University of Science and Technology Bannu

Data Mining

Lesson 8

February 21, 2024

Association Rule Mining

Learning Objectives

Lesson outline

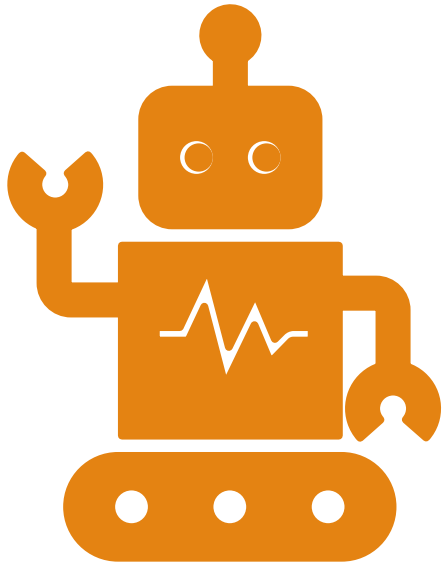
- ✓ What is Association Rule and frequent pattern mining?
- ➢ **The Apriori Algorithm for Association Rule**
- ❑ How FP-Growth Algorithm Work in Association Rule?
- ❑ Practical Implementation of the Apriori Algorithm.
- ❑ Practical Implementation of FP-Growth Algorithm.

Frequent patterns and association rules

- Imagine that you are a sales manager at ***AllElectronics***
- Primary responsibility "**Sales Growth**"
- Customer purchased PC and Digital Camera
- You are recommending new item to her/him
- You can recommend a new item to her/him if you have a knowledge
- Knowledge of "which products are frequently purchased by your customers following their purchases of a PC and a digital camera in sequence"
- Frequent patterns and association rules are the knowledge that you want to mine in such a scenario.

Apriori Algorithm:

- **Apriori** is a seminal algorithm proposed by **R. Agrawal and R. Srikant in 1994** for mining frequent itemsets.
- Apriori employs an **iterative approach** known as a level-wise search, where k-itemsets are used to explore k +1 itemsets.
- First, the set of **frequent 1-itemsets** is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted by L1.
- Next, L1 is used to find L2, the set of frequent 2-itemsets, which is used to find L3, and so on, until no more frequent k-itemsets can be found.
- The finding of each Lk requires one full scan of the database.



Apriori Property

Apriori property: All nonempty subsets of a frequent itemset must also be frequent.

By definition, if an item set "**I**" does not satisfy the minimum support threshold, min_sup, then "***I***" is not frequent, that is, P(I)< min sup. If an item A is added to the itemset I, then the resulting itemset (i.e., $I \cup A$) cannot occur more frequently than "**I**". Therefore, $I \cup A$ is not frequent either, that is, P($I \cup A$) <min_sup

Steps

- A two-step process is followed, consisting of **join step** and **prune actions step**
- **1 Join Step:** We use the Join step to generate itemsets that can possibly be frequent itemsets. For this, we start by creating sets containing single items. If there are N items in the dataset, we create N candidate sets.
- **2. Prune Step:** The pruning step in the apriori algorithm is based on the concept that a subset of a frequent itemset must also be a frequent itemset. In other words, if we have an itemset having a subset that is not a frequent itemset, the itemset cannot be a frequent itemset.

Example

Let’s look at a concrete example, based on the AllElectronics transaction database, D, There are nine transactions in this database, that is, D 9 and min_support=2

TID	List of items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Step-1

STEP-1 Scan database D to find Unique Items or Item set one.

TID	List of items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

C1=

One-Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

Step-2

Step-2 Compare candidate support count with minimum support Count to get L1. Our min support count is 2

C1=

One-Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

L1=

Frequent One-Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

Step-3

Step-3 Create Frequent Itemset with Two Items: To create frequent itemset with two items, we will first create the candidate itemset with two items. For this, we will join all the frequent items set with one item with each other. After joining, we will get the following itemset.

Frequent One-Itemset	Support Count
I1	6
I2	7
I3	6
I4	2
I5	2

(L1, l2), (l1,l3),(l1,l4),(l1,l5),(l2,l3),(l2,l4),(l2,l5),(I3,I4),(I3,I5),(I4,I5)

After creating the itemset with two items, we need to prune the itemset having subsets that are not frequent itemset. As the {I1}, {I2}, {I3}, {I4}, and {I5} all are frequent itemset, no itemset will be removed from the above list while pruning.

