

Современные методы и алгоритмы интеллектуального анализа данных

(часть I)

М.Л. Цымблер

Содержание

- Понятие интеллектуального анализа данных
- Базовые задачи интеллектуального анализа данных
 - Классификация
 - Кластеризация
 - Поиск шаблонов
- Некоторые алгоритмы поиска шаблонов
- Интеграция интеллектуального анализа данных и СУБД

Интеллектуальный анализ данных

- Совокупность алгоритмов и методов для обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений.
- Английские термины
 - Data Mining
 - Knowledge Discovery in Databases

Данные, подвергаемые анализу

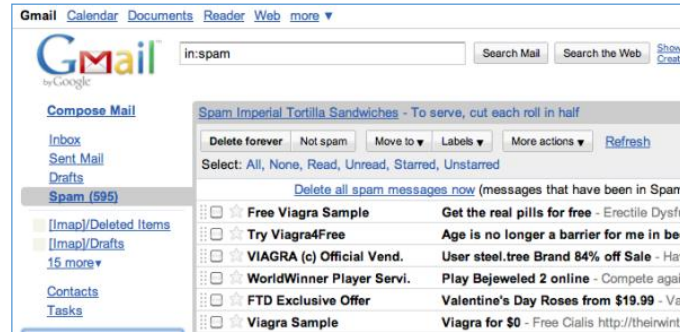
- Базы данных
 - Реляционные таблицы
- Хранилища данных
 - Многомерные массивы
- Транзакционные данные
 - «Ключ-значение»
- Временные ряды
 - Хронологически упорядоченные последовательности
- Геопространственные данные
 - GPS координаты, ассоциированные с др. информацией
- Мультимедиа
 - Видео, аудио, изображения
- Тексты
 - TXT, HTML, XML
- Графы
 - WWW, соц. сети, биологические сети, RDF
- ...

Базовые задачи анализа данных

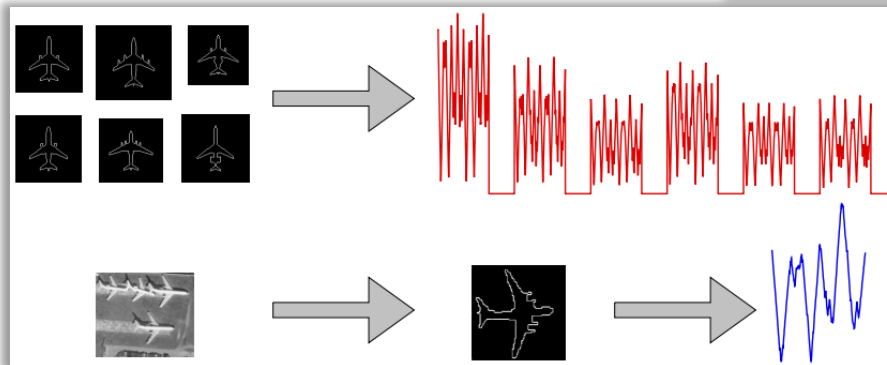
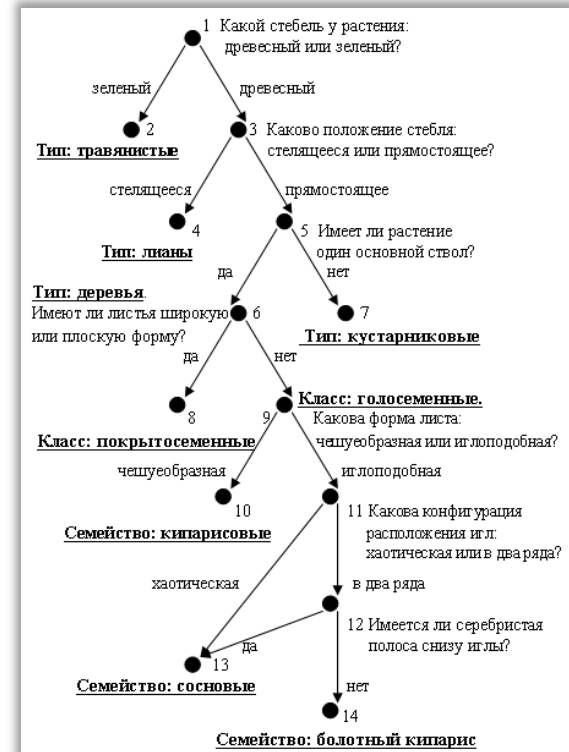
- Классификация (classification)
 - Разбиение объектов на группы схожести, семантика групп заранее известна
- Кластеризация (clustering)
 - Разбиение объектов на группы схожести, семантика групп заранее НЕ известна
- Поиск шаблонов (frequent patterns)
 - Нахождение частых зависимостей между объектами

Примеры задач классификации

- Спам-фильтр
- Предсказание оттока клиентов
- Биологическая классификация
- Определение типа наземного объекта по аэрофотоснимку

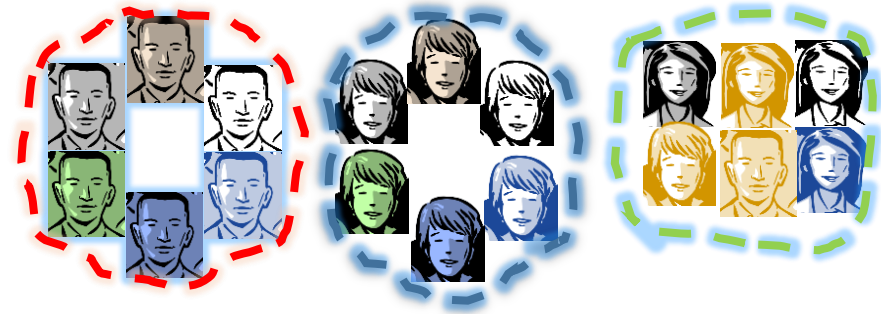


Talk	Text	Purchases	Data	Age	Churn?
148	72	0	33.6	50	TRUE
85	66	0	26.6	31	FALSE
183	64	0	23.3	32	TRUE
89	66	94	28.1	21	FALSE
115	0	0	35.3	29	FALSE
166	72	175	25.8	51	TRUE
100	0	0	30	32	TRUE
118	84	230	45.8	31	TRUE
171	110	240	45.4	54	TRUE
159	64	0	27.4	40	FALSE

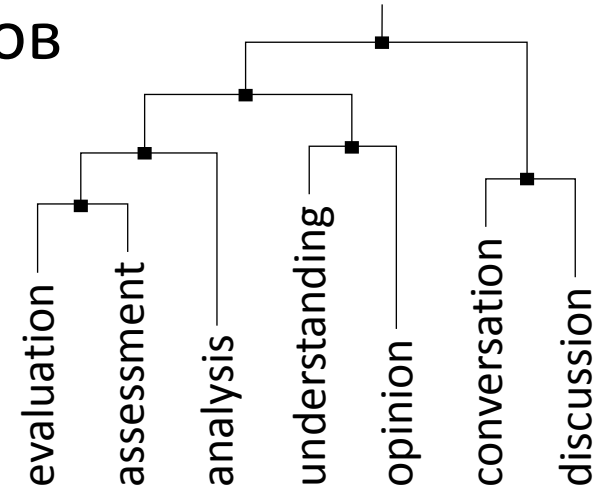


Примеры задач кластеризации

- Таргетирование
 - Различная реклама, акции, скидки для различных целевых групп



- Автоматическая рубрикация документов



Примеры задач на поиск шаблонов

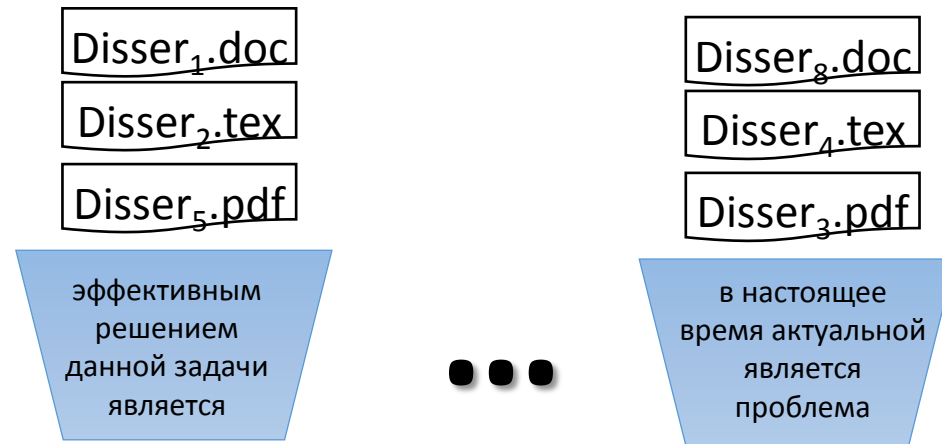
○ Супермаркет: какие товары часто покупают совместно?

