецов и др., 1996] Кузнецов С.О., Финн В.К. О модели обучения и классификации, основанной на операции сходства // Обозрение прикладной и промышленной математики 3, №1, 1996.

[Нечепуренко и др., 1990] Нечепуренко М.И., Попков В.К., Кохов В.А. и др. Алгоритмы и программы решения задач на графах и сетях. – Новосибирск: Наука, 1990. 515 с.

[Финн, 2009] Финн В.К. Автоматическое порождение гипотез в интеллектуальных системах. – М.: Едиториал УРСС. 2009. 528 с.

[Bunke et al., 1998] Bunke H., Sharer K. A Graph Distance Metric Based on the Maximum Common Subgraph. // Pattern Recognition Letters, vol. 19, no. 3-4, 1998, pp. 255–259.

[Johnson et al., 1987] Johnson M. A., Nicholson V., Naim M., and Tsai C. A Comparison of the Substructure Metric to some Fragment-Based Measures of Inter-Molecular Structural Similarity, QSAR in Drug Design and Toxicology, D. Hadzi and B. Jerman-Balzac (ed.), Elsevier, Amsterdam, 1987, pp. 67–69.

[Kokhov, 2012] Kokhov V.A. Two Approaches to Determining Similarity of Two Digraphs. // Journal of Computer and Systems Sciences International, 2012, Vol. 51, pp. 695–714.

[Raymond et al., 2002] Raymond J. W., Gardiner E. J., Willett P. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. // Computer Journal, vol. 45, №.6, 2002. P. 631–644.

УДК 004.89:303.7

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ПРОБЛЕМЫ ПОДДЕРЖКИ СОЦИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ¹

М.А. Михеенкова (*mmikh@viniti.ru*)

В.К. Финн (*finn@viniti.ru*)

Всероссийский институт научной и технической информации РАН,
Москва

В работе рассматриваются проблемы развития компьютерных интеллектуальных систем, являющихся инструментом исследования в социальных науках. Человеко-машинные системы такого рода реализуют формализованные эвристики, воспроизводящие когнитивные рассуждения на основе анализа баз фактов, формирование которых обеспечивается предварительной содержательной работой социолога.

Введение

Метафорический характер наименования «Искусственный интеллект» (ИИ) требует от специалистов в этой области рефлексии относительно понимания основных задач, объединяющих широкий спектр исследований в единое научное направление. Это составляет предмет рассмотрения как довольно давних работ [Поспелов и др., 1991], так и относительно современных [McCarthy, 2007]. В последней формулируется проблема «очеловечивания» систем ИИ: от реализации методов ИИ в компьютерных системах к конструктивной аппроксимации человеческого мышления с включением имитации психологических особенностей (интуиции, эмоций, ассоциаций). Указанные полюса исследований характеризуются в [Boden, 2006] как «технологический» и «психологический» ИИ, соответственно. Однако наиболее плодотворным представляется движение в сторону создания систем, имитирующих рациональные аспекты естественного интеллекта (ЕИ) [Финн, 2009]. При

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14–07–00856а) и президиума РАН (программа П15, проект № 209).

современном уровне развития наук о человеке и обществе «очеловеченные» системы (human-level AI) – дело не самого близкого будущего.

1. Интеллектуальные системы как инструмент имитации познавательной деятельности

Компьютерные интеллектуальные системы (ИС) – основной конечный продукт ИИ [Арский и др., 2008] – объединяют технологические достижения с формальными средствами воспроизведения познавательной деятельности, являясь инструментом организации научных исследований. Потребность в такого рода системах особенно выражена в областях, где в значительной степени отсутствует формальный аппарат, преобладают идеи, а не точно сформулированные понятия, процедуры формирования теорий носят эвристический характер. К таким наукам относится, в частности, современная социология, накопившая обширные массивы неколичественных (качественных) данных, которые нуждаются в структурировании и систематизации с последующим преобразованием накопленного материала в знания – формированием теорий.

В архитектуре систем, воспроизводящих доступную феноменологию рационального сознания с точки зрения ИИ [Финн, 2009], отображается естественное сопоставление соответствующих компонентов. Системе знаний ЕИ соответствует Информационная среда ИС – подсистема накопления знаний, включающая базу фактов БФ и базу знаний БЗ. Продуктивному мышлению отвечает реализующий познавательные процедуры Решатель задач, субъективный мир личности (СМЛ) находит свое отражение с помощью развитых инструментов восприятия и отображения результатов – Интеллектуального интерфейса.

Функциональные возможности Решателя настроены на извлечение знаний из БФ, автоматическое порождение гипотез и объяснение имеющихся фактов на основании порожденных гипотез, а также на осуществление дедуктивного вывода из исходных и полученных знаний, БЗ. Фактически это означает возможность формального воспроизведения исследовательских эвристик действующих социологов – процедур выявления причинных зависимостей из анализа имеющихся данных, индуктивных стратегий создания теорий и связанных с этим процессов абдуктивного объяснения.

Это способствует уточнению не всегда ясно определяемой – ввиду отсутствия аппарата – структуры познавательного процесса в науках о человеке и обществе. Поскольку ценность выявленных причинных зависимостей проверяется возможностью объяснения рассматриваемого явления (в том числе и с учетом имеющегося теоретического знания) и дальнейшего предвидения возможных эффектов, можно говорить об уточнении познавательной активности в виде схемы «анализ данных – предсказание – объяснение» [Финн, 2009], для реализации которой требуется формализация различного рода познавательных процедур.

1.1. Интеллектуальный анализ социологических данных

Указанная схема есть не что иное, как прямое следование парадигме «понимающей социологии» М. Вебера, «...стремящейся, истолковывая, понять социальное действие и тем самым каузально объяснить его процесс и воздействие» [Вебер, 2006; с. 453]. Однако анализ личностного повседневного опыта и социальных взаимодействий индивидуумов неосуществим широко распространенными статистическими инструментами, имеющими дело с обезличенным субъектом и преимущественно количественными данными. Обнаружение нового (относительно имеющихся БФ и БЗ) знания, извлечение интерпретируемых зависимостей между различными факторами, неявно содержащимися в массивах эмпирических социологических данных, прогнозирование новых эффектов возможны лишь с использованием современных информационных технологий – ИС на основе методов ИИ.

Для перехода от феноменологии к системе знаний в науках со слаборазвитым формальным аппаратом требуется решение классических задач ИИ: представления данных и знаний и автоматизации аналитических процедур извлечения нового знания из эмпирического материала. Это означает создание формального языка с дескриптивными и аргументативными функциями для первичной структуризации данных и знаний (с возможностью определения их сходства), формирования системы отношений и представления процедур анализа, допускающих фальсификацию и возможную верификацию результатов.

Построение теории на основе эмпирических фактов представляет собой так называемую «восходящую» (индуктивную) стратегию качественного анализа социологических данных (в то время

как для количественного (статистического) анализа характерна «нисходящая» стратегия проверки априорных теорий). Отсутствие формальных инструментов рядом специалистов рассматривается как неотъемлемая характеристика такого анализа, позволяющая сохранять связь с «живой жизнью». Это, однако, делает полученные выводы уязвимыми с точки зрения обоснованности и объективности и заставляет других исследователей искать возможности для преодоления этих трудностей.

Наиболее широко распространенным инструментом формализованного качественного анализа является система fsQCA, реализующая качественный сравнительный анализ (Qualitative Comparative Analysis) американского социолога Ч. Рейджина [Rihoux et al., 2009]. QCA использует булеву алгебру для реализации фундаментального принципа качественного анализа «сходство фактов влечет наличие (отсутствие) изучаемого эффекта и его повторяемость»; расширенные версии включают элементы алгебры нечетких множеств и многозначных логик. Система предназначена для выявления интерпретируемых причинно-следственных связей конъюнктивно-дизъюнктивного типа, которые могут служить основой для построения теорий социологом. Таким образом, формальные средства и реализующие их компьютерные системы здесь реализуют лишь первую часть познавательного цикла: предсказание и объяснение существенным образом зависят от опыта, теоретического фундамента и профессиональной интуиции исследователя.

1.2. Интеллектуальная система типа ДСМ для поддержки социологических исследований

Полноценная реализация когнитивной схемы познавательной деятельности осуществляется в интеллектуальных системах типа ДСМ (ИС-ДСМ), обеспечивающих построение теории на основе эмпирических фактов средствами формализованных познавательных процедур. Составная часть Решателя интеллектуальных ДСМ-систем – Рассуждатель – реализует процедуры (и их комбинации – стратегии) ДСМ-метода автоматического порождения гипотез [Финн, 2013], формализующие соответствующие эвристики анализа данных. На основе этого универсального ядра создаются ИС-ДСМ для различных предметных областей, в их числе – система JSM-Socio [Михеенкова и др., 2013] для формализованного каче-

ственного анализа социологических данных. Особенности предметной области отражаются в принципах формирования информационной среды и особенностях пользовательского интерфейса, учитывающего потребности эксперта-социолога.

JSM-Socio предназначена для реализации исследовательской программы «понимающей социологии»: изучения индивидуально-поведения и порождения его детерминант с последующей типологизацией социума на их основе, анализа и прогнозирования мнений (как варианта поведения), распознавания рациональности мнений (в т.ч. степени рациональности мнений данной социальной общности) и отклонений от нее. Существенным при решении этих задач является выяснение влияния ситуации на поведение индивидуума.

Успешность выполнения этой программы зависит от соответствия имеющихся эмпирических данных существенным онтологическим допущениям ДСМ-метода. Это, прежде всего, детерминистский характер рассматриваемых социальных явлений и открытость знаний о предметной области. Заметим, что причинная обусловленность здесь понимается как предрасположенность к совершению поведенческих актов (действий, установок, мнений) при отсутствии противодействующих влияний. Поскольку гипотетические причины наблюдаемых эффектов представляются в виде структурного (не метрического и не статистического) сходства фактов, необходима его формализуемость. Принципиальны также наличие в БФ позитивных (+)- и негативных (-)-примеров (\pm)-фактов) изучаемого эффекта поведения и предполагаемая (\pm)-квазисимметрия причин (существенных влияний). Выполнение последнего условия позволяет автоматически порождать фальсификаторы порожденных гипотез и может рассматриваться как основание для абдуктивного принятия индуктивных гипотез о причинах.

Реализация описанной выше схемы познавательного процесса обеспечивается в ДСМ-методе синтезом познавательных процедур. Анализ БФ осуществляется формализованными процедурами эмпирической *индукции*, опирающимися на формальные расширения и уточнения индуктивных методов английского мыслителя Д.С. Милля. Предсказание – перенос порожденных гипотез о причинах на случаи с неопределенным исходом – представлено выводом на основе *структурной аналогии*. Объяснение же начальных данных

средствами *абдуктивного вывода* Ч.С. Пирса означает принятие гипотез на достаточном основании.

1.3. Стратегии анализа социологических данных

В ДСМ-языке для представления субъектов поведения вводятся индивидуальные переменные X, Z, V, \dots 1-го сорта (быть может, с нижними индексами) и константы C, C_1, C_2, \dots , являющиеся значениями переменных для объектов и подобъектов X, Z, V и т.д. Свойства объектов (например, эффекты поведения субъектов) представим индивидуальными переменными 2-го сорта Y, U, W, \dots (быть может, с нижними индексами) и константами Q, Q_1, Q_2 . Для параметров ситуации рассмотрим переменные $S, S_1, \dots, S_n, \dots$ и константы $\bar{S}, \bar{S}_1, \dots, \bar{S}_n$ 3-го сорта. Объекты (субъекты поведения) $X \in 2^{U^{(1)}}$, множества свойств (эффекты поведения) $Y \in 2^{U^{(2)}}$, ситуации $S \in 2^{U^{(3)}}$. Здесь $U^{(i)}$ – конечные множества, $\mathcal{B}_i = \langle 2^{U^{(i)}}, \emptyset, U^{(i)}, \text{---}, \cap, \cup \rangle$ – булевы алгебры, соответственно, $i = 1, 2, 3$. $U^{(1)} = \{d_1, \dots, d_n\}$ – множество характеристик социальных субъектов (описание социального характера, индивидуальных и биографических особенностей личности), $U^{(2)} = \{a_1, \dots, a_{r_2}\}$ – множество эффектов их поведения (действий и установок), $U^{(3)} = \{s_1, \dots, s_{r_3}\}$ – внешние (ситуационные) факторы. Мнение как специфический вид поведения формируется на основе оценки респондентом утверждений p_1, \dots, p_n , которые раскрывают содержание темы опроса T^* [Finn et al., 2011] и аргументируют отношение к теме. Функция оценки $v[p_i]$ ($i = 1, \dots, n$) принимает значения $v \in \{1, -1, 0, \tau\}$ («фактическая истина», «фактическая ложь», «фактическое противоречие», «неопределенность», соответственно), являющиеся типами истинностных значений бесконечнозначной логики ДСМ-метода. Пусть $?J_v p_i$ – терм, который понимается как вопрос «верно ли, что $v[p_i] = v?$ ». $J_v p = t$, если $v[p_i] = v$; в противном случае $J_v p = f$; t и f – истинностные значения двузначной логики «истина» и «ложь», соответственно. Мнением индивида назовем максимальную конъюнкцию $\Phi_j \equiv J_{n_1^{(j)}} p_1 \& \dots \& J_{n_n^{(j)}} p_n$, где $v_i^{(j)} \in \{\pm 1, 0, \tau\}$, $i = 1, \dots, n$; $j = 1, \dots, 4^n$.

