© 2022 г. Д.В. ВИНОГРАДОВ, д-р физ.-мат. наук
(vinogradov.d.w@gmail.com)
(Вычислительный центр им. А.А. Дородницына
Федерального исследовательского центра
“Информатика и управление” РАН, Москва)

АЛГЕБРАИЧЕСКОЕ МАШИННОЕ ОБУЧЕНИЕ: УПОР НА ЭФФЕКТИВНОСТЬ[1]

Представлен обзор современного состояния исследований по алгебраическому машинному обучению. Основной упор сделан на вопросы вычислительной сложности. Ключевыми идеями являются использование методов теории решеток и вероятностных алгоритмов, основанных на цепях Маркова.

Ключевые слова: вероятностные алгоритмы, сложность вычислений, решетки, машинное обучение, переобучение.

DOI: 10.31857/S0005231022060022, **EDN:** ACGDEV

1. Введение

В последнее время на волне успехов обучения нейронных сетей разных видов (рекуррентных, сверточных, LSTM и др.) наблюдается тенденция отказа классическим (“old-school symbolical”) методам извлечения знаний (например, ДСМ-методу [1] порождения гипотез) в праве считаться методами искусственного интеллекта. В этой ситуации отчасти несут ответственность создатели таких методов, не уделяющие достаточного внимания вопросам сложности вычислений.

Цель настоящей статьи — попытаться исправить это ложное мнение, объяснив, как использование вероятностных алгоритмов может победить “проклятие размерности”. Сосредоточимся на варианте алгебраического машинного обучения, основанного на бинарной операции сходства. Описанный подход назовем “алгебраическим”, потому что он пытается приближенно восстановить зависимость как набор элементов конечно-порожденной алгебры — решетки кандидатов (в гипотезы о причинах). Будем использовать идеи и результаты из Анализа формальных понятий (Formal Concept Analysis, FCA) [2] — современного раздела теории решеток [3].

Для понимания статьи требуется минимальное знакомство с теорией вероятностей (в объеме базового курса) и цепей Маркова (теорема о невозвратных состояниях). План работы следующий.

[1] Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 18-29-03063мк).

Раздел 2 будет посвящен парадигме Л. Вэльянта [4] — В.Н. Вапника–А.Я. Червоненкиса [5] “вероятно приближенно-корректного” (ВПК) обучения. Мы продемонстрируем этот подход в случае конъюнктивных понятий.

В разделе 3 будут обсуждаться проблемы классического подхода к извлечению знаний с помощью бинарной операции сходства (ДСМ-метода). Для этого предварительно напомним некоторые понятия FCA [2]. Затем обсудим случай булевой алгебры, который ставит барьер для прямого подхода к поиску гипотез с точки зрения теории сложности вычислений. Наконец, будет представлен результат автора о переобучении — возникновении “фантомных” кандидатов, который обосновывает вероятность допущения ошибок при предсказании.

Раздел 4 содержит описание вероятностных алгоритмов поиска сходств и исследование их алгоритмических свойств.

В разделе 5 будут описаны процедуры извлечения знаний с помощью вероятностного подхода. Будет выведена улучшенная оценка на число гипотез (кандидатов, не имеющих контрпримеров), чтобы с заданной надежностью правильно предсказать все достаточно важные тестовые примеры, предъявленные на прогноз наличия целевого свойства для оценки качества порожденных гипотез.

Наконец, в разделе 6 позволим себе сформулировать критерий дискриминации данных от знаний с точки зрения специалиста по теории сложности вычислений. Приводятся примеры задач Эйлера о мостах и коммивояжера, в котором такое различие видится наиболее ярко.

2. Вероятно приближенно-корректное обучение

Для пропедевтических целей начнем с антропоморфного примера вероятно приближенно-корректного обучения по Вэльянту–Вапнику–Червоненкису.

В некоем племени начинают готовить преемника стареющему вождю. Тестируют способность обучаться новому. Для этого кандидат в вожди должен отправиться с шаманом в джунгли с целью обучиться правилу, какие растения лекарственные, а какие нет. Перед выходом ученик должен сказать, сколько растений m он хочет изучить. Если это число окажется неразумно большим, то его дисквалифицируют.

Для очередного растения $\vec{x}_t \in X_n$ шаман говорит, лекарственное ли это растение (положителен ли $c(\vec{x}_t) = 1$ этот пример) или нет (контрпример для искомого правила — отрицательный пример $c(\vec{x}_t) = 0$). Все примеры $\vec{x}_t \in X_n$ появляются независимо и с одинаковой вероятностью (Это допущение не слишком реалистично, но именно оно позволяет работать усиленному закону больших чисел.). Когда они увидят объявленное учеником число m обучающих примеров $S = \{\langle \vec{x}_1, c(\vec{x}_1)\rangle, \ldots, \langle \vec{x}_m, c(\vec{x}_m)\rangle\}$, шаман с учеником возвращаются домой.

Затем ученик должен найти правило $h = h_S : X_n \to \{0, 1\}$, как распозна-

вать лекарственные растения. Эффективность этого шага иногда игнорируется, но это очень важно!

Наконец, ученик с шаманом отправляются в (те же самые) джунгли. Ученик обязан классифицировать первое встретившееся растение (тестовый пример) $\vec{x} \in X_n$, который появляется независимо от обучающей выборки и с распределением, совпадающим с обучающим. Если классификация ученика не совпадет с классификацией шамана (= неверна) $h_S(\vec{x}) \neq c(\vec{x})$, то ученика дисквалифицируют.

Задачами ученика являются:

1) выбрать язык описания объектов (задать область X_n обучающих примеров);

2) научиться оценивать снизу число m случайных обучающих примеров $S = \{\langle \vec{x}_1, c(\vec{x}_1)\rangle, \ldots, \langle \vec{x}_m, c(\vec{x}_m)\rangle\}$, чтобы из них можно с заранее выбранной вероятностью $\geqslant 1 - \delta$ получать правило $h : X_n \to \{0,1\}$, ошибающееся в не более чем заданной доле $\varepsilon > 0$ случаев (для любого $c : X_n \to \{0,1\}$);

3) придумать алгоритм $A(S) = h_S$ нахождения такого правила за ограниченное (полиномом от n, $\frac{1}{\varepsilon}$ и $-\log \delta = \log\left(\frac{1}{\delta}\right)$) число шагов;

4) реализовать предсказание тестовых примеров $\vec{x} \in X_n$ с помощью найденного правила h_S.

Продемонстрируем успех парадигмы ВПК-обучения на фольклорном примере обучения конъюнктивным понятиям.

