РОССИЙСКАЯ АКАДЕМИЯ НАУК

ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
(ВИНИТИ)

НАУЧНО • ТЕХНИЧЕСКАЯ
ИНФОРМАЦИЯ

Серия 2. ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ И СИСТЕМЫ

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ СБОРНИК

Издается с 1961 г. № 9 Москва 2012

ОБЩИЙ РАЗДЕЛ

УДК 004.89:510.64

О.М. Аншаков

ДСМ-метод: теоретико-множественное объяснение

Представлено оригинальное изложение основных понятий и правил ДСМ-метода на языке обычной теории множеств (без использования аппарата неклассических логик). Рассматриваются архитектура и принципы работы ДСМ-системы. Обсуждается связь ДСМ-метода и анализа формальных понятий. Даются практические рекомендации для разработчиков нестандартных версий ДСМ-систем.

Ключевые слова: *ДСМ-метод, интеллектуальный анализ данных, правдоподобные рассуждения, анализ формальных понятий, соответствие Галуа*

ВВЕДЕНИЕ

ДСМ-метод автоматического порождения гипотез был предложен В.К. Финном в работах [1–3]. Основные публикации по ДСМ-методу, вышедшие до 2009 г., можно найти в сборниках [4, 5]. В настоящее время ДСМ-метод рассматривается как оригинальная совокупность логико-комбинаторных технологий интеллектуального анализа данных, использующая формализованные (с помощью неклассических логик) правила правдоподобных рассуждений. Если говорить более подробно, то ДСМ-метод обнаруживает закономерности в данных с помощью *индукции* в стиле Д.С. Милля [6], порождает правила устранения неопределенности в данных, используя *фальсификацию* в стиле К. Поппера [7], формирует предсказания с помощью оригинальной версии рассуждений по *аналогии* и оценивает правдоподобие порожденных гипотез с помощью некоторого варианта *абдукции* Ч.С. Пирса [8]. Аббревиатура «ДСМ» — это инициалы Джона Стюарта Милля, формализованные правила индуктивной логики которого являются идеологическим фундаментом ДСМ-метода.

Современный взгляд на ДСМ-метод, представленный его основателем, содержится в статьях [9–11] и книге [12]. В классическом ДСМ-методе для обнаружения закономерностей использовался формализованный аналог метода сходства Д.С. Милля. В статьях [9, 10] В.К.Финн дает формализованные описания

всех правил индуктивной логики Д.С. Милля. В настоящее время идет интенсивная работа над созданием систем анализа данных, поддерживающих разные методы индуктивной логики Д.С. Милля. В связи с этим, прежде всего, необходимо упомянуть статьи А.Ю. Волковой, например [13].

В большинстве работ по ДСМ-методу правила правдоподобных рассуждений описывались с помощью многозначных логик с J-операторами в стиле Россера и Тьюркетта [14]. Это обстоятельство делает изучение основ ДСМ-метода непростой задачей для разработчиков программного обеспечения, не имеющих специальной подготовки в области неклассических логик. Возможно, это является одной из причин того факта, что ДСМ-метод является не настолько широко распространенным методом анализа данных, насколько он этого заслуживает.

Попытки объяснить ДСМ-метод с помощью другой математической техники делались неоднократно, но нельзя сказать, что этих попыток было много. В связи с этим следует упомянуть статью В.Б. Борщева [15], которая, по-видимому, была первой работой, содержащей нестандартное изложение ДСМ-метода.

В статье О.М. Аншакова [16] ДСМ-метод был изложен на языке обычной теории множеств. Используя упрощенную версию ДСМ-метода, описанную в [16], исследователи из Института механики МГУ им. М.В.Ломоносова создали систему автоматической классификации структур двухфазных сплавов. Подробное описание этой системы имеется в кандидатской диссертации А.С. Шундеева [17]. Работа [16] имеет также педагогическое значение, так как упрощенное изложение ДСМ-метода облегчает его понимание и использование для разработки интеллектуальных систем. В учебном пособии Г.С. Осипова [18] ДСМ-метод излагается по материалам статьи [16].

В дальнейшем теоретико-множественный подход к ДСМ-методу использовался в работах А.А. Липкина, в частности, в его кандидатской диссертации [19]. Теоретико-множественный подход к правилам ДСМ-метода позволил автору этих работ сравнительно просто определить правила обобщенного ДСМ-метода для случая так называемого ранжированного ДСМ-метода и разработать прототип соответствующей ДСМ-системы.

Резюмируя сказанное выше, можно сделать вывод о том, что наличие упрощенного (даже вульгаризированного) описания ДСМ-метода является полезным обстоятельством. Оно способствует снижению барьера вхождения в сообщество разработчиков ДСМ-систем (или ДСМ-подобных систем) для независимых исследователей и инженеров. Кроме того, упрощенное описание ДСМ-метода дает возможность достаточно быстро (хотя и поверхностно) ознакомиться с ним начинающим исследователям и разработчикам, которые в дальнейшем могут продолжить изучение ДСМ-метода со всеми его тонкостями и подробностями.

Настоящая статья посвящена упрощенному описанию элементов ДСМ-метода на языке наивной теории множеств. Содержание этой статьи не дублирует содержание работы [16] (и вообще имеет с ним мало общего), несмотря на то, что статья [16] также посвящена изложению ДСМ-метода на теоретико-множественном языке.

В статье [16] автор делает попытку рассмотреть ДСМ-метод с некоторой общей точки зрения, например, обобщить понятие *фрагмента* исследуемого объекта, используя понятие *характеристики* и формализованного *описания*. Опыт показал, что такое обобщение оказалось невостребованным при программной реализации ДСМ-систем. Кроме того, в [16] предполагалось, что структура исследуемых объектов может быть представлена различными способами: в виде вектора, строки, графа, множества, мультимножества и т.п. (конкретный способ представления объектов в [16] не назывался). Однако случаи использования в ДСМ-системах представления объектов, отличного от представления в виде множества, очень редки, поэтому вряд ли в популярном изложении имеет смысл о них упоминать. И, наконец, в [16] все-таки говорится о нестандартных истинностных значениях, вводятся соответствующие обозначения, описываются (хотя и неформально) правила правдоподобных рассуждений, т.е., дается представление о логических основаниях ДСМ-метода. В настоящее время автор считает, что в популярном изложении ДСМ-метода можно обойтись и без этого.

В предлагаемой читателю статье, в отличие от [16], акцент делается не на логических, а на архитектурных и алгоритмических проблемах.

1. НАЗНАЧЕНИЕ, КОМПОНЕНТЫ И ОСНОВНОЙ АЛГОРИТМ ДСМ-МЕТОДА

В работе [9] В.К. Финн выделяет пять компонентов ДСМ-метода:

(1) условия применимости,

(2) ДСМ-рассуждения,

(3) представление знаний в виде открытых квазиаксиоматических теорий (КАТ),

(4) метатеоретические принципы и средства исследования рассуждений и предметных областей (в том числе дедуктивная имитация рассуждений, процедурная семантика и препроцессинг, результатом которого является выбор стратегий рассуждения и соответствующей им процедурной семантики),

(5) интеллектуальные системы типа ДСМ (ИС-ДСМ).

В статье [11] к ним добавляется компонент:

(6) обнаружение эмпирических закономерностей завершающих процесс knowledge discovery в базе фактов (БФ) посредством ИС-ДСМ.

В настоящей работе нас будет интересовать только компонент (5) — ДСМ-системы, остальных компонентов мы будем касаться лишь в том случае, если это будет необходимо для объяснения работы ДСМ-системы. Под *ДСМ-системой* будем понимать систему интеллектуального анализа данных, т.е. извлечения знаний из данных (обнаружения закономерностей в данных) с помощью формализованных правдоподобных рассуждений, основанных на (довольно сильно модифицированных) «канонах Д.С. Милля» [6].

ДСМ-система предназначена для обнаружения связи *между структурой объекта и его поведением,* которая интерпретируется как причинно-следственная связь. Как правило, *характерные черты структуры объекта считаются причинами особенностей его поведения*, но в некоторых случаях принята обратная интерпретация: совокупность элементов поведения считается причиной множества характеристик его структуры. Это принято в обратном ДСМ-методе, введенном в статье С.М. Гусаковой, М.А. Михеенковой и В.К. Финна [20], (см. также [4, ч. III, гл. 3]).

Элементы структуры объекта будем называть *атомами*. Элементы поведения будем называть (целевыми) *свойствами*. Исследуемый *объект* представляется в виде конечного множества атомов. Он может обладать (или не обладать) некоторым множеством целевых свойств. Предполагается, что как у наличия, так и у отсутствия набора целевых свойств может быть причина (не обязательно единственная), эта причина является *фрагментом* структуры объекта. Так как объекты представлены множествами, фрагменты представляются подмножествами объектов, т.е. опять же конечными множествами атомов. Модель причинно-следственных связей, используемая ДСМ-методом, может быть описана графом, изображенном на рис. 1.

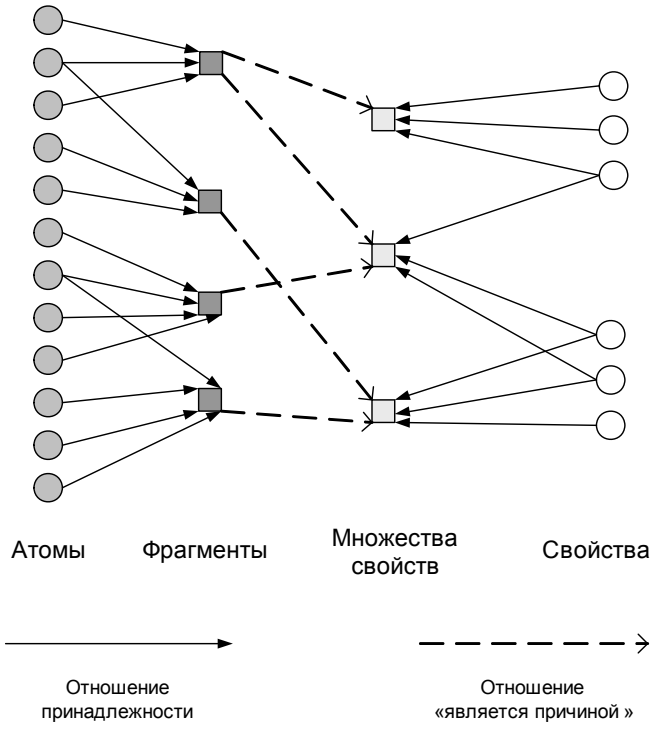


Рис. 1. Сущности ДСМ-метода и связи между ними

ДСМ-система решает две основные задачи:

(1) формирование гипотез о возможных причинах наличия (отсутствия) свойств у объектов (возможные причины являются фрагментами — подмножествами — объектов),

(2) формирование гипотез о наличии (отсутствии) свойств у тех объектов, для которых это было неизвестно.

Формирование гипотез о возможных причинах происходит с помощью *формализованных индуктивных рассуждений*. Формирование гипотез о наличии (отсутствии) свойств можно интерпретировать как применение *формализованных рассуждений по аналогии*.

Классический алгоритм ДСМ-метода, включающий работу ДСМ-системы, представляет собой итеративную процедуру, которую можно изобразить диаграммой деятельности UML* (рис. 2). Рассмотрим основные блоки этой процедуры.

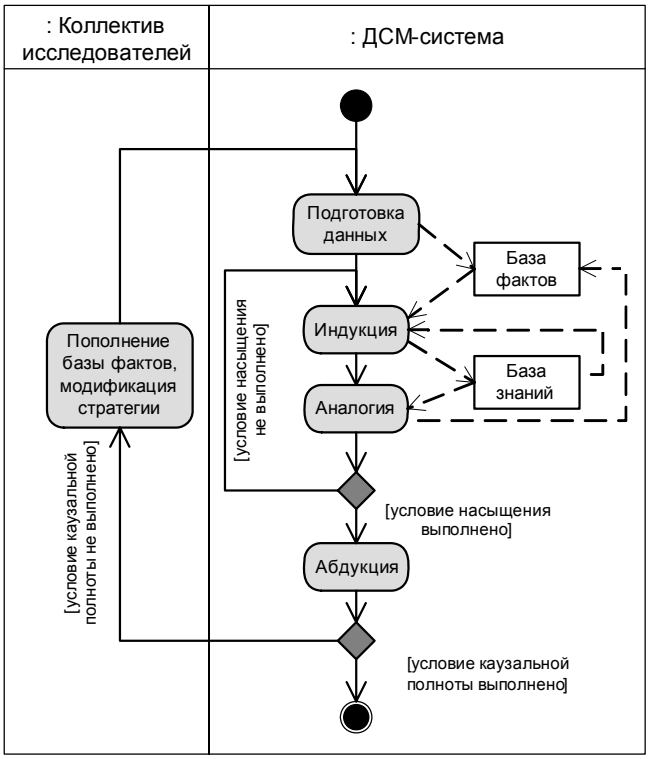


Рис. 2. Алгоритм ДСМ-метода

Подготовка данных в контексте работы ДСМ-системы означает преобразование данных во внутренний формат ДСМ-системы. Исходные данные могут быть представлены в некотором стандартном формате, например в виде CSV-файла, где каждому объекту соответствует строка таблицы. Для работы ДСМ-системы объекты необходимо представить в виде множеств. В результате процедуры подготовки данных формируется *база фактов*, представленных во внутреннем формате ДСМ-системы.

Индукция — применение формализованных правил индуктивных рассуждений — предназначена для порождения гипотез о возможных причинах наличия (отсутствия) целевых свойств. Гипотезы о возможных причинах представляют некоторые общие закономерности, которые включаются в *базу знаний* ДСМ-системы.

Аналогия — применение формализованных правил рассуждений по аналогии — предназначена для порождения гипотез о наличии (отсутствии) целевых

* По языку UML существует обширная литература, сошлемся, например, на наиболее научное и непредвзятое, по мнению автора, изложение из книги [21].

свойств у тех объектов, для которых это неизвестно. Гипотезы о наличии (отсутствии) целевых свойств имеют такой же формат, как исходные факты и включаются в *базу фактов*.

Проверка условия насыщения — условие насыщения считается выполненным, если невозможно получить новые гипотезы о наличии (отсутствии) свойств у тех объектов, для которых это было неизвестно. Поскольку количество таких объектов конечно, насыщение обязательно (рано или поздно) наступает. Если насыщение наступило, происходит выход из цикла *Индукция–Аналогия* и переход к блоку абдукции.

Абдукция — это проверка условия (критерия) каузальной полноты. Условие каузальной полноты считается выполненным, если для каждого объекта, обладающего (не обладающего) требуемым набором целевых свойств, найдена причина наличия (отсутствия) этого набора свойств. Если условие каузальной полноты выполнено, то работа ДСМ-системы завершается, в противном случае происходит расширение исходного набора данных и, возможно, происходит модификация стратегии работы ДСМ-системы.

Пополнение базы фактов и модификация стратегии производится пользователями и администраторами (настройщиками) ДСМ-системы. Пополняется набор исходных данных, который преобразуется в базу фактов уже самой ДСМ-системой. Модификация стратегии включает различные операции, в частности, может быть выбрана альтернативная разновидность ДСМ-метода, изменены параметры правил индукции и аналогии и т.п.

ДСМ-рассуждением будем называть часть алгоритма ДСМ-метода, полученную удалением процедур *подготовки данных* и *пополнения базы фактов и модификации стратегии*. Эта часть относится только к ДСМ-системе и играет в ней очень важную роль. Введем некоторые понятия, относящиеся к работе ДСМ-рассуждения, следуя, в основном, формулировкам из статьи В.К. Финна [9]:

Шагом ДСМ-рассуждения будем называть однократное выполнение процедуры индукции или процедуры аналогии.

Тактом ДСМ-рассуждения будем называть однократное последовательное выполнение процедур индукции и аналогии.

Этапом I ДСМ-рассуждения будем называть внутренний цикл работы ДСМ-метода, состоящий из последовательных тактов, выполняемых до тех пор, пока не будет выполнено условие насыщения.

Этапом II ДСМ-рассуждения будем называть выполнение процедуры абдукции и проверку условия каузальной полноты.

В настоящей статье нас будет интересовать, главным образом, *Этап I* ДСМ-рассуждений. Мы будем называть этот этап *ядром ДСМ-системы*. Здесь же изложим некоторые общие соображения относительно процедуры подготовки данных.

2. АРХИТЕКТУРА ДСМ-СИСТЕМЫ

Архитектура ДСМ-системы представлена на рис. 3. Она включает: ДСМ-решатель, базу фактов (данных), базу знаний и интерфейс, разделенный на два компонента: интерфейс конечного пользователя и интерфейс для настройки системы. Конечный пользователь формулирует конкретные задачи, а настройщик системы определяет стратегию и методы подготовки данных, которые будут применяться для различных задач конечного пользователя. Роль настройщика для ДСМ-системы аналогична роли инженера по знаниям для продукционных экспертных систем.

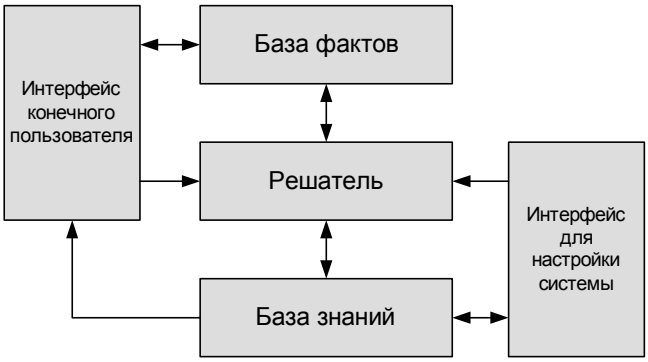


Рис. 3. Архитектура ДСМ-системы

База знаний ДСМ-системы имеет сложную структуру. Она включает:

- *предметные знания* (они описывают известные закономерности предметной области и используются, в основном, для подготовки данных) — включают правила или процедуры перевода данных во внутренний формат и правила или процедуры для определения структуры объектов,
- *метазнания* — правила или процедуры, представляющие формализованные правдоподобные рассуждения, — индукцию и аналогию,
- *гипотезы о причинах наличия (отсутствия) целевых свойств* — они имеют формат, аналогичный формату данных, но описывают общие закономерности, поэтому должны интерпретироваться как знания.

База фактов содержит:

- сведения о *наличии или отсутствии целевых свойств*,
- сведения о *структуре объектов*.

Интерфейс конечного пользователя позволяет организовать ввод исходных данных и сформулировать задание для ДСМ-системы.

Интерфейс для настройки системы позволяет модифицировать правила и/или процедуры индукции и аналогии, выбрать разновидность ДСМ-метода и задать его параметры.

Решатель исполняет ту часть алгоритма ДСМ-метода, которая относится к работе ДСМ-системы (см. рис. 2). Решатель содержит *вычислитель* и *рассуждатель*. Вычислитель исполняет процедуры подготовки данных и обнаружения сходства объектов, рассуждатель реализует процедуры правдоподобных рассуждений: индукции, аналогии и абдукции.

По отношению к внутреннему циклу из алгоритма, изображенного на рис. 2 (ядру ДСМ-системы), база фактов и база знаний делятся на *постоянную* и *модифицируемую* части. Постоянная часть базы фактов — это сведения о структуре объектов. Модифицируемая

часть базы фактов — это сведения о наличии (отсутствии) у объектов целевых свойств. Постоянная часть базы знаний — это предметные знания и метазнания. Модифицируемая часть базы знаний — это гипотезы о возможных причинах наличия (отсутствия) целевых свойств объектов. Структура базы фактов и базы знаний изображена на рис. 4.

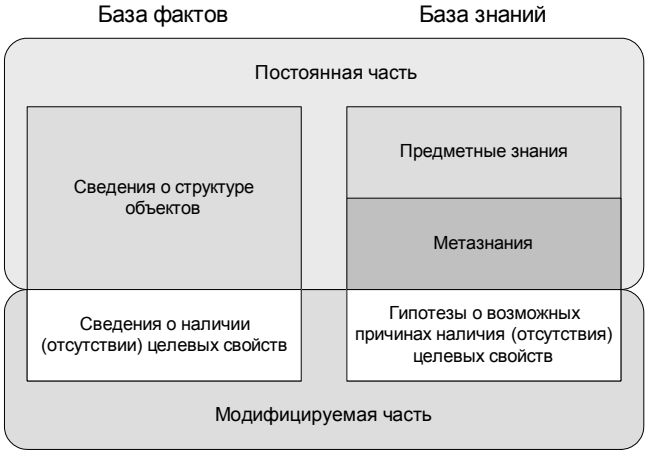


Рис. 4. Структура базы фактов и базы знаний

Модифицируемая часть базы знаний изменяется с помощью процедуры индукции, модифицируемая часть базы фактов изменяется с помощью процедуры аналогии. Постоянные части базы фактов и базы знаний не изменяются ядром ДСМ-системы.

Подчеркнем, что относительно знаний нигде в явной форме не говорилось, имеем ли мы в виду *процедурные знания* (алгоритмы) или *декларативные знания* (системы правил). Это связано с тем обстоятельством, что каждый компонент базы знаний (см. рис. 4) может быть реализован и как система процедурных, и как система декларативных знаний. Исключением являются гипотезы о возможных причинах, которые по форме напоминают факты и являются декларативными знаниями. Предметные знания, как правило, реализуются как процедурные. Метазнания могут быть реализованы и как процедурные, и как декларативные знания, хотя с точки зрения теории они являются декларативными знаниями (системой правил формализованных правдоподобных рассуждений).

3. АТОМЫ, ОБЪЕКТЫ, СВОЙСТВА, БАЗОВЫЕ ЛОГИЧЕСКИЕ ФУНКЦИИ

Рассмотрим теоретико-множественную версию математической модели предметной области, лежащую в основе ДСМ-системы.

Как уже было сказано, подлежащие рассмотрению *объекты* представляются в виде конечных множеств. Все возможные элементы таких множеств образуют *универсум атомов*, который будем обозначать через $\mathbf{A}$. Множество подлежащих рассмотрению объектов — *универсум объектов* — будем обозначать через $\mathbf{O}$. Очевидно $\mathbf{O} \subseteq 2^{\mathbf{A}}$, где через $2^{\mathbf{A}}$ обозначено множество всех подмножеств множества $\mathbf{A}$. Каждое подмножество множества $\mathbf{A}$ будем называть *фрагментом*, а множество $2^{\mathbf{A}}$ — *универсумом фрагментов*. Универсум фрагментов будем обозначать через $\mathbf{F}$.

Разумеется, для реальной ДСМ-системы универсум фрагментов — это просто абстракция. Никто никогда не пытался целиком построить этот универсум, да это и невозможно, поскольку в реальных задачах универсум атомов $\mathbf{A}$ может содержать несколько сотен, а то и тысяч, элементов.

Множество всех (целевых) свойств, которыми (по нашим представлениям) могут обладать объекты, будем называть *универсумом целевых свойств* и обозначать через $\mathbf{T}$. Совокупность подлежащих рассмотрению множеств целевых свойств будем называть *универсумом значимых множеств целевых свойств* и обозначать через $\mathbf{P}$. Относительно множеств целевых свойств необходимо учитывать следующие обстоятельства:

- как правило, множество целевых свойств воспринимается как единое целое,
- обычно, множество $\mathbf{T}$ невелико по объему или просто одноэлементно,
- в большинстве случаев, элементы множества $\mathbf{P}$ являются одноэлементными подмножествами множества $\mathbf{T}$.

Через $\mathbf{B}$ будем обозначать множество классических истинностных значений: *false* и *true*. Обозначать эти значения будем через 0 и 1, соответственно, т.е., $\mathbf{B} = \{0,1\}$.

Будем предполагать, что у нас есть возможность определить следующие вычислимые функции со значениями из $\mathbf{B}$:

– $\mathrm{P}^+ : \mathbf{O} \times \mathbf{P} \to \mathbf{B}$ — функция «обладает множеством свойств», для любых $o \in \mathbf{O}$, $p \in \mathbf{P}$

$$\mathrm{P}^+(o,p) = \begin{cases} 1, & \text{если объект } o \text{ обладает множеством} \\ & \text{свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

– $\mathrm{P}^- : \mathbf{O} \times \mathbf{P} \to \mathbf{B}$ — функция «не обладает множеством свойств», для любых $o \in \mathbf{O}$, $p \in \mathbf{P}$

$$\mathrm{P}^-(o,p) = \begin{cases} 1, & \text{если объект } o \text{ не обладает множеством} \\ & \text{свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

– $\mathrm{C}^+ : \mathbf{F} \times \mathbf{P} \to \mathbf{B}$ — функция «является возможной причиной наличия множества свойств», для любых $s \in \mathbf{F}$, $p \in \mathbf{P}$

$$\mathrm{C}^+(s,p) = \begin{cases} 1, & \text{фрагмент } s \text{ является причиной} \\ & \text{наличия множества свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

– $\mathrm{C}^- : \mathbf{F} \times \mathbf{P} \to \mathbf{B}$ — функция «является возможной причиной отсутствия множества свойств», для любых $s \in \mathbf{F}$, $p \in \mathbf{P}$

$$\mathrm{C}^-(s,p) = \begin{cases} 1, & \text{фрагмент } s \text{ является причиной} \\ & \text{отсутствия множества свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

Необходимо отметить, что значения логических функций P^+, P^-, C^+ и C^- зависят от результатов наблюдений и/или экспериментов или от результатов *недостоверных* рассуждений. Сведения, на основании которых определяются значения функций P^+, P^-, C^+ и C^-, могут быть неполными, неточными и даже противоречивыми. Поэтому мы не можем определенно говорить о наличии или отсутствии свойств и, тем более, о возможных причинах этого наличия или отсутствия.