ДСМ-рассуждение, формализованное средствами бесконечнозначной логики предикатов (1-го порядка для конечных моделей и слабой логики предикатов 2-го порядка для бесконечных моделей), состоит из последовательного и итерируемого применения индуктивных выводов и выводов по аналогии, повторяющегося до стабилизации множества гипотез. В исходной базе фактов ИС-

ДСМ представлены предикаты $X \Rightarrow_1 Y$ «субъект X демонстрирует эффекты поведения Y ». С помощью индуктивного поиска сходства исходных примеров из предиката $X \Rightarrow_1 Y$ порождаются предикаты $V \Rightarrow_2 W$ (прямой ДСМ-метод, анализ «от причины – к следствию») или $W \Leftarrow_3 V$ (обратный ДСМ-метод, «от следствия – к причине»), т.е. в автоматическом режиме формируются фрагменты базы знаний интеллектуальной системы. Производные предикаты $V \Rightarrow_2 W$ и $W \Leftarrow_3 V$ означают, что «подмножество характеристик V есть причина эффекта поведения W » и «эффект поведения W есть следствие подмножества характеристик V ». В выводе по аналогии порожденные гипотезы $V \Rightarrow_2 W$ и $W \Leftarrow_3 V$ о причинах наличия/отсутствия изучаемых эффектов используются для расширения и уточнения начального состояния БФ. ДСМ-рассуждение завершается применением абдукции – процедуры объяснения исходного множества фактов на основании так называемой «аксиомы каузальной полноты» (АКП) [Финн, 2013]. АКП требует, чтобы каждый пример из БФ был объяснен порожденными гипотезами. Если гипотез недостаточно для объяснения, БФ может быть пополнена новыми примерами, и ДСМ-рассуждение выполняется заново.

Полная категориальная характеристика социального поведения задается термом \bar{X} , где \bar{X} – полный объект $\bar{X} = \langle X, S, [\varphi] \rangle$, X – субъект поведения, S – контекст (ситуация) поведения, φ – мнение субъекта (о ситуации, о возможном поведении), $[\varphi]$ – множество элементов соответствующей максимальной конъюнкции. Пусть в исходном состоянии БФ представлена предикатами $\bar{X} \Rightarrow_1 Y$ ($\langle X, S, [\varphi] \rangle \Rightarrow_1 Y$), где Y – эффект (действие или установка к действию). Тогда применение прямого ДСМ-метода порождает гипотезы о причинах вида $\bar{V} \Rightarrow_2 Y$, где \bar{V} – сходство объектов \bar{X} , $\bar{V} = \bar{X}_1 \cap \dots \cap \bar{X}_k$, $\bar{V} = \langle V, S', [\psi] \rangle$ ($V = X_1 \cap \dots \cap X_k$, $S' \subseteq S_1 \cap \dots \cap S_k$, $[\psi] \subseteq [\varphi_1] \cap \dots \cap [\varphi_k]$). База фактов для анализа такого вида поведения как мнение представлена предикатами $\langle X, S \rangle \Rightarrow_1 \langle [\varphi], T \rangle$, где T – одна из возможных тем опроса (в случае проведения опроса по одной теме представление упрощается до $\langle X, S \rangle \Rightarrow_1 [\varphi]$). В этом случае применяется обратный метод и порождаются гипотезы вида $W \Leftarrow_3 \langle V, S' \rangle$, $W/[\psi]$, $V = X_1 \cap \dots \cap X_k$, $S' \subseteq S_1 \cap \dots \cap S_k$, $[\psi] = [\varphi_1] \cap \dots \cap [\varphi_k]$.

Оба подхода были применены для ситуационного анализа трудовых отношений на российских предприятиях [Климова и др., 2012] средствами ИС JSM-Socio. Изучались такие виды поведения,

как готовность либо неготовность добиваться справедливости в случае нарушения трудовых прав, мнения относительно ситуации на предприятиях, а также стабильность и лояльность по отношению к своему предприятию.

ИС-ДСМ является, в частности, инструментом поддержки закрытого социологического опроса и реализации предсказательного опроса. Формально m -значный (когда респонденту предлагается $m \geq 2$ вариантов ответа) предсказательный социологический опрос задается как $O_m = \langle J_m, P, \Sigma, K, R, X \Rightarrow_1 Y \rangle$. Логическими средствами его формализации являются m -значные логики J_m и исчисление эквивалентных формул ИЭФ- J_m [Finn et al., 2011]. Множество оценок переменных $V_m = \{0, \frac{1}{m-1}, \dots, \frac{m-2}{m-1}, 1\}$ должно быть социологически интерпретируемым. R соответствует множеству респондентов, участвующих в опросе, $|R| = r$, $K = \{\phi_1, \dots, \phi_r\}$ – множеству мнений. Непротиворечивое множество $\Sigma = \{\psi_1, \dots, \psi_s\}$ формул ψ_1, \dots, ψ_s логики J_m выражает заложенные исследователем логические зависимости между элементами p_1, \dots, p_n каркаса P , которые в идеале должны распознаваться рационально мыслящим респондентом.

2. Проблемы развития интеллектуальных систем поддержки социологических исследований

Формализованные эвристики типа «анализ данных – предсказание – объяснение» и их конкретизация в виде схемы правдоподобных рассуждений «индукция – аналогия – абдукция» применительно к анализу социологических данных нуждаются в объективизации. При анализе мнений это может достигаться уже на уровне представления данных с использованием логики аргументации – когда оценка отношения к высказываниям p_1, \dots, p_n формируется на основе выбора аргументов «за» и «против» из заданного внешним образом множества аргументов A [Финн, 2012]. Понимание рациональности как обобщенной целеценностной рациональности, представляющей аргументированное принятие решений, позволяет сформулировать процедурный подход к ее анализу на основе логических критериев непротиворечивости [Finn et al., 2011].

Развитием и усилением идеи объективизации результатов ДСМ-рассуждений является поиск эмпирических закономерностей [Финн, 2010]: выделение инвариантного ядра, т.е. множества гипотез, генерируемых в последовательностях расширяющихся БФ

в неизменном виде. Если все гипотезы удовлетворяют этому условию, можно говорить о выявлении эмпирических законов, если же доля таких гипотез не менее 0,8, речь идет об эмпирических тенденциях. Это может свидетельствовать о выявлении реальных причинных зависимостей в рассматриваемой области. Разумеется, важнейшую роль при этом играют выбор адекватного языка представления данных и знаний и непротиворечивость и полнота исходной концептуальной модели.

Для выявления эмпирических закономерностей существенным оказывается выбор стратегий, наиболее адекватных эмпирической ситуации исследования. Напомним, что для формализации индукции и порождения гипотез о возможных причинах наличия изучаемых свойств определяются предикаты $M_{a,n}^+(V, W)$ и $M_{a,n}^-(V, W)$ (параметр n отображает число применений правил правдоподобного вывода к БФ), представляющие индуктивный метод сходства – необходимую компоненту аналогов всех индуктивных методов Д.С. Милля, формализованных в ДСМ-методе. Подформулы предикатов содержат экзистенциальное условие (существование k (+)-фактов вида $J_{(1,n)}(X_i \Rightarrow_1 Y_i), i = 1, \dots, k, J_{(1,n)}\Psi = \bigvee_{j=1}^n J_{(1,j)}\Psi$); условие сходства (+)-фактов $(\bigcap_{i=1}^k X_i = V) \& (V \neq \emptyset)$; условие эмпирической зависимости V и W и условие исчерпываемости сходных (+)-фактов $\forall X \forall Y ((J_{(1,n)}(X \Rightarrow_1 Y) \& (V \subset X)) \rightarrow ((W \subseteq Y) \& (\bigvee_{i=1}^k (X = X_i))))$; условие нижней границы числа сходных (+)-фактов $k \geq 2$ (в практических целях в ИС она может быть увеличена). Аналогично определяется $M_{a,n}^-(V, W)$ для порождения кандидатов в гипотезы о причинах (-)-фактов. В результате применения правил правдоподобного вывода 1-го рода $(I_n^+, (I_n^-, (I_n^0$ и $(I_n^t$, в посылки которых входят $M_{a,n}^+(V, W) \& \neg M_{a,n}^-(V, W), \neg M_{a,n}^+(V, W) \& M_{a,n}^-(V, W), M_{a,n}^+(V, W) \& M_{a,n}^-(V, W)$ и $\neg M_{a,n}^+(V, W) \& \neg M_{a,n}^-(V, W)$, порождаются, соответственно, гипотезы $J_{(1, n+1)}(V \Rightarrow_2 W), J_{(\leftarrow, n+1)}(V \Rightarrow_2 W), J_{(0, n+1)}(V \Rightarrow_2 W)$ и $J_{(\tau, n+1)}(V \Rightarrow_2 W)$.

Предикаты сходства могут быть усилены дополнительными условиями, в том числе позволяющими формализовать и другие индуктивные методы Д.С. Милля (см. [Финн, 2013]). Пусть I^+ – множество имен (индексов) усиления $M_{a,n}^+(V, W)$, I^- – множество имен усиления $M_{a,n}^-(V, W)$, $x \subseteq I^+$ и $a^+ \in x$, а $y \subseteq I^-$ и $a^- \in y$. Тогда стратегией ДСМ-метода $\text{Str}_{x,y}$ будем называть множества правил

$(I)_n^\sigma$ ($\sigma \in \{+, -, 0, \tau\}$) таких, что они образованы возможными комбинациями предикатов $M_{x,n}^+(V, W)$ и $M_{y,n}^-(V, W)$ (например, $M_{x,n}^+(V, W) \& M_{y,n}^-(V, W)$ для $(I)_n^+$). В [Финн, 2014] подробно рассматриваются дистрибутивные решетки элементарных и неэлементарных (включающих условия миллевских индуктивных методов) ДСМ-стратегий, являющиеся спецификацией общего подхода к формированию стратегий; реализация некоторых стратегий в ИС-ДСМ представлена в [Волкова, 2011].

Подобные усиления формулируются также и в случае расширенного представления данных с включением параметра ситуации. Рассмотрим усиление ситуационного метода сходства, представляющее модификацию метода различия (аналогично [Финн, 2013]). В исходной БФ ИС-ДСМ влияние ситуации отображается тернарным предикатом $P(X, Y, S)$ – «субъект X демонстрирует эффекты поведения Y в ситуации S ». Для индуктивного порождения причин $R_i(\langle V, S_0 \rangle, W)$ – «пара \langle подмножество характеристик субъекта V и характеристики ситуации $S_0 \rangle$ есть причина эффектов поведения W » – используются минимальные по выразительной силе предикаты (ситуационного) сходства ${}_i M_{a,n}^\sigma(V, W, S_0)$ ($\sigma = +, -$), входящие в посылки индуктивных правил правдоподобного вывода 1-го рода. Основой эмпирических зависимостей в этих предикатах является сходство не только субъектов поведения и их мнений, но и ситуации проявления этих мнений. Ситуационный метод различия может учитывать различие в объектах, и тогда формализуется посредством предиката ${}_i M_{d_{0x},n}^+(V, W, S_0) = {}_i M_{a,n}^+(V, W, S_0) \& (d_{0x}^+)$, где

$$d_{0x}^+ = \forall X \forall Y \forall S \forall Z \forall U ((J_{(1,n)} P(X, Y, S) \& (W \subseteq Y) \& (V \subset X) \& (S_0 \subseteq S) \& ((X \setminus V) \subset Z) \& ((X \setminus V) \neq \emptyset \& \neg (V \subset Z))) \rightarrow (\neg J_{(1,n)} P(Z, U, S) \vee \neg (W \subseteq U))).$$

Различие в ситуациях описывается предикатом ${}_i M_{d_{0s},n}^+(V, W, S_0) = {}_i M_{a,n}^+(V, W, S_0) \& (d_{0s}^+)$, где $d_{0s}^+ = \forall X \forall Y \forall S \forall S' \forall U ((J_{(1,n)} P(X, Y, S) \& (W \subseteq Y) \& (V \subset X) \& (S_0 \subseteq S) \& ((S \setminus S_0) \subset S') \& ((S \setminus S_0) \neq \emptyset \& \neg (S_0 \subset S'))) \rightarrow (\neg J_{(1,n)} P(X, U, S') \vee \neg (W \subseteq U))).$

Таким образом, повышение объективности исследований достигается сочетанием внешних усилий по корректному формированию БФ и применением внутренних возможностей формирования стратегий ДСМ-анализа в следующих версиях JSM-Socio.

