Математические и инструментальные методы машинного обучения

9. Методы поиска ассоциативных правил

Поиск ассоциативных правил

Ассоциативные правила позволяют находить закономерности между связанными событиями. Примером такого правила служит утверждение, что покупатель, приобретающий «Хлеб», приобретет и «Молоко». Впервые эта задача была предложена для поиска ассоциативных правил для нахождения типичных шаблонов покупок, совершаемых в супермаркетах, поэтому иногда ее еще называют анализом рыночной корзины (*market basket analysis*).

Пусть имеется база данных, состоящая из покупательских транзакций. Каждая **транзакция — это набор товаров, купленных покупателем за один визит**. Такую транзакцию еще называют рыночной корзиной. Целью анализа является установление следующих зависимостей: если в транзакции встретился некоторый набор элементов X, то на основании этого можно сделать вывод о том, что другой набор элементов Y также должен появиться в этой транзакции. Установление таких зависимостей дает возможность находить очень простые и интуитивно понятные правила.

Понятие ассоциативного правила

Пусть дан контекст $\mathbb{K} := (G, M, I)$, где G — множество объектов, M — множество признаков (items), $I \subseteq G \times M$
Ассоциативным правилом контекста $\mathbb{K}$ называется выражение вида $A \to B$, где $A, B \subseteq M$.

Поддержка и достоверность

Основными характеристиками таких правил являются **поддержка** и **достоверность**. Правило "Из X следует Y" имеет поддержку s, если $s\%$ транзакций из всего набора содержат наборы элементов X и Y. Достоверность правила показывает, какова вероятность того, что из X следует Y. Правило «Из X следует Y» справедливо с достоверностью c, если $c\%$ транзакций из всего множества, содержащих набор элементов X, также содержат набор элементов Y. Покажем на конкретном примере: пусть 75% транзакций, содержащих хлеб, также содержат молоко, а 3% от общего числа всех транзакций содержат оба товара. 75% — это достоверность правила, а 3% —это поддержка.

Поддержка и достоверность

Алгоритмы поиска ассоциативных правил предназначены для нахождения всех правил вида «из X следует Y», причем поддержка и достоверность этих правил должны находиться в рамках некоторых наперед заданных границ, называемых соответственно минимальной и максимальной поддержкой и минимальной и максимальной достоверностью. Пороговые значения параметров поддержки и достоверности выбираются таким образом, чтобы ограничить количество найденных правил.

Определение

Поддержкой (support) ассоциативного правила $A \to B$ называется величина $supp(A \to B) = \frac{|(A \cup B)'|}{|G|}$.

Значение $supp(A \to B)$ показывает какая доля объектов G содержит $A \cup B$. Часто поддержку выражают в $\%$.

Достоверностью (confidence) ассоциативного правила $A \to B$ называется величина $conf(A \to B) = \frac{|(A \cup B)'|}{|A'|}$.

Значение $conf(A \to B)$ показывает какая доля объектов обладающих A также содержит $A \cup B$. Величину поддержки также часто выражают в $\%$.

Расчёт поддержки и достоверности

Клиент \ Товар	Пиво	Чипсы	Молоко	Мюсли	Пряники
Клиент 1	1	1	0	0	0
Клиент 2	0	0	1	1	1
Клиент 3	1	1	1	1	0
Клиент 4	1	1	1	0	1
Клиент 5	0	1	1	1	1

$Supp(\{Пиво\} \rightarrow \{Чипсы\}) = |3| / |5| = 60\%$

$Conf(\{Пиво\} \rightarrow \{Чипсы\}) = |3| / |3| = 100\%$

$Supp(\{Мюсли, Пряники\} \rightarrow \{Молоко\}) = |2| / |5| = 40\%$

$Conf(\{Мюсли, Пряники\} \rightarrow \{Молоко\}) = |2| / |2| = 100\%$

Алгоритм Apriori

Поскольку $\varphi(x) = \bigwedge_{f \in \varphi} f(x)$ — конъюнкция, имеет место

Свойство антимонотонности:
для любых $\psi, \varphi \subset \mathscr{F}$ из $\varphi \subset \psi$ следует $\nu(\varphi) \geqslant \nu(\psi)$.