Более правильно говорить о том, что имеются аргументы за (или против) наличия или отсутствия свойств и т.п. Например, функцию $\mathrm{P}^+ : \mathbf{O} \times \mathbf{P} \to \mathbf{B}$ правильнее было бы описать так: для любых $o \in \mathbf{O}$, $p \in \mathbf{P}$

$$\mathrm{P}^+(o,p) = \begin{cases} 1, & \text{если имеется достаточно убедительный} \\ & \text{набор аргументов ЗА то,} \\ & \text{что объект } o \text{ обладает множеством} \\ & \text{свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

А функцию $\mathrm{C}^+ : \mathbf{F} \times \mathbf{P} \to \mathbf{B}$ правильнее описать следующим образом: для любых $s \in \mathbf{F}$, $p \in \mathbf{P}$

$$\mathrm{C}^+(s,p) = \begin{cases} 1, & \text{если имеется достаточно убедительный} \\ & \text{набор аргументов ЗА то,} \\ & \text{что фрагмент } s \text{ является причиной} \\ & \text{наличия множества свойств } p, \\ 0, & \text{в противном случае.} \end{cases}$$

Аналогично, могут быть неформально описаны функции P^- и C^-. Что понимается под «достаточно убедительным набором аргументов *за*», зависит от конкретной разновидности ДСМ-метода, а иногда, и от конкретной задачи, решаемой ДСМ-системой.

4. ПОЛОЖИТЕЛЬНЫЕ И ОТРИЦАТЕЛЬНЫЕ ПРИМЕРЫ И ГИПОТЕЗЫ

Будем использовать введенные в предыдущем разделе обозначения для универсумов атомов, объектов, фрагментов и т.п.

Объект $o \in \mathbf{O}$ будем называть *положительным примером* для множества свойств $p \in \mathbf{P}$, если $\mathrm{P}^+(o,p) = 1$, т.е. если существует достаточно убедительный набор аргументов *за* то, что o *обладает* множеством свойств p.

Множество всех положительных примеров для набора свойств p будем обозначать через $\mathbf{O}^+(p)$.

Объект $o \in \mathbf{O}$ будем называть *отрицательным примером* для множества свойств $p \in \mathbf{P}$, если $\mathrm{P}^-(o,p) = 1$, т.е. если существует достаточно убедительный набор аргументов *за* то, что o *не обладает* множеством свойств p.

Множество всех отрицательных примеров для набора свойств p будем обозначать через $\mathbf{O}^-(p)$.

Объект $o \in \mathbf{O}$ будем называть *противоречивым примером* для множества свойств $p \in \mathbf{P}$, если он одновременно является и положительным, и отрицательным примером для этого множества свойств, т.е. если $\mathrm{P}^+(o,p) = 1$ и $\mathrm{P}^-(o,p) = 1$. Это возможно лишь в случае достаточно убедительного набора аргументов как *за* наличие, так и *за* отсутствие у объекта o множества свойств p.

Множество всех противоречивых примеров для набора свойств p будем обозначать через $\mathbf{O}^0(p)$.

Объект $o \in \mathbf{O}$ будем называть *неопределенным примером* для множества свойств $p \in \mathbf{P}$, если он не является ни положительным, ни отрицательным примером для этого множества свойств, т.е. если $\mathrm{P}^+(o,p) = 0$ и $\mathrm{P}^-(o,p) = 0$. Это возможно в случае отсутствия достаточно убедительного набора аргументов и *за* наличие, и *за* отсутствие у объекта o множества свойств p.

Множество всех неопределенных примеров для набора свойств p будем обозначать через $\mathbf{O}^\tau(p)$.

Объект $o \in \mathbf{O}$ будем называть *чисто положительным примером* для множества свойств $p \in \mathbf{P}$, если $\mathrm{P}^+(o,p) = 1$ и $\mathrm{P}^-(o,p) = 0$. В этом случае существует достаточно убедительный набор аргументов *за* то, что o *обладает* множеством свойств p и не существует достаточно убедительного набора аргументов *за* то, что o *не обладает* множеством свойств p.

Множество всех чисто положительных примеров для набора свойств p будем обозначать через $\mathbf{O}^{(+)}(p)$.

Объект $o \in \mathbf{O}$ будем называть *чисто отрицательным примером* для множества свойств $p \in \mathbf{P}$, если $\mathrm{P}^-(o,p) = 1$ и $\mathrm{P}^+(o,p) = 0$. В этом случае существует достаточно убедительный набор аргументов *за* то, что o *не обладает* множеством свойств p и не существует достаточно убедительного набора аргументов *за* то, что o *обладает* множеством свойств p.

Множество всех чисто отрицательных примеров для набора свойств p будем обозначать через $\mathbf{O}^{(-)}(p)$.

Очевидно, что между множествами $\mathbf{O}^+(p)$, $\mathbf{O}^-(p)$, $\mathbf{O}^0(p)$, $\mathbf{O}^\tau(p)$, $\mathbf{O}^{(+)}(p)$ и $\mathbf{O}^{(-)}(p)$ имеют место следующие соотношения:

$$\mathbf{O}^0(p) = \mathbf{O}^+(p) \cap \mathbf{O}^-(p),$$
$$\mathbf{O}^{(+)}(p) = \mathbf{O}^+(p) \setminus \mathbf{O}^0(p),$$
$$\mathbf{O}^{(-)}(p) = \mathbf{O}^-(p) \setminus \mathbf{O}^0(p),$$
$$\mathbf{O}^\tau(p) = \mathbf{O} \setminus \left(\mathbf{O}^+(p) \cup \mathbf{O}^-(p)\right),$$
$$\mathbf{O}^\tau(p) = \mathbf{O} \setminus \left(\mathbf{O}^{(+)}(p) \cup \mathbf{O}^{(-)}(p) \cup \mathbf{O}^0(p)\right),$$
$$\mathbf{O} = \mathbf{O}^+(p) \cup \mathbf{O}^-(p) \cup \mathbf{O}^\tau(p),$$
$$\mathbf{O} = \mathbf{O}^{(+)}(p) \cup \mathbf{O}^{(-)}(p) \cup \mathbf{O}^0(p) \cup \mathbf{O}^\tau(p).$$

Термин «пример» мы используем по отношению к объектам в контексте обладания набором целевых свойств. Термин «гипотеза» мы будем использовать по отношению к фрагментам, которые могут быть возможными причинами наличия или отсутствия наборов целевых свойств.

Фрагмент $s \in \mathbf{F}$ будем называть *положительной гипотезой* для множества свойств $p \in \mathbf{P}$, если $C^+(s,p)=1$, т.е. если существует достаточно убедительный набор аргументов *за* то, что s *является возможной причиной наличия* множества свойств p.

Множество всех положительных гипотез для набора свойств p будем обозначать через $\mathbf{F}^+(p)$.

Фрагмент $s \in \mathbf{F}$ будем называть *отрицательной гипотезой* для множества свойств $p \in \mathbf{P}$, если $C^-(s,p)=1$, т.е. если существует достаточно убедительный набор аргументов *за* то, что s *является возможной причиной отсутствия* множества свойств p.

Множество всех отрицательных гипотез для набора свойств p будем обозначать через $\mathbf{F}^-(p)$.

Фрагмент $s \in \mathbf{F}$ будем называть *противоречивой гипотезой* для множества свойств $p \in \mathbf{P}$, если он одновременно является и положительной, и отрицательной гипотезой для этого множества свойств, т.е. если $C^+(s,p)=1$ и $C^-(s,p)=1$. Это возможно лишь в случае достаточно убедительного набора аргументов как *за* то, что s является возможной причиной наличия, так и *за* то, что s является возможной причиной отсутствия множества свойств p.

Множество всех противоречивых гипотез для набора свойств p будем обозначать через $\mathbf{F}^0(p)$.

Фрагмент $s \in \mathbf{F}$ будем называть *неопределенной гипотезой* для множества свойств $p \in \mathbf{P}$, если он не является ни положительной, ни отрицательной гипотезой для этого множества свойств, т.е. если $C^+(s,p)=0$ и $C^-(s,p)=0$. Это возможно в случае отсутствия достаточно убедительного набора аргументов и *за* то, что s является возможной причиной наличия, и *за* то, что s является возможной причиной отсутствия множества свойств p.

Множество всех неопределенных гипотез для набора свойств p будем обозначать через $\mathbf{F}^\tau(p)$.

Фрагмент $s \in \mathbf{F}$ будем называть *чисто положительной гипотезой* для множества свойств $p \in \mathbf{P}$, если $C^+(s,p)=1$ и $C^-(s,p)=0$. В этом случае существует достаточно убедительный набор аргументов *за* то, что s является возможной причиной *наличия* множества свойств p, и не существует достаточно убедительного набора аргументов *за* то, что s является возможной причиной *отсутствия* множества свойств p.

Множество всех чисто положительных гипотез для набора свойств p будем обозначать через $\mathbf{F}^{(+)}(p)$.

Фрагмент $s \in \mathbf{F}$ будем называть *чисто отрицательной гипотезой* для множества свойств $p \in \mathbf{P}$, если $C^-(s,p)=1$ и $C^+(s,p)=0$. В этом случае существует достаточно убедительный набор аргументов *за* то, что s является возможной причиной *отсутствия* множества свойств p, и не существует достаточно убедительного набора аргументов *за* то, что s является возможной причиной *наличия* множества свойств p.

Множество всех чисто отрицательных гипотез для набора свойств p будем обозначать через $\mathbf{F}^{(-)}(p)$.

Между множествами гипотез $\mathbf{F}^+(p)$, $\mathbf{F}^-(p)$, $\mathbf{F}^0(p)$, $\mathbf{F}^\tau(p)$, $\mathbf{F}^{(+)}(p)$ и $\mathbf{F}^{(-)}(p)$ имеют место соотношения, аналогичные соотношениям между соответствующими множествами примеров, а именно:

$$\mathbf{F}^0(p)=\mathbf{F}^+(p)\cap\mathbf{F}^-(p),$$
$$\mathbf{F}^{(+)}(p)=\mathbf{F}^+(p)\setminus\mathbf{F}^0(p),$$
$$\mathbf{F}^{(-)}(p)=\mathbf{F}^-(p)\setminus\mathbf{F}^0(p),$$
$$\mathbf{F}^\tau(p)=\mathbf{F}\setminus\left(\mathbf{F}^+(p)\cup\mathbf{F}^-(p)\right),$$
$$\mathbf{F}^\tau(p)=\mathbf{F}\setminus\left(\mathbf{F}^{(+)}(p)\cup\mathbf{F}^{(-)}(p)\cup\mathbf{F}^0(p)\right),$$
$$\mathbf{F}=\mathbf{F}^+(p)\cup\mathbf{F}^-(p)\cup\mathbf{F}^\tau(p),$$
$$\mathbf{F}=\mathbf{F}^{(+)}(p)\cup\mathbf{F}^{(-)}(p)\cup\mathbf{F}^0(p)\cup\mathbf{F}^\tau(p).$$

5. СХОДСТВА И КЛАСТЕРЫ. СООТВЕТСТВИЯ ГАЛУА

Будем находить возможную причину набора целевых свойств как общую часть (пересечение) некоторого семейства объектов, этим набором свойств обладающих. В работах по ДСМ-методу общую часть объектов называют их *сходством*, т.е. термин «сходство» употребляется в смысле *операции* (пересечения), а не отношения, обладающего свойствами рефлексивности и симметричности (такие отношения называются отношениями *толерантности* [22]).

Здесь нужно отметить следующее обстоятельство: мы допускаем, что *у одного и того же* набора целевых свойств могут быть *различные* причины. Если мы рассмотрим множество объектов, обладающих некоторым набором свойств p, то каждая возможная причина наличия этого свойства порождает свой *кластер* (в один кластер собираются объекты, имеющие общую причину наличия у них свойства p). Заметим, что возможная причина (фрагмент), которая определяется как пересечение некоторого семейства объектов, должна включаться в каждый объект, принадлежащий этому семейству.

Теперь дадим формальные определения двум двойственным понятиям: *сходства* семейства объектов и *кластера* фрагмента.

Пусть $O \subseteq \mathbf{O}$ — некоторое семейство объектов, представленных в виде множеств. *Сходством* этого семейства будем называть фрагмент, равный пересечению множеств, представляющих объекты. Обозна-

чать сходство семейства объектов O будем через O^{Si}. Таким образом, по определению,

$$O^{\mathrm{Si}} = \bigcap_{o \in O} o = \{a \in \mathbf{A} \mid (\forall o \in O)(a \in o)\}. \qquad (1)$$

Пусть $s \in \mathbf{F}$ — некоторый фрагмент, т.е. подмножество универсума $\mathbf{A}$. *Кластером* фрагмента s назовем семейство всех объектов из $\mathbf{O}$, включающих s. Обозначать кластер фрагмента s будем через s^{Cl}. Таким образом, по определению,

$$s^{\mathrm{Cl}} = \{o \in \mathbf{O} \mid s \subseteq o\} = \{o \in \mathbf{O} \mid (\forall a \in s)(a \in o)\}. \qquad (2)$$

Цепочки равенств (1) и (2) мы в дальнейшем рассмотрим более подробно, когда будем говорить о связи ДСМ-метода и анализа формальных понятий.

На рис. 5 дана наглядная иллюстрация определениям сходства семейства объектов и кластера фрагмента. Объекты на этом рисунке изображены в виде прямоугольников. Фрагмент может быть любой фигурой (не обязательно прямоугольником). На левой части рисунка хорошо видно, что семейство объектов, порождающих сходство, не обязано включать *все* объекты, которые порождают соответствующее пересечение. На правой части рисунка видно, что фрагмент, порождающий кластер, не обязательно является *пересечением* всех элементов кластера.

Определив сходство семейства объектов и кластер фрагмента, мы, фактически, определили два отображения: ${}^{\mathrm{Si}}: 2^{\mathbf{O}} \to 2^{\mathbf{A}}$ и ${}^{\mathrm{Cl}}: 2^{\mathbf{A}} \to 2^{\mathbf{O}}$. Заметим, что на семействах множеств $2^{\mathbf{O}}$ и $2^{\mathbf{A}}$ теоретико-множественное включение $\subseteq$ является отношение частичного порядка. Пара отображений $\langle {}^{\mathrm{Si}}, {}^{\mathrm{Cl}} \rangle$ обладает некоторым набором свойств, описывающим связь этих отображений с частичным порядком $\subseteq$ на семействах множеств $2^{\mathbf{O}}$ и $2^{\mathbf{A}}$.

Для любых $O, O_1, O_2 \in 2^{\mathbf{O}}$ и любых $s, s_1, s_2 \in 2^{\mathbf{A}}$ верно:

(i) $O_1 \subseteq O_2$ влечет $O_2{}^{\mathrm{Si}} \subseteq O_1{}^{\mathrm{Si}}$ (антитонность отображения ${}^{\mathrm{Si}}$),

(ii) $s_1 \subseteq s_2$ влечет $s_2{}^{\mathrm{Cl}} \subseteq s_1{}^{\mathrm{Cl}}$ (антитонность отображения ${}^{\mathrm{Cl}}$),

(iii) $O \subseteq O^{\mathrm{Si}\circ\mathrm{Cl}}$ (экстенсивность композиции ${}^{\mathrm{Si}\circ\mathrm{Cl}}$),

(iv) $s \subseteq s^{\mathrm{Cl}\circ\mathrm{Si}}$ (экстенсивность композиции ${}^{\mathrm{Cl}\circ\mathrm{Si}}$).

Справедливость утверждений (i)–(iv) означает, что пара отображений $\langle {}^{\mathrm{Si}}, {}^{\mathrm{Cl}} \rangle$ является соответствием Галуа в смысле определения из [23].

Понятие *соответствия Галуа* (Galois connection) относится к частично упорядоченным множествам. В [23] это понятие определяется следующим образом:

Пусть $\langle M, \leq \rangle$ и $\langle M', \leq' \rangle$ — два частично упорядоченных множества. Упорядоченная пара отображений $\langle \varphi, \psi \rangle$, где $\varphi: M \to M'$, $\psi: M' \to M$, называется *соответствием Галуа* (между этими частично упорядоченными множествами), если для любых $a, b \in M$ и любых $a', b' \in M'$ верно:

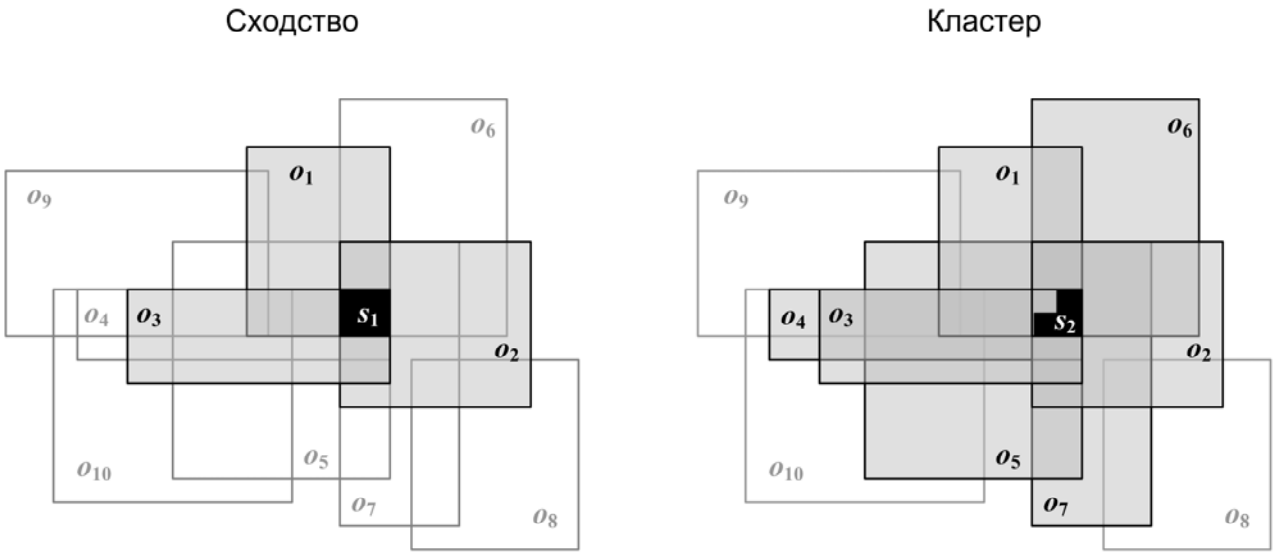
(a) $a \leq b$ влечет $b^{\varphi} \leq' a^{\varphi}$ (антитонность отображения φ),

(b) $a' \leq' b'$ влечет $b'^{\psi} \leq a'^{\psi}$ (антитонность отображения ψ),

(c) $a \leq a^{\varphi\circ\psi}$ (экстенсивность композиции $\varphi \circ \psi$),

(d) $a' \leq' a'^{\psi\circ\varphi}$ (экстенсивность композиции $\psi \circ \varphi$).

Здесь мы используем постфиксную форму представления значения функции, т.е. вместо $\varphi(a)$ пишем a^{φ} (аналогично, вместо $\psi(a')$ пишем a'^{ψ}).



$O = \{o_1, o_2, o_3\}$, $\quad s_1 = O^{\mathrm{Si}} = o_1 \cap o_2 \cap o_3$ $\qquad$ $s_2{}^{\mathrm{Cl}} = \{o \in O \mid s_2 \subseteq o\} = \{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}$

Рис. 5. Сходство и кластер

Можно показать, что пара антитонных отображений $\langle\varphi,\psi\rangle$, где $\varphi: M \to M'$, $\psi: M' \to M$, является соответствием Галуа (между частично упорядоченными множествами $\langle M,\leq\rangle$ и $\langle M',\leq'\rangle$) в том и только в том случае, если для любых $a \in M$ и $a' \in M'$:

$a' \leq' a^{\varphi}$ тогда и только тогда, когда $a \leq a'^{\psi}$.

Этот факт позволяет дать соответствию Галуа эквивалентное определение, несколько более короткое, чем определение из [23].

Применяя приведенное выше утверждение к соответствию Галуа $\langle^{\mathrm{Si}},^{\mathrm{Cl}}\rangle$, получаем, что для любых $O \in 2^{\mathbf{O}}$ и $s \in 2^{\mathbf{A}}$:

$$O \subseteq s^{\mathrm{Cl}} \text{ тогда и только тогда, когда } s \subseteq O^{\mathrm{Si}}. \quad (3)$$

Рис. 5 с $O = \{o_1, o_2, o_3\}$ и $s = s_2$ является наглядной иллюстрацией соотношения (3). Однако нас прежде всего будет интересовать тот случай, когда в соотношении (3) включения будут заменены на равенства. Для того чтобы рассматривать этот случай более содержательно, введем следующее определение:

Пусть $O \in 2^{\mathbf{O}}$, $s \in 2^{\mathbf{A}}$. Будем говорить, что пара $\langle O,s\rangle$ удовлетворяет *условию исчерпываемости* относительно множества $\mathbf{O}$, если справедливы равенства:

$$O = s^{\mathrm{Cl}}, \quad (4)$$

$$s = O^{\mathrm{Si}}. \quad (5)$$

Равенства (4) и (5), а также (1) и (2), потребуются нам при обсуждении связи ДСМ-метода с анализом формальных понятий. Условие исчерпываемости будет необходимо при определении так называемых решающих предикатов ДСМ-метода.

Связь конструкций ДСМ-метода с соответствиями Галуа обнаружил С.О. Кузнецов (см., например, [24]).

6. КАНДИДАТЫ В ВОЗМОЖНЫЕ ПРИЧИНЫ

Пусть s — фрагмент, т.е. $s \in \mathbf{F} = 2^{\mathbf{A}}$. Будем говорить, что s является *кандидатом в возможные причины наличия набора свойств* $p \in \mathbf{P}$, если существует $O \subseteq \mathbf{O}^{(+)}(p)$ такое, что выполняются следующие условия:

(i) пара $\langle O,s\rangle$ удовлетворяет условию исчерпываемости относительно $\mathbf{O}^{(+)}(p)$;

(ii) $s \neq \varnothing$;

(iii) $|O| \geq 2$.

Здесь через $|O|$ обозначена мощность множества O.

Приведем некоторые аргументы в пользу именно такого определения кандидата в возможные причины. Рассмотрим сначала условие (i). Напомним, что условие исчерпываемости предполагает справедливость равенства (5). Значит, $s = O^{\mathrm{Si}}$, т.е. s является сходством (пересечением) множества объектов O.

ДСМ-метод основан на предположении о том, что возможная причина *наличия* набора свойств должна определяться как нечто общее (сходство) некоторого множества *положительных* примеров для этого набора свойств. В данном случае мы поступаем более жестко и отбираем только *чисто положительные* примеры.

Условие исчерпываемости кроме равенства (5) включает также равенство (4). Значит, $O = s^{\mathrm{Cl}}$. Требование справедливости равенства (4) обусловлено менее очевидными соображениями, связанными с эффективностью алгоритмов порождения всех различных пересечений.

Заметим, что один и тот же фрагмент s может быть получен как пересечение различных (иногда даже дизъюнктных) семейств объектов, но существует *единственное* семейство O, для которого пара $\langle O,s\rangle$ удовлетворяет условию исчерпываемости относительно некоторого заранее выбранного универсума объектов. Этот факт позволяет определять такие алгоритмы построения всех (разных) пересечений, во время работы которых каждое такое пересечение порождается ровно один раз, т.е. эти алгоритмы будут иметь сложность *линейную относительно результата* (количества порожденных пересечений). Разумеется, в наихудшем случае даже такие алгоритмы имеют сложность *экспоненциальную относительно размера исходных данных* (количества объектов). Однако данные реальных задач таковы, что наихудший случай для них практически невозможен.

Условие (ii) требует, чтобы кандидат в возможные причины был непуст. Это условие можно объяснить исходя из элементарного здравого смысла. Отсутствие чего бы то ни было не может быть возможной причиной наличия непустого набора свойств.

Условие (iii) связано с классической формулировкой метода сходства (method of agreement) Д.С. Милля [6], в котором присутствуют «… два или более случаев рассматриваемого явления…». Каждому такому случаю соответствует объект из семейства O.

Рассмотрим теперь вопрос о том, как можно изменить определение кандидата в возможные причины с целью его адаптации для решения собственных прикладных задач. Сначала необходимо описать общую структуру этого определения. *Оно содержит преамбулу и три условия.* В преамбуле говорится о том, что существует семейство объектов O, каким-то образом связанное с фрагментом s, который рассматривается в качестве кандидата в возможные причины. Это семейство объектов O фактически порождает фрагмент s. В работах по ДСМ-методу такое семейство объектов может быть названо по-разному: орбита, множество образующих, множество порождающих, множество родителей (фрагмента s). В данной работе мы для краткости будем называть семейство O *родителем* фрагмента s.