C2=

TID	List of items
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

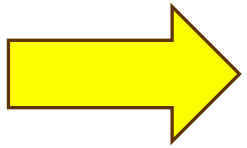
Two Itemset	Min Support count
(l1, l2)	4
(l1, l3)	4
(l1, l4)	1
(l1,l5)	2
(l2,l3)	4
(l2,l4)	2
(l2,l5)	2
(I3,I4)	0
(I3,I5)	1
(I4,I5)	0

Step-4

Step-4 Compare candidate support count with minimum support Count to get L2.

C2

Two Itemset	Min Support count
(l1, l2)	4
(l1, l3)	4
(l1, l4)	1
(l1,l5)	2
(l2,l3)	4
(l2,l4)	2
(l2,l5)	2
(I3,I4)	0
(I3,I5)	1
(I4,I5)	0



L2

Frequent Two Itemset	Min Support count
(I1, I2)	4
(I1, I3)	4
(I1, I5)	2
(I2, I3)	4
(I2, I4)	2
(I2, I5)	2

Step-5 Create Frequent Itemset with Three Items

To create frequent itemset with three items, we will first create the candidate itemset with three items. For this, we will join all the frequent items set with one item with each other. After joining, we will get the following itemset.

Unique Items={(I1, I2, I3,I4,I5)}

={(I1, I2, I3),(I1,I2,I4),(I1,I2,I5),(I2,I3,I4),(I2,I3,I5),(I3,I4,I5),(I1,I3,I5), (I2, I4, I5)}

3-Itemset	SUBSET	All the subsets are frequent itemset?
(I1, I2, I3)	(I1,I2),(I1,I3),(I2,I3)	YES
(I1, I2,I4)	(I1, I2), (I1,I4),(I2,I4)	NO
(I1,I2,I5)	(I1, I2),(I1,I5),(I2,I5)	YES
(I2,I3,I4)	(I2,I3),(I2,I4),(I3,I4)	NO
(I2,I3,I5)	(I2,I3),(I2,I5),(I3,I5)	NO
I3,I4,I5	(I3,I4),(I3,I5),(I4,I5)	NO
(I1,I3,I5)	(I1,I3),(I1,I5),(I3,I5)	NO
(I2, I4, I5)	(I2,I4),(I2,I5),(I4,I5)	NO

Apriori property (pruning)

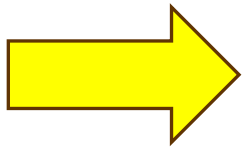
L2

Frequent Two Itemset	Min Support count
(I1, I2)	4
(I1, I3)	4
(I1, I5)	2
(I2, I3)	4
(I2, I4)	2
(I2, I5)	2

3-Itemset	SUBSET	All the subsets are frequent itemset?
(I1, I2, I3)	(I1,I2),(I1,I3),(I2,I3)	YES
(I1, I2,I4)	(I1, I2), (I1,I4),(I2,I4)	NO
(I1,I2,I5)	(I1, I2),(I1,I5),(I2,I5)	YES
(I2,I3,I4)	(I2,I3),(I2,I4),(I3,I4)	NO
(I2,I3,I5)	(I2,I3),(I2,I5),(I3,I5)	NO
(I3,I4,I5)	(I3,I4),(I3,I5),(I4,I5)	NO
(I1,I3,I5)	(I1,I3),(I1,I5),(I3,I5)	NO
(I2, I4, I5)	(I2,I4),(I2,I5),(I4,I5)	NO

Therefore, C3 = (I1, I2, I3) , (I1, I2, I5) after pruning.