Следствия:

1. если ψ частый, то все его подмножества $\varphi \subset \psi$ частые.
2. если φ не частый, то все наборы $\psi \supset \varphi$ также не частые.
3. $\nu(\varphi \cup \psi) \leqslant \nu(\varphi)$ для любых φ, ψ.

Два этапа поиска ассоциативных правил:

1. поиск частых наборов
   (многократный просмотр транзакционной базы данных).
2. выделение ассоциативных правил
   (простая эффективная процедура в оперативной памяти).

Поиск частых наборов (frequent item-set)

Вход: X^ℓ — обучающая выборка;
минимальная поддержка δ; минимальная значимость $\varkappa$;
Выход: $R = \{(\varphi, y)\}$ — список ассоциативных правил;

1: множество всех частых исходных признаков:
$G_1 := \{f \in \mathscr{F} \mid \nu(f) \geqslant \delta\}$;
2: **для всех** $j = 2, \ldots, n$
3: множество всех частых наборов мощности j:
$G_j := \{\varphi \cup \{f\} \mid \varphi \in G_{j-1},\ f \in G_1 \setminus \varphi,\ \nu(\varphi \cup \{f\}) \geqslant \delta\}$;
4: **если** $G_j = \varnothing$ **то**
5: **выход** из цикла по j;
6: $R := \varnothing$;
7: **для всех** $\psi \in G_j,\ j = 2, \ldots, n$
8: AssocRules $(R, \psi, \varnothing)$;

Выделение ассоциативных правил

Этап 2. Простой алгоритм, выполняемый быстро, как правило, полностью в оперативной памяти.

Вход и Выход: R — список ассоциативных правил;
(φ, y) — ассоциативное правило;

1: **ПРОЦЕДУРА** AssocRules (R, φ, y);
2: **для всех** $f \in \varphi$: $\mathrm{id}_f > \max\limits_{g \in y} \mathrm{id}_g$ (чтобы избежать повторов y)
3: $\varphi' := \varphi \setminus \{f\}$; $y' := y \cup \{f\}$;
4: **если** $\nu(y'|\varphi') \geqslant \varkappa$ **то**
5: добавить ассоциативное правило (φ', y') в список R;
6: **если** $|\varphi'| > 1$ **то**
7: AssocRules (R, φ', y');

