

АСОЦІАТИВНІ ПРАВИЛА DATA MINING

АЛГОРИТМИ APRIORI, ECLAT

ЗАДАЧА АНАЛІЗУ РИНКОВИХ КОШИКІВ

Надія І. Недашківська n.nedashkivska@gmail.com

Постановка задачі аналізу ринкових кошиків



В задачі пошуку **асоціативних правил** мова піде про пошук логічних закономірностей в масивах даних.

Традиційне застосування асоціативних правил – це **аналіз ринкових кошиків** - виявлення наборів товарів, які часто купуються разом або ніколи не купуються разом.

Постановка задачі аналізу ринкових кошиків

Дано: $I = \{i_1, i_2, \dots, i_n\}$ - множина об'єктів

$T = \{i_k \mid i_k \in I\} \subset I$ - транзакція – множина подій, які відбуваються разом

$D = \{T_1, T_2, \dots, T_m\}$ - множина транзакцій

Потрібно: - Знайти набори об'єктів, які зустрічаються в I найчастіше – часті набори.

- Побудувати множину асоціативних правил.

Асоціативне правило – це висловлювання виду

якщо X тоді Y : $X \Rightarrow Y$, де $X \subset I$, $Y \subset I$, $X \cap Y = \emptyset$

Приклад: магазин

Ідентифікатор товару	Назва товару	Ціна товару
0	Сир Голандський	367.00
1	Оселедець	125.00
2	Огірки солоні	76.00
3	Банани	44.00
4	Йогурт	18.50
5	Вода Моршинська (2 л)	37.00
6	Кава Якобс (200 г)	248.00

Приклад: магазин: множина транзакцій

Ідентифікатор транзакції	Ідентифікатор товару	Назва товару	Ціна товару
0	1	Оселедець	125.00
0	2	Огірки солоні	76.00
0	5	Вода Моршин (2 л)	37.00
1	1	Сир Голандський	367.00
1	3	Банани	44.00
1	4	Йогурт	18.50
1	5	Вода Моршин (2 л)	37.00
1	6	Кава Якобс (200 г)	248.00

Приклад АП: якщо {оселедець, огірки} тоді {вода}

Постановка задачі

- Множина транзакцій, які містять об'єкт i_k :

$$D_{i_k} = \{T_r \mid i_k \in T_r, k = 1, \dots, n, r = 1, \dots, m\} \subseteq D$$

- $F \subseteq I$ - довільний набір об'єктів

- q -елементний набір - набір, який складається з q об'єктів.

- Множина транзакцій, які містять набір F :

$$D_F = \{T_r \mid F \subseteq T_r, r = 1, \dots, m\} \subseteq D$$

Постановка задачі

Підтримкою набору F називається $Supp(F) = \frac{|D_F|}{|D|}$

Властивість антимонотонності:

$\forall E, F \subseteq I$ з умови $E \subseteq F$ випливає $Supp(E) \geq Supp(F)$

Набір F називається **частим**, якщо $Supp(F) \geq Supp_{\min}$

Параметр $Supp_{\min}$ - **мінімальна підтримка набору**

$L = \{F \mid Supp(F) \geq Supp_{\min}\} - ?$

Показники корисності асоціативних правил

□ Підтримка АП $Supp(X \Rightarrow Y) = p(X \cap Y) = \frac{|D_{F=X \cup Y}|}{|D|}$

$|D_{F=X \cup Y}|$ в чисельнику – це кількість транзакцій, які містять X і Y

□ Достовірність (значущість) АП

$$Conf(X \Rightarrow Y) = p(Y | X) = \frac{p(X \cap Y)}{p(X)} = \frac{Supp(X \Rightarrow Y)}{Supp(X)}$$

Більше знач $Conf$ свідчить про більшу корисність правила. Проте, це не жорстка імплікація, тобто не завжди із X слідує Y , але часто. Наскільки часто – це нам показує параметр $Conf_{min}$. Наприклад, в половині випадків. Цього часто буває достатньо.

$Supp(Y)$ – безумовна підтримка, $Conf(X \rightarrow Y)$ – умовна підтримка набору Y .

Постановка задачі пошуку асоціативних правил

Знайти АП, такі що

1) Набори X і Y часто зустрічаються разом:

$$Supp(X \Rightarrow Y) \geq Supp_{\min}$$

2) Якщо зустрічається X , то часто зустрічається також і Y :

$$Conf(X \Rightarrow Y) \geq Conf_{\min}$$

Параметр $Supp_{\min}$ - мінімальна підтримка

Параметр $Conf_{\min}$ - мінімальна достовірність

Показники корисності асоціативних правил

Якщо X і Y незалежні, то $Supp(X \Rightarrow Y) \approx Supp(X) \cdot Supp(Y)$

Тоді $Conf(X \Rightarrow Y) \approx Supp(Y)$

Хоча умова і наслідок часто зустрічаються разом,
не менш часто вони зустрічаються і окремо.

