Lec 3:
Mining Association Rules

T.Heba AL.Marwai

Association rule mining

- Proposed by Agrawal et al in 1993.
- It is an important data mining model studied extensively by the database and data mining community.
- Assume all data are categorical.
- No good algorithm for numeric data.
- Initially used for Market Basket Analysis to find how items purchased by customers are related.

Bread $\rightarrow$ Milk [sup = 5%, conf = 100%]

The model: data

- $I = \{i_1, i_2, \ldots, i_m\}$: a set of *items*.
- Transaction *t* :
  - *t* a set of items, and $t \subseteq I$.
- Transaction Database *T*: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$.

Transaction data: supermarket data

- Market basket transactions:

  t1: {bread, cheese, milk}

  t2: {apple, eggs, salt, yogurt}

  … …

  tn: {biscuit, eggs, milk}

- Concepts:
  - An *item*: an item/article in a basket
  - *I*: the set of all items sold in the store
  - A *transaction*: items purchased in a basket; it may have TID (transaction ID)
  - A *transactional dataset*: A set of transactions

Transaction data: a set of documents

- **A text document data set. Each document is treated as a “bag” of keywords**

doc1:	Student, