Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования

Колмаков Евгений Александрович

Метрический подход к модификации алгоритмов классификации на основе анализа формальных понятий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:
к.ф-м.н., доцент
Гуров Сергей Исаевич

Москва, 2015

Содержание

Аннотация

Алгоритмы классификации, основанные на анализе формальных понятий (АФП), могут работать с небинарным описанием объектов по-разному: использовать их напрямую или переходить к бинарным признакам с помощью процедуры шкалирования. Общим недостатком классификаторов второго типа является то, что они “забывают” метрическую структуру исходного признакового пространства. Основная идея этой работы — использовать исходную метрическую информацию наряду с теоретико–порядковыми отношениями между объектами и признаками. В рамках данного исследования рассматривается два способа модификации алгоритмов классификации на основе АФП с помощью метрического подхода. Во-первых, предлагается модель классификаторов, использующая как метрическую информацию из исходного признакового пространства, так и объектно-признаковые порядковые зависимости. Данная модель обобщает некоторые существующие классификаторы на основе АФП. Во-вторых, рассматривается подход, основанный на введении подходящей меры расстояния между формальными понятиями. Для этого на произвольной конечной решётке вводится семейство псевдометрик, некоторые из которых имеют отчётливую интерпретацию в терминах формальных понятий. Также предлагаются способы возможных модификаций данной меры расстояния для учёта особенностей решётки формальных понятий и варианты её использования для модификации алгоритмов классификации на основе АФП.

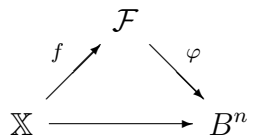
1 Введение

Анализ формальных понятий (АФП, *англ.* Formal Concept Analysis) является направлением прикладной теории решёток, позволяющим формализовать некоторые модели машинного обучения. С его помощью решаются задачи обработки и представления знаний, интеллектуального анализа данных. В частности, задачи классификации и кластеризации.

АФП и его методы предоставляют математический аппарат для исследования и представления иерархий данных, отражающих их объектно-признаковые зависимости. В обычной жизни, используя слово «понятие», мы подразумеваем некоторую абстрактную идею, выделяющую объекты некоторого класса по их общим свойствам и признакам. На основе этой идеи АФП формализует термин «понятие», что позволяет строго описывать (с помощью решёток формальных понятий) теоретико-порядковые отношения и зависимости между объектами и признаками.

Математическую основу АФП составляют отдельные главы теории решёток и теория связей Галуа. Основным объектом данной теории является решётка Галуа, отражающая иерархию понятий. Методы классификации на основе АФП обладают рядом особенностей. В частности, отсутствуют какие-либо предположения относительно статистических моделей данных.

Существует множество алгоритмов классификации на основе АФП [11], и многие из них предполагают, что объекты $x \in \mathbb{X}$ описываются при помощи бинарных (двоичных) признаков. Однако часто в прикладных задачах объекты описываются с помощью вещественных чисел, графов и других признаков. Некоторые алгоритмы используют исходное признаковое описание напрямую, например, узорные структуры [1, 3], но многие классификаторы используют признаки только после шкалирования. Для использования таких методов необходим переход (осуществляемый некоторым отображением φ) от исходного признакового пространства $\mathcal{F}$ к пространству бинарных векторов фиксированной длины, то есть булеву кубу B^n:



Обычно в задачах классификации исходное признаковое пространство $\mathcal{F}$ наделено дополнительной структурой, например, метрического пространства. Во многих случаях при отображении φ значительная часть метрической информации теряется. В новом признаковом пространстве она используется лишь в "слабой" форме, а именно как результат шкалирования.

Целью данной работы является разработка методов, совмещающих в себе метрический подход к решению задач классификации с подходом АФП, и исследование различных форм использования метрического подхода для обобщения и модификации алгоритмов классификации на основе АФП. Ниже приведено краткое описание работы по разделам.

Во втором разделе формально описывается постановка задачи классификации, излагаются необходимые определения, формулировки теорем и утверждений теории частично упорядоченных множеств, теории решёток и АФП, а также вводятся обозначения, используемые далее. В конце раздела описывается используемый в экспериментах алгоритм генерации решётки формальных понятий.

В третьем разделе проводится обзор существующих методов классификации на основе АФП, анализируются их сильные и слабые стороны. Подробно рассмотрены следующие алгоритмы: Rulearner [8], GALOIS [9], классификация на основе гипотез [6] и узорные структуры [3].

В четвертом разделе подробно описывается предлагаемый подход. Вводятся формулы оценок за классы, использующие как исходную метрическую информацию, так и решётку формальных понятий. На их основе строится модель алгоритмов классификации, обобщающая некоторые из алгоритмов, приведённых в третьем разделе, рассматриваются её аналогии с ABO [12] и метрическими алгоритмами классификации. Предложенный подход позволяет устранить общий недостаток рассмотренных алгоритмов, использующих шкалирование. Затем на произвольной конечной решётке вводится семейство псевдометрик. Указываются значения параметров, при которых введённая мера расстояния имеет простой смысл в терминах понятий и предлагаются варианты её использования для модификации алгоритмов классификации на основе АФП. В конце раздела вводится понятие пространства версий между минимальными

α-гипотезами и их минимальными генераторами, которое в дальнейшем исследуется с использованием введённой модели.

В пятом разделе описываются проведённые эксперименты по сравнению предложенного подхода с существующими алгоритмами классификации на основе АФП, а также по исследованию описанного в конце четвертого раздела пространства версий.

2 Основные определения

2.1 Постановка задачи классификации

Будем рассматривать задачу классификации (обучения по прецедентам) в следующей постановке. Задано некоторое множество *объектов* $\mathbb{X} = \bigsqcup_{y \in Y} \mathbb{X}_y$, которое разделено на конечное число попарно непересекающихся подмножеств $\mathbb{X}_y$, называемых *классами*. Принадлежность элементов множества объектов к этим классам известна только для конечного подмножества $X \subset \mathbb{X}$, называемого *обучающей выборкой*.

Каждый объект $x \in \mathbb{X}$ описывается с помощью конечного набора *признаков* $\{f_i\}_{i=1}^n$, то есть отображений $f_i \colon X \to D_i$, при этом $f_i(x)$ — значение i-го признака объекта x. Множество $D_1 \times \cdots \times D_n$ обозначается $\mathcal{F}$ и называется исходным *признаковым пространством*. Обычно $\mathbb{X}$ отождествляется с $\mathcal{F}$.

По информации о разбиении множества объектов на классы, содержащейся в обучающей выборке, необходимо построить *алгоритм классификации* $a \colon \mathbb{X} \to Y$, который для каждого нового объекта $x \in \mathbb{X}$ указывал бы метку $a(x) \in Y$ содержащего его класса, либо некоторое выделенное значение, которое обозначает отказ от классификации и соответствует ситуации, в которой алгоритм не смог решить вопрос о принадлежности объекта к одному из заданных классов.

Для оценки качества реализованных алгоритмов использовалась техника скользящего контроля. Метод t-кратной кросс–валидации заключается в следующем. Выборка случайным образом разбивается на t непересекающихся блоков одинаковой (или почти одинаковой) длины $k_1, \ldots, k_t$:

$$X^L = X_1^{k_1} \cup \cdots \cup X_t^{k_t}, \quad k_1 + \cdots + k_t = L.$$

Каждый блок по очереди становится контрольной подвыборкой, при этом обучение классификатора $a(x)$ производится по остальным $t-1$ блокам. После этого полученная характеристика $\chi(a, X)$ классификатора (например, ошибка на обучающей и контрольной выборке, число отказов и другие) усредняются:

$$CV_\chi(a, X^L) = \frac{1}{t}\sum_{i=1}^{t} \chi\left(a(X^L \setminus X_i^{k_i}), X_i^{k_i}\right),$$

где $a(X)$ — классификатор, обученный по выборке X.

2.2 Теория решёток и АФП

В этой работе мы пользуемся стандартной терминологией теории решёток и АФП. Данный раздел содержит краткое описание всех необходимых понятий и обозначений.

Пусть G и M — произвольные непустые множества, называемые *множеством объектов* и *множеством признаков*, а $I \subseteq G \times M$ — соответствие между G и M. Упорядоченная тройка $\mathbb{K} = (G, M, I)$ называется *формальным контекстом*. В случае конечных множеств объектов и признаков формальный контекст может быть задан с помощью объектно-признаковой матрицы.

Для любых $A \subseteq G$ и $B \subseteq M$ определим отображения $(\cdot)'$ следующим образом:

$$A' = \{m \in M | gIm \text{ для всех } g \in A\}, \quad B' = \{g \in G | gIm \text{ для всех } m \in B\}.$$

Эти отображения задают *соответствие Галуа* между множествами 2^G и 2^M. Мы пишем g' и m' вместо $\{g\}'$ и $\{m\}'$ для любых $g \in G, m \in M$.

