Лекция 7.1

ПОИСК АССОЦИАТИВНЫХ ПРАВИЛ

Ассоциация - одна из задач Data Mining.

Целью поиска ассоциативных правил (association rule) является нахождение закономерностей между связанными событиями в базах данных.

Очень часто покупатели приобретают не один товар, а несколько. В большинстве случаев между этими товарами существует взаимосвязь. Так, например, покупатель, приобретающий макаронные изделия, скорее всего, захочет приобрести также кетчуп. Эта информация может быть использована для размещения товара на прилавках.

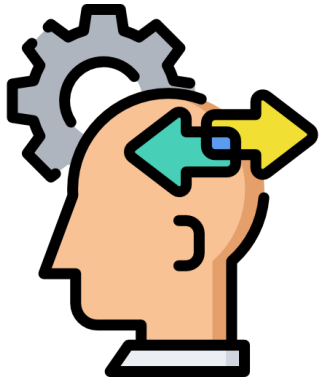
Объект 1 *Объект 2* *Найденная взаимосвязь*

Сферы применения

- **розничная торговля**: определение товаров, которые стоит продвигать совместно; выбор местоположения товара в магазине; анализ потребительской корзины; прогнозирование спроса;
- **перекрестные продажи**: если есть информация о том, что клиенты приобрели продукты А, Б и В, то какие из них вероятнее всего купят продукт Г?
- **маркетинг**: поиск рыночных сегментов, тенденций покупательского поведения;
- **сегментация клиентов**: выявление общих характеристик клиентов компании, выявление групп покупателей;

Мерчендайзинг

Поиск шаблонов поведения покупателей



Сегментация

Введение в ассоциативные правила

Рассмотрим основы на примере торговой компании.

- **Транзакция -** это множество событий, которые произошли одновременно.
- Регистрируя все бизнес-операции в течение всего времени своей деятельности, торговые компании накапливают огромные собрания транзакций.
- Каждая такая транзакция представляет собой набор товаров, купленных покупателем за один визит.
- ***Транзакционная или операционная база данных*** (Transaction database) представляет собой двумерную таблицу, которая состоит из номера транзакции (TID) и перечня покупок, приобретенных во время этой транзакции.
- **TID -** уникальный идентификатор, определяющий каждую сделку или транзакцию

Введение в ассоциативные правила

- На основе имеющейся базы данных нам нужно найти закономерности между событиями, то есть покупками.
- **Ассоциативное правило состоит из двух наборов предметов, называемых условием и следствием, записываемых в виде X→Y.**
- Допустим, имеется транзакционная база данных D. Присвоим значениям товаров переменные

TID	Приобретенные покупки
100	Хлеб, молоко, печенье
200	Молоко, сметана
300	Молоко, хлеб, сметана, печенье
400	Колбаса, сметана
500	Хлеб, молоко, печенье, сметана
600	Конфеты

→

TID	Приобретенные покупки
100	a, b, c
200	b, d
300	b, a, d, c
400	e, d
500	a, b, c, d
600	f

Введение в ассоциативные правила

- Рассмотрим набор товаров (Itemset), включающий, например, {Хлеб, молоко, печенье}.
- Выразим этот набор с помощью переменных:

abc={a,b,c}

- Этот набор товаров встречается в нашей базе данных три раза, т.е. п**оддержка этого набора товаров равна 3:**

SUP(abc)=3.

- При минимальном уровне поддержки, равной трем, набор товаров abc является часто встречающимся шаблоном.

Введение в ассоциативные правила

- **Поддержкой** называют количество или процент транзакций, содержащих определенный набор данных.
- Для данного набора товаров поддержка, выраженная в процентном отношении, равна 50%.

$$SUP(abc)=(3/6)*100\%=50\%$$

- Поддержку иногда также называют **обеспечением набора.**
- Таким образом, набор представляет интерес, если его поддержка выше определенного пользователем **минимального значения (min support).**
- Эти наборы называют часто **встречающимися (frequent).**

Поддержка ассоциативного правила

Рассмотрим правило «из покупки молока следует покупка печенья» для базы данных, которая была приведена ранее.