- Мерчендайзинг
- Дисконтные программы
- Подсказки «с этим товаром часто покупают ...»



○ Плагиат или нет?

- Предложения/фразы и документы, содержащие их
- Документы, которые часто встречаются совместно, указывают на плагиат



Поиск шаблонов: определения

- Объект (*item*). Набор (*itemset*) – конечное множество объектов.
- *k*-набор (*k-itemset*) – множество из $k \geq 1$ объектов.
- Транзакция – пара (TID, *k*-набор). Набор транзакций – *D*.
- Поддержка набора (*itemset support*)
 - абсолютная – количество транзакций, в которые входит набор
 - относительная – доля транзакций, в которые входит набор (вероятность того, что транзакция содержит данный набор).

D

TID	Набор
10	орешки, пиво, подгузники
20	кофе, пиво, подгузники
30	кофе, пиво, подгузники, хлеб
40	молоко, орешки, хлеб
50	кофе, молоко, орешки, подгузники, хлеб

○ Пример:

- $sup(\{\text{пиво}\}) = 3/5$
- $sup(\{\text{орешки, пиво}\}) = 1/5$
- $sup(\{\text{кофе, подгузники}\}) = 2/5$
- $sup(\{\text{пиво, подгузники}\}) = 3/5$
- $sup(\{\text{кофе, подгузники, хлеб}\}) = 2/5$

Задача рыночной корзины

- Минимальная поддержка (*min support*) – пороговое значение поддержки.
- Частый k -набор (*frequent itemset*) L_k – k -набор с поддержкой не ниже *minsup*.
- Задача 1: анализ рыночной корзины
 - для заданных D и *minsup* найти все L_k ($k \geq 1$).

D

TID	Набор
10	орешки, пиво, подгузники
20	кофе, пиво, подгузники
30	кофе, пиво, подгузники, хлеб
40	молоко, орешки, хлеб
50	кофе, молоко, орешки, подгузники, хлеб

○ Пример:

- При *minsup*=3
 - $L_1 = \{\{\text{орешки}\}, \{\text{пиво}\}, \{\text{подгузники}\}, \{\text{хлеб}\}\}$
 - $L_2 = \{\{\text{пиво, подгузники}\}\}$
- При *minsup*=2
 - $L_1 = \{\{\text{кофе}\}, \{\text{молоко}\}, \{\text{орешки}\}, \{\text{пиво}\}, \{\text{подгузники}\}, \{\text{хлеб}\}\}$
 - $L_2 = \{\{\text{кофе, пиво}\}, \{\text{пиво, подгузники}\}, \dots\}$
 - $L_3 = \{\{\text{кофе, подгузники, хлеб}\}\}$

Алгоритм Apriori: идеи

- Последовательность шагов
 - Найти L_1
 - Для всех $k \geq 2$ использовать L_{k-1} при нахождении L_k .
- Кандидаты
 - На основе L_{k-1} генерировать C_k , множество наборов-кандидатов в частые наборы и вычислять их поддержку.
- Отбрасывание
 - Вычислять поддержку не для всех кандидатов, отбрасывать заведомо не частые наборы-кандидаты.
 - Количество кандидатов n -элементного множества объектов: $2^n - 1$.

Генерация кандидатов

- Полагаем объекты в наборе L_{k-1} упорядоченными
 - I_1, I_2 — наборы из L_{k-1}
 - $I_i[1] < I_i[2] < \dots < I_i[k-1]$
- $C_k = L_{k-1} \text{ JOIN } L_{k-1}$
 - **условие соединения:**
 $(I_1[1]=I_2[1]) \text{ and } (I_1[2]=I_2[2]) \dots \text{ and } (I_1[k-2]=I_2[k-2]) \text{ and } (I_1[k-1]<I_2[k-1])$
 - **результат соединения:**
 $\{I_1[1], I_1[2], \dots, I_1[k-2], I_1[k-1], I_2[k-1]\}$
- **Пример:**
 - $L_3 = \{abc, abd, acd, ace, bcd\}$
 - $C_4 = L_3 \text{ JOIN } L_3 = \{abcd, acde\}$

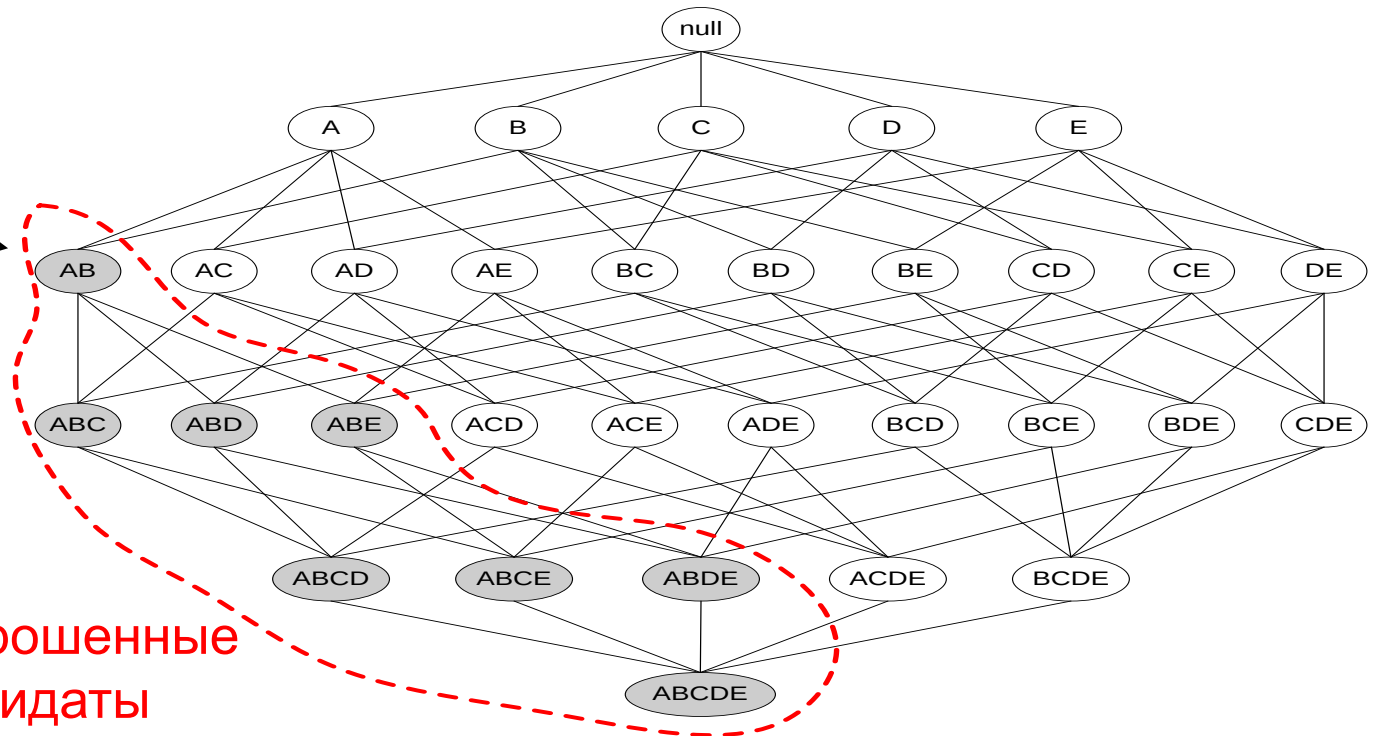
Отбрасывание кандидатов (принцип Apriori)

