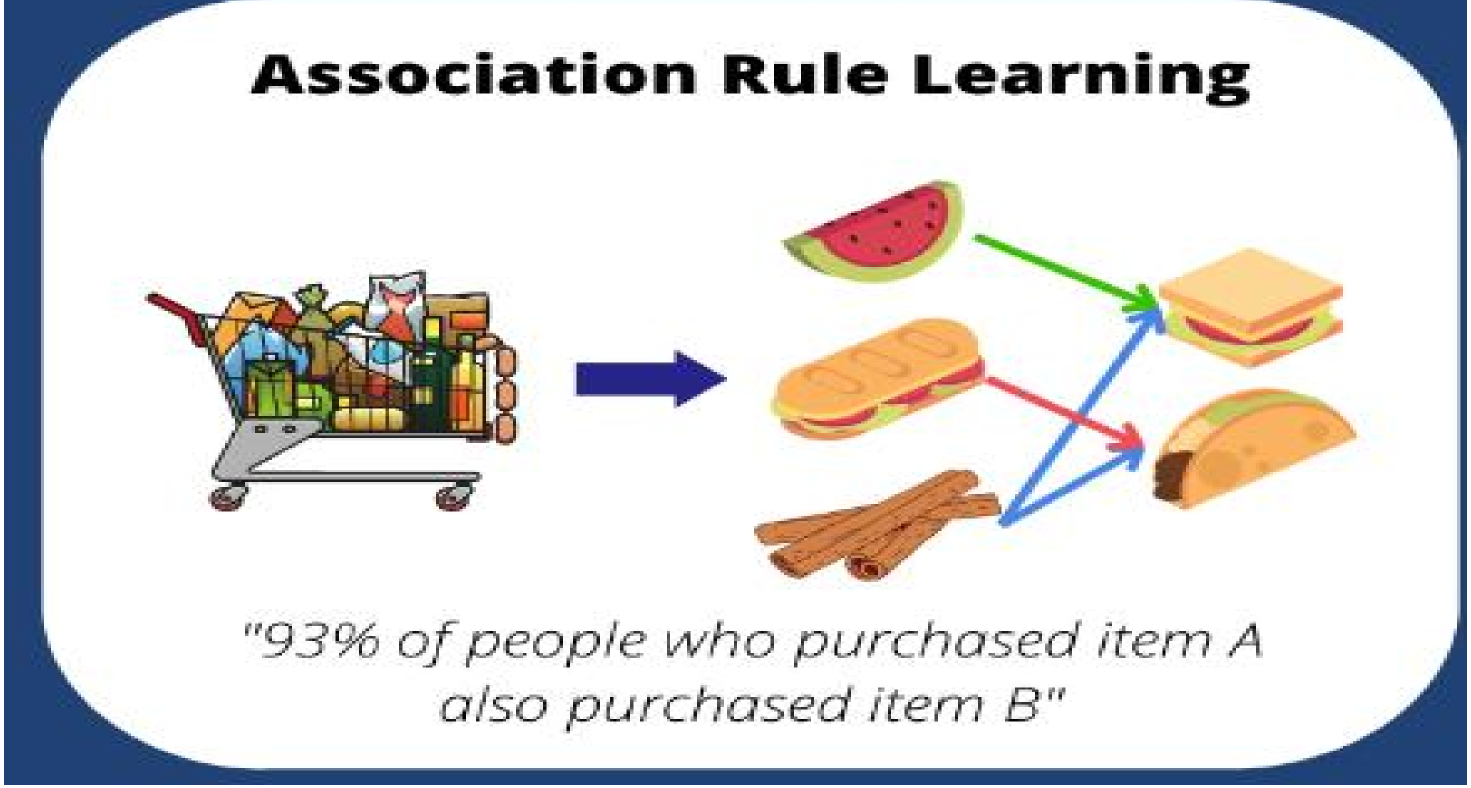


Association Rule Mining-Apriori Algorithm

By
Dr. Siddique Ibrahim
Assistant Professor
VIT-AP University
Amaravati

Case Study

- Imagine that you are a sales manager at Vijayawada Electronics, and you are talking to a customer who recently bought a LED TV and a Sound bar from the store.

What should you recommend to her/him next???

Information about which products are frequently purchased by your customers following their purchases of a LED TV and a Sound bar in sequence would be very helpful in making your recommendation.

Frequent patterns and association rules are the knowledge that you want to mine in such a scenario.

Introduction

- Data mining is the discovery of knowledge and useful information from the large amounts of data stored in databases.