Тому використовується ще один показник корисності АП:

□ **Покращення (improvement) або ліфт (lift)**

$$Improv(X \Rightarrow Y) = \frac{Supp(X \Rightarrow Y)}{Supp(X)Supp(Y)}$$

$Improv(X \Rightarrow Y) > 1$ - правило корисніше за вгадування

Вправи

- ❑ Знайти значення підтримки для всіх наборів прикладу.
- ❑ Чи будуть правила, побудовані на основі одного і того ж набору, мати однакову підтримку/ однакову достовірність?
- ❑ Чи можна поєднувати наступні правила

$$X = \{i_1, i_2\} \Rightarrow Y = \{i_3\}$$

$$X = \{i_1, i_2\} \Rightarrow Y = \{i_4\}$$

в одне правило $X = \{i_1, i_2\} \Rightarrow Y = \{i_3, i_4\}$

АЛГОРИТМ APRIORI

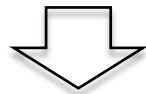
ПОБУДОВИ АСОЦІАТИВНИХ ПРАВИЛ

Надія І. Недашківська n.nedashkivska@gmail.com

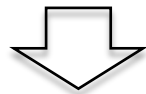
Найпростіший алгоритм пошуку АП

- ❑ Розглянути всі можливі комбінації умов і наслідків
- ❑ Розрахувати для них підтримку (достовірність, покращення)
- ❑ Відібрати ті, що задовольняють пороговим обмеженням

Якщо в БД k предметів і АП є бінарними



$k \cdot 2^{k-1}$ АП, наприклад, $1000 \cdot 2^{999}$



дуже високі обчислювальні витрати

Загальний метод пошуку АП

Пошук асоціативних правил:

- 1) знаходження частих наборів об'єктів
- 2) побудова асоціативних правил на основі частих наборів

Властивість антимонотонності:

$\forall E, F \subseteq I$ з умови $E \subseteq F$ випливає $Supp(E) \geq Supp(F)$

- якщо F частий, то всі його підмножини також часті,
- якщо F не частий, то всі набори $E \supset F$ також не часті,
- $\forall E, F \subseteq I$ має місце $Supp(E \cup F) \leq Supp(F)$

Загальний метод пошуку АП

Ідея: аналізувати не всі, а тільки часті набори.

Властивість антимонотонності значно зменшує простір пошуку АП. Якщо набір F не є частим, то додавання нового об'єкту E до набору F не робить його більш частим.

З іншого боку, якщо ми додаємо нові об'єкти до набору, і на якомусь кроці отримали нечастий набір, то все, більше об'єктів до цього набору додавати не треба.

1) Задача пошуку частих наборів об'єктів вважається складною задачею: це потребує багаторазового перегляду бази даних транзакцій і часто неможливо для надвеликих баз, напр. 1 млн. записів. Таку БД складно повністю помістити в оперативну пам'ять. Ми будемо намагатися прочитати БД тільки один раз !

2) Побудова АП на основі знайдених частих наборів вважається простою ефективною процедурою в оперативній пам'яті.

Знаходження частих наборів об'єктів.

Алгоритм Apriori

□ L_k - множина k -елементних **частих** наборів:

$$L_k = \{F_j \mid F_j = \{i_1, i_2, \dots, i_k\}, \text{Supp}(F_j) \geq \text{Supp}_{\min}\}$$

□ C_k - множина кандидатів в k -елементні часті набори, отримана шляхом зв'язування множини L_{k-1} з собою

k -набори є **зв'язуваними**, якщо вони мають спільні перші $k-1$ елементів.

Етапи алгоритму Apriori:

- 1) Генерація кандидатів в часті набори
- 2) Скорочення множини кандидатів, використовуючи властивість антимонотонності
- 3) Відбір частих кандидатів на основі параметру Supp_{\min}

Знаходження частих наборів об'єктів.

Алгоритм Apriori

1. Побудова множини одноелементних частих наборів:

$$L_1 = \{i \mid i \in I, \text{Supp}(i) \geq \text{Supp}_{\min}\}$$

2. Для всіх $k = 2, \dots, n$

Генерація k -елементних частих наборів

$$L_k = \{F \cup \{i\} \mid F \in L_{k-1}, i \in L_1 \setminus F, \text{Supp}(F \cup \{i\}) \geq \text{Supp}_{\min}\}$$

3. Якщо $L_k = \emptyset$ то вихід із циклу по k .

Знаходження частих наборів об'єктів.

Алгоритм Apriori

Ми перебираємо всі часті підмножини деякої множини.

Спочатку беремо **одноеlementні часті набори** L_1 . Далі додаємо до них по одному об'єкту, тобто формуємо двоелементні множини і т.д. Нечасті набори ми відкидаємо і так поступово нарощуємо потужність наборів, фільтруючи від нечастих наборів. **Об'єм перебору суттєво залежить від** параметру **Supp_{min}**: якщо значення Supp_{min} зменшуємо, то кількість частих наборів буде більшою, якщо Supp_{min} збільшуємо – то меншою.

Це дуже критичний параметр щодо кількості знайдених частих наборів.

k – це потужність наборів, які аналізуються на одній ітерації циклу.

Знаходження частих наборів об'єктів.

