УДК 004.93

ВЕРОЯТНОСТНЫЕ ФОРМАЛЬНЫЕ ПОНЯТИЯ В НЕКОТОРЫХ ЗАДАЧАХ КЛАССИФИКАЦИИ

Е.Е. Витяев[1], В.В. Мартынович[2]

Институт математики им. С.Л. Соболева СО РАН, Новосибирск, Россия
[1] *vityaev@math.nsc.ru,* [2] *vilco@yandex.ru*

Аннотация

Рассматривается определение формальных понятий как неподвижных точек импликаций. На основе этого определения водится понятие вероятностных формальных понятий путем замены импликаций на специальные максимально специфические вероятностные правила, для которых ранее было доказано, что неподвижные точки для них логически непротиворечивы. Определяется алгоритм *ProbClosure* обнаружения вероятностных формальных понятий. Для разработки алгоритмов кластеризации и классификации контекст рассматривается как выборка из генеральной совокупности. Обобщая алгоритм *ProbClosure*, определяются алгоритмы кластеризации *ConcClosure* и *StatClosure* путем введения различных функционалов энергии, определяющих степень непротиворечивости правил в неподвижной точке. Алгоритмы классификации получаются путем применения алгоритмов кластеризации к новым данным. Проведено сравнение полученных алгоритмов классификации с решающими деревьями C4.5, ID3 и методом классификации, основанным на решётке формальных понятий. Сравнение проведено на данных репозитория UCI. Полученные результаты показали сравнительно большую точность разработанных алгоритмов по сравнению с указанными методами.

Ключевые слова: *анализ формальных понятий, вероятность, ассоциативные правила, классификация.*

Цитирование: *Витяев, Е.Е.* Вероятностные формальные понятия в некоторых задачах классификации / Е.Е. Витяев, В.В. Мартынович // Онтология проектирования. – 2017. – Т. 7, №4(26). - С. 473-486. – DOI: 10.18287/2223-9537-2017-7-4-473-486.

Введение

Анализ формальных понятий (АФП) [1] содержит в себе удобный инструментарий для представления и обработки различных данных. Формальные понятия образуют целостные объединения групп свойств и объектов и поэтому являются очевидными кандидатами на классификационные единицы. Это позволяет использовать их при решении задач кластеризации и классификации.

В рамках АФП изучается весь спектр задач анализа данных. За последние годы опубликованы работы по извлечению паттернов атрибутов [2, 3], работы по алгоритмам кластеризации и извлечению ассоциативных правил, предпринят ряд попыток построения алгоритмов классификации и др. [4-7]. Для нас особый интерес представляет работа [7], где предложен алгоритм построения решающих деревьев на основе решетки формальных понятий, а результаты экспериментов представлены в виде таблицы с измерениями точности работы алгоритмов. Это позволяет сравнить точность предлагаемых алгоритмов классификации, разработанных на основе вероятностных формальных понятий и соответствующего алгоритма кластеризации с результатами этих алгоритмов.

Раздел 1 предлагаемой статьи посвящен определению вероятностных и статистических формальных понятий. В разделе 2 приводятся основные алгоритмы их построения: *Prob-*

Closure для вероятностных, *StatClosure* для статистических формальных понятий. Центральным можно считать раздел 3, где рассматриваются основные практические модификации процедуры *StatClosure* – алгоритмы *ClassifyInCluster, ClassifyOverClusters,* - а также различные параметризации этих алгоритмов, позволяющие придать предлагаемому подходу требуемую гибкость. В разделе 4 приводятся результаты классификации, полученные этими алгоритмами. Протоколы экспериментов и сравнение с близкими по семантике методами классификации даются в разделе 5. В качестве опорных рассматриваются результаты, полученные в [7] с помощью построения решающих деревьев на основе решетки формальных понятий, что позволяет более наглядно показать эффективность методов *ClassifyInCluster* и *ClassifyOverClusters*.

1 Вероятностные и статистические формальные понятия

Напомним базовые определения АФП [1].

***Определение* 1**. Формальный контекст – $K = (G, M, I)$, где G – множество объектов, M – множество атрибутов и $I \subseteq G \times M$ – отношение принадлежности атрибутов объектам.

***Определение* 2**. $A \subseteq G$, $B \subseteq M$. Тогда:

- $A^{\uparrow} = \{m \in M \mid \forall g \in A, (g, m) \in I\}$;
- $B^{\downarrow} = \{g \in G \mid \forall m \in B, (g, m) \in I\}$;
- (A, B) – формальное понятие, если $A^{\uparrow} = B$ и $B^{\downarrow} = A$.

***Определение* 3**. $R = (B, C)$ – импликация, $R \in \Re = \mathrm{Imp}(K)$, если $B^{\downarrow} \subseteq C^{\downarrow}$ и $B, C \subseteq M$. При этом $B = R^{\leftarrow}$ называется посылкой, а $C = R^{\rightarrow}$ – заключением импликации. Оператор логического вывода, использующий множество импликаций $\Re$, добавляет к некоторому множеству атрибутов L другие, выводимые из него атрибуты:

$$\Pi_{\Re}(L) = L \cup \{C \mid \exists R \in \Re : R^{\leftarrow} \subseteq L, R^{\rightarrow} = C\}.$$

Вероятностное обобщение формальных понятий можно получить [8], опираясь на следующий результат.

***Теорема* 1** [1]. Множество неподвижных точек оператора логического вывода совпадает с множеством формальных понятий. Для любого множества $B \subseteq M$, $\Pi_{\Re}(\mathrm{B}) = \mathrm{B} \Leftrightarrow \mathrm{B}^{\downarrow\uparrow} = \mathrm{B}$.

Вероятностные формальные понятия мы получим как неподвижные точки соответсвующего вероятностного оператора логического вывода. Для его определения построим логико-вероятностную модель, описывающую формальный контекст K.