- *Association Rules:* Describing association relationships among the attributes in the set of relevant data.

Frequent patterns

- Frequent patterns are patterns (e.g., itemsets, subsequences, or substructures) that appear frequently in a data set.

For example:

A set of items, such as milk and bread, that appear frequently together in a transaction data set is a frequent itemset.

A subsequence,

- such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (frequent) sequential pattern.

A substructure

- Can refer to different structural forms, such as subgraphs, subtrees, or sublattices, which may be combined with itemsets or subsequences.
- If a substructure occurs frequently, it is called a (frequent) structured pattern.

- 10 customer purchased Bread
- 8Cus Bread
- 2 Cust Bread & suger
- 5 Cust Bread & coffee powder
- 6 Cust Bread & Milk
- 9 Cust Bread & Jam

Why Mining frequent pattern?

- Finding frequent patterns plays an essential role in mining associations, correlations, and many other interesting relationships among data.

- Moreover, it helps in data classification, clustering, and other data mining tasks.

- Thus, frequent pattern mining has become an important data mining task and a focused theme in data mining research

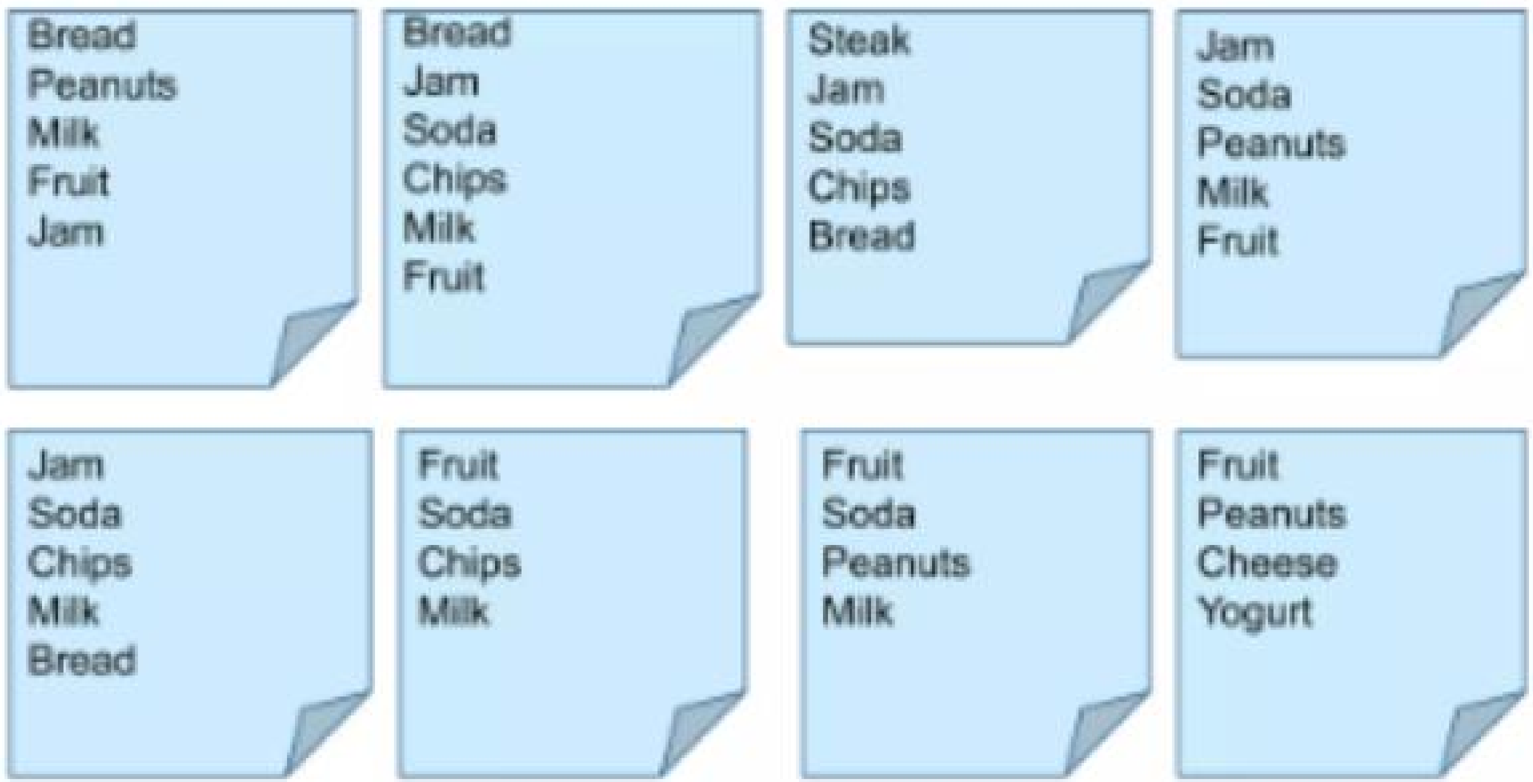
Frequent itemset mining

- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.
- With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases.
- The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes such as catalog design, cross-marketing, and customer shopping behavior analysis.

