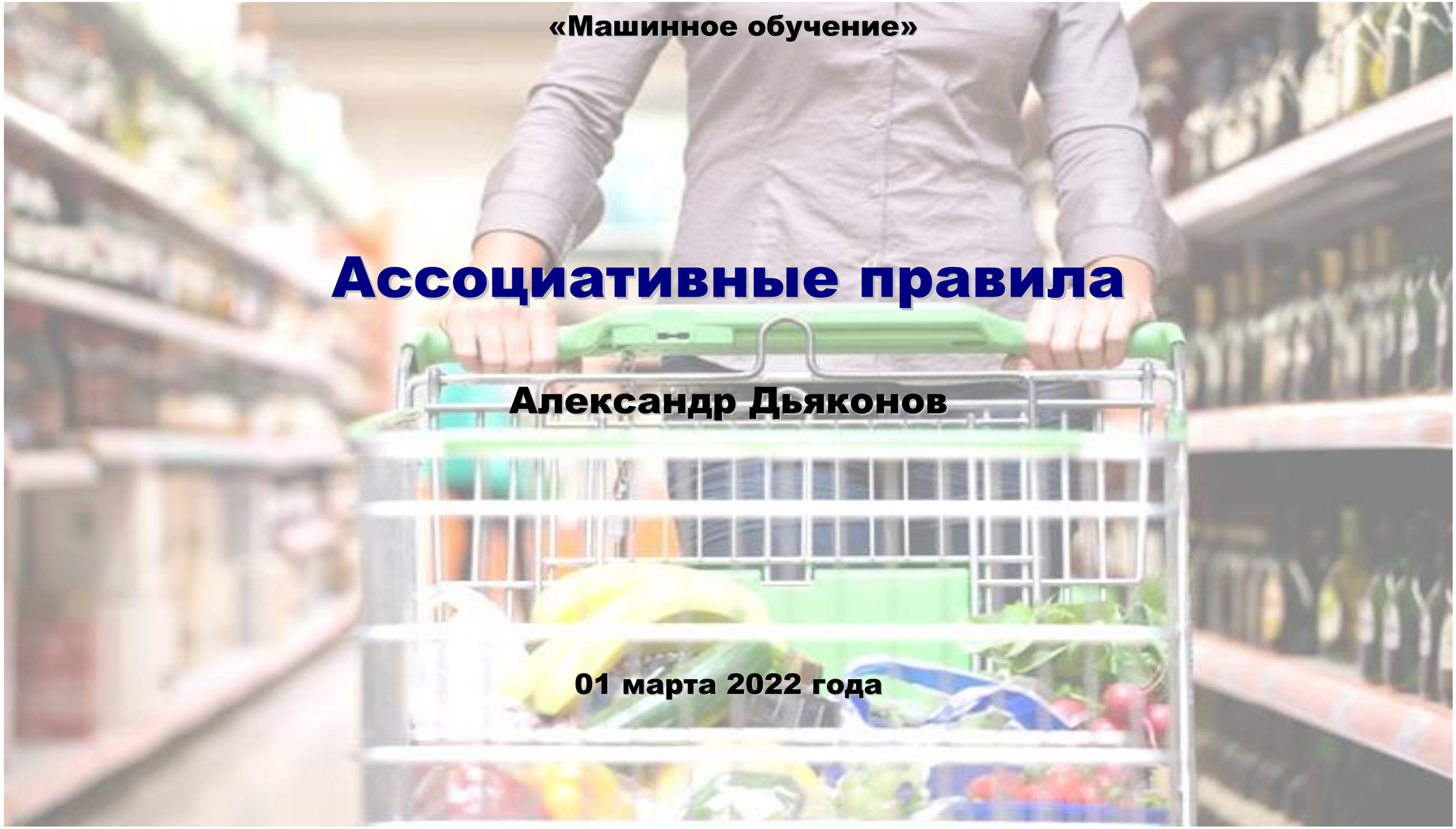


«Машинное обучение»

Ассоциативные правила

Александр Дьяконов

01 марта 2022 года

План

Ассоциативные правила (Association Rules)

Поддержка (support)

Достоверность (confidence)

Улучшение (Lift)

Apriori

«частое множество» (frequent itemset)

FP-Growth

Логические закономерности

Анализ покупательских корзин (Market Basket Analysis)

Интересный факт: у Сергея Брина есть несколько статей по ассоциативным правилам

Ассоциативные правила (Association Rules)

Обнаружение взаимосвязей переменных

Чаще – при анализе покупательских корзин (Market Basket Analysis)

Чаще – в терминах «если … , то…»

Ключевые термины:

Поддержка (Support)

Уверенность (Confidence)

Улучшение (Lift)

Apriori Algorithm

Ассоциативные правила

Товары (items): $I = \{i_j\}_{j=1}^{n}$ **{хлеб, масло, молоко, …}**

Объект (корзина, transaction): $x = \{i_j\}_j \subseteq I$ **{хлеб, кефир}**

Правило: $A \to B,\ A, B \subseteq I,\ A \cap B = \varnothing$ **{креветки, чипсы}** $\to$ **{пиво}**

«market basket transactions»:

корзина	товар
1	хлеб, соль,
2	перец, сахар, соль,
3	водка,
4	хлеб, соль, перец, сахар,

Ассоциативные правила

Поддержка (support) – частота вхождения данного множества в обучение

$$\text{support}(A) = \frac{|\{A \subseteq x \,|\, x \in X_{\text{train}}\}|}{|X_{\text{train}}|}$$

иногда $\text{support}(A \to B) = \text{support}(A \cup B)$

насколько частое правило

Достоверность / уверенность / значимость (confidence) ~ вероятность правильности правила

$$\text{confidence}(A \to B) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$$

насколько надёжное правило

Ассоциативные правила

Lift, иногда Улучшение (improvement) ~ полезнее ли правило случ. угадывания

$$\text{lift}(A \to B) = \frac{\text{support}(A \cup B)}{\text{support}(A) \cdot \text{support}(B)} =$$

$$= \frac{|\{A \cup B \subseteq x \,|\, x \in X_{\text{train}}\}| \cdot |X_{\text{train}}|}{|\{A \subseteq x \,|\, x \in X_{\text{train}}\}| \cdot |\{B \subseteq x \,|\, x \in X_{\text{train}}\}|}$$

есть и много других характеристик правил!