Определение 1. Пусть каждый объект x описывается множеством бинарных признаков $f_1, \ldots, f_n$, т.е. множество объектов $X \subseteq \{0,1\}^n$, где n – число признаков. Чтобы узнать значение признака на объекте $\vec{x}$, введем булевские переменные $p_1, \ldots, p_n : \{0,1\}^n \to \{0,1\}$, соответствующие проецированию на компоненты. Переменные вместе с их отрицаниями назовем ***литералами****.*

Понятие *— подмножество объектов $\{\vec{x} \in X : c(\vec{x}) = 1\}$, задаваемое булевой функцией $c : X \to \{0,1\}$, называемой* ***индикатором****. Если $c(\vec{x}) = 1$, то объект $\vec{x} \in X$ называется положительным примером понятия, в противном случае — отрицательным примером. Если индикатор представим конъюнкцией литералов, то такое понятие назовем* ***конъюнктивным****.*

Гипотеза *— список литералов $h = \{l_{i_1}, \ldots, l_{i_k}\}$, конъюнкция которых представляет кандидата на обучаемое понятие. Предсказание с помощью гипотезы осуществляется по булевой функции $h = l_{i_1} \wedge \ldots \wedge l_{i_k}$ в качестве индикатора.*

Легко проверить, что, отождествляя объект $\vec{x} \in \{0,1\}^n$ с максимально непротиворечивым множеством литералов $\{l_1, \ldots, l_n\}$ ($l_j \in \{\neg f_j, f_j\}$), гипотеза h доопределяет пример $\vec{x}$ положительно, если и только если $h \subseteq \vec{x}$.

В задаче ВПК-обучения конъюнктивным понятиям предполагается, что искомое понятие c задается конъюнкцией $c = \{l'_{i_1}, \ldots, l'_{i_r}\}$ литералов.

Определение 2. Объем m выборки $S = \{\langle \vec{x}_1, c(\vec{x}_1)\rangle, \ldots, \langle \vec{x}_m, c(\vec{x}_m)\rangle\}$ назовем ***большим****, если $m \geqslant \frac{2n \cdot [\ln(2n) - \ln(\delta)]}{\varepsilon}$.*

Алгоритм 1 (пересечения).

1. Перечислим элементы $S = \{\langle \vec{x}_1, c(\vec{x}_1)\rangle, \ldots, \langle \vec{x}_m, c(\vec{x}_m)\rangle\}$ в некотором порядке.

2. Положим $h_0 = L = \{\neg p_1, p_1, \ldots, \neg p_n, p_n\}$ (множество всех литералов).

3. По порядку проверяем $h_j(\vec{x}_{j+1}) \neq c(\vec{x}_{j+1})$. Если ошибка не происходит, то гипотеза не изменяется. Когда случается ошибка, из гипотезы удаляются те литералы, которых нет в неправильно предсказанном примере:

$$h_j(\vec{x}_{j+1}) \neq c(\vec{x}_{j+1}) \Rightarrow h_{j+1} = h_j \setminus (L \setminus \vec{x}_{j+1}).$$

Легко проверить, что $h_j \setminus (L \setminus \vec{x}_{j+1}) = h_j \cap \vec{x}_{j+1}$ как множества литералов, что и дает название алгоритму 1.

Сформулируем несколько лемм, чтобы доказать, что выборка большого объема достаточна, чтобы с заранее выбранной вероятностью $\geqslant 1 - \delta$ алгоритм 1 находил правило $h : X_n \to \{0, 1\}$, ошибающееся в не более чем заданной доле $\varepsilon > 0$ случаев.

Лемма 1. Никакой литерал из обучаемого конъюнкта $c = \{l'_{i_1}, \ldots, l'_{i_k}\}$ не удаляется алгоритмом пересечения, т.е. $c \subseteq h_j$ для всех j.

Лемма 2. Алгоритм 1 может ошибаться только на положительных примерах, т.е. случай $h_j(\vec{x}_j) = 1 \neq 0 = c(\vec{x}_j)$ не возможен.

Определение 3. Литерал l назовем ***плохим****, если $\mathbb{P}[c(\vec{x}) = 1 \& l \notin \vec{x}] > > \frac{\varepsilon}{2n}$.*

Лемма 3. Если h не содержит плохих литералов, то $\mathbb{P}[h(\vec{x}) \neq c(\vec{x})] \leqslant \varepsilon$.

Лемма 4. Если $m \geqslant \frac{2n \cdot [\ln(2n) - \ln(\delta)]}{\varepsilon}$, то $2n \cdot \left(1 - \frac{\varepsilon}{2n}\right)^m \leqslant \delta$.

Теорема 1. Алгоритм пересечения доказывает эффективную ВПК-обучаемость конъюнктивным понятиям.

Доказательство. Докажем, что при большом объеме выборки m вероятность того, что в гипотезе h останется хоть один плохой литерал, не превзойдет δ.

Пусть l – плохой литерал. Тогда $\mathbb{P}[l$ сохранится после одного примера$] = = 1 - \mathbb{P}[l$ удалится первым примером $] \leqslant 1 - \frac{\varepsilon}{2n}$.

Поэтому $\mathbb{P}[l$ сохранится после m примеров$] \leqslant \left(1 - \frac{\varepsilon}{2n}\right)^m$.

Так как всего литералов ровно $2n$, то утверждение теоремы следует из неравенства Буля $\mathbb{P}\left[\cup_{j=1}^k A_j\right] \leqslant \sum\limits_{j=1}^k \mathbb{P}[A_j]$ и леммы 4.

Ясно, что $\frac{2n \cdot [\ln(2n) - \ln(\delta)]}{\varepsilon}$ мажорируется полиномом от n, $\frac{1}{\varepsilon}$ и $-\log \delta$, и число шагов алгоритма пересечения для обработки одного примера линейно зависит от n.

3. Проблемы извлечения знаний с помощью операции сходства

Если изучаемое понятие $c : \{0,1\}^n \to \{0,1\}$ не является конъюнктивным, то алгоритм 1 не работает, так как выдает единственный (чаще всего пустой) конъюнкт. Более широким классом булевых функций, для которого возможно эффективное ВПК-обучение, являются списки решений [6]. Для общего случая дизъюнктивных нормальных формул вопрос остается открытым. Если ограничиться конъюнкцией хорновских дизъюнктов, то в [7, 8] предложены алгоритмы ВПК-обучения с использованием эмуляции запросов о принадлежности и эквивалентности с помощью случайных оценок.

Хотя индуктивные методы извлечения знаний восходят к трудам английского логика XIX в. Д.С. Милля [9], интерес к ним возродился с возникновением искусственного интеллекта на Западе в 1950-е гг. Некоторые эксперименты с индуктивным образованием понятий в логике высказываний описаны в книге [10], изданной в 1966 г. Важным был доклад С.А. Вере [11] на Четвертой Объединенной международной конференции по ИИ, проходившей в Тбилиси в 1975 г., где операция сходства в логике предикатов была реализована через унификацию.

В пионерских работах В.К. Финна [1, 12] был предложен метод извлечения знаний из обучающих примеров, описываемых бинарными признаками, с учетом контрпримеров, описываемых таким же образом. При этом допускалась множественность целевых свойств (тоже как набор бинарных переменных). В.К. Финн рассматривал симметричную ситуацию: нахождение сходств как положительных примеров, не допускающих включение ни в какой отрицательный контрпример, так и отрицательных примеров с запретом положительных контрпримеров. Это привело к красивой теории логики аргументации [13, 14], но дополнительно увеличило вычислительную сложность. Но более существенным недостатком первоначального подхода было ограничение булевой алгеброй как носителя универсума примеров и их сходств. Это привело к проблемам формализации методов различия [15] и остатков [16] Д.С. Милля, так как они использовали операцию относительного дополнения. Для булевых алгебр она определяется однозначно, что не так для недистрибутивных решеток.