***Определение* 4**. Для конечного контекста $K = (G, M, I)$ определим сигнатуру контекста σ_K, содержащую лишь множество предикатных символов, совпадающее с M. Для сигнатуры σ_K и контекста K как модели определим интерпретацию предикатных символов следующим образом: $K \vDash m(x) \Leftrightarrow (x, m) \in I$.

***Определение* 5**. Определим классические логические конструкции:

1) $\mathrm{Term}(K)$ – множество термов состоит из символов переменных;
2) $\mathrm{At}(K)$ – атомами являются выражения $m(t)$, где $m \in \sigma_K$ и $t \in Term(K)$;
3) $\mathrm{Lit}(K)$ – литеры включают все атомы $m(t)$ и их отрицания $\neg m(t)$;
4) $\mathrm{For}(K)$ – определяется индуктивно: всякий атом – формула, и для любых $\Phi, \Psi \in \mathrm{For}(K)$ синтаксические конструкции $\Phi \wedge \Psi, \Phi \vee \Psi, \Phi \rightarrow \Psi, \neg\Phi$ - тоже формулы.

***Определение* 6**. Рассмотрим произвольную вероятностную меру μ на множестве G, определенную в колмогоровском смысле. Определим контекстную вероятностную меру на множестве формул как:

$$\nu : \mathrm{For}(K) \to [0,1], \nu(\phi) = \mu(\{g \in G \mid g \vDash \phi\}).$$

Определим правила на контексте, как аналог импликаций, а также их составные части.

***Определение* 7**. Пусть $C, H_i \in \mathrm{Lit}(K)$, $C \notin \{H_1, H_2, ...H_k\}, k \geq 0$, тогда:

1) *Правило* $R = (H_1, H_2 ..., H_k \to C)$ есть импликация $(H_1 \wedge H_2 ... \wedge H_k \to C)$;
2) Посылкой $R^{\leftarrow}$ правила R называется набор литер $\{H_1, H_2 ..., H_k\}$;
3) Заключением правила является $R^{\rightarrow} = C$;
4) Длиной правила назовём мощность его посылки $|R^{\leftarrow}|$;
5) Если $R_1^{\leftarrow} = R_2^{\leftarrow}$ и $R_1^{\rightarrow} = R_2^{\rightarrow}$, то $R_1 = R_2$.

***Определение* 8**. Вероятностью правила R является значение

$$\eta(R) = \nu(R^{\rightarrow} \mid R^{\leftarrow}) = \frac{\nu(R^{\leftarrow} \wedge R^{\rightarrow})}{\nu(R^{\leftarrow})}.$$

Если знаменатель $\nu(R^{\leftarrow})$ равен 0, то вероятность правила неопределена.

***Определение* 9**. Правило R назовем максимально специфичным $R \in \mathrm{MSR}(K)$, если нет правила $\tilde{R}$ с более длинной посылкой $R^{\leftarrow} \subset \tilde{R}^{\leftarrow}$ и более высокой вероятностью $\eta(\tilde{R}) > \eta(R)$.

Правила определения 7 позволяют установить вероятностный оператор замыкания. Для этого заменим множество импликаций $Imp(K)$ на множество максимально специфических вероятностных правил. Поэтому ниже будем предполагать, что $\Re$ – множество максимально специфичных правил. По аналогии с теоремой 1 определим вероятностные формальные понятия как неподвижные точки оператора логического вывода, использующего множество правил $\Re$.

***Определение* 10**. Замыканием $\bar{L}$ множества литер L будем называть наименьшую неподвижную точку оператора логического вывода, содержащую L:

$$\bar{L} = \Pi_{\Re}(\bar{L}) = \Pi_{\Re}^{\infty}(L) = \bigcup_{k \in \mathrm{N}} \Pi_{\Re}^{k}(L).$$

***Определение* 11**. Пусть $\Re \subset \mathrm{MSR}(K)$ – множество максимально специфических правил. Тогда B – вероятностное формальное понятие, если $\Pi_{\Re}(B) = B$.

***Теорема* 2** [9]. Пусть $\Re$ – множество максимально специфических правил, тогда: если L непротиворечиво, то $\Pi_{\Re}(L)$ также непротиворечиво.

На основе определения 11 и теоремы 2 нетрудно предложить алгоритм замыкания *ProbClosure*, который для заданного множества литер B строит замыкание $\bar{B}$, являющееся минимальной неподвижной точкой, содержащей множество B, и, в силу определения 11, вероятностным формальным понятием. Алгоритм *ProbClosure* не требует разрешения противоречий, так как в силу теоремы 2 исключается ситуация, когда в процессе вывода обнаруживается одновременно литера и ее отрицание.

Алгоритм 1. *ProbClosure*. Замыкание набора литер оператором вывода.
Вход: $\Re \subseteq \mathrm{For}(K)$, $K = (G, M, I)$, $B \subseteq \mathrm{Lit}(K)$
Выход: $C \subseteq \mathrm{Lit}(K)$ – вероятностное формальное понятие

1: **Функция** `ProbClosure`$(K, \mathfrak{R}, B)$
2: $B_0 \leftarrow B$
3: $k \leftarrow 0$
4: **Повторять**
5: $B_{k+1} \leftarrow \Pi_{\mathfrak{R}}(B_k)$
6: $k \leftarrow k+1$
7: **До тех пор пока** $B_k \neq B_{k-1}$
8: **Вернуть** B_k
9: **Конец функции**

В практических задачах контекст полностью неизвестен, а известна только некоторая выборка из контекста. Адекватной моделью данных, применяемой в большинстве методов машинного обучения [10], можно считать следующую:

- источник данных e – многомерная случайная величина с заданным распределением;
- обучающая выборка $G_{teach} = \{(g_{(1)}, ..., g_{(n)})\}$ – выборка из генеральной совокупности, где $g_{(i)}$ попарно независимые случайные величины с распределением e.