В условии (i) раскрывается характер связи между фрагментом s и его родителем O. В условии (ii) накладываются ограничения на сам фрагмент s. В условии (iii) ограничения накладываются на родителя O.

Предположим, что мы собираемся сохранить общую структуру этого определения. Вопрос о том, как можно изменить преамбулу и условие (i) рассмотрим

позже. А сейчас рассмотрим возможные модификации условий (ii) и (iii).

Модификация условия (ii) может потребоваться для того, чтобы более точно отразить характер предметной области. Предположим, что экспертам предметной области известен некоторый набор фрагментов, которые ни при каких обстоятельствах не могут быть возможными причинами множества целевых свойств p. Обозначим этот набор «незначащих» фрагментов через N. Будем предполагать, что $\varnothing \in N$. Тогда условие (ii) будет записано в виде

$$s \notin N.$$

Теперь предположим, что множество атомов $\mathbf{A}$ содержит подмножество S «незначащих элементов». Примерами таких элементов могут служить стоп-слова, от которых принято избавляться при анализе текстов. Логично предположить, что фрагмент, полностью состоящий из «незначащих элементов» сам будет «незначащим» и не может рассматриваться в качестве кандидата в возможные причины. В этом случае условие (ii) представляется в виде

$$s \not\subseteq S.$$

Возможна также ситуация, когда специалистов в предметной области, прежде всего, будут интересовать такие возможные причины, размер которых находится в определенном диапазоне. Тогда условие (ii) будет записано в виде

$$a \le |s| \le b,$$

где $0 < a < b$.

Теперь рассмотрим возможные варианты модификации условия (iii). Во-первых, может быть изменено пороговое значение мощности родителя. Например, условие (iii) может быть переписано в виде

$$|O| \ge 10.$$

Тогда от фрагмента s будет требоваться, чтобы он являлся пересечением по крайней мере 10 различных чисто положительных примеров.

Во-вторых, пороговое значение мощности может зависеть от количества чисто положительных примеров или от общего количества примеров. В этом случае, мы будем учитывать частоту встречаемости фрагмента в рассматриваемых примерах. Например, можно записать условие (iii) в виде

$$|O| \ge 0{,}1 \cdot \left|\mathbf{O}^{(+)}(p)\right|.$$

Тогда фрагмент s должен быть общей частью по крайней мере 10% чисто положительных примеров.

В некоторых случаях (например, при анализе текстов) наиболее полезными могут оказаться фрагменты, которые встречаются не слишком часто, но и не слишком редко. Для таких случаев условие (iii) может быть переписано в виде

$$a \cdot \left|\mathbf{O}^{(+)}(p)\right| \le |O| \le b \cdot \left|\mathbf{O}^{(+)}(p)\right|,$$

где $0 < a < b < 1$.

Что касается преамбулы и условия (i), то наиболее естественной их модификацией была бы замена множества *чисто* положительных примеров $\mathbf{O}^{(+)}(p)$ на множество положительных примеров $\mathbf{O}^{+}(p)$. Эта замена (вместе с некоторыми изменениями стратегии ДСМ-рассуждений) позволила бы рассматривать аргументы «за» и аргументы «против» совершенно независимо друг от друга, оставляя последнее решающее слово эксперту.

Возможно и более радикальное изменение условия (i), включающее замену условия исчерпываемости на более простое условие. Автор настоящей статьи неоднократно высказывал мысль о необходимости «отделить в ДСМ-методе логику от комбинаторики» и отдельно рассматривать вопросы о *порождении сходств* (пересечений множеств, представляющих объекты) и *проверке этих сходств на соответствие условиям*, выраженным средствами логики предикатов первого порядка. Это идея не была поддержана членами сообщества исследователей, занимающихся ДСМ-методом, и впоследствии автор от нее отказался.

Следует отметить, что с условием исчерпываемости связаны весьма непростые и тонкие вопросы математического, концептуального и технологического характера. Их подробное обсуждение в относительно популярной работе автору кажется неуместным. Однако некоторые замечания все-таки необходимо сделать.

В условие исчерпываемости входит равенство

$$s = O^{\mathrm{Si}} = \bigcap_{o \in O} o.$$

Если попытаться выразить его непосредственно через бинарную операцию сходства (пересечения), то в результете будет получена формула переменной длины, содержащая квантор по натуральным числам

$$\exists n \exists o_1 \ldots \exists o_n \left(s = o_1 \cap \ldots \cap o_n\right). \qquad (6)$$

Эта формула включается в формулу, описывающую так называемый решающий предикат ДСМ-метода (фактически, в определение кандидата в возможные причины), встречающуюся во многих работах по ДСМ-методу. В статье В.К.Финна [9] определение решающего предиката сопровождается подробными комментариями.

Однако, формула (6) не является формулой логики предикатов первого порядка. Это осложняет представление алгоритмов ДСМ-метода средствами логического программирования, хотя для ДСМ-метода, использующего рассуждения по формальным правилам, аппарат логического программирования является вполне естественным.

Теоретическая основа для применения логического программирования в ДСМ-методе была заложена в работах Д.В. Виноградова [25, 26]; наиболее известная практическая реализация ДСМ-метода на языке логического программирования описана в работе М.А. Михеенковой и Т.Л. Феофановой [27].

Д.В. Виноградовым в [28] было доказано, что формулу (6) (а значит и условие исчерпываемости) в контексте некоторой теории ДСМ-метода можно представить с помощью формулы логики предикатов первого порядка. Это весьма интересный теоретический результат, но, к сожалению, первопорядковое

представление для формулы (6) не очень удобно для практического использования.

Заканчивая отступление, посвященное условию исчерпываемости, и возвращаясь к основной теме данного раздела, необходимо отметить, что мы рассмотрели лишь малую часть вариантов определения кандидата в возможные причины. Подводя итог обсуждению вопроса о модификациях этого определения, можно сказать, что различные его варианты позволяют достаточно тонко настраивать ДСМ-подобную систему, приспосабливая ее к решению конкретных задач.

Определение *кандидата в возможные причины отсутствия набора свойств* $p \in \mathbf{P}$ является *двойственным* определению кандидата в возможные причины наличия этого набора свойств, данному в начале текущего раздела. Требуется просто заменить $\mathbf{O}^{(+)}(p)$ на $\mathbf{O}^{(-)}(p)$. Возможные модификации нового определения будут аналогичны возможным модификациям исходного определения.

7. РЕШАЮЩИЕ ПРЕДИКАТЫ. ПРАВИЛА ИНДУКЦИИ

Пусть $s \in \mathbf{F}$, $p \in \mathbf{P}$. Тот факт, что s является кандидатом в возможные причины наличия множества свойств p, будем обозначать через $\mathrm{M}^{+}(s,p)$. Тот факт, что s является кандидатом в возможные причины отсутствия множества свойств p, будем обозначать через $\mathrm{M}^{-}(s,p)$.

Условия $\mathrm{M}^{+}(s,p)$ и $\mathrm{M}^{-}(s,p)$ будем называть *положительным и отрицательным решающими предикатами индукции*, соответственно. Решающие предикаты позволяют дать краткую формулировку правилам индукции, которые мы будем интерпретировать как правила вычисления значений булевых функций C^{+} и C^{-}. Заметим, что формулировка правил в данной статье будет существенно проще традиционной формулировки правил ДСМ-метода. Это связано с тем обстоятельством, что в этой статье используется расширенный набор базовых функций (их четыре: P^{+}, P^{-}, C^{+} и C^{-}), а не две, как в традиционном ДСМ-методе.

Оператор присваивания будем обозначать через «$:=$». Это достаточно традиционное обозначение для оператора присваивания, оно должно восприниматься однозначно, кроме того это обозначение (хотя и немного) отличается от знака равенства.

Итак, мы имеем два правила индукции:

$$\frac{\mathrm{C}^{+}(s,p)=0, \mathrm{M}^{+}(s,p)}{\mathrm{C}^{+}(s,p) := 1} \qquad \text{(I+)}$$

$$\frac{\mathrm{C}^{-}(s,p)=0, \mathrm{M}^{-}(s,p)}{\mathrm{C}^{-}(s,p) := 1} \qquad \text{(I–)}$$

Этим правилам будут соответствовать условные операторы:

(I+) **if** $\mathrm{C}^{+}(s,p)=0$ **and** $\mathrm{M}^{+}(s,p)$ **then** $\mathrm{C}^{+}(s,p) := 1$;

(I–) **if** $\mathrm{C}^{-}(s,p)=0$ **and** $\mathrm{M}^{-}(s,p)$ **then** $\mathrm{C}^{-}(s,p) := 1$.

Подчеркнем, что эти правила и условные операторы имеют отношение к математической модели, а не к конкретной программной реализации.

Фрагменты (элементы множества $\mathbf{F}$) заранее неизвестны, они порождаются в процессе работы ДСМ-системы, и мы полагаем, что, по умолчанию, значение функций C^{+} и C^{-} для любого нового (только что построенного) фрагмента равно 0. Так что условия $\mathrm{C}^{+}(s,p)=0$ и $\mathrm{C}^{-}(s,p)=0$ можно не проверять, они всегда будут выполнены после того, как был построен фрагмент s.

Проверка условия $\mathrm{M}^{+}(s,p)$ сводится к проверке условий (i)–(iii) из определения кандидата в возможные причины наличия набора свойств p. Самое сложное из них — условие (i) не нужно проверять, так алгоритмы порождения фрагментов, используемые в ДСМ-методе, строят только такие фрагменты, которые заведомо удовлетворяют условию (i). Остается только проверка сравнительно простых условий (ii) и (iii). То же самое можно сказать о проверке условия $\mathrm{M}^{-}(s,p)$.

Из правил (I+) и (I–) можно получить следствия, связывающие решающие предикаты M^{+} и M^{-} с принадлежностью фрагмента одному из множеств гипотез: $\mathbf{F}^{(+)}(p)$, $\mathbf{F}^{(-)}(p)$, $\mathbf{F}^{0}(p)$ и $\mathbf{F}^{\tau}(p)$. Эти следствия можно интерпретировать как правила «перетаскивания» гипотезы из множества неопределенных гипотез $\mathbf{F}^{\tau}(p)$ в одно из перечисленных выше множеств гипотез. В частности, гипотеза может остаться в множестве $\mathbf{F}^{\tau}(p)$. Приведем список этих правил:

$$\text{(Ind +)} \quad \frac{s \in \mathbf{F}^{\tau}(p), \mathrm{M}^{+}(s,p), \neg\mathrm{M}^{-}(s,p)}{s \in \mathbf{F}^{(+)}(p)}$$

$$\text{(Ind –)} \quad \frac{s \in \mathbf{F}^{\tau}(p), \mathrm{M}^{-}(s,p), \neg\mathrm{M}^{+}(s,p)}{s \in \mathbf{F}^{(-)}(p)}$$

$$\text{(Ind 0)} \quad \frac{s \in \mathbf{F}^{\tau}(p), \mathrm{M}^{+}(s,p), \mathrm{M}^{-}(s,p)}{s \in \mathbf{F}^{0}(p)}$$

$$\text{(Ind }\tau\text{)} \quad \frac{s \in \mathbf{F}^{\tau}(p), \neg\mathrm{M}^{+}(s,p), \neg\mathrm{M}^{-}(s,p)}{s \in \mathbf{F}^{\tau}(p)} \; .$$

Те читатели, которые знакомы с ДСМ-методом, легко поймут, что приведенные выше правила «перетаскивания» суть сформулированные несколько иначе классические *правила простого ДСМ-метода без запрета на контрпример*.

Автор не ставил себе задачу описать на теоретико-множественном языке все используемые в настоящее время разновидности ДСМ-метода. Задача была более скромная: показать, что теоретико-множественное описание основных понятий ДСМ-метода, не использующее аппарат неклассических логик, в принципе возможно. Поэтому никакие другие версии ДСМ-метода в этой статье рассматриваться не будут.

8. РЕШАЮЩИЕ ПРЕДИКАТЫ. ПРАВИЛА АНАЛОГИИ

Правила аналогии позволяют делать предсказания о наличии или отсутствии у объектов набора целевых свойств. Общая идея этих правил такова:

- если объект содержит причину наличия набора свойств p, то он *должен* обладать этим набором свойств,
- если же объект содержит причину отсутствия набора свойств p, то он *не должен* обладать этим набором свойств.

Разумеется, противоречия в этом случае возможны, но они корректно обрабатываются.

Дадим сначала некоторые определения. Будем говорить, что объект $o \in \mathbf{O}$ является *кандидатом на наличие набора свойств* $p \in \mathbf{P}$, если существует $s \in \mathbf{F}^{(+)}(p)$ такое, что $s \subseteq o$. Аналогично, будем говорить, что объект $o \in \mathbf{O}$ является *кандидатом на отсутствие набора свойств* $p \in \mathbf{P}$, если существует $s \in \mathbf{F}^{(-)}(p)$ такое, что $s \subseteq o$.

Тот факт, что o является кандидатом на *наличие* набора свойств p, будем обозначать через $\Pi^+(o,p)$. Тот факт, что o является кандидатом на *отсутствие* набора свойств p, будем обозначать через $\Pi^-(o,p)$.

Условия $\Pi^+(o,p)$ и $\Pi^-(o,p)$ будем называть *положительным и отрицательным решающим предикатом аналогии, соответственно*. Решающие предикаты позволяют дать краткую формулировку правилам аналогии, которые мы будем интерпретировать как правила вычисления значений булевых функций P^+ и P^-. Использование решающих предикатов делает формулировку правил аналогии очень похожей на формулировку правил индукции. Приведем эти правила:

$$\frac{\mathrm{P}^+(o,p)=0, \Pi^+(o,p)}{\mathrm{P}^+(o,p) \coloneqq 1} \qquad (\mathrm{A}+)$$

$$\frac{\mathrm{P}^-(o,p)=0, \Pi^-(o,p)}{\mathrm{P}^-(o,p) \coloneqq 1} \qquad (\mathrm{A}-)$$

Этим правилам будут соответствовать условные операторы:

(A+) **if** $\mathrm{P}^+(o, p) = 0$ **and** $\Pi^+(o, p)$ **then** $\mathrm{P}^+(o, p) = 1$;

(A–) **if** $\mathrm{P}^-(o, p) = 0$ **and** $\Pi^-(o, p)$ **then** $\mathrm{P}^-(o, p) = 1$;

Как и в случае правил индукции, можно получить следствия из правил (A+) и (A–), которые можно интерпретировать как правила «перетаскивания» объекта из множества неопределенных примеров $\mathbf{O}^\tau(p)$ в одно из множеств: $\mathbf{O}^{(+)}(p)$, $\mathbf{O}^{(-)}(p)$, $\mathbf{O}^0(p)$ или сохранения его в множестве $\mathbf{O}^\tau(p)$. Приведем список этих правил:

$$(\mathrm{An}+) \quad \frac{o \in \mathbf{O}^\tau(p), \Pi^+(o,p), \neg\Pi^-(o,p)}{s \in \mathbf{O}^{(+)}(p)}$$

$$(\mathrm{An}-) \quad \frac{o \in \mathbf{O}^\tau(p), \Pi^-(o,p), \neg\Pi^+(o,p)}{s \in \mathbf{O}^{(-)}(p)}$$

$$(\mathrm{An}\ 0) \quad \frac{o \in \mathbf{O}^\tau(p), \Pi^+(o,p), \Pi^-(o,p)}{o \in \mathbf{O}^0(p)}$$

$$(\mathrm{An}\ \tau) \quad \frac{o \in \mathbf{O}^\tau(p), \neg\Pi^+(o,p), \neg\Pi^-(o,p)}{s \in \mathbf{O}^\tau(p)}\ .$$

9. РАБОТА ДСМ-СИСТЕМЫ. ИТЕРАЦИЯ ПРИМЕНЕНИЯ ПРАВИЛ

Ядро ДСМ-метода составляют две фазы:

Фаза индукции. С помощью правил индукции формируются гипотезы о возможных причинах набора целевых свойств. С точки зрения математической модели, фаза индукции состоит в вычислении значений функций C^+ и C^-. Но не нужно забывать, что в этой же фазе один из аргументов этих функций — фрагмент s — только формируется. Алгоритм построения множества фрагментов, используемых в качестве аргументов функций C^+ и C^-, вносит наибольший вклад в вычислительную сложность процедур ДСМ-метода. Построенные фрагменты образуют множества положительных и отрицательных гипотез (возможных причин наличия и возможных причин отсутствия набора целевых свойств, соответственно). Напомним, что положительная гипотеза находится как сходство (пересечение) положительных примеров (объектов, обладающих набором целевых свойств), а отрицательная гипотеза находится как сходство (пересечение) отрицательных примеров (объектов, не обладающих набором целевых свойств).

Фаза аналогии. С помощью правил аналогии формируются гипотезы о наличии или отсутствии набора целевых свойств. С точки зрения математической модели происходит вычисление значений функций P^+ и P^-. Никаких новых объектов при этом не строится но в соответствии с определениями из раздела 4 изменяются множества примеров. Множество неопределенных примеров сужается. Множества положительных и отрицательных примеров расширяются.

После формирования гипотез о наличии или отсутствии набора целевых свойств Этап I ДСМ-метода (в смысле статьи [9]) может быть закончен. Это происходит в случае так называемого *одношагового* ДСМ-метода. Но, поскольку у нас изменилось соотношение между положительными, отрицательными и неопределенными примерами, мы можем попробовать еще раз применить процедуры индукции и аналогии. В этом случае ядро представляет собой циклическую процедуру, а соответствующий вариант ДСМ-метода называется *итеративным*.

Выход из цикла осуществляется при выполнении *условия насыщения*, когда множества примеров перестают изменяться. Поскольку множество неопределенных примеров может только уменьшаться, а множества положительных и отрицательных примеров — только расти, и все эти множества — конечны, рано или поздно условие насыщения должно стать выполненным.

В случае итеративного ДСМ-метода завершение Этапа I происходит после выполнения условия насыщения и наступает Этап II — применение абдукции (проверка условия каузальной полноты). В этой статье Этап II не рассматривается.

Среди исследователей, занимающихся ДСМ-методом, нет единства в вопросе о необходимости итераций в ядре ДСМ-метода. Существуют как сторонники, так и убежденные противники итераций.

Противники итераций утверждают, что надежность новых примеров, полученных с помощью правил аналогии, достаточно низка; повторяя применение правил, мы с каждым шагом уменьшаем правдоподобие получаемых гипотез и, в конце концов, получаем гипотезы, не имеющие никакого отношения к исследуемому набору целевых свойств. Для обоснования этой позиции иногда приводится (искусственный) пример из работы [29] (см. также [5, гл. 5]), в которой доказывается существование сколь угодно длинных последовательностей непересекающихся гипотез.

Позиция автора настоящей статьи по этому вопросу такова:

- математического доказательства того факта, что итерация *всегда* бесполезна, — нет;
- случаи, когда итерация оказывается бесполезной, — возможны;
- для экспериментального доказательства бесполезности (как и полезности) итерации — недостаточно данных;
- можно привести (искусственный) пример, демонстрирующий возможность получения с помощью итерации процедур индукции и аналогии новых содержательных гипотез.

Завершим этот раздел описанием упомянутого выше примера. Идея этого примера такова:

- в качестве объектов берем множества ключевых слов;
- единственное целевое свойство — иметь отношение к оружию;
- в качестве отрицательных примеров берутся множества ключевых слов, имеющих отношение к сельскохозяйственным орудиям.

Порождение гипотез и примеров демонстрирует эволюцию вооружений и сельскохозяйственной техники. Работа процедур ДСМ-метода представлена в табл. 1.

В этом примере неопределенными являются те гипотезы, которые еще не построены, т.е. не получено соответствующее пересечение.

10. ДСМ-МЕТОД И АНАЛИЗ ФОРМАЛЬНЫХ ПОНЯТИЙ

Основополагающей по анализу формальных понятий (formal concept analysis) считается работа Р. Вилле [30]. Наиболее основательное изложение результатов этого направления исследований содержится в книге [31]. ДСМ-метод и анализ формальных понятий появились примерно в одно время и около десяти лет развивались независимо друг от друга.

Анализ формальных понятий позиционировался как ветвь прикладной алгебры (более конкретно — теории решеток). Исследователей в этой области интересовали, прежде всего, алгебраические и алгоритмические вопросы. Проблемы практического применения теоретических результатов не были для данного направления центральными, хотя иногда основоположники анализа формальных понятий занимались и прикладными проблемами (см., например, [32]).

ДСМ-метод позиционировался, прежде всего, как система технологий интеллектуального анализа данных, предназначенная для решения практических задач в разных предметных областях: фармакологии, медицине, социологии и т.п. В ранних работах по ДСМ-методу можно выделить два направления:

- исследования неклассических логик, являющихся фундаментом для описания процедур ДСМ-метода (наиболее типичной работой такого рода является статья [33]),
- прикладные исследования, посвященные применению ДСМ-метода в различных предметных областях (например, работы [34-36]).

Публикаций, посвященных проектированию эффективных алгоритмов, в первое десятилетие исследования ДСМ-метода было сравнительно немного. Среди немногих исключений следует отметить, например, статью [37]. В это же время в рамках анализа формальных понятий велась достаточно интенсивная работа по описанию разнообразных алгоритмов порождения всех формальных понятий для заданного формального контекста.

Связь между ДСМ-методом и анализом формальных понятий обнаружил С.О. Кузнецов — автор одного из наиболее эффективных алгоритмов ДСМ-метода — алгоритма «Замыкай-по-Одному» [38]. После обнаружения этой связи в ДСМ-системах стали широко использоваться алгоритмы, разработанные для порождения формальных понятий (подробный анализ таких алгоритмов имеется в работе С.О. Кузнецова и С.А. Объедкова [39]). Особенно популярным стал алгоритм Норриса [40], который в настоящее время является стандартом де факто для ДСМ-систем.

Прежде чем объяснять связь между ДСМ-методом и анализом формальных понятий (АФП) введем основные термины, относящиеся к анализу формальных понятий.

Формальным контекстом будем называть упорядоченную тройку $\langle G, M, I \rangle$, где G — непустое *множество объектов*, M — непустое *множество признаков*, $I \subseteq G \times M$, т.е. I задает некоторое соответствие между элементами множеств G и M.

Тот факт, что $\langle g, m \rangle \in I$ будем, как обычно, обозначать через $g\, I\, m$. Если $g\, I\, m$, то будем говорить, что *объект g обладает признаком m*.

Пусть A — множество объектов, т.е. $A \subseteq G$. *Содержанием* (*интенсионалом*) множества объектов A будем называть множество признаков, которыми обладают все объекты из A. Интенсионал множества A будем обозначать через A^{In}. По определению,

$$A^{\mathrm{In}} = \{ m \in M \mid (\forall g \in A)(g\, I\, m) \}. \qquad (7)$$

Таблица 1

Демонстрация возможностей итерации процедур индукции и аналогии

Номер шага	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Фаза	С	И	А	И	А	И	А	И	А	И	А	И	А	И	А
Объект															
{лук, меч},	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
{лук, копье},	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
{меч, копье},	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
{лук, арбалет},			+	+	+	+	+	+	+	+	+	+	+	+	+
{меч, арбалет},			+	+	+	+	+	+	+	+	+	+	+	+	+
{арбалет, шпага},					+	+	+	+	+	+	+	+	+	+	+
{меч, шпага},			+	+	+	+	+	+	+	+	+	+	+	+	+
{шпага, мушкет},							+	+	+	+	+	+	+	+	+
{шпага, пистолет},							+	+	+	+	+	+	+	+	+
{арбалет, мушкет},					+	+	+	+	+	+	+	+	+	+	+
{арбалет, пистолет},					+	+	+	+	+	+	+	+	+	+	+
{мушкет, ружье},									+	+	+	+	+	+	+
{пистолет, ружье},									+	+	+	+	+	+	+
{ружье, винтовка},											+	+	+	+	+
{пистолет, винтовка},									+	+	+	+	+	+	+
{винтовка, пулемет},													+	+	+
{пистолет, пулемет},									+	+	+	+	+	+	+
{пистолет, револьвер},									+	+	+	+	+	+	+
{винтовка, револьвер},													+	+	+
{мотыга, серп},	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
{серп, соха},	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
{соха, мотыга},	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
{серп, коса},			–	–	–	–	–	–	–	–	–	–	–	–	–
{мотыга, коса},			–	–	–	–	–	–	–	–	–	–	–	–	–
{мотыга, плуг},			–	–	–	–	–	–	–	–	–	–	–	–	–
{соха, плуг},			–	–	–	–	–	–	–	–	–	–	–	–	–
{коса, косилка},					–	–	–	–	–	–	–	–	–	–	–
{плуг, косилка},					–	–	–	–	–	–	–	–	–	–	–
{плуг, трактор},					–	–	–	–	–	–	–	–	–	–	–
{косилка, трактор}.							–	–	–	–	–	–	–	–	–
Фрагмент															
{лук},		+	+	+	+	+	+	+	+	+	+	+	+	+	+
{копье},		+	+	+	+	+	+	+	+	+	+	+	+	+	+
{меч},		+	+	+	+	+	+	+	+	+	+	+	+	+	+
{арбалет},				+	+	+	+	+	+	+	+	+	+	+	+
{шпага},						+	+	+	+	+	+	+	+	+	+
{мушкет},								+	+	+	+	+	+	+	+
{пистолет},								+	+	+	+	+	+	+	+
{ружье},										+	+	+	+	+	+
{винтовка},												+	+	+	+
{пулемет},														+	+
{револьвер},														+	+
{мотыга},		–	–	–	–	–	–	–	–	–	–	–	–	–	–
{серп},		–	–	–	–	–	–	–	–	–	–	–	–	–	–
{соха},		–	–	–	–	–	–	–	–	–	–	–	–	–	–
{коса},				–	–	–	–	–	–	–	–	–	–	–	–
{плуг}				–	–	–	–	–	–	–	–	–	–	–	–
{косилка},						–	–	–	–	–	–	–	–	–	–
{трактор},								–	–	–	–	–	–	–	–

Условные обозначения: И – индукция; А — аналогия; С — «старт» (начальное состояние); «+» — положительный(ая) пример (гипотеза); «–» — отрицательный(ая) пример (гипотеза); пустая клетка — неопределенный(ая) пример (гипотеза).