Ранее в теории решеток [3] возникло новое направление — анализ формальных понятий (FCA) [2]. Уже в 1982 г. Рудольф Вилле доказал свою Фундаментальную теорему, которая позволила эффективно работать с общими конечными решетками.

Следующие шаги в развитии отечественного подхода к извлечению знаний с помощью бинарной операции сходства были сделаны в трудах С.О. Кузнецова. Сначала им был предложен алгоритм [17] “Замыкай-по-одному” для эффективного построения остовного дерева в решетке всех кандидатов.

Затем С.О. Кузнецов [18, 19] показал, как для извлечения знаний из обучающих примеров, описываемых бинарными признаками, использовать тех-

Таблица 1. Минимальная выборка для булевой алгебры

I	f_1	f_2	$\ldots$	f_n
o_1	0	1	$\ldots$	1
o_2	1	0	$\ldots$	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
o_n	1	1	$\ldots$	0

нику FCA, расширяя теорию с булевых алгебр обучающих примеров на произвольные конечные (полу-)решетки.

Более того, Фундаментальная теорема Р. Вилле ответила на вопрос о достаточности описания примеров бинарными признаками: любая конечная решетка может быть с точностью до изоморфизма порождена как решетка кандидатов некоторой обучающей выборки из примеров, описываемых битовыми строками.

С помощью техники FCA Д.В. Виноградову [20] удалось создать алгоритм и доказать его корректность для представления объектов, описываемых признаками, чьи значения образуют произвольную конечную нижнюю полурешетку, с помощью битовых строк.

Собирая вместе битовые строки положительных (обучающих) примеров O, получаем "обучающую выборку" (или, в терминологии FCA, формальный контекст) $I \subseteq O \times F$, где $F = \{f_1, \ldots, f_n\}$ – множество бинарных признаков.

Определение 4. Для подмножества $A \subseteq O$ объектов его общим ***фрагментом*** *называется подмножество признаков $A' = \{f \in F : \forall o \in A [oIf]\} \subseteq F$. Ясно, что $\emptyset' = F$. A' вычисляется побитовым умножением строк, соответствующих отобранным во множество A объектам.*

Для подмножества $B \subseteq F$ признаков его ***родителями*** *называется подмножество объектов $B' = \{o \in O : \forall f \in B\, [oIf]\} \subseteq O$. Очевидно, что $\emptyset' = O$. B' можно вычислить побитовым умножением столбцов, соответствующих отобранным во множество B признакам.*

Пару $\langle A, B \rangle$ назовем ***кандидатом****, если $A = B' \subseteq O$ и $B = A' \subseteq F$.*

В FCA [2] кандидаты называются "формальными понятиями", но предпочтем сменить имя, так как "кандидаты" из нашего определения играют эту роль в ДСМ-методе [1], где они превращаются в "гипотезы о причинах" после фальсификации некоторых из них дополнительными проверками.

Случаю булевой алгебры соответствует выборка (см. таблицу 1).

Здесь множество обучающих примеров $O = \{o_1, o_2, \ldots, o_n\}$ – коатомы, где $o_i I f_j$ (т.е. пример o_i имеет признак f_j), если и только если $i \neq j$.

Ясно, что любая пара $\langle O \setminus \{o_{j_1}, \ldots, o_{j_k}\}, \{f_{j_1}, \ldots, f_{j_k}\} \rangle$ будет кандидатом, поэтому имеем булеву алгебру всех 2^n подмножеств признаков (= битовых строк).

При $n = 32$ обучающая выборка занимает $n \cdot n = 2^{10}$ бит = 128 байт. Но чтобы записать результат, требуется $n \cdot 2^n = 2^{37}$ бит, т.е. ровно 16 Гбайт!

Этот пример указывает на вычислительную трудность реализации первоначальной идеи В.К. Финна о порождении всех кандидатов, а затем фальсификации тех из них, которые вкладываются (как битовые строки) в отрицательные примеры (так называемый "запрет контрпримеров").

Но проблема не ограничивается сложностью вычислений по памяти. Д.В. Виноградов [21] обнаружил эффект "переобучения" — возникновение "фантомных" кандидатов, не устранимых сколь угодно большим количеством контрпримеров.

Мы предполагаем, что имеется по крайней мере 2 различных набора бинарных признаков (= 2 причины), включение любого из которых в описание объекта делает его положительным. Допустим, имеется n не входящих ни в одну из этих причин признаков, называемых **сопутствующими**.

Рассмотрим следующую вероятностную модель:

1) в обучающей выборке имеется четыре обучающих примера, два из которых в точности совпадают с различными причинами;

2) остальные два обучающих примера содержат по одной из причин, а сопутствующие бинарные признаки образуют последовательности Бернулли с вероятностью успеха p;

3) имеется m контрпримеров, каждый из которых не имеет ни одной из причин, а сопутствующие признаки образуют последовательности Бернулли с вероятностью успеха p, все случайные биты предполагаются независимыми.

Нетривиальное сходство второй пары обучающих примеров назовем **фантомным**. Заметим, что возникновение такого сходства мешает правильному предсказанию целевого свойства у тестовых примеров, так как его включение объявит этот пример истинным, хотя он может не содержать ни одной из исходных (правильных) причин. Это сходство возникает из-за попытки извлечь максимум информации из обучающей выборки, но объясняется случайным совпадением нескольких сопутствующих признаков у обучающих примеров второй пары, каждый из которых имеет свою причину, отличную от таковой у другого примера.

Довольно легко доказать лемму 5.

Лемма 5. С вероятностью $\sum\limits_{j=0}^{m} \binom{m}{j} \cdot (-1)^j \cdot (1 - p^2 + p^{2+j})^n$ *возникнет фантомное сходство, которое не устранится ни одним из* m *случайных контрпримеров.*

С помощью леммы 5 доказывается теорема 2.

Теорема 2 [21]. *При числе сопутствующих признаков* $n \to \infty$ *и вероятности появления этих признаков у контрпримеров и обучающих примеров, равной* $p = \sqrt{\frac{a}{n}}$ $(a \leqslant 1)$, *вероятность возникновения фантомного сходства двух обучающих примеров, не устраненного никаким из* $m = c \cdot \sqrt{n}$ *контрпримеров, будет стремиться к*

$$1 - e^{-a} - a \cdot e^{-a} \cdot \left[1 - e^{-c \cdot \sqrt{a}}\right] > 0.$$

Легко проверить, что вероятность $p \geqslant \sqrt{\frac{a}{n}}$ гарантирует, что возникнет нетривиальное фантомное сходство, т.е. фрагмент, который содержит хотя бы один сопутствующий признак; при $p < \sqrt{\frac{a}{n}}$ гарантии нет даже безотносительно наличия контрпримеров.

Левая часть последнего неравенства теоремы при $c \to \infty$ уменьшится до $1 - e^{-a} - a \cdot e^{-a}$, что совпадает с вероятностью того, что пуассоновская случайная величина со средним a примет значение больше единицы, что не может стать равным нулю. Это значит, что даже для сколь угодно большого числа контрпримеров вероятность возникновения фантомного сходства не может быть обнулена.

Другими словами, этот результат указывает на возможность некорректного предсказания тестовых примеров, т.е. приводит к аналогу ВПК-обучения. А в этом случае можно не ограничиваться детерминированными алгоритмами.