Заключение

Рефлексия относительно познавательной деятельности с неизбежностью приводит к осознанию необходимости реализации логик рассуждений и аргументации в ИС для представления и формирования систем знаний, в особенности в науках о человеке и обществе. Интеллектуальные системы, будучи партнерскими человеко-машинными системами, оказываются в этом случае полноценным инструментом исследования. Плодотворное взаимодействие ИС-ДСМ с исследователем требует высокопрофессиональной деятельности по использованию стандартных эвристик формирования массива данных. Но и выбор стратегии ДСМ-рассуждений, формирование БФ и БЗ для достижения абдуктивной сходимости (увеличения доли объясненных фактов), выявление эмпирических тенденций невозможны без экспертной содержательной работы социолога. Включение в ИС новых процедур для распознавания закономерностей в расширяющихся базах фактов способствует построению эмпирических теорий, расширению баз знаний и приданию научной объективности исследованиям в социальных областях с помощью современных информационных технологий.

СПИСОК ЛИТЕРАТУРЫ

- [Арский и др., 2008] Арский Ю.М., Финн В.К. Принципы конструирования интеллектуальных систем // Информационные технологии и вычислительные системы. 2008. № 4.
- [Вебер, 2006] Вебер М. Избранное. – М.: РОССПЭН, 2006.
- [Волкова, 2011] Волкова А.Ю. Алгоритмизация процедур ДСМ-метода автоматического порождения гипотез // НТИ. Сер. 2. 2011. № 5.
- [Климова и др., 2012] Климова С.Г., Михеенкова М.А. Формальные средства ситуационного анализа: опыт применения // НТИ, сер. 2. 2012. № 10.
- [Михеенкова и др., 2013] Михеенкова М.А., Волкова А.Ю. Спецификация интеллектуальной системы типа ДСМ // НТИ, сер. 2. 2013. № 7.
- [Поспелов и др., 1991] Поспелов Д.А., Левитин К.Е. (редакторы-составители) Будущее искусственного интеллекта. – М.: Наука, 1991.
- [Финн, 2009] Финн В.К. К структурной когнитологии: феноменология сознания с точки зрения искусственного интеллекта // Вопросы философии. 2009. № 1.

[Финн, 2010] Финн В.К. Об определении эмпирических закономерностей посредством ДСМ-метода автоматического порождения гипотез // Искусственный интеллект и принятие решений. 2010. № 4.

[Финн, 2012] Финн В.К. Тезисы об аргументационных системах // НТИ, сер. 2. 2012. № 9.

[Финн, 2013] Финн В.К. Эпистемологические основания ДСМ-метода автоматического порождения гипотез // НТИ, сер. 2. 2013. № 9; № 12 (Ч. I и II).

[Финн, 2014] Финн В.К. Дистрибутивные решетки индуктивных процедур // (в печати).

[Boden, 2006] Boden M. A. *Mind as Machine. A History of Cognitive Science.* – Oxford: Clarendon press, 2006.

[Finn et al., 2011] Finn V. K., Mikheyenkova M. A. *Plausible Reasoning for the Problems of Cognitive Sociology* // *Logic and Logical Philosophy.* 2011. Vol. 20.

[McCarthy, 2007] McCarthy J. *From here to human-level AI* // *Artificial Intelligence.* 2007. Vol. 171.

[Rihoux et al., 2009] Rihoux B., Ragin C.C. (eds). *Configurational Comparative Methods. Qualitative Comparative Analysis (QCA) and related techniques (Applied Social Research Methods).* – CA and London: SAGE Publications, 2009.

ИТОГИ И ПЕРСПЕКТИВЫ РАЗВИТИЯ МОДЕЛИ КЛАССИФИКАЦИИ НА ОСНОВЕ ХОРОШИХ ДИАГНОСТИЧЕСКИХ ТЕСТОВ (Часть 1)

К.А. Найденова (*ksennaidd@gmail.com*)
Военно-медицинская академия, Санкт-Петербург

Дается ретроспективный обзор работ в направлении развития модели классификации и распознавания образов на основе хороших классификационных тестов. Рассматриваются некоторые перспективы развития этого направления.

Введение

Развитие методов логического анализа данных и распознавания образов на основе хороших диагностических тестов началось с открытия Н.Г.Болдырева, профессора ЛЭТИ им. Ульянова (Ленина), что распознающую систему можно рассматривать как алгебраическую решетку. В работе [Boldyrev, 1974] предлагалась формализация распознающей системы как алгебры с двумя бинарными операциями утончения и обобщения (генерализации), определяемыми с помощью системы аксиом, включающей аксиомы решетки. Коротко опишем формальную систему, предложенную Н.Г. Болдыревым.

Пусть P – полный ансамбль паттернов (в нашей терминологии это множество свойств). Две бинарные операции $a * b$ и $a \bullet b$ определяются для любой пары подмножеств ансамбля P . Предположим, что результаты этих операций есть либо подмножества полного ансамбля паттернов, либо равны одному из символов ω и α , показывающих невыполнимость этих операций. Первая операция, обозначаемая $*$, называется операцией генерализации (G -операция) и вторая операция, обозначаемая \bullet , называется операцией утончения (R -операция).

Пусть 2^P есть множество всех подмножеств множества P . Тогда эти операции отображают каждую пару $a, b \subseteq P$ во множество 2^P . Если пара (a, b) не имеет образа в отображении, определяемом операцией R , тогда мы полагаем, что результат R -операции равен

символу α . Это значит, что не существует объекта, который обладает свойствами a и b одновременно. Аналогично, если пара (a, b) не имеет образа в отображении, определяемом G -операцией, тогда мы полагаем, что результат G -операции равен символу ω . Смысл символа ω есть: « a не похоже на b ». Эти операции определяются следующей системой постулатов [Boldyrev, 1974].

1. Постулаты утончения:

if $a \bullet b \neq \alpha$, $a \bullet c \neq \alpha$, и $b \bullet c \neq \alpha$, then $(a \bullet b) \bullet c \neq \alpha$ и $a \bullet (b \bullet c) \neq \alpha$.

2. Постулаты решетки:

$$(a \bullet b) \bullet c = a \bullet (b \bullet c); (a * b) * c = a * (b * c);$$

$$a \bullet b = b \bullet a; a * b = b * a;$$

$$a*(a \bullet b) = a; a \bullet (a * b) = a.$$

3. Постулаты включения:

$$a \bullet b \subseteq a; a \subseteq a * b;$$

$$a \bullet b \subseteq b; b \subseteq a * b.$$

Нетрудно доказать, что результат операции генерализации $c = a * b$ есть наименьший элемент в решетке, удовлетворяющий отношениям $c \bullet a = a$, $b \bullet c = b$. Аналогично, результат операции утончения $c = a \bullet b$ есть наибольший элемент в решетке, удовлетворяющий отношениям $c * a = a$, $c * b = b$.

К работам по применению алгебраических решеток для распознавания и классификации образов с 1974 года подключились некоторые аспиранты Н.Г.Болдырева, в частности Т.Чебоксарова, Г.Корнева, Б.Градобойнов и автор этой статьи. В основном развивалась модель распознающей системы как булевой решетки и были построены алгоритмы распознавания с применением четверичных кодов, которые сводились к новым методам минимизации Булевых функций. Другое направление было связано с распознаванием образов, описываемых целочисленными (символьными) кодами. Задача, сформулированная Болдыревым, была следующей: повторить с помощью программы на компьютере процесс классификации природных объектов, который осуществляет человек. В качестве природных объектов выступали индикатрисы рассеяния света, а в качестве естественной классификации – научная классификация индикатрис, построенная физиком О.Д. Бартеновой [Бартенева и др., 1960]. В рамках этого задания было естественным познакомиться с книгой [Piaget et al., 1959] по генезису классификацион-

ных структур. Исследования способности к формальной классификации у детей дошкольного и раннего школьного возраста привели Пиаже и его сотрудников к открытию, что эта способность связана с формированием когнитивных структур в виде взаимосвязанных алгебраических решеток на объектах, классах объектов и свойствах объектов с операциями объединения и пересечения.

В семидесятые годы прошлого века сложилась традиция отдельно рассматривать задачи, связанные с классификацией и распознаванием, такие как выделение понятий или генерализованное описание заданных классов объектов, диагностика классов объектов, поиск наиболее информативных признаков (минимизация признаков). Отдельно рассматривались процессы собственно распознавания, которые часто сводились к построению решающих деревьев на признаках. Отдельно от распознавания образов рассматривалась и задача кластеризации. Рассмотрение первых двух задач в рамках теории решеток привело к следующим результатам.

1. Три эквивалентных определения диагностического теста

1. Была построена интерпретация распознающей системы как модулярной решетки в целочисленном пространстве признаков. Пусть $A = \{a_1, a_2, \dots, a_k\}$ конечное множество целых чисел и пусть \leq есть отношение совпадения. Это отношение выполняется для пары элементов $a_i, a_j, i \neq j, a_i, a_j \in A$, если и только если $a_i = a_j$ и обладает всеми свойствами частичной упорядоченности. Дополним множество A двумя символами ω и α такими, что $\forall a_i \in A, a_i < \omega, \alpha < a_1$. Тогда множество $A' = \{A, \omega, \alpha\}$ с отношением \leq образует полную решетку $L = (A', \leq)$. L также задается с помощью бинарных операций $*$ и \bullet , определяющих $\sup(a_i, a_j)$ и $\inf(a_i, a_j)$, соответственно, для всех $a_i, a_j \in A$, как показано в таблицах 1 и 2.

Таблица 1

*	a_1	a_2	...	a_k	ω	α
a_1	a_1	ω	ω	ω	ω	a_1
a_2		a_2	ω	ω	ω	a_2
..				ω	ω	...
a_k				a_k	ω	a_k
ω					ω	ω
α						α

Определим решетку L^n на множестве $A^n = A_1' \times A_2' \times \dots \times A_n'$ как прямое произведение решеток $L^i = (A_i', \leq)$, $i = 1, 2, \dots, n$. Операции $\{*, \bullet\}$ в L^n есть покомпонентные операции, а именно, условие $r_1 \leq r_2$, $r_1, r_2 \in A^n$ выполняется, если и только если условие $x_j[r_1] \leq x_j[r_2]$, $j = 1, 2, \dots, n$ выполняется для каждой пары соответствующих компонент. $L^n = (A^n, \leq)$ является полной модулярной решеткой. Единичный элемент в этой решетке есть n -компонентный кортеж, все элементы которого равны ω , и нулевой элемент в этой решетке есть n -компонентный кортеж, все элементы которого равны α .

Таблица 2

\bullet	a_1	a_2	a_k	ω	α
a_1	a_1	α	α	α	a_1	α
a_2		a_2	α	α	a_2	α
..				α	...	α
a_k				a_k	a_k	α
ω					ω	α
α						α

Элементы в множестве A^n можно рассматривать как описания объектов, определяемых посредством атрибутов (признаков) A_1, A_2, \dots, A_n . Отношение $r_1 \leq r_2$, $r_1, r_2 \in A^n$ может быть прочитано как выражение « r_1 обладает признаком r_2 ». Тогда операция $*$ для пары $r_1, r_2 \in A^n$ образует описание ближайшего общего признака $r_3 \in A^n$, такого, что $r_1, r_2 \leq r_3$, и для любого $r_4 \in A^n$, $r_4 \neq r_3$ и $r_1, r_2 \leq r_4$ мы имеем $r_3 < r_4$. Операция \bullet для $r_1, r_2 \in A^n$ образует описание ближайшего элемента r_5 , обладающего как признаком r_1 , так и признаком r_2 , то есть $r_5 \leq r_1, r_2$ и для любого $r_6, r_6 \in A^n$, $r_6 \neq r_5$ такого, что $r_6 \leq r_1, r_2$, мы имеем $r_6 < r_5$. Единичный элемент в решетке имеет семантику «никакой не признак», и нулевой элемент решетки имеет семантику «никакой не объект». Естественно отображать все кортежи, содержащие, по крайней мере, один символ α в нулевой элемент α в L^n (в n -компонентный кортеж, все элементы которого равны α).

2. В основу разработанного алгоритма классификации был положен математический аппарат теории решеток. Объекты исходного множества являются свободными образующими полурешетки, порождаемой с помощью операции обобщения. Алгоритм классификации заключается в нахождении упорядоченных цепочек признаков и соответствующих им упорядоченных по отноше-

нию включения подмножеств объектов, обладающих этими признаками. На алгебраическом языке алгоритм строит упорядоченные по вложению цепочки главных (principal) идеалов (фильтров) в решетке. Признак, оканчивающий цепочку, называется признаком класса объектов. Процесс построения цепочки заканчивается при получении признака, в котором число значений удовлетворяет заданным ограничениям. Затем строится следующая цепочка признаков и образуется следующий класс объектов. Этот алгоритм представлял собой по сути построение решетки с выбором признаков по некоторому формальному критерию (число значений) и неформальному предпочтению экспериментатора. Однако важно, что признак здесь двойственный объект и классификация есть процесс построения решетки.