Market basket analysis.

- A typical example of frequent itemset mining is market basket analysis. This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”

Market basket Transactions



What is Association Rule?

- Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories

- Applications
  - Basket data analysis
  - Cross-marketing
  - Catalog design

Find any interesting information from the transaction.

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Association Rule Mining

- Association rule mining searches for interesting relationships among items in a given data set.
- Which groups/sets of items are customers likely to purchase on a given trip to the store?
- Which are product are moving fast?
- Which combination will be pushed hardly for purchase?

Association Rule Mining

- Result will be used for advertising strategies, as well as catelog design.

Measures

- A set of items is referred to as an itemset.
- The set {Laptop, Anti-virus software} is a 2-itemset.
- The occurrence frequency of an itemset is the number of transactions that contain the itemset.
- This is known as freqency, support_Count, or count of the itemset.

Support Measure

- Support indicates how frequently a rule or an itemset appears in the dataset. It represents the proportion of transactions in which the itemset occurs. In other words, it shows how popular or common an item or a combination of items is within all transactions.

- Support for an itemset X: $\text{Support}(X) = \frac{\text{Number of transactions containing } X}{\text{Total number of transactions}}$

- For example, if 100 transactions were recorded, and 20 of them contained milk and bread together, then the support for {milk, bread} would be 20/100 = 0.2 or 20%.

Confidence Measure

- Confidence measures how often a rule is found to be true. In association rules, it is the likelihood that a rule's consequence occurs given that its premise has occurred. In other words, confidence measures the conditional probability that items in the consequent (right-hand side) of the rule are also present in transactions that contain the antecedent (left-hand side).
  - Confidence for the rule $X \rightarrow Y$:

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cup Y)}{\text{Support}(X)}$$

- For example, if the rule is {milk} → {bread} and the confidence is 80%, it means that 80% of the transactions that contain milk also contain bread.

Finding frequent itemsets and strong rule

- A itemset satisfies minimum support if the occurrence frequency of the itemset is greater than or equal to the min_sup and total no of transaction.
- A rules that satisfy both a minimum support threshold(Min_sup) and a minimum confidence threshold (Min_conf) are called Strong.

Classification of ARM

- Boolean Association Rule: If a rule concerns associations between the presence or absence of items.

$$computer \Rightarrow antivirus\_software\ [support = 2\%, confidence = 60\%].$$

Classification of ARM

- Quantitative Association Rule: If a rule describe associations between quantitative items or attributes are partitioned into intervals.

- age(X,”30..40”) ^ income(X,”50k...75k) =>buys (X,iphone)

Apriori Algorithm

- Apriori is an influential algorithm for mining frequent itemsets for Boolean association rule.
- The name is based on the fact that the algorithm uses prior knowledge of freqent itemsets properties.
- Apriori is iterative and level wise search.

- First the algorithm generate 1-itemset this is denoted as L1.

- L1 is used to find L2(set of frequent 2 itemsets)which is used to find L3 and so on.

Apriori Property

- To improve the efficiency of the level wise generation of frequent itemsets.

- An important property called the Apriori property.
- It is used to reduce the search space.

- All nonempty subsets of a frequent itemset must also be frequent.

- All subsets of a freqent itemset must also be frequent.

Example

- If an itemset *I* does not satisfy the minimum support threshold(*Min_sup*), then I is not freqent. i.e p(I) < *min_sup*.

- If an item A is added to the itemset *I*, then the resulting itemset (i.e *I U A*) cannot satisfy *min_sup*.
- Therefore *I U A* is not frequent itemset.

Anti-Monotone

- If a **set** cannot pass the test, all of its **superset** will fail the same test as well.

- It is called *anti-monotone* because the property is monotonic in the contest of failing a test.

Apriori Algorithm

1. In the first iteration of the algorithm, each item is a member of the set of candidate 1-itemsets, C_1. The algorithm simply scans all of the transactions to count the number of occurrences of each item.

