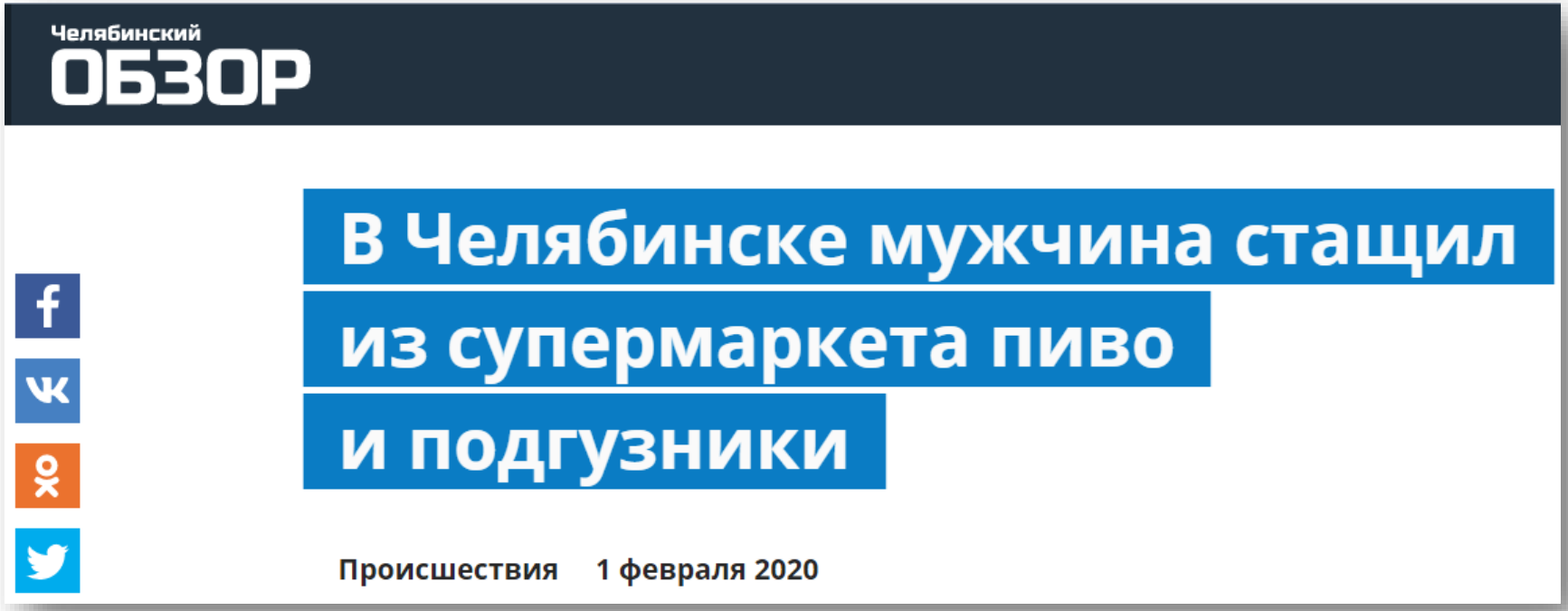
Задача поиска шаблонов в данных

Все в мире повторяется.

Ф. Бэкон


Челябинский ОБЗОР

В Челябинске мужчина стащил из супермаркета пиво и подгузники

Происшествия 1 февраля 2020

https://obzor174.ru/v-chelyabinske-muzhchina-stashchil-iz-supermarketa-pivo-i-podguzniki

Содержание

- Частые наборы и шаблоны
- Алгоритм Apriori поиска частых наборов
- Меры полезности шаблонов

Пример: анализ рыночной корзины

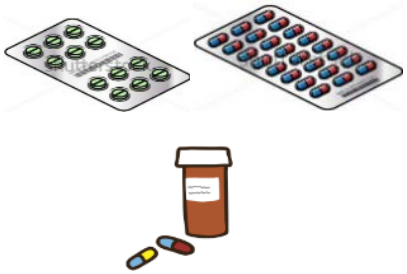
- Какие наборы товаров в супермаркете часто покупают совместно?

Пример: поиск побочных эффектов лекарств

- Какие симптомы у пациентов часто встречаются совместно с принимаемыми лекарствами?