Алгоритм Apriori

На кроці 2 шукаємо часті набори потужності k , за умови, що часті набори потужності $(k-1)$ нам відомі. Тобто, **Apriori** – це рекурсивний метод. Беремо по одному елементу F з $(k-1)$ -елементного частого набору L_{k-1} , та по одному об'єкту i з множини L_1 без F та намагаємося їх об'єднати. Якщо підтримка цього об'єднання $\geq \text{Supp}_{\min}$, то записуємо це об'єднання в L_k .

На кроці 3, якщо не знайшли жодного частого набору потужності k , то виходимо з циклу по k , бо вже жодного частого набору в подальшому не отримаємо.

Приклад: магазин

$$Supp_{\min} = 0.5$$

ідентифікатор	Набір L_1	Supp
0	{1}	0.50
1	{2}	0.50
2	{3}	0.50
3	{5}	0.75
4	{6}	0.50

Приклад: магазин

$$L_1 = \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}\}$$

$$L_2 = \{\{1,2\}, \{1,5\}, \{2,5\}, \{3,6\}\}$$

ідентифікатор	Кандидати в часті набори	Supp
0	{1,2}	0.50
1	{1,3}	< Supp _{min}
2	{1,5}	0.50
3	{1,6}	< Supp _{min}
4	{2,3}	< Supp _{min}
5	{2,5}	0.50
6	{2,6}	< Supp _{min}
7	{3,5}	< Supp _{min}
8	{3,6}	0.50
9	{5,6}	< Supp _{min}

Приклад: магазин

$$L_2 = \{\{1,2\}, \{1,5\}, \{2,5\}, \{3,6\}\}$$

$$L_3 = \{\{1,2,5\}\}$$

ідентифікатор	Кандидати в часті набори	Supp
0	$\{1,2,5\}$	0.50

$$L = L_1 \cup L_2 \cup L_3 = \{\{1\}, \{2\}, \{3\}, \{5\}, \{6\}, \{1,2\}, \{1,5\}, \{2,5\}, \{3,6\}, \{1,2,5\}\}$$

Побудова асоціативних правил: етап 2

Дано: $\{L_k \mid k = 1, 2, \dots, n\}$ – k -елементні часті набори

Знайти: R – список асоціативних правил

$R := \emptyset$

Для всіх $k = 2, \dots, n$, для всіх $F \in L_k$ $Y := \emptyset$

Процедура BuildAssocRules(R, F, Y)

1. Для всіх $f \in F$ - можливих піднаборів f набору F

1.1. $X := F \setminus \{f\}$ $Y := Y \cup \{f\}$ розщепляємо частий набір F на умову X і наслідок Y

1.2. Якщо $Conf(X \Rightarrow Y) \geq Conf_{\min}$ то

1.2.1. Додати асоціативне правило $X \Rightarrow Y$ в R

1.2.2. Якщо $|X| > 1$ то BuildAssocRules(R, X, Y)

Побудова асоціативних правил: етап 2

Пошук асоціативних правил:

- 1) знаходження частих наборів об'єктів
- 2) побудова асоціативних правил на основі частих наборів

Нехай часті набори знайдено на етапі 1. Тоді побудова АП на етапі 2 вважається простою і ефективною процедурою в оперативній пам'яті.

Ідея побудови АП: ми розщепляємо частий набір F на умову і наслідок, перевіряючи достовірність $Conf$ отриманого правила.

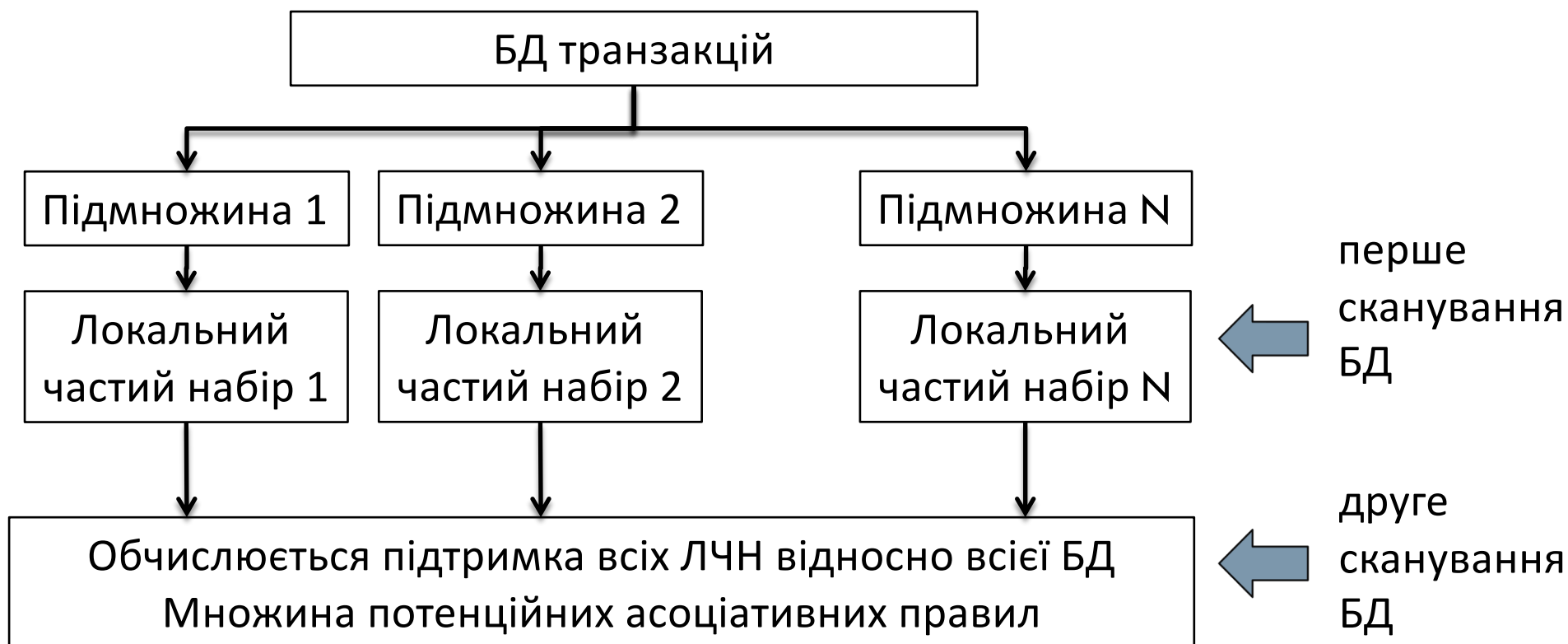
Процедура є рекурсивною. Коли вона викликається перший раз, то список правил R порожній, F – деякий непорожній частий набір, Y – порожня.

