УДК 004.853

В.И. Городецкий, О.Н. Тушканова

АССОЦИАТИВНАЯ КЛАССИФИКАЦИЯ: АНАЛИТИЧЕСКИЙ ОБЗОР. ЧАСТЬ 2

Городецкий В.И., Тушканова О.Н. **Ассоциативная классификация: аналитический обзор. Часть 2.**

Аннотация. В работе продолжается рассмотрение основных результатов, моделей и методов, разработанных в области ассоциативной классификации, ориентированных на обработку данных большого объема. Дается анализ подходов, методов и алгоритмов, разработанных в области ассоциативной классификации к настоящему времени. В заключении формулируются достоинства и недостатки ассоциативной классификации как модели машинного обучения, а также дается оценка перспектив ее использования в интеллектуальном анализе больших данных.

Ключевые слова: большие данные, ассоциативное правило, ассоциативная классификация, паттерн, эмерджентный паттерн.

Gorodetsky V., Tushkanova O. **Associative Classification: Analytical Overview. Part 2.**

Abstract. The paper continues the survey of associative classification in context of big data processing. An extended overview and comparative analysis of the modern approaches, models and algorithms developed for associative classification form the main paper contents. In conclusion, the paper outlines the main advantages and drawbacks of associative classification, as well as evaluates its capabilities from big data processing perspective.

Keywords: associative classification, emerging pattern, big data.

1. Введение. Данная работа является продолжением работы [1], в которой были описаны более ранние результаты в области ассоциативной классификации. В данной работе представлены современные алгоритмы, модели и методы, разработанные в области ассоциативной классификации, которые уже в большей мере ориентированы на обработку данных большого объема.

Повторим кратко формальную постановку задачи ассоциативной классификации, приведенную более детально в первой части данной работы [1].

Пусть $\boldsymbol{D}$ – транзакционная база данных (множество данных), $D_i \in \boldsymbol{D}$ – произвольная транзакция, $\boldsymbol{X}$ – множество всех символов, которые используются для обозначения объектов (признаков, атрибутов) в транзакциях множества $\boldsymbol{D}$, $\boldsymbol{A}$ – подмножество символов из множества $\boldsymbol{X}$ и $\boldsymbol{D(A)}$ – подмножество множества транзакций из множества $\boldsymbol{D}$, каждая из которых содержит подмножество символов $\boldsymbol{A} \in \boldsymbol{X}$ в качестве подмножества. Для характеристики статистических свойств подмножества $\boldsymbol{A}$ в базе данных $\boldsymbol{D}$ используют отношение мощности n_A множества $\boldsymbol{D(A)}$ к мощности n всего множества транзакций $\boldsymbol{D}$. Эту величину

принято называть *поддержкой* (*support*) подмножества $\boldsymbol{A}$ во множестве транзакций $\boldsymbol{D}$:

$$supp(\boldsymbol{A}) = n_A / n. \quad (1)$$

Пусть даны два набора символов (объектов) $\boldsymbol{A} \in \boldsymbol{X}$ и $\boldsymbol{B} \in \boldsymbol{X}$, причем $\boldsymbol{A}$ и $\boldsymbol{B}$ не имеют общих элементов, и пусть σ и γ – вещественные числа из интервала [*0, 1*]. Говорят [2, 3], что выражение вида $\boldsymbol{A} \rightarrow \boldsymbol{B}$ есть *ассоциативное правило* с *порогом уверенности* $conf(\boldsymbol{A} \rightarrow \boldsymbol{B}) = \gamma$ и *порогом поддержки* $supp(\boldsymbol{A}) = \sigma$ (σ, γ – ассоциативное правило), если справедливы следующие неравенства:

$$n_{AB} / n \geq \sigma, \quad (2)$$

$$n_{AB} / n_A \geq \gamma, \quad (3)$$

где n_{AB} – количество транзакций во множестве $\boldsymbol{D}$, которые содержат объединение множества символов подмножеств $\boldsymbol{A}$ и $\boldsymbol{B}$. Модель ассоциативного правила, заданную условиями (2), (3), принято называть моделью типа *поддержка–уверенность*.

Подмножество (последовательность) элементов $\boldsymbol{A}$ принято называть посылкой ассоциативного правила $\boldsymbol{A} \rightarrow \boldsymbol{B}$, а подмножество (последовательность) $\boldsymbol{B}$ – его следствием. Обычно эти последовательности называют паттернами (*patterns*). В задачах ассоциативной классификации заключение правила может содержать только однолитерный паттерн, который является именем одного из классов. Поэтому в общем случае основная подзадача задачи ассоциативной классификации сводится к поиску множества (σ, γ)–ассоциативных правил для каждого класса. Эта подзадача называется обычно задачей *обучения* классификатора. Другая подзадача – это синтез классификатора на множестве найденных ассоциативных правил. Эта задача не является предметом данной работы.

В последующей части работы дается описание и сравнительный анализ основных результатов, полученных в области ассоциативного анализа к настоящему времени. В заключении формулируются достоинства и недостатки ассоциативной классификации, а также оцениваются перспективы использования методов ассоциативной классификации для интеллектуального анализа больших данных.

2. Алгоритмы ассоциативной классификации: современные подходы. Большая группа современных методов и алгоритмов генерации ассоциативных правил классификации основана на понятии эмерджентный паттерн. По сути, методы ассоциативной классифика-

ции, основанные на эмерджентных паттернах, сформировали новое направление в этой области, которое активно развивается и сегодня. Работы в этом направлении [4-7] во многом способствовали более глубокому пониманию специфики задач ассоциативной классификации.

Первой работой, в которой было введено понятие *эмерджентного паттерна* (*Emergent Pattern, EP*), далее для краткости *ЭП*, была работа [4]. В ней задача ассоциативной классификации была поставлена как задача дискриминации, т.е. как задача поиска правил, позволяющих отделить примеры одного класса от примеров другого класса. Заметим, что в работах, упомянутых ранее, прагматика ассоциативных правил как правил классификации при их генерации вообще не принималась во внимание. В работе [4], наоборот, в качестве базового критерия отбора правил ассоциативной классификации рассматривается их способность отличать примеры одного класса от примеров другого класса. В дальнейшей эволюции таких моделей эта прагматика оставалась неизменной, а совершенствование методов и реализующих их алгоритмов было направлено на повышение их вычислительной эффективности при генерации ассоциативных правил и при использовании этих правил в алгоритмах классификации.

Сформулируем понятие ЭП, следуя работе [4]. Пусть дана упорядоченная пара $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ транзакционных данных, которые либо относятся к разным классам, либо относятся к разным временным интервалам лога работы некоторой системы. Как и раньше (см. раздел 2), каждая транзакция из множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ может содержать элементы (атрибуты, переменные, предметы, объекты, англ. *items*) из (линейно упорядоченного) множества (последовательности) $\boldsymbol{X}$. Рассмотрим произвольный паттерн $A \subseteq \boldsymbol{X}$, который характеризуется поддержкой σ_1 во множестве $\boldsymbol{D_1}$ и поддержкой σ_2 во множестве $\boldsymbol{D_2}$. Отношение σ_2 / σ_1 авторы работы [4] называют *коэффициентом возрастания поддержки* (*Growth rate*) паттерна $\boldsymbol{A}$ от множества данных $\boldsymbol{D_1}$ к множеству $\boldsymbol{D_2}$. Формально значение показателя *GrowthRate(A)* определяется нижеследующей формулой:

$$GrowthRate(A)=\begin{cases} 0, \textit{если } \sigma_1(\boldsymbol{A})=0 \textit{ и } \sigma_2(\boldsymbol{A})=0, \\ \infty, \textit{если } \sigma_1(\boldsymbol{A})=0 \textit{ и } \sigma_2(\boldsymbol{A})>0, \\ \sigma_2(\boldsymbol{A})/\sigma_1(\boldsymbol{A}), \textit{в других случаях}. \end{cases} \quad (4)$$

Определение ЭП дается с использованием порогового значения ρ для величины *GrowthRate(**A**)*: паттерн $\boldsymbol{A}$ называется *эмерджентным паттерном* от множества $\boldsymbol{D_1}$ к множеству $\boldsymbol{D_2}$, если

GrowthRate(***A***) ≥ ρ. Таким образом, авторы вводят понятие ЭП с использованием порога ρ, чтобы выбором его значения можно было управлять их разделяющей способностью. Заметим, что понятие меры уверенности для ЭП на заданных множествах данных $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ становится ненужным, т.к. ее значение для обоих множеств равно 1, поскольку всем транзакциям каждого из множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ ставится в соответствие постоянное заключение, например, метка класса или имя временного интервала.

С учетом введенных понятий и определений задача поиска ассоциативных правил в работе [4] сводится к поиску эмерджентных паттернов $\boldsymbol{A_i}$ со значением меры *GrowthRate*($\boldsymbol{A_i}$) ≥ ρ. Обратим внимание на то, что задача поиска ЭП не использует понятие поддержки, а опирается на понятие коэффициента роста поддержки. Это означает, что "хорошие" паттерны могут иметь низкое значение поддержки в обоих множествах, что приводит к значительному возрастанию вычислительной сложности задачи поиска *EP*. Причины этого обусловлены тем, что, во-первых, паттернов с низким уровнем поддержки всегда бывает очень много. Во-вторых, при поиске паттернов по условию *GrowthRate*($\boldsymbol{A_i}$) ≥ ρ принципиально нельзя воспользоваться алгоритмом *Apriori*, поскольку для *EP* не соблюдается условие монотонного уменьшения значения поддержки паттерна при увеличении его длины за счет добавления новых атрибутов.

Таким образом, данная работа показала, что поиск ассоциативных правил для задач классификации является задачей, которая, во-первых, отличается от поиска обычных ассоциативных правил, и, во-вторых, имеет не так много общего с задачей поиска правил в машинном обучении.