Пара (A, B), где $A \subseteq G, B \subseteq M$ и $A' = B, B' = A$ называется *формальным понятием* контекста $\mathbb{K}$ с *формальным объёмом* A и *формальным содержанием* B. Определим пару отображений $ext : (A, B) \mapsto A$ и $int : (A, B) \mapsto B$.

Введём на множестве $\mathfrak{B}(\mathbb{K})$ отношение частичного порядка следующим образом:

$$(A_1, B_1) \leqslant (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2.$$

Теорема 1. *Частично упорядоченное множество* $\underline{\mathfrak{B}}(\mathbb{K}) = \langle \mathfrak{B}(\mathbb{K}), \leqslant \rangle$ *образует полную решётку, в которой*

$$\bigvee_{i \in I}(A_i, B_i) = \left(\left(\bigcup_{i \in I} A_i\right)'', \bigcap_{i \in I} B_i\right),$$

$$\bigwedge_{i\in I}(A_i, B_i) = \left(\bigcap_{i\in I} A_i, \left(\bigcup_{i\in I} B_i\right)''\right),$$

причем отображение $ext\colon (A, B) \mapsto A$ *является изоморфизмом между решётками* $\underline{\mathfrak{B}}(\mathbb{K})$ *и* $\underline{\mathfrak{B}}_{\mathfrak{ext}}(\mathbb{K})$, *а отображение* $int\colon (A, B) \mapsto B$ *является изоморфизмом между решёткой* $\underline{\mathfrak{B}}(\mathbb{K})$ *и решёткой, двойственной к* $\underline{\mathfrak{B}}_{\mathfrak{int}}(\mathbb{K})$.

Решётка $\mathfrak{B}(\mathbb{K})$ называется *решёткой формальных понятий.*

Теорема 2. *Пусть* L *— полная решётка, а* G *и* M *— множества. Предположим, что существуют отображения* $\gamma\colon G \to L$ *и* $\mu\colon M \to L$ *такие, что множество* $\gamma(G)$ *супремум–плотно в* L, *а множество* $\mu(M)$ *инфимум–плотно в* L. *Если мы определим соответствие* $I \subseteq G \times M$, *как* $gIm \Leftrightarrow \gamma(g) \leqslant \mu(m)\ \forall g \in G, m \in M$, *тогда* $L \cong \underline{\mathfrak{B}}(G, M, I)$. *В частности,* $L \cong \underline{\mathfrak{B}}(L, L, \leqslant)$.

Пусть $\langle L, \wedge, \vee\rangle$ — решётка и $x \in L$. Через $x^{\triangledown}(x^{\triangle})$ мы обозначаем *порядковый идеал* (*фильтр*), *порождённый элементом* x. Через $At(L), J(L)$ и $M(L)$ — множество всех *атомов*, $\bigvee$*-неразложимых* и $\bigwedge$*-неразложимых* элементов решётки L соответственно. Функция $f\colon L \to \mathbb{R}$ называется *супермодулярной*, если для всех $x, y \in L$:

$$f(x) + f(y) \leqslant f(x \wedge y) + f(x \vee y).$$

Более подробные определения и доказательства всех приведённых выше теорем можно найти в [4], [5] — по теории решёток, [6], [7] — по АФП.

2.3 Алгоритм построения решётки понятий

Как правило, размер решётки понятий экспоненциально зависит от числа объектов и признаков, и её построение — это вычислительно сложная задача. Использование t-кратного скользящего контроля увеличивает это время в t раз. В данной работе для построения решётки понятий был реализован алгоритм “Close By One” (или “Замыкай по-Одному”), описанный в [1]. Псевдокод алгоритма представлен в 1 и 2.

Сложность такого алгоритма описывается следующей теоремой (см. [6]).

Теорема 3. *Алгоритм “Замыкай-по-Одному” порождает множество всех понятий за время* $O(|L||G|^2|M|)$, *имея полиномиальную задержку* $O(|G|^3|M|)$ *с использованием не более* $O(|G||M| + |G|^2|L|)$ *памяти.*

Алгоритм 1 Close By One

Вход:

(G, M, I) — формальный контекст.

Выход:

L — решётка понятий.

1: $L = \varnothing$;

2: **для всех** $g \in G$

3: process$(\{g\}, g, (g'', g'))$;

Алгоритм 2 process(A, g, (C, D))

1: **если** $\{h \mid h \in C \setminus A \text{ and } h < g\} = \varnothing$ **то**

2: $L = L \cup \{(C, D)\}$;

3: **для всех** $f \in \{h \mid h \in G \setminus C \text{ and } g < h\}$

4: $Z = C \cup \{f\}$;

5: $Y = D \cap \{f'\}$;

6: $X = Y'$;

7: process$(Z, f, (X, Y))$;

Заметим, что в силу симметрии множества объектов и множества признаков для бинарных контекстов возможна двойственная формулировка данного алгоритма в терминах множеств признаков. Однако приведённое описание позволяет использовать этот же алгоритм для вычисления решётки узорных структур, в которой множество признаков заранее неизвестно.

3 Алгоритмы классификации на основе АФП

Существует множество алгоритмов классификации на основе АФП. Эти алгоритмы могут работать с небинарным описанием объектов по-разному: переходить к бинарным признакам с помощью процедуры шкалирования или использовать исходные признаки напрямую. Классификаторы первого типа можно условно разделить на следующие группы:

1. **Алгоритмы, основанные на использовании всей решётки понятий.** (Rulearner [8], GALOIS [9], GRAND, CITREC)

   Для построения классификатора методы данной группы используют всю решётку понятий целиком. Основным недостатком алгоритмов этой группы является большая сложность построения всей решётки.

2. **Алгоритмы, основанные на использовании гипотез.** ([2])

   Обучающая выборка порождает три контекста — положительный, отрицательный и недоопределенный. Для классификации используются некоторые специальные формальные содержания понятий из положительного и отрицательного контекстов, называемые гипотезами.

3. **Алгоритмы, основанные на использовании подмножеств решётки понятий.** (CLNN & CLNB [10], LEGAL)

   Алгоритмы из этой группы используют не всю решётку понятий, а только некоторое её подмножество (например, подрешётку или субиерархию Галуа (Attribute-Object-Concept poset) — частично упорядоченное множество, состоящее из объектных и признаковых понятий, то есть понятий вида (g'', g') и (m', m''), где $g \in G,\ m \in M$). Это значительно снижает сложность алгоритмов.

4. **Алгоритмы, основанные на использовании покрытия контекста.** (Induction of Product Rules [14])

   Покрытие контекста — это покрытие решётки, содержащее только понятия, локально минимизирующие некоторый функционал (например, информационную энтропию). На основе покрытий выводятся логические правила. Построение покрытий основано на эвристических методах, что несколько снижает сложность обучения.

Ниже подробно рассматриваются три алгоритма: Rulearner, GALOIS и алгоритм классификации на основе гипотез.

3.1 Алгоритм Rulearner

Данный алгоритм предназначен для поиска логических закономерностей в данных. Он использует всю решётку понятий в качестве пространства поиска для вывода правил. В основе алгоритма лежит идея о том, что решётка $\underline{\mathfrak{B}}(\mathbb{K})$ определяется множеством объектов обучающей выборки и множеством признаков (в случае приведённого контекста им соответствуют элементы множеств $J(L)$ и $M(L)$), которыми могут обладать объекты из обучающей выборки, и отражает объектно–признаковые зависимости в данных.

Алгоритм получает на вход решётку формальных понятий $\underline{\mathfrak{B}}(\mathbb{K})$, обучающую выборку C (в качестве меток отдельных элементов решётки), а также параметр N, определяющий, сколько объектов из обучающей выборки выведенное правило может классифицировать ошибочно. Результатом работы Rulearner является множество логических правил (возможно его преобразование в решающий список путём незначительных модификаций алгоритма). Выводимые правила имеют следующий вид: $f_1, f_2, \ldots f_n \Rightarrow c$, где $f_1, f_2, \ldots f_n$ — некоторый набор признаков, c — метка класса.