При знаходженні частих наборів (етап 1) використовуємо тільки параметр $Supp_{min}$. При побудові АП (етап 2) використовується ще й параметр $Conf_{min}$.

Підвищення ефективності алгоритму Apriori

Модифікація Partition-based Apriori algorithm

розбиття БД транзакцій на N достатньо малих підмножин



Підвищення ефективності алгоритму Apriori

Один з напрямків підвищення ефективності – скорочення необхідної кількості сканувань БД транзакцій. Алгоритм Apriori сканує БД декілька разів, залежно від числа елементів в частих наборах.

Алгоритм розділення Partition-based Apriori algorithm потребує всього два проходи по набору даних. Він базується на ідеї локальних частих наборів.

1. Весь набір даних розділяється на N підмножин, які не перетинаються. Кожна з підмножин достатньо мала, так щоб помістилася в оперативній пам'яті ПК. Під час першого сканування алгоритм считує кожен підмножину і отримує набори, які є частинами для даної підмножини (локальні часті набори).
2. Під час другого сканування алгоритм обчислює підтримку всіх локальних частих наборів відносно всієї БД. Друге сканування визначає множину всіх потенційних АП.

Підвищення ефективності алгоритму Apriori

- ❑ **Семплінг** – відбір випадкової вибірки з початкової БД і пошук частих наборів за цією вибіркою.

Шукається компроміс між точністю та ефективністю розв'язку. Розмір вибірки має бути таким щоб забезпечити прийнятні обчислювальні витрати. Ясно, що деякі часті набори будуть втрачені. Щоб звести втрати до мінімуму поріг підтримки Supp_{\min} вибирають нижчим ніж для всієї вибірки.

- ❑ Модифікації на основі **ієрархії об'єктів**.

- ❑ **Аналіз послідовностей об'єктів**.

АЛГОРИТМ ECLAT

Надія І. Недашківська n.nedashkivska@gmail.com

Знаходження частих наборів об'єктів.

Алгоритм Eclat

Метод пошуку у глибину (DFS Search Manner) за допомогою перетинів наборів елементів. Eclat є удосконаленим алгоритмом Apriori. Використовує **вертикальне відображення** даних.

Горизонтальний формат відображення даних

Номер транзакції	Товари
1	Хліб, масло, сік
2	Масло, вода
3	Масло, молоко
4	Хліб, масло, вода
5	Хліб, молоко
6	Масло, молоко
7	Хліб, молоко
8	Хліб, масло, молоко, сік
9	Хліб, масло, молоко

Вертикальний формат (TID-множини)

Товар	Номери транзакцій
Хліб	1,4,5,7,8,9
Масло	1,2,3,4,6,8,9
Молоко	3,5,6,7,8,9
Вода	2,4
Сік	1,8

При такому представленні підтримка набору дорівнює відношенню потужності множини до загальної кількості транзакцій.

Знаходження частих наборів об'єктів.

Алгоритм Eclat

1. Переводимо дані з горизонтального до вертикального формату

$$S_1 = \{ \{i, M_i\} \mid i \in I, M_i = D_i = \{T_r \mid i \in T_r, r = 1, \dots, m\} \}$$

2. Розраховуємо множину одноелементних частих наборів

$$L_1 = \{i \mid i \in I, \text{Supp}(i) \geq \text{Supp}_{\min}\}$$

Для всіх $k=2\dots n$

3. Генерація кандидатів в k -елементні часті набори:

Для всіх $F_a, F_b \in L_{k-1}$

$$S_k = \{ \{F', M'_F\} \mid F' = F_a \cup F_b, M'_F = \{M_{F_a} \cap M_{F_b}\}, a < b, M_{F_a}, M_{F_b} \in S_{k-1} \}$$

Відбір частих наборів: $L_k = \{F' \in S_k, \text{Supp}(F') \geq \text{Supp}_{\min}\}$

4. Якщо $L_k = \emptyset$, то вихід із циклу по k

Знаходження частих наборів об'єктів.