жар

тошнота
рвота



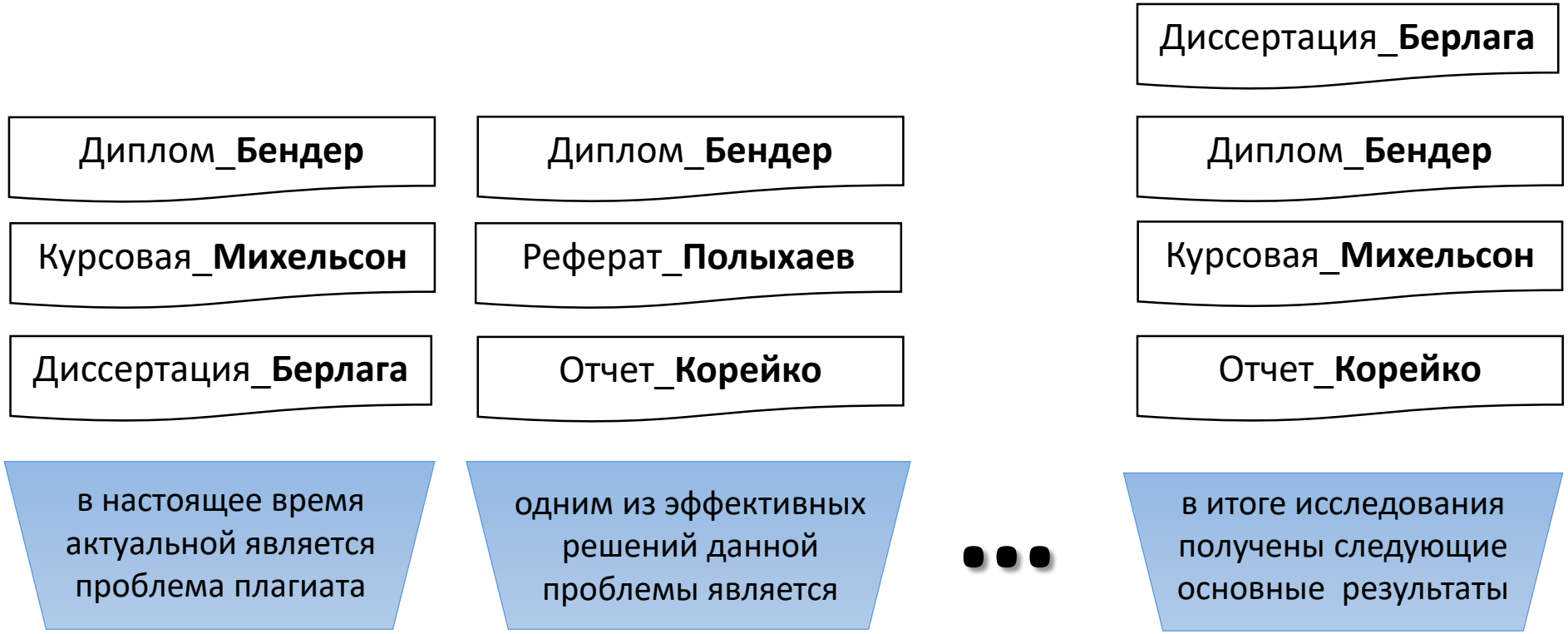
тошнота
жар

тошнота
жар
сыпь

...

Пример: выявление заимствований

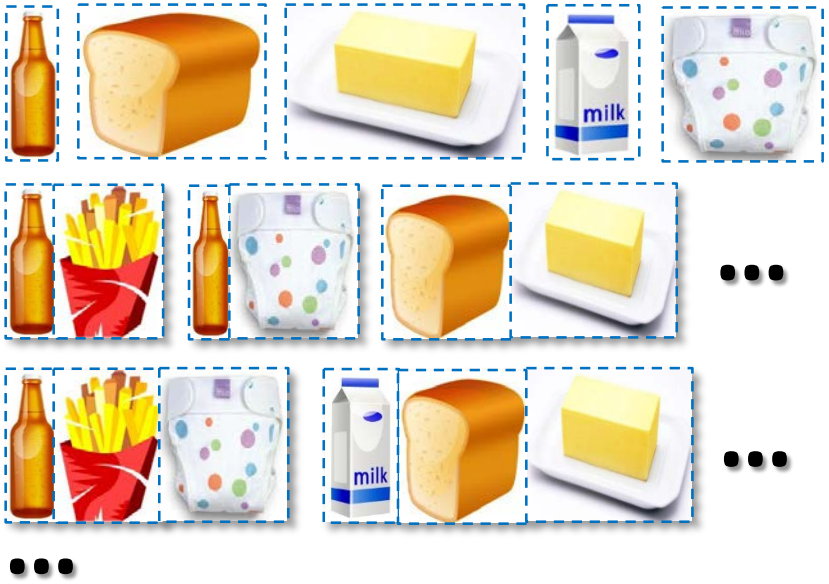
- Какие группы авторов часто используют одинаковые конструкции?



Частый набор *vs.* шаблон

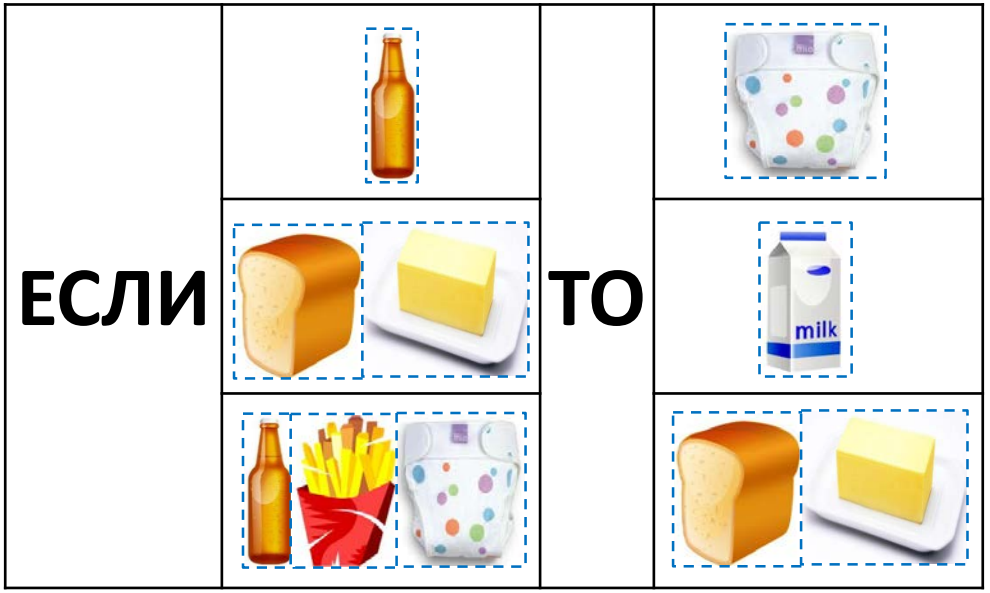
Частые k-наборы

$1 \leq k \leq k_{max}$



Шаблоны (ассоциативные правила)

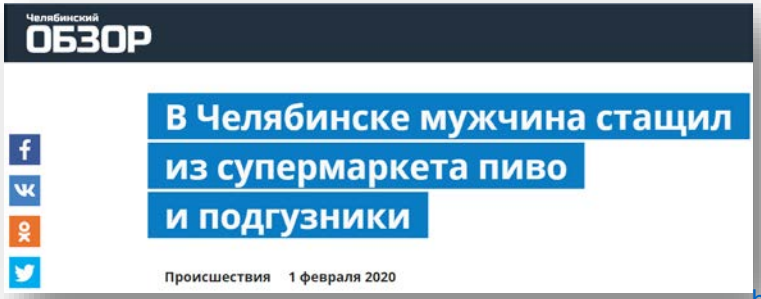
антецедент → *консеквент*



А при чем здесь  и  ?

