МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Інститут Комп'ютерних наук та інформаційних технологій
Кафедра Програмної інженерії та інтелектуальних технологій управління
Спеціальність 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки та інтелектуальні системи

ЛАБОРАТОРНА РАБОТА №2 за курсом

«Інтелектуальний аналіз даних та видобування знань»

Тема лабораторної роботи Ідентифікація асоціативних правил в WEKA

Виконав студент 5 курсу, групи КН-М422

Захар ПАРАХІН

(підпис, прізвище та ініціали)

Перевірила Оксана ІВАЩЕНКО

(підпис, прізвище та ініціали)

Харків 2022

ЗМІСТ

Вступ

Алгоритм апріорі існує для пошуку частих наборів елементів у наборі даних для логічного правила асоціації.

Алгоритм називається Apriori, оскільки він використовує попередні знання про часті властивості набору елементів.

WEKA має власну реалізацію цього алгоритма для проведення побудови правил асоціації.

Апріорний алгоритм може бути повільним. Основним обмеженням є час, потрібний для зберігання великої кількості наборів кандидатів із дуже частими наборами елементів, низькою мінімальною підтримкою або великими наборами елементів, тобто це неефективний підхід для великої кількості наборів даних.

1 Хід виконання роботи

1.1 Алгоритм Apriori

Алгоритм називається Apriori, оскільки він використовує попередні знання про часті властивості набору елементів. Щоб підвищити ефективність порівневої генерації частих наборів елементів, використовується важлива властивість, яка називається властивістю Apriori, яка допомагає зменшити простір пошуку.

Апріорна властивість – усі непорожні підмножини частих наборів елементів мають бути частими. Ключовою концепцією алгоритму Апріорі є його немонотонність опорної міри. Апріорі припускає, що усі підмножини частого набору елементів мають бути частими (властивість Apriori). Якщо набір елементів є рідкісним, усі його надмножини будуть рідкісними. Головне завдання Apriori реалізувати знаходження правил асоціації з певним рівнем довіри і впевненості, наприклад, якщо існує певне співпадіння у наборі елементів множини A і у 50-60 відсотках з множиною B, що включає A і є частим набором, то можна припустити, що при виборі елементів з B множини, можна брати і A елементи у коплекті. Алгоритм працює за частотами використання певних наборів елементів.

```
Apriori(T, ε)
    L_1 ← {large 1 - itemsets}
    k ← 2
    while L_{k-1} is not empty
        C_k ← Apriori_gen(L_{k-1}, k)
        for transactions t in T
            D_t ← {c in C_k : c ⊆ t}
            for candidates c in D_t
                count[c] ← count[c] + 1

        L_k ← {c in C_k : count[c] ≥ ε}
        k ← k + 1

    return Union(L_k)

Apriori_gen(L, k)
     result ← list()
     for all p ∈ L, q ∈ L where p_1 = q_1, p_2 = q_2, ..., p_{k-2} = q_{k-2} and p_{k-1} < q_{k-1}
         c = p ∪ {q_{k-1}}
         if u ∈ L for all u ⊆ c where |u| = k-1
             result.add(c)
      return result
```

Рисунок 1 - Приклад псевдокоду алгоритму:

1.2 WEKA майнінг для асоціацій

Вкладка “Assosiate” має схеми для навчання асоціативним правилам. Алгоритми можна обрати, налаштувати і виконати як для фільтрів, класифікацій так і для кластерів.Selectiing attributes - має перебір всіх можливих комбінацій властивостей лоя пошуку підмножини властивостей, для цього треба налаштувати оцінку атрибутів і метод пошуку, де оцінка визначає за яким методом назначаються значимості атрибутів, а пошук - стиль пошуку мідмножин. **Use full training set** - значимість визначається для повного набору даних. **Cross-validation -** значимість визначається шляхом кросс-валідації.

1.2.1 Набір даних weather

Relation: weather.symbolic

No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: **play** Nominal
1	rainy	mild	high	FALSE	yes
2	rainy	mild	high	TRUE	no
3	rainy	cool	normal	FALSE	yes
4	rainy	cool	normal	TRUE	no
5	sunny	cool	normal	FALSE	yes
6	rainy	mild	normal	FALSE	yes
7	overcast	hot	normal	FALSE	yes

Рисунок 2 - Набір даних з файлу weater_reduced.arff

minimal support = 0.6

1. Підраховуємо всі елементи

outlook		windy	
rainy	5	true	2
sunny	1	false	5
overcast	1		

temperature		play		
	mild	3	yes	5
	cool	3	no	2
	hot	1		
humidity				
	high	2		
	normal	5		