- **Поддержка ассоциативного правила – это число транзакций, которые содержат как условие, так и следствие.**

Например, для ассоциации A→B можно записать

support(A→B)=S(A→B)=P(A∩B)=

= (количество транзакций, содержащих A и B)/(общее число транзакций)

- Правило имеет поддержку s, если s% транзакций из всего набора содержат одновременно наборы элементов A и B или, другими словами, содержат оба товара.
- *Молоко - это товар A, печенье - это товар B. Поддержка правила "из покупки молока следует покупка печенья" равна 3, или 50%.*

Достоверность правила

Достоверность правила показывает, какова вероятность того, что из события A следует событие B.

confidence$(A \rightarrow B) = C(A \rightarrow B) = P(A|B) = P(A \cap B)/P(A) =$
(количество транзакций, содержащих A и B) /
(количество транзакций, содержащих только A)

Достоверность правила

- Правило "Из A следует B" справедливо с достоверностью c, если c% транзакций из всего множества, содержащих набор элементов A, также содержат набор элементов B.
- Число транзакций, содержащих молоко, равно четырем, число транзакций, содержащих печенье, равно трем, достоверность правила равна (3/4)*100%, т.е. 75%.
- Достоверность правила "из покупки молока следует покупка печенья" равна 75%, т.е. 75% транзакций, содержащих товар A, также содержат товар B.

Значимость

- Методики поиска ассоциативных правил обнаруживают все ассоциации, которые удовлетворяют ограничениям на поддержку и достоверность, наложенным пользователем.
- Это приводит к необходимости **рассматривать десятки и сотни тысяч ассоциаций**, что **делает невозможным обработку** такого количества данных вручную.
- **Число правил** желательно **уменьшить** таким образом, чтобы **проанализировать** только **наиболее значимые** из них.
- Значимость часто вычисляется как разность между поддержкой правила и в целом и произведением поддержки только условия и поддержки только следствия.

Субъективные меры значимости

- **Лифт** – это отношение частоты появления условия в транзакциях, которые также содержат и следствие, к частоте появления следствия в целом.

$$\text{lift}(A \rightarrow B) = L(A \rightarrow B) = C(A \rightarrow B)/S(B).$$

- **Левередж** – это разность между наблюдаемой частотой, с которой условие и следствие появляются совместно (т.е. с поддержкой ассоциации), и произведением частот появления (поддержек) условия и следствия по отдельности.

$$T(A \rightarrow B) = S(A \rightarrow B) - S(A)S(B).$$

- **Улучшение** показывает, полезнее ли правило случайного угадывания. Если $I(A \rightarrow B) > 1$, это значит, что вероятнее предсказать наличие набора B с помощью правила, чем угадать случайно.

$$I(A \rightarrow B) = S(A \rightarrow B)/(S(A)S(B)).$$

Методы и алгоритмы поиска ассоциативных правил

- **Алгоритм AIS.** В алгоритме AIS кандидаты множества наборов генерируются и подсчитываются "на лету", во время сканирования базы данных.
- **Алгоритм SETM.** Создание этого алгоритма было мотивировано желанием использовать язык SQL для вычисления часто встречающихся наборов товаров. Как и алгоритм AIS, SETM также формирует кандидатов "на лету", основываясь на преобразованиях базы данных.

Неудобство алгоритмов AIS и SETM - излишнее генерирование и подсчета слишком многих кандидатов, которые в результате не оказываются часто встречающимися. Для улучшения их работы был предложен алгоритм **Apriori**.

Алгоритм Apriori

В основе алгоритма **Apriori** лежит понятие частого набора. Под частотой понимается простое количество транзакций в которых содержится данный предметный набор.

Частый предметный набор – предметный набор с поддержкой больше заданного порога либо равной ему. Этот порог называется минимальной поддержкой.

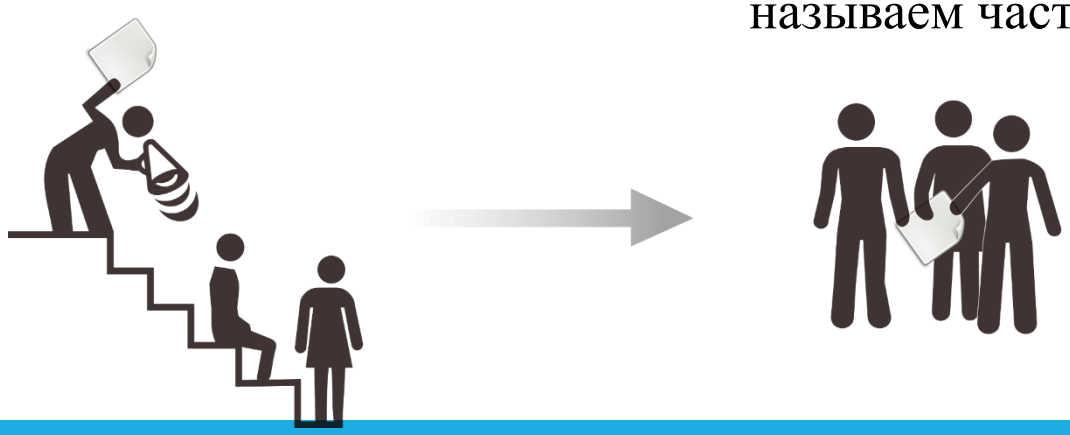
Работа данного алгоритма состоит из нескольких этапов, каждый из этапов состоит из следующих шагов:

1. формирование кандидатов;
2. подсчет кандидатов.