Описанный алгоритм был применен для классификации (без учителя) природных объектов – индикатрис рассеяния света, полученных с помощью нефелометра ИФ-14 в разных географических районах (данные О.Д. Бартеневой [Бартенева и др., 1960]), и спектров яркости слоистообразных облаков нижнего яруса, полученных с помощью спектрометра СПИ-2, установленного на самолете-лаборатории ГТО ИЛ-18 (данные Л.И. Чапурского).

Несмотря на простоту алгоритма, были получены неплохие результаты [Найденова и др., 1978]. При классификации индикатрис рассеяния света сначала проводилось целочисленное кодирование измерений. Описание индикатрисы включало 18 атрибутов. При автоматической классификации было образовано 4 класса индикатрис. Полученные группы объединяли индикатрисы одного и того же класса по классификации Бартеневой. Встречались лишь единичные попадания индикатрис не в свой класс. Во всех экспериментах на 20-25 разных кривых наблюдалось образование от 4 до 7 групп при числе значений в признаках от 3 до 4. Алгоритм позволил получить более точную и детальную классификацию по сравнению с той, которую выполняет человек.

Спектральные яркости облаков описывались 40 значениями яркости. Ограничения на число значений яркости в общих признаках задавались равными 10 или 15. В обоих случаях было выделено по 6 классов спектров с одинаковыми ядрами. Машинная классификация спектров яркости облачности в целом отражала физическую природу их формирования. Без участия человека, при отсутствии априорной информации было произведено разделение спектров

как по высотам Солнца, так и по оптическим толщинам облаков. В каждом классе преобладали облака с определенной микроструктурой, отличающиеся спектром распределения частиц по размерам, их счетной концентрацией, а также соотношением их жидких и твердых облачных элементов. Иными словами, классы облачности, полученные машинным алгоритмом, хорошо интерпретируются специалистом. Некоторые выделенные классы отражают пространственную структуру облачности, то есть особенности конвективных процессов в слоях облаков.

Эта работа показывает, что к 1978 году уже не только задача выделения и описания классов была сведена к порождению алгебраической решетки, но и алгоритм для решения этой задачи уже был создан и применялся на практике.

3. Одновременно был разработан алгоритм получения диагностических тестов на основе построения решетки разбиений заданного множества объектов с операциями умножения и объединения на разбиениях. Разбиения объектов по значениям заданных атрибутов рассматриваются как атомарные элементы строящейся решетки. Для построения классификационных тестов использовался тот же алгоритм построения упорядоченных цепочек главных идеалов (фильтров) в решетке, а в качестве управляющего параметра использовалось число различных пар объектов в порождаемых разбиениях. Хотя этот алгоритм не был опубликован, но он представлен в диссертации [Найденова, 1979], а также докладывался на конференциях и семинарах.

4. Все основные задачи классификации (порождение и описание классов объектов, диагностика и выделение подмножеств информативных признаков) решаются на основе одной и той же математической модели и сводятся к построению алгебраических решеток.

В работе [Найденова и др., 1979] предлагался двунаправленный алгоритм построения решеток с использованием одновременно двух операций решетки. Правила сокращения перебора с помощью запрещенных интервалов в решетке, сформулированные в этой статье, используются во всех задачах поиска тестов при декомпозиции на подзадачи. Изложение в статье идет в терминах решетки, а в прикладных задачах решетка может иметь различные интерпретации.

В работе [Найденова, 1982] устанавливается связь структуры функциональных зависимостей (ФЗ) в данных с алгебраической решеткой на разбиениях. Доказывается У т е р ж д е н и е 1: Пусть

$R(X)$ – отношение и таблица $Tn(X)$ – его реализация. Пусть $X' \subset X$ и $X'' \subset X$ совокупности атрибутов, такие что разбиения строк таблицы $Tn(X)$, порождаемые ими, находятся в отношении $P(X'') \subset P(X')$. Тогда совокупности X' и X'' в $Tn(X)$ находятся в ФЗ $X'' \rightarrow X'$. Если разбиения $P(X'')$ и $P(X')$ эквивалентны, то совокупности X' и X'' находятся во взаимно однозначной ФЗ $X'' \leftrightarrow X'$.

Для целочисленных таблиц вводятся два способа представления разбиений строк таблицы по значениям атрибутов и даются примеры выполнения операций решетки разбиений для этих двух способов представления разбиений.

Устанавливается эквивалентность задач поиска ФЗ и задач поиска диагностических тестов. Доказываются следующие теоремы.

Т е о р е м а 1. Пусть $R(X)$ – некоторое отношение и таблица $Tn(X)$ – его реализация. Пусть Q – множество строк таблицы. Следующие две задачи эквивалентны: 1) найти минимальную по мощности совокупность атрибутов $X' \subset X$ такую, что в таблице $Tn(X)$ имеет место $X' \leftrightarrow X$. 2) найти минимальную по мощности совокупность атрибутов $X' \subset X$, по значениям которой различаются все пары строк множества Q .

Т е о р е м а 2. Пусть $R(X)$ – некоторое отношение и таблица $Tn(X)$ – его реализация. Пусть Q – множество строк таблицы и $P(X'')$ – разбиение множества Q , порождаемое совокупностью атрибутов $X'' \subset X$. Следующие две задачи эквивалентны: 1) найти минимальную по мощности совокупность атрибутов $X' \subset X$ такую, что в таблице $Tn(X)$ имеет место функциональная зависимость $X' \rightarrow X''$. 2) найти минимальную по мощности совокупность атрибутов $X' \subset X$, различающую все пары строк, принадлежащие разным классам в разбиении $P(X'')$.

Доказаны также следующие утверждения. **У т в е р ж д е н и е 2.** Если совокупность атрибутов $X^* \subset X$ является минимальным диагностическим тестом в таблице $Tn(X)$, то X^* содержит только взаимно-функционально независимые атрибуты. **У т в е р ж д е н и е 3.** Пусть $R(X)$ – некоторое отношение и таблица $Tn(X)$ – его реализация и Q – множество строк таблицы. Пусть $P(X'')$ – разбиение множества Q , порождаемое совокупностью атрибутов $X'' \subset X$.

Если $X' \subset X$ минимальная по мощности совокупность атрибутов, различающая все пары строк таблицы, принадлежащих разным классам в разбиении $P(X'')$, то она является совокупностью функционально взаимно-независимых атрибутов.

Описываются процессы построения алгебраической решетки как в случае решетки разбиений, так и в случае решетки общих признаков объектов. Процесс построения решеток и в том и в другом случаях есть двойственный процесс. В первом случае поиск диагностических тестов совокупности строк Q заключается в одновременном построении двух цепочек: цепочки упорядоченных по включению совокупностей атрибутов $X^0 \subset X^1 \subset X^2 \subset X^3 \subset \dots \subseteq X^k$, $X^k \subset X$ и цепочки порождаемых ими разбиениями множества Q , т.е. $P(X^0) \subset P(X^1) \subset P(X^2) \subset P(X^3) \subset \dots \subseteq P(X^k)$. Во втором случае используется восходящий процесс с образованием цепочек признаков с помощью операции обобщения и соответствующих цепочек упорядоченных по отношению включения подмножеств объектов множества Q , обладающих этими признаками [Naidenova, 1984].

В результате была показана связь алгебраических решеток с основными классификационными задачами – диагностика и построение описаний классов в терминах признаков; этот подход дает право говорить об алгоритмической эквивалентности разных классификационных задач.

В статье [Найденова, 1982] также дана декомпозиция классификационных задач на элементарные подзадачи, в которых ищутся решения, удовлетворяющие заданным ограничениям одновременно по числу атрибутов в тестах и по числу пар различимых строк (объектов), принадлежащих одному или разным классам объектов. Различаются алгоритмы, которые дают: одно решение, упорядоченный набор решений, все решения, удовлетворяющие заданным ограничениям, оптимальные в некотором смысле решения (минимальный тест (по числу атрибутов), минимальное число классов и т.п.). В результате были даны три эквивалентных определения диагностического теста.

Определение 1. Набор $X \subseteq U$ атрибутов является диагностическим тестом для заданной классификации в таблице $T(U)$ примеров, если и только если одно из следующих эквивалентных условий выполняется:

$$\begin{aligned} & \mathbf{(e_1)} \quad \forall (i, j), i, j \in S, i \neq j \\ & t_i[K] \neq t_j[K] \rightarrow t_i[X] \neq t_j[X] \quad (t_i[X] = t_j[X] \rightarrow t_i[K] = t_j[K]); \\ & \mathbf{(e_2)} \quad X \rightarrow K; \\ & \mathbf{(e_3)} \quad P(X) \subseteq P(K) \quad (X \rightarrow K \equiv X * K = X \equiv X + K = K), \end{aligned}$$

где S – множество индексов рассматриваемых объектов.

Плодотворным был период работы с 1980 по 1985 г. в «Лес-проекте» (Министерства лесного хозяйства СССР), где небольшая группа математиков и программистов под рук. К Найденовой решала задачи автоматизации методов определения состава насаждений по признакам, регистрируемым с фотографий, а также задачи сегментации и анализа изображений. Решение этих задач позволило, с одной стороны, изучить структуры знаний специалистов-лесников и их методы рассуждений при описании и распознавании типов леса в лаборатории и природных условиях, а также прийти к понятию хорошего диагностического (классификационного) теста, наилучшим образом аппроксимирующего заданную классификацию объектов.

СПИСОК ЛИТЕРАТУРЫ

[Бартенева и др., 1960] Бартенева О.Д., Довгяло Е.Н., Полякова Е.А. Экспериментальные исследования оптических свойств приземного слоя атмосферы // Труды ГГО, 1960, вып. 220.

[Найденова, 1979] Найденова К.А. Автоматизация процессов классификации экспериментальных данных на основе теории алгебраических решеток // Автореферат на соискание ученой степени к.т.н. Ленинград. 1979.

[Найденова, 1982] Найденова К.А. Реляционная модель анализа экспериментальных данных // Известия АН СССР, серия «Техническая кибернетика». 1982. № 4.

[Найденова и др., 1978] Найденова К.А., Чапурский Л.И. Применение алгебраического подхода при автоматической классификации природных объектов // Проблемы физики атмосферы. 1978. Вып. 15.

[Найденова и др., 1979] Найденова К.А., Чебоксарова Т.Н. Стратегия двунаправленного поиска в решении прикладных задач // Семиотика и информатика. Вып. 12. – М.: ВИНТИ, 1979.

[Boldyrev, 1974] Boldyrev, N. G. 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. 1974.

[Naidenova, 1984] Naidenova, X.A. A Formal Model of Knowledge Interpretation on the Base of Classification Process // Proceeding of the IFAC Symposium of Artificial Intelligence, Ser. 9. Pergamon Press. 1984.

[Piaget et al., 1959] Piaget, J., and Inhelder, B.. La genèse des structures logiques élémentaires: Classifications et sériations. – Neuchâtel: Delachaux & Niestlé. 1959.

ИТОГИ И ПЕРСПЕКТИВЫ РАЗВИТИЯ МОДЕЛИ КЛАССИФИКАЦИИ НА ОСНОВЕ ХОРОШИХ ДИАГНОСТИЧЕСКИХ ТЕСТОВ (Часть 2)

К.А. Найденова (*ksennaidd@gmail.com*)
Военно-медицинская академия, Санкт-Петербург

Дается ретроспективный обзор работ в направлении развития модели классификации и распознавания образов на основе хороших классификационных тестов. Рассматриваются некоторые перспективы развития этого направления.

1. Понятие хорошего диагностического теста

Так как диагностический тест как набор атрибутов или значений атрибутов порождает разбиение объектов на непересекающиеся блоки, то встал вопрос, какие же тесты (в смысле разбиения) будут максимально близки к заданной классификации объектов. Определение хорошего теста и первый алгоритм его нахождения были даны в работе [Найденова и др., 1986] и позже на английском языке [Naidenova et al., 1991].

Пусть U есть множество атрибутов, описывающих рассматриваемые объекты, $X \subseteq U$ есть тест для заданного целевого атрибута K , и разбиения $P(X)$, $P(K)$ порождаются в таблице описаний объектов T по значениям X и K , соответственно. Обозначим через $Q(K)$ множество всех диагностических тестов для K : $Q(K) = \{X: X \subseteq U: P(X) \leq P(K)\}$.

Для определения хорошего теста мы используем зависимость на разбиениях: $P(X) \subseteq P(K)$ ($X \leq K \equiv X * K = X \equiv X + K = K$). Если $P(X) = P(K)$, тогда X есть идеальная аппроксимация классификации K .