3-Itemset	SUBSET	All the subsets are frequent itemset?
(I1, I2, I3)	(I1,I2),(I1,I3),(I2,I3)	YES
(I1, I2,I4)	(I1, I2), (I1,I4),(I2,I4)	NO
(I1,I2,I5)	(I1, I2),(I1,I5),(I2,I5)	YES
(I2,I3,I4)	(I2,I3),(I2,I4),(I3,I4)	NO
(I2,I3,I5)	(I2,I3),(I2,I5),(I3,I5)	NO
(I3,I4,I5)	(I3,I4),(I3,I5),(I4,I5)	NO
(I1,I3,I5)	(I1,I3),(I1,I5),(I3,I5)	NO
(I2, I4, I5)	(I2,I4),(I2,I5),(I4,I5)	NO



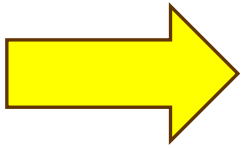
C3

3-Itemset
(I1, I2, I3)
(I1, I2, I5)

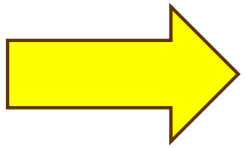
Step-6

Scan D from count each candidate and Compare candidate support count with minimum support count.

3-Itemset
(I1, I2, I3)
(I1, I2, I5)



3-Itemset	Min Support count
(I1, I2, I3)	2
(I1, I2, I5)	2



L3

Frequent 3-Itemset	Min Support count
(I1, I2, I3)	2
(I1, I2, I5)	2

Calculate Frequent Itemset with Four Items

Now, we will calculate the frequent itemset with four items. For this, we will first join the items in the frequent itemset with three items to create itemset with four items. below.

Frequent 3-Itemset	Min Support count
(I1, I2, I3)	2
(I1, I2, I5)	2

We will get only one itemset as shown**:(I1, I2, I3, I5)**

Step-7

(I1, I2, I3, I5)

The above itemset has four subsets with three elements i.e. ***{I2, I3, I5}, {I1, I3, I5}, {I1, I2, I5}, {I1, I2, I3}. In these itemset, {I1, I2, I5} and {I1, I2, I3} are not frequent itemset in L3***. Hence, we will prune the itemset {I1, I2, I3, I5}. Thus, we have no candidate set for itemset with 4 items. Hence, the process of frequent itemset generation stops here.

L3

Frequent 3-Itemset	Min Support count
(I1, I2, I3)	2
(I1, I2, I5)	2

Thus, C4 = Ø, and the algorithm terminates, having found all of the frequent itemset.

Generating association rules:

Once the frequent itemsets from transactions in database D have been found, it is straightforward to generate strong association rules from them. This can be done using below equation for confidence:

$$\boldsymbol{Confidence\ (A \rightarrow B) = P(B/A) = \frac{support\ count(A \cup B)}{support\ count(A)}}$$