Л.А. Якимова [22] смогла экспериментально обнаружить фантомные гипотезы при работе алгоритма исчерпывающего порождения гипотез (ДСМ-метода) на массиве Mushrooms из репозитория данных для тестирования алгоритмов машинного обучения Университета Калифорнии в г. Ирвайн [23]. Особенно настораживающим был тот факт, что фантомные гипотезы смогли заставить ДСМ-систему неправильно объявить несколько ядовитых грибов съедобными.

4. Вероятностные алгоритмы поиска сходств

Отказавшись от детерминизма, переходим к идее порождения достаточно большого числа кандидатов (в гипотезы о причинах), каждая своей траекторией случайного блуждания по решетке кандидатов.

Базовые шаги этих блужданий и их корректность устанавливаются в следующей легко доказываемой лемме-определении.

Лемма 6. Операция ***замыкай-по-одному-вниз*** *на кандидате* $\langle A, B\rangle$ *и объекте* $o \in O$ *порождает кандидат*

$$CbODown(\langle A, B\rangle, o) = \big\langle (A \cup \{o\})'', B \cap \{o\}' \big\rangle.$$

Операция ***замыкай-по-одному-вверх*** *на кандидате* $\langle A, B\rangle$ *и признаке* $f \in F$ *порождает кандидат*

$$CbOUp(\langle A, B\rangle, f) = \big\langle A \cap \{f\}', (B \cup \{f\})'' \big\rangle.$$

Возможно небольшое ускорение вычислений, если заметить, что $o \in A \Rightarrow$ $\Rightarrow CbODown(\langle A, B\rangle, o) = \langle A, B\rangle$ и $f \in B \Rightarrow CbODown(\langle A, B\rangle, f) = \langle A, B\rangle$, так как проверка принадлежности значительно быстрее, чем вычисление операций *CbO*.

В случае булевой алгебры все еще сильнее ускоряется: если $o_j \notin A$, то $CbODown(\langle A, B\rangle, o_j) = \langle A \cup \{o_j\}, B \setminus \{f_j\}\rangle$, и $CbODown(\langle A, B\rangle, o_j) = \langle A, B\rangle$ в противном случае. Аналогично если $f_j \notin B$, то $CbOUp(\langle A, B\rangle, f_j) =$ $= \langle A \setminus \{o_j\}, B \cup \{f_j\}\rangle$, и, как обычно, $CbOUp(\langle A, B\rangle, f_j) = \langle A, B\rangle$ при $f_j \in B$.

Простейшее случайное блуждание порождается алгоритмом 2.

Алгоритм 2 (немонотонный).

1. Сформируем обучающую выборку $I \subseteq O \times F$, где $O := (+)$-примеры, $F :=$ признаки.

2. Начинаем с наименьшего кандидата: положим $A := O$; $B := O'$.

3. До некоторого шага T делаем цикл из пунктов 4 и 5.

4. Формируем множество свободных примеров и признаков: $R := (O \setminus A) \cup \cup (F \setminus B)$.

5. Выбираем случайный элемент $r \in R$.

Если выбран объект $r \in O \setminus A$, то применяем операцию CbODown: $\langle A, B\rangle := CbODown(\langle A, B\rangle, r)$.

Если выбран признак $r \in F \setminus B$, то применяем операцию CbOUp: $\langle A, B\rangle := := CbOUp(\langle A, B\rangle, r)$.

6. После цикла выдается результат $\langle A, B\rangle$ – кандидат, на котором алгоритм 2 остановился на шаге T.

Очевидно, что в случае булевой алгебры алгоритм 2 с равной $\frac{1}{n}$ вероятностью переходит с текущего подмножества на одного из n его соседей, т.е. представляет собой случайное блуждание по соответствующему гиперкубу.

Алгоритм 3 (монотонный).

1. Сформируем обучающую выборку $I \subseteq O \times F$, где $O := (+)$-примеры, $F :=$ признаки.

Формируем дизъюнктное объединение множеств примеров и признаков: $R := O \cup F$.

2. Начинаем с наименьшего кандидата: положим $A := O$; $B := O'$.

3. До некоторого шага T делаем цикл.

4. Выбираем случайный элемент $r \in R$.

Если выбран объект $r \in O$, то применяем операцию CbODown: $\langle A, B\rangle := := CbODown(\langle A, B\rangle, r)$.

Если выбран признак $r \in F$, то применяем операцию CbOUp: $\langle A, B\rangle := := CbOUp(\langle A, B\rangle, r)$.

5. После цикла выдается результат $\langle A, B\rangle$ – кандидат, на котором алгоритм 3 остановился на шаге T.

В случае булевой алгебры алгоритм 3 с вероятностью $\frac{1}{2}$ никуда не сдвигается, а с равной $\frac{1}{2 \cdot n}$ вероятностью переходит с текущего подмножества на одного из n его соседей, т.е. представляет собой ленивое случайное блуждание по соответствующему гиперкубу.

Заметим, что корректность алгоритмов 2 и 3 (т.е. то, что в результате их работы обязательно получим кандидата) следует из леммы 6.

Основная проблема с алгоритмами 2 и 3, применяемыми к произвольным обучающим выборкам, состоит в том, что неизвестна никакая полиномиальная оценка на "время остановки" T, которая обеспечивала бы хорошую "пе-

ремешиваемость" соответствующей цепи Маркова. Таким образом, вопрос о выборе параметра T (фактически, о длине цикла) в этих алгоритмах остается открытым.

По этой причине и после экспериментов с алгоритмами 2 и 3 [24] было решено отказаться от них в пользу семейства спаривающих цепей Маркова [25]. Простейший вариант задается алгоритмом 4.

Алгоритм 4 (спаривающий).

1. Сформируем обучающую выборку $I \subseteq O \times F$, где $O :=$ (+)-примеры, $F :=$ признаки.

2. Положим $R := O \cup F$; $\mathrm{Min} := \langle O, O' \rangle$ – наименьший кандидат; $\mathrm{Max} :=$ $:= \langle F', F \rangle$ – наибольший кандидат.

3. Пока $\mathrm{Min} \neq \mathrm{Max}$, т.е. оба кандидата не станут равны, делать цикл.

4. Выбираем случайный элемент $r \in R$.

Если выбран объект $r \in O$, то одновременно вычисляются

$$\mathrm{Min} := CbODown(\mathrm{Min}, r); \quad \mathrm{Max} := CbODown(\mathrm{Max}, r).$$

Если выбран признак $r \in F$, то одновременно вычисляются

$$\mathrm{Min} := CbOUp(\mathrm{Min}, r); \quad \mathrm{Max} := CbOUp(\mathrm{Max}, r).$$

5. После цикла выдается результат – любой (из совпадающих) кандидатов, например, Min.

Для доказательства останавливаемости алгоритма 4 с вероятностью единица, нам нужно определение 5.

Определение 5. ***Порядок*** *на кандидатах:* $\langle A_1, B_1 \rangle \leqslant \langle A_2, B_2 \rangle$, *если* $B_1 \subseteq B_2$.

Порядок, задаваемый этим определением, двойственен порядку, рассматриваемому в FCA [2]: $\langle A_1, B_1 \rangle \leqslant \langle A_2, B_2 \rangle$, если $A_1 \subseteq A_2$.

Теперь легко доказать лемму 7.