- В 1992 г. в США был проведен анализ 1.2 млн. рыночных корзин в 25 магазинах формата «у дома» компании Osco, который выявил частый набор  для покупок в рабочие дни с 5 до 7 час. вечера

  - Руководство Osco не стало ставить эти товары рядом на полках (им были неясны причины такого частого набора)
  - *Объяснение*: молодая семья приходит с работы домой, жена отправляет мужа в ближайший магазин купить для ребенка  , а муж дополнительно покупает себе 

- 

Сотрудники Росгвардии в Челябинске задержали мужчину, подозреваемого в краже из супермаркета. «Уловом» грабителя стали три упаковки подгузников и три банки пива.

https://obzor174.ru/v-chelyabinske-muzhchina-stashchil-iz-supermarketa-pivo-i-podguzniki

Объекты, наборы, транзакции

- Объекты («товары»):

$$\mathcal{I} = \{i_1, \dots, i_m\}$$

- Наборы:

$$I \subseteq \mathcal{I},\ I \neq \emptyset;\ k\text{-набор: } I \subseteq \mathcal{I}, |I| = k$$

- Транзакции:

$$\mathcal{D} = \{(tid; I) \mid I \subseteq \mathcal{I}\}$$

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$\mathcal{I} = \{\text{Beer, Bread, Cola, Diaper, Milk}\}$

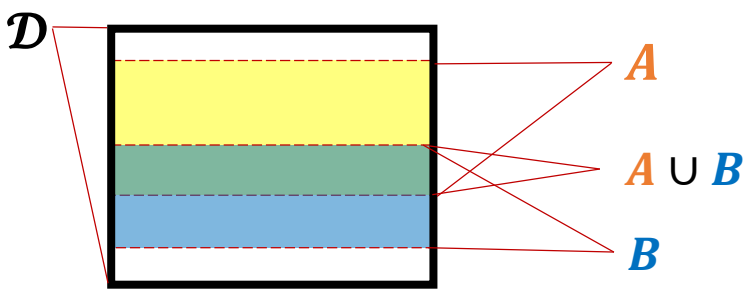
Шаблон, поддержка, достоверность

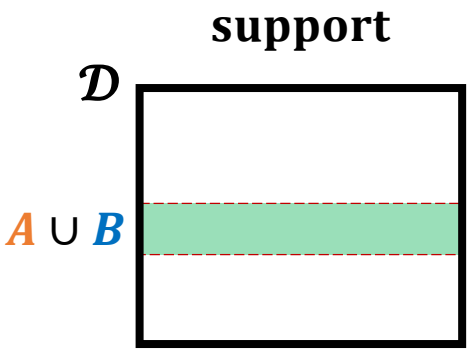
- Шаблон: $A \to B, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset$
- Поддержка

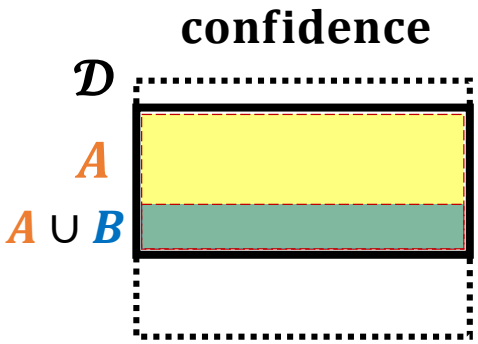
$$\sup(A \to B) = P(A \cup B) = \frac{|\{t \in \mathcal{D} \mid (\mathrm{A} \cup \mathrm{B}) \subseteq t.I\}|}{|\mathcal{D}|}$$

- Достоверность

$$\mathrm{conf}(A \to B) = P(B|A) = \frac{\sup(A \cup B)}{\sup(A)}$$







Поддержка и достоверность шаблона

- Поддержка:

$$\sup(A \to B) = P(A \cup B)$$

- Достоверность:

$$\text{conf}(A \to B) = P(B|A)$$

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	**Beer**, Cola, **Diaper, Milk**
4	**Beer**, Bread, **Diaper, Milk**
5	Bread, Cola, **Diaper, Milk**

$$\{\text{Diaper}, \text{Milk}\} \to \text{Beer}$$

$$sup = \frac{2}{5} = 0.4, \qquad \text{conf} = \frac{2}{3} = 0.67$$

Частый набор

- $minsup$ – порог поддержки (параметр)
- $I \subseteq \mathcal{I}$ – частый $\Leftrightarrow \sup(I) \geq minsup$
- Множество всех частых наборов: $\mathcal{L} = \bigcup_{k=1}^{k_{max}} \mathcal{L}_k$,
  $\mathcal{L}_k$ – множество частых k-наборов

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$$\boldsymbol{minsup = 0.6}$$

$$\mathcal{L}_1 = \{\text{Beer, Bread, Diaper, Milk}\}$$

$$\mathcal{L}_2 = \left\{\begin{matrix}\{\text{Beer, Diaper}\}, \{\text{Bread, Diaper}\}, \\ \{\text{Diaper, Milk}\}\end{matrix}\right\}$$

$$\boldsymbol{minsup = 0.1}$$

$$\mathcal{L}_3 = \left\{\begin{matrix}\{\text{Beer, Bread, Diaper}\}, \\ \{\text{Beer, Diaper, Milk}\}, \\ \{\text{Bread, Diaper, Milk}\}\end{matrix}\right\}$$

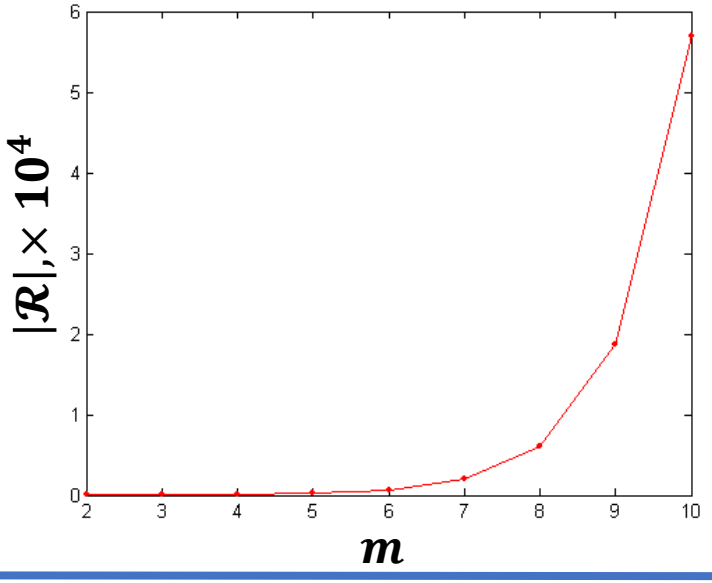
Устойчивый шаблон

- $minconf$ – порог достоверности (параметр)
- Шаблон $A \to B$ устойчив, если $\sup(A \to B) \geq minsup \wedge \text{conf}(A \to B) \geq minconf$
- **Задача поиска шаблонов**
  - Дано: $\mathcal{I} = \{i_1, \dots, i_m\}, \mathcal{D} = \{t_1, \dots, t_n\}, minsup, minconf$
  - Найти: $\mathcal{R} = \{A \to B \mid A, B \subseteq \mathcal{I}, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset, \sup(A \to B) \geq minsup, \text{conf}(A \to B) \geq minconf\}$