2. Suppose that the minimum support count required is 2, that is, $min\_sup = 2$. (Here, we are referring to *absolute* support because we are using a support count. The corresponding relative support is $2/9 = 22\%$.) The set of frequent 1-itemsets, L_1, can then be determined. It consists of the candidate 1-itemsets satisfying minimum support. In our example, all of the candidates in C_1 satisfy minimum support.

3. To discover the set of frequent 2-itemsets, L_2, the algorithm uses the join $L_1 \bowtie L_1$ to generate a candidate set of 2-itemsets, C_2.[7] C_2 consists of $\binom{|L_1|}{2}$ 2-itemsets. Note that no candidates are removed from C_2 during the prune step because each subset of the candidates is also frequent.

Apriori Algorithm

4. Next, the transactions in D are scanned and the support count of each candidate itemset in C_2 is accumulated, as shown in the middle table of the second row in Figure 6.2.

5. The set of frequent 2-itemsets, L_2, is then determined, consisting of those candidate 2-itemsets in C_2 having minimum support.

6. The generation of the set of the candidate 3-itemsets, C_3, is detailed in Figure 6.3. From the join step, we first get $C_3 = L_2 \bowtie L_2 = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$. Based on the Apriori property that all subsets of a frequent itemset must also be frequent, we can determine that the four latter candidates cannot possibly be frequent. We therefore remove them from C_3, thereby saving the effort of unnecessarily obtaining their counts during the subsequent scan of D to determine L_3. Note that when given a candidate k-itemset, we only need to check if its $(k-1)$-subsets are frequent since the Apriori algorithm uses a level-wise

(a) Join: $C_3 = L_2 \bowtie L_2 = \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\}$

$\bowtie \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\}$

$= \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}.$

(b) Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?

- The 2-item subsets of {I1, I2, I3} are {I1, I2}, {I1, I3}, and {I2, I3}. All 2-item subsets of {I1, I2, I3} are members of L_2. Therefore, keep {I1, I2, I3} in C_3.
- The 2-item subsets of {I1, I2, I5} are {I1, I2}, {I1, I5}, and {I2, I5}. All 2-item subsets of {I1, I2, I5} are members of L_2. Therefore, keep {I1, I2, I5} in C_3.
- The 2-item subsets of {I1, I3, I5} are {I1, I3}, {I1, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I1, I3, I5} from C_3.
- The 2-item subsets of {I2, I3, I4} are {I2, I3}, {I2, I4}, and {I3, I4}. {I3, I4} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I4} from C_3.
- The 2-item subsets of {I2, I3, I5} are {I2, I3}, {I2, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I5} from C_3.
- The 2-item subsets of {I2, I4, I5} are {I2, I4}, {I2, I5}, and {I4, I5}. {I4, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I4, I5} from C_3.

(c) Therefore, $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}$ after pruning.

Transaction Database D

TID	items
T1	I1, I2 , I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

Confidence

6.2.2 Generating Association Rules from Frequent Itemsets

Once the frequent itemsets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence). This can be done using the following equation for confidence, where the conditional probability is expressed in terms of itemset support count:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)},$$

where $support\_count(A \cup B)$ is the number of transactions containing the itemsets $A \cup B$, and $support\_count(A)$ is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "$s \Rightarrow (l - s)$" if $\frac{support\_count(l)}{support\_count(s)} \geq min\_conf$, where min_conf is the minimum confidence threshold.

Since the rules are generated from frequent itemsets, each one automatically satisfies minimum support. Frequent itemsets can be stored ahead of time in has

2 Let's try an example based on the transactional data for *AllElectronics* shown in Figure 6.2. Suppose the data contain the frequent itemset $l = \{I1, I2, I5\}$. What are the association rules that can be generated from l? The nonempty subsets of l are $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, and $\{I5\}$. The resulting association rules are as shown below, each listed with its confidence:

$I1 \wedge I2 \Rightarrow I5$, *confidence* $= 2/4 = 50\%$
$I1 \wedge I5 \Rightarrow I2$, *confidence* $= 2/2 = 100\%$
$I2 \wedge I5 \Rightarrow I1$, *confidence* $= 2/2 = 100\%$
$I1 \Rightarrow I2 \wedge I5$, *confidence* $= 2/6 = 33\%$
$I2 \Rightarrow I1 \wedge I5$, *confidence* $= 2/7 = 29\%$
$I5 \Rightarrow I1 \wedge I2$, *confidence* $= 2/2 = 100\%$

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, since these are the only ones generated that are strong.

6.2.3 Im