Алгоритм Eclat



Етапи алгоритму Eclat аналогічні відповідним етапам алгоритму Apriori, крім функції обчислення підтримки кандидатів.

Розрахунок значень підтримки наборів тепер не потребує сканування бази.

Знаходження частих наборів об'єктів.

Алгоритм Eclat

Товар	Набори транзакцій
{Хліб}	1,4,5,7,8,9
{Масло}	1,2,3,4,6,8,9
{Молоко}	3,5,6,7,8,9
{Вода}	2,4
{Сік}	1,8

Товар	Набори транзакцій
{Хліб, Масло}	1,4,8,9
{Хліб, Молоко}	5,7,8,9
{Хліб, Вода}	4
{Хліб, Сік}	1,8
{Масло, Молоко}	3,6,8,9
{Масло, Вода}	2,4
{Масло, Сік}	1,8
{Молоко, Сік}	8

Товар	Набори транзакцій
{Хліб, Масло, Молоко}	8,9
{Хліб, Масло, Сік}	1,8
{Хліб, Масло, Вода}	4
{Хліб, Молоко, Сік}	8
{Масло, Молоко, Вода}	-
{Масло, Молоко, Сік}	8
{Масло, Вода, Сік}	-

Товар	Набори транзакцій
{Хліб, Масло, Молоко, Сік}	8

Знаходження частих наборів об'єктів.

Алгоритм Eclat

Переваги Eclat

1. Зазвичай алгоритм значно швидший за Apriori.
2. Не потрібно проводити кожного разу сканування бази даних, щоб знайти рівень підтримки для $(k+1)$ наборів елементів, при $k \geq 1$. Це суттєво зменшує ресурсні затрати та є найважливішою перевагою цього алгоритму.
3. БД сканується лише один раз.

Недоліки Eclat

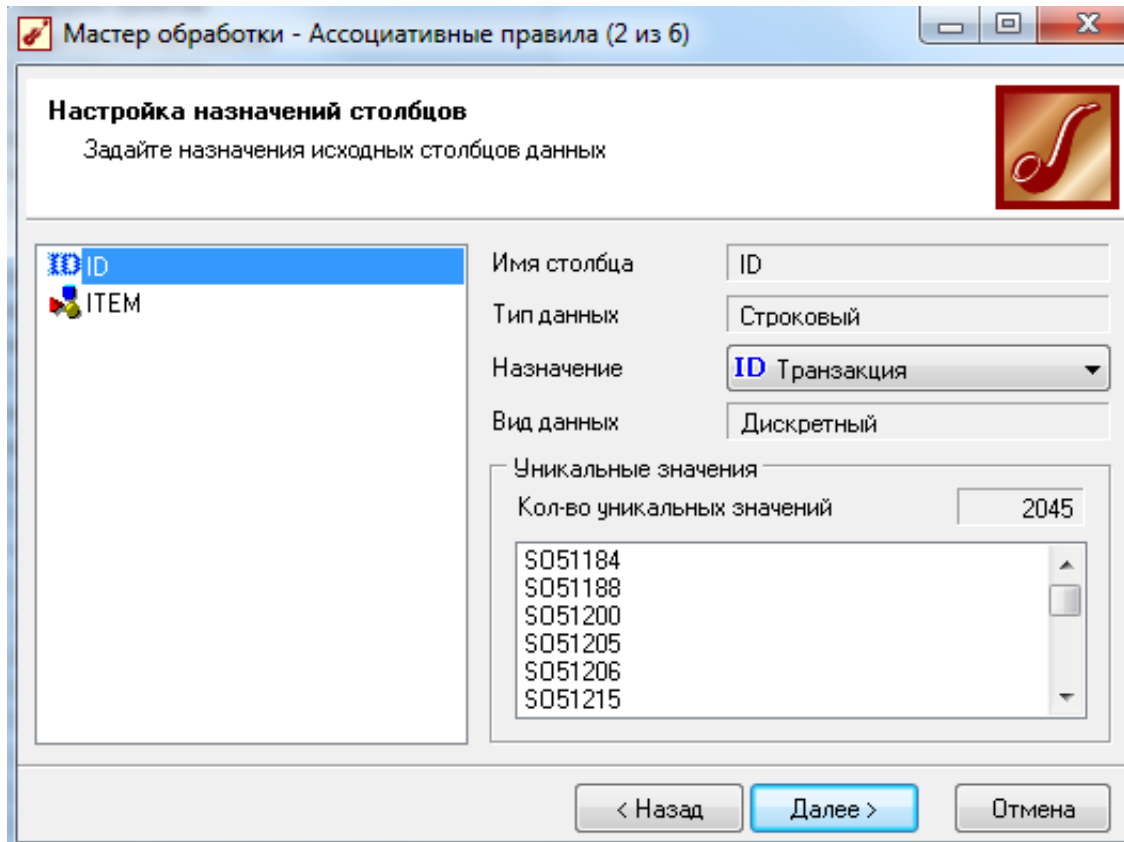
1. TID-множини можуть бути досить довгими, особливо при великій БД. Тому операції з ними можуть займати довгий час.
2. Велика кількість згенерованих кандидатів при малому значенні мінімальної підтримки (властивий також Apriori).