Поиск шаблонов: полный перебор

1. Сгенерируем $\mathcal{R} = \{A \to B \mid A, B \subseteq \mathcal{I}, A \neq \emptyset, B \neq \emptyset, A \cap B = \emptyset\}$
2. $\forall r \in \mathcal{R}$ вычислим $\sup(r)$, $\text{conf}(r)$
3. Отбросим $\forall r \in \mathcal{R}$: $\sup(r) < minsup$, $\text{conf}(r) < minconf$

- $|\mathcal{R}| = \sum_{k=1}^{m-1}\left[C_k^m \cdot \sum_{i=1}^{m-k} C_i^{m-k}\right] = 3^m - 2^{m+1} + 1$



m	$\lvert\mathcal{R}\rvert$
6	602
10	57 002

Поиск устойчивых шаблонов полным перебором вычислительно невозможен

Поиск шаблонов: идеи

TID	Items
1	Bread, Milk
2	Beer, Bread, Diaper, Eggs
3	Beer, Cola, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Bread, Cola, Diaper, Milk

$\{Milk,Diaper\} \rightarrow \{Beer\}$ (s=0.4, c=0.67)
$\{Milk,Beer\} \rightarrow \{Diaper\}$ (s=0.4, c=1.0)
$\{Diaper,Beer\} \rightarrow \{Milk\}$ (s=0.4, c=0.67)
$\{Beer\} \rightarrow \{Milk,Diaper\}$ (s=0.4, c=0.67)
$\{Diaper\} \rightarrow \{Milk,Beer\}$ (s=0.4, c=0.5)
$\{Milk\} \rightarrow \{Diaper,Beer\}$ (s=0.4, c=0.5)

- Шаблоны примера – разбиения одного набора
- Разбиения одного набора имеют равную поддержку, но могут иметь разную достоверность
- Для поиска шаблонов обработку поддержки и достоверности можно отделить друг от друга

Поиск шаблонов: алгоритм

1. Найдем все частые наборы (с поддержкой не ниже $minsup$)
2. Сгенерируем шаблоны, выполняя разбиение каждого частого набора; устойчивыми будут шаблоны с достоверностью не ниже $minconf$

Поиск шаблонов: алгоритм

$\mathcal{L} := \cup_{k=1}^{k_{max}} \{I \subseteq \mathcal{I} \mid |I| = k, \sup(I) \geq minsup\}$

$\mathcal{R} := \emptyset$

for all $I \in \mathcal{L}$

  for all $S \in \mathcal{P}(I) \setminus \emptyset$ **do**

    $conf := \frac{\sup(I)}{\sup(S)}$

    if $conf \geq minconf$ **then**

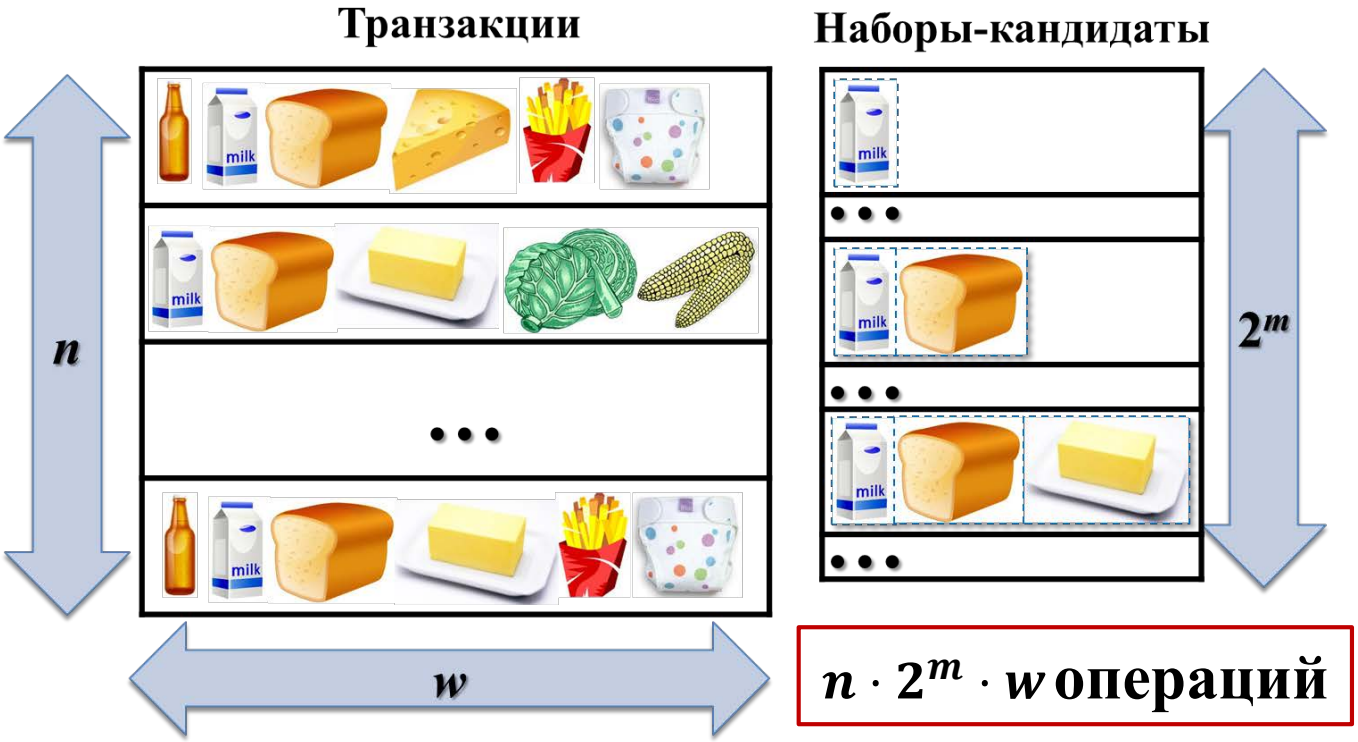
      $pattern := "S \rightarrow I \setminus S"$

      $\mathcal{R} := \mathcal{R} \cup pattern$

$\mathcal{P}(I)$ – множество всех подмножеств I

Поиск частых наборов: полный перебор

1. Сгенерируем кандидатов в частые наборы: $\mathcal{C} = \mathcal{P}(\mathcal{I}) \setminus \emptyset$
2. $\forall c \in \mathcal{C}$ вычислим $\sup(c)$
3. Отбросим $\forall c \in \mathcal{C}$: $\sup(c) < minsup$