Пусть B — множество признаков, т.е. $B \subseteq M$. *Объемом* (*экстенсионалом*) множества признаков B будем называть множество объектов, которые обладают всеми признаками из B. Экстенсионал множества B будем обозначать через B^{Ex}. По определению,

$$B^{\mathrm{Ex}} = \{g \in G \mid (\forall m \in B)(g\, I\, m)\}. \quad (8)$$

Введя понятия интенсионала и экстенсионала, мы, фактически, определили два отображения: $^{\mathrm{In}}: 2^G \to 2^M$ и $^{\mathrm{Ex}}: 2^M \to 2^G$. Можно показать, что пара отображений $\langle {}^{\mathrm{In}}, {}^{\mathrm{Ex}} \rangle$ является соответствием Галуа (между частично упорядоченными множествами $\langle 2^G, \subseteq \rangle$ и $\langle 2^M, \subseteq \rangle$). Т.е. мы получили ситуацию, достаточно похожую на рассмотренную в разделе 5, где были определены отображения $^{\mathrm{Si}}: 2^{\mathbf{O}} \to 2^{\mathbf{A}}$ и $^{\mathrm{Cl}}: 2^{\mathbf{A}} \to 2^{\mathbf{O}}$, такие, что пара $\langle {}^{\mathrm{Si}}, {}^{\mathrm{Cl}} \rangle$ является соответствием Галуа (между частично упорядоченными множествами $\langle 2^{\mathbf{O}}, \subseteq \rangle$ и $\langle 2^{\mathbf{A}}, \subseteq \rangle$).

Для простоты будем отождествлять объект g с множеством признаков, которыми этот объект обладает. Тогда вместо $g\, I\, m$ (g обладает признаком m) мы можем писать $m \in g$.

Интенсионал множества объектов A в этом случае будет состоять из всех признаков, общих для объектов из A (т.е. будет пересечением объектов из A). Тогда равенство (7) может быть переписано следующим образом:

$$A^{\mathrm{In}} = \bigcap_{g \in A} g = \{m \in M \mid (\forall g \in A)(m \in g)\}. \quad (9)$$

Экстенсионал множества признаков B будет состоять из всех объектов, содержащих все признаки из B. В этом случае равенство (8) может быть переписано следующим образом:

$$B^{\mathrm{Ex}} = \{g \in G \mid B \subseteq g\} = \{g \in G \mid (\forall m \in B)(m \in g)\}. \quad (10)$$

Теперь сходство между парами равенств (1), (2) и (9), (10) стало совершенно очевидным.

Итак, непосредственной проверкой можно убедиться в том, что упорядоченная тройка $\langle \mathbf{O}, \mathbf{A}, э \rangle$, где $\mathbf{O}$ — множество объектов ДСМ-метода, $\mathbf{A}$ — множество атомов (каждый объект является подмножеством множества атомов), э — отношение «содержит» (o э a тогда и только тогда, когда $a \in o$), является формальным контекстом, для которого интенсионал совпадает со сходством семейства объектов ($O^{\mathrm{In}} = O^{\mathrm{Si}}$), а экстенсионал — с кластером фрагмента ($s^{\mathrm{Ex}} = s^{\mathrm{Cl}}$).

Подчеркнем, что при погружении ДСМ-метода в анализ формальных понятий роль признаков будут играть атомы, из которых состоят объекты.

Введем теперь еще один термин из анализа формальных понятий.

Пусть $\langle G, M, I \rangle$ — формальный контекст, $A \subseteq G$, $B \subseteq M$. Упорядоченную пару $\langle A, B \rangle$ назовем формальным понятием, если верны следующие равенства:

$$A = B^{\mathrm{Ex}}, \quad (11)$$

$$B = A^{\mathrm{In}}. \quad (12)$$

Множество A будем называть объемом (экстенсионалом) формального понятия $\langle A, B \rangle$, а множество B — его содержанием (интенсионалом).

Для рассмотренного выше формального контекста $\langle \mathbf{O}, \mathbf{A}, э \rangle$ равенства (11) и (12) будут равносильны равенствам (4) и (5), соответственно. В этом случае, пара $\langle O, s \rangle$ (где $O \subseteq \mathbf{O}$, $s \subseteq \mathbf{A}$) является формальным понятием тогда и только тогда, когда она удовлетворяет условию исчерпываемости относительно множества $\mathbf{O}$ (см. раздел 5).

Процедуры ДСМ-метода должны находить все пары, удовлетворяющие условию исчерпываемости. Для этого достаточно найти (породить) все формальные понятия в некотором формальном контексте. Как уже было сказано выше, в таком направлении исследований, как анализ формальных понятий было разработано немало алгоритмов, эффективно решающих эту задачу.

11. ПОДГОТОВКА ДАННЫХ

Данные для ДСМ-системы могут быть разными. Их внешнее представление может существенно отличаться от внутреннего, с которым работают процедуры ДСМ-метода. Например, в анализе данных по фармакологии объектами считаются химические соединения. Исходные данные в этом случае содержат представления структурных формул (или трехмерной структуры молекул) химических соединений в каком-либо стандартном формате.

Для работы ДСМ-системы необходимо представить соединения в виде множеств. Для этого используется особая система кодирования, разработанная специалистами в предметной области, — так называемый фрагментарный код суперпозиций подструктур (ФКСП) [41]. Задача автоматизации перевода внешнего представления во внутреннее для этого случая весьма нетривиальна. Решение этой задачи включает разработку оригинальных алгоритмов обработки химических графов и создание эффективной компьютерной программы, выполняющей такой перевод. Этому и была посвящена диссертация Д.А. Добрынина [42].

Существуют и другие предметные области, для которых формирование внутреннего представления данных для ДСМ-системы является нетривиальной задачей и требует участия эксперта. Однако в большинстве случаев данные для анализа записаны в прямоугольную таблицу, в которой каждому объекту соответствует единственная строка таблицы. Столбцы таблицы содержат значения параметров объекта, и существует единственный выделенный столбец, соответствующий целевому свойству. Для определенности будем считать, что в этом столбце могут

содержаться только три возможных значения: «1» (объект обладает свойством), «0» (объект не обладает свойством) и «?» или пустая клетка (информация о наличии свойства у объекта отсутствует).

В случае такого табличного представления совокупности объектов часто говорят, что каждый объект представлен в виде вектора. Это не совсем правильно, так как термин «вектор» принято употреблять по отношению к структуре, все элементы которой имеют один и тот же тип. Однако на практике в разных столбцах таблицы могут содержаться значения разных типов (целые, вещественные, строковые и т.п.), т.е. в общем случае объект представлен в виде записи (кортежа) или строки таблицы реляционной базы данных. Мы все-таки будем использовать по отношению к такой смешанной структуре термин «вектор», подразумевая, что наш вектор является гибридным или обобщенным.

Предположим, что таблица, представляющая данные, содержит n столбцов. Один столбец (обычно первый) содержит идентификаторы объектов, и еще один столбец (обычно последний) содержит значения целевого свойства. Оставшиеся $n-2$ столбца образуют *представления объектов*. У таблицы имеется *заголовочная строка*, содержащая имена (идентификаторы) атрибутов, другими словами — *заголовки столбцов*. Разумеется, у разных столбцов должны быть разные заголовки.

Наша задача — заменить векторы, представляющие объекты, на подмножества некоторого универсума, сохранив (насколько это возможно) смысл информации, содержавшейся в исходном векторе, и не допустив чрезмерного увеличения размеров универсума.

Прежде всего, нам необходимо разбить атрибуты на два класса:

- индивидуальные (не требующие группировки) — атрибуты, для которых исходная таблица содержит *мало* различных значений,
- групповые (требующие обязательной группировки) — атрибуты, для которых исходная таблица содержит *много* различных значений.

Типичным примером индивидуального атрибута является целевое свойство. Соответствующий столбец может содержать всего три возможных значения. Типичный пример группового атрибута — идентификатор объекта. Все значения в соответствующем столбце разные. Но ни целевое свойство, ни идентификатор в представление объекта не входят.

Пороговые значения, отличающие «малое количество возможных значений» от «большого количества возможных значений», зависят от конкретной задачи. В любом случае «мало» означает «намного меньше количества строк в таблице», а много — «равно или не намного меньше количества строк в таблице».

Чаще всего, групповые атрибуты соответствуют данным, представляющим результаты измерений, например: рост, масса тела, температура, возраст и т.п. Обычно такие атрибуты имеют вещественный тип. Для группировки результатов измерений наиболее естественно использовать интервалы значений. О том, на сколько интервалов следует разбить множество возможных значений атрибута и какие нужно установить пороговые значения, лучше выяснить у эксперта предметной области.

Однако может получиться и так, что групповой атрибут является *номинальным*, т.е. для него неизвестно естественное отношение линейного порядка. Рассмотрим, например, атрибут «цвет». Предположим, что в столбце «цвет» содержится 256 вариантов различных значений. Это довольно много. Кроме того, предположим, что (в контексте рассматриваемой задачи) на множестве значений параметра «цвет» нельзя определить естественное отношение порядка. Тогда мы должны выбрать какой-либо осмысленный способ классификации, не использующий интервалы. Например, поместить в один класс «оттенки серого», в другой — «оттенки красного» и т.д.

Если же никакого естественного способа классификации для группового атрибута найти не удается, то такой атрибут следует исключить из представления объекта, так как все равно никакая закономерность, в которой было бы существенным участие этого атрибута, обнаружена не будет.

Для единообразия будем предполагать, что множество возможных значений любого (группового или индивидуального) атрибута разбивается на конечное число классов. Каждый класс индивидуального атрибута содержит в точности один элемент.

Через $\mathrm{Partition}(A)$ обозначим множество (имен) классов, на которые разбито множество возможных значений атрибута A. Через R обозначим множество атрибутов, участвующих в векторном представлении объектов. Через U обозначим универсум для представления объектов в виде множеств. Положим по определению,

$$U = \{\langle A, C\rangle \mid A \in R, C \in \mathrm{Partition}(A)\}.$$

Пусть v — векторное представление объекта. Через $v(A)$ будем обозначать значение атрибута A в строке v. Через $\mathrm{Set}(v)$ будем обозначать представление объекта в виде множества, соответствующее векторному представлению v. Положим по определению,

$$\mathrm{Set}(v) = \{\langle A, C\rangle \mid A \in R, C \in \mathrm{Partition}(A), v(A) \in C\}.$$

Предположим, что наши данные записаны в табл. 2.

Таблица 2

Исходные данные (векторное представление)

Идентификатор	Глаза	Нос	Губы	Целевое свойство
Маша	Большие	Средний	Средние	1
Даша	Средние	Средний	Пухлые	1
Саша	Маленькие	Длинный	Тонкие	0

Для простоты будем считать, что все атрибуты индивидуальны. Тогда для универсума получим следующее представление.

$$U = \{\langle\text{Глаза, Большие}\rangle, \langle\text{Глаза, Средние}\rangle, \langle\text{Глаза, Маленькие}\rangle,$$
$$\langle\text{Нос, Средний}\rangle, \langle\text{Нос, Длинный}\rangle,$$
$$\langle\text{Губы, Средние}\rangle, \langle\text{Губы, Пухлые}\rangle, \langle\text{Губы, Тонкие}\rangle\}.$$

В качестве имени векторного представления будем использовать идентификатор строки. Тогда объекты будут представлены в виде множеств следующим образом:

$$\mathrm{Set}(\text{Маша}) = \{\langle\text{Глаза, Большие}\rangle, \langle\text{Нос, Средний}\rangle, \langle\text{Губы, Средние}\rangle\},$$

$$\mathrm{Set}(\text{Даша}) = \{\langle\text{Глаза, Средние}\rangle, \langle\text{Нос, Средний}\rangle, \langle\text{Губы, Пухлые}\rangle\},$$

$$\mathrm{Set}(\text{Саша}) = \{\langle\text{Глаза, Маленькие}\rangle, \langle\text{Нос, Длинный}\rangle, \langle\text{Губы, Тонкие}\rangle\}.$$

Теперь обратим внимание на то, что в ДСМ-методе причина *отсутствия* набора целевых свойств всегда описывается через *наличие* каких-либо атомов во фрагменте. Атомы могут интерпретироваться как признаки или компоненты структуры. Но вполне возможна такая ситуация, когда причиной отсутствия набора свойств является не наличие каких-либо компонентов, а их отсутствие. Например, *отсутствие* качественного профессионального образования является возможной причиной *отсутствия* престижной работы.

Более того, отсутствие каких-либо компонентов может быть причиной наличия целевых свойств. Например, *отсутствие* в организме витамина B1 является причиной *наличия* у пациента болезни бери-бери.

В связи с этими обстоятельствами, следует отнестись к подготовке данных более внимательно и попытаться представить «отсутствие» через «наличие». Для этого мы должны в два раза увеличить наш универсум, добавив «отрицательные» классы. Для нашего примера это означает, что универсум должен включать не только пары

$$\langle\text{Глаза, Большие}\rangle, \langle\text{Глаза, Средние}\rangle, \langle\text{Глаза, Маленькие}\rangle,$$

но и пары

$$\langle\text{Глаза, не Большие}\rangle, \langle\text{Глаза, не Средние}\rangle, \langle\text{Глаза, не Маленькие}\rangle.$$

Теперь опишем эту ситуацию более формально. Через $\mathrm{Dom}(A)$ будем обозначать *домен* атрибута A, т.е. множество его возможных значений. Для любого класса $C \in \mathrm{Partition}(A)$ положим

$$\overline{C} = \mathrm{Dom}(A) \setminus C.$$

Через $\mathrm{Cover}(A)$ будем обозначать *расширение разбиения* множества значений атрибута A на классы или *покрытие* домена атрибута A. Положим по определению,

$$\mathrm{Cover}(A) = \mathrm{Partition}(A) \cup \{\overline{C} \mid C \in \mathrm{Partition}(A)\}.$$

Определим *расширенный универсум* EU следующим образом:

$$EU = \{\langle A, C\rangle \mid A \in R, C \in \mathrm{Cover}(A)\}.$$

Для каждого векторного представления v *определим расширенное представление в виде множества* следующим образом:

$$\mathrm{ESet}(v) = \{\langle A, C\rangle \mid A \in R, C \in \mathrm{Cover}(A), v(A) \in C\}.$$

Теперь посмотрим, какую интерпретацию получат эти определения в контексте нашего примера. Сначала договоримся вместо $\overline{C}$ писать просто «не C». Для расширенного универсума получим следующее представление:

$$\begin{aligned} EU = \{&\langle\text{Глаза, Большие}\rangle, \langle\text{Глаза, Средние}\rangle, \langle\text{Глаза, Маленькие}\rangle,\\ &\langle\text{Нос, Средний}\rangle, \langle\text{Нос, Длинный}\rangle,\\ &\langle\text{Губы, Средние}\rangle, \langle\text{Губы, Пухлые}\rangle, \langle\text{Губы, Тонкие}\rangle,\\ &\langle\text{Глаза, не Большие}\rangle, \langle\text{Глаза, не Средние}\rangle, \langle\text{Глаза, не Маленькие}\rangle,\\ &\langle\text{Нос, не Средний}\rangle, \langle\text{Нос, не Длинный}\rangle,\\ &\langle\text{Губы, не Средние}\rangle, \langle\text{Губы, не Пухлые}\rangle, \langle\text{Губы, не Тонкие}\rangle\}. \end{aligned}$$

Теперь рассмотрим расширенные представления объектов в виде множеств:

$$\begin{aligned} \mathrm{ESet}(\text{Маша}) = \{&\langle\text{Глаза, Большие}\rangle, \langle\text{Нос, Средний}\rangle, \langle\text{Губы, Средние}\rangle,\\ &\langle\text{Глаза, не Средние}\rangle, \langle\text{Глаза, не Маленькие}\rangle, \langle\text{Нос, не Длинный}\rangle,\\ &\langle\text{Губы, не Пухлые}\rangle, \langle\text{Губы, не Тонкие}\rangle\}, \end{aligned}$$

$$\begin{aligned} \mathrm{ESet}(\text{Даша}) = \{&\langle\text{Глаза, Средние}\rangle, \langle\text{Нос, Средний}\rangle, \langle\text{Губы, Пухлые}\rangle,\\ &\langle\text{Глаза, не Большие}\rangle, \langle\text{Глаза, не Маленькие}\rangle, \langle\text{Нос, не Длинный}\rangle,\\ &\langle\text{Губы, не Средние}\rangle, \langle\text{Губы, не Тонкие}\rangle\}, \end{aligned}$$

$$\begin{aligned} \mathrm{ESet}(\text{Саша}) = \{&\langle\text{Глаза, Маленькие}\rangle, \langle\text{Нос, Длинный}\rangle, \langle\text{Губы, Тонкие}\rangle,\\ &\langle\text{Глаза, не Большие}\rangle, \langle\text{Глаза, не Средние}\rangle, \langle\text{Нос, не Средний}\rangle,\\ &\langle\text{Губы, не Средние}\rangle, \langle\text{Губы, не Пухлые}\rangle\}. \end{aligned}$$

Несмотря на кажущуюся избыточность, расширенные представления в виде множеств несут много полезной информации. Из таблицы 2 видно, что объекты «Маша» и «Даша» являются положительными примерами для целевого свойства. Если взять «узкое» представление этих объектов в виде множеств, т.е. Set (Маша) и Set (Даша), то их сходство (пересечение) содержит лишь один элемент — $\langle\text{Нос, Средний}\rangle$. Это довольно бедное содержание, чтобы его всерьез рассматривать в качестве кандидата в возможные причины наличия целевого свойства.

Однако, если мы возьмем пересечение множеств ESet (Маша) и ESet (Даша), то оно, в частности, содержит такие элементы, как

$$\langle\text{Глаза, не Маленькие}\rangle, \langle\text{Нос, не Длинный}\rangle, \langle\text{Губы, не Тонкие}\rangle,$$

что выглядит гораздо более содержательным, т.е. в этом случае мы действительно смогли обнаружить кандидата в возможные причины целевого свойства.

В заключение этого раздела подчеркнем, что представление объектов в виде множеств — это всего лишь абстракция. Чтобы использовать его в программе, необходимо выбрать одно из возможных машинных представлений множеств. Заметим, что в ДСМ-системах множества обычно представляют в виде битовых строк.

ЗАКЛЮЧЕНИЕ

В настоящей работе предложен нестандартный подход к описанию конструкций ДСМ-метода. Этот подход был использован для определения решающих предикатов и правил ДСМ-метода. Обсуждалась также связь между ДСМ-методом и анализом формальных понятий, обосновывалась возможность использования процедурами ДСМ-метода алгоритмов порождения формальных понятий.

Разумеется, эта статья не охватывает всех (и даже всех основных) проблем, связанных с ДСМ-методом. Автор планирует написать продолжение

этой статьи, в котором собирается рассмотреть следующие вопросы:

- обобщенный и несимметричный ДСМ-метод;
- алгоритмы ДСМ-метода;
- проектирование ДСМ-системы;
- нестандартные варианты ДСМ-метода.

Автор желает читателям успехов в самостоятельном изучении ДСМ-метода и применении его для решения прикладных задач.

СПИСОК ЛИТЕРАТУРЫ

1. Финн В.К. О возможностях формализации правдоподобных рассуждений средствами многозначных логик // Всесоюз. симпозиум по логике и методологии науки.— Киев: Наукова думка, 1976.— С. 82–83.
2. Финн В.К. Базы данных с неполной информацией и новый метод автоматического порождения гипотез // Диалоговые и фактографические системы информационного обеспечения.— М., 1981.— С. 153–156.
3. Финн В.К. О машинно-ориентированной формализации правдоподобных рассуждений в стиле Ф. Бэкона — Д.С. Милля // Семиотика и информатика. — 1983. — Вып. 20. — С. 35–101.
4. Автоматическое порождение гипотез в интеллектуальных системах / сост. Е.С. Панкратова, В.К. Финн; под. общ. ред. В.К. Финна. — М.: ЛИБРОКОМ, 2009. — 528 с.
5. ДСМ-метод автоматического порождения гипотез: Логические и эпистемологические основания / сост. О.М. Аншаков, Е.Ф. Фабрикантова; под. общ. ред. О.М. Аншакова. — М.: ЛИБРОКОМ, 2009. — 433 с.
6. Милль Д.С. Система логики силлогистической и индуктивной. - М.: ЛЕНАНД, 2011, 832 с.
7. Поппер К.Р. Логика и рост научного знания. - М.: Прогресс, 1983. - 605 с.
8. Peirce C.S. Abduction and induction // Philosophical Writings of Peirce / ed. J. Buchler. - NY: Dover, 1995. – P. 150–156.
9. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть I // Искусственный интеллект и принятие решений. — 2010. — № 3. — С. 3–21.
10. Финн В.К. Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Часть II // Искусственный интеллект и принятие решений. — 2010. — № 4. — С. 14–40.
11. Финн В.К. Об определении эмпирических закономерностей посредством ДСМ - метода автоматического порождения гипотез // Искусственный интеллект и принятие решений. — 2010. — № 4. — С. 41–48.
12. Финн В.К. Искусственный интеллект: Методология, применения, философия. - М.: КРАСАНД, 2011. — 448 с.
13. Волкова А.Ю. Алгоритмизация процедур ДСМ-метода автоматического порождения гипотез // НТИ. Сер 2. — 2011. — № 5.— С. 6–12.
14. Rosser J.B., Turquette A.R. Many-valued logics. - Amsterdam: North-Holland, 1951.
15. Борщев В.Б. О постулатах ДСМ-метода // Новости искусственного интеллекта. Спец. вып. К 60-летию В.К. Финна.— М.: 1993.— С. 16–26.
16. Аншаков О.М. Об одной интерпретации ДСМ-метода автоматического порождения гипотез // НТИ. Сер 2. — 1999. — № 1–2.— С. 45–53.
17. Шундеев А.С. Логико-языковые средства автоматизации производственных процессов: дис. … канд. физ.-мат. наук. - М., 2005. - 168 с.
18. Осипов Г.С. Лекции по искусственному интеллекту. - М.: УРСС, 2009. - 272 с.
19. Липкин А.А. ДСМ-метод порождения гипотез для объектов, описываемых атрибутами с весами: дис. …канд. тех. наук. - М., 2008. - 117 с.
20. Гусакова С.М., Михеенкова М.А., Финн В.К. О логических средствах автоматизированного анализа мнений // НТИ. Сер 2. — 2001. — № 5.— С. 4–24.
21. Новиков Ф.А., Иванов Д.Ю. Моделирование на UML. Теория, практика, видеокурс. — СПб.: Профессиональная литература, Наука и Техника, 2010. — 640 с.
22. Шрейдер Ю.А. Равенство, сходство, порядок. — М.: Наука, 1971. — 257 с.
23. Математическая Энциклопедия. Т. 1 (А–Г) / Ред. коллегия: И.М. Виноградов (глав. ред.) [и др.] — М., «Советская Энциклопедия», 1977. - 1152 с.
24. Кузнецов С.О. ДСМ-метод на языке соответствий Галуа // НТИ. Сер 2. — 2006. — № 12. — С. 1–7.
25. Виноградов Д.В. Логические программы для квазиаксиоматических теорий // НТИ. Сер 2. — № 1–2. — 1999. — С. 61–64
26. Виноградов Д.В. Корректные логические программы для правдоподобных рассуждений // НТИ. Сер. 2. — 2001. — № 5. — С. 25–28.
27. Михеенкова М.А., Феофанова Т.Л. Обучающая ДСМ-система для анализа социологических данных // Вестник Российского государственного гуманитарного университета. Сер. «Информатика. Информационная безопасность. Математика». – 2009. - Вып. 10. - С. 152 –169.
28. Виноградов Д.В. Формализация правдоподобных рассуждений в логике предикатов первого порядка // НТИ. Сер 2. — 2000. — № 11. — С. 17–20.
29. Аншаков О.М., Скворцов Д.П., Финн В.К. О дедуктивной имитации некоторых вариантов ДСМ-метода // Семиотика и информатика.— 1993.— Вып. 33.— С. 164–233.
30. Wille R. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts // Ordered

Sets / Ed. by I. Rival. — Dordrecht; Boston: Reidel, 1982. — P. 445–470.