ПОБУДОВА АСОЦІАТИВНИХ ПРАВИЛ В DEDUCTOR

Надія І. Недашківська n.nedashkivska@gmail.com

Побудова Асоціативних правил в Deductor

Таблиця даних про продаж товарів в деякій торгівельній точці має тільки два поля "Номер чеку" і "Товар".



Мастер обработки - Ассоциативные правила (2 из 6)

Настройка назначений столбцов
Задайте назначения исходных столбцов данных

Имя столбца: ID

Тип данных: Строковый

Назначение: ID Транзакция

Вид данных: Дискретный

Уникальные значения

Кол-во уникальных значений: 2045

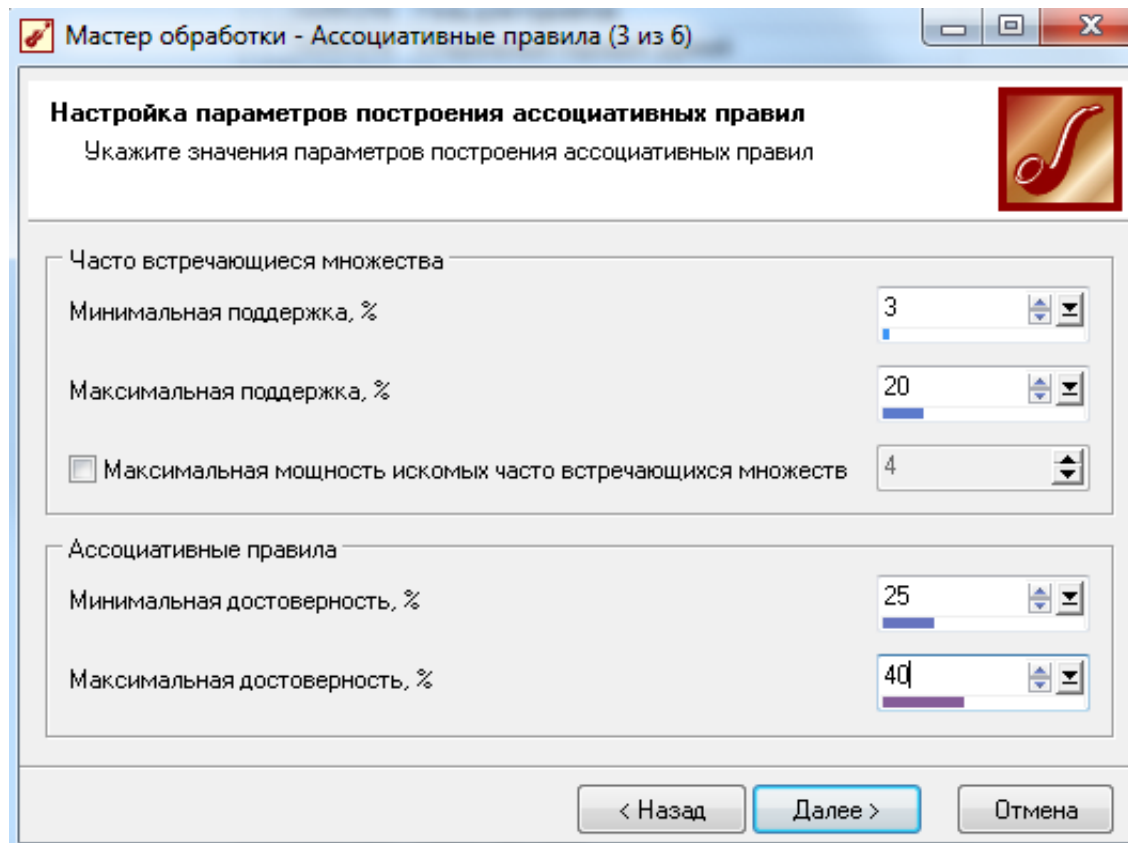
S051184
S051188
S051200
S051205
S051206
S051215

< Назад Далее > Отмена

Необхідно вирішити задачу аналізу споживчої корзини з метою подальшого використання результатів для стимулювання продаж.

Побудова Асоціативних правил в Deductor

Параметри побудови асоціативних правил: мінімальна і максимальна підтримка, мінімальна і максимальна достовірність, максимальна потужність частих наборів.