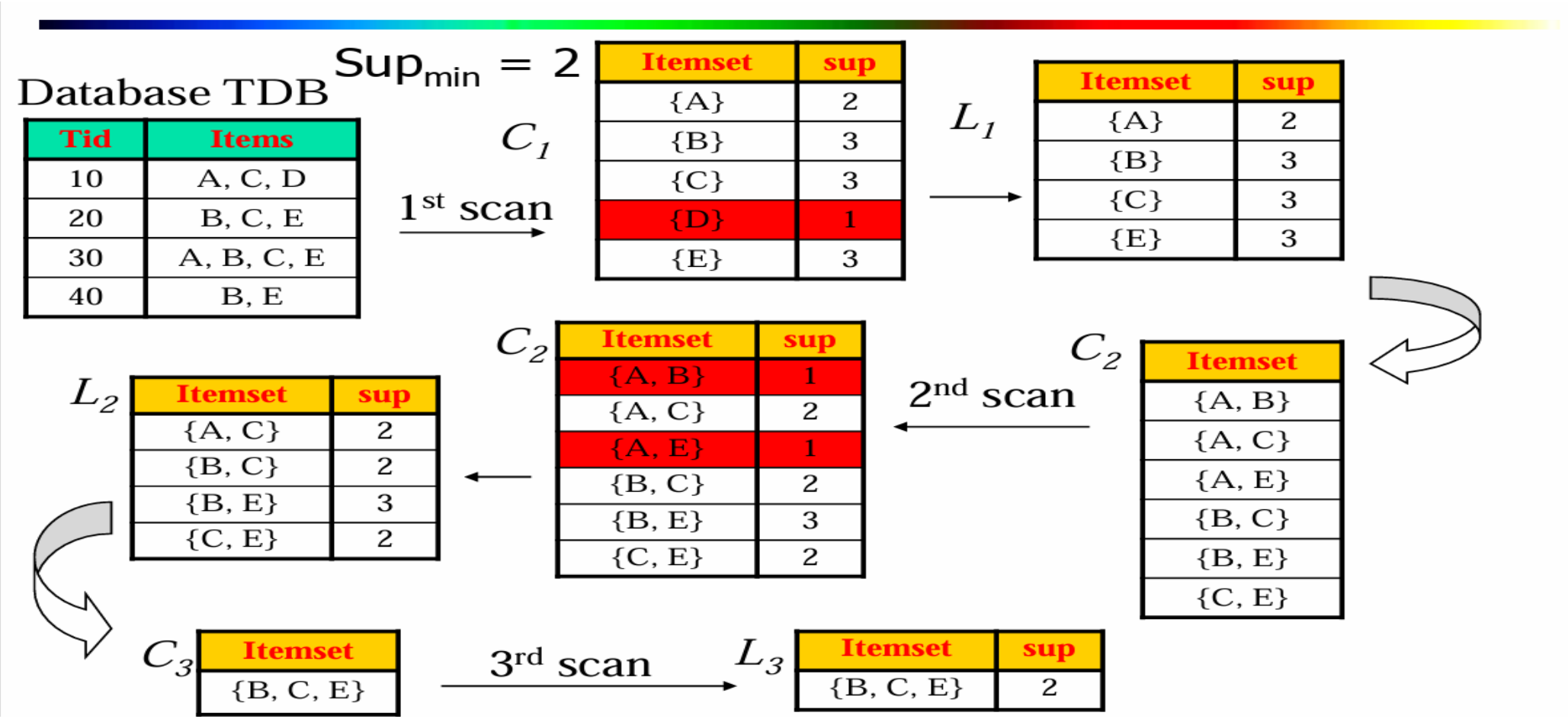
The conditional probability is expressed in terms of itemset support count, where $\boldsymbol{support\ count(A \cup B)}$ is the number of transactions containing the itemsets A B, and support count(A) is the number of transactions containing the itemset A.

Example

Example: The data contains frequent itemset **X = {I1, I2, I5}**. What are the association rules that can be generated from X?

Frequent item set	Subset	Association Rule
X = {I1, I2, I5}	(I1, I2)	$\{I1, I2\} \Rightarrow I5, \quad confidence = 2/4 = 50\%$
	(I1, I5)	$\{I1, I5\} \Rightarrow I2, \quad confidence = 2/2 = 100\%$
	(I2, I5)	$\{I2, I5\} \Rightarrow I1, \quad confidence = 2/2 = 100\%$
	(I1)	$I1 \Rightarrow \{I2, I5\}, \quad confidence = 2/6 = 33\%$
	(I2)	$I2 \Rightarrow \{I1, I5\}, \quad confidence = 2/7 = 29\%$
	(I5)	$I5 \Rightarrow \{I1, I2\}, \quad confidence = 2/2 = 100\%$

2nd Example


Database TDB
Sup_min = 2
Tid | Items
10 | A, C, D
20 | B, C, E
30 | A, B, C, E
40 | B, E
C1
1st scan
Itemset | sup
{A} | 2
{B} | 3
{C} | 3
{D} | 1
{E} | 3
L1
Itemset | sup
{A} | 2
{B} | 3
{C} | 3
{E} | 3
C2
Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}
2nd scan
C2
Itemset | sup
{A, B} | 1
{A, C} | 2
{A, E} | 1
{B, C} | 2
{B, E} | 3
{C, E} | 2
L2
Itemset | sup
{A, C} | 2
{B, C} | 2
{B, E} | 3
{C, E} | 2
C3
Itemset
{B, C, E}
3rd scan
L3
Itemset | sup
{B, C, E} | 2

The Apriori Algorithm

- Pseudo-code:

C_k: Candidate itemset of size k
L_k : Frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do**
begin
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in C_{k+1} that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
end
return $\cup_k L_k$;

Limitations of Apriori

- Apriori suffer from two nontrivial costs:

1.It may still need to generate a huge number of candidate sets. For example, if there are 104 frequent 1-itemsets, the Apriori algorithm will need to generate more than 10^7 candidate 2-itemsets.

2.It may need to repeatedly scan the whole database and check a large set of candidates by pattern matching. It is costly to go over each transaction in the database to determine the support of the candidate itemsets.