Для любого элемента $u \in \mathfrak{B}(\mathbb{K})$ обозначим $\mathrm{cover}(u)$ — мощность множества $J(u) = J(L) \cap u^{\triangledown}$, $\mathrm{label}(u)$ — метка класса элемента u. Для простоты изложения будем полагать, что число классов равно двум. В этом случае процедура вывода логических правил выглядит следующим образом:

- в решётке размечаются элементы, соответствующие объектам обучающей выборки, то есть атомы (их метки указаны в C);
- размечаются оставшиеся элементы $u \in \mathfrak{B}(\mathbb{K})$: если все объекты выборки из множества $J(u)$ принадлежат одному классу (допускается некоторое число объектов другого класса, задаваемое параметром N), то элементу u присваивается метка соответствующего класса, в противном случае, этот элемент помечается как MIXED;
- всем элементам присваивается метка active;
- находятся все элементы с пометкой active, которые имеют метку конкретного класса (не MIXED);

- среди таких элементов выбирается элемент с наибольшим значением $\mathrm{cover}(u)$; если таких элементов несколько, то среди них выбирается элемент v, для которого множество $M(v) = M(L) \cap v^{\triangle}$ имеет наименьшую мощность;
- пусть $f_1, f_2, \ldots f_n$ — признаки, соответствующие элементам из $M(u)$, тогда выводится правило: $f_1, f_2, \ldots f_n \Rightarrow \mathrm{label}(u)$;
- все элементы из множества $u^{\triangledown}$ помечаются как inactive;
- правила выводятся, пока все элементы не будут помечены как inactive.

Важным свойством данного алгоритма является устойчивость к шумовым данным, она обеспечивается использованием параметра N при присваивании меток элементам решётки понятий. Примеры применения данного алгоритма для решения задач классификации и результаты вычислительных экспериментов подробно описаны в статье [8].

3.2 Алгоритм GALOIS

Данный алгоритм предназначен для инкрементного построения решётки понятий, однако он также может быть использован и для решения задач классификации и кластеризации. Авторы статьи [9] предлагают две процедуры классификации. Первая из них заключается в следующем:

- после построения решётки в ней выделяется множество *непротиворечивых* понятий (то есть понятий, формальный объём которых содержит объекты только одного класса);
- для каждого объекта x тестовой выборки вычисляется близость между этим объектом и каждым из устойчивых понятий по следующей формуле:

$$\Gamma_C(x) = \|C \to x\|,$$

где C, x — бинарные векторы (соответствующие формальному содержанию понятия и признаковому описанию объекта), $\to$ — операция побитовой импликации, $\|\cdot\|$ — число единичных координат в бинарном векторе;

- объекту x присваивается метка класса понятия C с наибольшим значением близости $\Gamma_C(x)$.

Вторая процедура классификации выглядит следующим образом:

- выделяется множество непротиворечивых понятий;
- для каждого объекта x тестовой выборки выбираются устойчивые понятия, формальные содержания которых вложены в множество признаков, которыми обладает объект;
- объекту x присваивается метка того класса, понятий которого больше всего в множестве, построенном на предыдущем шаге.

3.3 Классификация на основе гипотез

Предположим, что в задаче классификации заданы множества положительных и отрицательных примеров относительно некоторого целевого свойства z, а также множество недоопределённых примеров — объектов, для которых неизвестно значение предиката обладания свойством z.

Такие входные данные могут быть заданы с помощью трёх контекстов: $\mathbb{K}_+ = (G_+, M, I_+)$, $\mathbb{K}_- = (G_-, M, I_-)$ и $\mathbb{K}_\tau = (G_\tau, M, I_\tau)$ — положительный, отрицательный и недоопределённый контексты соответственно. Операторы Галуа в этих контекстах обозначаются верхними индексами $+, -$ и τ соответственно. Формальное понятие (A_+, B_+) положительного контекста называется *положительным понятием*, при этом A_+ называется *положительным формальным объёмом*, а B_+ — *положительным формальным содержанием*. Аналогично для *отрицательных* и *недоопределённых понятий*. Положительное формальное содержание B_+ называется:

- *положительной предгипотезой*, если оно не является формальным содержанием ни одного отрицательного понятия;
- *положительной гипотезой*, если оно не является подмножеством формального содержания какого-либо понятия вида (g^{--}, g^-) (такие понятия называются *отрицательными примерами*);

- *фальсифицированной положительной гипотезой*, если оно является подмножеством формального содержания понятия некоторого отрицательного примера.

Аналогично для *отрицательных предгипотез*, *гипотез* и *фальсифицированных гипотез*. Если формальное содержание недоопределенного примера $g \in G_\tau$ содержит положительную (отрицательную) гипотезу, то говорят, что эта гипотеза является *гипотезой в пользу положительной* (*отрицательной*) *классификации* g. Классификация недоопределённого объекта $g \in G_\tau$ происходит так:

- строятся множества положительных и отрицательных гипотез;
- если g^τ содержит хотя бы одну положительную гипотезу и ни одной отрицательной, то g относится к положительному классу;
- если g^τ содержит хотя бы одну отрицательную гипотезу и ни одной положительной, то g относится к отрицательному классу;
- если g^τ содержит как положительную, так и отрицательную гипотезы, либо g^τ не содержит ни положительных, ни отрицательных гипотез, то происходит отказ от классификации.

Следует отметить, что приведённый выше алгоритм обычно даёт большую долю отказов от классификации. Более подробное описание алгоритма в [2], результаты вычислительных экспериментов в [11]. Данный подход также носит название *ДСМ-метод* в честь английского философа, логика и экономиста Джона Стюарта Милля.

Достоинства рассмотренных алгоритмов:

- нет предположений относительно статистической модели данных;
- алгоритмы используют объектно-признаковые зависимости;
- все алгоритмы путём достаточно несложных модификаций могут быть сделаны устойчивыми к шуму.

Недостатки рассмотренных алгоритмов:

- в общем случае экспоненциальная сложность построения решётки;

- метрическая информация используется не в явном виде, а только как результат шкалирования;
- существенная доля отказов от классификации.

3.4 Узорные структуры

Примером алгоритмов второго типа является подход, основанный на введении операции сходства на множестве описаний объектов [1, 3]. Аналогом формального контекста здесь является множество объектов $g \in G$ вместе с множеством описаний $d \in D$, для которых задана операция "сходства" $\sqcap$, при этом каждому объекту $g \in G$ сопоставляется некоторое описание $\delta(g) \in D$.

Упорядоченная тройка $\mathbb{P} = (G, \underline{D}, \delta)$, где G — некоторое множество, $\underline{D} = (D, \sqcap)$ — нижняя полурешётка, $\delta\colon G \to D$ — отображение такое, что множество $\delta(G)$ порождает полную подполурешётку $(D_\delta, \sqcap)$ в $\underline{D}$, называется *узорной структурой*. Элементы множества D называются *узорами*. На $\underline{D}$ определён порядок поглощения:

$$c \sqsubseteq d \iff c \sqcap d = c.$$

Условие порождения множеством $\delta(G)$ полной подполурешётки выполняется автоматически в двух важных ситуациях: когда полурешётка $\underline{D}$ полна и когда множество G конечно. Для узорной структуры $(G, \underline{D}, \delta)$ введём пару отображений:

$$A^\diamond = \prod_{g \in A} \delta(g) \quad \forall A \subseteq G,$$

$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \quad \forall d \in D.$$

Утверждение 1. *Операторы $\diamond$ задают соответствие Галуа между множествами $(G, \subseteq)$ и $(D, \sqsubseteq)$.*

Пара (A, d), где $A \subseteq G$ и $d \in M$, называется *узорным понятием* структуры $\mathbb{P}$, если $A^\diamond = d$ и $d^\diamond = A$. При этом A называется *объёмом* понятия, а d — *узорным содержанием* понятия.

Ниже приведены примеры используемых на практике узорных структур:

- Пусть задан контекст (G, M, I), тогда $(G, (2^M, \cap), (\cdot)')$ есть узорная структура. Её узорные понятия — это формальные понятия исходного контекста.

- Пусть объекты описываются с помощью вещественных векторов размера n. Интервальные узорные структуры:

$$[a_1, b_1] \sqcap [a_2, b_2] = [\min(a_1, a_2), \max(b_1, b_2)],$$

$$[a_1, b_1] \sqsubseteq [a_2, b_2] \Leftrightarrow a_1 \leqslant a_2 \ \& \ b_1 \geqslant b_2.$$

  $d \in D$ есть n-мерный вектор из интервалов, а операция $\sqcap$ применяется покомпонентно. При этом $\delta((x_1, \ldots, x_n)) = ([x_1, x_1], \ldots, [x_n, x_n])$.

Особо стоит выделить узорные структуры на графах [1]. Пусть P есть множество всех конечных графов с пометками из множества $(L, \leqslant)$. Такой граф имеет вид $\Gamma = ((V, l), E)$, где l есть функция приписывания меток вершинам.

$\Gamma_1 = ((V_1, l_1), E_1)$ *поглощает* $\Gamma_2 = ((V_2, l_2), E_2)$, если существует инъективное отображение $\varphi\colon V_2 \to V_1$, которое:

- сохраняет ребра: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$;
- учитывает порядок на пометках вершин: $l_2(v) \leqslant l_1(\varphi(v))$.