31. Ganter B., Wille R. Formal Concept Analysis: Mathematical Foundations, Berlin: Springer-Verlag, 1999.
32. Ganter B., Wille R. “Conceptual scaling,” in Applications of combinatorics and graph theory to the biological and social sciences / ed. by F. Roberts. - New York: Springer-Verlag, 1989. - P. 139–167.
33. Anshakov O. M., Finn V. K., Skvortsov D. P. On axiomatization of many-valued logics associated with formalization of plausible reasonings // Studia Logica. — 1989. — Vol. 48, N 4. — P. 423–447.
34. Панкратова Е.С., Ивашко В.Г., Авидон В.В., Блинова В.Г., Бодягин Д.А. Экспериментальная проверка новой версии ДСМ-метода автоматического порождения гипотез // НТИ. Сер. 2.— 1988.— № 2.— С. 18–21.
35. Панкратова Е.С., Блинова В.Г., Финн В.К. О возможности применения ДСМ-метода в задаче распознавания химического канцерогенеза // Экпертные системы: состояние и перспективы / под ред. Д.А.Поспелова.— М.: Наука, 1989.— С. 131–138.
36. Панкратова Е.С., Ивашко В.Г., Блинова В.Г., Попов Д.В. Применение ДСМ-метода порождения гипотез для прогноза противоопухолевой активности и токсичности соединений, принадлежащих к различным классам химических соединений // Экпертные системы: состояние и перспективы / под ред. Д.А. Поспелова.— М.: Наука, 1989.— С. 139–146.
37. Забежайло М.И., Ивашко В.Г., Кузнецов С.О., Михеенкова М.А., Хазановский К.П., Аншаков О.М. Алгоритмические и программные средства ДСМ-метода автоматического порождения гипотез // НТИ. Сер. 2. — 1987. — № 10. — С. 1–14.
38. Кузнецов С.О. Быстрый алгоритм построения всех пересечений объектов из конечной полурешетки // НТИ, Сер.2. — 1993. — № 1. — С. 17–20.
39. Kuznetsov S.O., Obiedkov S.A. Comparing Performance of Algorithms for Generating Concept Lattices // Journal of Experimental and Theoretical Artificial Intelligence. — 2002. — Vol. 14. — N. 2–3. - P. 189–216.
40. Norris E.M. An Algorithm for Computing the Maximal Rectangles in a Binary Relation // Revue Roumaine de Mathématiques Pures et Appliquées. — 1978. — N 23(2). — P. 243–250.
41. Avidon V.V., Pomerantsev I.A., Golender V.E., Rozenblit A.B. Structure-activity relationship oriented languages for chemical structure representation // Journal of Chemical Information and Computer Sciences. — 1982. — Vol. 22, N 4. — P. 207–214.
42. Добрынин Д.А. Инструментальные средства для представления информации о структуре химических соединений и их сходстве в интеллектуальных системах: дис. … канд. тех. наук.- М., 2003. — 118 с.

Материал поступил в редакцию 05.07.12.

Сведения об авторе

АНШАКОВ Олег Михайлович – доктор физико-математических наук, профессор РГГУ, Москва

E-mail: oansh@yandex.ru

УДК 004.89:510.64

Д.В. Виноградов

Вероятностное порождение гипотез в ДСМ-методе с помощью простейших цепей Маркова*

*Описывается вероятностный подход к порождению гипотез в правдоподобных рассуждениях типа ДСМ с помощью специальных цепей Маркова. Этот подход подобен методам Монте-Карло, активно применяемым в задачах приближенного подсчета комбинаторных конфигураций и вычислений объемов выпуклых тел. Обсуждаются простейшие цепи Маркова: немонотонная, монотонная и «склеивающая в будущем». Доказываются их свойства, а для склеивающей доказывается останавливаемость с вероятностью 1.***

Ключевые слова: *ДСМ-гипотеза, цепь Маркова, стохастическая матрица, стационарное распределение, правдоподобные рассуждения типа ДСМ*

ВВЕДЕНИЕ

Одним из самых разработанных в отечественном искусственном интеллекте является ДСМ-метод правдоподобных рассуждений [1], объединяющий в себе индуктивное обобщение обучающих примеров в гипотезах о причинах, доопределение свойств неизвестных примеров по аналогии с обучающими, абдуктивное принятие гипотез. Хорошо известно, что основная вычислительная сложность алгоритмов ДСМ-метода сосредоточена на этапе индуктивного порождения гипотез (см. [2]). На практике довольно часто число гипотез на порядок превосходит объем обучающей выборки. Несколько раз в экспериментах наблюдался «комбинаторный взрыв» числа гипотез. Автор полагает, что результаты, полученные в результате излишне затраченных вычислительных ресурсов (несколько часов на выборках умеренного объема), не имеют и содержательного смысла. Эксперту сложно отобрать среди множества гипотез, часто отличающихся одним признаком (фрагментом кода ФКСП, например), те, которые могут им быть объявлены «причинами» исследуемых свойств (например, фармакофорами).

В последнее время в ДСМ-сообществе наметился интерес к методам уменьшения числа гипотез (см. [3]). Однако такие методы основаны на прореживании числа всех порожденных гипотез с сохранением объясняющей и предсказательной силы. В настоящей работе мы предложим другой подход, основанный на многократных запусках специальных цепей Маркова, результат выдачи которых образует случайную выборку из гипотез.

1. НЕОБХОДИМЫЕ ОПРЕДЕЛЕНИЯ И ПРИМЕРЫ

Определение 0. ***Выборкой*** обучающих примеров одного знака называется множество объектов O, каждый из которых описывается подмножеством множества признаков F. Отношение «объект $o \in O$ имеет признак $f \in F$» обозначим $(o, f) \in E \subseteq O \times F$ или, как обычно для бинарных отношений, oEf.

Пример 1. Выборку из трех примеров $O = \{o_1, o_2, o_3\}$, каждый из которых описывается некоторым подмножеством из множества признаков $F = \{f_1, f_2, f_3, f_4\}$ удобно представлять таблицей:

$O \setminus F$	f_1	f_2	f_3	f_4
o_1	+	+	-	-
o_2	+	-	+	-
o_3	-	-	-	+

Пример 2. Рассмотрим конечное множество объектов $O = \{o_1, \ldots, o_n\}$, j-ый из которых o_j описывается подмножеством $F^{-j} = \{f_1, \ldots, f_{j-1}, f_{j+1}, \ldots, f_n\}$ ($j = 1, \ldots, n$) множества признаков $F = \{f_1, \ldots, f_n\}$, непересекающегося с O. Случаю трех примеров соответствует таблица:

$O \setminus F$	f_1	f_2	f_3
o_1	-	+	+
o_2	+	-	+
o_3	+	+	-

* Работа выполнена при поддержке программы фундаментальных исследований Президиума РАН «Математическое моделирование и интеллектуальные системы» на 2012 г.

** Цепи Маркова, основанные на идеях «склеивания из прошлого» и схемах «с отверганием», будут исследованы в следующей статье.

Определение 1. ***Пересечением*** подмножества $A \subseteq O$ объектов назовем подмножество $A' = \{f \in F : \forall o \in A[oEf]\}$. Другими словами, пересечение $A' \subseteq F$ подмножества $A \subseteq O$ объектов — это все признаки, общие для всех объектов из множества *A*. Заметим, что пересечением пустого подмножества объектов по определению является все множество признаков $\varnothing' = F$.

Пример 3. Результаты операции пересечения различных подмножеств объектов из примера 1 представлены в таблице:

$A \subseteq O$	$A' \subseteq F$
$\varnothing$	$F = \{f_1, f_2, f_3\}$
$\{o_1\}$	$\{f_1, f_2\}$
$\{o_2\}$	$\{f_1, f_3\}$
$\{o_3\}$	$\{f_4\}$
$\{o_1, o_2\}$	$\{f_1\}$
$\{o_1, o_3\} = \{o_2, o_3\} = \{o_1, o_2, o_3\}$	$\varnothing$

Пример 4. Результат операции пересечения подмножества $\{o_{j_1}, \ldots, o_{j_k}\}$ объектов из примера 2 получается вычитанием из *F* подмножества признаков $\{f_{j_1}, \ldots, f_{j_k}\}$.

Определение 2. ***Пересечением*** подмножества $B \subseteq F$ признаков назовем подмножество $B' = \{o \in O : \forall f \in B[oEf]\}$. Другими словами, пересечение $B' \subseteq O$ множества $B \subseteq F$ признаков — это все объекты, обладающие всеми признаками из множества B. Пересечением пустого подмножества признаков является все множество объектов $\varnothing' = O$.

Пример 5. Результаты операции пересечения различных подмножеств признаков из примера 1 представлены в таблице:

$B \subseteq F$	$B' \subseteq O$
$\varnothing$	$O = \{o_1, o_2, o_3\}$
$\{f_1\}$	$\{o_1, o_2\}$
$\{f_2\} = \{f_1, f_2\}$	$\{o_1\}$
$\{f_3\} = \{f_1, f_3\}$	$\{o_2\}$
$\{f_4\}$	$\{o_3\}$
$\{f_1, f_4\} = \{f_2, f_3\} = \{f_2, f_4\} =$ $= \{f_3, f_4\} = \{f_1, f_2, f_3\} =$ $= \{f_1, f_2, f_4\} = \{f_1, f_3, f_4\} =$ $= \{f_2, f_3, f_4\} = \{f_1, f_2, f_3, f_4\}$	$\varnothing$

Пример 6. Результат операции пересечения подмножества $\{f_{j_1}, \ldots, f_{j_k}\}$ признаков из примера 2 получается вычитанием из *O* подмножества объектов $\{o_{j_1}, \ldots, o_{j_k}\}$.

Утверждение 1. ***Существует обучающая выборка для ДСМ-метода с*** n ***объектами одного знака, для которой возможно возникновение*** $2^n - n - 2$ ***ДСМ-сходств.***

Доказательство. Достаточно рассмотреть пример 2. Согласно примеру 4, все 2^n подмножеств будут пересечениями соответствующих подмножеств объектов. Но ДСМ-метод исключит из списка сходств пересечение пустого подмножества, n одноэлементных подмножеств объектов и пересечение всех объектов, так как оно будет иметь пустое подмножество признаков.

Определение 3. **(*Конечной*) *цепью Маркова*** на множестве состояний $S = \{0, \ldots, k-1\}$ назовем $k \times k$-матрицу $\mathbf{P} = (p_{i,j})$ из неотрицательных действительных чисел, удовлетворяющую условию, что все суммы по строкам равны 1: $\forall i \left[\sum p_{i,j} = 1\right]$. Матрицы такого вида называются ***стохастическими***. Элемент $p_{i,j}$ этой матрицы — вероятность перехода в момент времени $t+1$ в состояние $X_{t+1} = j$, стартуя в момент времени t из состояния $X_t = i$.

Пример 7. Пусть состояниями будут булевы вектора длины n. Два состояния i и j назовем ***смежными***, если они отличаются в единственном разряде. Тогда матрица $\mathbf{P} = (p_{i,j})$ с элементами, равными $1/n$, если i и j являются смежными, и равными 0 в противном случае, является цепью Маркова, соответствующей случайному блужданию на булевых векторах (= равновероятный переход из вершины гиперкуба в смежную вершину по ребру).

$i \setminus j$	000	001	010	011	100	101	110	111
000	0	1/3	1/3	0	1/3	0	0	0
001	1/3	0	0	1/3	0	1/3	0	0
010	1/3	0	0	1/3	0	0	1/3	0
011	0	1/3	1/3	0	0	0	0	1/3
100	1/3	0	0	0	0	1/3	1/3	0
101	0	1/3	0	0	1/3	0	0	1/3
110	0	0	1/3	0	1/3	0	0	1/3
111	0	0	0	1/3	0	1/3	1/3	0

Определение 4. ***Стационарным распределением*** цепи Маркова $\mathbf{P} = (p_{i,j})$ на множестве состояний $S = \{0, \ldots, k-1\}$ назовем такую вектор-строку $\pi = (\pi_i)$ длины k из неотрицательных действительных чисел,

удовлетворяющую условию $\sum \pi_i = 1$, что $\forall j \left[\sum \pi_i \cdot p_{i,j} = \pi_j \right]$. π_i - вероятность находиться в состоянии $i \in S$. Тогда в следующий момент времени вероятность находиться в любом состоянии $j \in S$ тоже будет равна π_j.

Пример 8. Для случайного блуждания по вершинам гиперкуба из примера 7 стационарным распределением будет равномерное распределение $\forall i \left[\pi_i = 2^{-n} \right]$, так как суммы по столбцам у матрицы из примера 7 тоже равны 1: $\forall j \left[\sum_i 2^{-n} \cdot p_{i,j} = 2^{-n} \cdot \sum_i p_{i,j} = 2^{-n} \right]$.

2. ЦЕПИ МАРКОВА ДЛЯ ОБРАЗОВАНИЯ СЛУЧАЙНОЙ ВЫБОРКИ ИЗ ДСМ-ГИПОТЕЗ

Определение 1. ***ДСМ-кандидатом*** из множества объектов O, каждый из которых описывается подмножеством множества признаков F, называется всякая пара $\langle A, B \rangle$, состоящая из подмножества (называемого ***списком родителей***) объектов $A \subseteq O$ и подмножества признаков $B \subseteq F$ (называемого ***пересечением объектов***), удовлетворяющая условиям $A' = B$ и $B' = A$ (последнее условие называется ***исчерпываемостью***).

Пример 1. Используя примеры 1.1, 1.3 и 1.5, перечислим все ДСМ-кандидаты в таблице:

i	A	B
0	$\varnothing$	$\{f_1, f_2, f_3, f_4\}$
1	$\{o_1\}$	$\{f_1, f_2\}$
2	$\{o_2\}$	$\{f_1, f_3\}$
3	$\{o_1, o_2\}$	$\{f_1\}$
4	$\{o_3\}$	$\{f_4\}$
5	$\{o_1, o_2, o_3\}$	$\varnothing$

Опишем теперь вероятностные алгоритмы для нахождения ДСМ-кандидатов.

Алгоритм 1. В момент времени $t = 0$ начинаем с ДСМ-кандидата $\langle F', F \rangle$. Если в произвольный момент времени t цепь Маркова находится в состоянии, соответствующем ДСМ-кандидату $\langle A, B \rangle$, где $O \setminus A = \{o_{j_1}, \ldots, o_{j_p}\}$ и $F \setminus B = \{f_{l_1}, \ldots, f_{l_q}\}$, то случайно с вероятностью $1/(p+q)$ выбирается один из элементов $\{o_{j_1}, \ldots, o_{j_p}\} \cup \{f_{l_1}, \ldots, f_{l_q}\}$. Если выбран некий объект $o \in O$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние, соответствующее ДСМ-кандидату $\langle (A \cup \{o\})'', (A \cup \{o\})' \rangle$. Если выбран некий признак $f \in F$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние, соответствующее ДСМ-кандидату $\langle (B \cup \{f\})', (B \cup \{f\})'' \rangle$. На шагах с номерами, кратными *T*, выдадим ДСМ-кандидатов, соответствующих состояниям, в которых находимся, до тех пор, пока число выданных ДСМ-кандидатов не достигнет требуемого числа. Выбор числа *T* определяется экспериментально.

Пример 2. В случае примера 1.2 алгоритм 1 соответствует случайному блужданию по вершинам гиперкуба, описанному в примере 1.7.

Пример 3. Для примера 2.1 матрица $\mathbf{P} = (p_{i,j})$ представлена ниже:

$i \setminus j$	$\varnothing$	$\{f_1\}$	$\{f_4\}$	$\{f_1, f_2\}$	$\{f_1, f_3\}$	$\{f_1, f_2, f_3, f_4\}$
$\varnothing$	0	1/4	1/4	1/4	1/4	0
$\{f_1\}$	1/4	0	0	1/4	1/4	1/4
$\{f_4\}$	2/5	0	0	0	0	3/5
$\{f_1, f_2\}$	1/4	1/4	0	0	0	2/4
$\{f_1, f_3\}$	1/4	1/4	0	0	0	2/4
$\{f_1, f_2, f_3, f_4\}$	0	0	1/3	1/3	1/3	0

Определение 2. ***Порядок*** на ДСМ-кандидатах $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$ определяется как (A_1, B_1) или, что эквивалентно, как $A_2 \subseteq A_1$.

Определение 3. Подмножество $J \subseteq S$ элементов частично упорядоченного множества (***ч.у.м.***) $\langle S, \le \rangle$, удовлетворяющее условию $\forall j \in J \forall i \left[i \le j \Rightarrow i \in J \right]$, называется (***порядковым***) ***идеалом***.

Определение 4. Цепь Маркова X_n с ч.у.м. состояний $\langle S, \le \rangle$ назовем ***монотонной***, если для любых $i \le l$ и для любого порядкового идеала J выполняется $\mathbf{P}\left[X_1 \in J \mid X_0 = i \right] = \sum p_{i,j} \ge \mathbf{P}\left[X_1 \in J \mid X_0 = l \right] = \sum p_{l,j}$.

Пример 4. Случайные блуждания по вершинам гиперкуба, описанные в примере 1.7, представляют собой монотонную цепь Маркова.

Пример 5. Цепь Маркова из примера 3 не является монотонной. Рассмотрим порядковый идеал $J = \left\{ \langle \{o_1, o_2, o_3\}, \varnothing \rangle < \langle \{o_3\}, \{f_4\} \rangle \right\}$. Рассмотрим также возрастающую пару состояний $i = \langle \{o_1\}, \{f_1, f_2\} \rangle < l = \langle \varnothing, \{f_1, f_2, f_3, f_4\} \rangle$. Тогда $\mathbf{P}\left[X_1 \in J \mid X_0 = i \right] = 1/4 > 1/3 = \mathbf{P}\left[X_1 \in J \mid X_0 = l \right]$.

Алгоритм 2. В момент времени $t = 0$ начинаем с состояния X_0, соответствующего паре ДСМ-кандидату $\langle F', F \rangle$. В каждый момент времени *t* случайно равновероятно выбирается один из элементов $O \cup F$. Пусть в этот момент времени t цепь Маркова находилась в состоянии X_t, соответствующем ДСМ-кандидату $\langle A, B \rangle$. Если был выбран некий объект $o \in O$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние X_{t+1}, соответствующее ДСМ-кандидату $\langle (A \cup \{o\})'', (A \cup \{o\})' \rangle$. Если

был выбран некий признак $f \in F$, то в следующий момент времени $t+1$ цепь Маркова переходит в состояние X_{t+1}, соответствующее ДСМ-кандидату $\langle (B \cup \{f\})', (B \cup \{f\})'' \rangle$. На шагах с номерами, кратными *T*, выдадим ДСМ-кандидатов, соответствующих состояниям, в которых находимся, до тех пор, пока число выданных ДСМ-кандидатов не достигнет требуемого числа. Выбор числа *T* определяется экспериментально.

Пример 6. Для примера 2.1 матрица $\mathbf{P} = (p_{i,j})$ монотонной цепи Маркова равна:

$i \setminus j$	$\varnothing$	$\{f_1\}$	$\{f_4\}$	$\{f_1, f_2\}$	$\{f_1, f_3\}$	$\{f_1, f_2, f_3, f_4\}$
$\varnothing$	3/7	1/7	1/7	1/7	1/7	0
$\{f_1\}$	1/7	3/7	0	1/7	1/7	1/7
$\{f_4\}$	2/7	0	2/7	0	0	3/7
$\{f_1, f_2\}$	1/7	1/7	0	3/7	0	2/7
$\{f_1, f_3\}$	1/7	1/7	0	0	3/7	2/7
$\{f_1, f_2, f_3, f_4\}$	0	0	1/7	1/7	1/7	4/7

Алгоритм 3. Состоянием склеивающей цепи Маркова будет пара ДСМ-кандидатов $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$. В момент времени $t = 0$ начинаем с состояния X_0, соответствующего паре ДСМ-кандидатов $\langle O, O' \rangle \le \langle F', F \rangle$. В каждый момент времени *t* случайно равновероятно выбирается один из элементов $O \cup F$. Пусть в этот момент времени *t* цепь Маркова находилась в состоянии X_t, соответствующем паре ДСМ-кандидатов $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$. 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 ДСМ-кандидатов $\langle (A_1 \cup \{o\})'', (A_1 \cup \{o\})' \rangle \le \langle (A_2 \cup \{o\})'', (A_2 \cup \{o\})' \rangle$.

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 ДСМ-кандидатов $\langle (B_1 \cup \{f\})', (B_1 \cup \{f\})'' \rangle \le \langle (B_2 \cup \{f\})', (B_2 \cup \{f\})'' \rangle$.

Когда будет достигнуто состояние, соответствующее паре совпадающих ДСМ-кандидатов $\langle A, B \rangle \le \langle A, B \rangle$, то выдается ДСМ-кандидат $\langle A, B \rangle$, и процесс перезапускается с начального состояния X_0: $\langle O, O' \rangle \le \langle F', F \rangle$. Алгоритм останавливается после выдачи требуемого числа ДСМ-кандидатов.

Пример 7. Для примера 2.1 матрица $\mathbf{P} = (p_{i,j})$ склеивающей цепи Маркова равна: (см. таблицу)

3. СВОЙСТВА АЛГОРИТМОВ

Лемма 1. ***Для любых двух подмножеств*** $A_2 \subseteq A_1 \subseteq O$ ***объектов имеем*** $A_1' \subseteq A_2' \subseteq F$. ***Для любых двух подмножеств*** $B_1 \subseteq B_2 \subseteq F$ ***признаков имеем*** $B_2' \subseteq B_1' \subseteq O$.

Доказательство использует определения 1.1 и 1.2.

Лемма 2. ***Для любого подмножества*** $B \subseteq F$ ***признаков имеем*** $B \subseteq B''$ ***и*** $B''' = B'$. ***Для любых двух подмножеств*** $B_1 \subseteq B_2 \subseteq F$ ***признаков имеем*** $B_1'' \subseteq B_2'' \subseteq F$. ***Для любого подмножества*** $A \subseteq O$ ***объектов имеем*** $A \subseteq A''$ ***и*** $A''' = A'$. ***Для любых двух подмножеств*** $A_2 \subseteq A_1 \subseteq O$ ***объектов имеем*** $A_2'' \subseteq A_1'' \subseteq O$.

Доказательство использует определения 1.1 и 1.2 и лемму 1.

Лемма 3. $\langle F', F \rangle$ ***и*** $\langle O, O' \rangle$ ***являются ДСМ-кандидатами. Для любого подмножества*** $B \subseteq F$ ***признаков*** $\langle B', B'' \rangle$ ***является ДСМ-кандидатом. Для любого подмножества*** $A \subseteq O$ ***объектов*** $\langle A'', A' \rangle$ ***является ДСМ-кандидатом. Для любого ДСМ-кандидата*** $\langle F', F \rangle$ ***выполнено*** $A = A''$ ***и*** $B = B''$.

Доказательство использует определение 2.1 и лемму 2.

Лемма 4. ***Для любого ДСМ-кандидата*** $\langle F', F \rangle$ ***выполнено*** $\langle A, B \rangle \le \langle F', F \rangle$ ***и*** $\langle O, O' \rangle \le \langle A, B \rangle$. ***В частности,*** $\langle O, O' \rangle \le \langle F', F \rangle$. ***Для любого ДСМ-кандидата*** $\langle F', F \rangle$ ***и любого признака*** $f \in F$ ***выполнено*** $\langle A, B \rangle \le \langle (B \cup \{f\})', (B \cup \{f\})'' \rangle$. ***Для любого ДСМ-кандидата*** $\langle F', F \rangle$ ***и любого объекта*** $o \in O$ ***выполнено*** $\langle (A \cup \{o\})'', (A \cup \{o\})' \rangle \le \langle A, B \rangle$.

Доказательство комбинирует определение 2.2 и леммы 1 и 2.

Лемма 5. ***Для любой упорядоченной пары ДСМ-кандидатов*** $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$ ***и любого признака*** $f \in F$ ***выполнено*** $\langle (B_1 \cup \{f\})', (B_1 \cup \{f\})'' \rangle \le \langle (B_2 \cup \{f\})', (B_2 \cup \{f\})'' \rangle$. ***Для любой упорядоченной пары ДСМ-кандидатов*** $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$ ***и любого объекта*** $o \in O$ ***выполнено*** $\langle (A_1 \cup \{o\})'', (A_1 \cup \{o\})' \rangle \le \langle (A_2 \cup \{o\})'', (A_2 \cup \{o\})' \rangle$.

Доказательство комбинирует определение 2.2 и леммы 1 и 2.

Теорема 1. ***Цепь Маркова из алгоритма*** 2 ***является монотонной.***

Доказательство использует лемму 5.

Лемма 6. ***Для любой упорядоченной пары ДСМ-кандидатов*** $\langle A_1, B_1 \rangle \le \langle A_2, B_2 \rangle$ ***склеивающая цепь Маркова не изменяет своего состояния с вероятностью*** $\dfrac{|A_1 \cap A_2| + |B_1 \cap B_2|}{|O| + |F|}$.

