

Московский государственный университет имени М.В. Ломоносова

Факультет Вычислительной Математики и Кибернетики

Кафедра Математических Методов Прогнозирования

Янаков Дмитрий Спартакович

**Анализ формальных понятий в задаче классификации по
прецедентам**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

д.ф.-м.н., доцент

Дюкова Елена Всеволодовна

Научный консультант:

аспирант ФИЦ ИУ РАН

Масляков Глеб Олегович

Москва, 2023

Содержание

1	Введение	2
2	Направления CVP и FCA. Классический случай	4
2.1	Процедуры корректного голосования (CVP)	4
2.2	Анализ формальных понятий (FCA)	5
3	Направления CVP и FCA. Случай частично упорядоченных данных	8
3.1	Задача поиска максимальных независимых элементов произведения частичных порядков	8
3.2	Процедуры корректного голосования	9
3.3	Анализ формальных понятий	10
3.3.1	Обобщение основных понятий	10
3.3.2	Сведение к классическому случаю	12
4	Алгоритм поиска формальных понятий Close By One	13
5	ДСМ-классификатор	15
5.1	Обучение и распознавание	15
5.2	Связь CVP и FCA	16
6	Экспериментальное исследование	17
6.1	Модельные данные	17
6.2	Реальные данные	18
7	Заключение	22
	Список литературы	23

1 Введение

Одной из основных задач машинного обучения является классификация по прецедентам. Задача ставится следующим образом.

Исследуется некоторое множество объектов M . Известно, что M представимо в виде объединения l непересекающихся подмножеств K_1, \dots, K_l , называемых классами. Объекты множества M описываются признаками x_1, \dots, x_n . Имеется конечный набор объектов $S_1, \dots, S_m \in M$, о которых известно, каким классам они принадлежат. Эти объекты называются прецедентами или обучающими объектами, а их описания имеют вид $S_i = (a_{i1}, \dots, a_{in})$, где a_{ij} – значение признака x_j для объекта S_i . Требуется по предъявленному набору значений признаков (b_1, \dots, b_n) , описывающему некоторый объект из M , о котором, вообще говоря, неизвестно какому классу он принадлежит, определить (распознать) этот класс.

Для решения этой задачи успешно применяется аппарат дискретной математики. Главным преимуществом рассматриваемого подхода, известного как логический или дискретный, является получение результата без использования дополнительных предположений вероятностного характера и при небольшом количестве прецедентов. Логический подход включает три основных направления: Correct Voting Procedures (CVP), Formal Concept Analysis (FCA) и Logical Analysis of Data (LAD).

В настоящей работе подробно описывается направление FCA. Выявляется связь между FCA и CVP.

Идеи, лежащие в основе FCA, были предложены Рудольфом Вилле в начале 1980-х годов [24]. Данное направление изучает, как объекты можно группировать иерархически с учетом их общих признаков. Для математической формализации FCA использует теорию множеств и теорию решеток. В 1976 году В.К. Финн предложил метод автоматического порождения гипотез или ДСМ-метод (как инструмент для формализации правдоподобных и достоверных выводов) [12]. В дальнейшем ДСМ-метод был адаптирован для решения задачи классификации с бинарными данными, представленными положительными и отрицательными примерами.

В России методы FCA развиты в работах С.О. Кузнецова, Д.И. Игнатова, М.И. Забежайло [20, 21, 22, 23]. За рубежом идеи FCA излагаются в работах Гантера [16, 17].

В CVP [3, 8], как и в других направлениях логического подхода, центральными являются вопросы построения моделей классификаторов, которые безошибочно распознают обучающие объекты, т.е. являются корректными. Фундаментальную роль в создании отечественных методов CVP сыграли работы С.В. Яблонского, в которых введено хорошо известное в дискретной математике понятие теста [13], и работы Ю.И. Журавлева, опубликованные в 70-х и 80-х годах прошлого века. Основы проблематики заложены также в статьях российских ученых М.М. Бонгарда и М.Н. Вайнцвайга, в которых описывался распознающий алгоритм «Кора» [1]. В дальнейшем это направление в основном развивалось в работах Ю.И. Журавлева, Е.В. Дюковой, Н.В. Пескова, П.А. Прокофьева, Г.О. Маслякова и др. [7, 15].

Довольно часто встречаются задачи классификации, в которых каждый признак принимает значения из некоторого конечного частично упорядоченного множества. В последнее время появились методы направления CVP, позволяющие работать с такими данными [6, 7].

Одним из основных недостатков ДСМ-классификатора является низкое качество классификации ввиду слишком строгой процедуры распознавания. В настоящей работе предложена модификация данной процедуры, позволяющая повысить точность классификации за счет изменения решающего правила. Модифицированный ДСМ-классификатор с решающим правилом, применяемым в процедурах CVP, обобщен на случай частично упорядоченных данных. С использованием алгоритма *Close By One* [23] реализованы на C++ классический ДСМ-классификатор и его модификация, а также модель ДСМ-классификатора, ориентированная на работу с данными, представленными в виде декартова произведения линейных порядков. Тестирование построенных классификаторов проведено на модельных и реальных данных.

2 Направления CVP и FCA. Классический случай

В данном разделе введены основные понятия логического анализа данных без частичных порядков.

2.1 Процедуры корректного голосования (CVP)

Будем предполагать, что множество допустимых значений каждого признака состоит из целых чисел и это множество конечно.

Определение 1. Пусть H – набор из r различных признаков вида $H = \{x_{j_1}, \dots, x_{j_r}\}$, $\sigma = (\sigma_1, \dots, \sigma_r)$, σ_i – допустимое значение признака x_{j_i} , $i = \overline{1, r}$. Пару (σ, H) назовем *элементарным классификатором (эл.кл.)*.