2. Прибираємо, ті що мають меншу підтримку ніж 0.6

$$\text{Support (A)} = \frac{\text{Number of transaction in which A appears}}{\text{Total number of transactions}}$$

Отримуємо такі елементи:

outlook rainy	5
windy false	5
play yes	5
humidity normal	5

3. Комбінуємо елементи

outlook{rainy} windy{false}	3
outlook{rainy} play{yes}	3
outlook{rainy} humidity{normal}	3
windy{false} play{yes}	5
windy{false} humidity{normal}	4
play{yes} humidity{normal}	4

windy{false} => play{yes} відповідає min_sup > 0.6,
також play{yes} => windy{false}
humidity{normal} play{yes} => windy{false}
humidity{normal} windy{false} => play{yes}

тобто маємо логіку:

windy{false} => play{yes}
play{yes} => windy{false}

Що ми отримуємо при використанні WEKA.

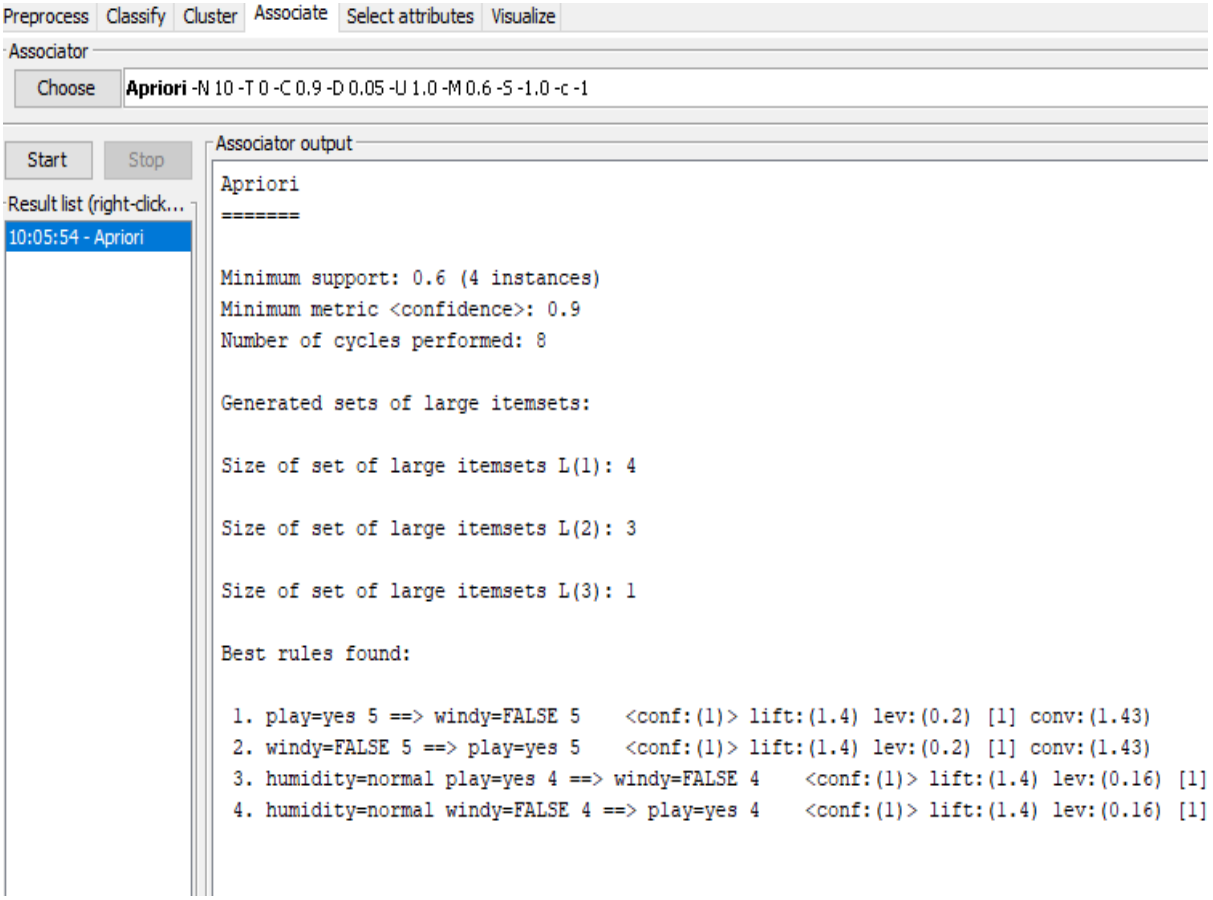

Рисунок 2 - Результати Apriori в WEKA

З цього бачимо наступну логіку:

windy{false} => play{yes}
play{yes} => windy{false}
humidity{normal} play{yes} ==> windy{false}
humidity{normal} windy{false} => play{yes}
Результати співпадають, система одразу розглядає і комбінацію декількох елементів.