Для пояснения существа задачи поиска ЭП, а также для ее декомпозиции авторы используют двухмерное представление области локализации различных типов ЭП в координатах $<\sigma_2, \sigma_1>$, т.е. в координатах мер поддержки паттернов во множестве данных $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$. В этой области (рисунок 1) все ЭП располагаются правее прямой l_1, тангенс угла α наклона которой к оси $O\sigma_2$ равен величине *1/ρ*. Все ЭП располагаются в треугольнике *ACE*. Однако посылки правил ассоциативной классификации, должны иметь значение поддержки, превышающее минимально допустимое значение $\sigma_{2\,min}$ во множестве $\boldsymbol{D_2}$. Кроме того, во множестве $\boldsymbol{D_1}$ они должны иметь значение поддержки меньше, чем $\sigma_{2\,min}$. Паттерны с такими значениями мер поддержки σ_2

и σ_1 располагаются в прямоугольнике *BCDG,* и именно они являются целью поиска в работе [4].

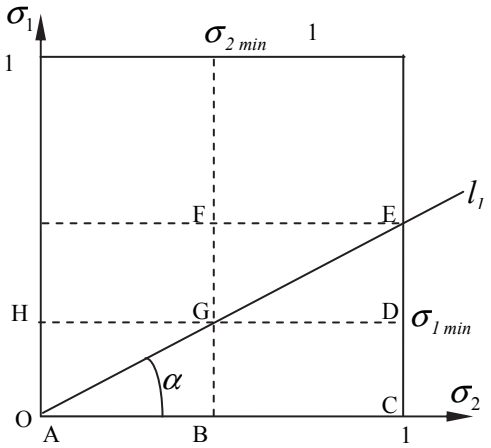

Рис. 1. Пояснение к алгоритму поиска эмерджентных паттернов

Прежде чем рассматривать алгоритм поиска таких ЭП, введем дополнительно важное понятие *интервально замкнутого множества*, использование которого позволяет авторам повысить эффективность алгоритма поиска эмерджентных правил. Заметим, что хотя авторы претендуют на авторство во введении этого понятия, оно давно и хорошо известно в алгебре, в частности, в теории алгебраических структур. Однако использование этого понятия для эффективного поиска ассоциативных правил, несомненно, принадлежит авторам [4].

Интервально замкнутое множество $\Im$ определяется следующим образом. Пусть дана упорядоченная пара множеств подмножеств [***L, R***], $\boldsymbol{L} = \{\boldsymbol{A}_i\}_{i=1}^{k}$ и $\boldsymbol{R} = \{\boldsymbol{B}_j\}_{j=1}^{r}$, при этом любое подмножество $\boldsymbol{A_i} \subseteq \boldsymbol{B}_j$ для некоторого значения индекса *j*, а все подмножества $\boldsymbol{A} \in \{\boldsymbol{A}_i\}_{i=1}^{k}$, как и все подмножества $\boldsymbol{B} \in \{\boldsymbol{B}_j\}_{j=1}^{r}$, являются несравнимыми по отношению включения. Тогда множество *всех* подмножеств $\{\boldsymbol{Z}_l\}_{l=1}^{s}$, такое, что для любого $\boldsymbol{Z}_l$ найдется пара подмножеств $\boldsymbol{A_i} \in \boldsymbol{L}$ и $\boldsymbol{B_i} \in \boldsymbol{R}$, для которых выполнено условие $\boldsymbol{A_i} \subseteq \boldsymbol{Z}_l \subseteq \boldsymbol{B}_j$, называется интервально замкнутым множеством подмножеств $\Im$=[**L**, **R**] с грани-

цами **L** и **R**. Множества подмножеств $\boldsymbol{L}=\{\boldsymbol{A}_i\}_{i=1}^{k}$ и $\boldsymbol{R}=\{\boldsymbol{B}_j\}_{j=1}^{r}$ называются *правой* и *левой границами* множества $\Im$ соответственно. Заметим, что границы множества $\Im$ принадлежат ему. Для заданных границ интервально замкнутое множество определяется единственным образом. Справедливо и обратное: если множество $\Im$ является интервально замкнутым, то его нижние и верхние границы определяются единственным образом.

Приведем пример интервально замкнутого множества подмножеств (рисунок 2) [4]. Пусть заданы границы множества: левая (нижняя) **L**=$\{\{\theta\}\}$, где θ – символ пустого множества, и правая (верхняя) **R** = $\{\{a_1,a_2,a_3\},\ \{a_1,a_4\}\}$. Тогда интервально замкнутое множество $\Im=[\{\{\theta\}\},\ \{\{a_1,a_2,a_3\},\ \{a_1,a_4\}\}]=\{\{a_1\},\ \{a_2\},\ \{a_3\},\ \{a_1,a_2\},\ \{a_1,a_3\},\ \{a_2,a_3\},\ \{a_1,a_2,a_3\},\ \{a_4\},\ \{a_1,a_4\}\}$.

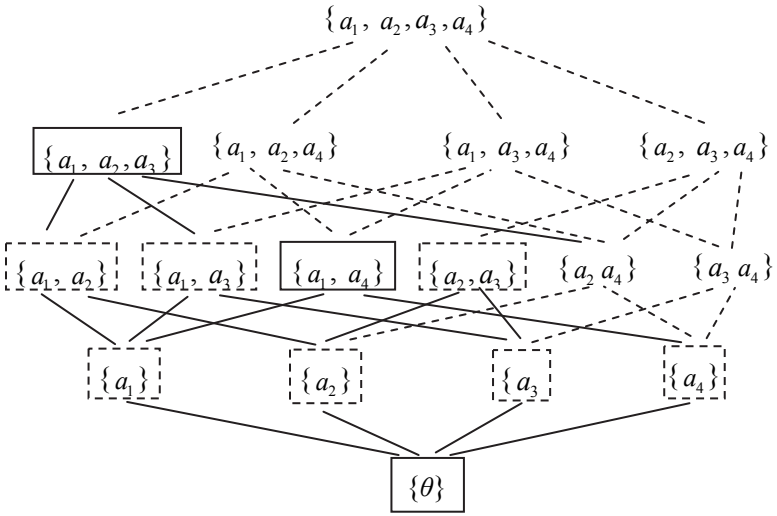

Рис. 2. Пример интервально замкнутого множества. Его элементы обведены прямоугольниками. Элементы нижней и верхней границ обведены прямоугольниками со сплошными сторонами

Способ построения интервально замкнутого множества по заданным его границам можно пояснить с помощью диаграммы Хассе частично упорядоченного множества $\Im$=[**L**, **R**], рассматриваемого в примере. Максимальные и минимальные элементы входят в интер-

вально замкнутое множество $\Im$. Остальные его элементы формируются как множества подмножеств его максимальных элементов, каждое из которых содержит в себе хотя бы один минимальный элемент. В рассматриваемом примере **L**={{θ}}, **R** = {{a_1, a_2, a_3}, {a_1, a_4}}. П содержит всего 4 разных элемента, а именно, a_1, a_2, a_3 и a_4. Диаграмма Хассе множества всех подмножеств, которые могут быть образованы из этих элементов, представлена на рисунке 3. В ней всего 16 элементов вместе с пустым множеством. Элементы нижней и верхней границ рассматриваемого примера на этой диаграмме обведены прямоугольниками со сплошными границами. Остальные элементы множества $\Im$=[**L**, **R**] на диаграмме Хассе, представленной на рисунке 3, обведены прямоугольниками с пунктирными границами. Отношения включения на множестве элементов $\Im$ показаны сплошными линиями. Таким образом, 9 множеств, входящих в интервально замкнутое множество $\Im$ (не считая пустого множества), задаются всего тремя граничными множествами.

Понятие интервально замкнутого множества паттернов (подмножеств) для поиска эмерджентных ассоциативных правил оказывается весьма полезным при построении эффективных алгоритмов их поиска, поскольку

а) множество всех паттернов, удовлетворяющих заданному ограничению на минимальное значение поддержки, является интервально замкнутым, и

б) алгоритм их поиска оказывается возможным свести к манипуляциям только с элементами границ интервально замкнутого множества. Например, вместо рассмотрения 9 паттернов в приведенном примере можно будет ограничиться рассмотрением только двух максимальных паттернов (в данном примере нижняя граница формируется пустым множеством, которое не соответствует какому-либо паттерну).

Среди интервально замкнутых семейств множеств авторы выделяют три специальных случая. Множество $\Im$=[**L,R**] называется *левокорневым*, если его левая граница состоит из одноэлементного множества. Если его правая граница состоит их одноэлементного множества, то оно называется *правокорневым*. В общем случае, если одноэлементными являются или множества, задающие правую границу, или множества, задающие левую границу, или оба эти множества являются корневыми, то множество $\Im$=[**L,R**] называют *корневым*. Авторы ограничиваются рассмотрением именно корневых интервально замкнутых множеств паттернов.

Опишем теперь кратко алгоритм поиска ЭП, предложенный в работе [4]. Он опирается на утверждение о том, что *множество всех паттернов (подмножеств элементов), которые имеют поддержку не меньше, чем некоторый заданный порог* $\sigma_{2\,min}$, *является левокорневым интервально замкнутым* множеством [4]. Аналогичное утверждение справедливо также для множества паттернов, которые имеют поддержку меньше, чем некоторый заданный порог $\sigma_{2\,min}$. Такое множество является *правокорневым* интервально замкнутым множеством. Если обратиться к рисунку 2, то правокорневым будет множество всех ЭП, для которых пары [σ_2, σ_1] отвечают точкам треугольника *ABG*. Множество всех ЭП для множества транзакций $\boldsymbol{D_2}$, которое отвечает четырехугольнику *BCEG*, будет левокорневым интервально замкнутым множеством паттернов. Напомним, что все эти паттерны являются эмерджентными от множества данных $\boldsymbol{D_1}$ к множеству данных $\boldsymbol{D_2}$ (поскольку эта область лежит ниже прямой l_1).