Определение 2. Близость объекта $S = (a_1, \dots, a_r) \in M$ и эл.кл. (σ, H) , $\sigma = (\sigma_1, \dots, \sigma_r)$, $H = \{x_{j_1}, \dots, x_{j_r}\}$, будем оценивать величиной $B(\sigma, S, H)$, определяемой следующим образом:

$$B(\sigma, S, H) = \begin{cases} 1, & \text{если } a_{j_t} = \sigma_t \text{ при } t = \overline{1, r}, \\ 0, & \text{иначе.} \end{cases}$$

Если $B(\sigma, S, H) = 1$, то будем говорить, что объект S содержит эл.кл. (σ, H) .

Определение 3. Эл.кл. (σ, H) является *корректным для класса K* , если нельзя указать пару обучающих объектов S' и S'' : $S' \in K$, $S'' \notin K$ и $B(\sigma, S', H) = B(\sigma, S'', H) = 1$.

Определение 4. Пусть $\overline{K} = \{K_1, \dots, K_l\} \setminus K$. Эл.кл. (σ, H) называется *представительным для класса K* , если ни один обучающий объект из \overline{K} не содержит (σ, H) и хотя бы один обучающий объект из K содержит (σ, H) .

Определение 5. Представительный эл.кл. (σ, H) для класса K называется *тупиковым*, если не является представительным для K любой эл.кл. вида (σ', H') , где $\sigma' \subset \sigma$, $H' \subset H$.

В классической модели алгоритма голосования по тупиковым представительным эл.кл. для каждого класса K строится множество тупиковых представительных

эл.кл., обозначаемое далее через $\mathcal{T}(K)$. Для нахождения $\mathcal{T}(K)$ используются алгоритмы монотонной дуализации, например, асимптотически оптимальные алгоритмы, впервые предложенные Е. В. Дюковой [4]. Эти алгоритмы на сегодняшний день являются лидерами по скорости счета. Задача монотонной дуализации относится к числу труднорешаемых задач дискретной математики и формулируется как построение сокращенной дизъюнктивной нормальной формы монотонной булевой функции, заданной конъюнктивной нормальной формой. Труднорешаемость дуализации обусловлена экспоненциальным ростом числа решений с ростом размера задачи и сложностью нахождения каждого нового решения.

Распознавание объекта S осуществляется на основе процедуры голосования. Для этого вычисляется оценка принадлежности объекта S классу K по следующей формуле:

$$G(S, K) = \frac{1}{|\mathcal{T}(K)|} \sum_{(\sigma, H) \in \mathcal{T}(K)} P_{(\sigma, H)} B(\sigma, S, H),$$

где $P_{(\sigma, H)}$ – вес эл.кл. (σ, H) . В качестве $P_{(\sigma, H)}$ обычно берется число обучающих объектов из K , содержащих (σ, H) .

Объект S относится к классу с наибольшей оценкой. Если таких классов несколько, то происходит отказ от классификации.

2.2 Анализ формальных понятий (FCA)

Определение 6. *Формальным контекстом* называется тройка вида $C = (M, X, I)$, где M – множество объектов, X – множество признаков, I – бинарное отношение между множествами M и X . Запись вида $(S, x) \in I$ означает, что объект $S \in M$ обладает признаком $x \in X$.

Другими словами, формальный контекст – это булева матрица L , строками которой являются признаковые описания объектов. Если объект S обладает признаком x , то соответствующий элемент матрицы L равен 1, в противном случае он равен 0.

Определение 7. Для множества $A \subseteq M$, положим $A' = \{x \in X | (S, x) \in I \ \forall S \in A\}$. Аналогично, для множества $B \subseteq X$, положим $B' = \{S \in M | (S, x) \in I \ \forall x \in B\}$. Полученные множества A' и B' называются *операторами вывода* для формального

контекста $C = (M, X, I)$. Полагаем, что для $\emptyset \subseteq M$ выполняется $\emptyset' = X$ и для $\emptyset \subseteq X$ выполняется $\emptyset' = M$.

То есть, A' – это оператор, возвращающий все столбцы матрицы L , которые в пересечении с заданными строками образуют подматрицу, все элементы которой равны 1. Аналогично, B' – это оператор, возвращающий все строки матрицы L , которые в пересечении с заданными столбцами образуют подматрицу, все элементы которой равны 1.

На рис. 1 приведен пример формального контекста с четырьмя геометрическими фигурами и четырьмя признаками.





№	$G \backslash X$	x_1	x_2	x_3	x_4
S_1		1	0	0	1
S_2		1	0	1	0
S_3		0	1	1	0
S_4		0	1	1	1

Рис. 1: Контекст геометрических фигур

Признак x_1 отвечает за то, что фигура имеет 3 угла, x_2 – 4 угла, x_3 – фигура содержит прямой угол, x_4 – фигура правильная.

В случае использования операторов вывода в данном контексте, можно получить, к примеру, следующие результаты:

- $\{S_1, S_4\}' = \{x_4\}$;
- $\{x_2, x_3\}' = \{S_3, S_4\}$;
- $\{S_1, S_2, S_3\}' = \emptyset$;

Определение 8. Пусть дан контекст $C = (M, X, I)$. Пара вида (A, B) , где $A \subseteq M$, $B \subseteq X$, такая, что $A' = B$, $B' = A$, называется *формальным понятием* данного контекста с *объемом* A и *содержанием* B .

Заметим, что пара (A, B) образует «максимальную» подматрицу матрицы L , все элементы которой равны 1. Причём число строк или столбцов в полученной подматрице нельзя увеличить.

В случае рассмотренного выше примера, формальными понятиями, например, являются:

- $(\{S_1\}, \{x_1, x_4\})$;
- $(\{S_3, S_4\}, \{x_2, x_3\})$;
- $(\{S_2, S_3, S_4\}, \{x_3\})$.

3 Направления CVP и FCA. Случай частично упорядоченных данных

В данном разделе введены основные понятия логического анализа данных с частичными порядками для CVP, а также обобщены понятия для FCA.

3.1 Задача поиска максимальных независимых элементов произведения частичных порядков

Сформулируем одну из центральных задач логического анализа данных с частичными порядками, а именно, задачу поиска максимальных независимых элементов произведения частичных порядков.

Пусть $M = N_1 \times \dots \times N_n$, где N_i , $i \in 1, 2, \dots, n$, – конечное множество значений признака x_i , на котором задан частичный порядок. Считается, что элемент $y = (y_1, \dots, y_n) \in M$ *следует* за элементом $z = (z_1, \dots, z_n) \in M$, если y_i следует за z_i при $i = \overline{1, n}$. Для обозначения того, что $y \in M$ следует за $z \in M$ далее используется запись $z \preceq y$ или $y \succeq z$. Запись $z \prec y$ ($y \succ z$) означает, что $z \not\preceq y$ и $y \neq z$. Элементы $z, y \in M$ называются *сравнимыми*, если либо $z \preceq y$, либо $y \succeq z$. В противном случае z и y называются *несравнимыми*.

Пусть $R \subseteq P$. Введём обозначения: $R^+ = R \cup \{x \in P \mid \exists a \in R, a \prec x\}$ – множество элементов, следующих за элементами из R .

Элемент x множества $P \setminus R^+$ называется *максимальным независимым* от R элементом множества P , если для любого другого элемента y множества $P \setminus R^+$ отношение $x \prec y$ не выполняется.

Обозначим через $I(R^+)$ множество, состоящее из максимальных независимых от R элементов множества P . Ставится задача построения для заданного R множества $I(R^+)$.

Одним из наиболее востребованных и изученных является случай, когда множества P_1, \dots, P_n – цепи, т.е. в каждом из этих множеств любые два элемента сравнимы. Для данного случая разработан алгоритм RUNC-M+ [6], который ищет специальные покрытия булевой матрицы, соответствующие максимальным независимым элементам.

Далее будем считать, что каждое множество N_i , $i \in \{1, 2, \dots, n\}$, имеет наибольший элемент, т.е. такой элемент k_i , для которого выполнено $a \preceq k_i$ для любого $a \in N_i$. Если наибольший элемент в N_i отсутствует, то N_i дополним таким элементом.

3.2 Процедуры корректного голосования

Определение 9. Близость объекта $S = (a_1, \dots, a_n)$ из M и эл.кл. (σ, H) , $H = \{x_{j_1}, \dots, x_{j_r}\}$, $\sigma = (\sigma_1, \dots, \sigma_r)$, $\sigma_i \in N_{j_i}$, при $i = \overline{1, r}$, будем оценивать величиной $\tilde{B}(\sigma, S, H)$, определяемой следующим образом:

$$\tilde{B}(\sigma, S, H) = \begin{cases} 1, & \text{если } a_{j_t} \preceq \sigma_t \text{ при } t = \overline{1, r}, \\ 0, & \text{иначе.} \end{cases}$$

Будем говорить, что объект S порождает эл.кл. (σ, H) , если $\tilde{B}(\sigma, S, H) = 1$.

Данные в разделе 2.1 понятия корректного эл.кл. класса K , представительного эл.кл. класса K полностью переносятся на рассматриваемый случай, если $B(\sigma, S, H)$ заменить на $\tilde{B}(\sigma, S, H)$.

Пусть (σ, H) — эл.кл., в котором $H = \{x_{j_1}, \dots, x_{j_r}\}$, $\sigma = (\sigma_1, \dots, \sigma_r)$, $\sigma_i \in N_{j_i}$, $i = \overline{1, r}$. Эл.кл. (σ, H) сопоставим набор $S_{(\sigma, H)} = (\gamma_1, \dots, \gamma_n)$ из $M = N_1 \times \dots \times N_n$, определяемый следующим образом:

$$\gamma_t = \begin{cases} \sigma_i, & \text{при } t = j_i \ (i = \overline{1, r}), \\ k_t, & t \notin \{j_1, \dots, j_r\}. \end{cases}$$

Определение 10. Представительный для класса K эл.кл. (σ, H) назовём *тупиковым*, если любой эл.кл. (σ', H') такой, что $S_{(\sigma, H)} \preceq S_{(\sigma', H')}$, не является представительным для класса K .

Обозначим через $R(K)$ — множество прецедентов класса K , а через $R(\overline{K})$ — множество прецедентов, не принадлежащих классу K . То есть $R(\overline{K}) = \{S_1, \dots, S_m\} \setminus R(K)$.

Справедливым является следующее утверждение [7].

Утверждение 1. Эл.кл. (σ, H) является тупиковым представительным для класса K относительно заданного частичного порядка тогда и только тогда, когда $S_{(\sigma, H)} \in I(R(\overline{K})^+)$ и $S_{(\sigma, H)} \in R(K)^+$.

Из утверждения 1 следует, что в случае частично упорядоченных данных при построении логических классификаторов, основанных на голосовании по тупиковым представительным эл.кл. класса, возникает задача построения максимальных независимых элементов произведения частичных порядков.

Заметим, что если описание обучающего объекта S класса K следует за описанием некоторого объекта из \overline{K} , то, очевидно, S не порождает ни одного представительного эл.кл. класса K . Поэтому в общем случае существование представительных для класса K эл.кл. не гарантировано.