1.2.2 Набір даних bank-data

Кількість елементів у цьому датасеті занадто, великі і для використання ручного розрахунку. Основним обмеженням є час, потрібний для зберігання великої кількості наборів кандидатів із дуже частими наборами елементів, низькою мінімальною підтримкою або великими наборами елементів, тобто це неефективний підхід для великої кількості наборів даних. В результаті використовуємо логіку індукції маємо факт, що алгоритм працює як у 1.2.1, також WEKA автоматизовано цей процес.

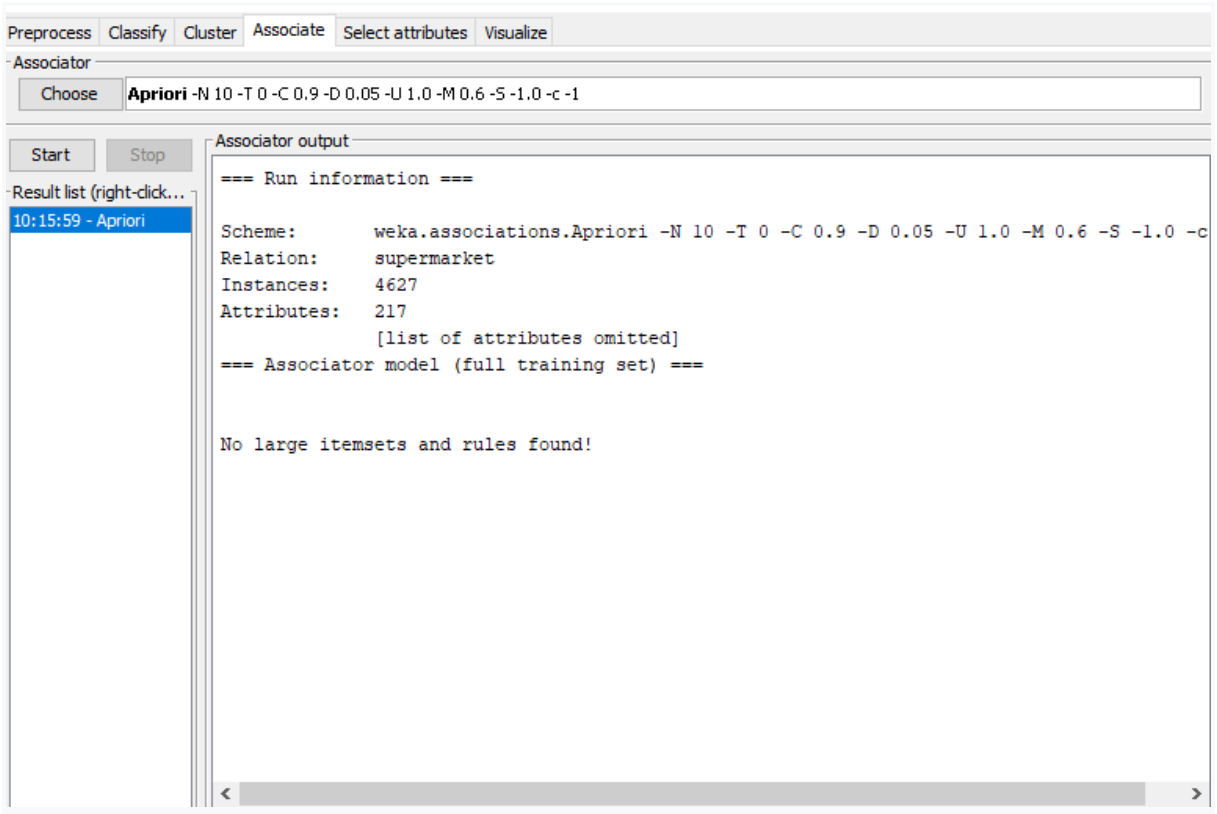

Рисунок 3 - результати для supermarket.arff

За результатами бачимо, що логічних правил не знайдено у цьому наборі даних.

1.2.3 Набір даних supermarket

Кількість елементів у цьому датасеті занадто, великі і для використання ручного розрахунку. Основним обмеженням є час, потрібний для зберігання великої кількості наборів кандидатів із дуже частими наборами елементів, низькою мінімальною підтримкою або великими наборами елементів, тобто це неефективний підхід для великої кількості наборів даних. В результаті використовуємо логіку індукції маємо факт, що алгоритм працює як у 1.2.1, також WEKA автоматизовано цей процес.