Обозначим *правую* границу множества паттернов для данных $\boldsymbol{D} \in \{\boldsymbol{D}_1, \boldsymbol{D}_2\}$ с поддержкой больше порога σ_{min} символом $LB_{\sigma_{min}}(\boldsymbol{D})$ (*Large Border*), а левую границу множества паттернов для данных $\boldsymbol{D}$ с поддержкой меньше порога σ_{min} – символом $SB_{\sigma_{min}}(\boldsymbol{D})$ (*Small Border*). Множество всех паттернов первого типа обозначим символом $\Im = \left(LB_{\sigma_{min}}, \boldsymbol{D}\right) = \left[\{\theta\}, LB_{\sigma_{min}}(\boldsymbol{D})\right]$, а множество всех паттернов второго типа обозначим символом $\Im = \left(SB_{\sigma_{min}}, \boldsymbol{D}\right) = \left[SB_{\sigma_{min}}(\boldsymbol{D}), \{I\}\right]$, где $\{\theta\}$ – пустое множество, а $\{I\}$–универсальное множество, т.е. множество всех паттернов, которые могут быть составлены из символов множества *I*.

Эмерджентные паттерны, которые являются кандидатами на использование их в качестве посылок ассоциативных правил, разделяющих множества $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$, должны быть σ_{2min}– паттернами множества данных $\boldsymbol{D_2}$. Но они не должны быть одновременно σ_{1min}– паттернами множества данных $\boldsymbol{D_1}$. Суть алгоритма, предложенного в рассматриваемой работе, состоит в поиске таких и только таких ЭП. Эффективность алгоритма обеспечивается тем, что он не оперирует с самими множествами $\Im = \left(LB_{\sigma_{1min}}, \boldsymbol{D_1}\right)$ и $\Im = \left(LB_{\sigma_{2min}}, \boldsymbol{D_2}\right)$. Он оперирует только их верхними и нижними границами.

Дадим теперь описание алгоритма поиска ЭП. Он строится на основе трех базовых алгоритмов, называемых далее алгоритмами 1, 2 и

3 соответственно. Рассмотрим их кратко. Детальные их описания и обоснования могут быть найдены в работе [4], где эти алгоритмы представлены в псевдокоде.

Алгоритм 1. В основу этого алгоритма положен алгоритм *Max-Miner*, предложенный в работе [8]. Он позволяет для заданного множества данных $\boldsymbol{D} \in \{\boldsymbol{D_1}, \boldsymbol{D_2}\}$ эффективно строить паттерны максимальной длины со значением поддержки не менее, чем заданное значение порога. Если говорить в терминах решаемой задачи поиска ЭП, то он позволяет быстро находить верхние (правые) границы $LB_{\sigma_{2min}}(\boldsymbol{D_2})$ и $LB_{\sigma_{1min}}(\boldsymbol{D_1})$ множеств паттернов для множеств данных $\boldsymbol{D_2}$ и $\boldsymbol{D_2}$ соответственно. Напомним, что левой границей для обоих множеств данных является одноэлементное множество $\{\theta\}$. Как отмечается в работе [8], алгоритм *Max-Miner* может эффективно отыскивать паттерны максимальной длины, содержащие до 13 элементов. Он строится на основе стандартного алгоритма *Apriori* [9]. Дополнительно к нему, алгоритм *Max-Miner* использует процедуру *просмотра вперед* (*look ahead*), генерируя в опережающей манере паттерны большей длины, чем те, которые обычно генерируются алгоритмом *Apriori* на его текущих шагах. Увеличение эффективности достигается за счет того, что просмотр вперед позволяет отсечь неперспективные направления продолжения поиска максимальных паттернов на более ранних шагах.

Итак, *алгоритм 1* отыскивает множества максимальных паттернов $LB_{\sigma_{2min}}(\boldsymbol{D_2})$ и $LB_{\sigma_{1min}}(\boldsymbol{D_1})$ при заданных значениях нижних порогов поддержки σ_{2min} и σ_{1min} соответственно. В общем случае они будут иметь вид:

$$LB_{\sigma_{1min}}(\boldsymbol{D_1}) = [\{\theta\}, \{\boldsymbol{A_1}, \boldsymbol{A_2}, \ldots, \boldsymbol{A_s}\}] \quad (5)$$

$$LB_{\sigma_{2min}}(\boldsymbol{D_2}) = [\{\theta\}, \{\boldsymbol{B_1}, \boldsymbol{B_2}, \ldots, \boldsymbol{B_k}\}]. \quad (6)$$

Как указывалось ранее, дальнейший поиск множества ЭП сводится к поиску границ множества, которое включает в себя паттерны множества $\Im = (LB_{\sigma_{2min}}, \boldsymbol{D_2})$, из которого удалены паттерны, входящие во множество $\Im = (LB_{\sigma_{1min}}, \boldsymbol{D_1})$. Этот поиск реализуется с использованием алгоритмов 2 и 3.

Алгоритм 2. Он реализует стандартную операцию поиска множества минимальных элементов (другими словами, левой границы) теоретико–множественной разности двух интервально замкнутых левокорневых множеств паттернов, имеющих в качестве левой границы

пустое множество $\{\theta\}$. Дополнительно к этому, первый элемент искомой разности должен быть одновременно и правокорневым множеством паттернов. Заметим, что в алгоритме поиска ЭП он применяется для поиска левой границы разности двух интервально замкнутых множеств вида $[\{\theta\},\{\boldsymbol{B}_j\}]$ и $[\{\theta\},\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_i\}]$, где $\boldsymbol{B}_j \in \{\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_k\}$ (см. (5) и (6)).

Авторы рассматривают два варианта реализации этой операции. В первом варианте сначала генерируются оба множества паттернов по их границам $[\{\theta\},\{\boldsymbol{B}_j\}]$ и $[\{\theta\},\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_k\}]$, а затем из первого множества паттернов удаляются те паттерны, которые одновременно содержатся и во втором множестве. Далее из полученного множества удаляются элементы, которые не являются минимальными в смысле частичного порядка по включению множеств. Второй вариант алгоритма, более эффективный, чем первый, отличается от него тем, что в нем построение верхней границы выполняется рекурсивно по отношению к множествам $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_k$ с удалением элементов, не являющихся минимальными, на каждом шаге. Таким способом достигается снижение мощности общего множества паттернов, генерируемых и просматриваемых в процессе построения границ разности двух левокорневых интервально замкнутых множеств указанного вида.

Алгоритм 3. Этот алгоритм использует алгоритмы 1 и 2. Он находит все ЭП, которые отвечают прямоугольнику BCDG, представленному на рисунке 1. Дадим краткое описание алгоритма 3. Его псевдокод можно найти в [4].

Входом алгоритма являются два интервально замкнутых множества паттернов (5) и (6) с паттернами множеств $LB_{\sigma_{2min}}(\boldsymbol{D}_2)$ и $LB_{\sigma_{1min}}(\boldsymbol{D}_1)$ в качестве правых границ множеств паттернов для данных $\boldsymbol{D}_2$, и $\boldsymbol{D}_1$ соответственно. Они находятся с помощью алгоритма 1, описанного выше.

Алгоритм реализуется последовательно для значений индексов $j \in 1,2,\ldots,k$ нижеследующим образом.

1. Для паттерна $\boldsymbol{B}_j$ строятся паттерны $\hat{\boldsymbol{A}}_1 = \boldsymbol{B}_j \cap \boldsymbol{A}_1$, $\hat{\boldsymbol{A}}_2 = \boldsymbol{B}_j \cap \boldsymbol{A}_2, \ldots, \hat{\boldsymbol{A}}_k = \boldsymbol{B}_j \cap \boldsymbol{B}_k$.

2. С помощью алгоритма 1 строится множество максимальных паттернов для найденного множества паттернов $\{\hat{\boldsymbol{A}}_1, \hat{\boldsymbol{A}}_2, \ldots, \hat{\boldsymbol{A}}_k\}$. Обозначим полученное множество символом $SB_{\sigma_{2min}}(\hat{\boldsymbol{A}}\boldsymbol{B}_j)$.

3. С помощью алгоритма 2 находится левая граница разности двух интервально замкнутых левокорневых множеств паттернов $\left[\{\theta\},\{\boldsymbol{B}_j\}\right]$ и $\left[\{\theta\},\left\{SB_{\sigma_{min}}(\hat{\boldsymbol{A}}\boldsymbol{B}_j)\right\}\right]$ и полученный результат, а именно множество минимальных паттернов (множество паттернов левой границы) этой разности добавляется в искомое множество ЭП.

Напомним, что описанный алгоритм имеет целью поиск ЭП, ля которых значения мер поддержки отвечают прямоугольнику *BCDG* (см. рисунок 2). Авторы замечают, что поиск ЭП, отвечающих треугольнику *ABG*, является вычислительно сложной задачей ввиду того, что в этой области паттерны, получаемые на основе данных множества $\boldsymbol{D_1}$, обладают низким значение поддержки, а потому их может быть катастрофически много. Наоборот, паттерны, которые получаются для множества данных $\boldsymbol{D_1}$, будут отвечать треугольнику *GDE*, обычно бывает немного и их можно проверить простым перебором.

Работа [4] рассматривает только вопрос о том, как находятся ЭП с заданными областями значений поддержки во множествах $\boldsymbol{D_2}$ и $\boldsymbol{D_1}$. Но эта работа не рассматривает, каким образом полученные ЭП используются далее в алгоритме ассоциативной классификации. Этому вопросу посвящена работа [5]. Дадим краткое описание технологии построения ассоциативного классификатора, представленной в этой работе.