Описаниями будут множества помеченных графов. Введём операцию сходства $\sqcap$. Для произвольной пары помеченных графов $X \sqcap Y$ — множество всех максимальных общих подграфов. Для произвольных описаний $\{X_1, \ldots, X_n\} \sqcap \{Y_1, \ldots, Y_m\}$ — множество максимальных элементов (относительно порядка поглощения) в множестве $\bigcup_{i,j}(X_i \sqcap Y_j)$.

Отличия вычислений в узорных структурах от вычислений в формальных контекстах:

- Множество признаков в исходных данных в явном виде отсутствует, и вычисление узорных понятий возможно лишь в стратегии “снизу-вверх” (от минимальных объемов и максимальных содержаний к максимальным объемам и минимальным содержаниям).
- Вычислительная сложность. Например, уже задача вычисления отношения $\sqsubseteq$ для помеченных графов является NP-полной (в силу NP-полноты задачи ИЗОМОРФИЗМ ПОДГРАФУ), а вычисление операции $X \sqcap Y$ NP-трудно, тогда как вычисление аналогичного отношения $\subseteq$ и операции $\cap$ в формальных контекстах связано с выполнением элементарных операций на битовых строках.

Обобщение ДСМ-метода на узорные структуры позволяет применять их для классификации. Пусть E_+ и E_- — множества положительных и отрицательных примеров по отношению к целевому аттрибуту w.

Узор $h \in D$ называется *положительной гипотезой*, если

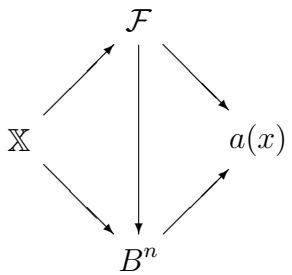
$$h^\diamond \cap E_- = \varnothing \text{ и } \exists A \subseteq E_+ \colon A^\diamond = h.$$

Если $g \in G_\tau$ — недоопределенный пример, то положительная гипотеза h есть гипотеза *в пользу положительной классификации* объекта g, если $h \sqsubseteq \delta(g)$. Аналогично для *отрицательных гипотез*. Классификация осуществляется аналогично случаю бинарных контекстов.

4 Метрический подход к модификации алгоритмов

Из обзора существующих методов в предыдущем разделе можно сделать следующие выводы. Узорные структуры позволяют обойти проблему бинарных контекстов и использовать исходное описание напрямую (но только при условии, что на нём можно определить операцию сходства), но зачастую алгоритмы порождения решётки, а значит, и классификации вычислительно сложнее (например, для графов), чем при использовании бинарных контекстов.

Общим недостатком всех рассмотренных алгоритмов, работающих с бинарными признаками, является то, что они используют признаки, полученные в результате процедуры шкалирования, из-за чего происходит потеря метрической информации. Возникает следующая идея: после шкалирования не отказываться от информации, доступной в исходном признаковом пространстве, а наоборот использовать её наряду с теоретико-порядковыми отношениями между объектами и признаками, которые задаются решёткой понятий. Тогда процесс построения классификатора $a(x)$ может быть представлен следующей схемой:



Важным является то, что пространства $\mathcal{F}$ и B^n вместе с дополнительными структурами на них (в первом пространстве — это метрика, во втором — это формальный контекст, задаваемый обучающей выборкой) используются одновременно, что предоставляет более богатый набор методов для построения алгоритма классификации и позволяет сохранить исходную метрическую информацию и использовать её. Предлагаемые ниже обобщения и модификации в первую очередь направлены на вторую группу алгоритмов, однако это идеи могут быть применены и для модификации алгоритмов классификации на основе узорных структур.

4.1 Введение метрических оценок

Пусть $\mathcal{H}_+$ и $\mathcal{H}_-$ — построенные по обучающей выборке множества понятий, содержания которых являются положительными и отрицательными гипотезами. $S(x, C)$ — мера близости между объектом x и множеством объектов C (основанная на функции расстояния из исходного признакового пространства $\mathcal{F}$).

Пусть $(\mathcal{F}, \rho)$ — метрическое пространство. Тогда разумно выбирать меру близости $S(x, C)$ как некоторую невозрастающую функцию от $\rho(x, C)$. Расстояние от точки x до множества C может быть определено разными способами, например:

- $\rho(x, C) = \inf_{c \in C} \rho(x, c)$
- $\rho(x, C) = \rho(x, c_*)$, где c_* — некоторый «центр масс» множества C.
- $\rho(x, C) = \frac{1}{|C|} \sum_{c \in C} \rho(x, c)$ и другие.

Выбор самой функции расстояния ρ является отдельным и достаточно непростым вопросом и здесь не рассматривается.

Определим оценки за положительный и отрицательный классы:

$$\Gamma_+(x) = \sum_{C \in \mathcal{H}_+} [int(C) \subseteq int(x)] S(x, ext(C)),$$

$$\Gamma_-(x) = \sum_{C \in \mathcal{H}_-} [int(C) \subseteq int(x)] S(x, ext(C)),$$

$$\Gamma(x) = \Gamma_+(x) - \Gamma_-(x).$$

Тогда итоговый классификатор будет иметь следующий вид:

$$a(x) = \text{sign } \Gamma(x).$$

Что можно сказать о качестве предложенного классификатора $a(x)$ по сравнению с рассмотренными выше алгоритмами? Справедливо следующее

Утверждение 2. *Если алгоритм классификации на основе гипотез верно распознаёт объект, то это же относится и к классификатору* $a(x) = \operatorname{sign} \Gamma(x)$.

Поскольку алгоритм $a(x)$ отказывается от классификации в меньшем числе случаев, чем раньше, но число ошибок (среди классифицированных объектов) может увеличиться. Можно предложить следующую модификацию:

$$a_R(x) = \begin{cases} \operatorname{sign} \Gamma(x), & |\Gamma(x)| > R; \\ \text{отказ от классификации}, & \text{иначе.} \end{cases}$$

Здесь R — некоторая положительная константа. $a_R(x)$ классифицирует уверенее, но с ростом R увеличивается число отказов. Такой подход влечёт за собой проблему выбора порога R.

4.2 Аналогия с алгоритмами вычисления оценок

В приведённом выше алгоритме при вычислении оценок используются множества положительных и отрицательных гипотез, то есть подмножества решётки понятий специального вида. Возникает идея обобщения данных оценок на подмножества произвольного вида, тем или иным образом характеризующие отдельные классы $y \in Y$.

Формализуем эту идею. Зафиксируем некоторое множество (формальных или узорных) понятий $\mathcal{C}$, которое будем называть *системой опорных понятий*. Предположим также, что каждое понятие $C \in \mathcal{C}$ характеризует некоторый класс $y \in Y$ и только его, то есть

$$\mathcal{C} = \bigsqcup_{y \in Y} \mathcal{C}_y, \text{ где } Y \text{ — множество классов.}$$

Тогда введём оценку объекта x за класс y следующим образом:

$$\Gamma_y(x) = \sum_{C \in \mathcal{C}_y} S(x, C).$$

Итоговый классификатор имеет стандартный вид:

$$a(x) = \arg\max_{y \in Y} \Gamma_y(x).$$

Оценки такого вида схожи с оценками за классы, которые используются в ABO [12], а множества $\mathcal{C}_y$ являются аналогами опорных множеств.

Приведём конкретные примеры системы опорных множеств $\mathcal{C}$ и функции близости $S(x, C)$ и рассмотрим получающиеся при этом алгоритмы:

- $\mathcal{C} = \mathcal{H}_+ \bigsqcup \mathcal{H}_-$ — множество положительных и отрицательных гипотез.

  $S(x, C) = [int(C) \subseteq x']\hat{S}(x, ext(C))$, где $\hat{S}(x, ext(C))$ — некоторая функция близости. При таком выборе $\mathcal{C}$ и $S(x, C)$ получаем алгоритм из пункта 4.1.

- $\mathcal{C} = \bigsqcup_{y \in Y} \mathcal{C}_y$ — множество непротиворечивых понятий.

  $S(x, C) = |\big(M\ int(C)\big) \cup x'|$ — модифицированный алгоритм GALOIS(1).

  $S(x, C) = [int(C) \subseteq x']$ — алгоритм GALOIS(2).

4.3 Аналогия с метрическими алгоритмами классификации

Пусть $\mathcal{C} = \{C_1, \ldots, C_n\}$ — множество опорных понятий. Предположим, что в исходном признаковом пространстве $\mathcal{F}$ введена мера расстояния ρ. Упорядочим $\mathcal{C}$ по возрастанию расстояний от C_i до классифицируемого объекта x:

$$\rho(x, C_x^{(1)}) \leqslant \rho(x, C_x^{(2)}) \leqslant \cdots \leqslant \rho(x, C_x^{(n)}),$$

$C_x^{(i)}$ — i-ый сосед объекта x среди $\mathcal{C}$, $y_x^{(i)}$ — класс, к которому относится понятие $C_x^{(i)}$.