Доказательство опирается на описание алгоритма 3. Если выбрать объект из $A_1 \cap A_2$, или признак из

$B_1 \cap B_2$, то монотонная цепь Маркова не изменит своего состояния.

Определение 1. ***Длина склеивания*** ДСМ-кандидатов для выборки O, описываемой множеством признаков F, определяется как $L = \min\left(|O|,|F|\right)$, т. е. минимум из числа объектов и числа признаков. ***Вероятность выбора*** (фиксированного объекта и признака) определяется как $p = \dfrac{1}{|O|+|F|}$.

Таблица

Матрица $\mathbf{P} = \left(p_{i,j}\right)$ склеивающей цепи Маркова равна:

$i \setminus j$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$1: \varnothing \le \varnothing$	$\frac{3}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0	0	0	0	0	0	0	0	0	0
$2: \{f_1\} \le \{f_1\}$	$\frac{1}{7}$	$\frac{3}{7}$	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0	0	0	0	0	0	0	0	0
$3: \{f_4\} \le \{f_4\}$	$\frac{2}{7}$	0	$\frac{2}{7}$	0	0	$\frac{3}{7}$	0	0	0	0	0	0	0	0	0	0	0
$4: \{f_1,f_2\} \le \{f_1,f_2\}$	$\frac{1}{7}$	$\frac{1}{7}$	0	$\frac{3}{7}$	0	$\frac{2}{7}$	0	0	0	0	0	0	0	0	0	0	0
$5: \{f_1,f_3\} \le \{f_1,f_3\}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	$\frac{3}{7}$	$\frac{2}{7}$	0	0	0	0	0	0	0	0	0	0	0
$6: \{f_1,f_2,f_3,f_4\} \le \{f_1,f_2,f_3,f_4\}$	0	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{4}{7}$	0	0	0	0	0	0	0	0	0	0	0
$7: \varnothing \le \{f_4\}$	$\frac{2}{7}$	0	$\frac{1}{7}$	0	0	0	$\frac{1}{7}$	0	0	0	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0
$8: \varnothing \le \{f_1\}$	$\frac{1}{7}$	$\frac{1}{7}$	0	$\frac{1}{7}$	$\frac{1}{7}$	0	0	$\frac{2}{7}$	0	0	0	0	$\frac{1}{7}$	0	0	0	0
$9: \{f_1\} \le \{f_1,f_2\}$	$\frac{1}{7}$	$\frac{1}{7}$	0	$\frac{1}{7}$	0	$\frac{1}{7}$	0	0	$\frac{2}{7}$	0	0	0	0	0	$\frac{1}{7}$	0	0
$10: \{f_1\} \le \{f_1,f_3\}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0	$\frac{2}{7}$	0	0	0	$\frac{1}{7}$	0	0	0
$11: \varnothing \le \{f_1,f_2\}$	$\frac{1}{7}$	0	0	$\frac{1}{7}$	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	0	$\frac{1}{7}$	0	$\frac{1}{7}$	0	$\frac{1}{7}$	0	0
$12: \varnothing \le \{f_1,f_3\}$	$\frac{1}{7}$	0	0	0	$\frac{1}{7}$	0	0	$\frac{1}{7}$	0	$\frac{1}{7}$	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0
$13: \{f_4\} \le \{f_1,f_2,f_3,f_4\}$	0	0	$\frac{1}{7}$	0	0	$\frac{3}{7}$	0	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0	0
$14: \{f_1,f_2\} \le \{f_1,f_2,f_3,f_4\}$	0	0	0	$\frac{1}{7}$	0	$\frac{2}{7}$	$\frac{1}{7}$	0	0	$\frac{1}{7}$	0	0	0	$\frac{2}{7}$	0	0	0
$15: \{f_1,f_3\} \le \{f_1,f_2,f_3,f_4\}$	0	0	0	0	$\frac{1}{7}$	$\frac{2}{7}$	$\frac{1}{7}$	0	$\frac{1}{7}$	0	0	0	0	0	$\frac{2}{7}$	0	0
$16: \{f_1\} \le \{f_1,f_2,f_3,f_4\}$	0	0	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	0	$\frac{1}{7}$	$\frac{1}{7}$	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0
$17: \varnothing \le \{f_1,f_2,f_3,f_4\}$	0	0	0	0	0	0	$\frac{1}{7}$	0	0	0	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{1}{7}$	0

Лемма 7. ***Если*** $|O| \le |F|$, ***то для любого натурального числа*** r ***и любой упорядоченной пары ДСМ-кандидатов*** $\langle A_1, B_1\rangle \le \langle A_2, B_2\rangle$ ***выполнено*** $\mathbf{P}\left[S_{r+L} = \langle A_1, B_1\rangle \le \langle A_1, B_1\rangle \,|\, S_r = \langle A_1, B_1\rangle \le \langle A_2, B_2\rangle\right] \ge p^L$. ***Если*** $|O| \ge |F|$, ***то для любого натурального числа*** r ***и любой упорядоченной пары ДСМ-кандидатов*** $\langle A_1, B_1\rangle \le \langle A_2, B_2\rangle$ ***выполнено*** $\mathbf{P}\left[S_{r+L} = \langle A_2, B_2\rangle \le \langle A_2, B_2\rangle \,|\, S_r = \right.$ $\left. = \langle A_1, B_1\rangle \le \langle A_2, B_2\rangle\right] \ge p^L$.

Доказательство использует определение 1 и описание алгоритма 3. Для склейки вниз необходимо постоянно выбирать объекты из $A_1 \setminus A_2$. Для склейки вверх необходимо постоянно выбирать признаки из $B_2 \setminus B_1$. Легко проверить, что если $|O| \le |F|$, то $|A_1 \setminus A_2| \le |O| = L$, а если $|O| \ge |F|$, то $|B_2 \setminus B_1| \le |F| = L$.

Теорема 2. ***Для склеивающей цепи Маркова из алгоритма*** 3 ***вероятность остановиться после*** n ***шагов стремится к*** 1 ***при*** $n \to \infty$.

Доказательство. Имеем оценку

$$\mathbf{P}\left[S_{k\cdot L} = \langle A_{1,k}, B_{1,k}\rangle < \langle A_{2,k}, B_{2,k}\rangle \,|\, S_0 = \langle O, O'\rangle < \langle F', F\rangle\right] \le$$

$$\le \prod \mathbf{P}\begin{bmatrix} S_{j\cdot L} = \langle A_{1,k}, B_{1,k}\rangle < \langle A_{2,k}, B_{2,k}\rangle \,|\, S_{(j-1)\cdot L} = \\ = \langle A_{1,j-1}, B_{1,j-1}\rangle < \langle A_{2,j-1}, B_{2,j-1}\rangle \end{bmatrix} \le (1-p)^{k\cdot L},$$

где $A_{1,0} = O$, $B_{1,0} = O'$, $A_{2,0} = F'$, $B_{2,0} = F$. Но $(1-p)^{k\cdot L} \to 0$ при $k \to \infty$.

Определение 2. Состояние, соответствующее упорядоченной паре ДСМ-кандидатов $\langle A_1, B_1\rangle < \langle A_2, B_2\rangle$, называется ***невозвратным.*** Состояние, соответствующее паре равных ДСМ-кандидатов $\langle A_1, B_1\rangle = \langle A_2, B_2\rangle$, называется ***эргодическим***. Квадратную матрицу переходов между эргодическими состояниями обозначим через ***S***. Квадратную матрицу переходов между невозвратными состояниями обозначим через ***Q***. Прямоугольную матрицу переходов из невозвратных переходов в эргодические обозначим через ***R***. Тогда матрица переходов монотонной цепи Маркова будет иметь следующий вид $P = \begin{pmatrix} S & \mathbf{0} \\ R & Q \end{pmatrix}$. Рассмотрим также ***остановленную*** матрицу $\hat{P} = \begin{pmatrix} I & \mathbf{0} \\ R & Q \end{pmatrix}$, где ***I*** – единичная матрица.

Пример 1. Матрица ***S*** для примера 2.7 совпадает с матрицей ***P*** для примера 2.6.

Лемма 8. ***Для любого натурального*** $k > 1$ ***для степеней матриц будут выполнены представления***

$$P^k = \begin{pmatrix} S^k & \mathbf{0} \\ \sum Q^t \cdot R \cdot S^{k-1-t} & Q^k \end{pmatrix} \text{ и } \hat{P}^k = \begin{pmatrix} I & \mathbf{0} \\ \sum Q^t \cdot R & Q^k \end{pmatrix}.$$

Доказательство комбинирует определение 2 и возможность блочного умножения матриц.

Следствие 1. ***При натуральном*** $n \to \infty$ ***имеем*** $Q^n \to \mathbf{0}$, ***т. е. степени матрицы Q стремятся к нулевой матрице.***

Доказательство прямо выводится из теоремы 2.

Следствие 2. ***При натуральном*** $k \to \infty$ ***имеем*** $\hat{P}^n \to \hat{P}^\infty = \begin{pmatrix} I & \mathbf{0} \\ \sum Q^t \cdot R & \mathbf{0} \end{pmatrix}$.

Доказательство комбинирует лемму 8 и следствие 1.

Следствие 3. ***Матрица вероятностей попасть в эргодическое состояния, стартуя из невозвратного состояния, равна*** $B = \sum Q^t \cdot R$.

Доказательство опирается на следствие 2.

Лемма 9. ***Любая степень стохастической матрицы A является стохастической матрицей.***

Доказательство использует определение 1.3 и ассоциативность умножения матриц, так как $n \times n$-матрица ***A*** является стохастической, если и только если $A \cdot \zeta_n = \zeta_n$, где $\zeta_n = (1,\ldots,1)^T$ - столбец из n штук единиц.

Следствие 4. ***Матрица*** $B = \sum Q^t \cdot R$ ***имеет суммы элементов по строкам равные*** 1, т. е. $A \cdot \zeta_n = \zeta_n$, ***где*** $\zeta_n = (1,\ldots,1)^T$ ***- столбец из*** n ***штук единиц.***

Доказательство выводится из следствия 2 и леммы 9.

Утверждение 1. (Перрон-Фробениус) ***Если некоторая степень*** S^k ***стохастической матрицы S имеет положительные элементы, то*** $\lim S^n = \zeta \cdot \pi$, ***где*** $\zeta = (1,\ldots,1)^T$ ***- столбец из одних единиц, а*** π ***- единственное стандартное распределение для S.***

Доказательство приводится во всех учебниках по теории цепей Маркова (например, [4]).

Следствие 5. ***При натуральном*** $k \to \infty$ ***имеем*** $P^n \to \begin{pmatrix} \zeta \cdot \pi & \mathbf{0} \\ \sum Q^t \cdot R \cdot \zeta \cdot \pi & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \zeta \cdot \pi & \mathbf{0} \\ \zeta \cdot \pi & \mathbf{0} \end{pmatrix}$.

Доказательство комбинирует следствие 4, леммы 6 и 8 и утверждение 1. Нужно лишь заметить, что для матрицы ***S*** имеется степень с положительными элементами, так как по лемме 6 модифицированная цепь, соответствующая матрице ***S***, апериодическая и неприводимая, как как любые состояния связаны через ДСМ-кандидата $\langle F', F\rangle$. Последнее равенство получается из следствия 4.

Пример 2. Распределение состояний, выданных склеивающей цепью Маркова из алгоритма 3 может не совпадать со стационарным распределением для монотонной цепи Маркова из алгоритма 2. Стационарное распределение алгоритма 2 находится как нормированный (сумма компонент которого равна 1) собственный вектор для транспонированной матрицы из примера 2.6. Примерно он равен (0.152344, 0.113281, 0.097656, 0.150391, 0.150391, 0.335937). Вектор вероятностей результатов алгоритма 3 является по следствию 3 последней строкой матрицы

$B = \sum Q^t \cdot R$, которая с помощью следствия 1 может быть преобразована в $B = (I - Q)^{-1} \cdot R$. Он приближенно равен (0.212891, 0.058501, 0.083147, 0.119978, 0.119978, 0.404406).

Определение 3. ***Расстояние (тотального) изменения*** между распределениями $\mu = (\mu_i)_{i \in S}$ и $\nu = (\nu_i)_{i \in S}$ на множестве состояний S ($\forall i \in S[\mu_i \geq 0], \forall i \in S[\nu_i \geq 0], \sum_{i \in S} \mu_i = \sum_{i \in S} \nu_i = 1$) задается как $\|\mu - \nu\|_{TV} = ½ \cdot \sum_{i \in S} |\mu_i - \nu_i|$.

Лемма 10. ***Для любых двух распределений*** $\mu = (\mu_i)_{i \in S}$ ***и*** $\nu = (\nu_i)_{i \in S}$ ***имеем*** $\|\mu - \nu\|_{TV} = \max_{A \subseteq S} |\mu(A) - \nu(A)|$, ***где*** $\mu(A) = \sum_{i \in A} \mu_i$ ***и*** $\nu(A) = \sum_{i \in A} \nu_i$.

Доказательство простое и опирается на определение 3.

Определение 4. ***Распределение выдачи*** алгоритма 3 на $S_{erg} \times \mathbb{N}$ задается равенством $\mathbf{P}(X_T = i, T \leq t) = (\mu(0) \cdot \hat{P}^t)_i$, где $i \in S_{erg}$ – эргодическое состояние, а $\mu(0) = (\delta_{i,l})_{i \in S}$ ($l = \langle O, O' \rangle \leq \langle F', F \rangle$) – начальное δ-распределение, сосредоточенное на стартовой паре $\langle O, O' \rangle \leq \langle F', F \rangle$ ДСМ-кандидатов. ***Распределение состояний*** алгоритма 3 определим как $\nu_i = \lim_{T \to \infty} \mathbf{P}(X_T = i, T \leq t) = \mathbf{P}(X_T = i)$, где $i \in S_{erg}$ – эргодическое состояние. ***Распределение времени склеивания*** алгоритма 3 равно $p_t = \sum_{i \in S_{erg}} \mathbf{P}(X_T = i, T = t) = \mathbf{P}(T = t)$.

Лемма 11. ***Распределение состояний вычисляется так:*** $\nu_i = (\bar{\mu}(0) \cdot \boldsymbol{B})_i$, ***где*** $\bar{\mu}(0)$ ***- начальное распределение, ограниченное на невозвратные состояния, а*** $i \in S_{erg}$ ***– эргодическое состояние.***

Доказательство опирается на следствие 3.

Лемма 12. ***Распределение времени склеивания равно*** $p_t = \bar{\mu}(0) \cdot \boldsymbol{Q}^{t-1} \cdot \boldsymbol{R} \cdot \boldsymbol{\zeta}$, ***где*** $\bar{\mu}(0)$ ***- начальное распределение, ограниченное на невозвратные состояния.***

Доказательство использует лемму 8, аналогично доказательству леммы 11.

Определение 5. Если $\hat{T}$ – независимая целочисленная случайная величина (ц.с.в.), имеющая распределение $\mathbf{P}[\hat{T} = t] = \hat{p}_t$, то ***условное распределение состояний*** относительно события $B = \{T \leq \hat{T}\}$ (при условии, что $\mathbf{P}(B) > 0$) определяется равенством: $\mu(\hat{T}) = c \cdot \sum_{t=0}^{\infty} \hat{p}_t \cdot \mu(0) \cdot \hat{P}^t$, где c — такая константа нормирования, чтобы $\mu(\hat{T}) \cdot \zeta = \sum \mu(\hat{T})_i = 1$. Так распределены состояния алгоритма 3, если прекратить вычисления, когда склеивание не наступит до момента времени $T \leq \hat{T}$.

Лемма 13. ***Условное распределение состояний вычисляется так:*** $\mu(\hat{T})_i = \dfrac{\mathbf{P}(\{X_T = i, T \leq \hat{T}\})}{\mathbf{P}(\{T \leq \hat{T}\})}$, ***где*** $i \in S_{erg}$ ***– эргодическое состояние.***

Доказательство опирается на независимость $\hat{T}$ и (X_T, T) и условие нормирования из определения 5.

Лемма 14. ***Если*** $\hat{T}$ ***– независимая ц.с.в., и событие*** $B = \{T \leq \hat{T}\}$ ***имеет ненулевую вероятность*** $\mathbf{P}(B) > 0$, ***то*** $\|\mu(\hat{T}) - \nu\|_{TV} \leq \dfrac{1}{\mathbf{P}(B)} - 1$.

Доказательство использует леммы 10 и 13. $\|\mu(\hat{T}) - \nu\|_{TV} = \max_{A \subseteq S} |\mu(\hat{T})(A) - \nu(A)|$. Фиксируем произвольное $A \subseteq S_{erg}$ и умножаем $|\mu(\hat{T})(A) - \nu(A)|$ на $\mathbf{P}(B) > 0$:

$$\mathbf{P}(T \leq \hat{T}) \cdot |\mu(\hat{T})(A) - \nu(A)| = \mathbf{P}(T \leq \hat{T}) \cdot \left| \sum \mu(\hat{T})_i - \sum \nu_i \right| =$$

$$= \mathbf{P}(T \leq \hat{T}) \cdot \left| \frac{\mathbf{P}(X_T \in A, T \leq \hat{T})}{\mathbf{P}(T \leq \hat{T})} - \mathbf{P}(X_T \in A) \right| =$$

$$= |\mathbf{P}(X_T \in A, T \leq \hat{T}) - \mathbf{P}(T \leq \hat{T}) \cdot \mathbf{P}(X_T \in A)| =$$

$$= |\mathbf{P}(T > \hat{T}) \cdot \mathbf{P}(X_T \in A) - \mathbf{P}(X_T \in A, T > \hat{T})| \leq$$

$$\leq \mathbf{P}(T > \hat{T}) = 1 - \mathbf{P}(T \leq \hat{T}).$$

Определение 5. Если $T_1, \ldots, T_r$ – независимые ц.с.в., имеющие распределение $\mathbf{P}(T_s = t) = \mu$, $s \in \{1, \ldots, r\}$, совпадающее с распределением времени склеивания T, то ***верхняя оценка склеивания*** по r испытаниям — ц.с.в. $\hat{T} = T_1 + \ldots + T_r$. На практике точное распределение времени склеивания неизвестно, так как неизвестна матрица переходов, поэтому можно, следуя предложению Проппа и Уилсона [5], провести r испытаний на время склеивания и в качестве верхней оценки времени склеивания взять их сумму.

Лемма 15. ***Если*** T ***и*** $\overline{T}$ ***– одинаково распределенные и независимые ц.с.в., то*** $\mathbf{P}(T > \overline{T}) \leq ½$.

Доказательство использует симметрию: $1 = \mathbf{P}(T > \overline{T}) + \mathbf{P}(\overline{T} > T) + \mathbf{P}(\overline{T} = T)$ и $\mathbf{P}(T > \overline{T}) = \mathbf{P}(\overline{T} > T)$.

Лемма 16. ***Если*** $\mathbf{P}(T = t) = \bar{\mu}(0) \cdot \boldsymbol{Q}^{t-1} \cdot \boldsymbol{R} \cdot \boldsymbol{\zeta}$ ***– распределение времени склеивания, то*** $\mathbf{P}(T > t_1 + t_2) \leq \mathbf{P}(T > t_1) \cdot \mathbf{P}(T > t_2)$.

Доказательство По леммам 8 и 9 $\mathbf{P}(T > t) = \bar{\mu}(0) \cdot \boldsymbol{Q}^t \cdot \boldsymbol{\zeta}$. По монотонности $\bar{\mu} \cdot \boldsymbol{Q}^t \cdot \boldsymbol{\zeta} \leq \bar{\mu}(0) \cdot \boldsymbol{Q}^t \cdot \boldsymbol{\zeta}$ для любого начального распределения $\bar{\mu}$ на невозвратных состояниях.

$$\mathbf{P}\left(T > t_1 + t_2\right) = \overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1+t_2} \cdot \boldsymbol{\zeta} = \overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{Q}^{t_2} \cdot \boldsymbol{\zeta} =$$

$$= \left(\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta}\right) \cdot \frac{\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1}}{\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta}} \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta} =$$

$$= \left(\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta}\right) \cdot \left(\overline{\mu} \cdot \boldsymbol{Q}^{t_2} \cdot \boldsymbol{\zeta}\right) \le$$

$$\le \left(\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta}\right) \cdot \left(\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_2} \cdot \boldsymbol{\zeta}\right) = \mathbf{P}\left(T > t_1\right) \cdot \mathbf{P}\left(T > t_2\right),$$

где мы положили $\overline{\mu} \cdot \frac{\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1}}{\overline{\mu}(0) \cdot \boldsymbol{Q}^{t_1} \cdot \boldsymbol{\zeta}}$ - вектор распределения невозвратных состояний после t_1 шагов.

Лемма 17. ***Если*** $\hat{T}$ ***– верхняя оценка склеивания по*** r ***испытаниям, то событие*** $B = \left\{T \le \hat{T}\right\}$ ***имеет вероятность*** $\mathbf{P}(B) \ge 1 - (½)^r$.

Доказательство использует леммы 15 и 16.

Теорема 3. ***Если*** $\hat{T}$ ***– верхняя оценка склеивания по*** r ***испытаниям, то*** $\left\|\mu\left(\hat{T}\right) - \nu\right\|_{TV} \le \frac{1}{2^r - 1}$.

Доказательство комбинирует леммы 14 и 17.

ЗАКЛЮЧЕНИЕ

Легко проверить, что стационарное распределение (0.161994, 0.120457, 0.129802, 0.159917, 0.159917, 0.267913) цепи Маркова алгоритма 1 не является равномерным, что было ясно из вида матрицы, которая не является симметричной (относительно транспонирования).

Этот пример подтверждает правдоподобную гипотезу о том, что ***стационарное распределение будет сконцентрировано в средних слоях диаграммы порядка*** на ДСМ-кандидатах по включению первых компонент. Это утверждение обосновывается следующим эвристическим рассуждением: если цепь Маркова находится внизу этой диаграммы, то в списке родителей уже имеются почти все родители, а список признаков мал. Поэтому с большей вероятностью алгоритм 1 выберет свободный признак, а не свободный пример, что будет соответствовать переходу вверх. Если же цепь Маркова находится вверху диаграммы порядка, то малым является список родителей, а признаков много, поэтому алгоритм 1 с большей вероятностью выберет свободный пример, а не свободный признак, что соответствует переходу вниз.

Алгоритм 1 легко может быть настроен на учет запрета контр-примеров: при попытке перехода на состояние, соответствующее ДСМ-кандидату, имеющему контр-пример, цепь Маркова просто остается в исходном состоянии и переходит к следующему выбору свободных примеров или свободных признаков. Как модифицировать алгоритмы 2 и 3 в этом случае представляется интересным исследовать. С другой стороны, против алгоритма 1 говорит его немонотонный характер, установленный в примере 2.5.

В работах по применению ДСМ-метода к медицинским задачам (см. [6]) признаки стали иметь разный статус: ***необходимые***, ***существенные*** и ***сопутствующие***. Представляется интересным исследовать вопрос о расширении предложенных алгоритмов на эту ситуацию.

Наконец, автор запланировал изложить в следующей статье модификации монотонного алгоритма 2, основанные на «***склеивании из прошлого***» (см. [5]) и «***схеме с отверганием***» (см. [7]). Как следует из примера 3.2, «склеивание в будущем» алгоритмом 3 приводит к ***смещению***, т. е. не совпадение со стационарным распределением монотонной цепи Маркова из алгоритма 2. Эти более сложные варианты позволяют устранить смещение. Вопрос, о необходимости устранения смещения, тоже нуждается в изучении.

* * *

Автор благодарит В.К. Финна за поддержку и Т.А. Волкову за полезные обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. 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.
2. Кузнецов С.О. О сложности обучения и классификации, основанных на операции сходства // Труды 3-ей Национальной Конференции по Искусственному Интеллекту (КИИ-92) . – Тверь. – 1992. – Том 1. – С. 32-34.
3. Шашкин Л.О. Сравнение приближенных способов поиска сходств для ДСМ-метода // НТИ. Сер. 2. – 2010. - № 12. - С. 10-13.
4. Кемени Дж., Снелл Дж. Конечные цепи Маркова. - М.: Наука, 1970.
5. Propp J.G., Wilson D.B. Exact sampling with coupled Markov chains and applications to statistical mechanics // Random Structures and Algorithms. – 1996. - Vol. 9. - P. 223-252.
6. Панкратова Е.С., Виноградов Д.В., Формальное описание настройки интеллектуальных ДСМ-систем на область клинической и лабораторной диагностики // НТИ. Сер.2. – 2011. - № 9. - С. 1-5.
7. Fill J.A. An interruptible algorithm for perfect sampling via Markov chain // Annals of Applied Probability. – 1998. - Vol. 8. – P. 131-162.

Материал поступил в редакцию 15.06.12.