Ассоциативные правила: примеры

корзина	товар
1	хлеб, соль,
2	перец, сахар, соль,
3	водка,
4	хлеб, соль, перец, сахар,

$$\text{confidence}(\{\text{соль}\} \to \{\text{перец}\}) = \frac{2/4}{3/4} = \frac{2}{3}$$

$$\text{confidence}(\{\text{соль}\} \to \{\text{хлеб}\}) = \frac{2/4}{3/4} = \frac{2}{3}$$

$$\text{confidence}(\{\text{соль}\} \to \{\text{перец, хлеб}\}) = \frac{1/4}{3/4} = \frac{1}{3}$$

могло ли получиться = 0?

Ассоциативные правила: примеры

корзина	товар
1	хлеб, соль,
2	перец, сахар, соль,
3	водка,
4	хлеб, соль, перец, сахар,

$$\text{support}(\{\text{хлеб}\}) = \frac{2}{4} \qquad \text{support}(\{\text{соль}\}) = \frac{3}{4}$$

$$\text{confidence}(\{\text{хлеб}\} \to \{\text{соль}\}) = \frac{2/4}{2/4} = 1$$

$$\text{confidence}(\{\text{соль}\} \to \{\text{хлеб}\}) = \frac{2/4}{3/4} = \frac{2}{3}$$

$$\text{confidence}(\{\text{хлеб}, \text{соль}\} \to \{\text{перец}\}) = \frac{1/4}{2/4} = \frac{1}{2}$$

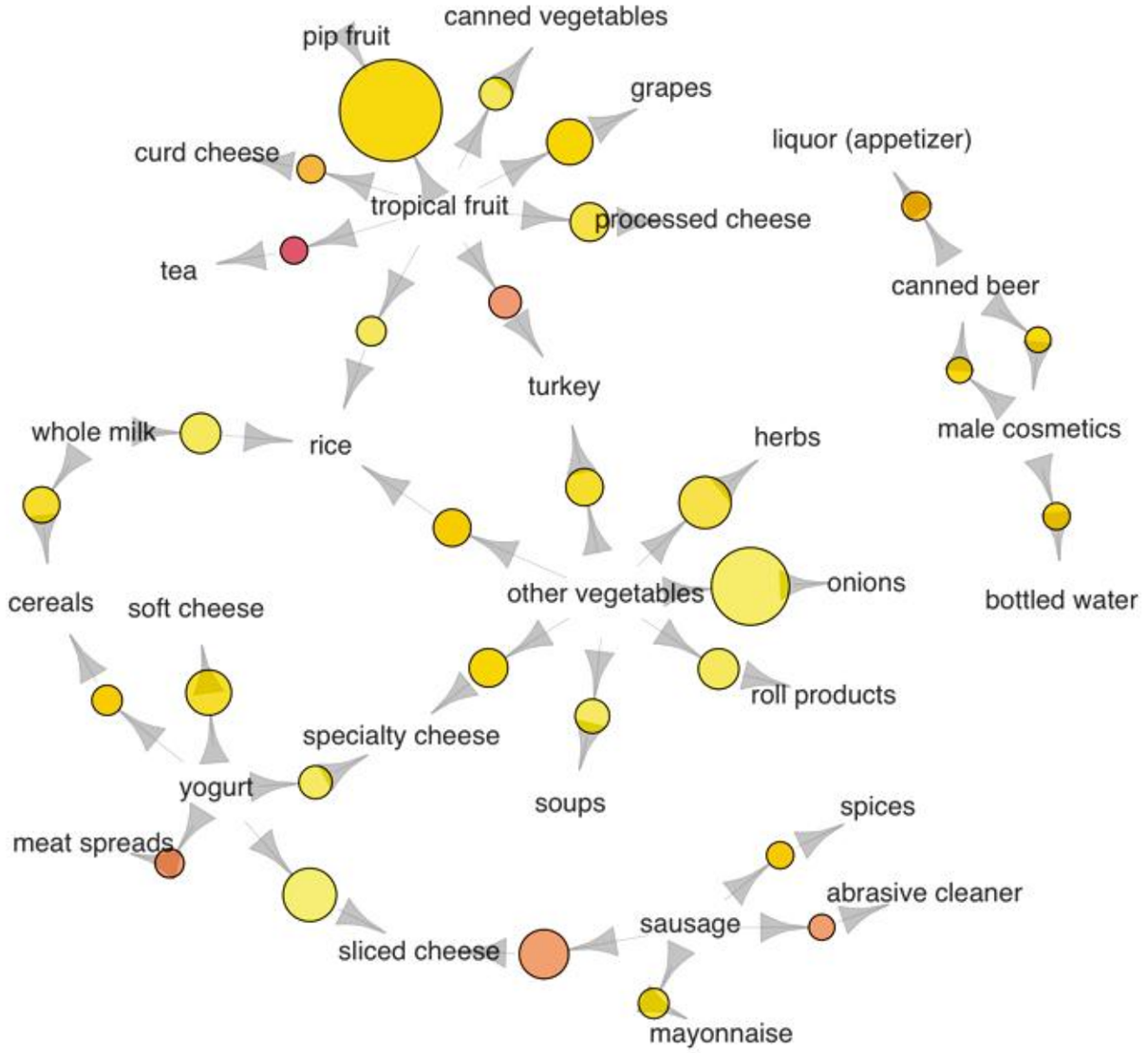
Ассоциативные правила: примеры

корзина	товар
1	хлеб, соль,
2	перец, сахар, соль,
3	водка,
4	хлеб, соль, перец, сахар,

$$\text{lift}(\{\text{хлеб}\} \to \{\text{соль}\}) = \frac{2 \cdot 4}{2 \cdot 3} = 1\frac{1}{3}$$

$$\text{lift}(\{\text{соль}\} \to \{\text{хлеб}\}) = \frac{2 \cdot 4}{3 \cdot 2} = 1\frac{1}{3}$$

$$\text{lift}(\{\text{хлеб}, \text{соль}\} \to \{\text{перец}\}) = \frac{1 \cdot 4}{2 \cdot 2} = 1$$



размер – support, цвет – lift (красный – больше)

https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html

Задача анализа АП (Association Rule Discovery)