Это означает, что моделью наблюдаемого контекста $K = (G_{teach}, M, I)$ является выборка из генеральной совокупности $K^* = (G, M, I)$, где каждый $g \in G_{teach}$ представлен многомерной бернуллиевской случайной величиной. Однако, задача классификации должна по-прежду пониматься в смысле исходного контекста K^*, образующего генеральную совокупность объектов. В таких условиях непротиворечивость логического вывода с помощью $\Pi_{\mathfrak{R}}$ может быть нарушена, поскольку максимально специфические правила, извлеченные из наблюдаемого контекста K, зачастую не будут являться таковыми по отношению к истиному контексту K^*.

Решить проблему противоречивости логического вывода возможно и в этом случае. Рассмотрим общий процесс преобразования набора литер. Пусть исходное множество литер $B = B_1$ проходит через цепочку преобразований $B_1, ..., B_n$ (такие преобразования происходят со стартовым множеством B в алгоритме *ProbClosure*). Предположим, что для алгоритма преобразования наборов литер существует некий критерий ϕ, минимизация которого определяет направление поиска в пространстве всех означиваний литер $B \in 2^{\mathrm{Lit}(K)}$. Такие алгоритмы очень удобны в вычислительном плане, поскольку позволяют определить процедуру минимизация итеративно и свести исходную задачу к задаче минимизации.

Для процесса преобразований конфигураций верно, что если первый и последний наборы совпадают $B_1 = B_n$, то он определяет тождественное преобразование, и тогда для критерия ϕ должно выполнятся определяемое ниже условие.

***Определение* 12**. Условие потенциальности

$$B_1 = B_n \Rightarrow \sum_{i=1,...,n-1} \phi(B_i, B_{i+1}) = 0 .$$

Функционал ϕ является аналогом физического потенциала. Условие в определении 12 является условием независимости потенциала от пути его вычисления, а, как известно, потенциал позволяет определить функцию энергии. Заметим, что идея введения функционала энергии не нова, в [11] она подробно изучена в контексте механизма обратной связи для глубинных нейронных сетей.

***Теорема* 3** [11]. Критерий ϕ может быть выражен с помощью потенциальной энергии E: $\phi(B,C) = E(C) - E(B)$; при этом $\phi(B,C)$ удовлетворяет условию потенциальности, а значение потенциала не зависит от точки начала отсчета энергии.

Зафиксируем некоторое множество правил $\Re$. Далее будем считать что все правила R берутся из этого универсума правил $\Re$.

***Определение* 13**. Пусть R – правило, а $B \subset \mathrm{Lit}(K)$.

- R применимо (или $R \in \mathrm{App}(B)$) к набору литер B, если $R^{\leftarrow} \subset B$.
- R подтверждается (или $R \in \mathrm{Sat}(B)$) на наборе B, если $R \in \mathrm{App}(B)$, и $R^{\rightarrow} \in B$.
- R опровергается (или $R \in \mathrm{Fal}(B)$) на наборе B, если $R \in \mathrm{App}(B)$, и $\neg R^{\rightarrow} \in B$.

***Определение* 14**. Энергией противоречий мы называем функционал энергии, определенный с помощью веса опровергающихся правил, за вычетом энергии подтверждающихся правил:

$$E(B) = \sum_{R \in \mathrm{Fal}(B)} \gamma(\mathrm{R}) - \sum_{R \in \mathrm{Sat}(B)} \gamma(\mathrm{R}),\ \gamma(R): \Re \to [0, \infty),\ \phi(B, \varnothing) = E(B).$$

Задача семейства алгоритмов состоит в том, чтобы минимизировать энергию противоречий $E(B) \to \mathbf{min}$, и, таким образом, найти максимально непротиворечивые комбинации литер (заметим, что можно также показать, что при наличии множества максимально специфичных правил алгоритм дает и абсолютно непротиворечивые комбинации литер, совпадающие с вероятностными формальными понятиями). Однако, полное решение задачи минимизации функционала энергии выглядит как полный перебор в пространстве означиваний литер $2^{\mathrm{Lit}(K)}$.

Вспомним, что мы работаем в вероятностном контексте, где точного решения исходной задачи классификации не требуется, а приемлемость решения определяется иными способами (например, предсказательной точностью классификатора - *Accuracy*). Поэтому абсолютной точностью при решении задачи минимизации функционала энергии можно пренебречь, а поставленная вычислительная проблема может быть решена субоптимальным образом. Предлагается вычислять приближенные решения посредством «жадного» итеративного алгоритма *StatClosure*, который минимизирует потенциал и выполняет поиск локально оптимальных решений соотношения $E(B) \to \mathbf{min}$. Свойство жадности опирается на то предположение, что для субоптимальности достаточно рассмотреть только потенциал перехода к ближайшим соседям, т.е. от конфигурации B к конфигурациям вида $B \pm l$, где $l \in \mathrm{Lit}(K)$.

Алгоритм 2. ***StatClosure***. Замыкание набора литер статистическим оператором вывода.

Вход: $\Re \subseteq \mathrm{For}(K)$, $K = (G, M, I)$, $B \subseteq \mathrm{Lit}(K)$

Выход: $C \subseteq \mathrm{Lit}(K)$ – статистическое формальное понятие

1: **Функция** `StatClosure`$(K, \Re, B)$
2: $B_0 \leftarrow B$
3: $k \leftarrow 0$
4: **Повторять**
5: $k \leftarrow k + 1$
6: $B_k \leftarrow B_{k-1}$
7: $\psi \leftarrow 0$
8: Candidates $\leftarrow \varnothing$

9: **Для всех** $L \in \mathrm{Lit}(K) \setminus (B_{k-1} \cup \neg B_{k-1})$ **выполнять**
10: Candidates $\leftarrow$ Candidates $\cup \{B_{k-1} \cup L\}$
11: **Для всех** $L \in B_{k-1}$ **выполнять**
12: Candidates $\leftarrow$ Candidates $\cup \{B_{k-1} \setminus L\}$
13: **Для всех** $C \in$ Candidates **выполнять**
14: $\alpha \leftarrow \phi(B_{k-1}, C)$
15: **Если** $\alpha < \psi$ **тогда**
16: $\psi \leftarrow \alpha$
17: $B_k \leftarrow C$
18: **Конец условия**
19: **Конец цикла**
20: **До тех пор пока** $\psi < 0$
21: **Вернуть** B_k
22: **Конец функции**

Полученные алгоритмом 2 неподвижные точки локально минимизируют функционал энергии противоречий, уменьшают количество противоречий до минимально возможного, а потому частично решают проблему противоречивости логического вывода.