- Любое непустое подмножество частого набора должно быть частым набором
 - Если набор не является частым, то все содержащие его наборы также не являются частыми: если набор из L_{k-1} не является частым, то он не входит в C_k

Пример

Не является частым

Отброшенные кандидаты



Алгоритм Apriori: пример

D , $minsup=2$

TID	Набор
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

$D, \text{minsup}=2$

C_1



TID	Набор
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

Набор	SUP

Алгоритм Apriori: пример

$D, \text{minsup}=2$

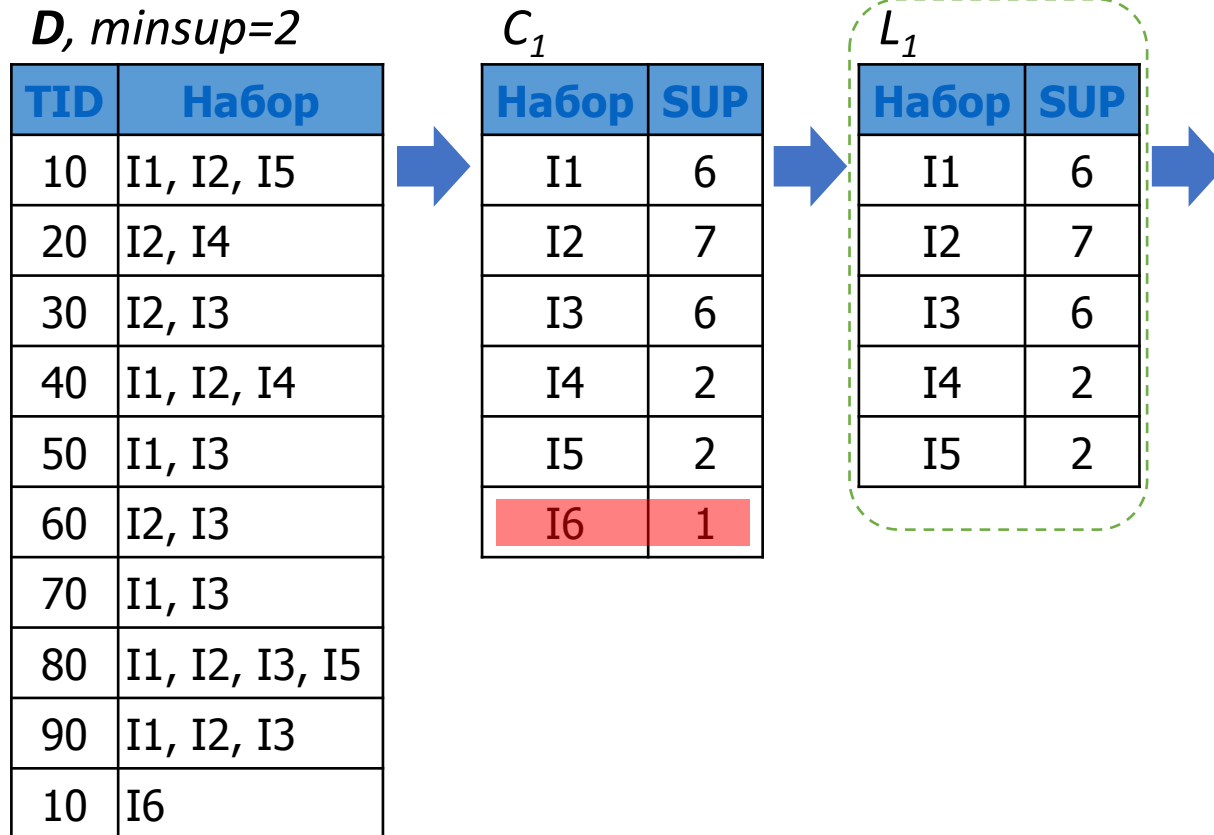
C_1



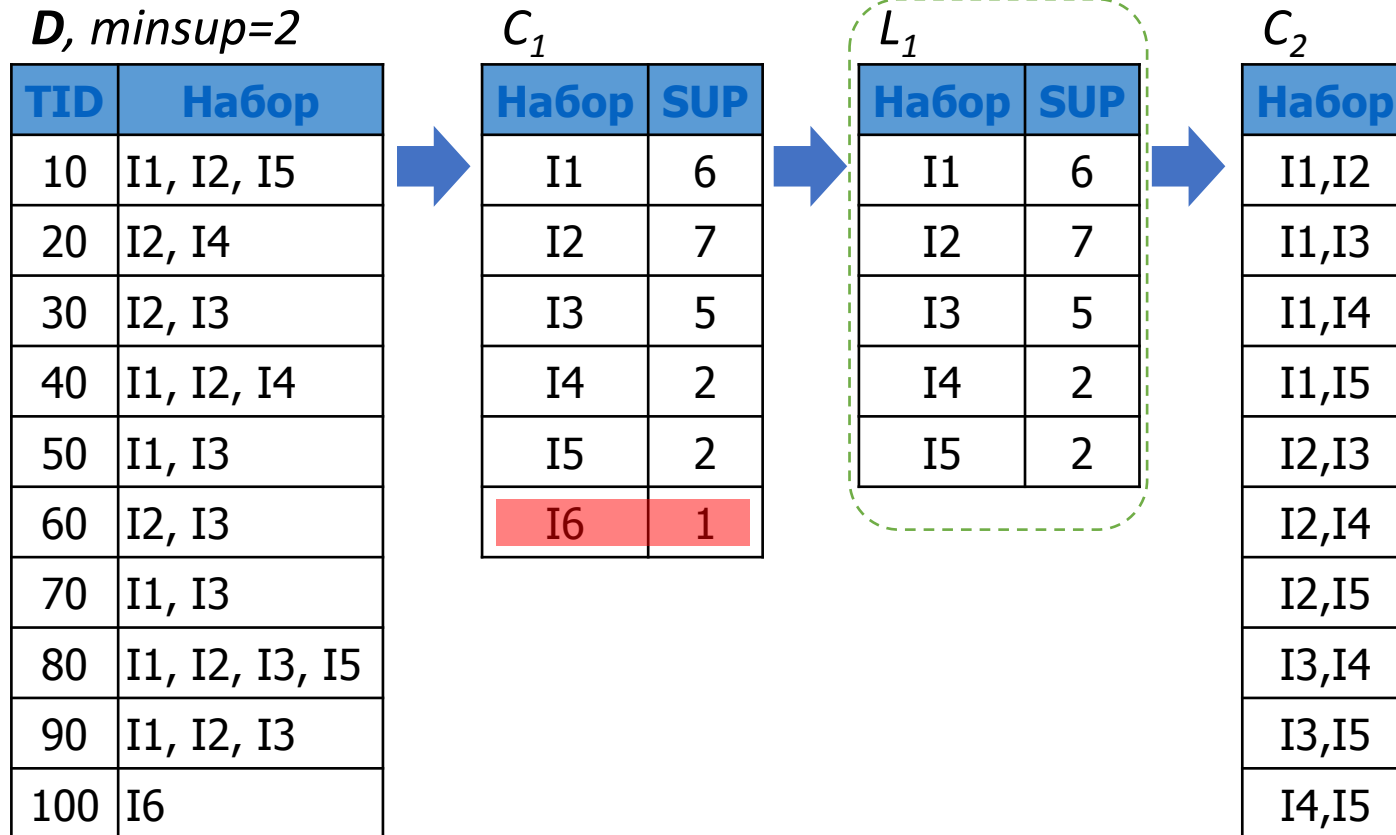
TID	Набор
10	I1, I2, I5
20	I2, I4
30	I2, I3
40	I1, I2, I4
50	I1, I3
60	I2, I3
70	I1, I3
80	I1, I2, I3, I5
90	I1, I2, I3
100	I6