$n \cdot 2^m \cdot w$ операций

Поиск частых наборов полным перебором вычислительно невозможен

Отбрасывание заведомо редких наборов

- Антимонотонность поддержки

$$\forall X, Y\colon X \subseteq Y \Leftrightarrow \sup(X) \geq \sup(Y)$$

  – поддержка набора всегда не больше поддержки любого своего подмножества

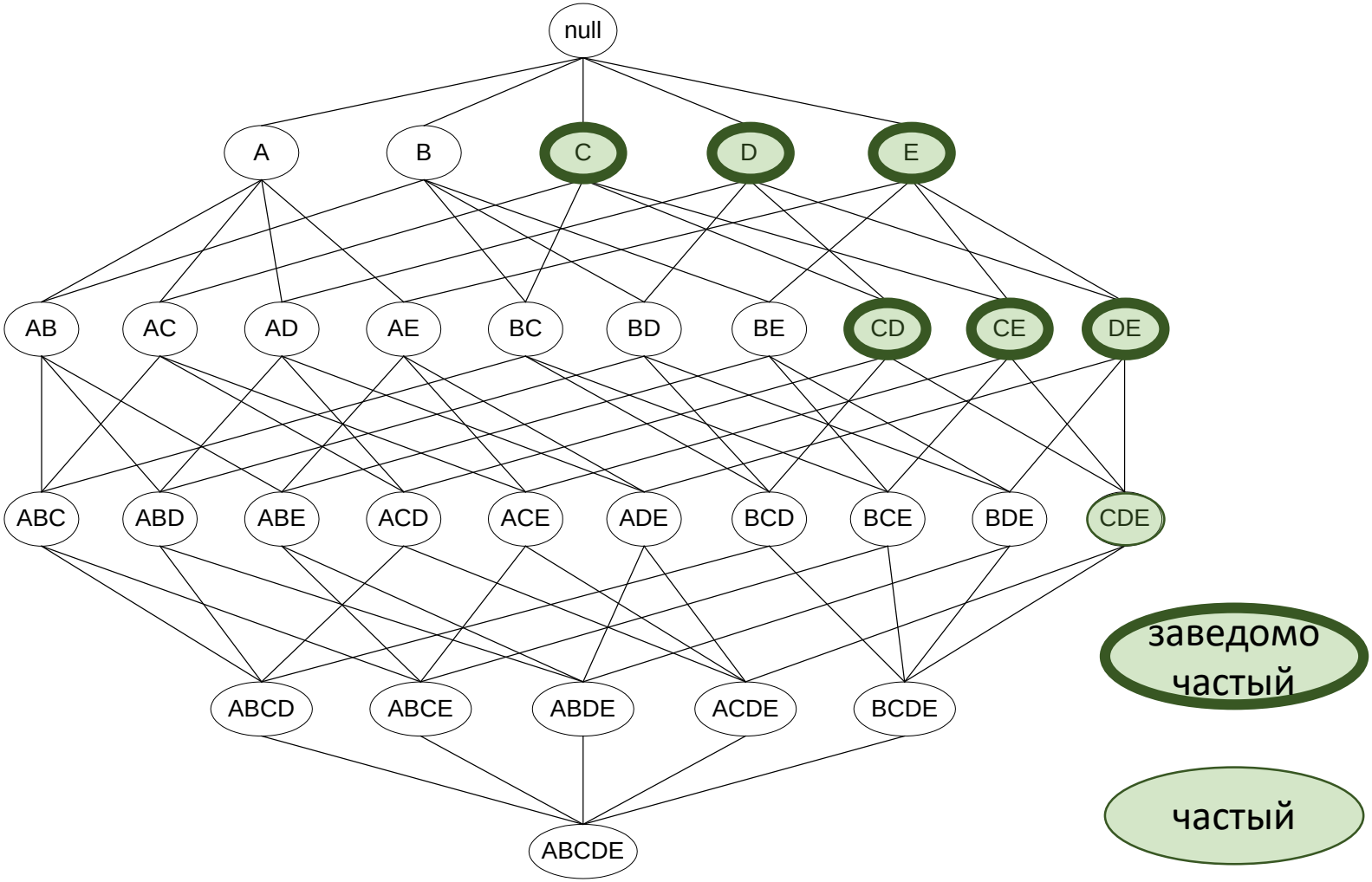
- Принцип Априори
  – любое подмножество частого набора является частым набором

$$\sup(Y) \geq minsup \Leftrightarrow \forall X \subseteq Y \sup(X) \geq minsup$$

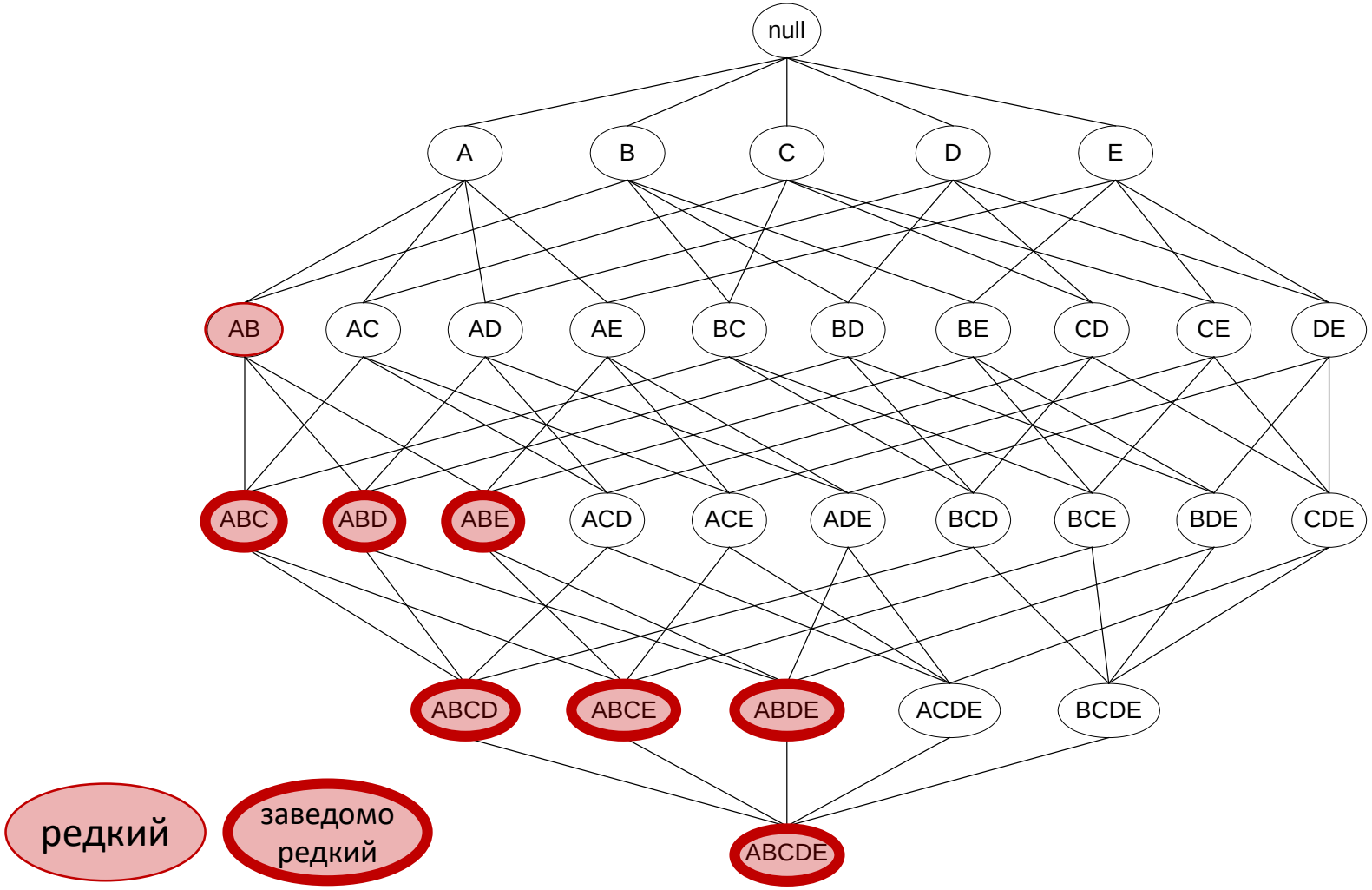
  – если некое подмножество набора является редким набором, то набор является редким