Определим оценку объекта x за класс y:

$$\Gamma_y(x) = \sum_{i=1}^{n} w_i(x)[y_x^{(i)} = y],$$

$w_i(x)$ — вес i-го соседа объекта x (неотрицательная и невозрастающая по i функция).

Введённые оценки полностью аналогичны оценкам, используемым в метрических алгоритмах классификации, за исключением того, что в качестве соседей выступают не объекты, а опорные понятия.

Таким образом, выбирая подходящие веса $w_i(x)$, получаем аналоги всех известных метрических методов (kNN, Parzen window, potential functions и других [16]), но в терминах понятий. Например:

- $w_i(x) = [i \leqslant k]$ — метод k ближайших соседей;

- $w_i(x) = [i \leqslant k]w_i$ — метод k взвешенных ближайших соседей, здесь w_i — вес, зависящий только от номера соседа;

- $w_i(x) = K\left(\frac{\rho(x,C_x^{(i)})}{h(x)}\right)$ — метод Парзеновского окна переменной ширины, здесь $K(z)$ — невозрастающая положительная функция на $[0,1]$, $h(x)$ — ширина окна (например, $h(x) = \rho(x, C_x^{(k+1)})$).

Предложенные в этом и предыдущих пунктах методы имеют следующие достоинства:

- использование метрической информации из исходного признакового пространства при вычислении близости $S(x,C)$ или расстояния $\rho(x,C)$ наряду с объектно-признаковыми зависимостями, задаваемыми решёткой понятий;

- предложенные алгоритмы являются обобщением некоторых из уже существующих и могут быть использованы для их модификации (например, алгоритма GALOIS или классификации на основе гипотез);

- доля отказов от классификации (по сравнению с рассмотренными методами) существенно ниже;

Следует отметить, что проблема сложности построения решётки понятий остаётся нерешённой. Возможный вариант решения — выбор системы опорных понятий $\mathcal{C}$ таким образом, чтобы избежать построения всей решётки. Также, засчёт использования дополнительной информации об объектах, возникают вопросы, касающиеся выбора метрики в исходном признаковом пространстве, выбора меры близости $S(x,C)$ и системы опорных понятий $\mathcal{C}$.

4.4 Псевдометрика на понятиях

Другой подход, использующий понятие близости в алгоритмах АФП, заключается в введении функции расстояния на множестве всех понятий. При выводе логических правил в алгоритме Rulearner [8] наиболее важными характеристиками элемента x решётки понятий при его сравнении с другими понятиями были значение функции $\mathrm{cover}(x) = |J(L) \cap x^{\triangledown}|$ и множество $M(x) = M(L) \cup x^{\triangle}$. В случае приведённого контекста, это согласуется с тем, что понятие характеризуется своим объёмом (отдельные

объекты соответствуют $\bigvee$-неразложимым элементам решётки) и содержанием (отдельные признаки соответствуют $\bigwedge$-неразложимым элементам решётки).

Таким образом, $\mathrm{cover}(x)$ соответствует числу объектов из обучающей выборки, покрываемых понятием x, а множество $M(x)$ — признакам, которые характеризуют данное понятие. Воспользуемся этими соображениями для введения функции близости на произвольной конечной решётке. В силу утверждений, двойственных к теоремам 3.1 и 3.3 из статьи [15], справедлива следующая

Теорема 4. *Пусть $\langle L, \wedge, \vee\rangle$ — решётка, и функция $f\colon L \to \mathbb{R}$ изотонна и супермодулярна. Тогда $d_f(x,y) = f(x) + f(y) - 2f(x \wedge y)$ является псевдометрикой на этой решётке.*

Рассмотрим произвольную конечную решётку $\langle L, \wedge, \vee\rangle$, непустое подмножество $D \subseteq L$ и функцию $f\colon L \to \mathbb{Z}_+$, определённую следующим образом:

$$f(x) = |D(x)|, \text{ где } D(x) = D \cap x^{\triangledown}.$$

Утверждение 3. *Функция $f(x)$ является изотонной и супермодулярной.*

Доказательство. Изотонность f следует из следующей цепочки импликаций:

$$x \leqslant y \Rightarrow x^{\triangledown} \subseteq y^{\triangledown} \Rightarrow D(x) \subseteq D(y) \Rightarrow f(x) = |D(x)| \leqslant |D(y)| = f(y).$$

Перейдём к доказательству супермодулярности:

$$f(x) + f(y) = |D(x)| + |D(y)| = |D(x) \cup D(y)| + |D(x) \cap D(y)| \leqslant$$
$$\leqslant f(x \vee y) + f(x \wedge y).$$

Докажем последнее неравенство. Включение $D(x) \cup D(y) \subseteq D(x \vee y)$ следует из включений:

$$x \leqslant x \vee y \Rightarrow D(x) \subseteq D(x \vee y),$$

$$y \leqslant x \vee y \Rightarrow D(y) \subseteq D(x \vee y),$$

Равенство $D(x) \cap D(y) = D(x \wedge y)$ следует из того, что $x^{\triangledown} \cap y^{\triangledown} = (x \wedge y)^{\triangledown}$. □

Таким образом, согласно теореме 4, функция $f(x)$ индуцирует псевдометрику $d_f(x,y)$ на решётке, определяемую следующей формулой:

$$d_f(x,y) = f(x) + f(y) - 2f(x \wedge y).$$

Значение функции $d_f(x,y)$ имеет простой смысл.

Утверждение 4. $d_f(x,y) = |D(x) \oplus D(y)|$, *где* $A \oplus B = (A \setminus B) \cup (B \setminus A)$.

Доказательство. На последнем шаге доказательства утверждения 3 установлено соотношение $D(x) \cap D(y) = D(x \wedge y)$.

$$\begin{aligned} f(x) + f(y) - 2f(x \wedge y) &= |D(x)| + |D(y)| - 2|D(x \wedge y)| = \\ &= |D(x)| + |D(y)| - 2|D(x) \cap D(y)| = \\ &= |D(x) \cup D(y)| + |D(x) \cap D(y)| - 2|D(x) \cap D(y)| = \\ &= |D(x) \cup D(y)| - |D(x) \cap D(y)| = |D(x) \oplus D(y)|. \end{aligned}$$

□

Следствие 1. Если $\langle L, \wedge, \vee \rangle$ — конечная булева алгебра и $D = At(L)$, то $d_f(x,y)$ — это в точности расстояние Хэмминга.

Стоит отметить, что в общем случае отношение эквивалентности $\sim$, заданное следующим образом:

$$x \sim y \iff d_f(x,y) = 0$$

не является конгруэнцией, поэтому с его помощью нельзя перейти от псевдометрического пространства (L, d_f) к метрическому, оставаясь в рамках теории решёток, поскольку фактор-множество $L/\sim$ не всегда будет решёткой.

Для сравнения формальных понятий разумно выбирать $D = J(L)$ или $D = At(L)$. В терминах этой псевдометрики два понятия тем ближе, чем меньше примеров (то есть понятий вида (g'', g'), где $g \in G$) покрывается только одним из данных понятий и не покрывается другим. Более того, функция $\mathrm{cover}(x)$ выражается через функцию $d_f(x,y)$ следующим образом: $\mathrm{cover}(x) = d_f(x, 0)$.

Одним из недостатков введённой меры расстояния является то, что число элементов из $D(x \wedge y)$ никак не учитывается, что в некоторых случаях может привести к неадекватным оценкам расстояния. Возможные модификации:

1. Учёт числа атомов из пересечения путём «нормировки», например:

$$d(x,y) = \frac{|D(x) \oplus D(y)|}{|D(x) \cup D(y)|}.$$

2. Добавление весов элементам D. Например, пусть w_d — доля гипотез (или непротиворечивых понятий), покрывающих $d \in D$. Тогда $d(x, y)$ примет вид:

$$d(x, y) = \sum_{d \in D(x) \oplus D(y)} w_d.$$

Расстояние между понятиями можно также применять для модификации классификаторов на основе АФП. Пусть, например, для классификации объекта x используется алгоритм на основе гипотез. Предположим, что имеется две гипотезы H_1^+, H_2^+ в пользу положительной классификации x и две гипотезы H_1^-, H_2^- в пользу отрицательной классификации x. В этом случае стандартный алгоритм отказывается от классификации.