Association Rule Mining Task

найти все $A \to B$:

$$\text{support}(A \to B) \geq \alpha \text{ (иногда } \text{support}(A) \geq \alpha \text{)}$$

$$\text{confidence}(A \to B) \geq \beta$$

алгоритм Apriori

1) ищем множества с достаточной поддержкой
«Frequent Itemset Generation»

2) формируем правила (проверяем достоверность)
«Rule Generation»

если нашли большую поддержку, например у {A, B, C},
то возможные правила

$\{A, B\} \to \{C\}$
$\{A, C\} \to \{B\}$
$\{B, C\} \to \{A\}$
$\{C\} \to \{A, B\}$
$\{B\} \to \{A, C\}$
$\{A\} \to \{B, C\}$

принципы Apriori: антимонотонность поддержки

$$\text{support}(A) \geq \alpha,\ A' \subseteq A \Rightarrow \text{support}(A') \geq \text{support}(A) \geq \alpha$$

нашли «большую поддержку» – все подмножества тоже большая поддержка

«частое множество» (frequent itemset)

«маленькая поддержка» – все надмножества тоже маленькая

«редкое множество» (infrequent itemset)

принципы Apriori: антимонотонность поддержки

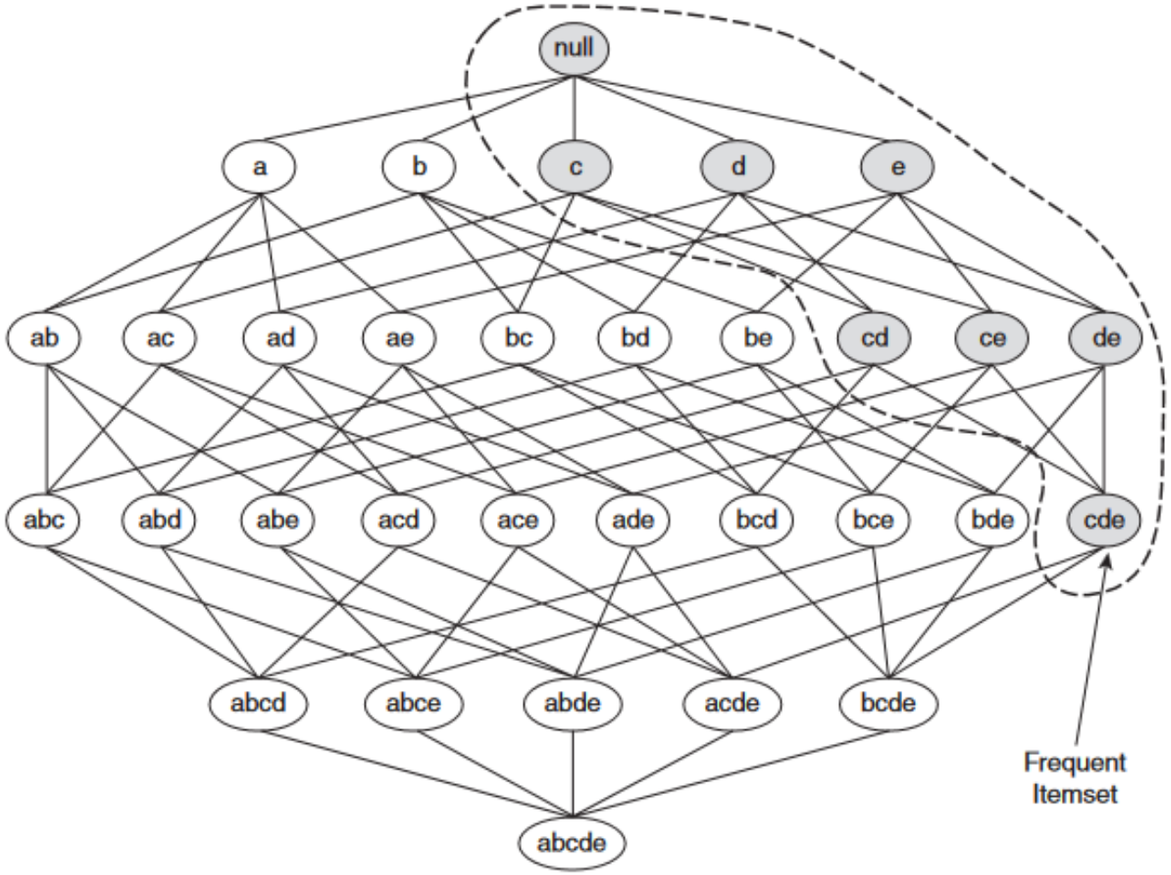


Figure 6.3. An illustration of the *Apriori* principle. If $\{c, d, e\}$ is frequent, then all subsets of this itemset are frequent.