Мастер обработки - Ассоциативные правила (3 из 6)

Настройка параметров построения ассоциативных правил
Укажите значения параметров построения ассоциативных правил

Часто встречающиеся множества

Минимальная поддержка, % 3

Максимальная поддержка, % 20

☐ Максимальная мощность искомых часто встречающихся множеств 4

Ассоциативные правила

Минимальная достоверность, % 25

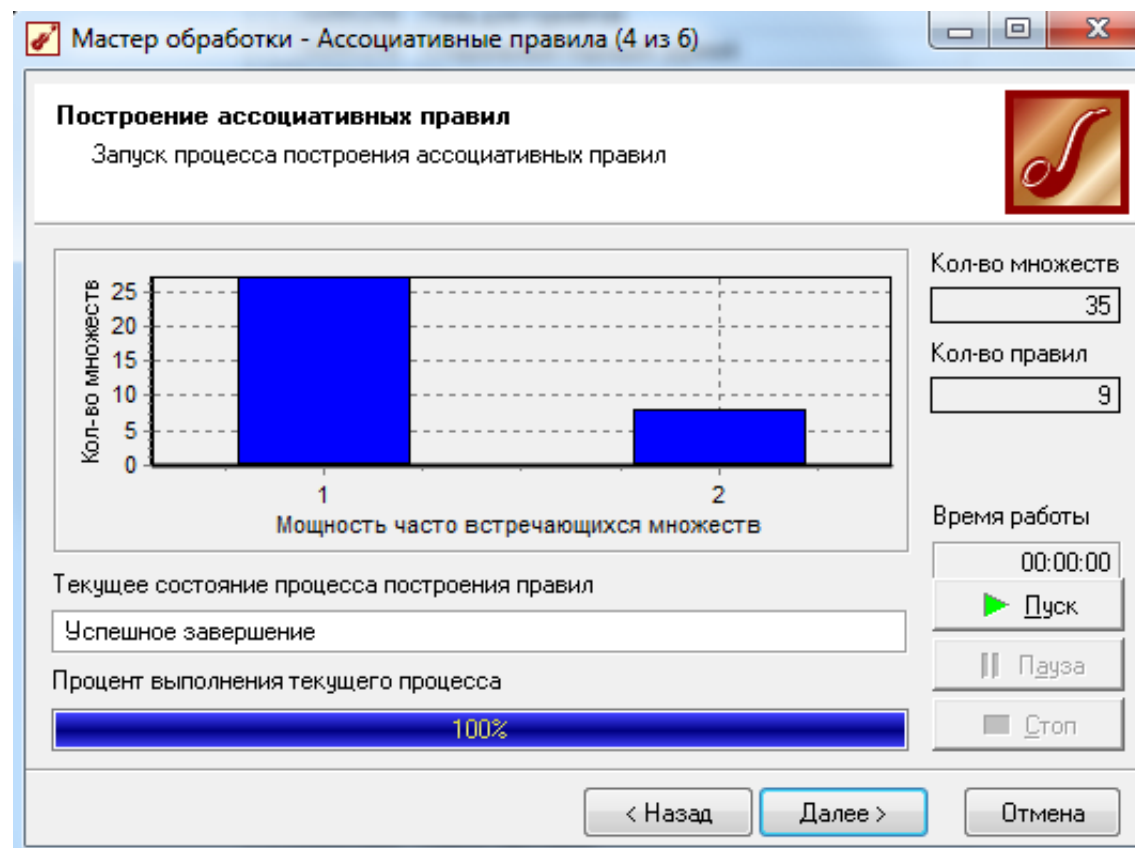
Максимальная достоверность, % 40

< Назад Далее > Отмена

Побудова Асоціативних правил в Deductor

Запуск процесу пошуку асоціативних правил. Відображається:

- загальна кількість частих наборів і знайдених правил,
- число частих наборів кожної з потужностей.



Візуалізатор “Часті набори”

Популярные наборы X Дерево правил X Что-если X					
ab. ▾ A↓ ▾ ▾ ▾					
Множеств: 35 из 35		Фильтр: Без фильтрации			
№	Номер множества	ab. Элементы	Поддержка		S Мощность
			Кол-во	%	
1	1	Антистатик спрей	101	4,94	1
2	2	Бумажное полотенце	87	4,25	1
3	3	Гель для туалетов	221	10,81	1
4	28	Гель для туалетов	66	3,23	2
		Мыло жидкое			
5	29	Гель для туалетов	66	3,23	2
		Стиральный порошок ручной			
6	4	Дозатор	85	4,16	1
7	5	Запасной баллон для освежителя	89	4,35	1
8	6	Зубная паста	288	14,08	1
9	30	Зубная паста	77	3,77	2
		Чистящий порошок универсальный			
10	7	Кондиционер для белья	93	4,55	1
11	31	Кондиционер для белья	79	3,86	2
		Стиральный порошок-автомат			
12	8	Мыло жидкое	200	9,78	1
13	32	Мыло жидкое	167	8,17	2
		Мыло кусковое			
14	9	Мыло кусковое	392	19,17	1
15	33	Мыло кусковое	131	6,41	2
		Средство для мытья посуды			
16	10	Освежитель воздуха	208	10,17	1

Візуалізатор “Асоціативні правила”