Предположим, что нам дополнительно известны расстояния $d(H_1^+, H_2^+), d(H_1^-, H_2^-)$ и $d(H_1^+, H_2^+) \gg d(H_1^-, H_2^-)$. Тогда имеет смысл отнести x к положительному классу, поскольку далёкие по отношению к введённой мере понятия являются менее "коррелированными" (так как они покрывают большое число различных примеров), а значит, их голоса более значимы.

Мера расстояния между понятиями также может быть использована для уменьшения размера системы опорных понятий (например, гипотез), используемой классификатором. Это способствует увеличению обобщающей способности классификатора, уменьшению переобучения и удалению "шумовых" понятий.

4.5 Пространство версий между минимальными гипотезами и минимальными генераторами

Впервые термин "пространство версий" был предложен в работе [13]. С тех пор с этим понятием связано одно из направлений в области машинного обучения. Пространства версий могут быть определены разными эквивалентными способами. Здесь мы будем использовать их определение через предикат соответствия (matching predicate).

Чтобы определить пространство версий, необходимо задать следующие объекты. *Язык примеров* L_i, средствами которого описываются примеры (instances). *Язык понятий* L_c, средствами которого описываются понятия (concepts). *Предикат соответствия (matching predicate)* $M(c, i)$, который указывает — покрывает или не

покрывает понятие c пример $i : M(c, i)$ имеет место тогда и только тогда когда i есть пример понятия c. Множество понятий частично упорядоченно отношением быть более общим чем понятие или быть эквивалентным ему (more general or equal) $\leqslant$: для $c_1, c_2 \in L_c$ имеет место $c_2 \leqslant c_1$ (формальное понятие c_1 более общее чем понятие c_2 или совпадает с ним), если каждый пример понятия c_2 есть пример понятия c_1. Множества I_+ и I_- *положительных и отрицательных примеров* целевого понятия (target concept). При этом $I_+ \cap I_- = \varnothing$. *Отношение согласованности* $\mathrm{cons}(c, I_+, I_-)$ имеет место тогда и только тогда когда для каждого $i \in I_+$ выполняется $M(c, i)$ и для каждого $i \in I_-$ выполняется $\neg M(c, i)$. Понятие для которого имеет место отношение согласованности называется *согласованным (consistent).*

Множество всех согласованных понятий в языке L_c называется *пространством версий (version space)* и обозначается $VS(L_c, L_i, M(c, i), I_+, I_-)$. Задача обучения в пространстве версий определяется следующим образом: по заданной пятёрке $L_c, L_i, M(c, i), I_+, I_-$ найти пространство версий $VS(L_c, L_i, M(c, i), I_+, I_-)$. Более подробное описание приводимых понятий и утверждения, касающиеся их, можно найти в [6].

Если задано множество признаков M и множества положительных и отрицательных примеров G_+ и G_-, то можно рассмотреть пространство версий

$$VS(2^M, 2^M, \subseteq, G_+, G_-).$$

Гипотезы и формальные понятия часто являются слишком "жёсткими" классификаторами, поэтому представляет интерес рассмотреть их ослабления. Ослаблением понятия гипотезы является α-гипотеза: формальное содержание B_+ положительного понятия (A_+, B_+) называется *положительной α-гипотезой*, если оно является подмножеством формального содержания g^- не более чем $\lceil \alpha |G_-| \rceil$ отрицательных примеров $g \in G_-$, т.е.

$$|\{g \in G_- \mid B_+ \subseteq g^-\}| \leqslant \alpha |G_-|.$$

Аналогично для *отрицательных α-гипотез*. Минимальные по включению гипотезы называются *минимальными*.

Ослаблением термина "понятие" являются генераторы. Множество признаков $D \subseteq B$ называется *генератором* формального содержания понятия (A, B), если $D'' = B$. Минимальные по включению генераторы называются *минимальными*.

С помощью предложенной модели классификаторов в данной работе было частично изучено пространство версий между минимальными α-гипотезами и минимальными генераторами для этих гипотез. Введём следующие обозначения: $\mathcal{H}_+^{\min}(\alpha), \mathcal{H}_-^{\min}(\alpha)$ — множества положительных и отрицательных минимальных α-гипотез, $\mathcal{G}_+^{\min}(\alpha), \mathcal{G}_-^{\min}(\alpha)$ — множества минимальных генераторов для α-гипотез из множеств $\mathcal{H}_+^{\min}(\alpha)$ и $\mathcal{H}_-^{\min}(\alpha)$ соответственно.

Различным вариантам выбора системы опорных понятий $\mathcal{C}$ с использованием описанных выше множеств отвечают четыре группы методов:

- $\mathcal{C} = \mathcal{H}_+^{\min}(\alpha) \bigsqcup \mathcal{H}_-^{\min}(\alpha)$ — положительные гипотезы против отрицательных гипотез.
- $\mathcal{C} = \mathcal{H}_+^{\min}(\alpha) \bigsqcup \mathcal{G}_-^{\min}(\alpha)$ — положительные гипотезы против отрицательных генераторов.
- $\mathcal{C} = \mathcal{G}_+^{\min}(\alpha) \bigsqcup \mathcal{H}_-^{\min}(\alpha)$ — положительные генераторы против отрицательных гипотез.
- $\mathcal{C} = \mathcal{G}_+^{\min}(\alpha) \bigsqcup \mathcal{G}_-^{\min}(\alpha)$ — положительные генераторы против отрицательных генераторов.

При этом в каждом случае возможно использование как классического ДСМ-метода, так и его модификации с помощью метрических оценок.

5 Тестирование предложенной модели

Для тестирования некоторых из предложенных методов и сравнения их с алгоритмами, рассмотренными в третьем разделе, были использованы два набора данных из UCI Machine Learning Repository [17]. Помимо алгоритмов из третьего раздела тестировались:

1. Модификация алгоритма GALOIS(1) (см. пример из пункта 4.2).
2. Модификации алгоритма классификации на основе гипотез с помощью введения метрических оценок (см. пункт 4.1). Была выбрана функция близости

$S(x, C) = K(\rho(x, C))$. В качестве $K(r)$ использовалась одна из функций:

$$K_1(r, a) = \frac{1}{1 + \exp(-ar)}, \qquad K_2(r, a) = \frac{1}{a + r}.$$

Для вычисления $\rho(x, C)$ использовались следующие варианты:

$$\rho_1(x, C) = \inf_{c \in C} \rho(x, c), \quad \rho_2(x, C) = \frac{1}{|C|} \sum_{c \in C} \rho(x, c), \quad \rho_3(x, C) = \sup_{c \in C} \rho(x, c).$$

Введём следующие обозначения:

ν_c — доля объектов, на которых не произошло отказа от классификации;

$\nu_r = 1 - \nu_c$ — доля отказов от классификации;

e_t — общая доля ошибок классификации (отказ также считается ошибкой);

e_r — доля ошибок классификации среди всех классифицированных объектов.

5.1 Сравнение алгоритмов

SPECT Heart Data Set. Задача состоит в прогнозировании состояния пациента на основе изображений его сердечной однофотонной эмиссионной компьютерной томографии. Данные разделены на обучающую (80 объектов) и контрольную (187 объектов) выборки. Для этой задачи существует два набора данных: SPECT и SPECTF Heart Data Set. В первом каждый объект описывается с помощью 22 бинарных признаков, а во втором — с помощью 44 числовых. В качестве функции расстояния во втором случае использовалась обычная евклидова метрика. Результаты тестирования представлены в таблице 1.

Liver Disorders Data Set. Задача состоит в прогнозировании наличия болезни печени на основе анализов крови и среднего употребления алкоголя. Данные были разделены на обучающую (150 объектов) и контрольную выборки (195 объектов). Изначально каждый объект описывался с помощью 6 числовых признаков. Была проведена простая процедура бинаризации: каждый признак линейным преобразованием переводился в $[0, 1]$, этот отрезок делился на 5 частей, и каждому признаку ставился в соответствие вектор из B^5 с единицей в k-ой позиции, где k — номер интервала, в который попал признак после такого преобразования. В исходном признаковом пространстве использовалась евклидова метрика. Результаты тестирования представлены в таблице 2.

Алгоритм	ν_r	ν_c	e_t	e_r
GALOIS(1)	0.27	0.73	0.33	0.1
Modified GALOIS(1)	0.15	0.85	0.2	0.07
GALOIS(2)	0.2	0.8	0.27	0.12
Rulearner	0.25	0.75	0.3	0.09
Hypotheses-based	0.41	0.59	0.62	0.18
$K = K_1, a = 0.0125, \rho = \rho_1$	0.2	0.8	0.3	0.13
$K = K_1, a = 0.0125, \rho = \rho_2$	0.2	0.8	0.25	0.09
$K = K_1, a = 0.0125, \rho = \rho_3$	0.2	0.8	0.27	0.12
$K = K_1, a = 1, \rho = \rho_2$	0.27	0.73	0.33	0.1
$K = K_2, a = 1, \rho = \rho_1$	0.2	0.8	0.29	0.12
$K = K_2, a = 1, \rho = \rho_2$	0.2	0.8	0.31	0.15

Таблица 1. Результаты тестирования алгоритмов. Задача SPECT.