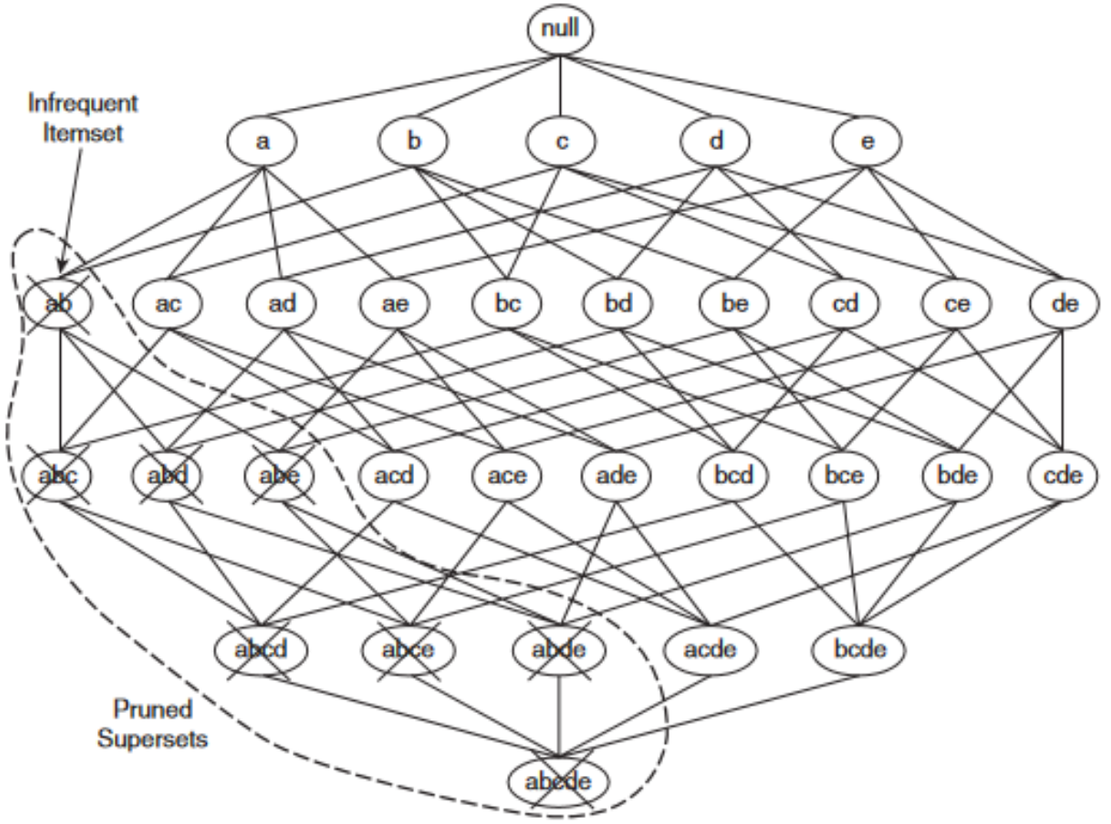


Figure 6.4. An illustration of support-based pruning. If $\{a, b\}$ is infrequent, then all supersets of $\{a, b\}$ are infrequent.

принципы Apriori: умный перебор подмножеств

корзина	товар
1	хлеб, соль,
2	перец, сахар, соль,
3	водка,
4	хлеб, соль, перец, сахар,

товар	поддержка
водка	1
перец	2
сахар	2
соль	3
хлеб	2

товар1	товар2	поддержка
хлеб	соль	2
хлеб	перец	1
хлеб	сахар	1
соль	перец	2
соль	сахар	2
перец	сахар	2

при генерации пар и $\alpha = 2$
водка не участвует

при генерации троек и $\alpha = 2$
пары с 1 не расширяются…

не совсем правильно, т.к. поддержка = доля

Apriori: множества с достаточной поддержкой

- k=1
- сгенерировать наборы длины k
- вычислить их поддержку
- оставить наборы с достаточной поддержкой $\geq \alpha$
- повтор
  - сгенерировать (k+1)-наборы-кандидаты из k-наборов достаточной поддержки
  - удалить из них те, которые имеют k-поднаборы не достаточной поддержки
  - посчитать поддержку для кандидатов
  - оставить кандидатов с достаточной поддержкой

есть и другие алгоритмы

Один из тонких моментов

как генерировать (k) → (k+1)

1. Наивный способ (Brute-Force Method):

{2,4}, {1,2}, {2,3}, {4,5}, {1,3} → {2,4,1}, {2,4,3}, {2,4,5}, …

добавлять все товары

2. Умнее – объединять k-шки (k × k)

	{2,4}	{1,2}	{2,3}	{4,5}	{1,3}
{2,4}		{1,2,4}	{2,3,4}	{2,4,5}	много
{1,2}			{1,2,3}	много	{1,2,3}
{2,3}				много	{1,2,3}
{4,5}					много
{1,3}					

3. Присоединять к k-шке частые товары (k × 1)