2 Алгоритмы классификации

Рассмотрим некоторые модификации алгоритма *StatClosure*. Он не является точным обобщением алгоритма вероятностного замыкания *ProbClosure*. Чтобы понять, в чём отличие, рассмотрим момент добавления литеры в содержание понятия. Пусть к B_{s-1} была добавлена литера L_s, в результате чего получилось множество литер $B_s = B_{s-1} + L_s$. Тогда $\phi(B_{s-1}, B_s)$ из процедуры статистического замыкания представляет собой комбинацию γ-весов из сумм по следующим группам правил:

1) $R \in \mathrm{Sat}(B_s)$ и $R^{\rightarrow} = L_s$, то есть набор B_s подтверждает заключение правила R;
2) $R \in \mathrm{Fal}(B_s)$ и $R^{\rightarrow} = \neg L_s$, то есть B_s опровергает заключение правила R;
3) $R \in \mathrm{Sat}(B_s)$ и $L_s \in R^{\leftarrow}$, т.е. L_s делает посылку правила верной;
4) $R \in \mathrm{Fal}(B_s)$ и $L_s \in R^{\leftarrow}$, т.е. L_s делает посылку верной и при этом R не верно на B_s.

***Определение* 15**. Обозначим правила, возникающие при рассмотрении литер-кандидатов на добавление к основному множеству литер B (первого и второго типов из перечисления выше) следующим образом:

- $\mathrm{ConcSat}(B, L) = \mathrm{App}(B) \cap \{R : R^{\rightarrow} = L\}$;
- $\mathrm{ConcFal}(B, L) = \mathrm{App}(B) \cap \{R : R^{\rightarrow} = \neg L\}$;
- $\mathrm{PreSat}(B, L) = \mathrm{Sat}(B + L) \cap \{R : L \in R^{\leftarrow}\}$;
- $\mathrm{PreFal}(B, L) = \mathrm{Fal}(B + L) \cap \{R : L \in R^{\leftarrow}\}$.

Оператор замыкания из *ProbClosure* устроен таким образом, что использует только правила типов ConcSat. Однако в случае возникновения противоречий необходимо также учитывать и ConcFal. Поэтому модификация алгоритма *StatClosure*, наиболее близкая к

ProbClosure и учитывающая ConcSat и ConcFal, приводит к следующему потенциалу энергии противоречий:

$$\varphi(B_{s-1}, B_{s-1} \cup \{L\}) = \sum_{R \in \text{ConcFal}(B_{s-1}, L)} \gamma(R) - \sum_{R \in \text{ConcSat}(B_{s-1}, L)} \gamma(R) .$$

***Замечание* 1**. Следует отметить, что отображение $\tilde{\phi}$ уже не будет потенциалом в смысле определения 12. Однако алгоритм 2 по-прежнему применим после замены $\phi \to \tilde{\phi}$. Модификацию алгоритма с учетом модификации потенциала непротиворечивости определим как *ConcConcepts*.

Рассмотрим ещё одну модификацию алгоритма *StatClosure*. Раз имеются два различных алгоритма *ConcClosure* и *StatClosure*, имеющих одинаковую природу, то можно использовать их композицию. Самой простой является линейная комбинация

$$\alpha \cdot StatConcepts + (1-\alpha) \cdot ConcConcepts .$$

Для этого мы смешиваем потенциалы:

$$\varphi_\alpha(B, B+L) = \alpha \cdot \varphi_{Stat}(B, B+L) + (1-\alpha) \cdot \varphi_{Conc}(B, B+L) =$$
$$\alpha \cdot [\sum_{\text{PreFal}(B,L)} \gamma(R) - \sum_{\text{PreSat}(B,L)} \gamma(R)] + \sum_{\text{ConcFal}(B,L)} \gamma(R) - \sum_{\text{ConcSat}(B,L)} \gamma(R)$$

***Определение* 16**. Параметр $\beta = 1/\alpha$ из формулы для φ_α назовем весом посылочных правил и обозначим как *PremiseFactor*.

Зачастую бывает желательно, чтобы проблема непротиворечивости решалась не только на уровне литер в описании понятия, но и на уровне правил, законов, которые это содержание описывают. В некоторых случаях допустим определённый уровень противоречий, определяемый количественным соотношением между подтверждающимися и опровергающимися правилами. В таком случае мы можем уменьшить вес правил, противоречащих добавлению литер. Это позволит добавлять в процессе выполнения процедуры замыкания больше литер, которые могут являться противоречивыми, но не более, чем того допускает выбранный уровень противоречивости w и, обратно, если требуется большая нетерпимость к противоречиям между различными правилами в логическом выводе, то уровень w следует увеличивать:

$$\varphi_w(B, B+L) =$$
$$w \cdot [\sum_{\text{PreFal}(B,L)} \gamma(R) + \sum_{\text{ConcFal}(B,L)} \gamma(R)] - [\sum_{\text{PreSat}(B,L)} \gamma(R) + \sum_{\text{ConcSat}(B,L)} \gamma(R)] =$$
$$w \cdot \sum_{\text{Fal}(B+L)} \gamma(R) - \sum_{\text{Sat}(B+L)} \gamma(R) .$$

***Определение* 17**. Параметр w из φ_w назовем весом противоречий.

Алгоритм *StatClosure*, а также его модификации с помощью веса посылочных правил и веса противоречий, могут успешно применяться для решения прикладных задач анализа данных аналогично тому, как для этого применяется АФП. Принципиальное отличие в том, что для применения алгоритма *StatClosure* и его модификаций не требуется безошибочность данных.