Лемма 7. Для всякой упорядоченной пары кандидатов $\langle A_1, B_1 \rangle \leqslant \langle A_2, B_2 \rangle$ *и любого* $o \in O$ *имеем* $CbODown(\langle A_1, B_1 \rangle, o) \leqslant CbODown(\langle A_2, B_2 \rangle, o)$.

Для всякой упорядоченной пары кандидатов $\langle A_1, B_1 \rangle \leqslant \langle A_2, B_2 \rangle$ *и любого* $f \in F$ *имеем* $CbOUp(\langle A_1, B_1 \rangle, f) \leqslant CbOUp(\langle A_2, B_2 \rangle, f)$.

С помощью леммы 7 легко доказать теорему 3.

Теорема 3 [25]. *Алгоритм 4 соответствует цепи Маркова.*

Эргодические состояния спаривающей цепи Маркова имеют вид $\langle A, B \rangle =$ $= \langle A, B \rangle$ (соответствуют совпадающим парам кандидатов). Невозвратные состояния имеют вид $\langle A_1, B_1 \rangle < \langle A_2, B_2 \rangle$ (склеивание кандидатов еще не произошло).

Теперь имеем результат о конечности траекторий с вероятностью единица как следствие классической теоремы о невозвратных состояниях [26].

Следствие 1. Вероятность того, что состояние $\langle A_1(t), B_1(t) \rangle \leqslant$ $\leqslant \langle A_2(t), B_2(t) \rangle$ *спаривающей цепи Маркова окажется невозвратным, стре-*

мится к нулю, когда $t \to \infty$. Следовательно, алгоритм 4 остановится с вероятностью единица.

Хотя вопрос о среднем времени работы алгоритма 4 остается открытым, Д.В. Виноградовым получены теорема о среднем времени остановки этого алгоритма и о сильной концентрации около этого среднего для частного случая булевой алгебры [27]. Здесь на вход алгоритма 4 подается выборка, определенная на странице 5, но мы имеем вероятностный алгоритм, следующий шаг которого определяется датчиком случайных чисел (для выбора объекта или признака для замыкания).

Теорема 4. Среднее время работы алгоритма 4 для n-мерной булевой алгебры равно

$$\mathbb{E}T = \sum_{j=1}^{n} \frac{n}{j} \approx n \cdot \ln(n) + n \cdot \gamma + \frac{1}{2}.$$

$\mathbb{P}\left[T \geqslant (1+\varepsilon) \cdot n \cdot \ln(n)\right] \to 0$ *при* $n \to \infty$ *для любого* $\varepsilon > 0$.

Эта теорема позволяет понять, насколько эффективно работает алгоритм спаривающей цепи Маркова. В случае 32-мерной булевой алгебры количество ее элементов составляет 4 294 967 296. Одна траектория авторского алгоритма спаренного случайного блуждания имеет среднюю длину $\mathbb{E}T \leqslant 130$. Из-за сильной концентрации все траектории имеют примерно ту же самую длину. При каждом шаге порождается (не всегда, так как возможны возвращения или неподвижность одного или обоих кандидатов из пары) не более двух кандидатов (один снизу и один сверху). Таким образом, при порождении, например, 1000 случайных кандидатов с помощью 1000 траекторий суммарно будет порождено около 260 000 элементов булевой алгебры.

Эксперименты Л.А. Якимовой [22] показали, что этот эффект (быстрая остановка и, как следствие, порождение малого числа кандидатов) наблюдается и в случае общих обучающих выборок.

В качестве практического средства для устранения наиболее длинных траекторий возможно применение следующей техники остановки алгоритма 4 (или его ленивого варианта в алгоритме 5 далее) и запуска его заново.

Определение 6. Если $T_1, \ldots, T_r$ – независимые целочисленные случайные величины, имеющие распределение времени склеивания T, то ***верхняя граница склеивания*** *по r испытаниям определяется как $\hat{T} = T_1 + \ldots + T_r$.*

На практике предлагается сделать r прогонов спаривающей цепи Маркова и взять оценку $t_1 + \ldots + t_r$ верхней границы склеивания.

Оценим, как изменяются вероятности попадания в эргодические состояния при остановке спаривающей цепи Маркова по r прогонам.

Определение 7. Для целочисленной случайной величины $\hat{T}$, независимой от целочисленной случайной величины T, ***условное распределение*** *со-*

стояний относительно события $B = \{T \leqslant \hat{T}\}$ *есть распределение*

$$\nu_i = \mu_{\hat{T},i} = \frac{\mathbb{P}[X_T = i, T \leqslant \hat{T}]}{\mathbb{P}[T \leqslant \hat{T}]}$$

для любого эргодического состояния i.

Используем расстояние тотального изменения (вариации) в качестве метрики между распределениями вероятности на конечном множестве.

Определение 8. Расстояние ***тотального изменения*** *между распределениями вероятностей* $\mu = (\mu_i)_{i \in U}$ *и* $\nu = (\nu_i)_{i \in U}$ *на конечном пространстве* U *определяется формулой:* $\|\mu - \nu\|_{TV} = \frac{1}{2} \cdot \sum\limits_{i \in U} |\mu_i - \nu_i|$.

Приведем ключевые леммы, необходимые для доказательства ключевой теоремы.

Лемма 8. $\|\mu - \nu\|_{TV} = \max_{R \subseteq U} |\mu(R) - \nu(R)|$.

Здесь подмножество R, на котором достигается максимум, можно задать, например, формулой: $R = \{i \in U \mid \mu_i > \nu_i\}$.

Лемма 9. $\|\mu - \mu_{\hat{T}}\|_{TV} \leqslant \frac{\mathbb{P}[T > \hat{T}]}{1 - \mathbb{P}[T > \hat{T}]}$ *, где* $\mu_{\hat{T}}$ *– распределение остановленной на верхней границе* $\hat{T}$ *склеивания по* $r > 1$ *испытаниям, а* μ *– распределение выдачи неостановленной цепи.*

Лемма 10. $\|\mu - \mu_{\hat{T}}\|_{TV} \leqslant \frac{1}{2^r - 1}$ *, где* $\mu_{\hat{T}}$ *– распределение остановленной на верхней границе склеивания по* $r > 1$ *испытаниям, а* μ *– распределение выдачи неостановленной цепи.*

Соединяя результаты лемм 8 и 10, получим теорему 5.

Теорема 5 [25]. *Для любого подмножества* R *решетки кандидатов если вероятность алгоритма 4 выбора какого-то элемента* R *равна* $\mu(R) = \sum\limits_{j \in R} \mathbb{P}[X_T = i]$, *то вероятность* $\mu_{\hat{T}}(R)$ *выбора какого-то элемента этого подмножества остановленным вариантом алгоритма 4 ограничена снизу* $\mu_{\hat{T}}(R) \geqslant \mu(R) - \frac{1}{2^r - 1}$, *если остановка совершается по верхней границе* $\hat{T}$ *склеивания для* $r > 1$ *предварительных траекторий.*

Впрочем, этот результат может не иметь практического смысла. Как показали эксперименты Л.А. Якимовой [22], обычно длинные траектории так редки (хотя и наблюдались), что проверка критерия остановки в цикле приводит к замедлению вычислений, не компенсируемым отбрасыванием длинных траекторий. Этот эмпирический факт находится в хорошем согласии со свойствами малой средней длины и сильной концентрации длин траекторий около среднего, которые для случая булевой алгебры были доказаны в теореме 4 выше.