Если для всех тестов X , $X \subseteq U$ имеет место зависимость $P(X) \subseteq P(K)$, то хороший тест будет такой X , что $P(X)$ есть ближайший к $P(K)$ элемент множества $Q(K)$, то есть для всех Y , $Y \subseteq U$ условие $(P(X) \subseteq P(Y) \subseteq P(K))$ влечет $P(X) = P(Y)$. Таким образом, мы приходим к следующему определению хорошего диагностического теста (ХТ).

Определение 1. Набор $X \subseteq U$ есть хороший тест или хорошая аппроксимация классификации K в T , если удовлетворяются следующие условия:

- а) $X \in Q(K)$;
 б) не существует такого набора атрибутов Z , $Z \subseteq U$, $X \neq Z$, что $Z \in Q(K)$ и $P(X) < P(Z) \leq P(K)$.

Введем понятие наилучшего диагностического теста следующим образом.

Определение 2. ХТ X , $X \subseteq U$ есть наилучший тест для заданной классификации K в T , если число классов в разбиении $P(X)$ есть наименьшее из возможных для всех тестов в $Q(K)$.

Введем следующую характеристику наборов атрибутов. Определим на множестве всех подмножеств атрибутов отношение конгруэнтности Θ следующим образом: $X \Theta Y$ если и только если $P(X) = P(Y)$ ($P(X) \subseteq P(Y)$, $P(Y) \subseteq P(X)$). Тогда удовлетворяются следующие зависимости на разбиениях $X^*Y = X$ and $X^*Y = Y$.

Определение 3. Набор атрибутов $X \subseteq U$ есть максимально избыточный набор, если для любого атрибута $A \notin X$, $A \in U$, наборы атрибутов AX и X принадлежат к разным классам эквивалентности по конгруэнтности, то есть $[XA] \Theta \neq [X] \Theta$ для всех $A \notin X$, $A \in U$.

Это означает, что если набор X есть максимально избыточный набор, то тогда добавление к X любого атрибута, ему не принадлежащего, приводит к изменению класса эквивалентности по конгруэнтности для полученного набора.

Определение 4. Набор X атрибутов называется хорошим максимально избыточным (ХМИТ) тестом для заданной классификации K , если он ХТ для K и, одновременно, он максимально избыточный набор атрибутов.

Если X есть ХМИТ для K , тогда после прибавления к нему любого атрибута $A \notin X$, $A \in U$ образуемый набор будет тестом для K , но он не будет ХТ.

Определение 5. Набор атрибутов X называется безыбыточным, если для всех Z , $Z \subset X$, $[X] \Theta \neq [Z] \Theta$.

Это означает, что если набор атрибутов X является безыбыточным, тогда удаление любого атрибута A , $A \in X$ из него приводит к изменению класса эквивалентности по конгруэнтности для образованного в результате удаления набора XA .

Определение 6. Набор X атрибутов есть хороший безыбыточный тест (ХБТ) для классификации K , если он ХТ для K и, одновременно, он также является безыбыточным набором атрибутов.

Особенностью первого алгоритма нахождения всех ХМИТ является то, что в нем нахождение этих тестов (в виде ФЗ) сводится

к нахождению хороших импликативных зависимостей для двух классов строк: первый («+» класс) содержит все такие подмножества атрибутов (точнее имен атрибутов), по значениям которых не различаются хотя бы два объекта, принадлежащие одному и тому же классу в заданном разбиении $P(K)$, а второй класс («-» класс) содержит все такие наборы имен атрибутов, по значениям которых не различаются хотя бы одна пара объектов, принадлежащих разным классам в разбиении $P(K)$. Тогда тест – это такой набор X первого множества (пусть M^+), который не включен ни в один набор второго множества и в силу этого свойства будет представлять собой левую часть импликативной зависимости $X \rightarrow M^+$. ХТ должен быть вложен в максимально возможное число наборов M^+ . Доказана Т е о р е м а 3 [Найденова и др., 1986; Найденова, 1992].

Т е о р е м а 3. ХМИТ для K либо принадлежит множеству M^+ , либо найдется такое число q , $2 \leq q \leq nt$, что этот тест будет равен пересечению точно q элементов M^+ , где nt есть размерность множества M^+ .

И первый алгоритм получения ХМИТ основан на этой теореме. Так как ХБТ (как и любой безызбыточный тест) принадлежит одному и только одному классу по конгруэнции Θ , он содержится в одном и только в одном ХМИТ. Это приводит к одному из возможных методов поиска ХБТ для заданной классификации на заданном множестве примеров:

- 1) найти все ХМИТ;
- 2) для ХМИТ, найти все ХБТ, содержащиеся в нем.

ХБТ можно получить непосредственно как элементы решетки разбиений, образующие которой есть атомарные разбиения, порождаемые заданными атрибутами. Эта идея заложена в алгоритме, предложенном в [Мегрецкая, 1989]. Статья [Найденова, 1992] содержит три эквивалентных определения теста, определение хорошего теста и его характеризацию. Кроме того, в ней дается подробное описание основного алгоритма для вывода функциональных и импликативных зависимостей, показывается связь диагностического подхода с основными направлениями машинного обучения.

В основном алгоритме ведущим процессом является порождение подмножеств индексов объектов, пересечение которых порождает тест. Перебор подмножеств ведется по уровням, содержащим все подмножества мощности k , $k = 1, nk$. Причем каждое подмножество мощности $k+1$ образуется тогда и только тогда, когда все его

собственные подмножества соответствуют тестам. Этот алгоритм был разработан независимо от алгоритма в [Agraval et al., 1994], где используется такой же метод перебора образуемых подмножеств.

Рассмотрим иллюстративный пример в таблице 3. ХМИТ для классификации $K: \{BG, DG, AE\}$. Безызбыточный, но не хороший тест для K : AB , так как $P(AB) = \{\{1,3\} \{2\} \{4\} \{5\} \{6\}\} \subset P(AE) = \{\{1,3,4\} \{2\} \{5\} \{6\}\}$.

ХМИТ для класса k_1 : $e_1 = \{1,3,4\}$, $b_1 = \{1,2,3\}$, $g_1 = \{1,2,4\}$. Отметим, что минимальные диагностические тесты необязательно будут хорошими. Например, минимальные тесты $c_2 = \{2,3\}$ и $d_1 = \{1,3\}$ не являются хорошими.

Таблица 1

T	A	B	C	D	E	F	G	K
t_1	a_1	b_1	c_1	d_1	e_1	f_1	g_1	k_1
t_2	a_2	b_1	c_2	d_2	e_2	f_2	g_1	k_1
t_3	a_1	b_1	c_2	d_1	e_1	f_3	g_2	k_1
t_4	a_1	b_2	c_3	d_2	e_1	f_1	g_1	k_1
t_5	a_2	b_2	c_3	d_2	e_1	f_1	g_2	k_2
t_6	a_3	b_3	c_3	d_3	e_3	f_1	g_3	k_2

В работе [Naidenova et al., 1995] используется декомпозиция главной задачи на атрибутивные подзадачи. В атрибутивной подзадаче порождаются тесты, которые включают заданный атрибут (или значение атрибута) и никакие другие тесты. Такая декомпозиция позволяет значительно сократить перебор за счет удалений уже использованных атрибутов (значений) и строк, которые после таких удалений более не содержат тестов. В работе [Naidenova, 2005] дан алгоритм DIAGARA, который использует декомпозицию главной задачи на объектные подзадачи. Каждая такая подзадача дает в результате тесты, вложенные в выбранную строку (описание объекта), и только такие тесты.

2. Хороший тест как двойственный объект

В конце 80-х годов прошлого века начал развиваться анализ формальных понятий (АФП), которому отводится главенствующая роль в решении задач классификации и анализа данных. ФКА, его развитию и применению посвящено много работ, в том числе

и отечественных ученых: [Ganter, 1984], [Ganter et al., 1991; 1999], [Wille, 1992], [Kuznetsov, 1993], [Kuznetsov et al., 2002; 2004]. Основными понятиями ФКА являются формальный контекст и формальное понятие. Формальное понятие в АФП есть тройка: $\mathbf{K} = (G, M, I)$, где G – множество объектов, M – множество признаков и $I \subseteq G \times M$ – отношение, показывающее, каким признаком обладает каждый объект. Для любого подмножества объектов и $A \subseteq G$ любого подмножества признаков $B \subseteq M$ определяются операторы Галуа: $A' = \{m \in M | \forall g \in A: (gIm)\}$ и $B' = \{g \in G | \forall m \in B: (gIm)\}$.

Формальное понятие есть пара (A, B) , такая, что $A \subseteq G, B \subseteq M, A' = B$, and $B' = A$. Множество A называется экстендом, а множество B называется интендом формального понятия $is(A, B)$. Для $g \in G$ и $m \in M$, $\{g\}'$ обозначается как g' и называется интендом объекта, и $\{m\}'$ обозначается как m' и называется экстендом атрибута.

Под влиянием АФП мы переопределили понятие ХДТ как двойственный объект [Найденова и др., 2001], в котором в явном виде объединялись бы две его составляющие – логический терм (правая часть ФЗ или импликации) и объекты или классы объектов, для которых выполняется условие, что логический терм является тестом. Оказалось, что можно дать диагностическому тесту определение, аналогичное формальному понятию только для тестов, которые образуют имплекативные зависимости между признаками. Кроме того, мы имели в виду многозначный контекст (табл. 3, например), в отличие от бинарного контекста, на котором определяется классическое формальное понятие.

Пусть $S = \{1, 2, \dots, N\}$ – множество индексов объектов (объектов, для краткости) и $T = \{m_1, m_2, \dots, m_j, \dots, m_m\}$ есть множество значений атрибутов (значений, для краткости). Каждый объект описывается набором значений из T .

Определения тестов основаны на соответствиях Галуа, то есть двух отображениях $S \rightarrow T, T \rightarrow S$. Пусть $s \subseteq S, t \subseteq T$. Обозначим через $t_i, t_i \subseteq T, i = 1, \dots, N$ описание объекта с индексом i . Определим отображения следующим образом:

$S \rightarrow T: t = \text{val}(s) = \{\cap \forall t_i: t_i \subseteq T, i \in s\}$ и $T \rightarrow S: s = \text{obj}(t) = \{i: i \in S, t \subseteq t_i\}$.

Операции $\text{val}(s)$, $\text{obj}(t)$ есть мыслительные операции, относящиеся к определению общего признака объектов, индексы которых принадлежат s , и к определению всех объектов, обладающих признаком t .

Пусть $R = R(+) \cup R(-)$ есть множество описаний положительных и отрицательных объектов, $S(+)$ и $S(-) = S \setminus S(+)$ есть множества индексов положительных и отрицательных объектов, соответственно.

A diagnostic test for $R(+)$ is a pair (s, t) such that $t \subseteq T$ ($A = \text{obj}(B) \neq \emptyset$), $s \subseteq S(+)$ and $t \not\subseteq \text{val}(g) \ \& \ t \neq \text{val}(g), \forall g, g \in R(-)$. Equivalently, $\text{obj}(t) \cap S(-) = \emptyset$.

В общем случае, множество t не замкнуто для диагностического теста (s, t) , то есть условие $\text{val}(\text{obj}(t)) = t$ не всегда выполняется, следовательно, диагностический тест не обязательно представляет собой формальное понятие. Условие замкнутости выполняется только для ХМИТ [Naidenova, 2012b].

Определение 7. Диагностический тест $(s, t), t \subseteq T$ ($s = \text{obj}(t) \neq \emptyset$) является хорошим тестом для $R(+)$, если и только если любое расширение $s' = s \cup i, i \notin s, i \in S(+)$ влечет, что $(s', \text{val}(s'))$ не является тестом для $S(+)$.

Определение 8. Хороший тест $(s, t), t \subseteq T$ ($s = \text{obj}(t) \neq \emptyset$) для $S(+)$ является безызбыточным, если любое его сужение $t' = t \setminus m, m \in t$ влечет, что $(\text{obj}(t'), t')$ не является тестом для $R(+)$.

Определение 9. Хороший тест для $S(+)$ является ХМИТ, если любое расширение $t' = t \cup m, m \notin t, m \in T$ влечет, что $(\text{obj}(t \cup A), t')$ не является хорошим тестом для $S(+)$.