Рассмотрим применение алгоритма *StatClosure* и его модификаций к задачам классификации. Пусть $K = (G^T \cup G^C, M \cup \Lambda, I)$ есть контекст, представляющий собой выборку из генеральной совокупности, где G^T – множество объектов обучения, а G^C – множество объектов контроля, а Λ – множество атрибутов разметки, определяющих класс объекта. Считаем, что роль учителя сводится к разметке объектов и присвоению им метки из

множества классов Λ. Логику учителя можно сформулировать в виде отображения $Teach: G^T \to \Lambda$. Задача алгоритма классификации – доопределить отображение *Teach* на контрольном множестве G^C в контексте $K^C = (G^C, M, I \cap (G^C \times M))$.

На первом этапе выполняется процедура кластеризации, обнаруживающая множество всех статистических формальных понятий на контексте $K^T = (G^T, M \cup \Lambda, I \cap (G^T \times (M \cup \Lambda)))$ относительно одной из описанных выше вариаций *StatClosure*, которую мы условно обозначим за $\text{Closure}(\cdot)$:

$$\Omega = \left\{ \text{Closure}(g^{\uparrow}) \,\middle|\, g \in G^T \right\}.$$

Далее классифицируемый объект поступает на обработку в процедуру ClassifyInCluster, описанную в алгоритме 3.

Алгоритм 3. ***ClassifyInCluster***. Классификация объекта.

Вход: $g \in G^{Control}$, $\text{Closure}(\cdot)$, Ω

Выход: $c \subseteq \Lambda$ – разметка объекта g

1: **Функция** `ClassifyInCluster`$(g, \text{Closure}, \Omega)$
2: $c \leftarrow \varnothing$
3: $\text{B} \leftarrow \text{g}^{\uparrow}$
4: $\overline{B} \leftarrow \text{Closure}(B)$
5: **Если** $\overline{B} \in \Omega$ **тогда**
6: $c \leftarrow \overline{B} \cap \Lambda$
7: **Конец условия**
8: **Вернуть** c
9: **Конец функции**

Алгоритм 3 и приводимый далее алгоритм 4, дают решение задачи классификации любым из описанных выше вариаций алгоритма *StatClosure*.

В [12] предлагается другой подход к задаче классификации. В работе решается задача распознавания транскрипционных факторов в последовательности ДНК. Идея заключается в том, чтобы определить степень принадлежности классифицируемого объекта ко всему спектру из найденных классов (кластеров).

Обратимся к определению 11 вероятностного оператора замыкания. Как нетрудно заметить, вероятностное формальное понятие полностью определяется множеством правил, которые его описывают. Действительно, по описанию прототипа класса $B \subseteq \text{Lit}(K)$ можно найти уже знакомое нам множество правил $\text{Sat}(B)$. И обратно, по множеству правил $\Re$ можем построить прототип класса $B = \bigcup_{R \in \Re} (R^{\leftarrow} \cup R^{\rightarrow})$. Получаем эквивалентное определение вероятностных формальных понятий через импликативные взаимосвязи.

Такое определение даёт возможность построить оценку близости классифицируемого объекта g к классу B, аналогично методам нечёткой кластеризации. Для этого следует вычислить значение энергии $E(g^{\uparrow})$ относительно множеств правил $\text{Sat}(B)$ каждого из классов. Тогда оценки принадлежности к классу будут следующими:

$$\lambda_B(g^{\uparrow}) = \sum_{R \in \text{Fal}(g^{\uparrow}) \cap \text{Sat}(B)} \gamma(R) - \sum_{R \in \text{Sat}(g^{\uparrow}) \cap \text{Sat}(B)} \gamma(R).$$

Алгоритм 4 выбирает два наиболее походящих класса B_I и B_{II} и в случае существенного смещения оценок принадлежности, задаваемого параметром $\lambda_* \le \lambda_{B_I}(\cdot)/\lambda_{B_{II}}(\cdot)$, даётся ответ в зависимости от вхождения признаков разметки из Λ в описание класса B_I.

Алгоритм 4. ***ClassifyOverClusters***. Классификация объектов.

Вход: $g \in G^{Control}$, Closure(·), Ω

Выход: $c \subseteq \Lambda$ – разметка классов для объектов $g \in G^{Control}$

Функция `ClassifyOverClusters`$(K, \Re, \Omega)$
  $\lambda_{Best}, \lambda_{Second} \leftarrow 0$
  $B_{Best}, B_{Second} \leftarrow \perp$
  $X \leftarrow g^{\uparrow}$
  $\overline{X} \leftarrow \text{Closure}(X)$
  Для всех $B \in \Omega$ **выполнять**
    $\lambda \leftarrow \lambda_B(\overline{X})$
    Если $\lambda > \lambda_{Best}$ **тогда**
      $\lambda_{Best} \leftarrow \lambda$
      $B_{Best} \leftarrow B$
    Иначе если $\lambda > \lambda_{Second}$
      $\lambda_{Second} \leftarrow \lambda$
      $B_{Second} \leftarrow B$
    Конец условия
  Конец цикла
  Если $\frac{\lambda_{Best}}{\lambda_{Second}} \ge \lambda_*$ **тогда**
    Вернуть $B_{Best} \cap \Lambda$
  Конец условия
  Вернуть $\varnothing$
Конец функции

3 Данные репозитория UCI

В последнее время активно изучается тематика построения базисов ассоциативных правил [13], анализа зашумленных контекстов [2, 3, 6], и эффективной классификации [4, 5] в рамках направления АФП. Все эти задачи в той или иной степени могут быть отнесены к анализу данных, поэтому представляется важным сопоставить эти методы анализа формальных понятий с предлагаемыми методами классификации, основанными на вероятностных формальных понятиях (ВФП). Для сравнения была выбрана статья [7], в которой метод классификации заключается в построении особого рода решающих деревьев на основе концептуальных решеток (обозначим его как TreeFCA).