Но бесспорным стало преимущество использования ленивого варианта спаривающей цепи Маркова. Чтобы понять, откуда здесь возникают возможности ускорения вычислений, необходимо подробнее рассмотреть базовые шаги случайных блужданий и их реализацию на современных компьютерах.

Фундаментальная теорема Р. Вилле оказывается полезной не только с идеологической (достаточно ограничиться описанием обучающих объектов и их сходств битовыми строками, т.е. наша постановка задачи максимально общая), но и с вычислительной точки зрения. Дело в том, что сходство двух битовых строк реализуется побитовым умножением, а эта операция на современных компьютерах реализована максимально эффективно. На CPU она занимает один такт (если битовая строка влезает в регистр), а их около 5 млрд. в секунду. На GPGPU нужны четыре волны (четыре такта с частотой более 1,5 ГГц), но зато современные видеокарты могут в параллель обрабатывать до (а сейчас уже и более) 4096 потоков. При этом алгоритмы, основанные на цепях Маркова, допускают максимальное распараллеливание: траектория определяется только датчиками случайных чисел.

Более того, современные языки программирования имеют специальные структуры данных (например, C++ использует boost::dynamic_bitset для работы с битовыми строками, причем даже переменной длины). Современные компиляторы умеют дополнительно оптимизировать побитовое умножение коротких строк, размещая несколько штук в длинные регистры и выполняя за один такт сразу несколько умножений (так называемый векторный параллелизм).

В базовом шаге $CbODown(\langle A, B\rangle, o) = \langle (A \cup \{o\})'', B \cap \{o\}'\rangle$ вторая компонента (вычисление фрагмента) соответствует побитовому умножению фрагмента старого кандидата B и описания $\{o\}'$ дополнительного объекта-родителя, т.е. вычисляется очень быстро. Первая же компонента $(A \cup \{o\})''$, так называемое "замыкание", требует нахождения всех родителей у соответствующего фрагмента. Хорошо еще, что $(A \cup \{o\})'' = (B \cap \{o\}')'$, т.е. можно использовать побитовое умножение столбцов из матрицы обучающей выборки. Но даже в этом случае количество перемножаемых столбцов может быть велико.

Аналогичное замечание верно и для $CbOUp$, только там первая и вторая компоненты меняются местами относительно сложности их вычисления.

Ключевым наблюдением будет то, что объекты и признаки выпадают сериями, а внутри серии нужно вычислять только быстрые операции. Когда серия прерывается, нужно один раз сделать вычислительно сложную операцию замыкания (для $CbOUp$ используется равенство $(B \cup \{f\})'' = (A \cap \{f\}')'$).

Собирая все нужные нам идеи, получаем алгоритм 5.

Алгоритм 5 (ленивый).

1. Сформируем обучающую выборку $I \subseteq O \times F$, где $O :=$ (+)-примеры, $F :=$ признаки.

2. Положим $R := O \cup F$; $A_1 := O$, $B_1 := O'$ – наименьший кандидат; $A_2 := F'$, $B_2 := F$ – наибольший кандидат.

Выбираем направление движения $moveUp := true$.

3. Пока $\mathrm{Min} \neq \mathrm{Max}$, т.е. оба кандидата не станут равны, делать цикл.

4. Выбираем случайный элемент $r \in R$.

5. Если выбран объект $r \in O$, то проверяем истинность $moveUp$.

Если это так, то сначала применяем замыкание $B_1 := A'_1$; $B_2 := A'_2$, и переопределяем $moveUp := false$.

Затем (независимо от $moveUp$) одновременно вычисляются сходства $B_1 := := B_1 \cap \{r\}'$; $B_1 := B_1 \cap \{r\}'$.

6. Если выбран признак $r \in F$, то проверяем истинность $moveUp$.

Если это не так, то сначала применяем замыкание $A_1 := B'_1$; $A_2 := B'_2$, и переопределяем $moveUp := true$;

Затем (независимо от $moveUp$) одновременно вычисляются сходства $A_1 := := A_1 \cap \{r\}'$; $A_1 := A_1 \cap \{r\}'$;

7. После цикла выдается результат — любой (из совпадающих) кандидатов, например, Min.

Возникает вопрос о степени экономии, достигаемой такой процедурой. Рассмотрим последовательность типов (объект или признак) элементов, выбираемых в ходе работы алгоритма 5. Ясно, что это – последовательность (вообще говоря, бесконечная) испытаний Бернулли $\langle \sigma_1, \ldots, \sigma_j, \ldots \rangle$ с вероятностью успеха (например, выбора признака), равной $p = \frac{n}{n+k}$, где n – число признаков, а k – число обучающих примеров.

Прежде всего зафиксируем два события $\{\sigma_1 = 0\}$ и $\{\sigma_1 = 1\}$.

В случае $\sigma_1 = 0$ нас интересует длина события $\{\sigma_1 = \ldots = \sigma_i = 0, \sigma_{i+1} = = \ldots = \sigma_j = 1, \sigma_{j+1} = 0\}$, а в случае $\sigma_1 = 1$ интересна длина $\{\sigma_1 = \ldots = \sigma_i = = 1, \sigma_{i+1} = \ldots = \sigma_j = 0, \sigma_{j+1} = 1\}$.

Рассмотрим случайную величину T суммы длин двух переходов от объектов к признакам и снова к объектам (при $\sigma = 0$) и суммы длин двух переходов от признаков к объектам (при $\sigma = 1$).

Теорема 6 [25]. *В ленивой схеме вычислений на каждую пару применений операции замыкания, одной для CbOUp и одной для CbODown, в среднем в классической схеме мы будем делать* $\mathbb{E}T = \frac{(n+k)^2}{k \cdot n}$ *операций замыкания.*

Ясно, что выигрыш тем больше, чем больше разница между k и n, где k – число обучающих примеров, а n – число признаков, используемых для описания объектов, так как $\frac{(n+k)^2}{k \cdot n} = 4 + \frac{(n-k)^2}{k \cdot n}$. Даже в худшем случае $k = n$ это сокращение вызовов трудоемкой операции не меньше двух раз, потому что $\frac{(2k)^2}{k \cdot k} = 4$.

Л.А. Якимова в ее выпускной квалификационной работе бакалавра [28], выполненной под руководством Д.В. Виноградова, продемонстрировала, что выигрыш от такого перехода на реальных данных может достигать очень большой величины, близкой к теоретическому предсказанию.

Сравнение алгоритмов 4 и 5 проводилось на массиве Mushrooms из репозитория данных для тестирования алгоритмов машинного обучения Университета Калифорнии в г. Ирвайн [23]. В этом массиве содержится описание 8124 грибов, из которых $k = 4208$ являются съедобными, а 3916 ядовитыми. Грибы кодировались битовыми строками длины $n = 124$ бита.

По теореме 6 средний выигрыш от применения ленивой схемы вычислений достигает на операциях замыкания $\frac{1}{2} \cdot \frac{(k+n)^2}{k \cdot n} \approx 18$ раз.

