Outline

- Basics
- Market Basket Analysis: A Motivating Example
- Preliminaries
- Frequent Itemset Mining
- Apriori Algorithm
- Frequent Pattern growth (FP-growth) Algorithm
- Frequent Closed Itemset Mining
- Frequent Closed Itemset Mining from High Dimensional Data

Basics

- Frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently.
- 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. If a substructure occurs frequently, it is called a frequent structured pattern.

Market Basket Analysis: A Motivating Example

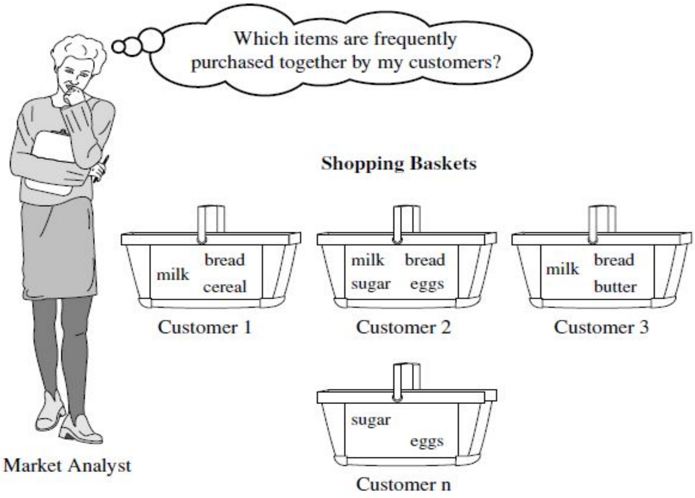

Figure 1. Apriori

Market Basket Analysis

- Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.

- A typical example of frequent itemset mining is market basket analysis.

- Data: collection of transactions of customers.

- Goal: find sets of products frequently occurring together.

- The discovery of associations helps in many business decision making processes, such as catalog design and customer shopping behavior analysis.

Applications

- Market basket analysis.
- Catalog design.
- Customer shopping behavior analysis.
- Web log analysis.
- DNA sequence analysis.
- Sale campaign analysis.
- Software bug detection.
- Chemical Compound Prediction.
- Text analysis.

Preliminaries

Let the Dataset *D* consist of *m* number of transactions (rows) and *n* of attributes or products (features)

- R $= \{r_1, r_2, \ldots r_m\}$
- F $= \{f_1, f_2, \ldots, f_n\}$
- Each row r_i has unique row identifier, *rid* and consist of set of products (features).
- A non-empty subset of features $X \subseteq$ F is defined as an itemset.
- Let $r(f_j)$ signify the rows in which j^{th} feature of the dataset is present.
- A non-empty subset of rids $Y \subseteq$ R is defined as rowset.
- Let $f(r_i)$ signify the features present in the i^{th} row of the dataset.

Preliminaries

Example 1

Table 1 shows an example of a Dataset *D* consisting of 8 rows, where each row is described with unique row identifier (*rid*), R= $\{1, 2, 3, 4, 5, 6, 7, 8\}$ and 11 features, F= $\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}\}$.

Table 1

Dataset *D*

row id (*rid*)	features
1	$f_1, f_2, f_4, f_6, f_{10}$
2	f_1, f_2, f_4, f_7, f_8
3	f_2, f_4, f_7, f_8
4	$f_1, f_2, f_6, f_8, f_9, f_{10}$
5	$f_1, f_3, f_4, f_7, f_8, f_{10}$
6	f_2, f_4, f_9
7	f_5, f_7
8	f_5, f_{11}

Preliminaries

Definition 1 (Support)

The number of rows in which an itemset X occurs is called the support of an itemset, denoted by *sup(X)*.

Example 2

In Table 1, the support of an itemset $X = \{f_2, f_4, f_7, f_8\}$, *sup(X)* is 2.

Definition 2 (Support Set)

The rows in which an itemset X occurs is called support set of an itemset, denoted by *supset(X)*.

Example 3

In Table 1, the support set of an itemset $X = \{f_2, f_4, f_7, f_8\}$, *supset(X)* is *23*.

Preliminaries

Definition 3 (Cardinality)

The number of items in an itemset X is called as the cardinality of an itemset, denoted by *card(X)*.

Example 4

In Table 1, the cardinality of an itemset $X = \{f_2, f_4, f_7, f_8\}$, *card(X)* is 4.

Definition 4 (Frequent Itemset)

An itemset X is called frequent itemset if and only if *sup(X)* $\geq$ *minsup*, where *minsup* is user specified least support threshold.

Example 5

In Table 1, the itemset $X = \{f_2, f_8\}$ is frequent itemset with minimum support threshold set to 2, *sup(X)* ≥ 2.

Preliminaries

Definition 5 (Association Rule)

Let A and B be the set of items. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset$ F, $B \subset$ F and $A \cap B = \emptyset$. The association rule $A \Rightarrow B$ holds in the dataset with **support** s and has **confidence** c.