---

id_f — порядковый номер признака f в $\mathscr{F} = \{f_1, \ldots, f_n\}$

Схема работы алгоритма Apriori/1

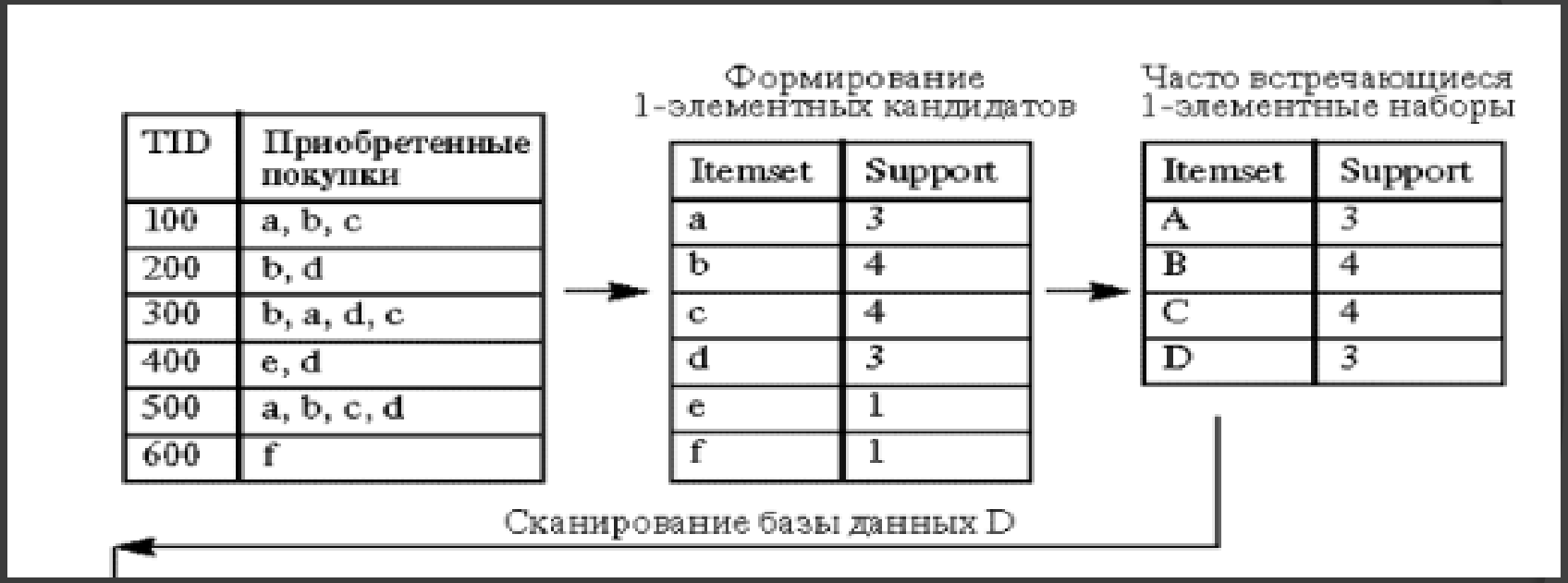


Схема работы алгоритма Apriori/2

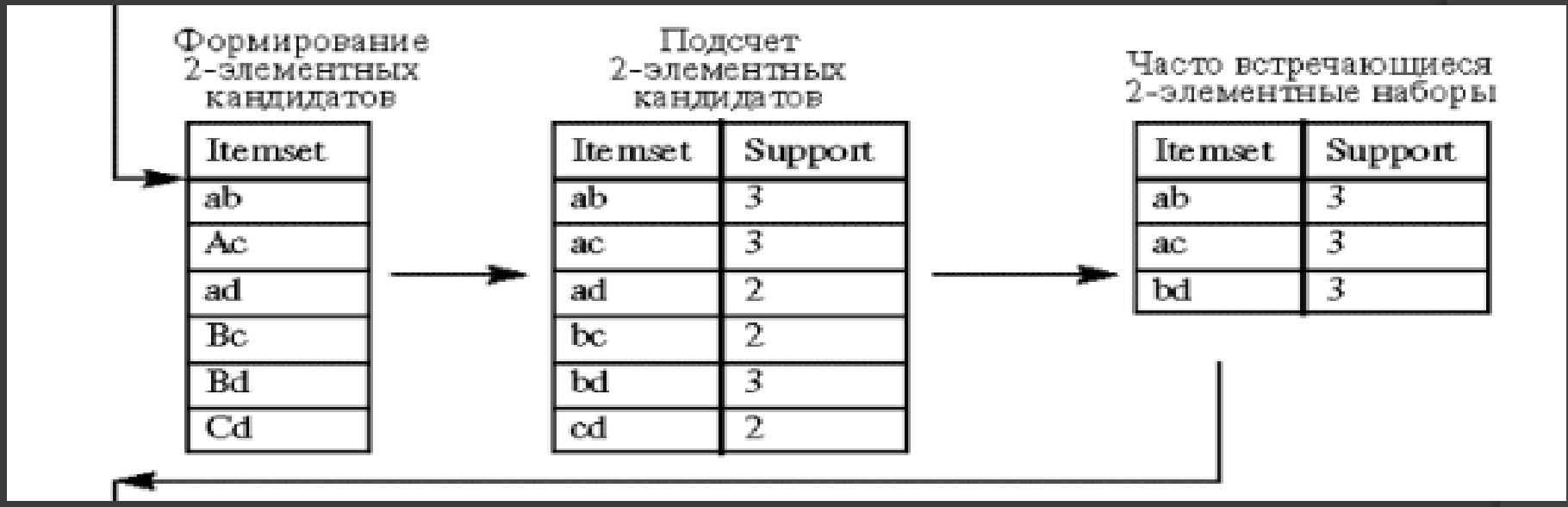
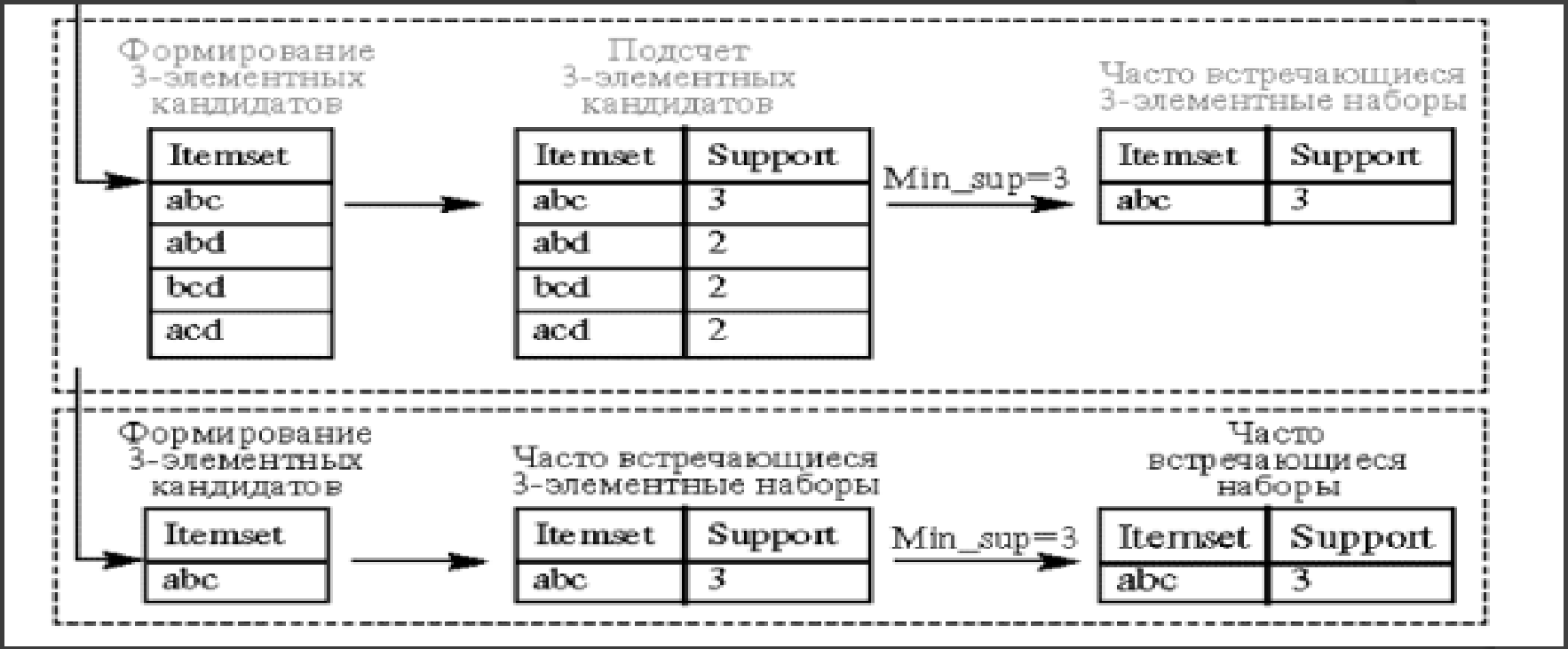