Алгоритмы спаривающих цепей Маркова в стандартном (алгоритм 4) и ленивом вариантах (алгоритм 5) сравнивались по времени вычисления заданного числа кандидатов. Соотношение этих промежутков времени (чуть более 17 раз) оказалось замечательно согласованным с теоретическим результатом, вычисленным выше. То, что выигрыш оказался несколько меньше, объясняется тем, что, кроме операций замыкания, в этих алгоритмах имеются еще операции побитового умножения, которые хотя и очень быстрые, тем не менее остаются в неизменном количестве. За подробностями читатель отсылается к [28].

5. Алгебраическое машинное обучение

Будем использовать классическую схему машинного обучения: сначала разделение выборки на обучающую и тестовую, потом извлечение знаний (индуктивное порождение гипотез), наконец, тестирование качества с помощью предсказания целевого свойства у тестовых примеров.

В классическом ДСМ-методе [1] имеется дополнительная процедура — абдуктивное принятие гипотез, но в нашем вероятностном случае она в значительной степени теряет свой смысл, так как некоторые гипотезы могут не возникнуть. Более того, ВПК-парадигма имеет свой способ обеспечения надежности результата, который в нашем случае будет составлять утверждение теоремы 7.

Индуктивное обобщение обучающих примеров осуществляется по алгоритму 6.

Алгоритм 6 (Индукция).

1. Зафиксируем число N порождаемых гипотез (например, из теоремы 7).

2. Сформируем обучающую выборку $I \subseteq O \times F$, где $O :=$ (+)-примеры, $F :=$ признаки. Сформируем список контрпримеров $C :=$ (–)-примеры.

3. Создадим несколько нитей для порождения гипотез, применяя алгоритм 5.

4. Пока число гипотез не достигнет заданного числа N, вычислять в различных нитях (threads) вычислений п. 5.

5. Если в нити возник кандидат $\langle A, B \rangle$, проверять наличие контрпримера: Если для некоторого $c \in C$ выполняется $B \subseteq c'$, то запустить заново алгоритм 5. Если контрпримеров не нашлось, то выдать $\langle A, B \rangle$ в качестве еще одной гипотезы.

После индуктивного обобщения обучающих примеров можно использовать тестовые примеры для проверки качества порожденных гипотез.

Алгоритм 7 (Аналогия).

1. Сформируем список тестовых примеров $X :=$ (τ)-примеры.

2. Во внешнем цикле выбираем текущий тестовый пример $o \in X$. Временно объявляем его отрицательным примером.

3. Во внутреннем цикле проверяем вложение фрагмента B гипотезы $\langle A, B\rangle$, порожденной алгоритмом 6, в описание тестового примера $B \subseteq o'$. Если $B \subseteq o'$, объявить пример o положительным; внутренний цикл прервать.

4. Если вложения не нашлось, то объект o останется отрицательным.

Алгоритм 7 называется Аналогией потому, что объявленный положительным тестовый пример o содержит в своем описании фрагмент B гипотезы $\langle A, B\rangle$, который также содержится в описаниях всех обучающих примеров из списка родителей $A \subseteq O$. Если верно, что была выявлена причина целевого свойства как сходство всех обучающих примеров из A, то и тестовый пример доопределен правильно по аналогии с ними.

Определение 9. ***Нижнее полупространство*** $H^{\downarrow}_{1/2}(o)$, *определяемое объектом* o *с фрагментом* $o' \subseteq F$, *задается линейным неравенством* $x_{j_1} + \ldots + x_{j_k} < \frac{1}{2}$, *где* $F \setminus o' = \{f_{j_1}, \ldots, f_{j_k}\}$.

Здесь переменные $x_1, \ldots, x_n \in \mathbb{R}$ находятся во взаимно-однозначном соответствии с признаками $F = \{f_1, \ldots, f_n\}$.

Так как обучающие и тестовые примеры и фрагменты лежат в $\{0,1\}^n$, то можно ограничиться булевым кубом.

Лемма 11. Пример o *предсказывается положительным с помощью гипотезы* $\langle A, B\rangle$ $(B \subseteq o')$ *тогда и только тогда, когда в его нижнем полупространстве* $H^{\downarrow}_{1/2}(o)$ *содержится точка, соответствующая* B.

Лемма 11 фактически переворачивает ВПК-парадигму: если в исходной постановке Вапника–Червоненкиса гипотезы были фиксированы и соответствовали подмножествам пространства обучающих примеров, а примеры выбирались независимо и случайно (что может быть невыполнимо на практике), то у нас тестовые примеры соответствуют нижним полупространствам булева куба, а фрагменты случайно и независимо порождаемых гипотез — точкам в этих полупространствах.

Зафиксируем $\varepsilon > 0$ – точность предсказания.

Определение 10. Объект o *назовем* ε-***важным***, *если суммарная вероятность появления таких гипотез* $\langle A, B\rangle$, *что* $B \in H^{\downarrow}_{1/2}(o)$, *будет больше* ε.

Теорема 7. Для n *признаков и любых* $\varepsilon > 0$ *и* $1 > \delta > 0$ *достаточно породить*

$$N \geqslant \frac{n \cdot \ln 2 - \ln \delta}{\varepsilon}$$

гипотез, чтобы с вероятностью, большей $1 - \delta$, *все* ε*-важные объекты были предсказаны положительно.*

Доказательство. Докажем, что при $N \geqslant \frac{n \cdot \ln 2 - \ln \delta}{\varepsilon}$ гипотезах вероятность того, что хотя бы один ε-важный пример будет пропущен, не превзойдет δ.

Пусть o будет ε-важным примером, а $H^{\downarrow}_{1/2}(o)$ — соответствующим нижним полупространством. Тогда $\mathbb{P}[o$ не доопределится одной гипотезой$] = 1 - \mathbb{P}[o$ доопределится первой гипотезой$] \leqslant 1 - \varepsilon$.

Поэтому $\mathbb{P}[o$ не доопределится N гипотезами$] \leqslant (1-\varepsilon)^N$ из-за независимости гипотез.

Так как различных нижних полупространств ровно 2^n, то по неравенству Буля $\mathbb{P}[$хотя бы один ε-важный o не доопределится N гипотезами$] \leqslant 2^n \cdot (1-\varepsilon)^N$.

Но $2^n \cdot (1-\varepsilon)^N \leqslant 2^n \cdot e^{-N\cdot\varepsilon} \leqslant 2^n \cdot e^{\ln\delta - n\cdot\ln 2} = \delta$, что завершает доказательство.

В публикации [25] использовалось прямое сведение к технике Вапника–Червоненкиса [5] "метод повторной выборки", что дало более грубую оценку. К тому же в ее формулировке там имеется опечатка.

Значение этой теоремы — оценка на число требуемых гипотез в алгоритме 6. В принципе, эту оценку можно немного улучшить, если заметить, что можно рассматривать только минимальные по включению примеры: они образуют антицепь в булевой алгебре, по теореме Шпернера их число не превосходит $\binom{n}{\lfloor \frac{n}{2} \rfloor}$, а не 2^n. Но это рассуждение дает дополнительное слагаемое $\left(-\frac{1}{2}\right) \cdot \ln\left(\frac{n\cdot\pi}{2}\right)$ в числителе, что не меняет порядок величины.

6. Заключение

Сформулируем критерий того, когда процедура приобретает знания, и обсудим, что такое знание в противовес простой информации. Эти вопросы стали актуальными в связи с широкими философскими дискуссиями об определении интеллектуальных систем.