3. Направления будущих исследований

Методы поиска хороших тестов отличаются от поиска всех формальных понятий или построения решетки Галуа над некоторым контекстом. При поиске только ХМИТ встает проблема определения зон поиска, в которых могут содержаться еще не найденные тесты, и отбрасывания таких зон, в которых заведомо новые тесты содержаться не могут. Для этого нами были введены понятия существенных объектов и значений. Разработка и анализ различных алгоритмов поиска ХМИТ позволили обнаружить, что эти алгоритмы представляют собой модели индуктивно-дедуктивных правдоподобных рассуждений. Порождаемые в процессе поиска импликации используются в дедуктивных правилах рассуждения для сокращения перебора, а для получения тестов используются индуктивные правила вывода, которые реализуют непосредственно модусы правдоподобных рассуждений Д.С. Милля (1872). К

этим правилам относятся индуктивные правила генерализации и спецификации, индуктивные диагностические правила, относящиеся как к объектам, так и к атрибутам и их значениям. Индуктивные правила рассуждений есть правила перехода от одного элемента в решетке к его ближайшим элементам по отношению порядка. Дедуктивные шаги при индуктивных рассуждениях состоят в выводе следствий из образованных импликаций и уже известных (найденных) хороших тестов (то есть знаний). Главные формы дедуктивных правил есть *modus ponens*, *modus tollens*, *modus ponendo tollens*, и *modus tollendo ponens*. [Найденова, 2009а; 2009b; 2006]. Новый алгоритм поиска ХМИТ-ов на базе индуктивных правил рассуждения, называемый NIAGARA, дан в [Naidenova, 2006].

Логический вывод при распознавании (с использованием полученных импликаций) новых объектов также основан на той же самой модели индуктивно-дедуктивных выводов. Дедуктивный механизм в этом процессе является ведущим, а индуктивный процесс помогает индукции, если для распознавания нужно расширить базу правил. При этом формируется локальный контекст, необходимый для разрешения трудной ситуации. Модель правдоподобных рассуждений при распознавании, близкая к естественным рассуждениям с помощью импликаций и индуктивных шагов вывода, описана в ряде работ, например, [Найденова, 2010; 2012а; 2012с].

Направление будущих исследований, естественно, связано с программной реализацией правдоподобных классификационных рассуждений, включающих как обучение (вывод правил), так и рассуждения при распознавании. Здесь важно отметить, что дедуктивные рассуждения реализуются алгоритмами поиска в алгебраической решетке теми же самыми, как и при поиске тестов, только при этом ведущей операцией будет операция утончения. Хотя мы до сих пор рассматривали задачи поиска хороших тестов, но получив ХМИТ и ХБТ быстрыми алгоритмами, нетрудно построить и полную решетку тестов или формальных контекстов для выделенных подконтекстов полного контекста.

Заключение

В статье дается ретроспективный обзор основных работ, связанных с теорией и применением ХДТ для распознавания образов, классификации и для синтеза правдоподобных рассуждений, что

связано в основном с будущими проектами интеллектуализации компьютеров.

СПИСОК ЛИТЕРАТУРЫ

[Мегрецкая, 1989] Мегрецкая И. А. Построение естественных классификационных тестов при создании баз знаний // Проблемы применения экспертных систем в народном хозяйстве. – Кишинев: Молдавия, Институт математики АН МССР. 1989.

[Найденова, 2009] Найденова К.А. Сведение одного класса алгоритмов машинного обучения к логическим операциям правдоподобных рассуждений // Известия Академии наук. Теория и системы управления. 2009. №3.

[Найденова, 2010] Найденова К.А. Организация правдоподобных рассуждений в интеллектуальных системах // Труды Всероссийской конференции «Управление знаниями и технологиями Semantic-Web». – СПб.: Университет информационных технологий, механики и оптики. 2010.

[Найденова, 2012a] Найденова К.А. Модель индуктивно-дедуктивных естественных рассуждений на основе классификационных рассуждений // Труды 13-й национальной конференции по искусственному интеллекту с международным участием. Том 1. – Белгород: БГТУ. 2012a.

[Найденова и др., 2001] Найденова К.А., Ермаков А.Е. Декомпозиция алгоритмов поиска хороших диагностических тестов // Труды 4-й Международной конференции «Computer-Aided design of Discrete Devices» (CAD DD'2001). Том 2. – Минск. 2001.

[Найденова и др., 1986] Найденова К.А., Полежаева Ю.Г. Алгоритм нахождения наилучших диагностических тестов // 4-ая Всесоюзная конференция «Применение методов мат. логики» / сб. Докладов конф. под ред. Минца Г.Е. и Лорентца П.П. – Таллин: Институт кибернетики АН ЭССР. 1986.

[Naidenova et al., 1991] Naidenova X.A., Polegaeva J. G. SIZIF – the System of Knowledge Acquisition from Experimental Facts // Industrial Application of Artificial Intelligence / Atly, I.L. and Mikulich, L.I. (Editors). 1991.

[Agrawal et al., 1994] Agrawal, R., & Srikant, R. Fast Algorithms for Mining Association Rules// Proceeding of the 20th VLDB Conference. – Santiago: Chile. 1994.

[Ganter, 1984] Ganter, B. Two basic algorithms in concepts analysis //FB4-Preprint, No. 831. – [Darmstadt: Technische Hochschule. 1984.

[Ganter et al., 1991] Ganter, B., & Reuter, K. Finding All Closed Sets: A General Approach // Order. 1991. 8.

[Ganter et al., 1999] Ganter B., Wille R. Formal Concept Analysis. Mathematical Foundations. – Berlin: Springer-Verlag. 1999.

[Kuznetsov, 1993] Kuznetsov, S.O. A Fast Algorithm for Computing All Intersections of Objects in a Finite Semi-lattice // *Automatic Documentation and Mathematical Linguistics*. 1993. 27(5).

[Kuznetsov et al., 2002] Kuznetsov, S.O., Ob'edkov, S.A. Comparing the performance of algorithms for generating concept lattices// *Journal of Experimental & Theoretical Artificial Intelligence*. 2002. 14 (2-3).

[Kuznetsov et al., 2004] Kuznetsov, S.O., Ob'edkov, S.A. Complexity of learning in context lattices from positive and negative examples // *Discrete Applied Mathematics*. 2004. 142.

[Mill, 1872] Mill, J. S. *The System of Logic Ratiocinative and Inductive Being a Connected View of the Principles of Evidence, and the Methods of Scientific Investigation*, Vol.1. – London, West Strand. 1972.

[Naidenova, 1992] Naidenova, X.A. Machine Learning as a Diagnostic Task // «*Knowledge-Dialog-Solution*» / *Materials of Short-term Scientific and Technical Seminar*, Arefiev, I. (ed.), – SPB. 1992.

[Naidenova, 2005] Naidenova, X.A. DIAGARA: An Incremental Algorithm for Inferring Implicative Rules from Examples // *International Journal «Information Theories & Applications*.2005. 12(2).

[Naidenova, 2006] Naidenova, X.A. An Incremental Learning Algorithm for Inferring Logical Rules from Examples in the Framework of the Common Reasoning Process // Triantaphyllou, E., & Felici, G. (Eds), *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*. – New York, USA: Springer. 2006.

[Naidenova, 2009b] Naidenova, X.A. Interconnecting a Class of Machine Learning Algorithms with Logical Commonsense Reasoning Operations. // R. Anbumani, & R. Nedunchezian (Eds), *Soft Computing Applications for Database Technologies: Techniques and Issues*, Chapter 12. – New York, USA: ISR. 2009b.

[Naidenova, 2012b] Naidenova, X.A. Good Diagnostic Tests as Formal Concepts // Domenach, D. I. Ignatov, and J. Poelmans (Eds.), *LNAI 7278*. – Heidelberg: Springer. 2012b.

[Naidenova, 2012c] Naidenova X.A. Classification reasoning as a model of human commonsense reasoning // *Workshop Notes of International Workshop «What can FCA do for AI?» (FCA4AI)* held at the European Conference on AI, (Editors: Sergei O. Kuznetsov, Amadeo Napoi, Sebastian Rudolph. P. – Montpellier, France. 2012c.

[Naidenova et al., 1995] Naidenova X.A., Shagalov, B.L., Plaksin M. V. Inductive Inferring All Good Classification Tests // *Proceedings of International Conference «Knowledge-Dialog-Solution» (KDS-95)*. Vol. 1. 1995.

[Wille, 1992] Wille, R. Concept lattices and knowledge systems // *Computer Mathematic Applied*. 1992. 23(6-9).

УДК 004.855

ЭВОЛЮЦИОННЫЙ МЕТОД ПОКРЫТИЙ ДЛЯ СОСТАВЛЕНИЯ БАЗЫ ФАКТОВ ДСМ-МЕТОДА¹

А.И. Панов (*pan@isa.ru*)

А.В. Швец (*shvets@isa.ru*)

Институт системного анализа РАН, Москва

В работе представлен метод предварительного обучения на данных большого объема для составления компактной базы фактов, используемой в задаче выявления причинно-следственных связей. Проводится сравнение методов, использующих классический алгоритм AQ-обучения и эволюционный алгоритм генерации покрытий. Оба алгоритма позволяют сокращать признаковое пространство, на котором проходит поиск гипотез ДСМ-методом. Приводятся результаты экспериментов на данных психологического тестирования.

Введение

В настоящее время в области автоматического анализа и извлечения данных высокую актуальность имеет задача выявления причинно-следственных (каузальных) связей. Каузальное отношение между явлениями (объектами и процессами) является одним из небольшого количества базисных отношений, используемых в моделях представления знаний как автоматизированных систем, так и человека [Осипов, 1990; Осипов, 1997]. Выделение причинно-следственных связей на данных из слабо формализованных предметных областей, для которых не созданы формальные модели, описывающие взаимосвязи объектов таких предметных областей, имеет большое практическое значение. Например, при анализе медицинских и психологических данных часто возникает задача прогнозирования - предсказания протекания болезни или поведения респондентов. Для составления таких прогнозов ключевую роль играют причинно-следственные связи.

Задача выявления каузальных отношений была поставлена достаточно давно и к настоящему времени накоплено большое количество методов ее решения. Все они могут быть разделены на две груп-

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 14-07-31194).

пы: статистические методы, которые применяются в подавляющем большинстве практических работ, и логические, которые призваны обеспечить большую правдоподобность выдвигаемых гипотез.

Обе группы методов обладают очевидными преимуществами и недостатками [Панов, 2013]. Основные недостатки статистических методов (корреляционного, факторного и др. типов анализа) – это требование к объему и репрезентативности выборки и неинтуитивный алгоритм решения задачи. В свою очередь логические методы, такие как ДСМ [Финн, 1991; Финн, 2000], подвержены проблеме полного перебора и трудно применимы для реального массива данных, возникающих в экспериментальных исследованиях. С другой стороны, основные преимущества логических методов: интуитивность, подобие алгоритма решения когнитивным процессам, протекающим у человека при решении той же задачи, слабые требования к свойствам выборки – имеют высокую ценность, что в свою очередь приводит к задаче включения статистических этапов в алгоритмы логических методов [Панов, 2010].

Один из подходов, направленных на сокращение пространства признаков, на котором осуществляется перебор при поиске гипотез о причинно-следственных связях, заключается в предварительной обработке данных. Такая обработка заключается в выборе из описания объектов признаков, наиболее важных с точки зрения отделения от других объектов. Именно на множестве этих признаков в дальнейшем и происходит поиск каузальных гипотез. В данной работе исследуется один подкласс задач, возникающих при анализе данных психологического тестирования, для которой предложен алгоритм AQ+JSM, основанный на логических методах, и показана корректность его работы [Панов, 2013]. Эффективность и работоспособность AQ+JSM были продемонстрированы на психологических и медицинских данных. Предлагается способ включения статистического этапа в данный алгоритм, проводится сравнение результатов работы этого этапа со стандартной логической (AQ-обучение [Michalski, 1973; Wojtusiak et al., 2006]) и с предлагаемой статистической (эволюционный метод покрытий) реализацией.

1. Постановка задачи

Исследуемый подкласс задач представляет собой поиск причинно-следственных связей для групп испытуемых среди результатов их психологического тестирования. При этом предполагает-

ся, что в рамках одного эксперимента (цикла тестирования) группы выделяются либо экспериментатором вручную, либо на основе разбиения целевой шкалы на интервалы. Например, возможно разбиение группы по полу (два класса), либо по уровню агрессивности (три класса: низкий, средний, высокий уровень агрессивности).

Так как в рамках рассматриваемой задачи нас интересуют причинно-следственные связи между свойствами группы в целом, а не между свойствами отдельных испытуемых, то для сокращения пространства признаков (базы фактов ДСМ-метода) в AQ+JSM проводится построение описания каждого класса.

Итак, в соответствии с [Панов, 2013] пусть имеется три множества: $O=\{o_i\}$ – множество из n объектов, которые представляют данные тестирования испытуемых, $C=\{c_k\}$ – множество классов объектов, которые соответствуют группам, на которые были разбиты испытуемые, $P=\{p_j\}$ – множество из m признаков, которые обычно соответствуют определенным шкалам психологических методик и опросников. Составляется матрица значений признаков $A=\{a_{ij}\}$, в которой каждому признаку соответствует столбец его значений из матрицы $A p_j \rightarrow (a_{1j}, a_{2j}, \dots, a_{nj})$, а каждому объекту соответствует его описание $o_i \rightarrow (p_1=a_{i1}, p_2=a_{i2}, \dots, p_m=a_{im})$, где пара $p_j=a_{ij}$ называется свойством объекта.