Алгоритм Apriori

Формирование кандидатов (candidate generation) - этап, на котором алгоритм, сканируя базу данных, создает множество i-элементных кандидатов (i - номер этапа).
На этом этапе поддержка кандидатов не рассчитывается.

Подсчет кандидатов (candidate counting) - этап, на котором вычисляется поддержка каждого i-элементного кандидата. Здесь же осуществляется отсечение кандидатов, поддержка которых меньше минимума, установленного пользователем (min_sup).
Оставшиеся i- элементные наборы называем часто встречающимися.



Алгоритм Apriori

- Чтобы сократить пространство поиска ассоциативных правил, алгоритм Apriori использует свойство **антимонотонности**.

- Свойство утверждает, что если предметный набор Z не является частым, то добавление некоторого нового предмета A к набору Z не делает его более частым. Данное полезное свойство позволяет значительно уменьшить пространство поиска ассоциативных правил.

- На первом этапе алгоритма Apriori формируются частые однопредметные наборы – множество F_1.

Алгоритм Apriori

- Для поиска F_k, то есть k-предметных наборов, алгоритм Apriori сначала создает множество F_k кандидатов в k-предметные наборы путем связывания множества F_{k-1} с самим собой.
- Затем F_k сокращается с использованием свойства антимонотонности.
- Предметные наборы множества F_k, которые остались после сокращения, формируют F_k.

Алгоритм Apriori

После того, как все частые предметные наборы найдены, можно переходить к *генерации на их основе ассоциативных правил*.

Для этого к каждому частому предметному набору s можно применить процедуру, состоящую из 2 шагов.

1). Генерируются все возможные поднаборы s.

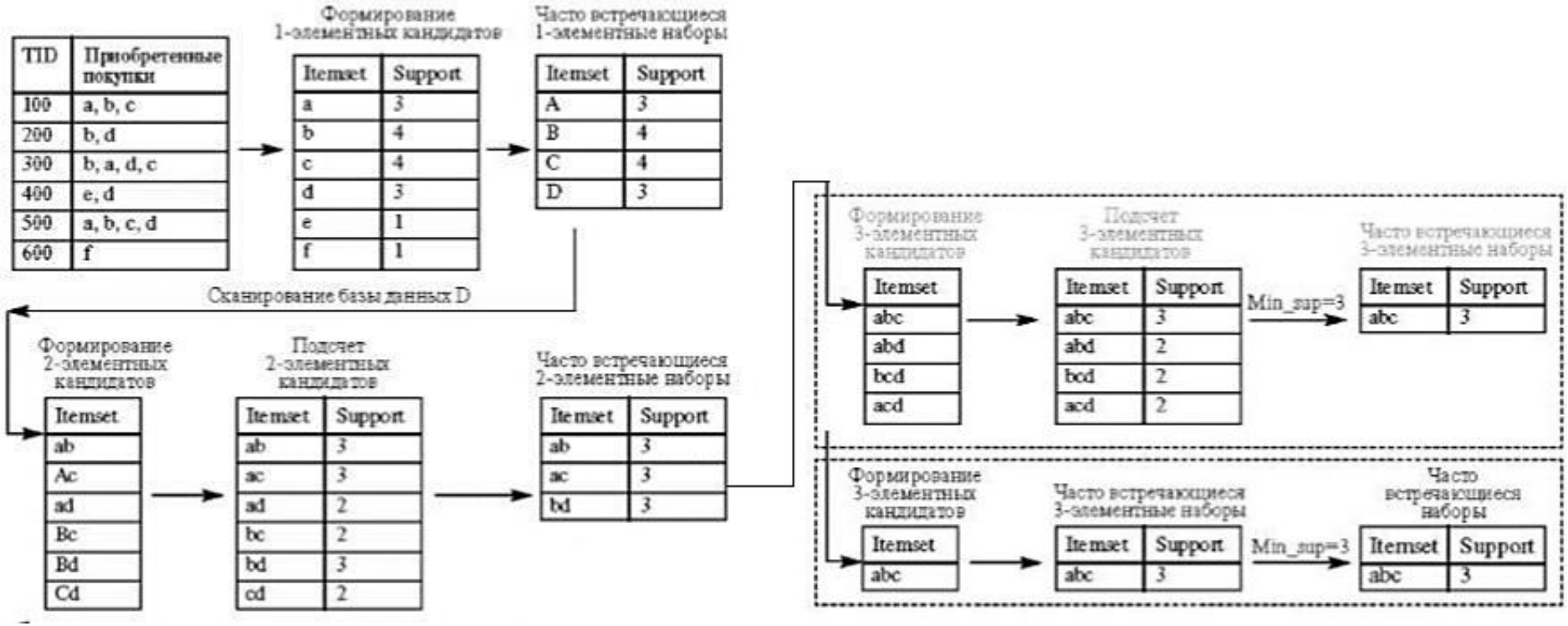
2). Если поднабор ss является непустым поднабором s, то рассматривается ассоциация R:ss→(s-ss), где s-ss представляет собой набор s без поднабора ss.