Можно показать, что в случае непересекающихся классов существует такое преобразование признакового описания множества M , в результате которого для каждого класса K всегда образуется непустое множество тупиковых представительных эл.кл. и любой прецедент из K порождает хотя бы один эл.кл. из данного множества [7]. Приведем это отображение.

Обозначим через \tilde{P} множество, совпадающее с множеством P , но с обратным отношением порядка, т.е. $x \preceq y$ в P тогда и только тогда, когда $y \preceq x$ в \tilde{P} .

Пусть $\tilde{M} = \tilde{N}_1 \times \dots \times \tilde{N}_n$. Зададим отображение $\phi : M \rightarrow M \times \tilde{M}$ следующим образом. Отображение ϕ переводит объект $S = (a_1, \dots, a_n)$ из M в объект $\phi(S) = (a_1, \dots, a_n, a_{n+1}, \dots, a_{2n})$ из $M \times \tilde{M}$, в котором $a_{n+i} = a_i$ при $i \in \{1, 2, \dots, n\}$. Иными словами, признаковое описание объекта S дублируется с обратным отношением порядка.

3.3 Анализ формальных понятий

3.3.1 Обобщение основных понятий

Приведем понятия, аналогичные тем, которые введены в разделе 2.2, учитывая, что на множестве значений каждого признака задан частичный порядок.

В данном случае *формальный контекст* C — это уже необязательно булева матрица L . Как следствие, пропадает бинарное отношение I , поэтому формальный контекст будем обозначать через $C = (M, X)$.

Определение 11.

- Для множества $A \subseteq M$, положим $A' = (\sigma, H)$:

1. $\tilde{B}(\sigma, S, H) = 1 \quad \forall S \in A;$
2. $S_{(\sigma, H)} \preceq S_{(\sigma', H')} \quad \forall (\sigma', H') : \tilde{B}(\sigma', S, H') = 1 \quad \forall S \in A.$

Причём из получившегося эл.кл. (σ, H) отбрасываются признаки x_{j_i} и соответствующие им значения σ_i : $\sigma_i = k_i$.

- Для эл.кл. $B = (\sigma, H)$, положим $B' = A \subseteq M$:

1. $\tilde{B}(\sigma, S, H) = 1 \quad \forall S \in A;$
2. $|A| = \max_{A' : \tilde{B}(\sigma, S, H)=1 \quad \forall S \in A'} |A'|$, где $|A|$ - мощность множества $|A|$.

- Полученные эл.кл. A' и множество B' называются *операторами вывода* для формального контекста $C = (M, X)$. Через (\emptyset) обозначим эл.кл., в котором нет ни одного признака, т.е. $H = \emptyset$. Полагаем, что для $\emptyset \subseteq M$ выполняется $\emptyset' = (\emptyset)$ и для эл.кл. (\emptyset) выполняется $(\emptyset)' = M$.

Проще говоря, A' – это оператор, возвращающий такой эл.кл. (σ, H) , у которого σ_i равно максимальному значению признака x_{j_i} среди объектов из множества A . Если $\sigma_i = k_i$, то такой признак отбрасывается из получившегося эл.кл.. B' – это оператор, возвращающий максимальное по мощности множество объектов A , порождающих эл.кл. $B = (\sigma, H)$.

Определение 12. Пусть дан контекст $C = (M, X)$. Пара вида (A, B) , где $A \subseteq M$, $B = (\sigma, H)$, такая, что $A' = B$, $B' = A$, называется *формальным понятием* данного контекста с *объемом* A и *эл.кл.* B .

Пусть $H = \{x_{j_1}, \dots, x_{j_r}\}$, $S \in M$. Обозначим через S_H - признаковое описание объекта S , из которого отброшены признаки из $X \setminus H$.

Пара (A, B) , где $A \subseteq M, B = (\sigma, H)$ образует подматрицу P матрицы L :

1. $S_H \preceq \sigma \quad \forall S \in A$, причем неравенство неверно $\forall(\sigma', H') : H' \supseteq H, \sigma' \preceq \sigma$ (в случае, если $|H'| > |H|$, σ дополняется максимальными элементами в соответствующих позициях);
2. Любая строка S из $M \setminus A$, добавленная к подматрице P , не удовлетворяет неравенству $S_H \preceq \sigma$.

3.3.2 Сведение к классическому случаю

Рассмотрим признак $x_i \in X$, который принимает значения из $N_i = \{a_{i1}, \dots, a_{in}\}$, причем $a_{i1} \prec a_{i2} \prec \dots \prec a_{in}$. Тогда введем следующую кодировку для признака x_i :

- $a_{i1} \rightarrow \underbrace{1\dots 11}_{n-1};$
- $a_{i2} \rightarrow \underbrace{1\dots 10}_{n-2};$
- ...
- $a_{in} \rightarrow \underbrace{0\dots 00}_{n-1}.$

Таким образом, 1 целочисленный признак заменяется на $n - 1$ бинарных.

Рассмотрим формальные контексты $C = (M, X)$ и $C' = (M', X', I)$, где C' получен из C с помощью применения кодировки. Тогда, несложно убедиться в том, что любое формальное понятие контекста C' взаимно однозначно сопоставляется с некоторым формальным понятием контекста C .

4 Алгоритм поиска формальных понятий Close By One

Рассмотрим один из алгоритмов нахождения всех формальных понятий формального контекста — *Close By One*, который был предложен Кузнецовым С.О. в [22]. С помощью него будут проведены дальнейшие эксперименты. Считается, что данные бинарные.

Данный алгоритм предполагает, что на множестве объектов задан лексикографический порядок. То есть, объект $S_i \prec S_j$, если $i < j$.