Далее проводится дискретизация всех интервальных шкал, т.е. перевод их в номинальный вид. При этом используется только три значения для получаемых номинальных признаков: w_3 – низкий уровень признака, w_2 – средний и w_1 – высокий. Далее начинается построение описания класса, которое представляет собой набор свойств (AQ-правил). Свойством h_j далее называется дизъюнкция интервалов значений признака и обозначается $p_j = \bigcup_q w_q$. В исходном варианте алгоритма AQ+JSM применялся стандартный метод AQ-обучения.

2. Применение AQ-обучения для построения базы фактов

На первом шаге AQ-обучения все объекты рассматриваемого класса объявляются положительными примерами, а все остальные объекты – отрицательными примерами. Выбирается произвольный опорный пример, на основе свойств которого строится начальное правило:

$$R_k^1 = \{p_1 = w_{i_1}, p_2 = w_{i_2}, \dots, p_m = w_{i_m}\}. \quad (2.1)$$

Затем происходит процедура расширения начального правила путем либо добавления интервала к какому-либо признаку, либо удаления одного из признаков. При этом проверяется, что получаемое расширенное правило не покрывает ни одного отрицательного примера. При повторении данной процедуры получается дерево правил, листьями которого служат наиболее общие правила, описывающие определенное количество положительных примеров. Из этих правил выбирается наилучшее по количеству покрытых положительных примеров и длине.

В итоге каждому классу c_k ставится в соответствие набор правил, представляющих собой конъюнкцию свойств:

$$\mathfrak{R}_k = \{R_{ki} \mid R_{ki} = \bigcap_j (p_j = \bigcup_q w_q)\}. \quad (2.2)$$

Полученные свойства из всех правил набора выступают тем множеством ключевых свойств, на котором производится поиск причинно-следственных связей. Построение каузальных гипотез происходит с помощью ДСМ-метода для каждого свойства из описания класса, при этом в качестве всего множества объектов (положительных и отрицательных примеров базы фактов) рассматриваются объекты всех классов.

Для решения поставленной задачи классический метод АQ-обучения помимо переборного характера обладает рядом других, критичных для поставленной цели, существенных недостатков. Первым из них является тот факт, что выбор первого примера и, в некоторых случаях признака для расширения, происходит случайным образом. Поэтому получающиеся правила не являются достоверным точным набором всех ключевых свойств класса. В связи с этим в АQ+JSM применялось обучение с накоплением, когда АQ запускался большое количество раз и выбирались самые встречаемые свойства, что существенно увеличивало время работы.

Вторым недостатком является тот факт, что в процессе построения множества правил в него не всегда включаются самые лучшие с точки зрения количества покрытых примеров правила. Это связано с тем, что производится сокращение количества правил, т.е. в АQ считается лучшей та ситуация, когда, например, имеется три правила с покрытием 10 примеров каждый, по сравнению с ситуацией, когда есть 5 правил, первое из которых покрывает 26 приме-

ров, а остальные по 1. Для цели же выделения ключевых свойств выгоднее получать в обязательном порядке те правила, которые покрывают максимальное количество примеров, т.е. использовать данный алгоритм.

Для решения поставленной задачи и устранения недостатков AQ-обучения в данной работе предлагается использовать эволюционный метод генерации покрытий объектов класса.

3. Применение ГА к задаче покрытия

Задачу порождения правил предлагается свести к задаче оптимизации – поиска оптимального набора правил. Поскольку задача имеет большую размерность (в связи с большим числом атрибутов и принимаемых ими значений), классические процедуры оптимизации не могут быть применены, поэтому целесообразно использовать генетический алгоритм (ГА), который применяют при решении сложных нестандартных оптимизационных задач с алгоритмическим заданием функций, сложной конфигурацией области допустимых значений, наличием многоэкстремальных функций, наличием большого числа переменных и т. д. [Емельянов и др., 2003].

Ранее была разработана модификация стандартного ГА, требующая настройки меньшего числа параметров по сравнению со стандартным ГА [Заблоцкий и др., 2011]. Эта модификация, названная коэволюционным асимптотическим генетическим алгоритмом (КАГА), представляет собой несколько параллельно работающих вероятностных асимптотических ГА, которые конкурируют за общий ресурс – число индивидов в популяции, и обмениваются лучшими найденными решениями. Участвующие алгоритмы имеют адаптивный оператор мутации и различаются типом селекции. Такая комбинация алгоритмов позволяет избежать выбора значений операторов селекции, рекомбинации и мутации, которые являются индивидуальными для каждой решаемой задачи.

В настоящей работе предлагается проводить итерационный процесс, на каждом шаге применяя КАГА для поиска оптимального правила, которое покрывает максимальное число положительных примеров и имеет минимальное число используемых атрибутов. Примеры, удовлетворяющие найденному правилу, исключаются и не учитываются на последующих шагах. Если два правила

имеют одинаковую длину и покрытие, то предпочтение отдается правилу, которое покрывает большее число примеров исходного множества.

Индивид представляет собой правило, закодированное в бинарной строке в виде последовательности атрибутов. Так как в решаемой задаче каждый атрибут имеет три значения, то атрибут кодируется тремя числами (0 или 1), которые интерпретируются как отсутствие или наличие соответствующего значения атрибута в правиле. Если всем значениям атрибута соответствуют единицы, то атрибут является несущественным и исключается из правила при декодировании.

Функцию пригодности предложено выбрать как следующую взвешенную сумму:

$$\alpha N_{cur} + \beta N_{init} - \gamma N_{gen} \rightarrow \max, \quad \alpha \gg \beta \gg \gamma > 0, \quad (3.1)$$

где N_{cur} – число покрытых положительных примеров текущего множества, N_{init} – число покрытых положительных примеров исходного множества, N_{gen} – число значений атрибутов, участвующих в правиле. Коэффициенты задаются один раз перед запуском алгоритма таким образом, чтобы первое слагаемое вносило в сумму наибольший вклад, а последнее слагаемое – наименьший вклад. Такой вид функции позволит в первую очередь выявить правила, покрывающие наибольшее число примеров, и только потом выбрать из них правило наименьшей длины.

Сравнение алгоритма AQ с предложенным алгоритмом проводилось на двух задачах. Первая задача содержала 2 класса объектов с 13 атрибутами по 15 и 28 объектов в каждом классе, вторая задача – 3 класса, 31 атрибут, в классах находилось 25, 20 и 25 объектов соответственно. Был выполнен многократный запуск алгоритмов для усреднения результатов.

Таблица 1

Зад.	Класс	Алг.	n_{avr}	n_{max}	r_{avr}	r_{min}	at_{avr}	at_{min}
1	1	AQ	7	10	4	3	8 (4)	2 (2)
		КАГА	7	10	3	3	8 (4)	5 (3)
	2	AQ	9	15	4	4	6 (4)	3 (2)
		КАГА	16	17	3	3	7 (4)	6 (3)

Продолжение таблицы 1

2	1	AQ	8	11	5	4	15 (8)	3 (3)
		КАГА	12	16	3	3	7 (4)	6 (3)
	2	AQ	5	7	5	4	11 (7)	4 (4)
		КАГА	8	10	4	4	8 (5)	7 (4)
	3	AQ	8	11	5	4	16 (9)	11 (7)
		КАГА	12	15	4	3	11 (6)	7 (4)
Средние значения	AQ	7	11	5	4	11 (6)	5 (4)	
	КАГА	11	14	3	3	8 (5)	6 (3)	

В таблице 1 приведены результаты экспериментов на тестовых данных из области психологического тестирования. Колонка r_{avr} означает среднее количество правил, r_{min} – минимальное количество правил, n_{avr} – среднее количество примеров, покрытых одним правилом (в минимальном наборе правил), n_{max} – максимальное количество примеров, покрытых одним правилом (в минимальном наборе правил), at_{avr} – среднее количество значений атрибутов в правиле (в минимальном наборе правил), at_{min} – минимальное количество значений атрибутов в правиле (в минимальном наборе правил). В последних двух столбцах в скобках дано количество атрибутов в правиле.

Из анализа таблицы видно, что на нескольких характерных для психологического тестирования задачах AQ в среднем, даже на небольших объемах данных, выдает менее эффективные с точки зрения количества покрытых примеров правила. С учетом того, что время работы КАГА меньше работы AQ, представляется целесообразным заменить на этапе формирования базы фактов ДСМ-метода в алгоритме AQ+JSM классический алгоритм AQ-обучения на эволюционный алгоритм генерации покрытий.

Заключение

В работе проведен анализ эффективности использования классического метода AQ-обучения для составления базы фактов ДСМ-метода в задаче выявления причинно-следственных связей для группы испытуемых. Были выявлены основные недостатки и пути их устранения, для чего был предложен эволюционный метод

генерации покрытий, на основе которых составляется описание группы испытуемых с необходимыми свойствами. Были проведены эксперименты на двух типичных в психологических исследованиях массивах данных, сравнивающие два метода. Была показана целесообразность использования алгоритма КАГА для составления базы фактов ДСМ-метода.

СПИСОК ЛИТЕРАТУРЫ

[Michalski, 1973] Michalski R.S. AQVAL/1-Computer Implementation of Variable-Valued Logic System VL1 and Examples of its Application to Pattern Recognition // Proc. Of the First Int. Joint Conf. on Pattern Recognition. Washington, DS, 1973. P. 3–17.

[Wojtusiak et al., 2006] The aq21 natural induction program for pattern discovery: Initial version and its novel features / Janusz Wojtusiak, Ryszard S. Michalski, Kenneth A. Kaufman, Jaroslaw Pietrzykowski // ICTAI.– 2006.– P. 523–526.

[Емельянов и др., 2003] Емельянов В.В., Курейчик В.В., Курейчик В.М. Теория и практика эволюционного моделирования. – М.: Физматлит, 2003.

[Заблоцкий и др., 2011] Заблоцкий С.Г., Семенкин Е.С., Швец А.В. Коэволюционный асимптотический генетический алгоритм для формирования предложений по слоговой модели в системе автоматического распознавания слитной речи // Вестник Сибирского государственного аэрокосмического университета имени академика М.Ф. Решетнёва. – 2011. – № 3.

[Осипов, 1990] Осипов Г.С. Построение моделей предметных областей. Ч.I. Неоднородные семантические сети // Изв. АН СССР. Техн. кибернетика. 1990. №5.

[Осипов, 1997] Осипов Г.С. Приобретение знаний интеллектуальными системами: основы теории и технологии. – М.: Физматлит, 1997.

[Панов, 2010] Панов А.И. Применение методов искусственного интеллекта в обработке психологических данных // Экспериментальная психология в России: традиции и перспективы / Под ред. В. А. Барабанщикова.– М.: Институт психологии РАН, 2010.

[Панов, 2013] Панов А.И. Выявление причинно-следственных связей в данных психологического тестирования логическими методами // Искусственный интеллект и принятие решений. – 2013. – №1.

[Финн, 1991] Финн В.К. Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Итоги науки и техники. 1991. Т. 15.

[Финн, 2000] Финн В.К. Каузальный анализ данных в интеллектуальных системах // Научно-техническая информация. Сер. 2. Информационные процессы и системы. 2000.

УДК 004.891 + 311.2

БАЙЕСОВСКАЯ СЕТЬ ДОВЕРИЯ КАК ИНСТРУМЕНТ ОЦЕНИВАНИЯ ИНТЕНСИВНОСТИ РИСКОВАННОГО ПОВЕДЕНИЯ РЕСПОНДЕНТОВ¹

А.В. Суворова (*sualv@mail.ru*)

Санкт-Петербургский институт информатики и автоматизации РАН,
Санкт-Петербург

Р.Р. Хайбуллин (*Ruslan.Khaibullin@lanit-tercom.com*)

Санкт-Петербургский государственный университет,
Санкт-Петербург

А.Л. Тулупьев (*alt@ias.spb.su*)

Санкт-Петербургский институт информатики и автоматизации РАН,
Санкт-Петербург

В докладе предложен подход к построению модели рискованного поведения по совокупности наблюдений, включающей сведения об эпизодах такого поведения. Описаны предложенная модель на основе байесовской сети доверия и разработанное программное обеспечение, приведены результаты численных экспериментов на статистических данных.

Введение

Байесовские сети доверия являются одним из классических теоретических аппаратов для представления и обработки данных и знаний с неопределенностью [Тулупьев и др., 2006; Pearl, 2000]. Байесовские сети доверия [Тулупьев и др., 2006] используются для моделирования связей в некоторой системе случайных элементов в тех случаях, когда в силу сложности системы или неполноты и неточности имеющейся информации о ее частях невозможно полностью описать совместное вероятностное распределение рассматриваемых случайных элементов. Аппарат байесовской сети доверия позволяет учитывать как имеющиеся статистические данные,

¹ Статья содержит материалы исследований, частично поддержанных грантами РФФИ 14-01-00580-а, субгрантом № M13A11589 (A06995) гранта НИИ № 5 R01AA017389-04 «Alcohol and HIV Risk Reduction in St. Petersburg, R.F.».