Один из тонких моментов

если пара (частых) наборов

$$I_1 = \{x_1, \ldots, x_{k-1}, x_k\}$$

$$I_2 = \{x_1, \ldots, x_{k-1}, x'_k\},\ x'_k > x_k$$

только если совпадают первые k-1 товаров

генерируем новый набор

$$I_{\text{new}} = \{x_1, \ldots, x_{k-1}, x_k, x'_k\}$$

Ещё тонкость

как вычислять support на каждом шаге

подробно не будем… можно использовать хэширование

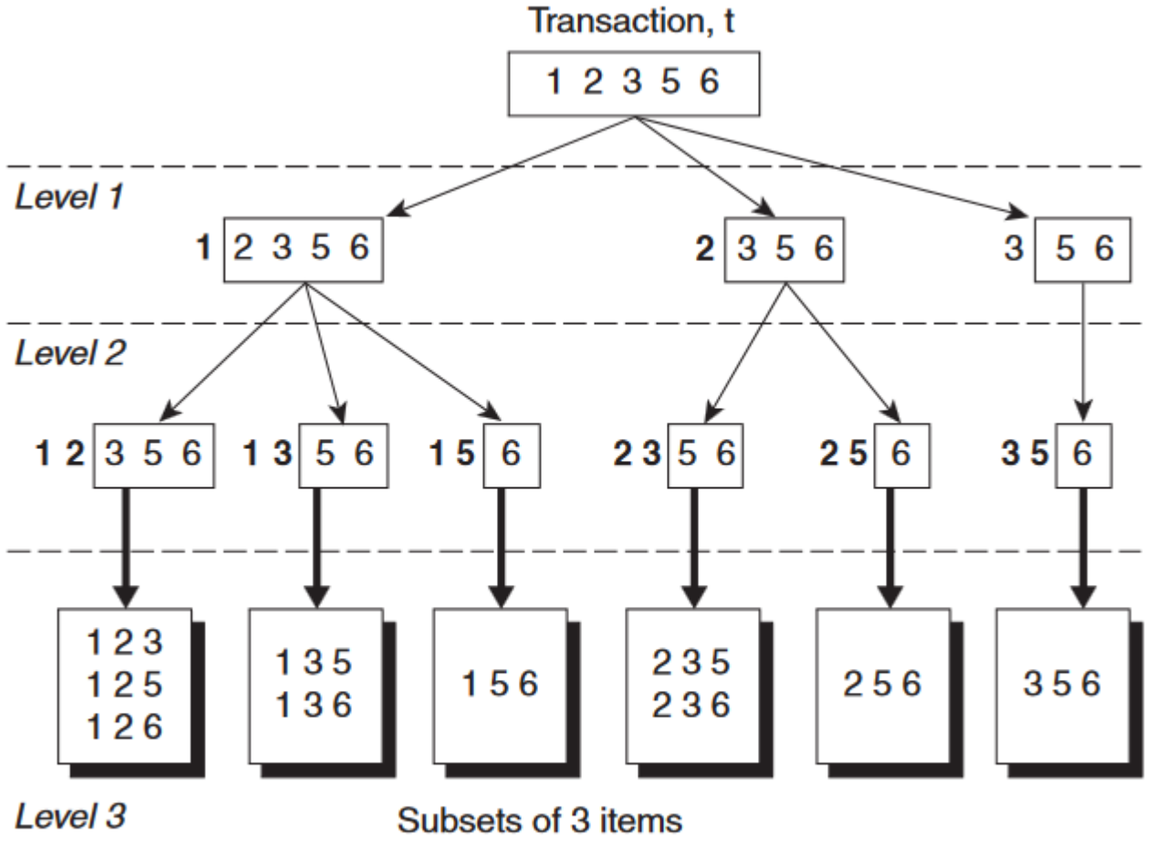


Figure 6.9. Enumerating subsets of three items from a transaction t.

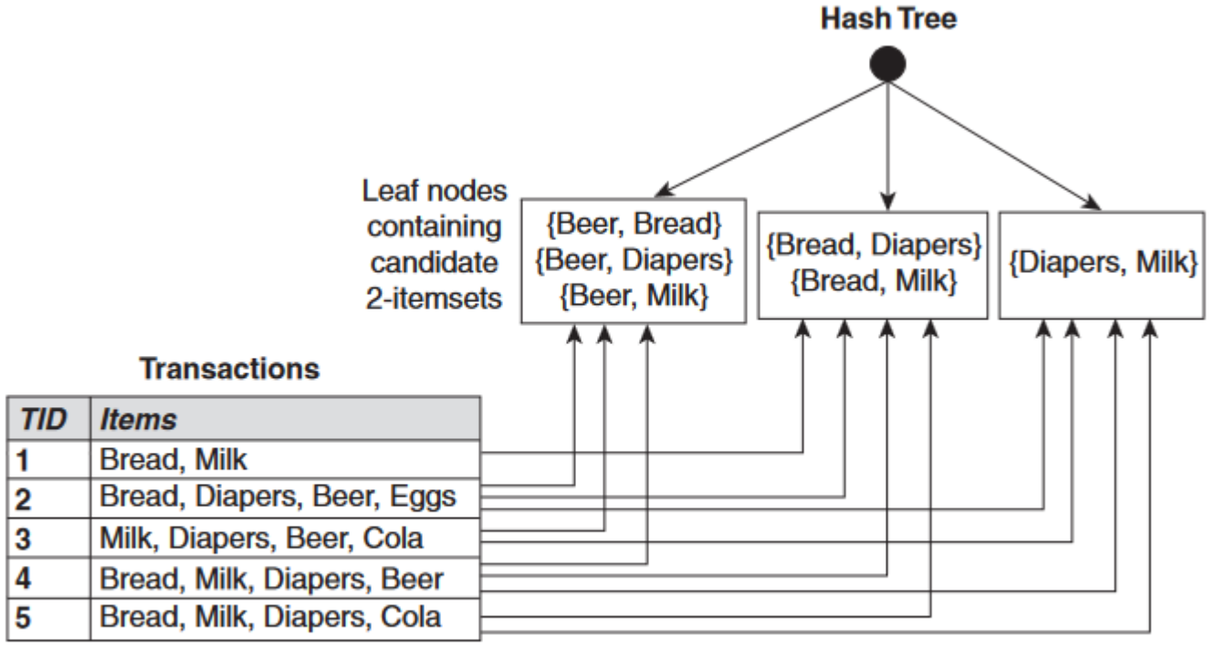


Figure 6.10. Counting the support of itemsets using hash structure.

алгоритм Apriori: генерация правил (Rule Generation)

проверяем достоверность

если нашли большую поддержку, например у {A, B, C},
то возможные правила

$\{A, B\} \to \{C\}$
$\{A, C\} \to \{B\}$
$\{B, C\} \to \{A\}$
$\{C\} \to \{A, B\}$
$\{B\} \to \{A, C\}$
$\{A\} \to \{B, C\}$

алгоритм Apriori: генерация правил (Rule Generation)