Один шаг алгоритма *Close By One* соответствует операции «Замыкай-по-одному-вниз» (*CbODown*), определяемой следующим образом:

$$CbODown((A, B), S) = ((A \cup \{S\})'', B \cap \{S\}')$$

Корректность алгоритма обеспечивается тем, что для любого формального понятия (A, B) и любого объекта $S \in M$, пара $CbODown((A, B), S)$ также является формальным понятием [2].

Если V — множество всех формальных понятий контекста, то сложность алгоритма — $\mathcal{O}(|M|^2|X||V|)$, а полиномиальная задержка — $\mathcal{O}(|M|^3|X|)$.

Псевдокоды алгоритма *Close By One* и вспомогательной рекурсивной процедуры *Process* приведены ниже.

Algorithm 1 Close By One

Input: $C = (M, X, I)$ — формальный контекст

Output: V — множество формальных понятий

```
1:  $V := \emptyset$ 
2: for  $S \in M$  do
3:    $\text{Process}(\{S\}, S, (S'', S'))$ 
4: end for
```

Algorithm 2 $\text{Process}((A, S, (C, D)))$, причем $C = A''$, $D = A'$

```
1: if  $\{H \in C \setminus A \mid H \prec S\} = \emptyset$  then  
2:    $V := V \cup \{(C, D)\}$   
3:   for  $F \in \{H \in M \mid S \prec H\}$  do  
4:      $Z := C \cup \{F\}$   
5:      $Y := D \cap \{F\}'$   
6:      $X := Y'$   
7:      $\text{Process}(Z, F, (X, Y))$   
8:   end for  
9: end if
```

Рассмотрим работу алгоритма более подробно:

1. к каждому объекту S из M применяется оператор вывода $'$, далее к S' применяется еще один оператор вывода, затем выполняется рекурсивная процедура Process ([алгоритм 1](#), шаг 3);
2. при входе в процедуру Process происходит так называемая проверка на "каноничность" генерируемого формального понятия, которая заключается в следующем: множество объектов $C \setminus A$ не должно содержать объекты, которые лексикографически меньше, чем объект S ([алгоритм 2](#), шаг 1);
3. если проверка на каноничность выполнена, то сгенерированное формальное понятие (C, D) добавляется в множество V ([алгоритм 2](#), шаг 2), иначе происходит выход из процедуры Process ;
4. далее каждый объект H , который лексикографически больше, чем S добавляется к множеству объектов C ([алгоритм 2](#), шаг 4), затем выполняется операция $CboDown((C, D), H)$ ([алгоритм 2](#), шаги 5 и 6);
5. после этого происходит вызов процедуры Process с новыми параметрами ([алгоритм 2](#), шаг 7).

Фактически, происходит полный перебор формальных понятий с уменьшением лишних действий за счет проверки на "каноничность". Она отсекает многие повторные генерации формальных понятий, которые уже были сгенерированы.

5 ДСМ-классификатор

ДСМ-метод изначально был сформулирован в терминах математической логики, однако позже была установлена связь между ДСМ-гипотезами и формальными понятиями [21].

Определение 13.

- Классический случай. Формальное понятие (A, B) формального контекста $C = (M, X, I)$ порождает эл.кл. вида (σ, B) , где все элементы σ равны 1. Полученный эл.кл. называется *гипотезой класса K* , если он является представительным для класса K .
- Случай частично упорядоченных данных. Формальное понятие (A, B) формального контекста $C = (M, X)$ порождает *гипотезу класса K* , если B — представительный эл.кл. для класса K .

Далее в работе будем считать, что рассматриваются бинарные данные. Случай частичных порядков будем сводить к классическому случаю с помощью кодировки, описанной в [разделе 3.3.2](#).

5.1 Обучение и распознавание

На этапе обучения ДСМ-классификатора необходимо найти все формальные понятия формального контекста $(R(K), X, I)$, которые порождают гипотезы класса K , а также все формальные понятия формального контекста $(R(\overline{K}), X, I)$, которые порождают гипотезы класса \overline{K} , где $\overline{K} = K \setminus \{K_1, \dots, K_l\}$.

Классификация распознаваемого объекта S происходит при помощи построенных гипотез. Для этого в ДСМ-классификаторе используется следующая процедура:

- если S содержит хотя бы одну гипотезу класса K и не содержит ни одну гипотезу класса \overline{K} то S **относится к классу K** ;
- если S содержит хотя бы одну гипотезу класса \overline{K} и не содержит ни одну гипотезу класса K то S **не относится к классу K** ;

- если S содержит как гипотезу класса K , так и \overline{K} , или если S не содержит ни одну гипотезу, то происходит **отказ от классификации**.

Стоит отметить, что на практике данная процедура приводит к большому числу отказов, что снижает качество классификации. Поэтому была предложена следующая процедура распознавания.

Пусть объект S содержит m гипотез класса K и n гипотез класса \overline{K} . Тогда:

- если $m > n$, то S **относится к классу K** ;
- если $m < n$, то S **не относится к классу K** ;
- если $m = n$ то происходит **отказ от классификации**.

5.2 Связь CVP и FCA

Как следует из [определения гипотезы](#), она является некоторым специальным представительным эл.кл., поскольку порождается из формального понятия.

Из этого следует, что при решении задач классификации в случае частично упорядоченных данных, необходимо дублировать признаковое описание объектов с обратным отношением порядка, чтобы гарантировать существование непустого множества гипотез для каждого класса.

6 Экспериментальное исследование

6.1 Модельные данные

Изначально была рассмотрена задача нахождения всех формальных понятий формального контекста, которая решалась, как уже было отмечено, с помощью алгоритма *Close By One*, реализованного на языке программирования C++.