Основным источником данных является UCI [14]. К его преимуществам можно отнести обширные библиографические списки, группированные по наборам данных, а также широкую распространённость предлагаемых наборов данных в литературе.

Сравнение с [7] проводилось на следующих данных:

1) ***zoo*** – содержит 17 булевозначных признаков, каждый из которых описывает отдельный аспект строения животной особи. Последний признак задаёт класс животных, к которому особь принадлежит (целочисленное значение от 1 до 7);
2) ***kp-vs-kr*** – содержит шахматные эндшпили типа король+ладья против король+пешка. Каждый атрибут описывает какую-либо особенность позиции (например, близость белого короля к черной пешке) и является номинальным. Целевой признак описывает класс: белые могут выиграть (win), или белые не могут выиграть (nowin);
3) ***votes*** – репозиторий включает бюллетени опросов (каждый состоял из 16 граф) респондентов, принадлежащих к двум политическим партиям (республиканцы и демократы). В данных присутствуют пропуски, которые были проинтерпретированы как шумы и дополнены случайными значениями; в остальном исходные данные содержат булевы признаки, которые удобно было представить в виде формального контекста. Обработанный набор представляет собой контекст $K = (G, M \cup C, I)$, где $|G| = 435$, $|M| = 16$ и $C = \{m_{class}\}$, $gIm \Leftrightarrow$ данные содержат “yes” для выбранного респондента g в графе m.

Для решения задач классификации были использованы алгоритмы *ClassifyInCluster* и *ClassifyOverClusters*. Для исследования точности (*Accuracy*) использовалась техника кросс-валидации [4], при которой исходные данные делятся на N равномерных частей, и каждая часть используется в качестве контрольной выборки, в то время как остальная часть – в качестве обучающего контекста. Оценка точности предсказаний *Accuracy* на каждой отдельной выборке равнялась отношению правильно предсказанных классификатором классов к общему количеству объектов в контрольной выборке за вычетом отказов от классификации. Итоговая оценка *Accuracy* равна средней оценке точности по всем итерациям.

4 Результаты классификации

Эксперимент по классификации состоял из двух частей: обучения и контроля. На этапе обучения были задействованы алгоритмы семантического вероятностного вывода [15] для получения множества статистически значимых правил. *StatConcepts* выявил множества статистических формальных понятий, а дальнейшая процедура классификации выполнялась алгоритмами *ClassifyInCluster* и *ClassifyOverClusters* из раздела 3. Для тяжелых вычислений, таких как поиск множества статистически значимых правил, были задействованы мощности Сибирского суперкомпьютерного центра [16].

Вся выборка разбивалась на N частей для использования техники Cross-Validation. Параметр N немного отличается для различных наборов данных; точное значение указано в таблице 1. В столбцах указаны наборы анализируемых данных, количество итераций с различными разбиениями исходного множества объектов на обучающее GT и контрольное GC, процент верно и неверно предсказанных классификатором объектов суммарно по всем итерациям.

Результаты приведены в виде двух таблиц. В таблице 1 указаны характеристики применения методов *ClassifyInCluster* и *ClassifyOverClusters* к различным наборам данных из репозитория UCI [14]. Сравнение с [7] сведено в таблицу 2, куда включены результаты точности альтернативных алгоритмов C4.5, ID3, TreeFCA из указанной статьи.

С целью изучения гибкости алгоритмов на репозитории ***votes*** [7] была проведена дополнительная серия экспериментов по изучению модификаций процедуры замыкания из раздела 3. Классификация включала в себя серию экспериментов, в течение которой

видоизменялось либо семейство используемых для классификации алгоритмов (выбирались разные процедуры *Closure* в алгоритме *ClassifyInCluster*), либо какие-то их параметры. Основным измеряемым показателем является точность предсказаний (*Accuracy*), а также количество отказов (*Declined*) алгоритма от прогнозов.

Таблица 1 – Протокол экспериментов по классификации методами *ClassifyInCluster* (In) и *ClassifyOverClusters* (Over) на репозиториях UCI

		Значения показателей		
Репозиторий		***zoo***	***kp_vs_kr***	***votes***
Метод		In	Over	In
Показатели	Итераций	101	20	42
	Объектов	101	1790	420
	Отказов	5%	25.98%	47.62%
	Верно	92%	60.50%	50.71%
	Неверно	3%	13.52%	1.67%

Таблица 2 – Точность различных алгоритмов на наборах данных из UCI

		Точность алгоритмов		
Репозиторий		***zoo***	***kp_vs_kr***	***votes***
Алгоритмы	ВФП	96.84%	81.74%	96.82%
	C4.5	92.69%	72.78%	86.50%
	ID3	95.04%	74.50%	89.28%
	TreeFCA	96.04%	74.65%	90.5%

Результаты экспериментов приведены на рисунке 1, где, в частности, видно, что алгоритмы обладают различными качественным свойствами, а результаты, полученные с их помощью, хорошо локализованы.