Аналогичный принцип монотонности

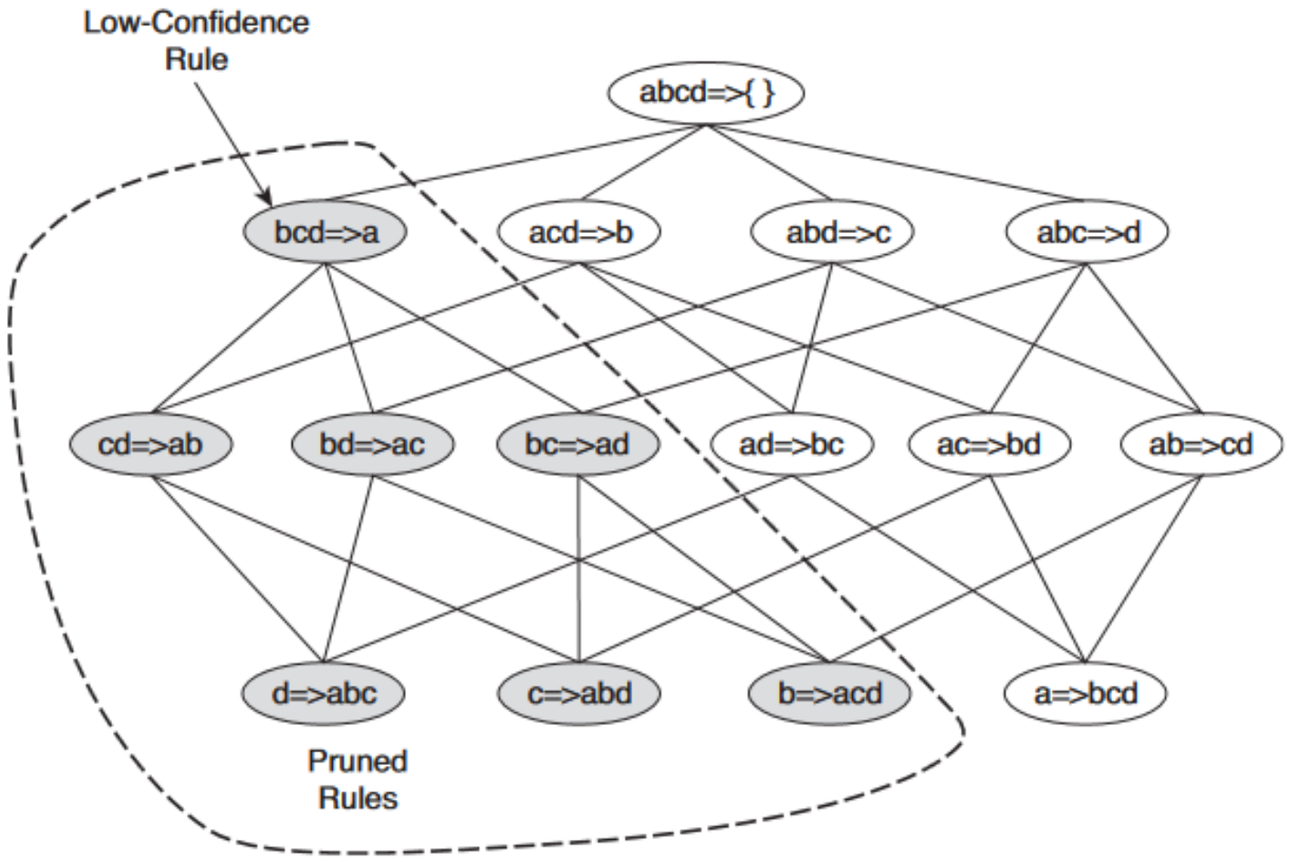


Figure 6.15. Pruning of association rules using the confidence measure.

https://www-users.cs.umn.edu/~kumar001/dmbook/ch6.pdf

Пример алгоритма

Номер транзакции	Номер товара	Наименование товара	Цена
0	1	Чипсы	12,00
0	3	Вода	4,00
0	4	Пиво	14,00
1	2	Кокосы	10,00
1	3	Вода	4,00
1	5	Орехи	15,00
2	5	Орехи	15,00
2	2	Кокосы	10,00
2	1	Чипсы	12,00
2	2	Кокосы	10,00
2	3	Вода	4,00
3	2	Кокосы	10,00
3	5	Орехи	15,00
3	2	Кокосы	10,00

Пример алгоритма, k=1

№	Набор	Supp
1	{0}	0
2	{1}	0,5
3	{2}	0,75
4	{4}	0,25
5	{3}	0,75
6	{5}	0,75

$\text{Supp}_{\min} = 0{,}5$ $L_1 = \{\{1\}, \{2\}, \{3\}, \{5\}\}$

Пример алгоритма, k=2

№	Набор	Supp
1	{1, 2}	0,25
2	{1, 3}	0,5
3	{1, 5}	0,25
4	{2, 3}	0,5
5	{2, 5}	0,75
6	{3, 5}	0,5

$L_2 = \{\{1, 3\}, \{2, 3\}, \{2, 5\}, \{3, 5\}\}$

Пример алгоритма, k=3

№	Набор	Supp
1	{2, 3, 5}	0,5

$L_3 = \{\{2, 3, 5\}\}$

4-элементные наборы создать нельзя

Ответ:

$$L = L_1 \cup L_2 \cup L_3 = \{\{1\}, \{2\}, \{3\}, \{5\}, \{1, 3\}, \{2, 3\}, \{2, 5\}, \{3, 5\}, \{2, 3, 5\}\}$$

Максимально частый набор (Maximal Frequent Itemsets)

– частый набор, любой наднабор которого не частый

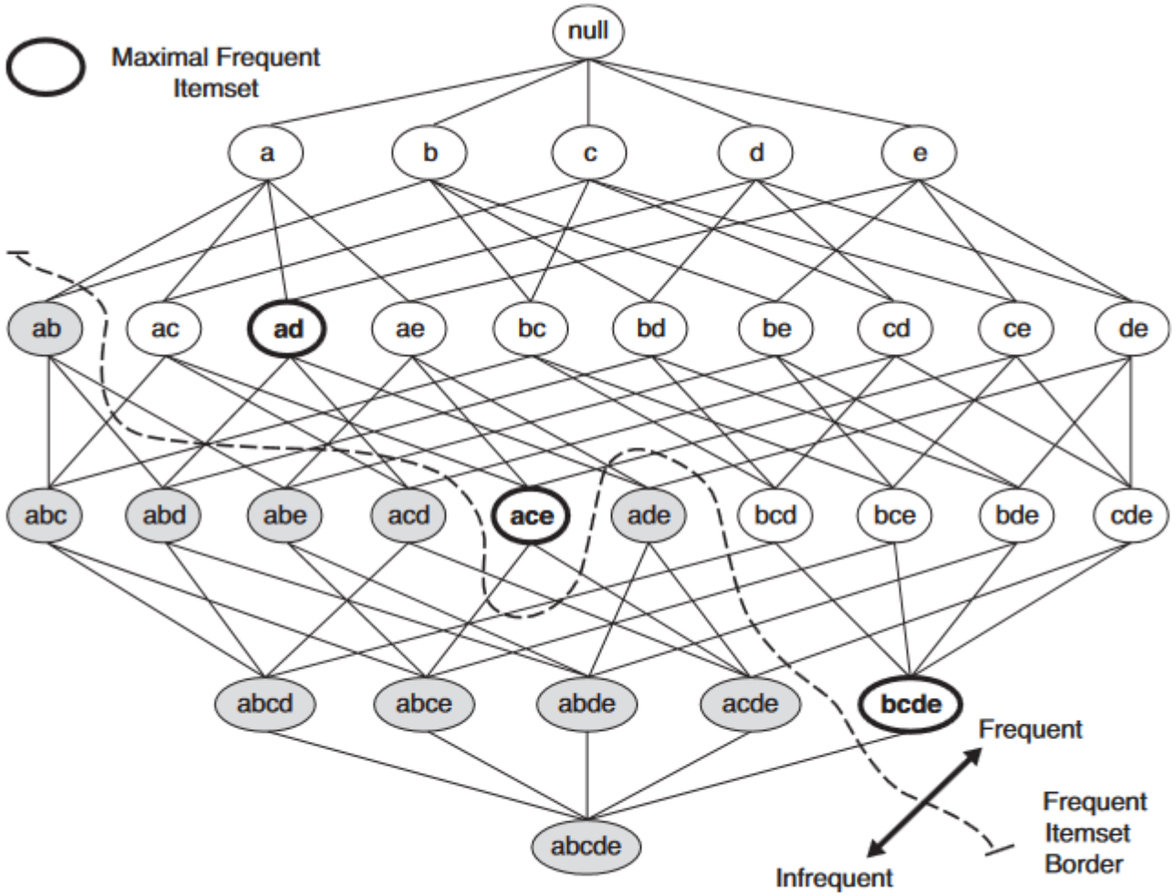


Figure 6.16. Maximal frequent itemset.