Рисунок 4 - результати при min_support = 0.1

Тут ми отримаємо наступний набір правил:

children='(-inf-0.3]' save_act=YES mortgage=NO pep=NO => married=YES

sex=FEMALE children='(-inf-0.3]' mortgage=NO pep=NO => married=YES

children='(-inf-0.3]' current_act=YES mortgage=NO pep=NO =>married=YES

children='(-inf-0.3]' mortgage=NO pep=NO => married=YES

children='(-inf-0.3]' car=NO mortgage=NO pep=NO 62 ==> married=YES

married=YES children='(-inf-0.3]' save_act=YES current_act=YES => pep=NO

married=YES children='(-inf-0.3]' save_act=YES mortgage=NO => pep=NO

married=YES children='(-inf-0.3]' current_act=YES mortgage=NO => pep=NO

sex=FEMALE married=YES children='(-inf-0.3]' mortgage=NO => pep=NO

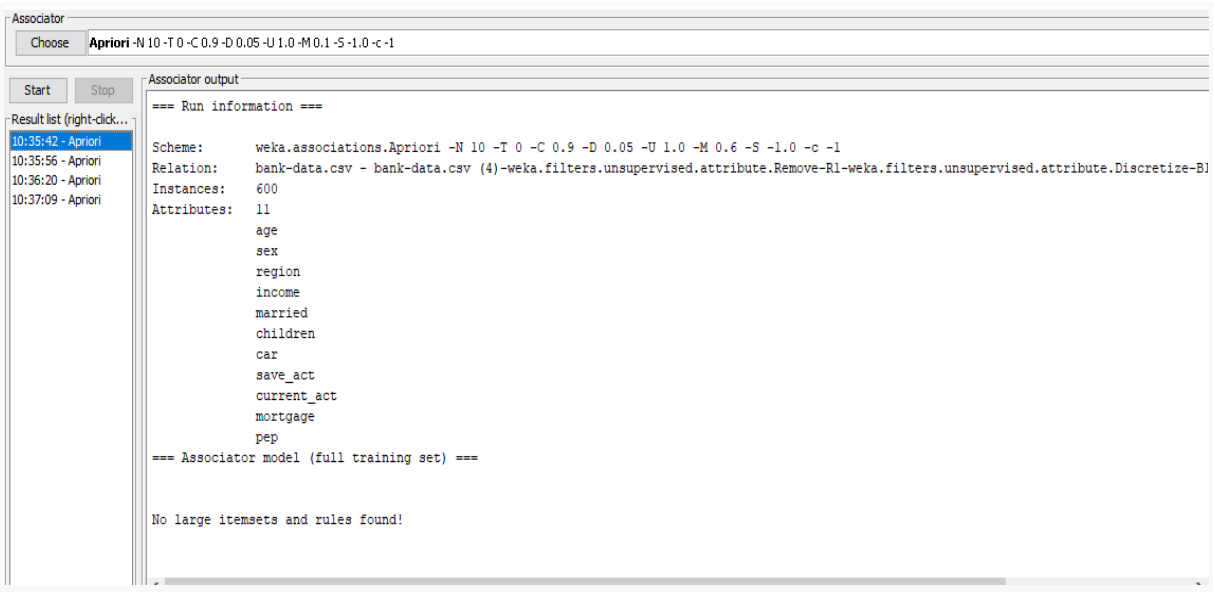

Рисунок 5 - при min support = 0.6

При збільшені мінімальної підтримки вже не отримуємо жодних правил, бо як видно з формули в 1.2.1 основний розрахунок, підтримки певного набору залежить від кількості загальної і при їх збільшенні приходиться або розбивати на менші набори і потім вже з цих нових даних шукати правила.

Висновки

Було вивчено алгоритм Apriori, використання ідентифікація асоціативних правил. Також виконані відповідні завдання з використання WEKA для трьох наборів даних з яких були отримані відповідні результати.

Закріплено знання з асоціативних правил і використання методів інтелектуального аналізу даних, розглянуто один з таких алгоритмів та їх обмеження.

Список джерел інформації

1 Data Mining: Practical Machine Learning Tools and Technique \ Ian Witten, Eibe Frank, Mark Hall

2 WEKA. Руководство по использованию \ Хабр online ресурс https://habr.com/ru/post/590565/

3 Інтелектуальний аналіз даних: Навчальний посібник \ А. О. Олійник, С. О. Субботін, О. О. Олійник