Схема работы алгоритма Apriori/3



Поиск последовательных шаблонов (Sequential Pattern Mining)

Использование шаблонов позволяет находить закономерности между связанными во времени событиями. Одним из примеров области применения таких знаний может послужить интернет-торговля. Поиск наиболее частых последовательностей покупок позволяет получить информацию о том, через какой промежуток времени после покупки товара “А” человек наиболее склонен купить товар “Б”, или в какой последовательности приобретаются товары. Получаемые закономерности в действиях покупателей можно использовать для персонализации клиентов, формирования более выгодного предложения, стимулирования продаж определенных товаров или управления запасами.

Поиск последовательных шаблонов (Sequential Pattern Mining)

Другим примером области применения задачи SPM может послужить веб-аналитика. Зная наиболее популярные последовательности переходов по страницам, можно размещать на соответствующих страницах определенный контент или изменить структуру сайта с целью более быстрого и удобного доступа к некоторым страницам.

Постановка задачи SPM

Рассмотрим постановку задачи поиска последовательных шаблонов, используя задачу анализа рыночной корзины, поскольку именно так она была изначально сформулирована в основополагающей статье Агравала и Шриканта. Пусть имеется база данных D, в которой каждая запись представляет собой клиентскую транзакцию. Каждая транзакция содержит следующие поля: идентификатор клиента, дата/время транзакции и набор купленных товаров. Добавим ограничение, что ни один клиент не имеет двух или более транзакций, совершенных в один и тот же момент времени.