$$\sup(Y) < minsup \Leftrightarrow \exists X \subseteq Y \sup(X) < minsup$$

Принцип Априори



Принцип Априори



Генерация наборов-кандидатов

- $k = 1$: $C_1 := \mathcal{I}$ (множество 1-наборов)
- $k = 2$: $C_2 := \mathcal{L}_1 \times \mathcal{L}_1$
  - Декартово произведение множества частых 1-наборов на себя
- $k \geq 3$:
  - Соединение множества частых $(k-1)$-наборов с самим собой
    - $C_k := \mathcal{L}_{k-1} \bowtie_\Theta \mathcal{L}_{k-1}$
      $X = (x_1, \ldots, x_{k-1})$, $Y = (y_1, \ldots, y_{k-1})$, $X, Y \in \mathcal{L}_{k-1}$, элементы в наборах лексикографически упорядочены
      $$\Theta = (\wedge_{i=1}^{k-2} x_i = y_i) \wedge (x_{k-1} < y_{k-1})$$
    - $X \bowtie_\Theta Y = (x_1, \ldots, x_{k-2}, x_{k-1}, y_{k-1})$
  - Отбрасывание заведомо редких наборов в C_k по принципу Априори
    - если $I \subseteq C_k$, но $I \notin \mathcal{L}_{k-1}$, то отбросить I

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $$C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$

  Вычисление поддержки

  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$

  Декартово произведение

  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$

  Вычисление поддержки

  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$

- $\mathcal{L}_2 = \{\mathbf{A}\mathrm{B}, \mathbf{A}\mathrm{C}, \mathbf{A}\mathrm{D}, \mathbf{A}\mathrm{E}, \mathbf{B}\mathrm{C}, \mathbf{B}\mathrm{D}, \mathbf{B}\mathrm{E}, \mathbf{C}\mathrm{D}, \mathbf{C}\mathrm{E}\}$

  Самосоединение

  $C_3 = \{\mathbf{A}\mathrm{BC}, \mathbf{A}\mathrm{BD}, \mathbf{A}\mathrm{BE}, \mathbf{A}\mathrm{CD}, \mathbf{A}\mathrm{CE}, \mathbf{A}\mathrm{DE}, \mathbf{B}\mathrm{CD}, \mathbf{B}\mathrm{CE}, \mathbf{B}\mathrm{DE}, \mathbf{C}\mathrm{DE}\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{A, B, C, D, E\}$, $\mathcal{L}_1 = \{A, B, C, D, E\}$
  Декартово произведение
  $C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$
  Вычисление поддержки
  $\mathcal{L}_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, \sout{DE}\}$
- $\mathcal{L}_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE\}$
  Самосоединение
  $C_3 = \{ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE\}$
  Вычисление поддержки
  $\mathcal{L}_3 = \{ABC, ABD, ABE, ACD, \sout{ACE}, \sout{ADE}, BCD, \sout{BCE}, BDE, CDE\}$
- $\mathcal{L}_3 = \{\mathbf{AB}C, \mathbf{AB}D, \mathbf{AB}E, ACD, BCD, BDE, CDE\}$
  Самосоединение
  $C_4 = \{\mathbf{AB}CD, \mathbf{AB}CE, \mathbf{AB}DE\}$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{A, B, C, D, E\}$, $\mathcal{L}_1 = \{A, B, C, D, E\}$
  Декартово произведение
  $C_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$
  Вычисление поддержки
  $\mathcal{L}_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, \sout{DE}\}$
- $\mathcal{L}_2 = \{AB, AC, AD, AE, BC, BD, BE, CD, CE\}$
  Самосоединение
  $C_3 = \{ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE\}$
  Вычисление поддержки
  $\mathcal{L}_3 = \{ABC, ABD, ABE, ACD, \sout{ACE}, \sout{ADE}, BCD, \sout{BCE}, BDE, CDE\}$
- $\mathcal{L}_3 = \{\mathbf{ABC}, \mathbf{ABD}, \mathbf{ABE}, ACD, BCD, BDE, CDE\}$
  Самосоединение
  $C_4 = \{\mathbf{ABCD}, \mathbf{ABCE}, \mathbf{ABDE}\}$
  Отбрасывание
  $C_4 = \{\mathbf{ABCD}, A\mathbf{BCE}, \mathbf{A}B\mathbf{DE}\}$ и $\mathbf{BCE} \notin \mathcal{L}_3$, $\mathbf{ADE} \notin \mathcal{L}_3$