для хранения частых наборов достаточно хранить только максимально частые

FP-Growth – Frequent Pattern Tree Approach:

– алгоритм построения дерева

Другой способ генерации наборов с достаточной поддержкой

Выбираем удачную структуру хранения данных!
FP-tree (frequent pattern tree)

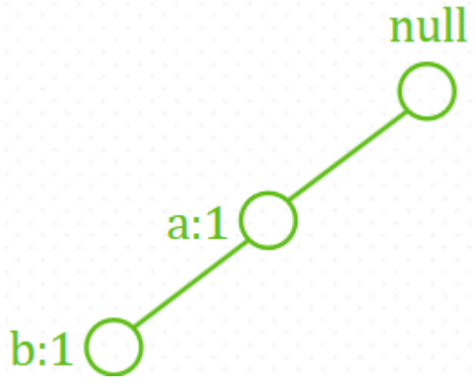
FP-tree (frequent pattern tree)

TID	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

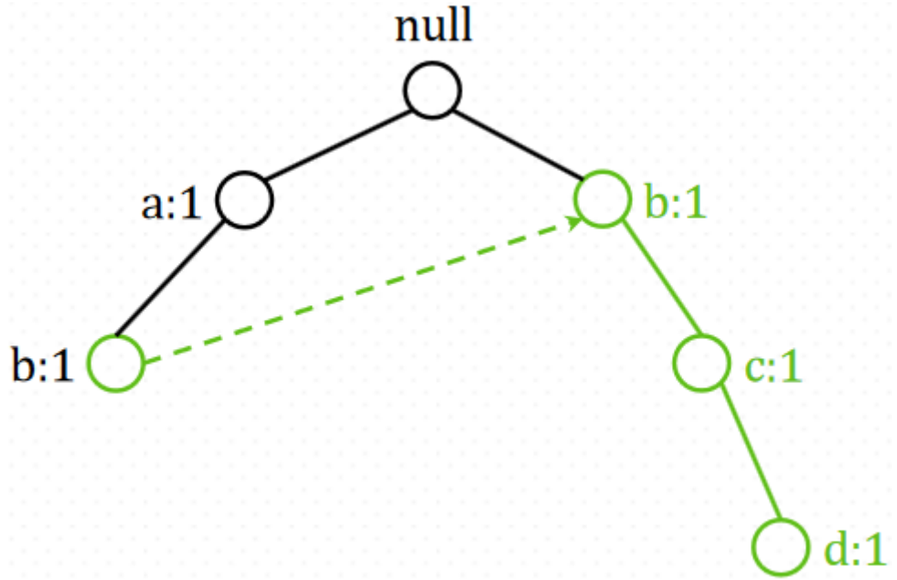
0. Сначала упорядочиваем по частоте: a, b, c, d, e