Сведения об авторе

ВИНОГРАДОВ Дмитрий Вячеславович – кандидат физико-математических наук, старший научный сотрудник ВИНИТИ РАН, Москва

E-mail: Vin@viniti.ru

УДК 004.8:510.6

В.А. Ковтун

Логико-комбинаторная система интеллектуального анализа данных, использующая аппарат нечеткого вывода

Описаны принципы работы системы анализа данных, комбинирующей элементы технологий ДСМ-метода и систем нечеткого вывода, а также приведены результаты компьютерных экспериментов, показывающие, что система, построенная на такой комбинации технологий, имеет определенные преимущества перед традиционной ДСМ-системой.

Ключевые слова: *интеллектуальный анализ данных, система нечеткого вывода, нечеткая логика, ДСМ-метод*

ВВЕДЕНИЕ

В настоящей работе описан метод анализа данных, представленных в виде вещественных векторов, включающий действия, аналогичные некоторым этапам работы системы нечеткого вывода, а также проведено сравнение этого метода с известным методом интеллектуального анализа данных, получившим название ДСМ-метода автоматического порождения гипотез. Описываемый метод комбинирует подходы ДСМ-метода и систем, основанных на нечеткой логике. Из ДСМ-метода взяты процедуры порождения гипотез о возможных причинах целевого свойства. Для решения задачи поиска возможных причин используется алгоритм Норриса, относящийся к теории формальных понятий. Из теории нечетких множеств, точнее, из систем нечеткого вывода взяты процедуры фаззификации и (частично) композиции. Это дает возможность говорить о степени уверенности в наличии (отсутствии) у объекта того или иного свойства (аналогичной степени принадлежности нечеткому множеству), а не просто делать утверждения о том, что объект обладает (или не обладает) этим свойством.

Основанная на предлагаемом методе система использует модуль автоматизированной подготовки данных и процедуру порождения нечетких продукций.

Эффективность работы предложенного метода сравнивается с эффективностью ДСМ-метода, использующего только «четкие» правила. Для этого в разработанной системе на этапе порождения гипотез о наличии у объекта целевого свойства применяются процедуры, аналогичные процедурам систем нечеткого вывода. Подобное сравнение необходимо для того, чтобы выяснить, имеет ли предлагаемый метод преимущества по сравнению с традиционным ДСМ-методом. Кроме того, обсуждаемый метод предполагает минимизацию участия экспертов в работе системы, в частности, на этапе предварительной подготовке данных.

ДСМ-МЕТОД

В середине XIX в. Джоном Стюартом Миллем [1] был предложен принцип поиска сходства, на основе которого в конце 1970-х гг. Виктор Константинович Финн разработал ДСМ-метод [2] (ДСМ – сокращение от «Джон Стюарт Милль»). ДСМ-метод используется как инструмент для анализа данных и обнаружения зависимостей, интерпретируемых как причинно-следственные связи. В качестве основных операций ДСМ-метод использует поиск сходства формализованных описаний объектов и применение формальных аналогов индуктивных методов Д.С. Милля [1].

ДСМ-метод использует структуру объекта для объяснения явлений, связанных с этим объектом. Поведение объекта описывается набором так называемых *целевых свойств*. Для простоты будем считать, что у нас имеется только одно целевое свойство (признак).

По отношению к целевому признаку множество объектов делится на четыре группы:

1) обладающих целевым признаком (множество положительных примеров);

2) не обладающих целевым признаком (множество отрицательных примеров);

3) тех, для которых сведения о наличии целевого признака противоречивы (множество противоречивых примеров);

4) тех, для которых сведения о наличии целевого признака отсутствуют (множество неопределенных примеров).

Объекты, с которыми работает ДСМ-метод, как правило, представлены в виде множеств. Возможная причина целевого признака в ДСМ-методе определяется как пересечение некоторого семейства множеств, представляющих объекты, обладающие целевым признаком.

Основные задачи ДСМ-метода:

- нахождение особенностей структуры объекта, являющихся возможными причинами наличия или отсутствия целевого свойства у объекта (порождение гипотез о возможных причинах целевых свойств);
- порождение гипотез о наличии (отсутствии) целевых свойств у новых (недоопределенных) объектов.

Структура объекта представляется в виде множества атомов. К такому представлению обычно приводится векторное представление объекта, при котором объект кодируется вещественным вектором, содержащим значения параметров объекта. Каждый параметр, в свою очередь, может быть представлен в виде признака или набора признаков, которые кодируются в виде булевого вектора, где 0 означает отсутствие, а 1 — наличие признака. После этого происходит обработка полученных векторов процедурами ДСМ-метода, для того чтобы в итоге определить наличие целевого свойства у новых (недоопределенных) объектов. Перед этапом определения наличия свойств у новых объектов из полученного множества причин удаляются противоречивые гипотезы.

В настоящей работе не будут использоваться противоречивые примеры, и применяемый алгоритм будет порождать только множества положительных и отрицательных гипотез.

АНАЛИЗ ФОРМАЛЬНЫХ ПОНЯТИЙ

В начале 1980-х гг. стало развиваться такое научное направление, как анализ формальных понятий (АФП). Первой работой, где были систематически изложены основы этого направления, считается статья Р. Вилле [3]. Анализ формальных понятий является одним из современных направлений прикладной алгебры, более конкретно — прикладной теории решеток. Наиболее полное изложение математических оснований анализа формальных понятий содержится в книге [4]. Анализ формальных понятий используется при решении различных задач искусственного интеллекта, но для нашей работы наиболее интересно применение анализа формальных понятий в интеллектуальном анализе данных.

Анализ формальных понятий оперирует объектами и признаками. Объект может быть представлен булевым вектором, размерность которого равна количеству признаков; *i*-й элемент (бит) этого булевого вектора равен 1, если объект имеет *i*-й признак, и 0, если у объекта этот признак отсутствует. Каждому такому булевому вектору соответствует множество признаков, которыми обладает объект.

При отождествлении объекта с множеством признаков, которыми он обладает, *формальное понятие* можно определить как упорядоченную пару множеств $\langle A,B\rangle$, где A — множество объектов, B — множество признаков,

$$B=\bigcap_{X\in A} X, \qquad A=\{Y\in G \mid B\subseteq Y\}$$

(здесь G — множество всех рассматриваемых объектов).

Поиск формальных понятий дает возможность найти все различные пересечения множеств признаков, представляющих объекты. Поэтому в ДСМ-методе могут использоваться алгоритмы, строящие множество всех формальных понятий, например алгоритм Норриса [5].

СИСТЕМЫ НЕЧЕТКОГО ВЫВОДА

Понятие нечеткого множества было введено Л. Заде в [6]. В книге [7] Л. Заде ввел очень важное понятие лингвистической переменной. Э. Мамдани использовал нечеткие множества и лингвистические переменные для создания первой системы нечеткого вывода, предназначенной для управления техническими устройствами [8].

Говоря неформально, лингвистическая переменная является переменной, которая может принимать как обычные числовые, так и лингвистические значения, представляющие собой словесную оценку количества или интенсивности. Например, переменная «температура» может принимать как числовые (36,6, 37,5, 39,2 и т.п.), так и лингвистические значения (нормальная, повышенная, высокая и т.п.). Каждому лингвистическому значению ставится в соответствие нечеткое множество со своей функцией принадлежности, показывающей, с какой степенью уверенности мы можем приписать переменной данное лингвистическое значение, если известно числовое значение этой переменной.

Системы нечеткого вывода позволяют по известным числовым значениям входных переменных получить заранее неизвестные числовые значения выходных переменных. Важным компонентом системы нечеткого вывода является база знаний, включающая нечеткие продукции — правила, содержащие имена и лингвистические значения входных и выходных переменных. Входные переменные содержатся в посылках, а выходные — в заключениях правил.

Пример нечеткой продукции: «Если температура высокая и давление высокое, то состояние — критическое». Здесь «температура» и «давление» — входные переменные, а «состояние» — выходная.

Работа системы нечеткого вывода делится на ряд стадий: фаззификация, нечеткий вывод, композиция и дефаззификация. На стадии нечеткого вывода вычисляется истинностное значение конъюнкции посылок нечеткой продукции (оно принадлежит отрезку $[0,1]$). Как правило, это истинностное значение становится истинностным значением заключения нечеткой продукции.

ОПИСАНИЕ ПРИНЦИПОВ РАБОТЫ СИСТЕМЫ, КОМБИНИРУЮЩЕЙ ДСМ-МЕТОД И СИСТЕМУ НЕЧЕТКОГО ВЫВОДА

Роль входных данных для описываемой системы играет числовая таблица. Строки таблицы соответствуют объектам, а столбцы — признакам. Имена объектов и признаков предполагаются известными. Один из столбцов таблицы соответствует целевому признаку. Здесь могут находиться значения 1 (наличие признака у объекта), 0 (отсутствие признака у объекта), «пусто» (незаполненная ячейка таблицы, что соответствует неопределенности). Все прочие столбцы должны быть заполнены числовыми значениями.

Подготовка данных включает следующие процедуры:

1) для каждого столбца (кроме целевого) находятся минимальное и максимальное значения, что позволяет определить диапазон изменения значений признака;

2) для каждого признака (кроме целевого) диапазон изменения делится на поддиапазоны, в простейшем случае происходит деление на три равных поддиапазона, соответствующих лингвистическим значениям «низкий», «средний», «высокий», для более тонкой настройки поддиапазонов требуются консультации с экспертами в предметной области;

3) строится бинарная (булева) таблица, в которой столбцы соответствуют поддиапазонам, в этих столбцах 1 означает попадание в поддиапазон;

4) может быть построена еще одна матрица, заполненная числами из отрезка [0,1], но в текущей версии алгоритма эта матрица не используется, поэтому строить ее необязательно.

В процессе тестирования (который будет описан далее) было выяснено, что исходные данные содержали большое количество «шумов», из-за которых существенная часть значений признаков попадала в один и тот же поддиапазон. В итоге на этапе формирования гипотез о возможных причинах наличия или отсутствия целевого признака порождалось большое количество противоречивых гипотез. Чтобы в дальнейшем избежать подобных трудностей, было решено модифицировать этап разбиения признаков на поддиапазоны. Для этого выполняются следующие действия:

– находится минимальное значение признака $V_{\min}$ и максимальное – $V_{\max}$;

– затем находится среднее значение признака M и множество значений признака разбивается на два класса: $K_<$ – содержащий значения признака, меньшие M, и $K_>$ – содержащий значения признака, большие M;

– потом находится среднее значение признака из $K_<$ (обозначим его через $M_<$) и из $K_>$ (обозначим его через $M_>$);

– итоговое разбиение на поддиапазоны получается следующим образом: первый поддиапазон – от $V_{\min}$ до $(M_< + M)/2$, второй поддиапазон – от $(M_< + M)/2$ до $(M + M_>)/2$, третий поддиапазон – от $(M + M_>)/2$ до $V_{\max}$.

Это позволяет получить более равномерное распределение признаков, а «шумы» в виде выбросов минимальных или максимальных значений не будут оказывать существенного влияния на разбиение. Однако в нашей статье описываемая система использует более простой способ кодирования данных, ориентированный на вычисление поддиапазонов только через минимальные и максимальные значения признаков.

Формирование гипотез о возможных причинах наличия или отсутствия целевого признака использует стандартную технику ДСМ-метода. Строки бинарной матрицы рассматриваются как представления множеств. Сходство объектов определяется как пересечение множеств. Для поиска пересечений используется один из классических алгоритмов, например алгоритм Норриса [6]. В результате выполнения описываемой процедуры будут сформированы гипотезы как о причинах наличия, так и причинах отсутствия целевого признака, причем каждая такая гипотеза является множеством признаков, представленным в виде булевого вектора.

Формирование нечетких продукций использует следующее обстоятельство: каждый столбец бинарной матрицы соответствует поддиапазону и может быть интерпретирован как пара $\langle$признак, лингвистическое значение$\rangle$. Для каждой гипотезы о возможной причине построим соответствующую нечеткую продукцию. Если в столбце, интерпретируемом как пара $\langle P, L\rangle$, находится 1, добавляем атомарную формулу P is L к посылкам этой нечеткой продукции. Если рассматриваемая гипотеза является гипотезой о причине *наличия* целевого признака T, то заключением правила будет формула T is present, в противном случае заключением правила является формула T is absent.

Для каждой атомарной формулы P is L, содержащейся хотя бы в одной из построенных нечетких продукций, определим функцию принадлежности.

Формирование гипотез о наличии или отсутствии целевого признака выполняется с использованием процедур системы нечеткого вывода. Для каждого неопределенного примера с помощью функций принадлежности находим истинностные значения посылок каждой продукции (фаззификация). Для каждой продукции находим истинностное значение ее заключения (нечеткий вывод). Если несколько продукций имеют одно заключение, то в качестве итогового истинностного значения заключения берем максимальное (часть процедуры композиции).

Для каждого объекта гипотеза о наличии/отсутствии целевого признака имеет следующий формат: $\langle o, p, a\rangle$, где o – имя (идентификатор) объекта, p – степень наличия целевого свойства (истинностное значение формулы T is present), a – степень отсутствия целевого свойства (истинностное значение формулы T is absent). Пороговые значения параметров p и a для принятия решений могут определяться с учетом мнения эксперта в предметной области.

Для общего описания работы алгоритма введем следующие обозначения:

O – множество неопределенных объектов,

Lpos – множество положительных гипотез (возможных причин наличия свойства),

Lneg – множество отрицательных гипотез (возможных причин отсутствия свойства),

i – идентификатор (ключ) объекта,

j – идентификатор (ключ) гипотезы,

k – номер бита в бинарном представление объекта и гипотезы,

0.. n – диапазон номеров битов,

Tnorm – t-норма, например, минимум или произведение,

SNorm – s-норма, например, максимум или алгебраическая сумма,

LinkF – функция, вычисляющая степень близости значений признаков, соответствующих биту k в объекте O[i] и гипотезе L[j].

Алгоритм будет представлен с помощью следующего псевдокода:

```
//  Обработка неопределенных объектов
For Each O[i] In O
    <o=i, p=0, a=0> // гипотеза о наличии целевого признака
//  Обработка положительных гипотез
    For Each L[j] In Lpos
        P = 1 // текущая степень наличия
        For Each k In 0..n
              P = Tnorm(P,
LinkF(L[j][k],O[i][k]))
        Next
        p = SNorm(p, P)  // общая степень наличия
    Next
//  Обработка отрицательных гипотез
    For Each L[j] In Lneg
          A=1 // текущая степень отсутствия
          For Each k In 1..n
                A=Tnorm(A,
LinkF(L[j][k],O[i][k]))
          Next
          a = SNorm(a, A)  // общая степень отсутствия
      Next
  Next
```

ОБЪЕКТНАЯ МОДЕЛЬ ДАННЫХ

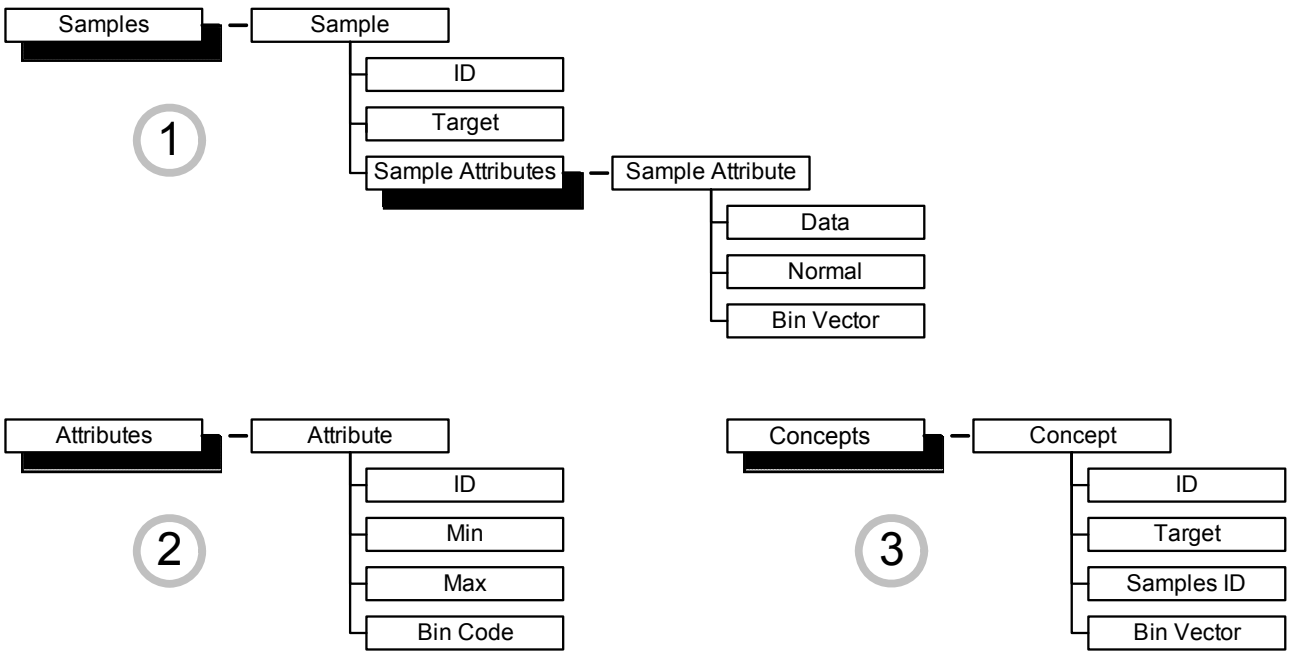
Для использования данных внутри системы была разработана модель, позволяющая хранить поступившую на вход информацию как в первоначальном (внешнем), так и в закодированном (внутреннем) представлении. Модель данных имеет древовидную структуру (см. рисунок), в ней три основных коллекции объектов:

1) входных примеров;
2) значений признаков;
3) гипотез.

Первая коллекция предназначена для хранения данных о входных примерах. Каждый пример имеет идентификатор и значения признаков, включая целевой. Множество нецелевых признаков представлено в виде трех векторов: вектора вещественных значений признаков, вектора нормированных значений и бинарного вектора, полученного в результате кодирования.

Коллекция значений признаков хранит информацию о максимальных и минимальных значениях признаков, которые используются при нормировании данных, а также бинарное представление данного признака. Оно используется для декодирования информации о значении данного признака из построенных гипотез.

Третья коллекция используется для хранения полученных гипотез о возможных причинах наличия у объектов целевого свойства. Каждая гипотеза состоит из идентификатора, множества номеров вошедших примеров и пересечения признаков этих примеров.



Коллекции для представления данных

ТЕСТИРОВАНИЕ СИСТЕМЫ

Для оценки эффективности работы системы использовались результаты объективных исследований онкологических заболеваний, взятые из общедоступного репозитория UCI Machine Learning Repository, содержащего данные для тестирования различных методов машинного обучения. Общий объем выборки составляет 569 объектов, каждый включает 30 признаков и 1 целевой атрибут – положительный или отрицательный результат анализа. Часть из поступивших на вход объектов отводилась на тестовую выборку. По этим объектам оценивалась точность и полнота порождения гипотез.

Целью тестирования было сравнение эффективности обычной ДСМ-системы и ДСМ-системы с использованием (на этапе предсказания наличия/отсутствия целевых свойств) нечетких продукций, описанной в настоящей статье. Обеим системам предоставлялись одинаковые наборы тестовых и обучающих примеров, а для определения сходства объектов использовался алгоритм Норриса [6]. Таким образом, множества гипотез о возможных причинах целевого свойства для обеих систем должны были совпадать. Это позволило оценить различие в работе методов на последнем этапе – при формировании гипотез о наличии/отсутствии у объектов целевого свойства.

Чтобы определить, какой объем выборки необходим для построения гипотез о наличии/отсутствии целевого свойства, изменялся процент объектов, отведенных на тестовую выборку. Минимальное количество тестовых объектов – 10% от общего количества примеров, максимальное – 70%. Тестовые объекты выбирались случайным образом.

Другим настраиваемым параметром была граница, после которой предсказанное нечетким модулем значение считалось положительным. Была проведена серия тестов для значений 0,9, 0,8 и 0,7.

Каждая серия состояла из 100 экспериментов, в которых оставались неизменными входное множество объектов, процент тестовых примеров и упомянутая выше граница. В ходе первых тестов было выявлено, что при порождении гипотез о возможных причинах происходил комбинаторный взрыв, и количество положительных и отрицательных гипотез часто превосходило 500 000. Для ускорения последующего анализа было решено отказаться от удаления положительных и отрицательных гипотез, имеющих одинаковый набор атрибутов.

На выходе система строила таблицы как для результатов отдельных экспериментов, так и для итогов всей серии экспериментов.

В табл. 1 представлены результаты работы нечетких правил порождения гипотез о наличии/отсутствии целевых свойств. Первый столбец – номера тестовых объектов, второй – степень доверия положительной гипотезы, третий – степень доверия отрицательной гипотезы, четвертый – итоговое значение, разность между степенями доверия к аргументам «за» и «против» наличия у объекта целевого свойства. Последний столбец – информация о действительном значении целевого свойства.

Таблица 1

Фрагмент таблицы результатов одного эксперимента

	За	Против	Итог	Действительное
1	1	0	1	1
2	1	1	0	0
3	1	0,25	0,75	1
4	1	0,25	0,75	1
5	1	0,25	0,75	1
6	1	0,5	0,5	1

Для оценки качества работы системы будем использовать понятия полноты и точности предсказания.

Точность определяется как отношение числа правильно определенных примеров к их общему числу:

$$\text{Precision} = \frac{\left|S_{posr} \cup S_{negr}\right|}{\left|S_{pos} \cup S_{neg}\right|},$$

где S_{posr} и S_{negr} — множество правильно определенных примеров, соответственно, обладающих и не обладающих целевым свойством, а S_{pos} и S_{neg} – множества всех доопределенных примеров положительных и отрицательных примеров, соответственно.

Полнота является отношением общего числа определенных примеров к общему количеству примеров:

$$\text{Recall} = \frac{\left|S_{pos} \cup S_{neg}\right|}{\left|S\right|}.$$

По итогам экспериментов было выявлено, что полнота нечеткого модуля составляет примерно 70%, а ДСМ-метода – 40%. Из-за наличия большого количества противоречивых гипотез ДСМ-метод находил как гипотезы «за», так и гипотезы «против» в случаях, когда тестовый объект не обладал целевым признаком. Поэтому вместо ответа –1 (отсутствие целевого признака) ДСМ-метод давал значение 0, т.е. неопределенность. Если же этап удаления противоречивых гипотез использовался, то из гипотез, порожденных алгоритмом Норриса, исключалось около 50%. Тогда полнота нечеткого модуля составляла 75–85%, а ДСМ — 70–80%. В таком случае время работы системы возрастало многократно, что не дало возможности провести достаточное для анализа количество экспериментов.

В итоге результаты экспериментов показали, что прирост полноты для нечеткой системы относительно ДСМ-системы составляет от 5% до 30% в зависимости от включения этапа удаления противоречивых гипотез. Этот прирост дает возможность прогнозировать с большей точностью наличие целевого признака у анализируемого объекта при использовании методов нечеткой логики.

РЕЗУЛЬТАТЫ РАБОТЫ СИСТЕМЫ

Кроме сравнения эффективности работы предложенного метода с эффективностью четкой ДСМ-системы, был проведен ряд экспериментов, чтобы выявить необходимое количество примеров, требуемое для обучения системы (табл. 2).

Таблица 2

Зависимость полноты и точности от процента тестовых примеров

Доля тестовых примеров, %	Полнота		Точность	
	Fuzzy	JSM	Fuzzy	JSM
70	66,9	35,3	84	81
60	68,2	36,9	86	84
50	68,4	39,7	87	85
40	70,2	40,2	87	87
30	73,9	44,9	91	92
20	70,1	45,4	89	91

Из полученных результатов можно сделать вывод, что полнота предложенного метода мало зависит от объема обучающей выборки. Кроме этого, не исключена возможность так называемого «переобучения» системы, когда при слишком большой обучающей выборке точность на тестовых примерах падает. Оптимальная доля обучающих примеров — 70%, что, примерено, составляет 398 из 569 объектов используемых в качестве входных данных. Таким образом, сравнительно небольшого количество обучающих примеров достаточно для получения лучшего результата предложенного метода. С другой стороны, как было сказано выше, тестовые объекты содержали большое количество шумов, что могло привести к потере точности. Внесение некоторых изменений в процесс кодирования, уменьшающий влияние шумов, может привести к лучшим результатам работы как используемого здесь варианта ДСМ-системы, так и предложенного метода, хотя применение нечеткого модуля на данном этапе уже показало его устойчивость к шумам.

СПИСОК ЛИТЕРАТУРЫ

1. Милль Д.С. Система логики силлогистической и индуктивной: Изложение принципов доказательства в связи с методами научного исследования / пер. с англ., изд. 5, испр. и доп. - М.: ЛЕНАНД, 2011. — 832 с.
2. Финн В.К. О возможности формализации правдоподобных рассуждений средствами многозначных логик // Всесоюзн. симп. по логике и методологии науки. — Киев: Наукова думка, 1976. — С. 82–83.
3. Wille R. Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts // Ordered Sets / ed. by I. Rival. — Dordrecht; Boston: Reidel, 1982. — P. 445–470.
4. Ganter B., Wille R. Formal Concept Analysis: Mathematical Foundations. – Berlin: Springer, 1999.
5. Norris E.M. An Algorithm for Computing the Maximal Rectangles in a Binary Relation // Revue Roumaine de Mathématiques Pures et Appliquées. – 1978. - № 23(2). - P. 243–250.
6. Zadeh L.A. Fuzzy sets // Information and Control. – 1965. -Vol.8, N 3. - P. 338–353.
7. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. - М.: Мир, 1976. — 167 с.
8. Mamdani E.H., Assilian S. An experiment in linguistic synthesis with a fuzzy logic controller // Int. J. Man-machine Studies. -1975. - Vol. 7. – P. 1–13.