Генерация наборов-кандидатов: пример

- $C_1 = \mathcal{I} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$, $\mathcal{L}_1 = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$
  Декартово произведение
  $C_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \mathrm{DE}\}$
  Вычисление поддержки
  $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}, \sout{\mathrm{DE}}\}$
- $\mathcal{L}_2 = \{\mathrm{AB}, \mathrm{AC}, \mathrm{AD}, \mathrm{AE}, \mathrm{BC}, \mathrm{BD}, \mathrm{BE}, \mathrm{CD}, \mathrm{CE}\}$
  Самосоединение
  $C_3 = \{\mathrm{ABC}, \mathrm{ABD}, \mathrm{ABE}, \mathrm{ACD}, \mathrm{ACE}, \mathrm{ADE}, \mathrm{BCD}, \mathrm{BCE}, \mathrm{BDE}, \mathrm{CDE}\}$
- $\mathcal{L}_3 = \{\mathbf{AB}\mathrm{C}, \mathbf{AB}\mathrm{D}, \mathbf{AB}\mathrm{E}, \mathrm{ACD}, \mathrm{BCD}, \mathrm{BDE}, \mathrm{CDE}\}$
  Самосоединение
  $C_4 = \{\mathbf{AB}\mathrm{CD}, \mathbf{AB}\mathrm{CE}, \mathbf{AB}\mathrm{DE}\}$
  Отбрасывание
  $C_4 = \{\mathbf{AB}\mathrm{CD}, \mathrm{A}\mathbf{BCE}, \mathrm{A}\mathbf{BDE}\} \Rightarrow C_4 = \{\mathrm{ABCD}\}$

Алгоритм Априори

Input:

- D, a database of transactions;
- min_sup, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)    L1 = find_frequent_1-itemsets(D);
(2)    for (k = 2; Lk−1 ≠ φ; k++) {
(3)        Ck = apriori_gen(Lk−1);
(4)        for each transaction t ∈ D { // scan D for counts
(5)            Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)            for each candidate c ∈ Ct
(7)                c.count++;
(8)        }
(9)        Lk = {c ∈ Ck | c.count ≥ min_sup}
(10)   }
(11)   return L = ∪k Lk;
```

Алгоритм Априори

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)     for each itemset l_1 ∈ L_{k-1}
(2)         for each itemset l_2 ∈ L_{k-1}
(3)             if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2])
                    ∧... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then {
(4)                 c = l_1 ⋈ l_2; // join step: generate candidates
(5)                 if has_infrequent_subset(c, L_{k-1}) then
(6)                     delete c; // prune step: remove unfruitful candidate
(7)                 else add c to C_k;
(8)             }
(9)     return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
          L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1)     for each (k-1)-subset s of c
(2)         if s ∉ L_{k-1} then
(3)             return TRUE;
(4)     return FALSE;
```

```
(1)   L_1 = find_frequent_1-itemsets(D);
(2)   for (k = 2; L_{k-1} ≠ ϕ; k++) {
(3)       C_k = apriori_gen(L_{k-1});
(4)       for each transaction t ∈ D { // scan D for counts
(5)           C_t = subset(C_k, t); // get the subsets of t that are candidates
(6)           for each candidate c ∈ C_t
(7)               c.count++;
(8)       }
(9)       L_k = {c ∈ C_k | c.count ≥ min_sup}
(10)  }
(11)  return L = ∪_k L_k;
```

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$C_1 = \mathcal{I}$

Items	SUP
I1	
I2	
I3	
I4	
I5	
I6	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$C_1 = \mathcal{I}$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2
I6	1

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

C_1

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2
I6	**1**

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	
I1,I3	
I1,I4	
I1,I5	
I2,I3	
I2,I4	
I2,I5	
I3,I4	
I3,I5	
I4,I5	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	4
I1,I3	4
I1,I4	1
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	0
I3,I5	1
I4,I5	0

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$C_2 = \mathcal{L}_1 \times \mathcal{L}_1$

Items	SUP
I1,I2	4
I1,I3	4
I1,I4	**1**
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2
I3,I4	**0**
I3,I5	**1**
I4,I5	**0**

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	
I1,I2,I5	
I1,I3,I5	
I2,I3,I4	
I2,I3,I5	
I2,I4,I5	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	
I1,I2,I5	
I1,**I3,I5**	
I2,**I3,I4**	
I2,**I3,I5**	
I2,**I4,I5**	

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$C_3 = \mathcal{L}_2 \bowtie \mathcal{L}_2$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

Алгоритм Apriori: пример

$\boldsymbol{D}, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

$C_4 = \mathcal{L}_3 \bowtie \mathcal{L}_3$

Items	SUP
I1,**I2,I3,I5**	

$\mathcal{L}_4 = \emptyset$

Items	SUP

Алгоритм Apriori: пример

$D, minsup = 0.2$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}$

$\mathcal{L}_1$

Items	SUP
I1	6
I2	7
I3	6
I4	2
I5	2

$\cup$

$\mathcal{L}_2$

Items	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2

$\cup$

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	2

Генерация шаблонов: пример

D,
$minsup = 0.2$
$minconf = 0.7$

TID	Items
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

$\mathcal{L}_3$

Items	SUP
I1,I2,I3	2
I1,I2,I5	**2**

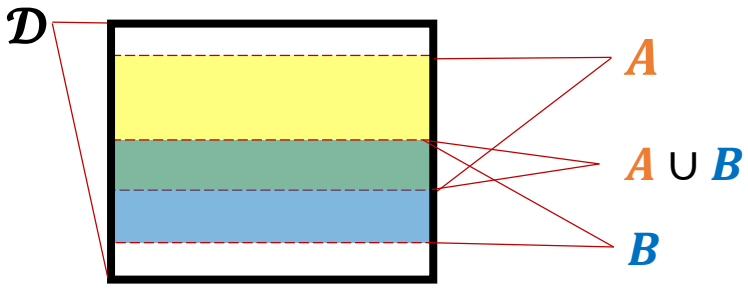
Шаблоны

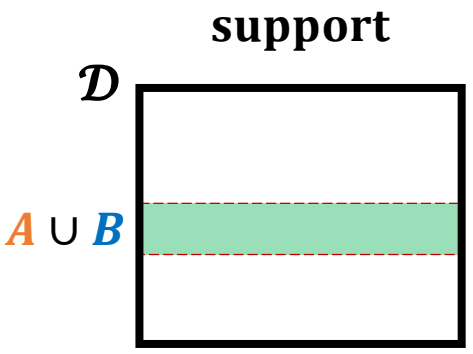
для {I1,I2,I5}

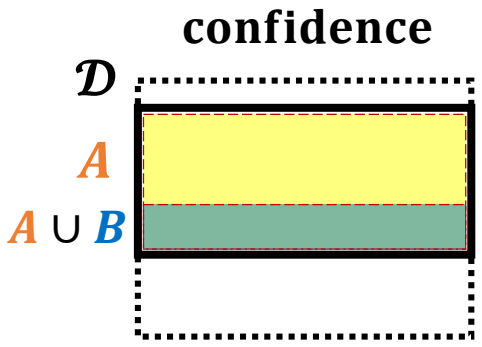
Rule	CONF
{I1,I2}→I5	2/4
{I1,I5}→I2	**2/2**
{I2,I5}→I1	**2/2**
I1→{I2,I5}	2/6
I2→{I1,I5}	2/7
I5→{I1,I2}	**2/2**