Представляется правильным посмотреть на это с точки зрения теории сложности вычислений: знания — это решение NP-трудных задач, а процедура способна приобретать знания, если она способна вычислительно эффективно решать класс NP-трудных задач (хотя бы приближенно).

Чтобы продемонстрировать отличие знаний от информации, рассмотрим две известные классические проблемы: обобщенную задачу Эйлера о Кенигсбергских мостах и задачу коммивояжера. В случае задачи Эйлера вопрос о построении требуемого маршрута решается жадным алгоритмом (а возможность проверяется за линейное время). В случае коммивояжера даже для существования маршрута необходимо решить NP-полную задачу. Решение такой задачи, если оно известно, следует хранить и передавать другим, так как в случае утери его нельзя будет быстро восстановить из исходных данных.

ДСМ-метод [1] может приобретать знания, но ценой экспоненциального времени работы. С точки зрения эффективности ДСМ-метод может работать на выборках очень малого объема. Напротив, описанный в этой статье метод алгебраического машинного обучения может считаться технологией приоб-

ретения знаний для выборок среднего объема и даже “BigData”, но ключевой проблемой в этом случае становится кодирование объектов со сложной структурой битовыми строками так, чтобы пересечение (побитовое умножение) выявляло бы общие фрагменты.

Альтернативой описанному подходу может служить метод последовательного расширения решетки кандидатов [29], развиваемый в последнее время С.О. Кузнецовым с коллегами.

Автор благодарен О.П. Кузнецову и С.О. Кузнецову за полезные обсуждения и поддержку и рецензентам настоящей статьи, которые способствовали существенному улучшению изложения. Особо хочу поблагодарить свою аспирантку Л.А. Якимову за совместную работу, однако за все неточности и ошибки, как обычно, несет ответственность исключительно автор.

СПИСОК ЛИТЕРАТУРЫ

1. *Финн В.К., Аншаков О.М.* (ред.) ДСМ-метод автоматического порождения гипотез: Логические и эпистемологические основания. М.: Эдиториал УРСС, 2009.
2. *Ganter B., Wille R.* Formal Concept Analysis. Transl. from German. Berlin: Springer-Verlag, 1999.
3. *Davey B.A., Priestley H.A.* Introduction to Lattices and Order. 2nd eds. Cambridge: Cambridge University Press, 2002.
4. *Valiant L.G.* A Theory of Learnable // Communications of the ACM. 1984. V. 27. No. 11. P. 1134–1142.
5. *Вапник В.Н., Червоненкис А.Я.* Теория распознавания образов (статистические проблемы обучения). М.: Наука, 1974.
6. *Rivest R.L.* Learning Decision Lists // Machine Learning. Springer. 1987. V. 2. No. 3. P. 229–246.
7. *Borchmann D., Hanika T., Obiedkov S.* Probably approximately correct learning of Horn envelopes from queries // Discrete. Appl. Math. 2020. V. 273. No. 1. P. 30–42.
8. *Yarullin R., Obiedkov S.* From Equivalence Queries to PAC Learning: The Case of Implication Theories // Int. J. Approx. Reason. 2020. V. 127. P. 1–16.
9. *Милль Дж.Ст.* Система логики силлогистической и индуктивной: Изложение принципов доказательства в связи с методами научного исследования. Пер. с англ. Изд. 5. М.: Эдиториал УРСС, 2011.
10. *Hunt E.B., Marin J., Stone P.J.* Experiments in Induction. N.Y.: Academic Press, 1966.
11. *Vere S.A.* Induction Of Concepts In The Predicate Calculus // Proc. 4th Int. Joint Conf. on Artificial Intelligence IJCAI-75. 1975. Tbilisi: JICAI. P. 281–287.
12. *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.
13. *Финн В.К.* Об одном варианте логики аргументации // НТИ. Сер. 2. 1996. № 5–6. С. 3–19.
14. *Виноградов Д.В.* Метод семантических таблиц для логики аргументации // НТИ. Сер. 2. 2006. № 11. С. 17–20.

15. *Финн В.К.* Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Ч. I // Искусственный интеллект и принятие решений. 2010. № 3. С. 3–21.

16. *Финн В.К.* Индуктивные методы Д.С. Милля в системах искусственного интеллекта. Ч. II // Искусственный интеллект и принятие решений. 2010. № 4. С. 14–40.

17. *Кузнецов С.О.* Быстрый алгоритм построения всех пересечений объектов из нижней полурешетки // НТИ. Сер. 2. 1993. № 1. С. 17–20.

18. *Кузнецов С.О., Финн В.К.* Об одной модели обучения и классификации, основанной на операции сходства // Обозрение Прикладной и Промышленной Математики. 1996. Т. 3. № 1. С. 66–90.

19. *Ganter B., Kuznetsov S.O.* Formalizing Hypotheses with Concepts // Proc. 8th Int. Conf. on Conceptual Structures ICCS–2000 (Mineau, G., Ganter, B. Eds.), Lecture Notes in Artifical Intelligence (Springer). 2000. V. 1867. P. 342–356.

20. *Виноградов Д.В.* О представлении объектов битовыми строками для ВКФ-метода // НТИ. Сер. 2. 2018. № 5. С. 1–4.

21. *Виноградов Д.В.* Скорость сходимости к пределу вероятности порождения случайного сходства при наличии контрпримеров // НТИ. Сер. 2. 2018. № 2. С. 21–24.

22. *Якимова Л.А.* Экспериментальное исследование поведения решателей, основанных на бинарной операции сходства. Магистерская диссертация по направлению подготовки 45.04.04 (Науч. руков. Виноградов Д.В.). М.: РГГУ, 2020.

23. *Dua D., Graff C.* UCI Machine Learning Repository. Irvine, CA:
[http://archive.ics.uci.edu/ml]. 2019.

24. *Сидорова Е.Ю.* Экспериментальная реализация вероятностных алгоритмов для поиска ДСМ-сходств. Дипломная работа по направлению подготовки 036000 (Науч. руков. Виноградов Д.В.). М.: РГГУ, 2013.

25. *Vinogradov D.V.* Machine Learning Based on Similarity Operation // Communications in Computer and Information Science, Springer. 2018. V. 934. P. 46–59.

26. *Кемени Дж., Снелл Дж.* Конечные цепи Маркова. Пер. с англ. М. Наука, 1970.

27. *Виноградов Д.В.* Сильная концентрация времени работы алгоритма поиска сходств // Матер. 8 Всеросс. мультиконф. по проблемам управления (МКПУ-2015) 2015. Т. 1. С. 42–45.

28. *Якимова Л.А.* Реализация ленивой схемы вычислений сходств в ВКФ-методе. Выпускная квалификационная работа бакалавра по направлению подготовки 45.03.04 (Науч. руков. Виноградов Д.В.). М.: РГГУ, 2018.

29. *Kuznetsov S., Napoli A., Makhalova T.* Gradual Discovery with Closure Structure of a Concept Lattice // Proc. 15th Int. Conf. CLA-2020. 2020. CEUR-WS. Iss. 2668. P. 145–157.

Статья представлена к публикации членом редколлегии О.П. Кузнецовым.

Поступила в редакцию 07.12.2021

После доработки 02.01.2022

Принята к публикации 26.01.2022