Support s, is the percentage of transactions in D that contain A ∪ B (i.e., the union of sets A and B, or say, both A and B).

Confidence c, is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, P(B|A).

$$support(A \Rightarrow B) = P(A \cup B) \tag{1}$$

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} \tag{2}$$

Frequent Itemset Mining

Table 2
Dataset D

TID	Items Bought
1	Beer, Nuts, Chips
2	Beer, Coffee, Chips
3	Beer, Chips, Eggs
4	Nuts, Eggs, Milk
5	Nuts, Coffee, Chips, Eggs, Milk

- Problem: To Mine the Frequent Itemsets with minimum support threshold (*minsup*) set to 50% and minimum confidence threshold (*minconf*) set to 50%.
- Frequent Itemsets are: Beer:3, Nuts:3, Chips:4, Eggs:3, {Beer, Chips}:3.
- Example of association rules
  Beer $\rightarrow$ Chips (60%, 100%).
  Chips $\rightarrow$ Beer (60%, 75%).

Frequent Itemset Mining

- Frequent Itemset Mining Algorithms
  - Apriori Algorithm
  - Frequent Pattern growth (FP-growth) algorithm
- Frequent Closed Itemset Mining Algorithms
- Frequent Maximal Itemset Mining Algorithms
- Frequent Colossal Itemset Mining Algorithms
- Frequent Colossal Closed Itemset Mining Algorithms

Apriori Algorithm

- Apriori is an algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets from transactional datasets for generating association rules.
- The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.
- 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.

Apriori Algorithm

- The resulting set is denoted L_1. Next, L_1 is used to find L_2, the set of frequent 2-itemsets, which is used to find L_3, and so on, until no more frequent k-itemsets can be found.
- The finding of each L_k requires one full scan of the database.
- To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property.
- Apriori property: All nonempty subsets of a frequent itemset must also be frequent.
- The property belongs to a special category of properties called antimonotone in the sense that if a set cannot pass a test, all of its supersets will fail the same test as well.

Apriori Algorithm

- Apriori Algorithm has two steps
  - The Join step
  - The Prune step
- **The Join step:**
  - To find L_k, a set of candidate k-itemsets is generated by joining L_{k-1} with itself. This set of candidates is denoted C_k.
  - Apriori assumes that items within a transaction or itemset are sorted in lexicographic order.
  - The join, $L_{k-1} \bowtie L_{k-1}$, is performed, where members of L_{k-1} are joinable if their first (k-2) items are in common.

Apriori Algorithm

- **The Prune step:**
  - C_k is a superset of L_k, that is, its members may or may not be frequent, but all of the frequent k-itemsets are included in C_k.
  - A scan of the database to determine the count of each candidate in C_k would result in the determination of L_k.
  - Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset.
  - If any (k-1)-subset of a candidate k-itemset is not in L_{k-1}, then the candidate cannot be frequent either and so can be removed from C_k.

Apriori Algorithm

Table 3
Dataset D

TID	List of items
1	I1, I2, I5
2	I2, I4
3	I2, I3
4	I1, I2, I4
5	I1, I3
6	I2, I3
7	I1, I3
8	I1, I2, I3, I5
9	I1, I2, I3

Apriori Algorithm

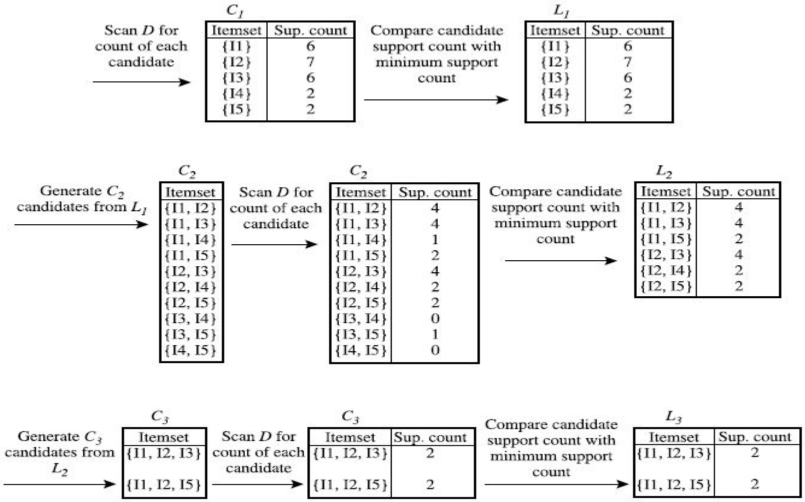

Figure 2. Steps Apriori Algorithm

Apriori Algorithm

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

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.

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

Apriori Algorithm

- Generating association rules. The frequent itemset considered is {I1, I2, I5}
- The nonempty subsets of frequent itemset 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 $= 2/2 = 50\%$

Apriori Algorithm

$Sup_{min} = 2$

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1^{st} scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2^{nd} scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3^{rd} scan

L_3

Itemset	sup
{B, C, E}	2