Полезность шаблонов

№	Шаблон	sup	conf	Польза
1	хлеб → масло	👍	👍	?
2	хлеб → авокадо	👍	👍	?
3	молоко → тунец	👍	👎	?
4	масло → стир. порошок	👎	👎	?
5	водка → икра красная	👎	👍	?
6	водка → икра баклажанная	👎	👍	?

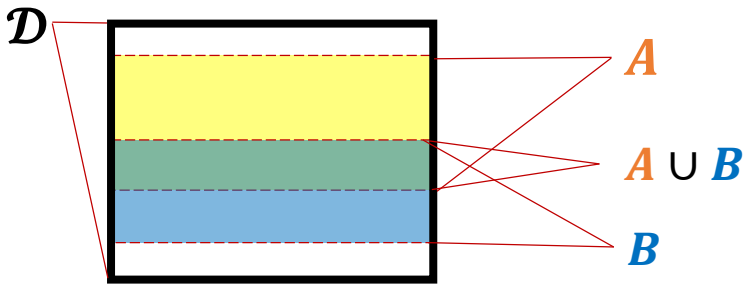


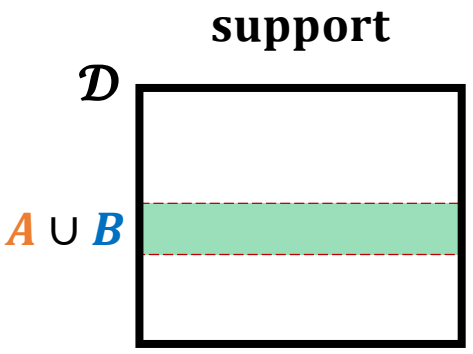


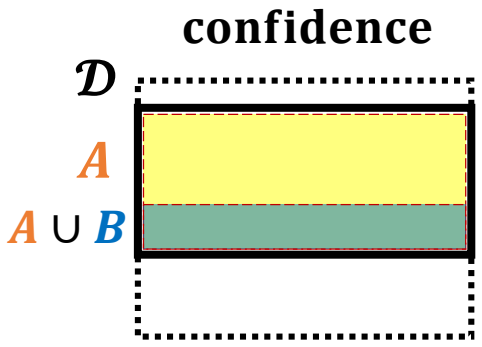


Полезность шаблонов

№	Правило	sup	conf	Польза
1	хлеб → масло	👍	👍	👎
2	хлеб → авокадо	👍	👍	👍
3	молоко → тунец	👍	👎	👎
4	масло → стир. порошок	👎	👎	👎
5	водка → икра красная	👎	👍	👎
6	водка → икра баклажанная	👎	👍	👍







Пример: шаблон устойчив, но бесполезен

- Таблица сопряженности

	$coffee$	$\overline{coffee}$	Σ
tea	4000	3500	7500
$\overline{tea}$	2000	500	2500
Σ	6000	4000	10000

- $coffee \rightarrow tea\ [sup = \frac{4000}{10000} = 0.4, conf = \frac{4000}{6000} = 0.66]$
- $\mathrm{P}(tea) = \frac{7500}{10000} = 0.75 > \mathrm{P}(tea|coffee) = conf$
- Проще угадать наличие tea, чем предсказать это с помощью шаблона

Пример: шаблон неустойчив, но полезен

- Таблица сопряженности

	$\boldsymbol{honey}$	$\overline{\boldsymbol{honey}}$	Σ
$\boldsymbol{tea}$	100	100	200
$\overline{\boldsymbol{tea}}$	20	780	800
Σ	120	880	1000

- $tea \rightarrow honey\ [sup = \frac{100}{1000} = 0.1, conf = \frac{100}{200} = 0.5]$
- $\mathrm{P}(honey) = \frac{120}{1000} = 0.12 < \mathrm{P}(honey|tea) = conf$
- $\mathrm{P}(\overline{tea} \cup honey) = \frac{20}{1000} = 0.025$
- Отбрасывание шаблона не учитывает информацию о предпочтении $honey$ при условии tea

Мера lift – дополнительная к sup и conf

- $\text{lift}(A, B) = \frac{P(A \cup B)}{P(A) \cdot P(B)}$
- Оценка
  - $\text{lift}(A, B) = 1$: A и B независимы
  - $\text{lift}(A, B) < 1$: A и B имеют отрицательную корреляцию (наличие A подразумевает отсутствие B)
  - $\text{lift}(A, B) > 1$: A и B имеют положительную корреляцию (наличие A подразумевает наличие B)
- $\text{lift}(A \to B) = \frac{P(B|A)}{P(B)} = \frac{\text{conf}(A \to B)}{\text{sup}(B)} = \frac{\text{sup}(A \to B)}{\text{sup}(A) \cdot \text{sup}(B)}$

Мера lift

- Пример:
  - Таблица сопряженности

	A	$\overline{A}$	Σ
B	4000	3500	7500
$\overline{B}$	2000	500	2500
Σ	6000	4000	10000

  - $A \rightarrow B$ $[sup = 0.4, conf = 0.66]$
- $\text{lift}(A, B) = \frac{P(A \cup B)}{P(A) \cdot P(B)} = \frac{0.4}{0.6 \cdot 0.75} = 0.89$
  - A и B имеют отрицательную корреляцию
  - $A \rightarrow B$ малополезно

Литература

- Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd Edition. Morgan Kaufmann, 2011. 740 p. ISBN 978-0123814791
  - Chapter 6. Mining Frequent Patterns, Associations, and Correlations: Basic Concepts and Methods, pp. 243-278
- Tan P.-N., Steinbach M., Karpatne A., Kumar V. Introduction to Data Mining. 2nd Edition. Pearson, 2019. 839 p. ISBN-13: 978-0-13-312890-1
  - 5. Association Analysis: Basic Concepts and Algorithms, pp. 357-450