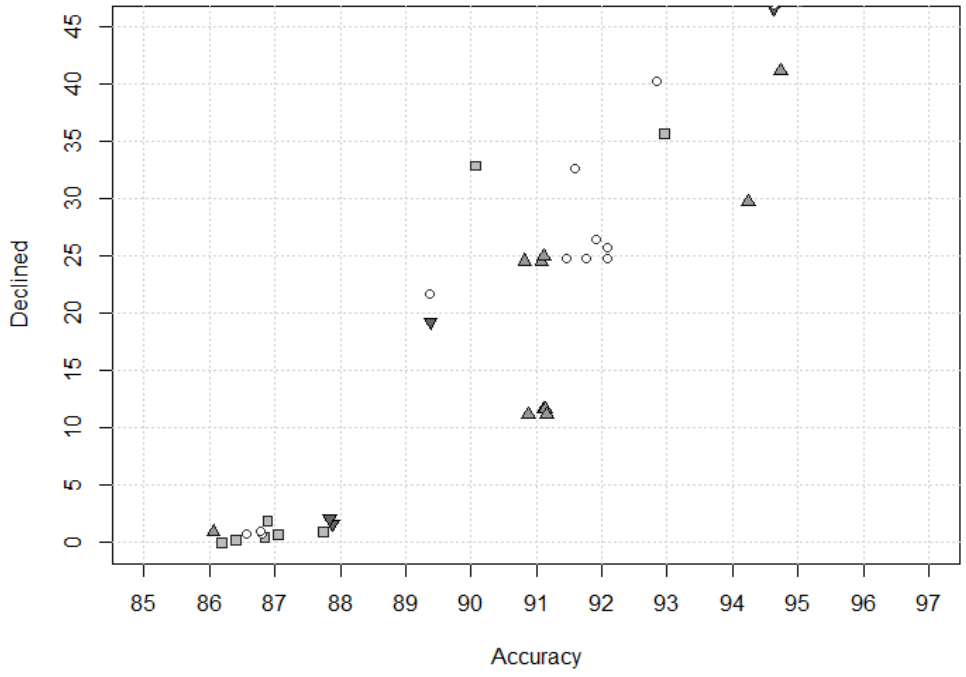

Рисунок 1 – Характеристики выполнения различных модификаций процедуры *Closure* при классификации данных ***votes***: △ - эксперименты близкие к методу *ConcConcepts*; ▽ - эксперименты с применением метода *StatConcepts*; □ - эксперименты смешанного метода с единичным весом противоречий; ○ - все остальные эксперименты

Заключение

Вероятностный подход к определению формальных понятий позволяет определить целое семейство алгоритмов кластеризации, смягчая проблему противоречивости логического вывода как для полностью определенных данных (формальных контекстов), так и для выборок из генеральной совокупности.

Алгоритмы классификации *ClassifyInCluster* и *ClassifyOverClusters*, построенные на основе статистических формальных понятий, позволяют успешно решать достаточно сложные задачи классификации, что было продемонстрировано на ряде наборов данных репозитория UCI, где они могут соперничать на равных с разработками АФП и классическими алгоритмами на основе решающих деревьев, имея в некоторых случаях ощутимое преимущество.

Статистические формальные понятия оказываются простыми в построении и полезными в прогнозировании. В то же время параметризация алгоритмов и их различные модификации обеспечивают необходимую гибкость при анализе данных. Заметна перспектива развития предлагаемого метода в рамках направления интеллектуального анализа данных: для этого следует провести более масштабные эксперименты, а также произвести интеграцию с уже существующими инструментами хранения и анализа данных.

Благодарности

Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант РФФИ 15-07-03410.

Список источников

[1] ***Ganter, B.*** Formal concept analysis. Mathematical Foundations / *B. Ganter* , *R. Wille*. - Berlin-Heidelberg: Springer-Verlag, 1999. – 290 p.

[2] ***Kuznetsov, S.O.*** Concept Stability as a Tool for Pattern Selection / S.O. Kuznetsov // ECAI 2014: CEUR Workshop proceedings. - 2014. - Vol. 1257. - P. 51-58.

[3] ***Klimushkin, M.*** Approaches to the Selection of Relevant Concepts in the Case of Noisy Data / Klimushkin, M., Obiedkov, S., Roth, C. // ICFCA 2010: LNAI. - 2010. - Vol. 5987. – P. 255-266.

[4] ***Prokasheva, O.*** Classification based on formal concept analysis and biclustering: Possibilities of the approach / Prokasheva, O., Onishchenko, A., Gurov, S. // Computational mathematics and modeling. - 2012. - Vol. 23(3).

[5] ***Quan, T.T.*** Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data / Quan, T.T., Hui, S.C., Cao, T.H. // CEUR Workshop proceedings, Belohlavek R., Snasel V. (Eds.). - 2004. - Vol. 110.

[6] ***Самойлов, Д.Е.*** Анализ неполных данных в задачах построения формальных онтологий / Д.Е. Самойлов, В.А. Семенова, С.В. Смирнов // Онтология проектирования. – 2016. – Т. 6, №3(21). - С. 317-339.

[7] ***Radim Belohlavek.*** Inducing decision trees via concept lattices / Radim Belohlavek, Bernard De Baets UGent, Jan Outrata and Vilem Vychodil // International journal of general systems. - 2009. – P. 455-467.

[8] ***Витяев, Е.Е.*** Вероятностное обобщение формальных понятий / Е.Е. Витяев, А.В. Демин, Д.К. Пономарев // Программирование. - 2012. - № 5. – С. 18-34.

[9] ***Витяев, Е.Е.*** Формализация естественной классификации и систематики через неподвижные точки предсказаний / Е.Е. Витяев, В.В. Мартынович // Сибирские электронные математические известия. Новосибирск: Изд-во института математики СО РАН. - 2015. - Т. 12. – С. 1006-1031.

[10] ***Goodfellow, I.*** Deep Learning / Goodfellow, I., Bengio, Y. and Courville, A. // MIT Press. 2016.

[11] ***LeCun, Y.*** A Tutorial on Energy-Based Learning / LeCun, Y. et al. // Predicting Structured Outputs, Bakir et al. (eds). MIT Press. - 2006.

[12] ***Vityaev, E.E.*** Transcription Factor Binding Site Discovery by the Probabilistic Rules / E.E. Vityaev, K.A. Lapardin, I.V. Khomicheva, A.L. Proskura // Proceedings of the 2nd workshop in data mining in functional genomics and proteomics.: The 18th European conference on Machine Learning and the 11th European conference on Principles and Practice of Knowledge Discovery in Databases. - 2007. - P. 104-109.