R считается ассоциативным правилом, если удовлетворяет условию заданного минимума поддержки и достоверности.

Данная процедура повторяется для каждого подмножества ss из s.

Алгоритм Apriori

Рассмотрим работу алгоритма Apriori на примере базы данных D. Минимальный уровень поддержки равен 3.



Разновидности Apriori

- **AprioriTid.** Интересная особенность этого алгоритма - то, что база данных D не используется для подсчета поддержки кандидатов набора товаров после первого прохода.
- С этой целью используется кодирование кандидатов, выполненное на предыдущих проходах.
- В последующих проходах размер закодированных наборов может быть намного меньше, чем база данных, и таким образом экономятся значительные ресурсы.

Разновидности Apriori

- **AprioriHybrid.** Анализ времени работы алгоритмов Apriori и AprioriTid показывает, что в более ранних проходах Apriori добивается большего успеха, чем AprioriTid.
- Однако AprioriTid работает лучше Apriori в более поздних проходах. Кроме того, они используют одну и ту же процедуру формирования наборов-кандидатов.
- Основанный на этом наблюдении, алгоритм AprioriHybrid предложен, чтобы объединить лучшие свойства алгоритмов Apriori и AprioriTid.
- AprioriHybrid использует алгоритм Apriori в начальных проходах и переходит к алгоритму AprioriTid, когда ожидается, что закодированный набор первоначального множества в конце прохода будет соответствовать возможностям памяти.
- Однако, переключение от Apriori до AprioriTid требует вовлечения дополнительных ресурсов.

Разновидности Apriori. AprioriHybrid.

- **Алгоритм DHP**, также называемый алгоритмом хеширования (J. Park, M. Chen and P. Yu, 1995 год). В основе его работы - вероятностный подсчет наборов-кандидатов, осуществляемый для сокращения числа подсчитываемых кандидатов на каждом этапе выполнения алгоритма Apriori.
- К другим усовершенствованным алгоритмам относятся: PARTITION, DIC, алгоритм "выборочного анализа".
- **PARTITION алгоритм** (A. Savasere, E. Omiecinski and S. Navathe, 1995 год). Этот алгоритм разбиения (разделения) заключается в сканировании транзакционной базы данных путем разделения ее на непересекающиеся разделы, каждый из которых может уместиться в оперативной памяти.
- **Алгоритм DIC**, Dynamic Itemset Counting (S. Brin R. Motwani, J. Ullman and S. Tsur, 1997 год). Алгоритм разбивает базу данных на несколько блоков, каждый из которых отмечается так называемыми "начальными точками" (start point), и затем циклически сканирует базу данных.

Ссылки на используемые источники:

Apriori — масштабируемый алгоритм поиска ассоциативных правил

Agrawal R, Imielinski T, Swami AN. «Mining Association Rules between Sets of Items in Large Databases.» SIGMOD. June 1993, 22(2):207-16

Market Basket Analysis

Выводы

- **Задачей поиска ассоциативных правил** является определение часто встречающихся наборов объектов в большом множестве наборов.

- **Секвенциальный анализ заключается** в поиске частых последовательностей. Основным отличием задачи секвенциального анализа от поиска ассоциативных правил является установление отношения порядка между объектами.

- **Наличие иерархии в объектах** и ее использование в задаче поиска ассоциативных правил позволяет выполнять более гибкий анализ и получать дополнительные знания.

Результаты решения задачи представляются в виде ассоциативных правил, условная и заключительная часть которых содержит наборы объектов.

Основными характеристиками ассоциативных правил являются поддержка, достоверность и улучшение.

Поддержка (support) показывает, какой процент транзакций поддерживает данное правило.

Достоверность (confidence) показывает, какова вероятность того, что из наличия в транзакции набора условной части правила следует наличие в ней набора заключительной части.

Улучшение (improvement) показывает, полезнее ли правило случайного угадывания.

Задача поиска ассоциативных правил решается в два этапа. На первом выполняется поиск всех частых наборов объектов. На втором из найденных частых наборов объектов генерируются ассоциативные правила.

Алгоритм Apriori использует одно из свойств поддержки, гласящее: поддержка любого набора объектов не может превышать минимальной поддержки любого из его подмножеств.