в наборах товары будут идти в таком порядке, т.е. не {c, a, b}, а {a, b, c}

1. Кодируем {a, b}

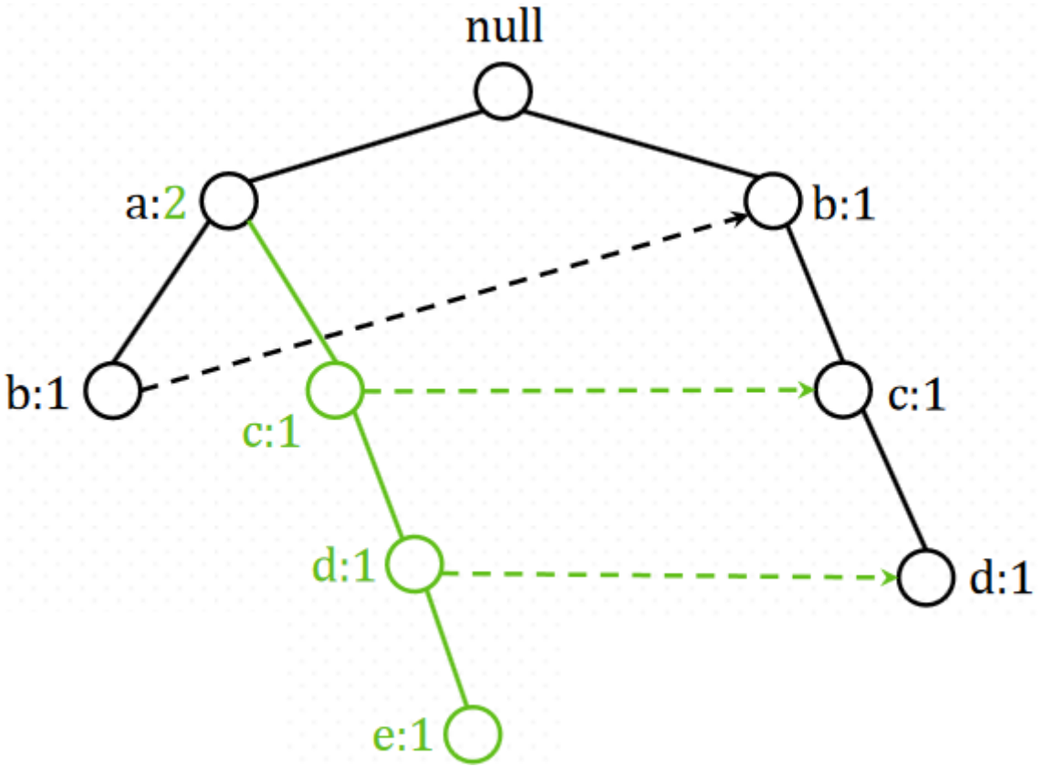


2. Кодируем {b, c, d}

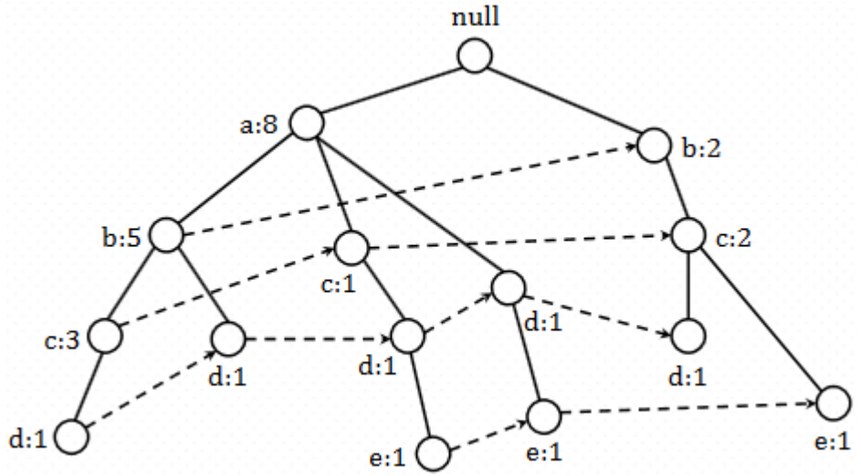


FP-tree (frequent pattern tree)

3. Кодируем {a, c, d, e}



Всё закодировали



Пунктиры – уровни дерева

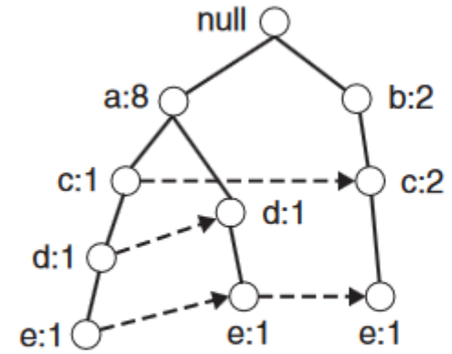
Числа – счётчики поддержки

счётчик(a) = 2

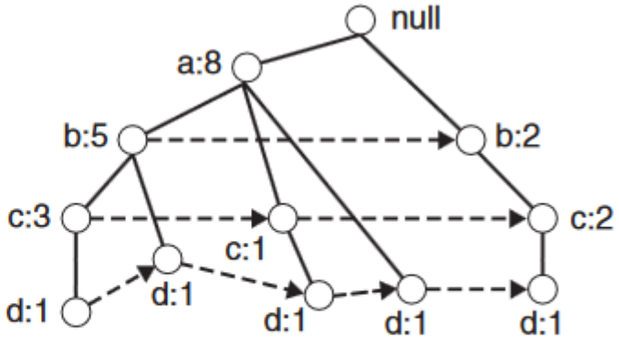
ДЗ как использовать такую структуру для поиска частых наборов?

FP-tree (frequent pattern tree): поиск частых наборов

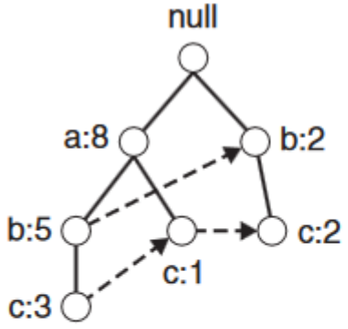
перебираем наборы «снизу-вверх» (которые кончаются на e,d,c,…), см:



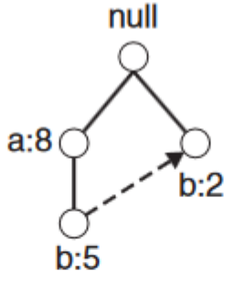
(a) Paths containing node e



(b) Paths containing node d



(c) Paths containing node c



(d) Paths containing node b

(e) Paths containing node a

Figure 6.26. Decomposing the frequent itemset generation problem into multiple subproblems, where each subproblem involves finding frequent itemsets ending in e, d, c, b, and a.

Table 6.6. The list of frequent itemsets ordered by their corresponding suffixes.

Suffix	Frequent Itemsets
e	{e}, {d,e}, {a,d,e}, {c,e},{a,e}
d	{d}, {c,d}, {b,c,d}, {a,c,d}, {b,d}, {a,b,d}, {a,d}
c	{c}, {b,c}, {a,b,c}, {a,c}
b	{b}, {a,b}
a	{a}

Ассоциативные правила: применение

оптимизация размещения товаров на полках

рекомендации

планирование промо-акций и исследований

Не только для товаров в магазине…

пример на результатах голосования

Table 6.4. Association rules extracted from the 1984 United States Congressional Voting Records.

Association Rule	Confidence
{budget resolution = no, MX-missile=no, aid to El Salvador = yes } ⟶ {Republican}	91.0%
{budget resolution = yes, MX-missile=yes, aid to El Salvador = no } ⟶ {Democrat}	97.5%
{crime = yes, right-to-sue = yes, physician fee freeze = yes} ⟶ {Republican}	93.5%
{crime = no, right-to-sue = no, physician fee freeze = no} ⟶ {Democrat}	100%

Ассоциативные правила: применение

правило не есть «естественная зависимость»
зависит от предложений, акций
(а не только от связи товаров и вкусов пользователей)

правило не означает зависимость!

Полезные правила на практике:
«если пиво.цена < 60, то чипсы.цена < 70»

Логические закономерности

Тест – множество столбцов, в которых все классы различаются

класс 1	00110100 01100100 00100101
класс 2	11111001 11011011 11101011
класс 3	00001000 00000100

Тупиковый тест – «несокращаемый» тест

Логические закономерности

Представительный набор – подописание, которое есть у какого-то объекта и которого нет в других классах

класс 1	00110100 01100100 00100101
класс 2	11111001 11011011 11101011
класс 3	00001000 00000100

Тупиковый представительный набор – «несокращаемый» представительный набор

Итоги

Поиск АП – обучение без учителя

АП – пример закономерности в данных
(м.б. полезной)

Есть разные приложения

Алгоритм APriory

Алгоритм FP-growth

Логические закономерности

Картинки взяты отсюда:
книга «Introduction to Data Mining»
https://www-users.cse.umn.edu/~kumar001/dmbook/ch6.pdf