Отметим, что одна из особенностей задачи классификации с использованием ЭП в качестве посылок правил состоит в том, что приходится использовать большое число правил, каждое из которых может покрывать только небольшое число примеров обучающей выборки. То же самое имеет место, как правило, и в режиме работы с новыми данными. Это означает, что с помощью таких правил нельзя строить алгоритмы классификации типа голосования правил. В отличие от этого, правила классификации, генерируемые большинством других методов с большим значением покрытия, можно рассматривать как самостоятельные классификаторы. Если каждый из них имеет вероятность правильной классификации больше, чем 0,5, то в таком случае в соответствии с теоремой Кондорсе результат голосования сходится с вероятностью 1 к правильному решению при увеличении числа правил [10, 11]. Поэтому для таких правил схемы голосования работают обычно хорошо. В отличие от этого, каждый ЭП работает правильно на очень небольшой доле обучающих данных, а вероятность правильной классификации с помощью ЭП на всем множестве данных может быть менее 0,01. Поэтому было бы правильнее интерпретировать ЭП как некоторые более удобные новые признаки, каждый из которых все еще не может рассматриваться как "хороший" классификатор. Именно поэтому авторы работы [5] рассматривают задачу построения классификаторов на основе ЭП как самостоятельную задачу.

Полагая, что алгоритм поиска ЭП имеется (см. [4]), авторы работы [5] рассматривают процедуру построения ассоциативного классификатора, в которой поиск ЭП является одной из готовых процедур. Такой подход позволяет хорошо понять достоинства и недостатки ассоциативной классификации на основе ЭП. Опишем кратко предложенную процедуру построения ассоциативного классификатора.

Первым ее шагом является преобразование обучающих данных к булевой форме, когда каждый элемент транзакции $\boldsymbol{A} \subseteq \boldsymbol{X}$ представляется булевой переменной, которая принимает значение *true*, если соответствующий элемент множества (последовательности) $\boldsymbol{X}$ в транзакции $\boldsymbol{A}$ присутствует, и значение *false*, в противном случае. Для перехода к такому представлению данных авторы используют самый простой метод – разбиение на интервалы числовых атрибутов с введением новой пропозициональной переменной для каждого интервала. Заметим, что такой вариант преобразования исходных типов данных задачи к булевой форме может привести к заметному увеличению размерности итогового пространства атрибутов, в котором придется далее решать задачу обучения классификатора.

На втором шаге отыскиваются ЭП для каждого класса, когда все множество данных обучения разбивается на два подмножества. Одно из них – это подмножество данных класса, для которого отыскиваются ЭП, а второе – это данные всех остальных классов. Заметим, что такой подход является типичным для задач классификации, в которой число классов больше двух.

Третий шаг построения классификатора является ключевым, и именно он содержит основные особенности. Его целью является агрегирование ЭП, имеющее целью преобразование подмножеств ЭП каждого класса в более выразительные структуры для увеличения их *дискриминационных* возможностей.

Сначала авторы рассматривают индивидуальные разделительные возможности ЭП. Эти возможности, обычно, очень ограничены. Например, если ЭП покрывает 3% примеров данных и при этом он классифицирует их правильно с вероятностью, например, 0,8, то вероятность правильного предсказания этим ЭП на всем множестве примеров обучающих данных будет порядка 0,024 [5]. Для оценки индивидуальных разделительных возможностей ЭП вводят формулу:

$$Score(E_i, C, s) = \frac{Growth\_Rate(E_i)}{Growth\_Rate(E_i)+1} \times support_C(E_i), \quad (7)$$

где $\boldsymbol{E_i}$ – ЭП, разделительные возможности которого оцениваются функцией (7) для примера s по отношению к классу $\boldsymbol{C}$, а величина $support_C(E_i)$ есть поддержка паттерна $\boldsymbol{E_i}$ в классе $\boldsymbol{C}$.

Авторы обращают внимание на то, что в выражении $\{Growth\_Rate(E_i)/ (Growth\_Rate(E_i)+1)\} \times support_C(E_i)$ первый сомножитель примерно равен *условной вероятности* события "*пример s, в котором имеется ЭП* $\boldsymbol{E_i}$*, принадлежит классу C*", а второй сомножитель – это доля примеров класса *C*, которые содержат данный ЭП. Сумму всех таких величин по множеству всех ЭП класса *C* авторы предлагают рассматривать в качестве величины, характеризующей разделительную силу построенного множества ЭП для этого класса:

$$Score\ (C, s) = \sum_{E_i \subseteq s, E_i \in E(C)} \frac{Growth\ \_\ Rate\ (E_i)}{Growth\ \_\ Rate\ (E_i) + 1} \times support\ _C\ (E_i) \qquad (8)$$

Этот выбор авторов очень уязвим. Эта величина была бы равна полной вероятности класса *C* в том случае, если бы ЭП класса на выборке класса *C* были бы независимыми случайными величинами. А это, в свою очередь, может иметь место в том и только в том случае, когда каждый паттерн покрывает некоторое множество данных, которые не покрываются ни одним другим паттерном. Такой случай нереален, поэтому мотивация авторов в этой части является неубедительной. Авторы понимают, что оценка разделительной способности множества паттернов в виде (8) не является вполне корректной. В реальных ситуациях сумма в (8) будет напрямую зависеть от числа ЭП, сгенерированных для класса, а также от меры зависимости паттернов класса между собой. Поэтому строить выбор классификатор по максимуму этой величины нельзя. Для ослабления названных недостатков меры (8) при ее использовании в качестве атрибута классификации авторы предлагают нормировать эту величину по множеству всех классов. В вычислении нормированных значений величин (8) для всех классов состоит основная задача третьего шага построения алгоритма ассоциативной классификации, рассматриваемого здесь.

Эта нормировка выбирается следующим образом. Нормирующий коэффициент строится таким образом, чтобы ослабить влияние того факта, что разные классы могут иметь разное количество ЭП. В качестве нормирующего коэффициента используется величина, которую авторы называют *base_score*(*C*). Она вычисляется как *медиана* множества значений величин *Score*(***C***,*s*) для всех тренировочных данных класса *C*. Медиана отвечает значению величины *Score*(***C***,*s*) для того примера данных класса *C*, для которого 50% всех тренировочных данных этого класса имеют значения больше него. Это значение *Score*(***C***,*s*) для конкретного примера берется в качестве значения нормирующего коэффициента *base_score*(***C***) класса ***C*** при вычислении

нормированного значения функции типа (8), используемой в алгоритме классификации:

$$Norm\_Score(\mathbf{C},s) = score(\mathbf{C},s,) / base\_score(\mathbf{C}). \quad (9)$$

Авторы, однако, замечают, что такой выбор не является обязательным. Их эксперименты показали, что выбор в качестве медианы любого примера в пределах (50–85)% всех примеров не сказывается существенно на качестве классификатора.

Что касается самого алгоритма CAEP, то в нем выбор величины *base_score*(*C*) выполняется автоматически с помощью последовательного повышения величины нижнего порога ρ для значения меры *Growth_Rate*($\underline{E_i}$) и выбора конкретного примера в качестве медианы.

Данный алгоритм проверен авторами на большом числе реальных данных. По их утверждению он показал хорошие свойства как по эффективности поиска ассоциативных правил классификации, так и по качеству решения самих задач классификации данных. В частности, экспериментальные результаты авторов показали, что алгоритм CAEP обладает лучшими свойствами по точности классификации по сравнению с классическим методом C4.5, а также по сравнению с методом CBA, который был рассмотрен ранее.

Однако авторы работ [4, 5], хотя и выражают оптимизм по поводу вычислительной эффективности ассоциативной классификации на основе ЭП, понимают, что достигнутого уровня вычислительной эффективности алгоритмов синтеза таких классификаторов явно недостаточно для работы с *большими данными*. Их исследования в течение последующего десятилетия позволили им существенно улучшить как эффективность, так и точность модели обучения ассоциативной классификации, в основе которой лежит понятие ЭП. Это достигнуто как за счет расширения понятия эмерджентного паттерна, так и за счет повышения эффективности алгоритмов их поиска.

В работе [7] авторы алгоритма CAEP вводят понятие *скачкообразного эмерджентного паттерна* (*Jumping Emergent Pattern, JEP*), далее для краткости *СЭП*, который отличается от понятия ЭП в следующем: скачкообразный эмерджентный паттерн – это ЭП, который в одном множестве данных имеет нулевое значение поддержки, а в другом – строго положительное.

Вообще говоря, правила классификации, которые имеют посылку с ненулевой поддержкой только в одном классе, рассматривались в алгоритмах AQ [12, 13], GK2 [14] и в других алгоритмах, которые брали за основу модусы сходства и различия Д.С.Милля [15, 16]. Новизна идеи использования понятия СЭП состоит только в том, что он рас-

сматривается в контексте задачи ассоциативной классификации. Что касается алгоритма поиска СЭП и построения алгоритма ассоциативной классификации по найденному множеству СЭП (авторы обозначают этот алгоритм аббревиатурой *JEP*), то отличия здесь не носят принципиального характера. Как и в случае алгоритма CAEP, алгоритм *JEP* использует понятие правой и левой границ множества ЭП и сводит поиск СЭП к поиску паттернов, задающих левую границу множества всех ЭП. Такой выбор обосновывается тем, что (а) паттерны, задаваемые элементами левой границы, имеют наименьшую длину на множестве других сравнимых с ними ЭП, а значит, и наибольшее значение меры поддержки по сравнению с ними, и (б) любое подмножество паттернов левой границы уже может не являться СЭП.

Алгоритм поиска СЭП в алгоритме *JEP* строится авторами во многом по аналогии с алгоритмом CAEP и с частичным его использованием. В нем, как уже говорилось, поиск СЭП сводится к поиску элементов левой границы всех ЭП. Эти СЭП авторы называют *наиболее выразительными* (*most expressive*) паттернами. Множество таких паттернов находится для каждого класса $\boldsymbol{C_p} \in \{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_q}\}$, для которого в качестве альтернативного класса выступает множество остальных классов $\overline{\boldsymbol{C}}_{\boldsymbol{p}} \in \{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_q}\} \setminus \boldsymbol{C_p}$. Соответственно, в качестве обучающих данных для класса C_p выступает множество D_p, а для класса $\overline{\boldsymbol{C}}_{\boldsymbol{p}}$ обучающие данные формируются как объединение обучающих данных всех классов, кроме данных класса C_p. Обозначим это множество данных символом $\overline{D}_p$. С учетом введенных обозначений для каждого класса пары $\{C_p, \overline{C}_p\}$ строится множество наиболее выразительных паттернов *MEJEP*($D_p, \overline{D}_p$). На этом обучение ассоциативного классификатора в модели СЭП заканчивается.