Целью экспериментов было выявить зависимость времени работы алгоритма от числа строк/столбцов булевой матрицы из дискретного равномерного распределения, а также подсчитать число получаемых формальных понятий. Результаты вычислений, усредненные по десяти запускам, изображены на графиках ниже. Для наглядности, эксперименты проведены на квадратных и прямоугольных матрицах.

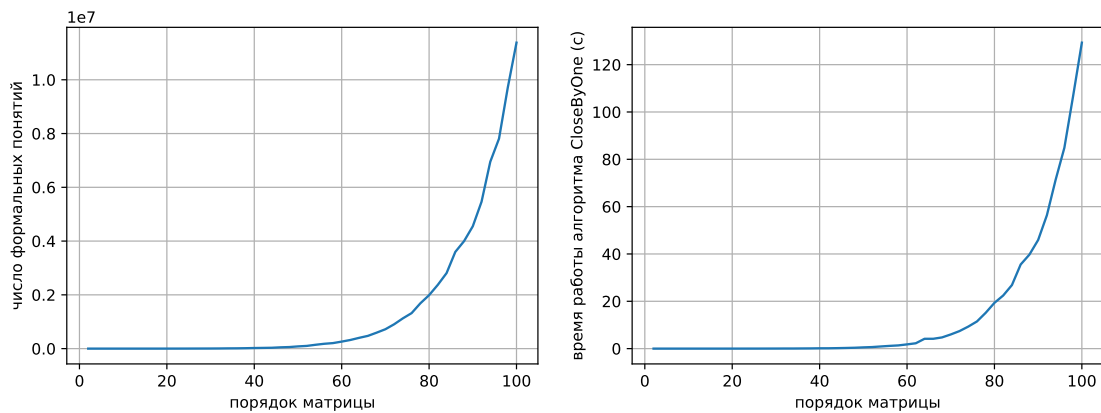


Рис. 2: Эксперименты на квадратных матрицах

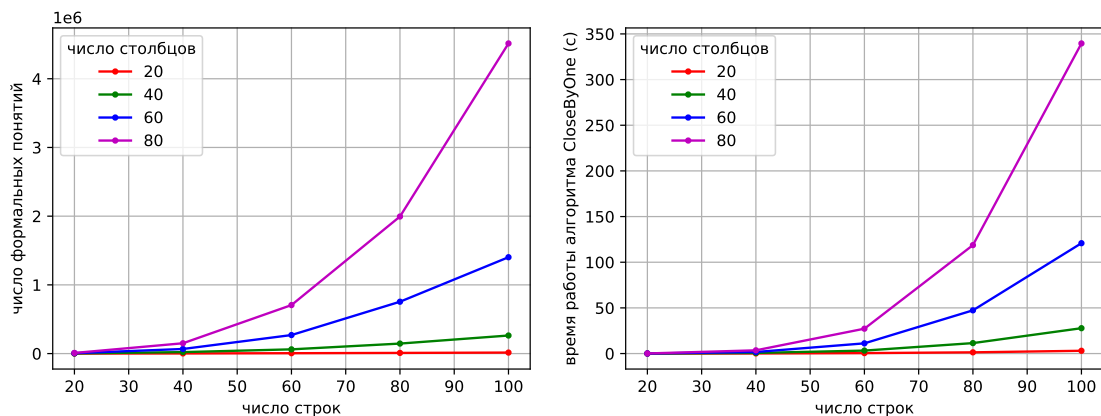


Рис. 3: Эксперименты на прямоугольных матрицах

Из графиков можно сделать следующие выводы:

- рост числа строк сильнее увеличивает время работы алгоритма, чем рост числа столбцов;
- время работы алгоритма зависит от числа столбцов линейно;
- время работы алгоритма зависит от числа строк кубически;
- число формальных понятий растёт пропорционально времени работы алгоритма.

Очевидно, что ДСМ-классификатор, использующий данный алгоритм для нахождения гипотез, будет быстрее работать при числе строк меньшем, чем число столбцов. Поэтому в случае, если число строк больше, чем число столбцов, происходит следующая процедура: вместо нахождения формальных понятий (A, B) исходной матрицы, ищутся формальные понятия вида (B, A) транспонированной матрицы, после чего в них меняются местами множества A и B .

6.2 Реальные данные

Эксперименты проводились на реальных данных из репозитория UCI [14]: дата-сетах «Шахматы» (Ш), «Крестики-нолики» (КН), «Машины» (М), «Детские сады» (ДС); и из репозитория ВЦ РАН [9]: «Инсульты» (И), «Неорганические соединения» (НС), «Остеогенная саркома» (ОС).

Основной целью было сравнить по времени и по качеству классический ДСМ-классификатор (обозначение — $ДСМ$) с предложенными модификациями: первая из которых позволяет увеличить точность классификации ($ДСМ-Г$), а вторая из которых позволяет работать с частично упорядоченными данными ($ДСМ+$). Также была рассмотрена модель, являющаяся комбинацией $ДСМ-Г$ и $ДСМ+$ — $ДСМ-Г+$, сочетающая в себе работу с частичными порядками и модифицированную процедуру распознавания. Помимо этого, в сравнении участвовали алгоритм голосования по всем тупиковым представительным эл.кл. в классическом случае (A_1) и в случае частично упорядоченных данных (A_1+), а также классические алгоритмы машинного обучения: логистическая регрессия (LR), случайный лес (RF) и градиентный бустинг (GB), запущенные на параметрах по умолчанию.

В качестве метрики качества была выбрана *ROC-AUC*.