Набор	SUP
I1	6
I2	7
I3	5
I4	2
I5	2
I6	1

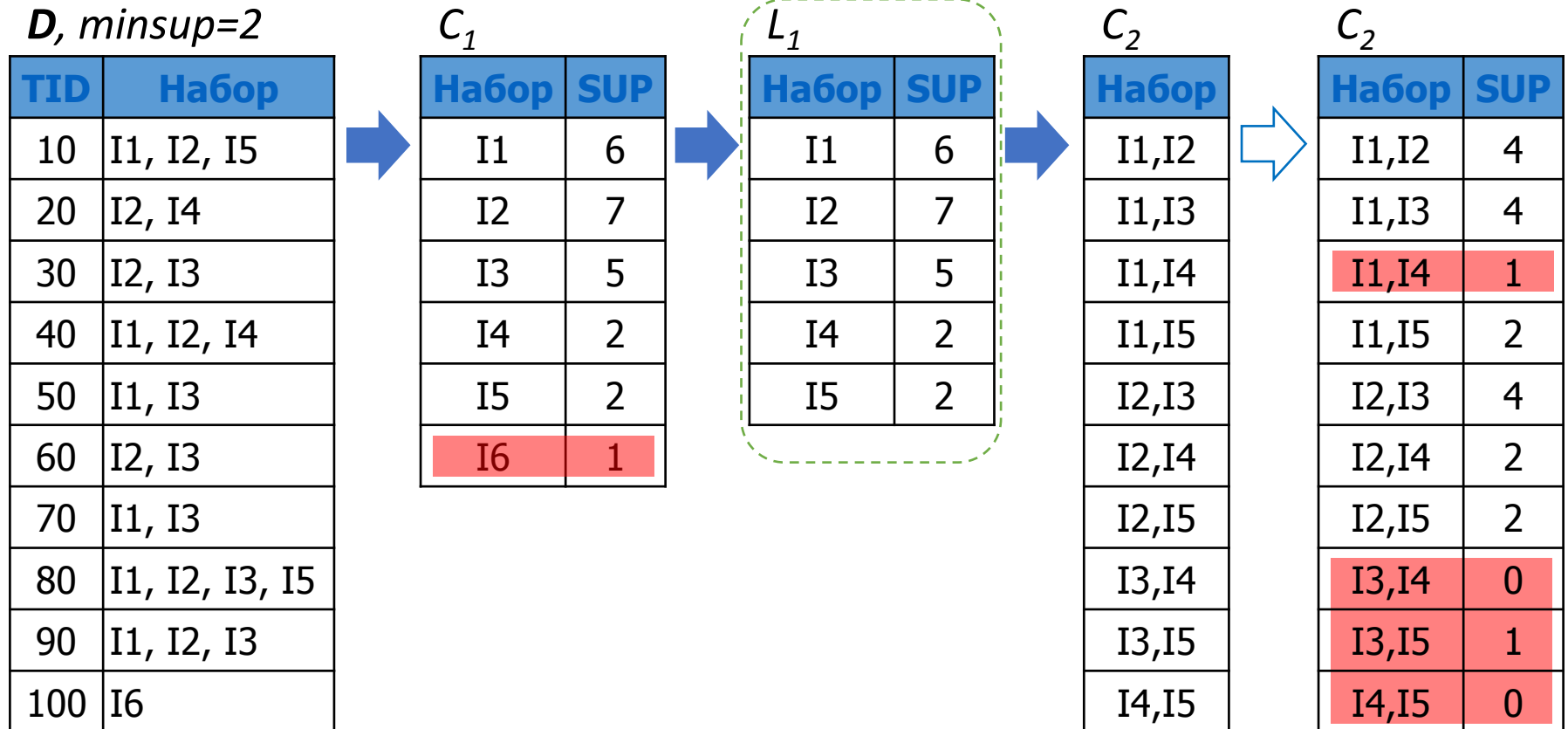
Алгоритм Apriori: пример



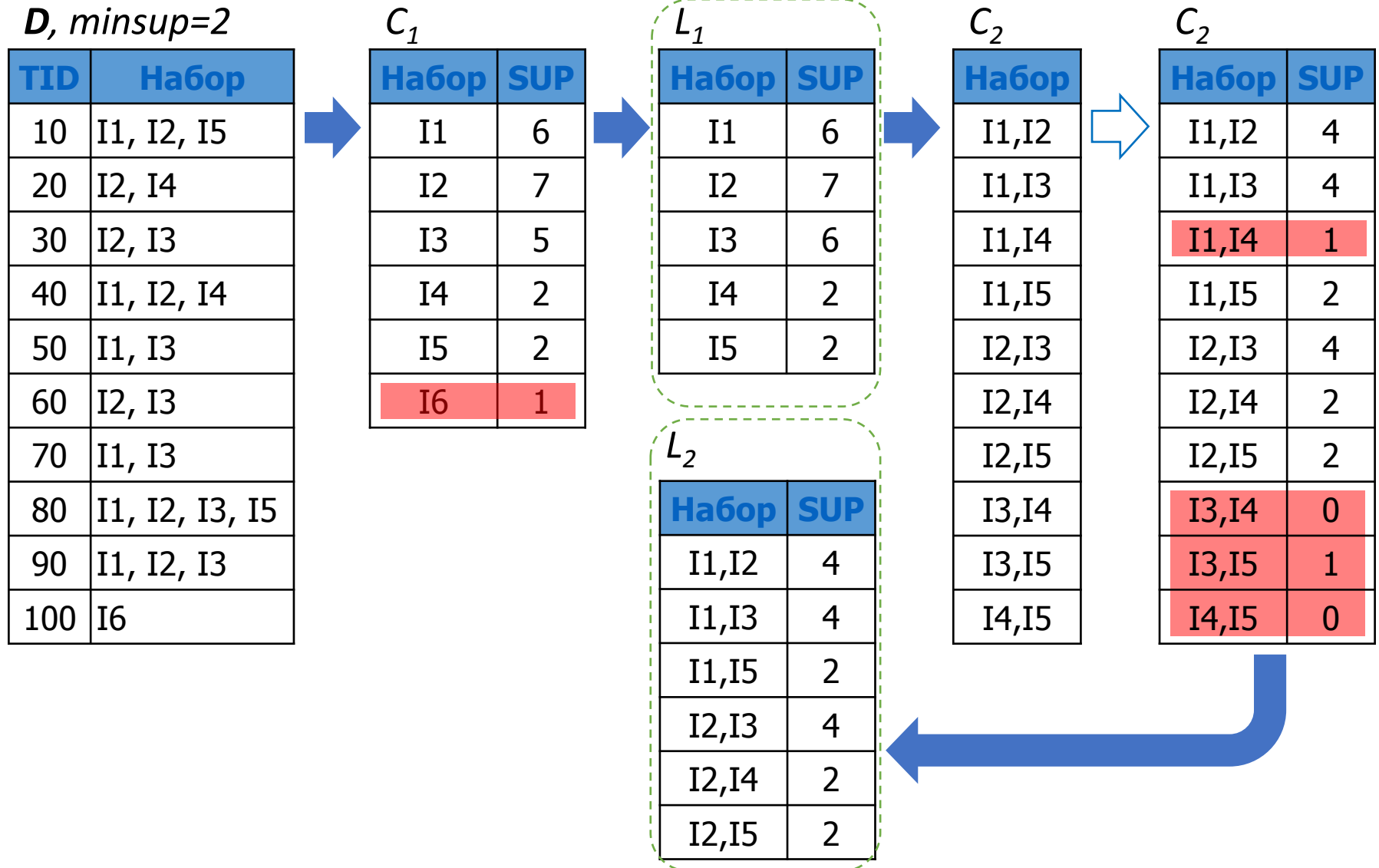
Алгоритм Apriori: пример



Алгоритм Apriori: пример



Алгоритм Apriori: пример



Алгоритм Apriori: пример