Рассмотрим, каким образом в алгоритме *JEP* работает классификатор. Обозначим, как и ранее, символом *s* новую транзакцию, для которой нужно предсказать класс принадлежности. Для решения этой задачи авторы работы [7] для каждого класса $\boldsymbol{C_p} \in \{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_q}\}$ предлагают вычислять значение меры, которую они называют *коллективным влиянием СЭП* (*collective impact, CI*). Эта мера вычисляется по такой формуле:

$$CI(C_p) = \sum_{i:\ СЭП_i \in ME-JEP(C_p, \overline{C}_p)\ \&\ СЭП \subseteq s} supp_{D_p}(СЭП_i) \cdot \qquad (10)$$

Решение принимается в пользу того класса C_p, для которого значение коллективного влияния $CI(C_p)$ максимально. Заметим, что такая модель принятия решения имеет много общего со схемой взвешенного голосования.

Основные различия алгоритмов *JEP* и *CAEP* состоит в том, что в них по-разному выбираются множества ЭП, а также по-иному строятся алгоритмы классификации. В алгоритме *JEP* не используется понятие *GrowthRate*. Оба алгоритма имеют примерно одинаковую точность принятия решений. Оба они, по заключению авторов, превосходят по точности и по вычислительной эффективности алгоритм *CBA* и классический алгоритм *C4.5*.

Алгоритмы *CAEP* и *JEP* имеют свои достоинства и недостатки. Например, алгоритм *CAEP* использует паттерны с некоторым пороговым значением поддержки (например, 1%), а СЭП с такой поддержкой могут не существовать. Хотя алгоритм *JEP* создан как некое развитие алгоритма *CAEP*, он не дает принципиального улучшения как вычислительной эффективности, так и по точности ассоциативной классификации по сравнению с аналогичными характеристиками алгоритма *CAEP*, но авторы выражают определенный оптимизм по поводу его возможностей.

В работе [17] используется некоторая модификация СЭП – СЭП с подсчетом встречаемости (англ. *Jumping Emerging Patterns with Occurrence Counts)*. Для нахождения СЭП в работе также используется алгоритм с построением границ. Авторы экспериментально показывают, что такое расширение СЭП хорошо работает в области классификации изображений.

В своих более поздних работах (см., например, [6] и др.) авторы алгоритмов *CAEP* и *JEP* признают, что хотя число наиболее выразительных СЭП, которые генерируются алгоритмом *JEP*, намного меньше, чем общее их число, тем не менее, даже для данных небольшого объема и размерности их получается слишком много. Например, в одной из задач для данных общим объемом 1000 примеров, которые описываются двадцатью атрибутами, общее число СЭП оказалось равным 32244. Из них было отобрано 2754 наиболее выразительных паттернов, однако и это число слишком велико для эффективной и устойчивой классификации. Попытка дальнейшего снижения числа используемых СЭП может потребовать продолжения фазы обучения на этапе построения классификатора. Для преодоления этих недостатков авторы пошли по пути поиска ЭП других типов, которые присутствуют в данных в меньшем числе, но обладают большей дискриминационной силой.

Авторы [18] также отказываются от использования СЭП в пользу обычных ЭП и экспериментально показывают, что ЭП дают более точную классификацию.

Первый, достаточно естественный, шаг – это введение ограничений на минимальное значение поддержки СЭП, что обеспечивает некоторый минимальный уровень покрытия им тренировочных данных. Такой паттерн авторы называют строгим СЭП и формально определяют его следующим образом [6]. Паттерн $A \in X$ называется *строгим скачкообразным ЭП* (*Strong Jumping Emergent Pattern, SJEP*) из множества D_1 во множество D_2, далее ССЭП, если для него выполнены следующие два условия:

1. $supp_{D_1}(A) = 0$ и $supp_{D_2}(A) \geq \delta$;

2. Любое собственное подмножество элементов паттерна A не удовлетворяет условию 1.

Этот шаг позволил авторам снизить число и повысить выразительность паттернов, на базе которых строится ассоциативный классификатор. Вторым, не менее важным шагом в направлении повышения вычислительной эффективности является введение ими для представления множества паттернов специальной структуры, которая обеспечивает не только эффективное их хранение, но также эффективный просмотр и поиск. Эта структура названа авторами *деревом контрастных паттернов* (*Contrast Pattern Tree Structure, CPT structure*) [6]. Следует заметить, что идея использования такого дерева заимствована ими из работы [19], где похожая структура была предложена для *представления возрастающих паттернов* (*FP-growth Tree*). В структуре *FP-growth Tree* множество паттернов задается в префиксной форме, в которой все паттерны, имеющие общий префикс, представляются общим путем из корня дерева до узла, которому соответствует последний общий символ паттернов (последний символ общего префикса). В работе [6] паттерны генерируются и структурируются точно так же, как и в алгоритме *FP-growth*. Однако содержание каждого узла структуры *CPT* намного богаче, чем содержание узла в дереве *FP-growth Tree*. Оно имеет целью обеспечить лучшую поддержку процессов формирования множеств ССЭП и представление информации о них.

Еще одно новшество структуры *CPT* состоит в том, что представляемые в нем данные предварительно упорядочиваются по отношению $\prec$ следующим образом. Пусть имеются два множества данных, и D_2, относящиеся к разным классам, и множество атрибутов этих данных $X = \{x_1, x_2, \ldots, x_n\}$, где любое $x_i \in \{x_1, x_2, \ldots, x_n\}$ есть одно-

элементное множество, символ паттерна, объект и т.п. Пусть, в соответствии с определением ССЭП, δ– это порог для минимального значения поддержки ССЭП. Определим величину $SupportRatio(x_i)$, называемую отношением поддержек элемента x_i в двух множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$, следующим образом [6]:

$$SupportRatio(x_i) = \begin{cases} 0, \ если \ [supp_{\boldsymbol{D_1}}(\{x_i\}) < \delta] \wedge [supp_{\boldsymbol{D_2}}(\{x_i\}) < \delta], \\ \infty, \ если \ [supp_{\boldsymbol{D_1}}(\{x_i\}) = 0] \wedge [supp_{\boldsymbol{D_2}}(\{x_i\}) \geq \delta] \vee \\ \quad [supp_{\boldsymbol{D_1}}(\{x_i\}) \geq \delta] \wedge [supp_{\boldsymbol{D_2}}(\{x_i\}) = 0], \\ \max\left(\frac{supp_{\boldsymbol{D_2}}(\{x_i\})}{supp_{\boldsymbol{D_1}}(\{x_i\})}, \ \frac{supp_{\boldsymbol{D_1}}(\{x_i\})}{upp_{\boldsymbol{D_2}}(\{x_i\})}\right), \ иначе. \end{cases} \quad (11)$$

Очевидно, что чем больше значение функции *SupportRatio(x_i),* тем лучше разделяющие свойства элемента $\{x_i\}$. Обычно эта функция принимает значения больше 1, кроме тех случаев, когда оба значения поддержки меньше значения порога δ, но такие паттерны являются бесполезными. Если же значение функции для одноэлементного паттерна $\{x_i\}$ равно ∞, то такой одноэлементный паттерн уже является ССЭП.

Отношение $\prec$ определяется в работе [6] следующим образом: для пары одноэлементных паттернов $\{x_i\}$ и $\{x_j\}$ справедливо $\{x_i\} \prec \{x_j\}$, если *SupportRatio(x_i)> SupportRatio(x_j)* или *SupportRatio(x_i)= SupportRatio(x_j), но* $x_i < x_j$ лексикографически.

Далее предполагается, что все элементы паттерна упорядочены в нем по введенному порядку, так что любой паттерн является упорядоченным списком. Аналогичным образом вводится порядок на множестве паттернов произвольной длины. Говорят, что паттерн $\{x_1, x_2, \ldots, x_m\} \prec \{y_1, y_2, \ldots, y_n\}$, если или (1) существует номер i, $1 \leq i \leq m$, такой, что когда $1 < j < i$ $x_j = y_j$, но $x_j \prec y_i$, или (2) для любого j из интервала $1 < j < m$ $x_j = y_j$, но $m < n$.

Дерево контрастных паттернов CPT строится таким образом, чтобы в нем все ветви, исходящие из корня, представляли паттерны, упорядоченные *слева направо* (чем левее паттерн, тем он важнее) и *от корня дерева вниз* – в каждой ветви (более важные одноэлементные паттерны находятся в ветви дерева ближе к корню). Каждый узел дерева представляет собой упорядоченное слева направо подмножество одноэлементных паттернов, каждому из которых поставлено в соответствие значение функции поддержки во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$. Рассмотрим при-

мер дерева *CPT*, заимствованный из работы [6], который поясняет введенное понятие порядка на множестве одноэлементных паттернов, а также структуру *CPT*. Пусть имеются данные, представленные в таблице 1 [6], в которой в последнем столбце экземпляры данных записаны в порядке, предусмотренном введенным отношением $\prec$. Этот порядок легко строится на основе данных таблицы.