Поскольку во многих задачах признаки целочисленные, то применялись следующие преобразования данных:

- в *ДСМ* и *ДСМ+* к данным применялось *OneHot*-кодирование (обозначение — *ОНЕ*);
- в *ДСМ-Г* и *ДСМ-Г+* к данным применялась кодировка, описанная в [разделе 3.3.2 \(COD\)](#).

Для алгоритмов, работающих с частичными порядками, исходные данные предварительно преобразовывались с помощью быстрой процедуры независимого линейного упорядочивания признаков [5].

Размеры задач приведены в таблице ниже. Выборки делились в следующем отношении: 80% — обучающая, 20% — тестовая.

Датасет	Количество признаков			Количество объектов	
	Изначально	После ОНЕ	После COD	Обучающая выборка	Тестовая выборка
М	6	21	15	1382	346
КН	9	27	18	766	192
ДС	8	21	19	10368	2592
И	81	81	81	63	16
ОС	19	566	547	213	54
НС	35	379	344	116	29
Ш	36	73	37	2556	640

Таблица 1: Размерности данных

Результаты экспериментов, усреднённые по десяти запускам, при случайных разбиениях датасетов на обучающую и тестовую выборки, приведены в таблицах ниже.

Датасет	ДСМ	ДСМ+	ДСМ-Г	ДСМ-Г+	A ₁	A ₁ +	LR	RF	GB
М	0.885	0.989	0.919	0.995	0.947	0.976	0.791	0.998	0.999
КН	0.830	0.845	0.999	0.845	0.995	0.992	0.627	0.996	0.989
ДС	0.959	0.999	0.992	1.000	0.999	0.854	0.956	1.000	1.000
И	0.510	0.662	0.666	0.684	0.657	0.607	0.792	0.750	0.701
ОС	0.492	0.958	0.566	0.977	0.644	0.864	0.690	0.697	0.670
НС	0.699	0.789	0.862	0.945	0.883	-	0.909	0.916	0.902
Ш	0.772	0.523	0.886	0.523	0.982	0.903	0.993	0.999	0.998

Таблица 2: Среднее значение точности (ROC-AUC)

Датасет	ДСМ	ДСМ+	ДСМ-Г	ДСМ-Г+	A ₁	A ₁ +	LR	RF	GB
М	0.12		0.07		0.02	0.44	< 0.01	0.12	0.10
КН	0.55		0.74		0.06	1.43	< 0.01	0.12	0.08
ДС	33.89		9.59		2.12	18.14	0.04	0.30	0.49
И	3.92		13.60		140.13	175.92	< 0.01	0.09	0.05
ОС	0.42		2969.18		0.04	325.28	0.01	0.10	0.07
НС	0.69		283.57		0.10	-	0.01	0.09	0.06
Ш	489.43		4.42		6.50	12.38	0.05	0.17	0.22

Таблица 3: Среднее значение времени (секунды)

Для алгоритма A₁ + не удалось получить результат на задаче «Неорганические соединения» ввиду слишком долгой работы программы.

Из проведенных экспериментов можно сделать следующие выводы по качеству классификации:

- на датасете «Машины» алгоритм ДСМ-Г+ показал практически такое же качество, как и градиентный бустинг, у которого лучший результат. Поэтому в данном случае алгоритмы можно считать сравнимыми по качеству;
- на датасете «Крестики-нолики» ДСМ-Г показал наилучшую точность среди всех тестируемых алгоритмов. В данной задаче случайный лес сравним с ним по качеству;

- на датасете «Детские сады» ДСМ-Г+ вместе со случайным лесом и градиентным бустингом показали идеальное качество;
- на датасете «Инсульты» наилучший показатель у логистической регрессии;
- на датасете «Остеогенная саркома» лидером является ДСМ-Г+. Он опередил следующий по качеству алгоритм A_1+ на 13%. В данной задаче введение частичных порядков позволило существенно увеличить точность классификации;
- на датасете «Неорганические соединения» лучший показатель точности также у ДСМ-Г+. В данном случае он опередил второй по качеству алгоритм — случайный лес на 3%;
- на датасете «Шахматы» случайный лес показал наилучший результат. Для ДСМ-классификаторов результаты получились хуже;
- введение частичных порядков не всегда приводит к увеличению качества: на задаче «Крестики-нолики» качество ДСМ-Г+ меньше, чем у ДСМ-Г.

Говоря про скорость работы алгоритмов, победителем является логистическая регрессия. ДСМ-классификаторы в основном работают за приемлемое время, однако на некоторых задачах могут отрабатывать достаточно долго. Связано это, в первую очередь, с большим количеством генерируемых гипотез. Например, на задаче «Остеогенная саркома» при введении частичных порядков время увеличилось почти на час, когда общее количество гипотез увеличилось с 5058 до 3907241. К тому же, на скорость работы ДСМ-классификаторов, как было показано в экспериментах на модельных данных, влияет размерность задачи: чем она больше, тем дольше время выполнения.

Таким образом, в результате проведенных экспериментов было установлено, что модификации ДСМ-Г и ДСМ-Г+ позволяют увеличить качество классификации по сравнению с классическим ДСМ-классификатором. Они также показывают наилучшее качество на некоторых задачах. Однако, при больших размерностях задач, ДСМ-классификаторы работают дольше, чем другие алгоритмы.