Алгоритм	ν_r	ν_c	e_t	e_r
GALOIS(1)	0.2	0.8	0.47	0.42
Modified GALOIS(1)	0.1	0.9	0.45	0.39
GALOIS(2)	0.1	0.9	0.43	0.38
Rulearner	0.03	0.97	0.46	0.44
Hypotheses-based	0.7	0.3	0.83	0.4
$K = K_1, a = 1, \rho = \rho_1$	0.12	0.88	0.52	0.45
$K = K_1, a = 0.01, \rho = \rho_2$	0.1	0.9	0.54	0.49
$K = K_1, a = 0.25, \rho = \rho_3$	0.12	0.88	0.53	0.47
$K = K_2, a = 200, \rho = \rho_1$	0.1	0.9	0.47	0.41
$K = K_2, a = 150, \rho = \rho_2$	0.1	0.9	0.41	0.36
$K = K_2, a = 150, \rho = \rho_3$	0.1	0.9	0.46	0.4

Таблица 2. Результаты тестирования алгоритмов. Задача Liver Disorders.

Целью проведённых экспериментов было не решение конкретных задач классификации, а сравнение методов на основе АФП между собой. В связи с этим применялась очень простая процедура бинаризации, чем объясняется высокая ошибка всех алгоритмов во второй задаче, и параметр a выбирался с помощью перебора по сетке небольшого размера. Выбор подходящей процедуры бинаризации признаков является отдельной темой для исследований и может существенно улучшить качество классификации. Например, можно использовать интервалы переменной длины.

На основе полученных результатов можно сделать следующие выводы об эффективности предложенных алгоритмов:

- Во всех случаях число отказов от классификации, по сравнению с алгоритмом классификации на основе гипотез, существенно уменьшено, при этом в первой задаче средняя относительная доля ошибок e_r существенно ниже, а во второй незначительно больше, чем e_r алгоритма классификации на основе гипотез.

- Общая доля ошибок классификации e_t модификаций алгоритма на основе гипотез в первой задаче сравнима с e_t алгоритма Rulearner, а во второй — с e_t алгоритма GALOIS. В обеих задачах она значительно ниже, чем у алгоритма классификации на основе гипотез.

- В обеих задачах модификация алгоритма GALOIS(1) улучшила качество классификации и уменьшила число отказов.

- Выбор конкретной функции $K(r, a)$ и варианта вычисления расстояния $\rho(x, C)$ почти не влияют на количество отказов от классификации, однако влияют на число ошибок. Таким образом, с помощью правильного их подбора можно улучшить качество классификации.

5.2 Минимальные α-гипотезы и минимальные генераторы

Для тестирования классификаторов на основе минимальных α-гипотез и их минимальных генераторов использовалось 5 наборов данных: Monk1, Monk2, Monk3, SPECT и Voting records. Ниже приведены результаты для задач Monk3, SPECT и Voting records. Использовалась 5-кратная кросс-валидация. Измерялись следующие

характеристики: TPR — true positive rate (доля верно классифицированных положительных примеров среди всех положительных примеров), SPC — specificity (доля верно классифицированных отрицательных примеров среди всех отрицательных примеров), ACC — accuracy (доля верно классифицированных примеров среди всех примеров), PPV — positive predictive value (доля верно классифицированных положительных примеров среди примеров, классифицированных положительно). Все характеристики, указанные ниже, представляют собой усреднённые значения соответствующих характеристик, полученных на каждой итерации кросс-валидации.

Результаты тестирования представлены ниже. Для каждого значения порога α представлена таблица с результатами, её строки соответствуют характеристикам классификаторов, а столбцы — используемой для классификации схеме:

1. (+)-гипотезы против (−)-гипотез (метрические оценки);
2. (+)-гипотезы против (−)-генераторов (метрические оценки);
3. (+)-генераторы против (−)-гипотез (метрические оценки);
4. (+)-генераторы против (−)-генераторов (метрические оценки);
5. (+)-гипотезы против (−)-гипотез (ДСМ-метод);
6. (+)-гипотезы против (−)-генераторов (ДСМ-метод);
7. (+)-генераторы против (−)-гипотез (ДСМ-метод);
8. (+)-генераторы против (−)-генераторов (ДСМ-метод);

Из приведённых результатов можно сделать следующие выводы о пространстве версий между минимальными α-гипотезами и минимальными генераторами для этих гипотез:

- При пороге $\alpha > 0{,}1$ классификаторы, использующие метрические оценки, сильно ошибаются, а обычный ДСМ-метод отказывается классифицировать более чем 60%, а в случае несбалансированных классов (например, задача SPECT) даже 90% выборки. Большинство отказов в этом случае — это отказы по противоречию.

Задача Monk3. $|G_+| = 288, |G_-| = 266, |M| = 15$. Число положительных понятий: 719. Число отрицательных понятий: 553.

Характеристика	1	2	3	4	5	6	7	8
TPR	0.98	0.96	0.97	0.95	0.98	1	0.98	1
SPC	0.98	0.88	0.98	0.88	0.98	0.97	0.97	0.97
ACC	0.98	0.89	0.97	0.89	0.98	0.98	0.98	0.98
PPV	0.98	0.78	0.98	0.79	0.98	0.93	0.98	0.94
Доля отказов	0.03	0.15	0.08	0.18	0.12	0.41	0.22	0.52

Таблица 3. Задача Monk3, $\alpha = 0$

Характеристика	1	2	3	4	5	6	7	8
TPR	0.74	0.74	0.71	0.71	0.98	0.94	0.98	0.94
SPC	0.99	0.95	0.99	0.94	0.95	0.95	0.35	0.35
ACC	0.78	0.78	0.75	0.75	0.97	0.94	0.97	0.92
PPV	1	0.99	1	0.99	0.99	0.98	0.99	0.98
Доля отказов	0.08	0.08	0.06	0.06	0.78	0.87	0.81	0.89

Таблица 4. Задача Monk3, $\alpha = 0.05$

Характеристика	1	2	3	4	5	6	7	8
TPR	0.61	0.61	0.61	0.61	0.55	0.55	0.55	0.55
SPC	0.98	0.98	0.98	0.98	0.4	0.4	0.4	0.4
ACC	0.64	0.64	0.64	0.64	0.56	0.56	0.56	0.56
PPV	1	1	1	1	0.6	0.6	0.6	0.6
Доля отказов	0.04	0.04	0.04	0.04	0.95	0.95	0.95	0.95

Таблица 5. Задача Monk3, $\alpha = 0.1$

Задача SPECT. $|G_+| = 272, |G_-| = 70, |M| = 22$. Число положительных понятий: 3355. Число отрицательных понятий: 88.

Характеристика	1	2	3	4	5	6	7	8
TPR	0.64	0.6	0.23	0.22	0.69	0.64	0.26	0.22
SPC	0.76	0.71	0.62	0.64	0.78	0.72	0.62	0.62
ACC	0.68	0.66	0.37	0.36	0.72	0.69	0.41	0.37
PPV	0.57	0.45	0.79	0.67	0.60	0.44	0.8	0.64
Доля отказов	0.48	0.47	0.33	0.33	0.52	0.53	0.5	0.53

Таблица 6. Задача SPECT, $\alpha = 0$

Характеристика	1	2	3	4	5	6	7	8
TPR	0.45	0.43	0.43	0.43	0.6	0.4	0.6	0.4
SPC	0.72	0.71	0.71	0.69	0.6	0.6	0.6	0.6
ACC	0.61	0.6	0.6	0.58	0.6	0.6	0.6	0.6
PPV	0.44	0.43	0.42	0.42	0.6	0.6	0.6	0.6
Доля отказов	0.2	0.2	0.2	0.2	0.78	0.93	0.81	0.89

Таблица 7. Задача SPECT, $\alpha = 0.05$

Характеристика	1	2	3	4	5	6	7	8
TPR	0.57	0.57	0.57	0.57	0	0	0	0
SPC	0.86	0.86	0.86	0.86	0	0	0	0
ACC	0.69	0.69	0.69	0.69	0	0	0	0
PPV	0.77	0.77	0.77	0.77	0	0	0	0
Доля отказов	0.45	0.45	0.45	0.45	0.99	0.99	0.99	0.99

Таблица 8. Задача SPECT, $\alpha = 0.1$

Задача Voting records. $|G_+| = 162, |G_-| = 186, |M| = 16$. Число положительных понятий: 475. Число отрицательных понятий: 1232.