Материал поступил в редакцию 02.07.12.

Сведения об авторе

КОВТУН Всеволод Александрович – аспирант Российского государственного гуманитарного университета, Москва

E-mail: kov-vsev@mail.ru

УДК [004.89:510.64]:007.52-026.26

Т.А. Волкова

Вариант динамического ДСМ-метода для обучения мобильного робота

Описывается создание адаптивного робота, обучение которого реализовано с помощью динамического ДСМ-метода. Приведена модификация ДСМ-метода, в которой примеры после порождения из них гипотез могут быть удалены из обучающей выборки. Описаны практические эксперименты с мобильным роботом.

Ключевые слова: *динамический ДСМ, интеллектуальная робототехника, машинное обучение*

ВВЕДЕНИЕ

В настоящее время наблюдается активное внедрение методов искусственного интеллекта в робототехнику в таких областях, как распознавание изображений, планирование, коллективное поведение.

В работе [1] приведена классификация роботов:

1. Программные (действующие по заранее заданной программе).

2. Адаптивные (способные изменять поведение под воздействием окружающей среды).

3. Интеллектуальные (способные выполнять задачу, поставленную в общем виде).

Программные роботы представлены в мире в основном промышленными роботами, выполняющими небольшой набор узкоспециализированных задач. Из самых известных адаптивных роботов современности стоит упомянуть известных шагающих роботов Big Dog и Little Dog компании Boston Dynamics, а также домашнего робота PR2 компании Willow Garage. Обзор современного состояния интеллектуальных роботов изложен в статье [2].

Особую роль в создании адаптивных и интеллектуальных роботов играют методы машинного обучения. В настоящей работе использовался ДСМ-метод автоматического порождения гипотез.

1. ПОСТАНОВКА ЗАДАЧИ ОБУЧЕНИЯ

Робот должен обучаться, чтобы совершенствовать своё поведение по мере накопления опыта. Робот существует в среде, все объекты которой имеют описания и принадлежат к некоторым классам из заранее определённого множества. Робот может сам получать факты из окружающей среды, поэтому обучение должно быть *индуктивным* - основанным на выявлении закономерностей в эмпирических данных. Наша задача - построение *индуктивного классификатора*.

В качестве *обучающей выборки* возьмем множество пар «обстановка – действие». Под *обстановкой* понимается состояние окружающей среды, зафиксированное рецепторами робота – освещенность, цвет поверхности, расстояние до препятствия, угол по отношению к некоторому ориентиру (например, к северу). Под *действием* будем понимать влияние эффекторов (исполнительных механизмов) робота на окружающую среду – вращение колес, свет, звук. Возможны более сложные действия, состоящие из простых: например, действие «Атака» является комбинацией езды вперед и света фонариком, а «Защита» - комбинацией езды назад и звука.

Обучение будет осуществляться на *примерах* (положительных и отрицательных), которые выдаёт учитель. Формирование положительного примера назовём *поощрением*, отрицательного – *наказанием*, обобщение нескольких примеров (только положительных или только отрицательных) - *гипотезой*.

Обучение (обобщение, получение гипотез) и *действие* (применение гипотез) повторяются циклически. Кроме того, должен существовать некоторый критерий, показывающий, что обучение проведено и теперь можно перейти к работе.

Обучение является *интерактивным*: сигналы поощрения и наказания выдаются непосредственно в ходе работы системы.

2. ДИНАМИЧЕСКИЙ ДСМ-МЕТОД

В настоящей работе в качестве метода обучения используется ДСМ-метод автоматического порождения гипотез [3], [4], который обобщает в гипотезах информацию, полученную из обучающей выборки (шаг индукции), затем применяет эти гипотезы для классификации неизвестных объектов (шаг аналогии), а также имеет критерий достаточного основания правдоподобного вывода (шаг абдукции).

Существующие ДСМ-системы работают с заранее заданным набором примеров. Примеры могут добавляться экспертом, если оказывается, что их недостаточно, но в этом случае каждый раз происходит перезапуск системы. Для робота такой вариант неприемлем из-за большой вычислительной сложности.

В работе [5] предложена модель *динамического ДСМ-метода*, в котором примеры могут добавляться в процессе функционирования системы. «В

отличие от классического ДСМ-метода, который работает с замкнутым множеством исходных примеров и заранее определёнными их свойствами, динамический ДСМ-метод позволяет работать в открытой среде с неизвестным заранее количеством примеров».

Такая особенность динамического ДСМ-метода накладывает определённые требования на алгоритм поиска пересечений. Очевидно, что он должен быть пошаговым, а не пакетным - множество пересечений не должно полностью пересчитываться каждый раз. Исходя из этого, в нашей работе был выбран алгоритм Норриса с некоторыми модификациями. Ключевой проблемой метода Норриса является определение каноничности порождения пересечения: пересечение объектов называется каноничным, если оно порождается из родителей в порядке возрастания их номеров. Это необходимо для того, чтобы не порождать одно и то же пересечение многократно.

Самым же существенным отличием динамического ДСМ-метода от классического является отсутствие критерия завершения. В классическом ДСМ-методе им является достижение насыщения – невозможность породить новые гипотезы. В динамическом варианте база фактов постоянно расширяется, и требуется сформулировать другое условие. Им может стать абдукция – познавательная процедура ДСМ-метода.

3. МОДЕЛЬНАЯ ЗАДАЧА

В мире робота существуют объекты, которые он учится классифицировать. Например, объектами могут быть совокупности параметров окружающей среды в некоторые моменты времени.

Для обучения робота была выбрана несложная модельная задача, которую можно было решить с имеющимся набором датчиков и исполнительных механизмов. Учитель, в роли которого выступает человек, обучает робота правилам вида «если – то», например, «если темно и видно препятствие – входи в режим защиты».

Признаки:

1. «Робот на черном/на белом».
2. «Темно/светло».
3. «Препятствие впереди/нет препятствия».
4. «Угол по отношению к северу составляет α».

Свойства (сложные действия)

1. Атака (Ехать вперед и светить фонариком).
2. Защита (Ехать назад и пищать).
3. Поиск (Повернуться влево на 90 градусов).

Для применения ДСМ-метода необходимо задать операции сходства на фрагментах и на множествах свойств. Обычно фрагменты и свойства представляются битовыми строками. Поэтому операция сходства соответствует побитовому умножению (рис. 1).

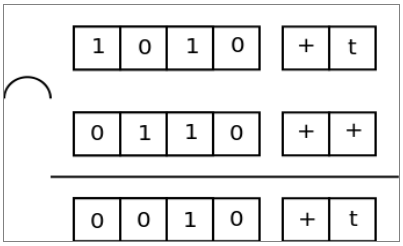


Рис.1. Операция пересечения

Чтобы строить ДСМ-объекты из данных рецепторов, необходимо количественные показания (угол по отношению к северу, уровень освещённости, показания дальномеров) преобразовать в битовые. Это осуществлялось введением нескольких порогов, но в будущем необходимо разработать адекватное представление для сигналов робота.

4. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ОБУЧЕНИЯ

Обучение робота может осуществляться двумя способами: выдачей роботу команд на исполнение, либо оцениванием автономного поведения робота. Первый способ удобен тем, что прост для учителя, но имеет недостаток: учитель может выдать только положительные примеры (в форме своих команд). Этот способ был реализован в работе словенских участников проекта XPERO [6], которые использовали ILP (индуктивное логическое программирование) в качестве метода машинного обучения, а также в задаче с обучением робота езде по линии [1]. При такой схеме становится невозможно переучить робота.

В статье [1] описан другой способ: реализация схемы поощрения и наказания, при которой после окончания автономного действия робот ждет оценки учителя некоторое время. Этот способ позволяет формировать как положительные, так и отрицательные примеры, но сложнее для учителя – необходимо отслеживать поведение робота и успевать реагировать.

Представляется разумным комбинирование этих двух способов: обучение путём ручной выдачи команд и последующая корректировка поведения робота сигналами поощрения и наказания.

В настоящей работе использован второй способ: процесс жизнедеятельности робота разделён на *фазы* действия и ожидания. В фазе действия робот фиксирует обстановку и выбирает правило, которое подходит для данного случая; если такое правило не находится, то робот совершает случайное действие. В фазе ожидания робот стоит некоторое время и ждёт сигнала от учителя. Правильный поступок робота учитель поощряет (робот получает положительный пример). Если же робот ошибается, то учитель наказывает его (робот получает отрицательный пример). Возможна реализация, при которой учитель явно предъявляет положительный пример (говорит, как именно нужно поступать в этой ситуации, какое действие выбрать из множества возможных). Это нужно для того, чтобы робот с самого начала обучения мог набрать достаточное количество положительных примеров (рис. 2).

Рис.2. Схема обучения робота

5. ПОСТАНОВКА ЗАДАЧИ «ЗАБЫВАНИЯ»

Максимальное число примеров экспоненциально зависит от количества признаков, поэтому возникает проблема хранения примеров в памяти робота. "Забывание примеров" существенно и по другой причине. Хотелось бы, чтобы со временем старая обучающая выборка не участвовала в порождении новых гипотез.

Замечено, что большинство примеров совпадает с пересечениями, у которых один родитель. Совместно с Д.В. Виноградовым (ВИНИТИ РАН) нами был разработан вариант ДСМ-метода, где примеры сохраняются как пересечения с одним родителем. После этого примеры могут быть забыты.

Для проверки каноничности итерация по примерам заменяется итерацией по пересечениям (недостаток - потенциально высокая вычислительная сложность). Однако, имеется выигрыш по объему занимаемой памяти: в классическом ДСМ-методе в худшем случае мы имеем 2^n пересечений и 2^n примеров, тогда как в настоящей реализации – только 2^n пересечений, среди которых есть и примеры.

Таким образом, этот вариант ДСМ-метода можно использовать, когда признаков мало, а примеров - много. У робота примеры непрерывно поступают, поэтому использование такой модификации оправданно.

В некоторых пересечениях «скрываются» примеры: их фрагменты совпадают с фрагментами примеров. Такие пересечения должны рассматриваться наравне с пересечениями с одним родителем. Обычно число пересечений со «скрытыми родителями» не превосходит трети от всех.

Чтобы учесть такие пересечения, нами была разработана модификация метода Норриса. В ней на стадии проверки на относительную и абсолютную каноничность итерация по предыдущим примерам заменяется итерацией по пересечениям со «скрытыми родителями». При добавлении нового пересечения или модификации старого проверяется, совпадают ли фрагмент текущего примера и фрагмент текущего пересечения. В этом случае у пересечения ставится флаг "скрытые родители". Такой же флаг ставится и при добавлении одноэлементного пересечения.

6. УЧЁТ ЧИСЛА ПРИМЕРОВ

Робот живет в изменяющейся среде (по терминологии Д.А. Поспелова, в мире с "переключающимися средами"). Он должен быть способен адаптироваться к изменениям - переобучаться при поступлении сигнала поощрения или наказания, причем максимально быстро.

Однако имеется и в некотором смысле противоположное утверждение: робот должен быть устойчивым к шумам и выбросам в данных. Это накладывает ограничения на используемые методы обучения: они должны быть устойчивы к одиночным (редким) ошибкам, таким, как ложное срабатывание одного из датчиков или недосмотр учителя.

В классическом ДСМ-методе существует стратегия "запрет на контрпример", в которой гипотеза уничтожается, если в обучающей выборке находится более чем n контрпримеров (порог n задается экспертом эмпирически). Контрпримером называется пример, который объясняется гипотезой, но имеет отличающийся от нее знак. "Запрет на контрпример" позволяет избавляться от противоречивых гипотез; в нашем случае такой запрет необходим, чтобы робот мог переучиваться – исключать старые и неактуальные гипотезы и приобретать новые.

В существующих ДСМ-системах новый пример проверяется на уникальность. Повторные примеры не рассматриваются, так как они не интересуют эксперта. В случае с роботом повторные примеры стоит учитывать: чем больше встретилось примеров, удовлетворяющих некоторой гипотезе, тем больше следует доверять гипотезе. Примеры приходят роботу постоянно, поэтому часто встречаются дубликаты.

Если использовать "запрет на контрпример", то остается открытым вопрос о том, как назначать пороги для отсеивания контрпримеров: назначение их постоянными не представляется разумным.

7. ЭКСПЕРИМЕНТЫ

Эксперименты проводились на мобильном роботе POP-BOT, который представляет собой автономную тележку с микроконтроллером Arduino, простыми рецепторами и эффекторами. Интеллектуальная система находится на компьютере и передает управляющие сигналы роботу по Bluetooth. Архитектура всего программного комплекса изображена на рис. 3.

В результате эксперимента робот успешно обучился нескольким стратегиям поведения. Кроме того, были проведены эксперименты по адаптации робота в искусственно измененной ситуации.

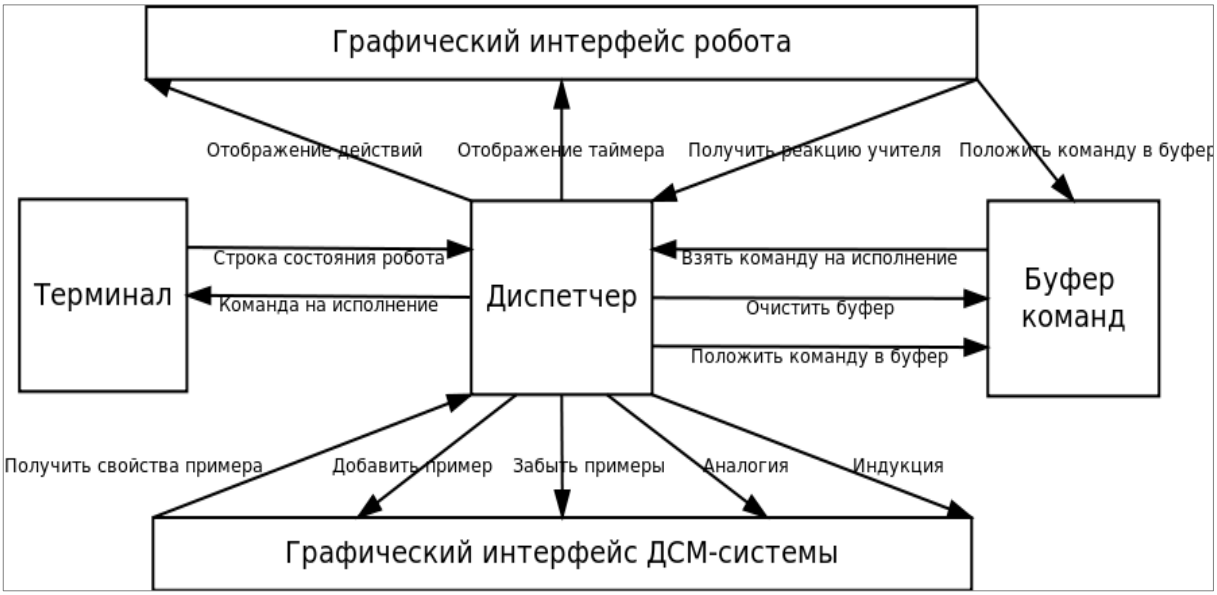


Рис.3. Архитектура системы

Вот пример обучения. Мы хотим научить робота, как вести себя при встрече с препятствием:

1) сперва научим его поведению в случае отсутствия препятствия. Выдадим два положительных примера: "если темно и нет препятствия – атака" и "если светло и нет препятствия – атака". Робот породит гипотезу "если нет препятствия – атака";

2) аналогично робот породит гипотезы "если темно и есть препятствие – защита" и "если светло и есть препятствие – поиск";

3) теперь допустим, что среда изменилась и робот должен при отсутствии препятствия всегда входить в режим поиска. Выдадим роботу хотя бы один контрпример: "если темно и нет препятствия – не атака", и полученная ранее гипотеза "если нет препятствия – атака" будет уничтожена. Теперь, как и в пункте 1, робот может быть научен новому правилу поведения в данной ситуации.

Пример хорошо иллюстрирует описанную проблему: гипотеза, подразумеваемая учителем и порожденная из двух примеров, уничтожается одним-единственным контрпримером.

ЗАКЛЮЧЕНИЕ

Проведенные эксперименты показывают, что простой перенос на роботов существующих ДСМ-систем был бы неэффективен из-за особенностей робота как системы реального времени.

Результаты нашей работы могут быть использованы для наглядной демонстрации обучения при помощи ДСМ-метода в экспозиции Политехнического музея.

В ближайшей перспективе предполагается разработать схему оценивания гипотез и реализовать специфическую для робота абдукцию.

В дальнейшем мы планируем применить и другие методы машинного обучения для адаптации робота.

СПИСОК ЛИТЕРАТУРЫ

1. Добрынин Д.А., Карпов В.Э. Моделирование некоторых форм адаптивного поведения интеллектуальных роботов // Информационные технологии и вычислительные системы. - 2006. - № 2. - С.45-56.
2. Добрынин Д.А. Интеллектуальные роботы: состояние и перспективы // XII национальная конференция по искусственному интеллекту с международным участием КИИ-2010 (20-24 сентября 2010 г., Тверь): Труды конференции в 4-т. Т.4. – М : Физматлит, 2010.
3. ДСМ-метод автоматического порождения гипотез: Логические и эпистемологические основания / сост. О.М. Аншаков, Е.Ф. Фабрикантова; под общ. ред. О.М. Аншакова. – М.: Книжный дом «Либроком», 2009. – 433 с.
4. Автоматическое порождение гипотез в интеллектуальных системах / сост Е.С. Панкратова, В.К. Финн; под общ. ред. В.К.Финна. Предисл. Ю.М. Арского. – М.: Книжный дом «Либроком», 2009.
5. Добрынин Д.А. Динамический ДСМ-метод в задаче управления интеллектуальным роботом. Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006 (25-28 сентября 2006 г., г. Обнинск): Труды конференции. В 3-т. Т.2. - М: Физматлит, 2006. - 310 с.
6. Leban Gregor, Zabkar Jure, Bratko Ivan. An experiment in robot discovery with ILP // Inductive Logic Programming, 18th International Conference (ILP 2008). - Prague, 10-12 September 2008, Proceedings. - P. 77-90.

Материал поступил в редакцию 02.07.12.

Сведения об авторе

ВОЛКОВА Татьяна Александровна – аспирант ВИНИТИ РАН, Москва
E-mail: robofreak@mail.ru

Российская академия наук

Федеральное государственное бюджетное учреждение науки
ВСЕРОССИЙСКИЙ ИНСТИТУТ НАУЧНОЙ И ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ
РОССИЙСКОЙ АКАДЕМИИ НАУК

предлагает научным работникам, аспирантам и другим специалистам в области естественных, точных и технических наук, желающим быстро и эффективно опубликовать результаты своей научной и научно-производственной деятельности, использовать способ публикации своих работ через *систему депонирования.*

«Депонирование (передача на хранение) – особый метод публикации научных работ (отдельных статей, обзоров, монографий, сборников научных трудов, материалов научных мероприятий – конференций, симпозиумов, съездов, семинаров) узкоспециального профиля, разрешенных в установленном порядке к открытому опубликованию, которые нецелесообразно издавать полиграфическим способом печати, а также работ широкого профиля, срочная информация о которых необходима для утверждения их приоритета. Депонирование предусматривает прием, учет, регистрацию, хранение научных работ и обязательное размещение информации о них в специальных информационных изданиях».

Подготовка и передача на депонирование научных работ происходит в соответствии с «Инструкцией о порядке депонирования научных работ по естественным, техническим, социальным и гуманитарным наукам» (М., 2003).

Результатом депонирования является публикация информации о депонированных научных работах в информационных изданиях ВИНИТИ РАН – реферативном журнале и аннотированном библиографическом указателе «Депонированные научные работы».

В соответствии с “Положением о порядке присуждения ученых степеней”, утвержденным Постановлением Правительства Российской Федерации от 30.01.2002 № 74 (в ред. Постановлений Правительства РФ от 20.04.2006 № 227, от 02.06.2008 № 424, от 20.06.2011 № 475), научные работы, депонированные в организациях государственной системы научно-технической информации, признаны публикациями, учитывающимися при защите кандидатских и докторских диссертаций.

Подать научную работу на депонирование можно обратившись в Отдел депонирования ВИНИТИ РАН по адресу:

125190, Москва, ул. Усиевича, 20.
ВИНИТИ РАН, Отдел депонирования научных работ.
Тел.: 8 (499) 155-43-28, Факс: 8 (499) 943-00-60.
e-mail: dep@viniti.ru

С инструкцией о порядке депонирования можно ознакомиться на сайте ВИНИТИ РАН: **http://www.viniti.ru**

БАЗА ДАННЫХ ВИНИТИ РАН

ВИНИТИ предлагает к использованию через WWW-сервер (**http://www.viniti.ru**) крупнейшую Федеральную базу отечественных и зарубежных публикаций по естественным, точным и техническим наукам. БД ВИНИТИ РАН генерируется с 1981 г., обновляется ежемесячно, пополнение составляет около 1 млн документов в год. БД ВИНИТИ представлена ретроспективными тематическими фрагментами и единой политематической БД (ретроспектива с 2001 г.), объединяющей все тематические фрагменты БД ВИНИТИ.

БД ВИНИТИ РАН в сети INTERNET

Сервер ВИНИТИ – http://www.viniti.ru – обеспечивает on-line доступ к Базе данных ВИНИТИ РАН круглосуточно без выходных.

На основе БД ВИНИТИ РАН предоставляются следующие услуги:

- Диалоговый поиск научно-технической информации **в режиме on-line**;
- **Демо-версия**, позволяющая ознакомиться с основными функциями поисковой системы, составом данных, формами представления документов и получить навыки работы с системой;
- **Поисковые эксперты** ВИНИТИ выполнят тематический поиск по разовым или постоянным запросам, а также окажут **консультационные услуги**.

БД ВИНИТИ РАН на CD-ROM

Любые наборы тематических фрагментов БД ВИНИТИ или их разделов могут быть предоставлены **на CD-ROM в поисковой системе (ИПС) "Сокол"**, обеспечивающей все поисковые функции, доступные в режиме on-line:

- Поиск можно вести в годовом или ретроспективном массиве (за несколько лет сразу) в одном или нескольких тематических фрагментах .
- Поиск по словам и любым словосочетаниям из заглавия, реферата, ключевых слов.
- Использование года, языка, рубрик, шифров тематических разделов БД для уточнения поиска.
- Поиск по словарю, выполняющему функции многоаспектного указателя, в том числе авторского, предметного, источников, индексов МПК, номеров патентных документов и депонированных рукописей и т.д.
- Возможность запоминания запросов для последующего использования и/или редактирования их.
- Чтение документов не только как в РЖ (последовательный просмотр документов одного номера за другим), но и чтение документов нужных тематических фрагментов (разделов) по оглавлению за весь период заказанной ретроспективы.

ИПС "Сокол" является прикладной программой Microsoft Windows.

Любые наборы тематических фрагментов БД ВИНИТИ или их разделов могут быть подготовлены **в коммуникативных форматах** ISO-2709, МЕКОФ, txt на любых видах электронных носителей.

Продукты предоставляются на договорной основе.

Информационная служба БД ВИНИТИ: 125190, Москва, ул. Усиевича 20, ВИНИТИ
Телефон: (499) 155-45-01, 155-45-02, **Факс:** (499) 152-62-31 **e-mail:** csbd@viniti.ru

Москва, ВИНИТИ РАН
28–30 ноября 2012 года

***8-я Международная конференция* «НТИ-2012»,**
посвященная 60-летию ВИНИТИ РАН

«АКТУАЛЬНЫЕ ПРОБЛЕМЫ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ НАУКИ, АНАЛИТИЧЕСКОЙ И ИННОВАЦИОННОЙ ДЕЯТЕЛЬНОСТИ»

Для участия **в «НТИ-2012»** приглашаются специалисты в области информационных технологий, телекоммуникаций, создатели и потребители информационных продуктов и услуг, ученые и специалисты РАН, вузовской и отраслевой науки, работники информационных центров и библиотек, служб распространения информационных продуктов и услуг.

Доклады или тезисы докладов, которые будут опубликованы в специальном сборнике, направлять в Оргкомитет конференции «НТИ-2012».

Планируется проведение пленарных заседаний, круглых столов и работа по секциям.

Адрес: Россия, 125190, Москва, ул. Усиевича, 20, Всероссийский институт научной и технической информации (ВИНИТИ РАН), ОНИИР, Оргкомитет **«НТИ-2012».**
Тел: (495) 155-44-22, 155-44-29, 152-64-41,
Факс: (495) 152-54-92, 943-00-60
E-mail: conf@viniti.ru , market@viniti.ru
http://www.viniti.ru