$D, \text{minsup}=2$

TID	Набор
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

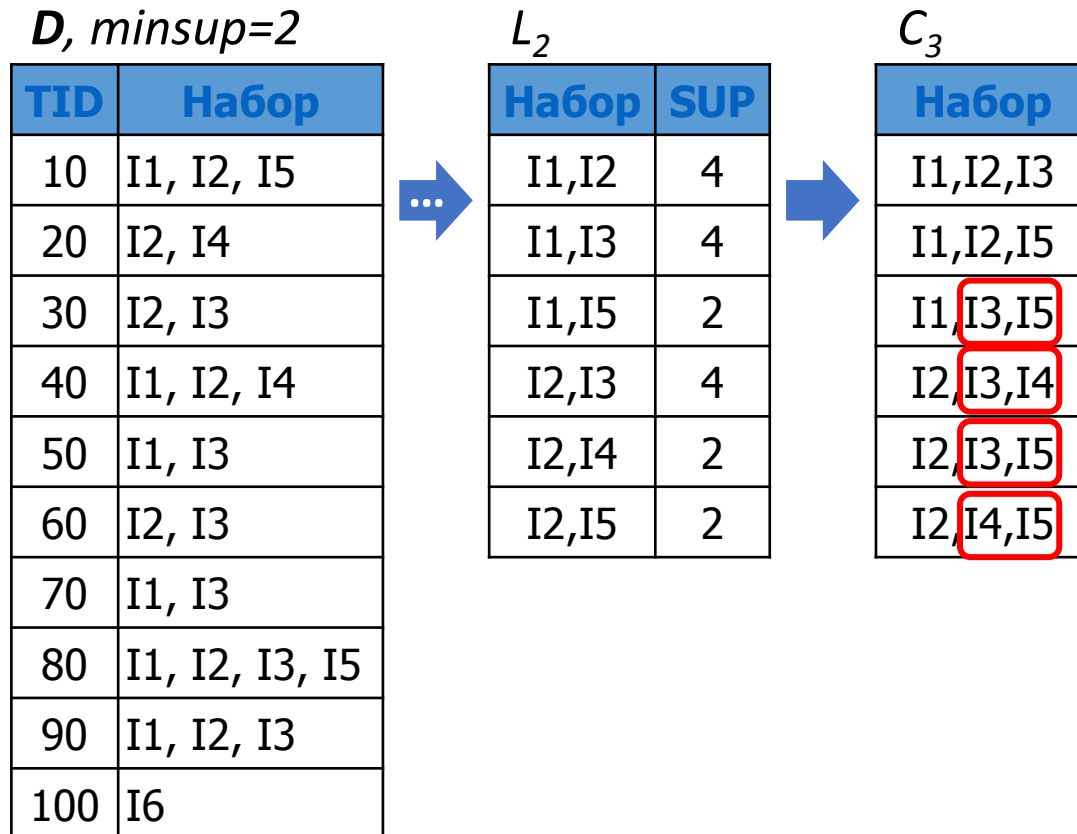


L_2

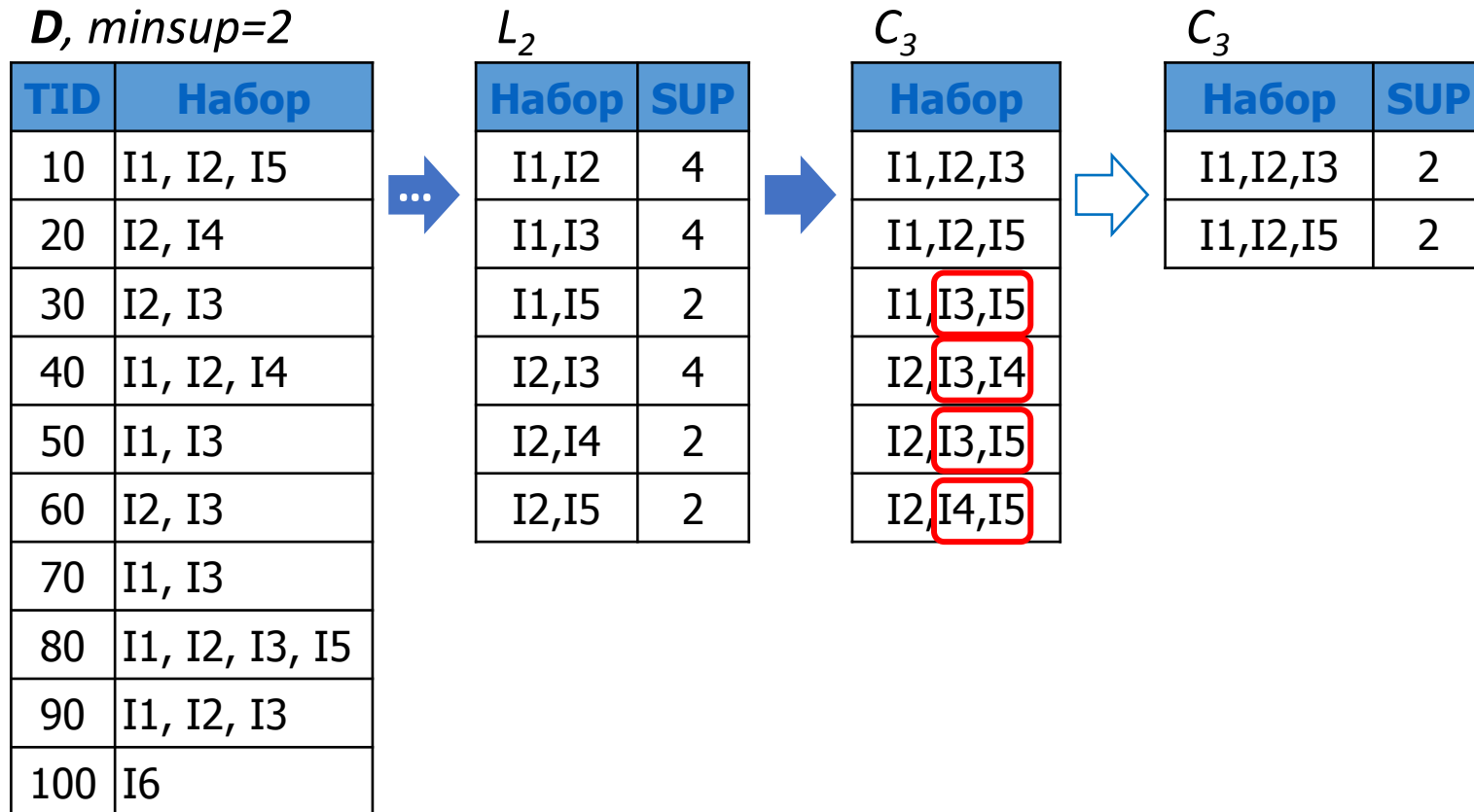
Набор	SUP
I1,I2	4
I1,I3	4
I1,I5	2
I2,I3	4
I2,I4	2
I2,I5	2