Характеристика	1	2	3	4	5	6	7	8
TPR	0.96	0.96	0.55	0.56	0.95	0.95	0.62	0.63
SPC	0.95	0.92	0.93	0.88	1	0.97	0.4	0.78
ACC	0.96	0.93	0.68	0.70	0.99	0.97	0.66	0.68
PPV	0.91	0.81	0.92	0.82	1	0.89	1	0.93
Доля отказов	0.45	0.42	0.40	0.40	0.49	0.48	0.69	0.72

Таблица 9. Задача Voting records, $\alpha = 0$

Характеристика	1	2	3	4	5	6	7	8
TPR	0.99	0.99	0.99	0.99	0.98	0.98	0.98	0.98
SPC	0.74	0.7	0.75	0.69	0.99	0.99	0.99	0.99
ACC	0.81	0.76	0.81	0.75	0.98	0.98	0.98	0.98
PPV	0.6	0.47	0.61	0.47	0.99	0.97	0.99	0.97
Доля отказов	0.04	0.04	0.04	0.04	0.34	0.4	0.56	0.62

Таблица 10. Задача Voting records, $\alpha = 0.05$

Характеристика	1	2	3	4	5	6	7	8
TPR	1	1	1	1	1	1	1	1
SPC	0.77	0.77	0.77	0.77	0.99	0.99	0.59	0.59
ACC	0.82	0.82	0.82	0.82	0.99	0.99	0.99	0.99
PPV	0.58	0.57	0.59	0.58	0.96	0.96	0.96	0.96
Доля отказов	0.09	0.09	0.09	0.1	0.52	0.54	0.7	0.72

Таблица 11. Задача Voting records, $\alpha = 0.1$

- Обычный ДСМ-метод имеет лучшие показатели TPR и SPC, по сравнению с голосованием с использованием метрических оценок. Это говорит о том, что его лучше применять в задачах с большой ценой ошибки первого рода.

- Во многих экспериментах при увеличении порога α и гипотезы, и генераторы выдают одинаковую классификацию (во многих случаях это связано с тем, что минимальные гипотезы и являются своими минимальными генераторами).

- Использование схемы вида "гипотезы против генераторов" имеет смысл в задачах с несбалансированными классами.

- Классификация с помощью минимальных генераторов требует ещё больших затрат и в среднем не улучшает показателей качества, при этом обычно имея большее число отказов. Это говорит о том, что в большинстве задач использование стандартной схемы "гипотезы против гипотез" предпочтительнее.

- Число генераторов для минимальных гипотез очень велико, однако среди них могут быть хорошие классификаторы, отличные от минимальных генераторов.

6 Заключение

В рамках данного исследования получены следующие результаты:

- На основе проведённого анализа существующих подходов к распознаванию на основе АФП предложена новая модель классификаторов, совмещающая подход АФП с метрическим подходом к классификации. В отличие от рассмотренных методов предложенные алгоритмы используют как метрическую информацию из исходного признакового пространства, так и объектно-признаковые порядковые зависимости. Показано, что предложенная модель классификаторов включает в себя некоторые из рассмотренных алгоритмов как частные случаи. Проведено экспериментальное сравнение данной модели с уже существующими алгоритмами классификации на основе АФП.

- На произвольной конечной решётке введено семейство расстояний. Доказано (см. утверждение 3), что любой элемент этого семейства является псевдометри-

кой, для которой получена формула в терминах множества $D(x)$ (см. утверждение 4). Также показано (см. следствие 1), что в случае конечных булевых алгебр это семейство содержит расстояние Хэмминга. Указаны значения параметров, при которых получаемая псевдометрика имеет понятную интерпретацию в терминах формальных понятий. Предложены способы модификации данной меры расстояния для учёта особенностей решётки формальных понятий и варианты её использования для модификации классификаторов на основе АФП.

- С использованием предложенной модели классификаторов было экспериментально исследовано пространство версий между минимальными α-гипотезами и их минимальными генераторами.

Перспективными вопросами для дальнейших исследований являются:

- Более детальное исследование алгоритмов классификации, получаемых при конкретном выборе системы опорных понятий $\mathcal{C}$ и меры близости $S(x, C)$.

- Формальное описание и изучение процедур классификации с использованием введённой на понятиях псевдометрики.

- Исследование возможности выбора системы $\mathcal{C}$ так, чтобы избежать построения всей решётки понятий целиком, а строить непосредственно элементы этой системы. Например, этого можно добиться путём введения более “сильных” операторов замыкания.

- Разработку алгоритмов, позволяющих эффективно “перемещаться” по пространству версий между минимальными гипотезами и минимальными генераторами, исключая порождение всех генераторов (или всего пространства версий целиком).

По теме исследований написана статья, с которой автор выступал на семинаре “What can FCA do for Artificial Intelligence?” FCA4AI, проведённом на конференции European Conference on Artificial Intelligence (ECAI 2014). Статья опубликована в сборнике трудов конференции [18] и в сборнике “Прикладная Математика и информатика” [19].

Список литературы

1. M. Kaytoue, S.O. Kuznetsov, A. Napoli, S. Duplessis. Mining gene expression data with pattern structures in formal concept analysis. Information Sciences, Volume 181, Issue 10, 15 May 2011, pp. 1989-2001, Information Science, 2011.

2. S.O. Kuznetsov. Complexity of Learning in Concept Lattices from Positive and Negative Examples. Discrete Applied Mathematics, 2004, No. 142(1–3), pp. 111-125.

3. S.O. Kuznetsov. Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: P. Maji, A. Ghosh, M.N. Murty, K. Ghosh, S.K. Pal, Eds., Proc. 5th International Conference Pattern Recognition and Machine Intelligence (PReMI'2013), Lecture Notes in Computer Science (Springer), Vol. 8251, pp. 30-41, 2013.

4. С.И. Гуров: Булевы алгебры, упорядоченные множества, решётки: определения, свойства, примеры. — Москва, 2013.

5. Г. Гретцер: Общая теория решёток. — М.: Мир, 1982.

6. С.О. Кузнецов: Теория решёток для интеллектуального анализа данных.

7. Lieve Lambrechts: Formal Concept Analysis. — Vrije Universiteit Brussel, 2011.

8. M. Sahami: Learning classification Rules Using Lattices. N. Lavrac and S. Wrobel eds., pp. 343–346, Proc ECML, Heraclion, Crete, Greece (Avril 1995).

9. C. Caprineto, G. Romano: GALOIS An order-theoretic approach to conceptual clustering. In proceedings of ICML93, pp. 33–40, Amherst, USA (July 1993).

10. Zhipeng Xie, Wynne Hsu, Zongtian Liu, Mong Li Lee: Concept Lattice based Composite Classifiers for high Predictability. Artificial Intelligence, vol. 139, pp. 253–267, Wollongong, Australia (2002).

11. O. Prokasheva, A. Onishchenko, S. Gurov. Classification methods based on Formal Concept Analysis. FCAIR 2013 – Formal Concept Analysis Meets Information Retrieval. Workshop co-located with the 35th European Conference on Information

Retrieval (ECIR 2013). March 24, 2013, Moscow, Russia. National Research University Higher School of Economics, pp. 95-104. ISSN 1613-0073

12. Ю.И. Журавлёв: Об алгебраическом подходе к решению задач распознавания или классификации. — Проблемы кибернетики: — 1978. — Т.33. — С.5-68.

13. T. Mitchell, Version Space: An Approach to Concept Learning, PhD thesis, Stanford University, 1978.

14. M. Maddouri: Towards a machine learning approach based on incremental concept formation. Intelligent Data Analysis, Volume 8, Issue 3, pp. 267–280 (2004).

15. Dan A. Simovici: Betweenness, Metrics and Entropies in Lattices. Proceedings of the 38th International Symposium on Multiple Valued Logic. 22-24 May, 2008, Dallas, TX, USA. IEEE Computer Society Washington, pp. 26-31. ISSN 0195-623X.

16. К.В. Воронцов: Машинное обучение (курс лекций).

17. Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

18. Evgeny Kolmakov, Metric Generalization and Modification of Classification Algorithms Based on Formal Concept Analysis. In: Sergei O. Kuznetsov, Amedeo Napoli, Sebastian Rudolph, Eds., Proc. 3rd Workshop "What can FCA do for Artificial Intelligence?"(FCA4AI 2014), CEUR Workshop Proceedings, Vol. 1257, pp. 43-50, 2014.

19. Е.А. Колмаков, Метрическое обобщение алгоритмов классификации на основе анализа формальных понятий, Сборник “Прикладная Математика и информатика”, серия “Труды факультета ВМК МГУ им. М.В. Ломоносова”, МАКС Пресс Москва, том 47, с. 122-136 (2014).