7 Заключение

В настоящей работе подробно рассмотрен подход анализа формальных понятий к решению задачи классификации по прецедентам. Предложена модификация классического ДСМ-классификатора, которая позволяет увеличить точность классификации за счёт применения для положительных гипотез процедуры голосования. Данная модификация обобщена на случай, когда данные представлены в виде декартова произведения частичных порядков. На языке программирования C++ реализованы полученные модели. Для них проведено экспериментальное исследование на случайных данных различной размерности и на реальных данных. В тестировании к тому же участвовали алгоритм голосования по всем тупиковым представительным эл.кл. и его обобщение на случай частично упорядоченных данных, а также логистическая регрессия, случайный лес и градиентный бустинг. Эксперименты показали, что предложенные классификаторы показывают достаточно хорошее качество на большинстве задач, однако при больших размерностях данных могут работать дольше других тестируемых алгоритмов.

Список литературы

- [1] Вайнцвайг М.Н. Алгоритм обучения распознаванию образов «Кора» // Алгоритмы обучения распознаванию образов. М.: Сов. Радио. 1973. С. 82–91.
- [2] Виноградов Д.В. Вероятностно-комбинаторный подход к автоматическому порождению гипотез // В кн.: Гуманитарные чтения. — М.: РГГУ, 2015. С. 771–775
- [3] Дюкова Е.В., Журавлёв Ю.И., Рудаков К.В. Об алгебраическом синтезе корректирующих процедур распознавания на базе элементарных алгоритмов // Ж. вычисл. матем. и матем. физ. 1996. Т. 36, №8. С. 215–223.
- [4] Дюкова Е. В., Инякин А. С. Об асимптотически оптимальном построении тупиковых покрытий целочисленной матрицы // Математические вопросы кибернетики. Вып. 17 — М.: Физматлит, 2008. С. 247–262.
- [5] Дюкова Е.В., Масляков Г.О. О выборе частичных порядков на множествах значений признаков в задаче классификации // Информатика и её применения, 2021. Т. 15. № 4. С. 72–78.
- [6] Дюкова Е.В., Масляков Г.О., Прокофьев П.А. О дуализации над произведением частичных порядков // Ж. машинное обучение и анализ данных, 2017. Т. 3, № 4. С. 239– 249
- [7] Дюкова Е.В., Масляков Г.О., Прокофьев П.А. О логическом анализе данных с частичными порядками в задаче классификации по прецедентам // Ж. вычисл. матем. и матем. физ., 2019. Т. 59, № 9. С. 1605– 1616
- [8] Дюкова Е. В., Песков Н. В. Построение распознающих процедур на базе элементарных классификаторов // Математические вопросы кибернетики. Вып. 14. — М.: Физматлит, 2005. С. 57– 92.
- [9] Журавлев Ю.И., Рязанов В. В., Сенько О. В. Распознавание. Математические методы. Программная система. Практические применения. — М.: ФАЗИС, 2006. 176 с.

- [10] Масич И.С. Поисковые алгоритмы решения задач условной псевдобулевой оптимизации // Ж. системы управления, связи и безопасности, 2016. №1, С. 1605–1616
- [11] Рязанов В. В. Логические закономерности в задачах распознавания (параметрический подход) // Ж. вычисл. матем. и матем. физ. 2007. Т. 47, №10. С. 1793–1808.
- [12] Финн В. К. О возможности формализации правдоподобных рассуждений средствами многозначных логик // Всесоюзн. симп. по логике и методологии науки. — Киев: Наукова думка, 1976. С. 82–83.
- [13] Чегис И. А., Яблонский С. В. Логические способы контроля электрических схем // Труды математического института им. В. А. Стеклова АН СССР. 1958. Т. 51.С. 270–360.
- [14] Asuncion A., Newman D. 2007. UCI machine learning repository. <https://archive.ics.uci.edu/ml/index.php>
- [15] Djukova E.V, Masliakov G.O., Correct classification over a product of partial orders // IEEE Proceedings of the VII International Conference on Information Technology and Nanotechnology, Samara, Russia, 2021. P. 1–5.
- [16] Ganter B., Kuznetsov S.O., Formalizing hypotheses with concepts. // In Proceedings 8th International Conference on Conceptual Structures, ICCS 2000, Lecture Notes in Artificial Intelligence, 1867, Darmstadt, Germany, pp. 342–356.
- [17] Ganter, B., Wille, R., Formal Concept Analysis: Mathematical Foundations (Heidelberg: Springer), 1999
- [18] Hammer P.L. Partially defined boolean functions and cause-effect relationships. // International Conference on Multi-attribute Decision Making Via OR-based Expert Systems. University of Passau. Passau. Germany, April, 1986.
- [19] Hammer P., Bonates T., Kogan A., Maximum patterns in datasets. Discrete Applied Mathematics. 2008, Vol. 156(6), P. 846–861.
- [20] Karpov N., Ignatov D.I., Braslavski P., Information Retrieval // 8th Russian Summer School, RuSSIR 2014, Nizhniy, Novgorod, Russia, August 18–22, 2014, Revised Selected Papers

- [21] Kuznetsov S.O., Mathematical aspects of concept analysis // Journal of Mathematical Science, 1996, Vol. 80, Issue 2, pp. 1654–1698.
- [22] Kuznetsov S.O., A fast algorithm for computing all intersections of objects from an arbitrary semilattice // Nauchno-Tekhnicheskaya Informatsiya. Seriya 2 — Informatsionnye protsessy i sistemy, 1993, No. 1, pp.17–20.
- [23] Kuznetsov S.O. and Obiedkov S.A., Comparing Performance of Algorithms for Generating Concept Lattices // Journal of Experimental and Theoretical Artificial Intelligence, 2002, Vol. 14, no. 2–3, pp. 189–216. Intelligence, 2002, Vol. 14, no. 2–3, pp. 189–216.
- [24] R. Wille, Restructuring lattice theory: An approach based on hierarchies of concepts // Rival, I., Ed., Ordered Sets: Proceedings. NATO Advanced Studies Institute, 83, Reidel, Dordrecht. 1982, 445–470.