так и экспертную информацию об интересующей исследователя области. Такие модели изучают и используют специалисты ряда научных дисциплин: искусственного интеллекта (ИИ) [Lee et al., 2008], математической статистики [Мусина, 2013], статистической физики [Wan et al., 2013], теоретической информатики, когнитивных исследований (например, [Martins et al., 2013; McCabe et al., 2008; Trucco et al., 2008]).

Данный доклад посвящен применению методов аппарата байесовских сетей доверия для моделирования рискованного поведения респондентов.

1. Постановка задачи

Моделирование поведения человека на основе ограниченного числа известных параметров необходимо при решении задач, относящихся к различным областям, включая исследования в психологии, экономике, социологии, медицине, эпидемиологии, компьютерной безопасности и т.д. [Азаров и др., 2012; Афанасьев, 2006; Тулупьева и др., 2008; Schmitt et al., 2013]. Подобной задачей является и разработка моделей и методов для обеспечения поддержки принятия решений в условиях, когда для получения данных невозможно организовать классические формы длительного наблюдения и многофакторного измерения параметров процесса, но имеются сведения, полученные от экспертов, предположения о классах и семействах таких процессов, а также ограниченное число измеряемых особенностей такого процесса [Тулупьева и др., 2008]. В частности, такие данные возникают при изучении поведения посредством опросов участников или экспертов.

В работах [Суворова и др., 2010; Тулупьева и др., 2008] предложен подход, позволяющий получать количественные данные об эпизодах поведения из опросов респондентов в условиях невозможности длительного наблюдения или измерения параметров, а также рассмотрены методы получения (средствами теории случайных процессов) оценки риска на основе данных о последних эпизодах рискованного поведения.

Сведение исходной задачи к построению байесовской сети доверия [Суворова, 2013] позволяет воспользоваться уже существующим мощным алгоритмическим аппаратом теории байесовских сетей доверия и свободно распространяемым программным ин-

струментарием (например, [GeNIe, 2013]) для проведения вычислительных экспериментов, а затем и для использования построенной модели в практических целях.

2. Описание модели

В основу предлагаемой модели [Суворова, 2013] положены сведения о последних эпизодах поведения, а также минимальном и максимальном интервалах между последовательными эпизодами. Такие данные получены в результате опроса или анкетирования респондентов.

На рис. 1 показана обобщенная модель $M = (G(V, L), \mathbf{P})$ в виде байесовской сети доверия, основывающаяся на большем числе величин. Структура модели представлена графом $G(V, L)$, где $V = \{t_{01}, t_{12}, t_{23}, t_{\min}, t_{\max}, \lambda, n\}$ – множество вершин, $L = \{(u, v) : u, v \in V\}$ – множество направленных связей между вершинами. Другими словами, на рис. 1 представлены случайные элементы, входящие в модель, и связи между ними.

Так, t_{ij} – случайная величина, характеризующая длину интервала между i -ым и j -ым с конца эпизодами, распределена экспоненциально (в предположении, что поведение представляет собой пуассоновский процесс). Кроме того, дополнительную информацию можно получить при включении в модель минимального и максимального интервалов между эпизодами (t_{\min} и t_{\max} соответственно).

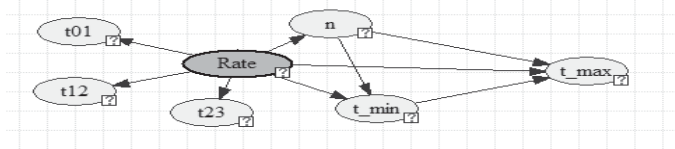


Рис. 1. Модель рискованного поведения, основанная на данных об эпизодах поведения

Тензоры \mathbf{P} условной вероятности, характеризующие переходы между узлами сети, где $\mathbf{P} = \{P(t_{j,j+1}|\lambda), P(t_{01}|\lambda), P(t_{\min}|n, \lambda), P(t_{\max}|n, \lambda, t_{\min}), P(n|\lambda), P(\lambda)\}$, определяются следующим образом: ($l_s = 1, \dots, k_s$, где k_s – число дизъюнктивных промежутков при дискретизации случайных величин; $s = 0, \dots, 4$; $j = 1, \dots, 2$; $i = 1, \dots, m$, где m – число дизъюнктивных промежутков при дискретизации величины λ) [Суворова, 2013]:

$$\begin{aligned}
p(t_{j,j+1}^{(j)} | \lambda^{(i)}) &= e^{-a\lambda^{(i)}} - e^{-b\lambda^{(i)}}, \quad j=0,1,2, \quad t_{j,j+1}^{(j)} = [a;b]; \\
p(t_{\min}^{(j)} | n, \lambda^{(i)}) &= e^{-an\lambda^{(i)}} - e^{-bn\lambda^{(i)}}, \quad t_{\min}^{(j)} = [a;b]; \\
p(n | \lambda^{(i)}) &= \frac{(\lambda^{(i)}T)^n}{n!} e^{-\lambda^{(i)}T}; \\
p(t_{\max}^{(j)} | n, \lambda^{(i)}, t_{\min}^{(j)}) &= e^{-(n-1)\lambda^{(i)}t_{\min}^{(j)}} \left(\left(e^{-\lambda^{(i)}t_{\min}^{(j)}} - e^{-\lambda^{(i)}b} \right)^{n-1} - \left(e^{-\lambda^{(i)}t_{\min}^{(j)}} - e^{-\lambda^{(i)}a} \right)^{n-1} \right), \\
t_{\max}^{(j)} &= [a;b].
\end{aligned}$$

После полного определения тензоров условных вероятностей для всех узлов модели численные расчеты апостериорного распределения интенсивности рискованного поведения при поступлении статистических данных будут выполнены программным обеспечением, например GeNIe&SMILE [GeNIe, 2013]. Апостериорное распределение позволяет судить об интервале оценки интенсивности, которого достаточно для дальнейшего принятия решения об уровне риска.

3. Апробация модели на данных

Апробация модели проведена на подвыборке объемом 100 случаев из данных, полученных в рамках полевого исследования на базе кожно-венерологического диспансера №9 Калининского района Санкт-Петербурга, в мае – ноябре 2011 года.

Для каждого респондента известны данные о последнем, предпоследнем и третьем с конца эпизодах потребления алкоголя, а также минимальном и максимальном интервалах между эпизодами.

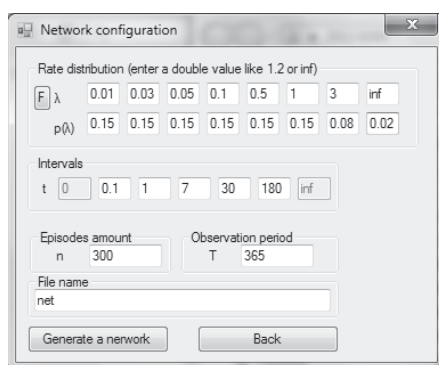


Рис. 2. Программа для автоматизации обработки данных с помощью байесовской сети доверия

С целью автоматизации обработки данных разработана программа (рис. 2), позволяющая задавать различные варианты дискретизации, а также производить вычисления для всех элементов выборки, не используя ручной ввод данных.

Интенсивность разбита на дискретные интервалы $[0; 0,01)$, $[0,01; 0,03)$, $[0,03; 0,05)$, $[0,05; 0,1)$, $[0,1; 0,5)$, $[0,5; 1)$, $[1; 3)$, $[3; \infty)$. Априорное распределение представлено на рис. 2. Апостериорное распределение получено GeNIe&SMILE [GeNIe, 2013] с использованием дополнительно разработанной программы для автоматизации обработки целого массива данных. В качестве значения интенсивности, вычисленной по данным об эпизодах потребления алкоголя, выбирается интервал с максимальной вероятностью. Полученное распределение интенсивности на выборке объемом 100 представлено на рис. 3. Так, у всех респондентов интенсивность меньше 0,1, то есть эпизоды рискованного алкопотребления происходят не чаще одного раза в 10 дней. У 30% выборки значение интенсивности принадлежит промежутку $(0,05; 0,1]$, то есть эпизоды потребления алкоголя происходят в среднем один раз в 10–20 дней. Таким образом, предложенная модель позволяет обрабатывать данные об эпизодах поведения, полученные в результате опросов, и на основе вычисленного апостериорного распределения делать выводы об интенсивности рискованного поведения.

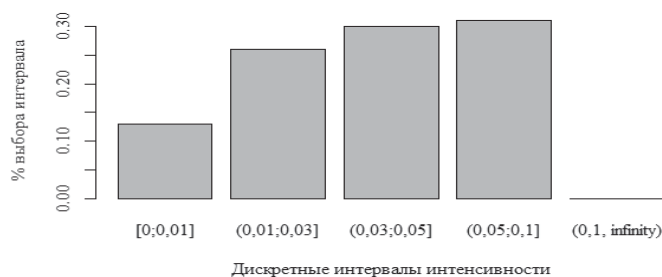


Рис. 3. Распределение апостериорных интенсивностей

Заключение

В докладе предложен подход к построению модели рискованного поведения по совокупности наблюдений, включающей сведения об эпизодах такого поведения. Применение аппарата байесовских сетей позволяет решать только содержательную задачу – постро-

ить байесовскую сеть – корректную модель предметной области, а вычисления обеспечиваются существующим программным обеспечением. Описаны предложенная модель и разработанное программное обеспечение, приведены результаты численных экспериментов на статистических данных.

СПИСОК ЛИТЕРАТУРЫ

[Азаров и др., 2012] Азаров А.А., Тулупьева Т.В., Фильченков А.А., Тулупьев А.Л. Вероятностно-реляционный подход к представлению модели комплекса «информационная система – персонал – критичные документы» // Труды СПИИ РАН. 2012. Вып. 1(20).

[Афанасьев, 2006] Афанасьев И.В. Возможности математического моделирования поведения аудитории с помощью динамических математических моделей // Актуальные проблемы современной науки. 2006. № 4.

[Мусина, 2013] Мусина В.Ф. Байесовские сети доверия как вероятностная графическая модель для оценки экономических рисков // Труды СПИИ РАН. 2013. Вып. 25.

[Суворова и др., 2010] Суворова А.В., Тулупьев А.Л., Пашенко А.Е., Тулупьева Т.В., Красносельских Т.В. Анализ гранулярных данных и знаний в задачах исследования социально значимых видов поведения // Компьютерные инструменты в образовании. №4. 2010.

[Суворова, 2013] Суворова А.В. Моделирование социально-значимого поведения по сверхмалой неполной совокупности наблюдений // Информационно-измерительные и управляющие системы. 2013. №9, т. 11.

[Тулупьев и др., 2006] Тулупьев А.Л., Николенко С.И., Сироткин А.В. Байесовские сети: логико-вероятностный подход. – СПб.: Наука, 2006.

[Тулупьева и др., 2008] Тулупьева Т.В., Пашенко А.Е., Тулупьев А.Л., Красносельских Т.В., Казакова О.С. Модели ВИЧ-рискованного поведения в контексте психологической защиты и других адаптивных стилей. – СПб.: Наука, 2008.

[GeNie, 2013] GeNie & SMILE // Decisions systems laboratory. School of Information Sciences. University of Pittsburg. URL: <http://genie.sis.pitt.edu/>.

[Lee et al., 2008] Lee S., Son Y.-J., Jin J. Decision field theory extensions for behavior modeling in dynamic environment using Bayesian belief network // Information Sciences. 2008. Vol. 178.

[Martins et al., 2013] Martins M.R., Maturana M.C. Application of Bayesian Belief networks to the human reliability analysis of an oil tanker operation focusing on collision accidents // Reliability Engineering and System Safety. 2013. Vol. 110.

[McCabe et al., 2008] McCabe B., Loughlin C., Munteanu R., Tucker S.,

Lam A. Individual safety and health outcomes in the construction industry // Can. J. Civ. Eng. 2008. Vol. 35.

[Pearl, 2000] Pearl J. Causality: Models, Reasoning, and Inference. – Cambridge: Cambridge University Press, 2000.

[Schmitt et al., 2013] Schmitt L., Brugere C. Capturing Ecosystem Services, Stakeholders' Preferences and Trade-Offs in Coastal Aquaculture Decisions: A Bayesian Belief Network Application // PLoS ONE. 2013. Vol. 8. Issue 10.

[Trucco et al., 2008] Trucco P., Cango E., Ruggeri F., Grande O. A Bayesian Belief Network modelling of organisational factors in risk analysis: A case study in maritime transportation // Engineering and System Safety. 2008. Vol. 93.

[Wan et al., 2013] Wan J., Zabarar N. A probabilistic graphical model approach to stochastic multiscale differential equations // Journal of Computational Physics. 2013. No. 250.