Таблица								
1 / 9								
	Номер правила	Номер элемента	Условие	Следствие	Поддержка		Достоверность	Лифт
					Кол-во	%		
▶	1	1	Гель для туалетов	Мыло жидкое	66	3,22738386308068	29,8642533936652	3,05361990950226
	2	1	Мыло жидкое	Гель для туалетов	66	3,22738386308068	33	3,05361990950226
	3	1	Гель для туалетов	Стиральный порошок ручн	66	3,22738386308068	29,8642533936652	3,13191785589976
	4	1	Стиральный порошок	Гель для туалетов	66	3,22738386308068	33,8461538461538	3,13191785589976
	5	1	Зубная паста	Чистящий порошок универ	77	3,76528117359413	26,7361111111111	1,87244339802131
	6	1	Чистящий порошок уни	Зубная паста	77	3,76528117359413	26,3698630136986	1,87244339802131
	7	1	Мыло кусковое	Средство для мытья посуд	131	6,40586797066015	33,4183673469388	4,49608955424275
	8	1	Чистящий порошок уни	Сода кальцинированная	96	4,69437652811736	32,8767123287671	5,05510351220517
	9	1	Чистящий порошок уни	Средство от накипи	76	3,71638141809291	26,027397260274	4,88312177956516

❑ Підтримка АП $Supp(X \Rightarrow Y) = p(X \cap Y) = \frac{|D_{F=X \cup Y}|}{|D|}$

❑ Достовірність АП $Conf(X \Rightarrow Y) = p(Y | X) = \frac{Supp(X \Rightarrow Y)}{Supp(X)}$

❑ Ліфт АП $Lift(X \Rightarrow Y) = \frac{Conf(X \rightarrow Y)}{Supp(Y)}$

Візуалізатор “Дерево правил”

Це дворівневе дерево.

1. Побудоване за умовою: на першому (верхньому) рівні знаходяться вузли з умовами, а на другому рівні - вузли з наслідком.

Правила відповідають на питання, що буде за заданої умови.

2. Побудоване за наслідком

Правила відповідають на питання, що потрібно, щоб був заданий наслідок.

Популярные наборы X Дерево правил X Что-если X				
Ассоциативные правила (по условию)				
Гель для туалетов (10,81%; 221)				
Мыло жидкое (9,78%; 200)				
Стиральный порошок ручной (9,54%; 195)				
Зубная паста (14,08%; 288)				
Чистящий порошок универсальный (14,28%; 292)				
Мыло хозяйственное (19,17%; 392)				
Количество правил: 3; Условие: Чистящий порошок универсальный				
Следствие	Поддержка		Достоверность, %	Лифт
	Кол-во	%		
Зубная паста	77	3,77	26,40	1,872
Сода кальцинированная	96	4,69	32,90	5,055
Средство от накипи	76	3,72	26,00	4,883

Якщо придбав порошок універсальний, то з імовірністю 26,4% придбає і зубну пасту.

Якщо придбав порошок універсальний, то з імовірністю 32,9% придбає і соду.

Якщо придбав порошок універсальний, то з імовірністю 26% придбає засіб від накипу.

Візуалізатор “Аналіз Що-Якщо”

Що отримаємо як наслідок, якщо виберемо дані умови.

Популярные наборы X Дерево правил X Что-если X

ab. 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

Элемент	Поддержка, %
Антистатик спрей	4,94
Бумажное полотенце	4,25
Гель для туалетов	10,81
Дозатор	4,16
Запасной баллон для освежителя	4,35
Зубная паста	14,08
Кондиционер для белья	4,55
Мыло жидкое	9,78
Мыло кусковое	19,17
Освежитель воздуха	10,17
Отбеливатель	9,73
Пена/соль для ванн	4,30
Перчатки резиновые	4,89
Покровное одноразовое для унитаза	3,52
Пятновыводитель	7,14
Салфетки бумажные	5,18
Сода кальцинированная	6,50
Средство для мытья пола	5,82
Средство для мытья посуды	7,43
Средство для ухода за мебелью	7,53
Средство для чистки кафеля	4,25
Средство для чистки плит	6,31
Средство от накипи	5,33
Средство по уходу за зеркалами и стеклами	4,35
Стиральный порошок ручной	9,54
Стиральный порошок-автомат	7,04
Чистящий порошок универсальный	14,28

Условие


Элемент	Поддержка, %
Зубная паста	14,08
Стиральный порошок ручной	9,54

Количество правил: 2

Следствие	Поддержка		Достоверность, %	Лифт
	Кол-во	%		
Гель для туалетов	66	3,23	33,80	3,132
Чистящий порошок универсальный	77	3,77	26,70	1,872

Застосування асоціативних правил



 **Аналіз ринкових кошиків** - виявлення наборів товарів, які часто купляються разом або ніколи не купляються разом

Мета:

- оптимізувати розміщення товарів на полицях,
- формувати персональні рекомендації,
- планувати рекламні компанії,
- більш ефективно управляти цінами і асортиментом

Застосування асоціативних правил

- ❑ **Виявлення долі клієнтів**, які позитивно відносяться до нововведень в їх обслуговуванні
- ❑ **Визначення профіля відвідувача веб-ресурсу**
- ❑ **Виявлення долі випадків**, в яких нові ліки викликають небезпечну побічну дію
- ❑ **Виявлення тематики в колекціях текстових документів** (тема – сукупність термінів, які часто зустрічаються разом у вузькій підмножині документів)

Недолік – достатньо жорстка вимога, щоб у тексті зустрічалися всі слова теми. Імовірнісні моделі більш адекватні. Метою аналізу може бути: - отримання ознак для знаходження термінів, - виявлення найбільш чітких тем, - формування початкових наближень для Topic Models.