Постановка задачи SPM

Пусть клиент совершил несколько упорядоченных во времени транзакций тогда, клиентская последовательность – это последовательность предметных наборов, соответствующих транзакциям, совершенным данным клиентом. Последовательность S называется поддерживаемой клиентом, если она содержится в клиентской последовательности данного клиента. Тогда поддержка последовательности определяется как число клиентов, поддерживающих данную последовательность.

Для базы данных клиентских транзакций задача поиска шаблонов заключается в обнаружении последовательностей, имеющих поддержку выше заданного порогового значения. Каждая такая последовательность является шаблоном последовательных событий.

Основные понятия SPM

Предметный набор — это непустой набор предметов (товаров), появившихся в одной транзакции.

Последовательность — это упорядоченный список предметных наборов.

Длиной последовательности будем называть количество предметов в этой последовательности, а последовательность длины k — k-последовательностью.

Далее будем называть последовательности, удовлетворяющие ограничению минимальной поддержки, частыми последовательностями.

Основные понятия SPM

Последовательность $S1$ содержится в другой последовательности $S2$, если все предметные наборы $S1$ содержатся в предметных наборах $S2$, при этом порядок надмножеств из $S2$ соответствует порядку предметных наборов $S1$.

Последовательность S называется максимальной, если она не содержится в какой-либо другой последовательности. Иногда требуется найти не все шаблоны последовательных событий, а только те из них, которые являются максимальными.

Алгоритмы SPM на основе Apriori

Первыми алгоритмами, разработанными для решения задачи *SPM*, являются *AprioriAll*, *AprioriSome* и *DynamicSome*, представленные в работе Агравала и Шриканта. Все эти алгоритмы используют подход генерации и отбора кандидатов часто встречающихся последовательностей, а также свойство **антимонотонности**. Работа данных алгоритмов состоит из нескольких фаз:

Фаза сортировки заключается в перегруппировке записей в таблице транзакций. Сначала записи сортируются по уникальному ключу покупателя, а затем по времени внутри каждой группы.

Фаза отбора кандидатов — в исходном наборе данных производится поиск всех частых предметных наборов. В частности, на этом этапе происходит поиск всех одноэлементных шаблонов.

Алгоритмы SPM на основе Apriori

Фаза трансформации. Производится для ускорения процесса проверки присутствия последовательностей в наборе транзакций покупателей. Трансформация заключается в замене каждой транзакции списком частых предметных наборов, которые в ней содержатся. При этом если в транзакции отсутствуют частые предметы, то данная транзакция не учитывается. Аналогичным образом не учитываются предметы, не являющиеся частыми, а также последовательности, транзакции которых не содержат частых предметных наборов.

Фаза генерации последовательностей — из полученных на предыдущих шагах последовательностей строятся более длинные шаблоны последовательностей.

Фаза максимизации — среди имеющихся последовательностей происходит поиск тех, что не входят в более длинные последовательности. Данный шаг является опциональным.

Алгоритм GSP (Generalized Sequential Patterns) SPM

Работа алгоритма GSP заключается в нескольких проходах по исходному набору данных. На первом проходе алгоритм вычисляет поддержку для каждого предмета и выделяет из них частые. Каждый подобный предмет представляет собой одноэлементную последовательность.

В начале каждого последующего прохода имеется некоторое число часто встречающихся последовательностей, выявленных на предыдущем шаге алгоритма. Из них будут формироваться более длинные последовательности-кандидаты. Каждый кандидат представляет собой последовательность, длина которой на один больше чем у последовательностей, из которых кандидат был сформирован. Таким образом, число элементов всех кандидатов одинаково.

Алгоритм GSP (Generalized Sequential Patterns) SPM

После формирования кандидатов происходит вычисление их поддержки. В конце шага определяется, какие кандидаты являются частыми. Найденные частые последовательности послужат исходными данными для следующего шага алгоритма. Работа алгоритма завершается тогда, когда не найдено ни одной новой частой последовательности в конце очередного шага, или когда невозможно сформировать новых кандидатов.

Таким образом, в работе алгоритма можно выделить две основные операции:

- генерация кандидатов;
- подсчёт поддержки кандидатов.

Недостатки GSP и Apriori

Из недостатков упомянутых выше алгоритмов можно выделить:

- большое количество обращений к базе данных, количество обращений соответствует длине максимального кандидата;
- большое число генерируемых кандидатов, что сильно сказывается на производительности для больших наборов данных.