Таблица 1. Обучающие данные. Пример заимствован из работы [6]

ID	Метка класса	Данные	Данные, упорядоченные отношением $\prec$
1	$\boldsymbol{D}_1$	{*a,c,d,e*}	{*e,a,c,d*}
2	$\boldsymbol{D}_1$	{*a*}	{*a*}
3	$\boldsymbol{D}_1$	{*b,e*}	{*e,b*}
4	$\boldsymbol{D}_1$	{*b,c,d,e*}	{*e,b,c,d*}
5	$\boldsymbol{D}_1$	{*a,b*}	{*a*,b}
6	$\boldsymbol{D}_1$	{*c,e*}	{*e*,c}
7	$\boldsymbol{D}_1$	{*a,b,c,d*}	{*a,b,c,d*}
8	$\boldsymbol{D}_1$	{*d,e*}	{*e,d*}

Структура данных *CPT* для этого множества имеет вид, представленный на рисунке 3а. В этом дереве все примеры с общим префиксом имеют общий отрезок пути из корня до некоторого узла. Например, экземпляры данных {*e,a,c,d*}, {*e,b*}, {*e,b,c,d*}, {*e,c*} и {*e,d*} имеют общий префикс {*e*}, поэтому все они имеют в дереве общий участок пути, который проходит через узел *e*. Экземпляры данных {*e,b*}, {*e,b,c,d*} имеют общий префикс {*e,b*}, поэтому общим участком в дереве для них является отрезок пути от корня до узла *b*. Каждому узлу $\{x_i\}$ дерева поставлены в соответствие значения функции $supp_{\boldsymbol{D}_i}(\{x_i\})$ во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ для паттернов, отвечающих множеству узлов на пути от корня структуры до соответствующего узла включительно. Заметим, что значения этой функции представлены в терминах абсолютных значений, т.е. в числе экземпляров данных, в котором соответствующий паттерн встречается в них. Каждому узлу поставлено в соответствие два таких числа. Одно из них представляет значение функции $supp_{\boldsymbol{D}_1}(\{x_i\})$, а другое – значение функции $supp_{\boldsymbol{D}_2}(\{x_i\})$. В работе [6] дано строгое описание алгоритма построения *CPT* для данных двух множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ в псевдокоде, поэтому здесь описание этого алгоритма опускается.

Структура *CPT* для представления данных используется далее алгоритмом генерации ССЭП. Формальное описание этого алгоритма

дано в работе [6] в псевдокоде, поэтому здесь ограничимся только его содержательными пояснениями на примере.

Для идентификации некоторого поддерева *CPT* в интересах последующего поиска и просмотра его содержания будем использовать имя корневого узла этого поддерева в форме префикса, т.е. в форме последовательности имен узлов от корня *CPT* до корневого узла поддерева. Например (рисунок 3а), идентификатор *R.e* ссылается на поддерево, корнем которого является узел *e*. Легко понять, например, о каком поддереве идет речь, если его корень имеет идентификатор *R.ea* или *R.ab*. Значение поддержки любого паттерн, заданного своим префиксом, во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$, указывается в соответствующем узле *CPT* явно (см. рисунок 3а) для обоих множеств.

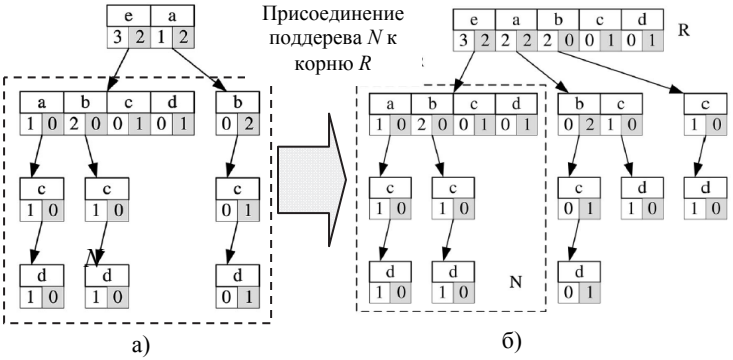

Рис. 3. а) Исходное представление данных табл. 1 в виде дерева контрастных паттернов. Рисунок заимствован из работы [6]; б) Скорректированное представление данных табл. 1 в виде дерева контрастных паттернов. Рисунок заимствован из работы [6]

В качестве входных данных алгоритм генерации множества ССЭП использует дерево *CPT*. Заметим, что в *CPT*–дереве на рисунок 3а одинаковые паттерны встречаются в нескольких узлах, но с разными префиксами. Например, это касается паттерна $\{a\}$, который встречается в узлах *R.a* (с пустым префиксом) и *R.ea* (с префиксом *e*). В таком дереве перечислять паттерны неудобно. Поэтому для решения задачи генерации паттернов авторы предлагают выполнять преобразование *CPT* –дерева, представляющего данные множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$, в более удобную форму. Заметим, что в алгоритме генерации ССЭП это преобразование является частью алгоритма и вызывается им по мере необходимости.

Поясним только общую идею этого преобразования, поскольку детальное его объяснение заняло бы слишком много места. В рабо-

те [6] этот алгоритм представлен в псевдокоде. Преобразование выполняется обходом узлов *CPT* – структуры в соответствии со стратегией *поиск в глубину*. Это означает, что обход узлов (на рисунок 3а) выполняется путем последовательного просмотра паттернов $\{e\}$, $\{e,a\}$, $\{e,a,c\}$, $\{e,a,c,d\}$, $\{e,b\}$, $\{e,b,c\}$, $\{e,b,c,d\}$, $\{e,c\}$ $\{e,d\}$, $\{a\}$, $\{a,b\}$, $\{a,b,c\}$ и $\{a,b,c,d\}$. Но заметим, что поддерево с корнем *R.e* содержит паттерны, которые встречаются и в поддереве *R.a*. Очевидно, что поддержка таких паттернов должна быть суммирована по множеству всех узлов, где они встречаются. С этой целью авторами работы [6] используется операция *присоединения поддеревьев* (*merge of subtrees*). Например, на рисунок 3а поддерево, обозначенное символом *N* (в ранее принятых терминах это дерево обозначалось бы символом *R.e*) присоединено к корню дерева *R* на рис. 3б. В полученной структуре в корне *R* суммируются значения поддержек одноэлементных паттернов, которые они имеют в исходном и в присоединенном деревьях. В рассматриваемом примере оказывается достаточным однократное использование операции присоединения поддеревьев. В алгоритме генерация ССЭП эта операция используется рекурсивно, когда это необходимо, по мере обхода узлов *CPT*–дерева в глубину и слева направо.

Напомним, что последовательность представления одноэлементных паттернов в любом узле дерева, как и порядок их следования вдоль ветвей дерева, определяются отношением $\prec$, а этот порядок на множестве одноэлементных паттернов вычисляется на основе данных множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$. Использование такого упорядочивания одноэлементных паттернов в узлах и в глубину дерева приводит к тому, что самые выразительные (самые полезные) паттерны располагаются ближе к корню *CPT*–дерева и в его ветвях, лежащих левее. С учетом этого свойства в алгоритме генерации ССЭП обычно требуется просмотр только небольшой части этого дерева в левой верхней его области. Покажем это на примере.

Примем в примере минимально допустимое значение этой поддержки $\delta = 2$. Поиск ССЭП начинается с анализа паттерна $\{e\}$, отвечающего корню *CPT*–дерева *R.e*, который имеет поддержку [3,2] во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ соответственно. Для него функция $SupportRatio(e)$ равна *1,5,* и поскольку она не равна бесконечности, то паттерн $\{e\}$ не может быть ССЭП–паттерном. Но так как функция $SupportRatio(e)$ не является антимонотонной по длине паттерна, то возможно, что символ *e* встречается в ССЭП большей длины. Поэтому просмотр ветвей поддерева *R.e* в глубину следует продолжить. При таком просмотре далее нужно анализировать узел *R. ea.* Но паттерн $\{ea\}$ имеет максимальное значение поддержки в одном из множеств, равное 1, а потому не является ССЭП. То же самое будет иметь место и для всех паттернов, которые могут находиться в поддереве с корнем

R. ea, поскольку значение поддержки вдоль ветвей дерева не может увеличиваться. Поэтому поддерево *R. ea* далее не рассматривается.

Очередным узлом для анализа является узел *R. eb* [2,0]. Ему соответствует паттерн {*eb*}, который удовлетворяет определению ССЭП для множества $\boldsymbol{D_1}$. Таким образом, один ССЭП уже найден, это паттерн {*eb*}[2,0]. Дальнейший просмотр ветвей поддерева *R. eb* в глубину не имеет смысла, поскольку паттерны, которые в нем могут встретиться, имеют большую длину, чем паттерн {*eb*}, а потому не могут иметь большее значение поддержки.

Следующий узел для анализа – это узел *R.ec.* С учетом всех случаев паттернов, которые начинаются символом *c,* узел *R. ec* имеет поддержку [2,1] во множествах $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$, соответственно. Паттерн {*ec*} не является ССЭП, поскольку его поддержка не равна нулю в одном из множеств $\boldsymbol{D_1}$ или $\boldsymbol{D_2}$. Но это не исключает, что более длинный паттерн по одной из ветвей, исходящих из узла *R. ec*, может таковыми оказаться для множества $\boldsymbol{D_1}$, в котором он имеет допустимое значение поддержки. Действительно, паттерн {*ecd*}[2,0] удовлетворяет определению ССЭП в пользу множества $\boldsymbol{D_1}$.

При дальнейшем обходе дерева будет найден еще один ССЭП, а именно {*ab*}[0,2]. Таким образом, для множества данных, представленных в табл. 1, алгоритм находит следующие ССЭП: {*eb*}[2,0], {*ecd*}[2,0] и {*ab*}[0,2].

Псевдокод алгоритма генерации ССЭП по данным, представленным структурой CPT–дерева, может быть найден в работе [6].

Достоинство описанного алгоритма состоит в его эффективности. Кроме того, его достоинство состоит также в том, что он выполняет поиск ССЭП одновременно для двух множеств $\boldsymbol{D_1}$ и $\boldsymbol{D_2}$ за один проход всей базы данных. В этом его неоспоримое преимущество по сравнению с алгоритмами поиска ЭП и СЭП, в которых этот поиск делается поочередно с использованием представления множества эмерджентных паттернов с помощью верхней и нижней границ.