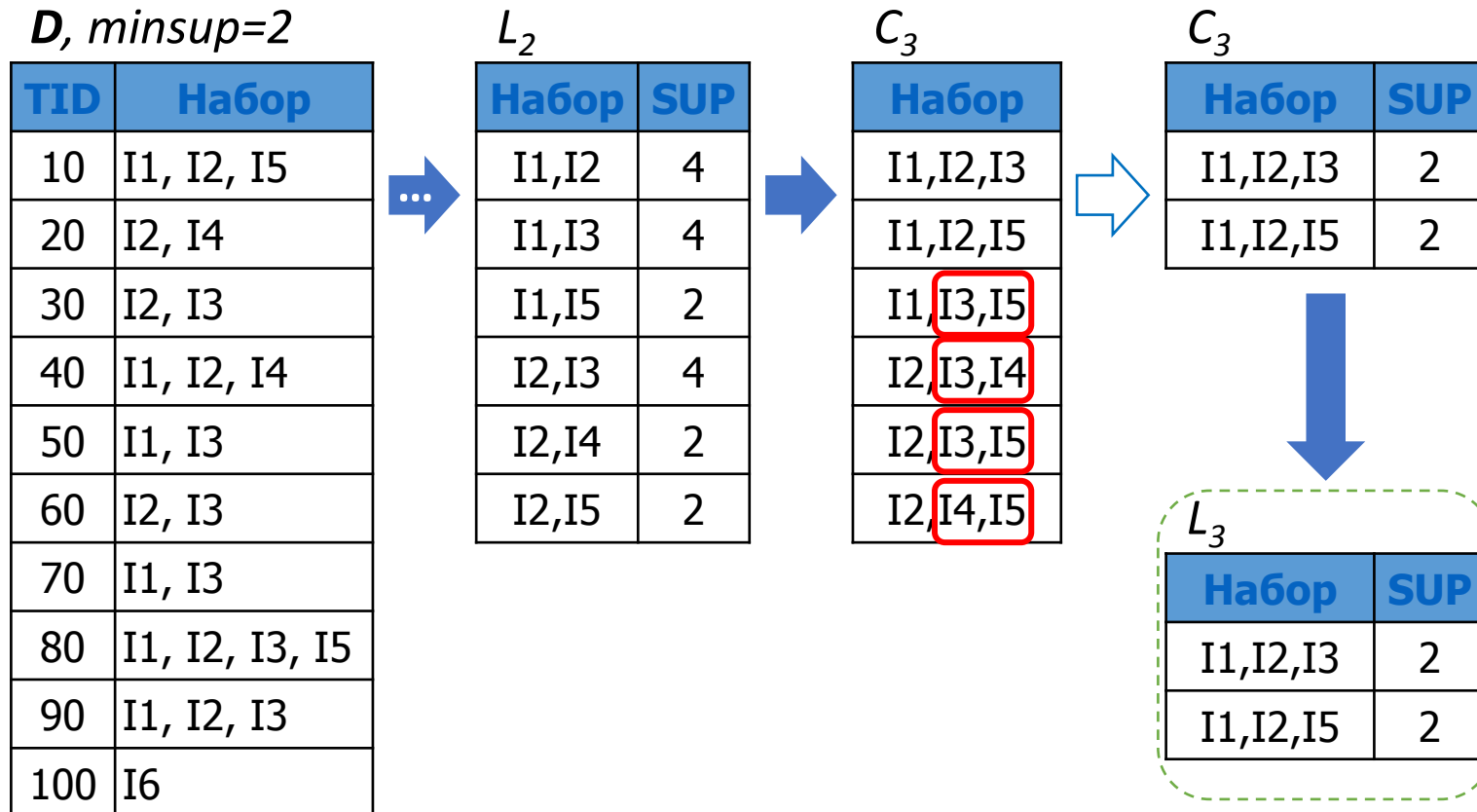
Алгоритм Apriori: пример



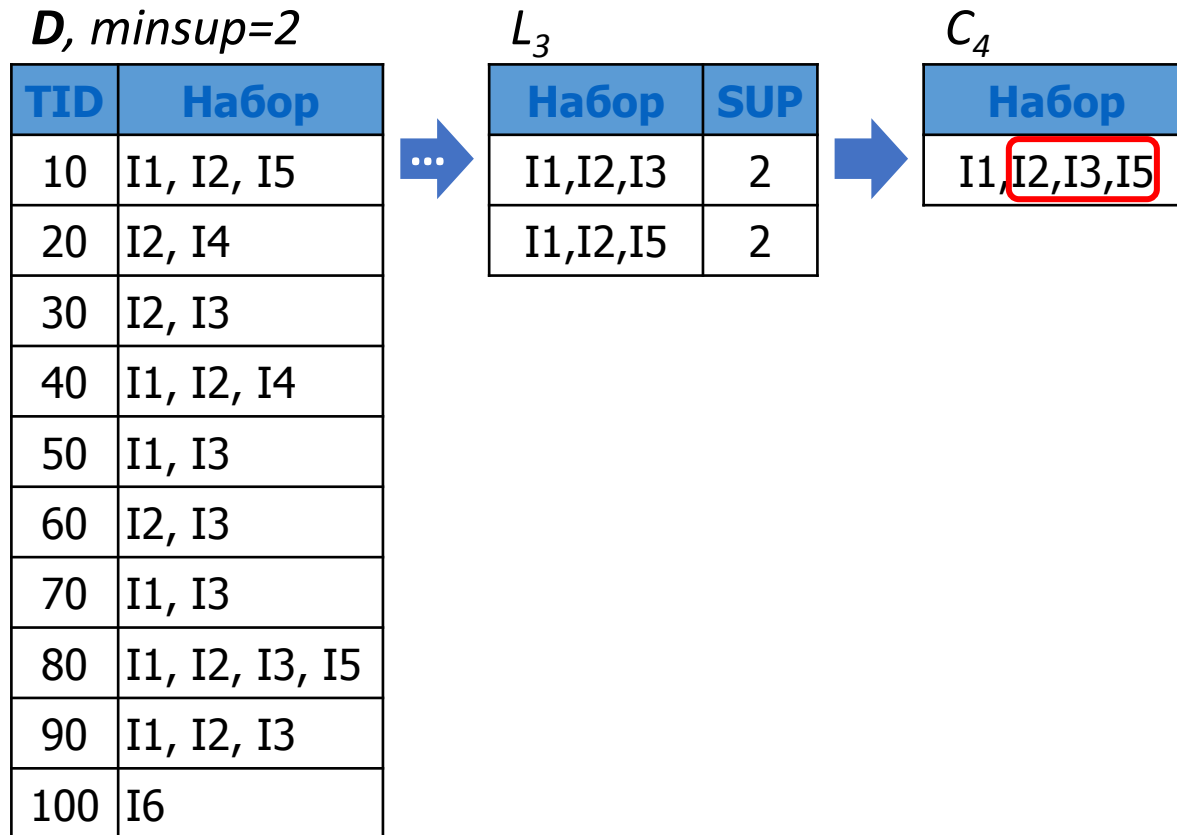
Алгоритм Apriori: пример



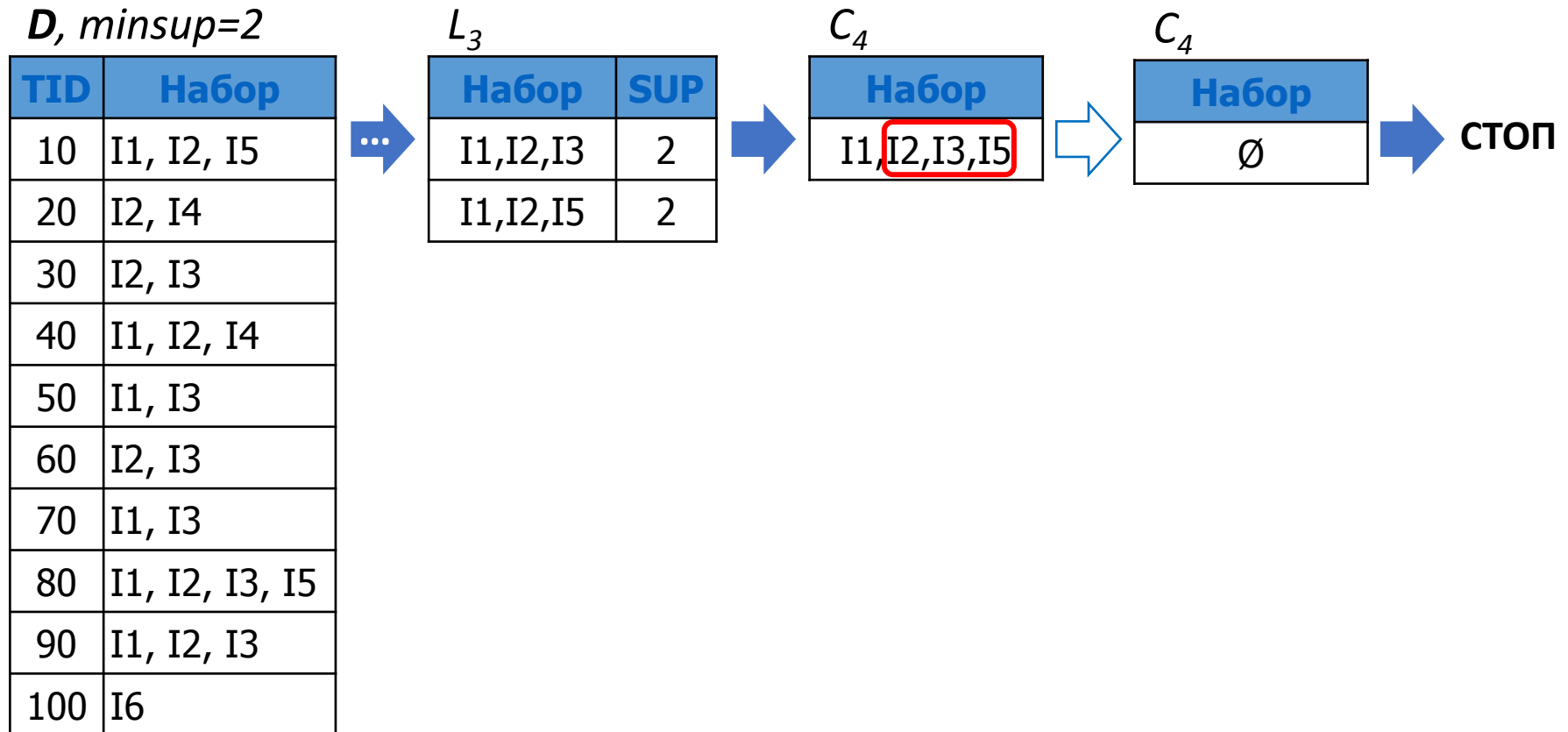
Алгоритм Apriori: пример



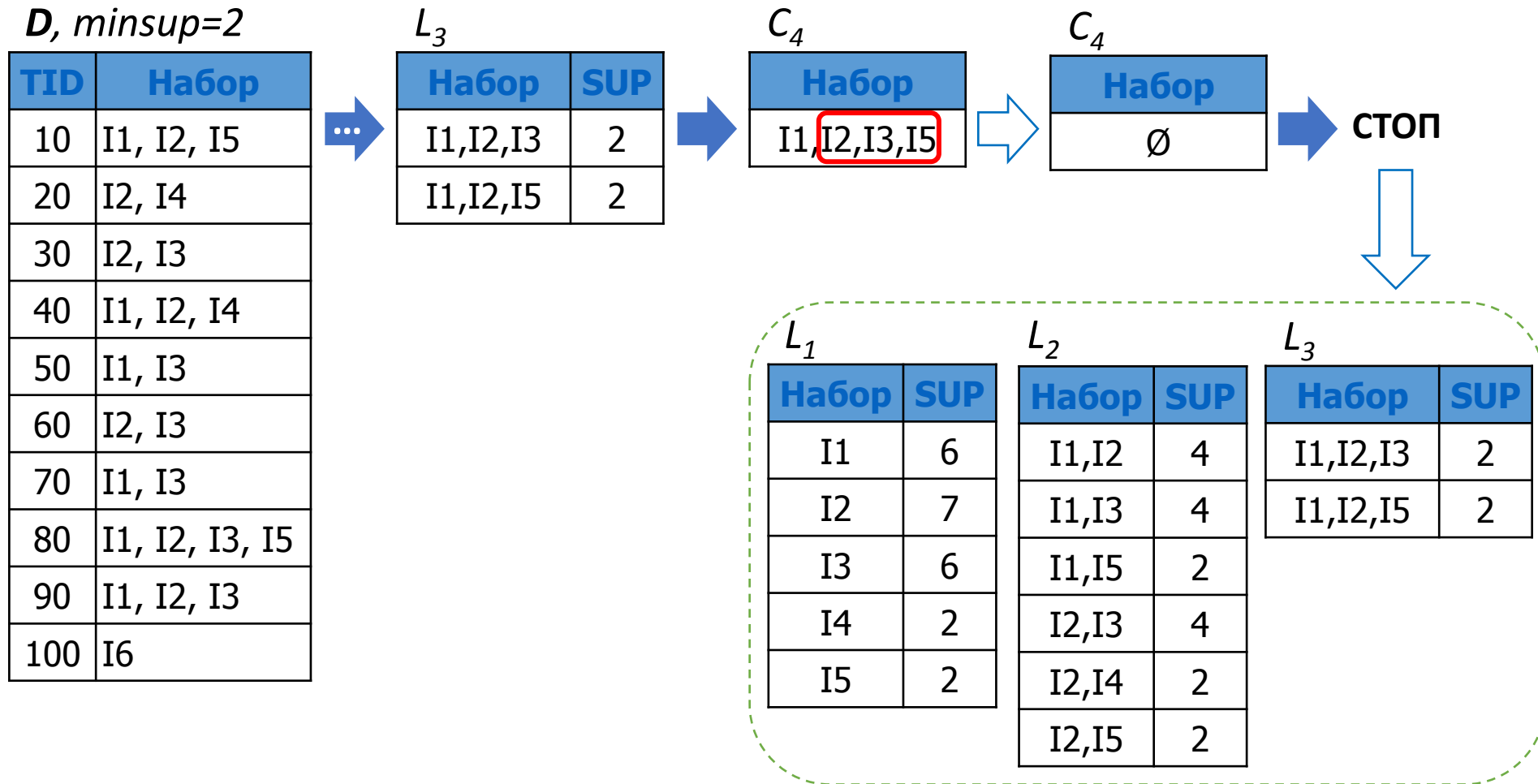
Алгоритм Apriori: пример



Алгоритм Apriori: пример



Алгоритм Apriori: пример



Алгоритм Apriori: ПСЕВДОКОД

Input:

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

Output: L , frequent itemsets in D .

Method:

```
(1)   $L_1 = \text{find\_frequent\_1-itemsets}(D);$ 
(2)  for ( $k = 2; L_{k-1} \neq \phi; k++$ ) {
(3)     $C_k = \text{apriori\_gen}(L_{k-1});$ 
(4)    for each transaction  $t \in D$  { // scan  $D$  for counts
(5)       $C_t = \text{subset}(C_k, t);$  // get the subsets of  $t$  that are candidates
(6)      for each candidate  $c \in C_t$ 
(7)         $c.\text{count}++;$ 
(8)    }
(9)     $L_k = \{c \in C_k \mid c.\text{count} \geq min\_sup\}$ 
(10) }
(11) return  $L = \cup_k L_k;$ 
```

Алгоритм Apriori: ПСЕВДОКОД

procedure **apriori_gen**(L_{k-1} :frequent $(k-1)$ -itemsets)

```
(1)   for each itemset  $l_1 \in L_{k-1}$ 