[13] Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04) / eds.: Bayardo Jr., R., Goethals B., Zaki M. CEUR-WS.org. - 2004.
[14] Репозиторий задач для методов Machine Learning. [Электронный ресурс]. URL: http://archive.ics.uci.edu/ml
[15] ***Воронцов, К.В.*** Комбинаторный подход к оценке качества обучаемых алгоритмов / К.В. Воронцов // Математические вопросы кибернетики. Под ред. О.Б. Лупанова. - М.: Физматлит, 2004. - Т. 13. - С. 5-36.
[16] Сибирский суперкомпьютерный центр. [Электронный ресурс]. URL: http://www2.sscc.ru/HKC-30T/HKC-30T.htm

PROBABILISTIC FORMAL CONCEPTS IN SOME CLASSIFICATION TASKS

E.E. Vityaev[1], V.V. Martynovich[2]

Sobolev institute of mathematics, SB RAS, Novosibirsk, Russia
[1]vityaev@math.nsc.ru, [2]vilco@yandex.ru

Abstract

The definition of formal concepts as fixed points of implication is considered. On the basis of this definition, the notion of probability formal concepts is introduced by replacing implications with special, maximally specific probability rules for which it was previously proved that fixed points for them are logically consistent. The ProbClosure algorithm for detecting probabilistic formal concepts is defined. To develop algorithms for clustering and classification, the context is considered as a sample from the general population. Generalizing the algorithm ProbClosure, algorithms for clustering ConcClosure and StatClosure are defined by introducing various energy functionals that determine the degree of non-contradiction of the rules at a fixed point. Classification algorithms are obtained by applying clustering algorithms to new data. Classification algorithms obtained are compared with the decision trees C4.5, ID3 and the classification method based on the lattice of formal concepts. The comparison was made on the data of the UCI repository. The obtained results showed comparatively high accuracy of the developed algorithms in comparison with these methods.

Key words: *formal concept analysis, probability, data mining, associative rules, classification, UCI.*

Citation: *Vityaev EE, Martynovich VV.* Probabilistic formal concepts in some classification tasks [In Russian]. *Ontology of designing.* 2017; 7(4): 473-486. DOI: 10.18287/2223-9537-2017-7-4-473-486.

References

[1] ***Ganter, B.*** Formal concept analysis. Mathematical Foundations. Berlin-Heidelberg: Springer-Verlag, 1999. – 290 p.
[2] ***Kuznetsov SO.*** Concept Stability as a Tool for Pattern Selection. ECAI 2014: CEUR Workshop proceedings. 2014; 1257: 51-58.
[3] ***Klimushkin M, Obiedkov S, Roth C.*** Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. ICFCA 2010: LNAI 5987. 2010: 255-266.
[4] ***Prokasheva O, Onishchenko A, Gurov S.*** Classification based on formal concept analysis and biclustering: Possibilities of the approach. Computational mathematics and modeling. 2012; 23(3).
[5] ***Quan TT, Hui SC, Cao TH.*** Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data. CEUR Workshop proceedings, Belohlavek R., Snasel V. (Eds.). 2004; 110.
[6] ***Samoilov DE, Semenova VA, Smirnov SV.*** Incomplete data analysis of for building formal ontologies [In Russian]. *Ontology of designing.* 2016; 6(3): 317-339.
[7] ***Belohlavek R, De Baets B, Outrata J, Vychodil V.*** Inducing decision trees via concept lattices. International journal of general systems. 2009; 455-467.
[8] ***Vityaev EE, Demin AV, Ponomaryov DK.*** Probabilistic Generalization of Formal Concepts [In Russian]. Programming. 2012; 38(5): 219–230.

[9] ***Vityaev EE, Martinovich VV.*** Formalization of «natural» classification and systematics as fix-points ofpredictions [In Russian]. Siberian Electronic Mathematical Reports. Novosibirsk: IM SD RAS. 2015; 12: 1006-1031.
[10] ***Goodfellow I, Bengio Y, Courville A.*** Deep Learning. - MIT Press. 2016.
[11] ***LeCun Y.*** et al. A Tutorial on Energy-Based Learning. Predicting Structured Outputs, Bakir et al. (eds). - MIT Press, 2006.
[12] ***Vityaev EE, Lapardin KA, Khomicheva IV, Proskura AL.*** Transcription Factor Binding Site Discovery by the Probabilistic Rules. Proceedings of the 2nd workshop in data mining in functional genomics and proteomics.: The 18th European conference on Machine Learning and the 11th European conference on Principles and Practice of Knowledge Discovery in Databases. 2007; 104-109.
[13] Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04) / eds.: Bayardo Jr., R., Goethals B., Zaki M. CEUR-WS.org. - 2004.
[14] Machine Learning repository. URL: http://archive.ics.uci.edu/ml
[15] ***Vorontsov KV.*** Kombinatorical approach to the quality of the learning algorithms estimation [In Russian]. Mathematical questions in kibernetics. (Ed.: O.B. Lupanov). - Moscow: Fizmatlit, - 2004; 13: 5-36.
[16] Siberian supercomputer center - URL: http://www2.sscc.ru/HKC-30T/HKC-30T.htm

Сведения об авторах

Витяев Евгений Евгеньевич, 1948 г. р. Окончил Новосибирский государственный университет в 1971 г., д.ф.-м.н. (2007). Профессор кафедры дискретной математики и информатики Новосибирского государственного университета. В списке научных трудов более 250 работ в области логики, интеллектуального анализа данных и искусственного интеллекта.

Vityaev Evgeny Evgenievich, born in 1948. He graduated the Novosibirsk State University in 1971, Doctor of Science (2007). Professor of the Department of Discrete Mathematics and Informatics at Novosibirsk State University. In the list of scientific works more than 250 works in the field of logic, data mining and AI.

Мартынович Виталий Валерьевич, 1990 г. р. В 2016 г. окончил аспирантуру Новосибирского государственного универсистета. С 2017 г. младший научный сотрудник института математики СО РАН. В списке научных трудов около 10 работ в области дискретной математики, методов анализа данных и интеллектуальных систем.

Vitaly Valerievich Martynovich (b. 1990) post-graduated from the Novosibirsk State University PhD program in 2016. Junior researcher at SB RAS Institute of Mathematics from 2017. Author and co-author of about 10 scientific articles around discrete mathematics, Data Mining methods and intelligent systems.