Дальнейшие усилия авторов данного направления были направлены на поиск других типов эмерджентных паттернов, которые могли бы лучше подойти для построения ассоциативных классификаторов и которые могут быть эффективно сгенерированы. Один из предложенных вариантов несколько обобщает понятие ССЭП, допуская, что поддержка таких паттернов не обязательно должна быть равна нулю в одном из классов. Оправданием такого допущения является возможность зашумления ССЭП, которое приведет к незначительному отклонению поддержки ССЭП в одном из классов от нуля. Это предположение представляется достаточно разумным, поскольку на тестовых данных практически всегда ССЭП имеет ненулевую поддержку в соответствующем классе. Заметим, что то же самое имеет место при тестировании правил, полученных алгоритмами индуктивного обучения, например, алгорит-

мами *AQ* [12, 13] или *GK2* [14]. Такие паттерны авторы называют *эмерджентными паттернами, устойчивыми к шуму* (*Noise-tolerant EPs, NEPs*). В формальном определении NEP допускается, что порог его поддержки в одном из множеств не превышает заданной малой величины, а порог поддержки в другом, наоборот, не меньше, чем заданное пороговое значение. Авторы вводят также понятие *обобщенного эмерджентного паттерна, устойчивого к шуму* (*Generalized Noise–tolerant EP, GNEP*). Обобщение состоит в том, что вместо традиционной меры различительной силы паттернов, задаваемой функцией *GrowthRate*, авторы допускают использование некоторых функций от значений продержки в двух множествах. Однако конструктивность и полезность такого понятия авторами не мотивируется.

Что касается использования полученных ССЭП в алгоритме ассоциативной классификации, то в этом отношении авторы не предлагают чего-либо нового и рассматривают варианты, аналогичные тем, что были предложены ими и другими исследователями в данной области.

Эффективность и качество алгоритмов генерации ССЭП, а также их использование в задачах классификации были тщательно исследованы авторами на большом количестве наборов данных. Полученные экспериментальные результаты сравнивались с результатами, полученными для наиболее известных алгоритмов обучения и классификации, например, для *CBA*, *C4.5* и его версиями, улучшенными за счет бустинга, и другими алгоритмами. Во всех случаях алгоритм *SJEP* оказывался существенно лучше по эффективности и показывал, в среднем, лучшие результаты по точности классификации. Причем эти оценки были получены на десятках различных наборов данных из UCI–репозитория [20]. Алгоритм *SJEP* сравнивался также с алгоритмом *JEP* и показал в среднем десятикратное ускорение решения задач и сравнимую точность при меньшем числе используемых паттернов. Несколько более осторожно авторы делают заключение о перспективности использования паттернов, которые являются обобщением ССЭП, а именно эмерджентных паттернов, устойчивых к шуму и их обобщений.

Очевидно, что модели ассоциативной классификации на основе различных видов эмерджентных паттернов являются важным шагом в области построения эффективных моделей классификации при работе с большими данными. Представляется, однако, что на текущий момент они исчерпали свои возможности по дальнейшему повышению эффективности. Одной из причин для такого заключения относительно модели, основанной на использовании ЭП, является необходимость сведения в ней любых данных к модели булевых данных. Такое преобразование всегда приводит к заметному увеличению размерности задачи, а значит, ставит дополнительные ограничения на возможности такого подхода при работе с большими данными.

3. Заключение. Появление модели эмерджентного паттерна, заимствованного из классической теории индуктивного обучения, было существенным шагом вперед в области поиска ассоциативных правил классификации. Это обусловлено тем, что в этой модели сама задача поиска ассоциативных правил классификации была сформулирована с учетом прагматики задачи классификации, суть которой состоит в том, что требуется построить правила, которые могли бы отделить экземпляры данных одного класса от экземпляров данных других классов. Эта прагматика была явно встроена в понятие ЭП. Особенно точно это выражено в понятии ССЭП. Это позволило авторам в дальнейшем сосредоточиться на эффективности алгоритмов поиска ЭП. Другое большое достижение в этой части – это введение структуры *CPT*–дерева для экономного представления данных обучения и построения эффективных алгоритмов их использования в процессах генерации ССЭП, отвечающих посылкам выразительных правил ассоциативной классификации. Обратим внимание на тот факт, что структура *CPT*–дерева, хотя первоначальная идея ее использования принадлежит и не авторам работ в области алгоритмов генерации ССЭП, является чрезвычайно продуктивной идеей в области обучения классификации. По-видимому, это дерево при некоторой его модификации может быть использовано в алгоритмах поиска минимальных правил в задачах индуктивного обучения при решении задач типа оптимального покрытия [12-14]. Оно может найти и другие применения.

Однако вопрос о пределах и перспективах использования моделей и алгоритмов ассоциативной классификации в задачах анализа больших данных в настоящее время вряд ли имеет однозначный ответ. К основному недостатку такой модели следует отнести ее ограниченные возможности при работе с гетерогенными данными сложной структуры, с данными, представленными текстами на естественном языке, изображениями и т.п. Модель ассоциативной классификации хорошо подходит для работы с дискретными данными (булевыми, номинальными и целочисленными). В случае других типов данных принципиальной проблемой становится проблема трансформации реальных данных к дискретной модели. Пути решения этой проблемы известны, однако их вряд ли можно считать удовлетворительными. В любом случае известные способы такого преобразования приводят к существенному увеличению размерности данных, возможно в десятки и сотни раз. А проблема размерности является ключевой проблемой больших данных и без их дискретизации.

Другая проблема, которая осложняет использование имеющихся моделей и алгоритмов ассоциативной классификации, вызвана тем фактом, что связь, которая называется ассоциацией, носит чисто синтаксический характер. Она является ненаправленной связью и потому ее нельзя интерпретировать как причинно–следственную связь, если

не обосновывать это методами или не использовать метрики, которые специально для этого разрабатываются. Во многих случаях ассоциация возникает как следствие "третьих факторов", от которых зависят и посылка и заключение ассоциативного фактора. Источником семантически нелепых связей, которые при этом могут быть обнаружены, является чисто синтаксический характер ассоциаций.

Анализируя способы построения классификаторов на основе ассоциативных правил, полученных с помощью процедур обучения, легко заметить, что такие способы базируются на здравом смысле, на эвристиках, на введении специальных метрик для оценки разделительной способности классификаторов. Для каждого варианта выбора модели объединения решений, даваемых различными правилами (по крайней мере, из тех вариантов, которые были рассмотрены в данной работе), всегда легко построить пример, когда предложенный вариант совсем не подходит, или когда предложенная эвристика не работает. Эта проблема достаточно хорошо известна, и она детально анализируется в теории объединения решений, которые вырабатываются множеством классификаторов, а каждое ассоциативное правило может интерпретироваться как простейший пример классификатора. Она называется проблемой разнообразия классификаторов (см., например, ее анализ в работе [21]). От успешности решения этой проблемы во многом будет зависеть успешность модели ассоциативной классификации при работе с большими данными.

Еще одна проблема, которая свойственна задаче принятия решений, в частности, задаче классификации при работе с большими данными, это учет контекста обучающих данных и, соответственно, контекста экземпляра данных, подлежащего классификации. Эта проблема исследуется особо в современной литературе по интеллектуальной обработке больших данных, в частности, она исследуется в задачах классификации, решаемых рекомендующими системами (см., например, работу [22]). Как известно, одним из предлагаемых решений проблемы учета контекста в задачах классификации больших данных является использование онтологий. Представляется, что модель ассоциативной классификации может успешно интегрироваться с моделью онтологического представления гетерогенных данных большого объема и размерности.

В заключение можно утверждать, что интеграция идей индуктивного обучения и ассоциативного анализа данных для построения моделей принятия решений при работе с большими данными, в частности, ассоциативной классификации представляется достаточно перспективной идеей. Эта интеграция, возможно, уже в ближайшее время сможет дать новый толчок в направлении эффективного решения проблемы обучения в задачах классификации и синтеза классификаторов применительно к большим данным. Однако успех будет сильно зави-

сеть от того, насколько эффективно удастся преодолеть отмеченные выше проблемы.

Литература

1. *Городецкий В.И., Тушканова О.Н.* Ассоциативная классификация: аналитический обзор. Часть 1 // Труды СПИИРАН. 2015. №1(38). С. 183–203.
2. *Городецкий В.И., Самойлов В.В.* Ассоциативный и причинный анализ и ассоциативные байесовские сети // Труды СПИИРАН. 2009. №9. С. 13–65.
3. *Adamo J.-M.* Data Mining for Association Rules and Sequential Patterns // Springer. 2000.
4. *Dong G., Li J.* Efficient Mining of Emerging Patterns: Discovering Trends and Differences // Proc. of the KDD'99. 1999. pp. 43–52.
5. *Dong G., Zhang X., Wong L., Li J.* CAEP: Classification by Aggregating Emerging Patterns // Proc. of the DS'99. 1999. pp. 30–42.
6. *Fan H., Ramamohanarao K.* Fast Discovery and the Generalization of Strong Jumping Emerging Patterns for Building Compact and Accurate Classifiers // IEEE Trans. Knowl. Data Eng. 2006. vol. 18(6). pp. 721–737.
7. *Li J., Dong G., Ramamohanarao K.* Making use of the most expressive jumping emerging patterns for classification // Proc. of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan. 2000. pp. 220–232.
8. *Bayardo Jr. R.J.* Efficiently Mining Long Patterns from Databases // Proc. of the SIGMOD Conference. 1998. pp. 85–93.
9. *Agrawal R., Sricant R.* Fast Algorithm for Mining Association rules // Proc. of the 20th Intern. Conference on Very Large Databases. Santiago, Chile. 1994. pp. 68–77.
10. *Condorcet N.C.* Essaisur l'application de l'analyse a la probabilite des decisions rendues a la pluralité des voix. Paris: Imprimerie Royale. 1785.
11. Condorcet's jury theorem. Wikipedia.org: the free encyclopedia // URL: http://en.wikipedia.org/wiki/Condorcet's_jury_theorem (дата обращения 20.06.2014 г.).
12. *Michalski R.S.* On the Quasi-Minimal Solution of the General Covering Problem// Proc. of the V International Symposium on Information Processing (FCIP-69), Bled, Yugoslavia. 1969. vol. A3. pp .125–128.
13. *Michalski R.S.* A Theory and Methodology of Inductive Learning. Machine Learning, vol.1 // Eds. Carbone J.G., Michalski R.S., Mitchel T.M. Tigoda. Palo Alto. 1983. pp. 83–134.
14. *Gorodetsky V., Karsaev O., Samoilov V.* Direct Mining of Rules from Data with Missing Values // Studies in Computational Intelligence. Chapter in book. Eds. Lin T.Y., Ohsuga S., Liau C.J., Hu X.T., Tsumoto S.. Foundation of Data Mining and Knowledge Discovery. Springer. 2005. vol. 6. pp. 233–264.
15. *Милль Дж.Ст.* Система логики силлогистической и индуктивной: Изложение принципов доказательства в связи с методами научного исследования Пер. с англ. // Изд. 5, испр. и доп. М.: ЛЕНАНД, 2011.
16. Пять канонов Джона Милля. Vikent.ru – портал И.Л. Викентьева // URL: http://vikent.ru/enc/834/ (дата обращения 20.06 2014 г.).
17. *Kobylinski L., Walczak K.* Efficient Mining of Jumping Emerging Patterns with Occurrence Counts for Classification // Transactions on Rough Sets XIII. LNCS 6499. 2011. pp. 73–88.
18. *Sherhod R., Judson P.N., et al.* Emerging Pattern Mining To Aid Toxicological Knowledge Discovery // Journal of Chemical Information Modeling. 2014. no. 54 (7). pp 1864–1879.
19. *Han J., Pei J., Yin Y.* Mining frequent patterns without candidate generation // Proc. of the ACM SIGMOD Intern. Conf. on Management of Data. 2000. pp. 1–12.