(2)     for each itemset  $l_2 \in L_{k-1}$ 
(3)       if ( $l_1[1] = l_2[1] \wedge l_1[2] = l_2[2] \wedge \dots \wedge l_1[k-2] = l_2[k-2] \wedge l_1[k-1] < l_2[k-1]$ ) then {
(4)          $c = l_1 \bowtie l_2$ ; // 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  $s \notin L_{k-1}$  then
(3)       return TRUE;
(4)   return FALSE;
```

Input:

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

Output: L , frequent itemsets in D .

Method:

```
(1)    $L_1 = \text{find\_frequent\_1-itemsets}(D)$ ;
(2)   for ( $k = 2; L_{k-1} \neq \emptyset; k++$ ) {
(3)      $C_k = \text{apriori\_gen}(L_{k-1})$ ;
(4)     for each transaction  $t \in D$  { // scan  $D$  for counts
(5)        $C_t = \text{subset}(C_k, t)$ ; // get the subsets of  $t$  that are candidates
(6)       for each candidate  $c \in C_t$ 
(7)          $c.\text{count}++$ ;
(8)     }
(9)      $L_k = \{c \in C_k \mid c.\text{count} \geq min\_sup\}$ 
(10)  }
(11) return  $L = \cup_k L_k$ ;
```

Ассоциативные правила

- Ассоциативное правило (*association rule*) для наборов A, B ($A \cap B = \emptyset$) – это импликация $A \rightarrow B$.

- Поддержка правила (*support*)

- $\text{sup}(A \rightarrow B) = P(A \cup B)$

- Достоверность правила (*confidence*)

- $\text{conf}(A \rightarrow B) = P(B|A) = \text{sup}(A \cup B) / \text{sup}(A)$

- Пример:

D

TID	Набор
10	орешки, пиво, подгузники
20	кофе, пиво, подгузники
30	кофе, пиво, подгузники, хлеб
40	молоко, орешки, хлеб
50	кофе, молоко, орешки, подгузники, хлеб

- $\text{sup}(\text{пиво} \rightarrow \text{подгузники}) = 3/5$
- $\text{conf}(\text{пиво} \rightarrow \text{подгузники}) = 1$
- $\text{sup}(\text{подгузники} \rightarrow \text{пиво}) = 3/5$
- $\text{conf}(\text{подгузники} \rightarrow \text{пиво}) = 3/4$

Поиск ассоциативных правил

- Минимальная поддержка (*min support*) и достоверность (*min confidence*) – пороговые значения поддержки и достоверности ассоциативных правил.
- Устойчивое правило (*strong rule*) имеет поддержку и достоверность не ниже минимальных.
- Задача 2: поиск ассоциативных правил
 - для заданных D , $minsup$ и $minconf$ найти все устойчивые правила.

Поиск устойчивых правил

1. Найти частые наборы (решить задачу рыночной корзины)
2. Для каждого частого набора l генерировать все его непустые подмножества
3. Для каждого непустого подмножества s
 - если $\text{conf}(s) = \sup(l) / \sup(s) \geq \text{minconf}$, то
выдать правило $s \rightarrow (l-s)$

Поиск устойчивых правил: пример

$D, \text{minsup}=2$

TID	Набор
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



L_1		L_2		L_3	
Набор	SUP	Набор	SUP	Набор	SUP
I1	6	I1,I2	4	I1,I2,I3	2
I2	7	I1,I3	4	I1,I2,I5	2
I3	5	I1,I5	2		
I4	2	I2,I3	4		
I5	2	I2,I4	2		
		I2,I5	2		



Правила для I1,I2,I5

Правило	CONF
$\{I1,I2\} \rightarrow I5$	2/4
$\{I1,I5\} \rightarrow I2$	2/2
$\{I2,I5\} \rightarrow I1$	2/2
$I1 \rightarrow \{I2,I5\}$	2/6
$I2 \rightarrow \{I1,I5\}$	2/7
$I5 \rightarrow \{I1,I2\}$	2/2



Устойчивые правила для $\{I1,I2,I5\}$ при $\text{minconf}=0.5$

Правило	CONF
$\{I1,I2\} \rightarrow I5$	2/4
$\{I1,I5\} \rightarrow I2$	2/2
$\{I2,I5\} \rightarrow I1$	2/2
$I5 \rightarrow \{I1,I2\}$	2/2

Устойчивое правило не всегда полезно

- Пример:
 - $|D|=10000$, $\text{minsup}=0.3$, $\text{minconf}=0.6$
 - $P(\text{игра})=6000$, $P(\text{видео})=7500$, $P(\text{игра} \cup \text{видео})=4000$
 - $\text{игра} \rightarrow \text{видео}$ [$\text{sup}=4000/10000=0.4$, $\text{conf}=4000/6000=0.66$]
 - Вероятность покупки видео 0.75 (его покупают часто независимо от покупки игры), и проще угадать наличие видео в покупке, чем предсказать этот факт с помощью правила.
- При $\text{conf}(A \rightarrow B) < \text{sup}(B)$ правило $A \rightarrow B$ бесполезно.

Оценка полезности правил

- $\text{Interest}(A \rightarrow B) = |\text{conf}(A \rightarrow B) - P(B)|$
 - Если $\text{Interest} \geq \text{threshold}$, правило полезно (обычно берется $\text{threshold} = 0.5$).
- $\text{Conviction}(A \rightarrow B) = (1 - \text{sup}(B)) / (1 - \text{conf}(A \rightarrow B))$
 - Если $\text{Conviction} < 1$, правило полезно.
- Дополнение правила $A \rightarrow B$ мерой корреляции A и B
 - *lift*
 - χ^2 (*Chi-square*)
 - *all_conf*
 - *max_conf*
 - *Kulc* (Kulczynski, 1927)
 - *Cosine*
 - *IR* (*Imbalance Ratio*)

Корреляция, мера *lift*

$$\circ \text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{P(B|A)}{P(B)} = \frac{\text{conf}(A \rightarrow B)}{\text{sup}(B)} = \text{lift}(A \rightarrow B)$$

○ Оценка

- $\text{lift} < 1$: A и B имеют отрицательную корреляцию (наличие A подразумевает отсутствие B)
- $\text{lift} > 1$: A и B имеют положительную корреляцию (наличие A подразумевает наличие B)
- $\text{lift} = 1$: A и B независимы

○ Пример:

- $P(\text{игра}) = 0.60$, $P(\text{видео}) = 0.75$, $P(\text{игра, видео}) = 0.40$
- $\text{lift}(\text{игра} \rightarrow \text{видео}) = 0.40 / (0.60 * 0.75) = 0.89$, т.е. игра и видео имеют отрицательную корреляцию

Алгоритм ECLAT

$D, \text{minsup}=2$

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

○ Вертикальный формат данных

- $\text{sup}(A) = |\text{TID_list}_A|$
- $\text{TID_list}_{A,B} = \text{TID_list}_A \cap \text{TID_list}_B$

Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор	Набор	TID_list	SUP
10	I1, I2, I5	I1	10, 40, 50, 70, 80, 90	6
20	I2, I4	I2	10, 20, 30, 40, 60, 80, 90	7
30	I2, I3	I3	30, 50, 60, 70, 80, 90	6
40	I1, I2, I4	I4	20, 40	2
50	I1, I3	I5	10, 80	2
60	I2, I3	I6	100	1
70	I1, I3			
80	I1, I2, I3, I5			
90	I1, I2, I3			
100	I6			



Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2
I6	100	1



L_1

Набор	SUP
I1	6
I2	7
I3	5
I4	2
I5	2

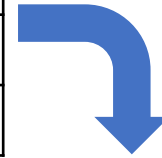
Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1

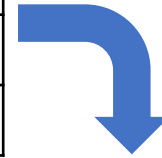
Алгоритм ECLAT

D , $minsup=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1



L_2

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

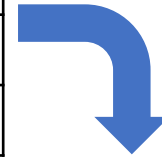
Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1



Набор
I1,I2,I3
I1,I2,I5
I1,I3,I5
I2,I3,I4
I2,I3,I5
I2,I4,I5

Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1



Набор
I1,I2,I3
I1,I2,I5
I1,I3,I5
I2,I3,I4
I2,I3,I5
I2,I4,I5

L_2

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

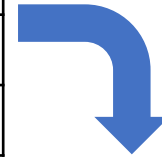
Алгоритм ECLAT

D , $minsup=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1,I2,I3	80, 90	2
I1,I2,I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1

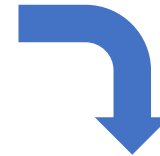
Алгоритм ECLAT

$D, \text{minsup}=2$

TID	Набор
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



Набор	TID_list	SUP
I1	10, 40, 50, 70, 80, 90	6
I2	10, 20, 30, 40, 60, 80, 90	7
I3	30, 50, 60, 70, 80, 90	6
I4	20, 40	2
I5	10, 80	2



Набор	TID_list	SUP
I1, I2	10, 40, 80, 90	4
I1, I3	50, 70, 80, 90	4
I1, I4	40	1
I1, I5	10, 80	2
I2, I3	30, 60, 80, 90	4
I2, I4	20, 40	2
I2, I5	10, 80	2
I3, I5	80	1



Набор	TID_list	SUP
I1,I2,I3	80, 90	2
I1,I2,I5	10, 80	2



L_3

Набор	SUP
I1,I2,I3	2
I1,I2,I5	2

Алгоритм ECLAT

- Плюс
 - Отсутствие необходимости просматривать все транзакции для подсчета поддержки
- Минус
 - При больших значениях $|TID_list|$ трудно хранить и вычислять пересечение TID_list
- Улучшение
 - Хранение разницы между $(k+1)$ -набором и k -набором. Пример:
 $TID_list(\{I1\}) = \{10, 40, 50, 70, 80, 90\}$
 $TID_list(\{I1, I2\}) = \{10, 40, 80, 90\}$
 $diffset(\{I1\}, \{I1, I2\}) = \{50, 70\}$