20. *Blake C.L., Murphy P.M.* UCI Repository of machine learning database. University of California, Department of Information and Computer Science. Irvine, CA. 1998 // URL: http://www.cs.uci.edu/mleam/mlrepository.html (дата обращения 20.06.2014).
21. *Городецкий В.И., Серебряков С.В.* Методы и алгоритмы коллективного распознавания // Автоматика и Телемеханика. 2008. № 11. С. 3–40.
22. *Gorodetsky V., Samoylov V., Serebryakov S.* Ontology–based Context–dependent Personalization Technology // Proc. of the WI/IAT 2010. Toronto. 2010. pp. 278–283.

References

1. Gorodetsky V.I., Tushkanova O.N. [Associative classification: analytical overview. Part 1]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2015. no. 1(38). pp. 183–203. (In Russ.).
2. Gorodetsky V.I., Samoylov V.V. [Associative and causal analysis and associative Bayesian networks]. *Trudy SPIIRAN - SPIIRAS Proceedings*. 2009. no. 9. pp. 13–65. (In Russ.).
3. Adamo J.-M. Data Mining for Association Rules and Sequential Patterns. Springer, 2000.
4. Dong G., Li J. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. Proc. of the KDD'99. 1999. pp. 43–52.
5. Dong G., Zhang X., Wong L., Li J. CAEP: Classification by Aggregating Emerging Patterns. Proc. of the DS'99. 1999. pp. 30–42.
6. Fan H., Ramamohanarao K. Fast Discovery and the Generalization of Strong Jumping Emerging Patterns for Building Compact and Accurate Classifiers. IEEE Trans. Knowl. Data Eng. 2006. vol. 18(6). pp. 721–737.
7. Li J., Dong G., Ramamohanarao K. Making use of the most expressive jumping emerging patterns for classification. Proc. of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan, 2000. pp. 220-232.
8. Bayardo R.J.Jr. Efficiently Mining Long Patterns from Databases. Proc. of the SIGMOD Conference. 1998. pp. 85–93.
9. Agrawal R., Sricant R. Fast Algorithm for Mining Association rules. Proc. of the 20th Intern. Conference on Very Large Databases. Santiago, Chile. 1994. pp. 68–77.
10. Condorcet N.C. Essaisur l'application de l'analyse a la probabilite des decisions rendues a la pluralité des voix. Paris: Imprimerie Royale. 1785.
11. Condorcet's jury theorem. Wikipedia.org: the free encyclopedia. Available at: http://en.wikipedia.org/wiki/Condorcet's_jury_theorem (accesses: 20.06.2014).
12. Michalski R.S. On the Quasi-Minimal Solution of the General Covering Problem. Proc. of the V International Symposium on Information Processing (FCIP-69), Bled, Yugoslavia. 1969. vol. A3. pp .125–128.
13. Michalski R.S. A Theory and Methodology of Inductive Learning. Machine Learning, vol.1. Eds. Carbone J.G., Michalski R.S., Mitchel T.M. Tigoda. Palo Alto. 1983. pp. 83–134.
14. Gorodetsky V., Karsaev O., Samoilov V. Direct Mining of Rules from Data with Missing Values. Studies in Computational Intelligence, Chapter in book. Eds. Lin T.Y., Ohsuga S., Liau C.J., Hu X.T., Tsumoto S. Foundation of Data Mining and Knowledge Discovery. Springer. 2005. vol. 6. pp. 233–264.
15. Mill J.S. *Sistema logiki sillogisticheskoy i induktivnoy: Izlozheniye printsipov dokazatel'stva v svyazi s metodami nauchnogo issledovaniya* [System of syllogistic logic and inductive: Statement of principles of proof in relation to the methods of scientific investigation]. Moscow: LENAND. 2011. 832 p. (In Russ.).
16. Five Canons of John Mill. Vikent.ru - portal I.L. Vikent'yeva [Vikent.ru - portal of I.L.Vikent'yev]. Available at: http://vikent.ru/enc/834/ (accessed: 20.06 2014). (In Russ.).
17. Kobylinski L., Walczak K. Efficient Mining of Jumping Emerging Patterns with Occurrence Counts for Classification. Transactions on Rough Sets XIII. LNCS 6499. 2011. pp. 73–88.

18. Sherhod R., Judson P.N., et al. Emerging Pattern Mining To Aid Toxicological Knowledge Discovery. Journal of Chemical Information Modeling. 2014. no. 54(7). pp. 1864–1879.
19. Han J., Pei J., Yin Y. Mining frequent patterns without candidate generation. Proc. of the ACM SIGMOD Intern. Conf. on Management of Data. 2000. pp. 1–12.
20. Blake C.L., Murphy P.M. UCI Repository of machine learning database. University of California, Department of Information and Computer Science. Irvine, CA. 1998. Available at: http://www.cs.uci.edu/mleam/mlrepository.html (accessed: 20.06.2014).
21. Gorodetsky V., Serebryakov S. [Methods and algorithms for collective recognition] *Avtomatika i Telemehanika – Automation and Remote Control*. 2008. no. 11. pp. 3–40. (In Russ.).
22. Gorodetsky V., Samoylov V., Serebryakov S. Ontology–based Context–dependent Personalization Technology. Proc. of the WI/IAT 2010. Toronto. 2010. pp. 278–283.

Тушканова Ольга Николаевна — аспирант, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук. Область научных интересов: машинное обучение, интеллектуальный анализ данных, извлечение знаний, многоагентные системы, рекомендующие системы, облачные технологии, онтологии. Число научных публикаций — 12. tushkanova.on@gmail.com; 14 линия, д. 39, Санкт-Петербург, 199178; р.т.: +79817343119.

Tushkanova Olga Nikolaevna — Ph.D. student, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences. Research interests: data mining, multi-agent systems, recommender systems, cloud computing, ontologies, knowledge extraction technologies. The number of publications — 12. tushkanova.on@gmail.com; 39, 14-th Line, St. Petersburg, 199178, Russia; office phone: +79817343119.

Городецкий Владимир Иванович — д-р техн. наук, заведующий лабораторией интеллектуальных систем, Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук. Область научных интересов: искусственный интеллект, технология многоагентных систем, распределенное обучение, извлечение знаний из баз данных, анализ и объединение данных различных источников, P2P сети принятия решений и P2P методы извлечения знаний из данных, обработка больших данных, планирование и составление расписаний, алгоритмы улучшения изображений, рекомендующие системы. Число научных публикаций — 200. gor@mail.iias.spb.su; 14 линия, д. 39, Санкт-Петербург, 199178; р.т.: +7-812-328-3311.

Gorodetski Vladimir Ivanovich — Ph.D., head of laboratory of intelligent systems, St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences. Research interests: intelligent data analysis, information fusion, P2P data mining and machine learning, multi-agent systems technology and software tools, agent-based applications, recommender systems, mobile image enhancement. The number of publications — 200. gor@mail.iias.spb.su; 39, 14-th Line, St. Petersburg, 199178, Russia; office phone: +7-812-328-3311.

РЕФЕРАТ

Городецкий В.И., Тушканова О.Н. **Ассоциативная классификация: аналитический обзор. Часть 2.**

В данной работе продолжается рассмотрение методов и алгоритмов ассоциативной классификации. В работе кратко дается постановка задачи ассоциативной классификации. В основной части работы выполнены анализ и сравнение современных моделей, методов и алгоритмов, разработанных в области ассоциативной классификации, основанной на эмерджентных паттернах, применительно к работе с данными большого объема. В заключении формулируются достоинства и недостатки методов ассоциативной классификации и дается оценка перспектив использования этого подхода для интеллектуального анализа больших данных.

SUMMARY

Gorodetsky V., Tushkanova O. **Associative Classification: Analytical Overview. Part 2.**

The paper continues the review of associative classification intended for processing of big data. It shortly formulates corresponding problem statement of associative classification. The main part of the paper represents an overview and comparative analysis of the modern methods, models and algorithms developed for associative classification based on emerging patterns. In conclusion, the paper outlines the main advantages and drawbacks of associative classification, as well as evaluates its capabilities from big data processing perspective.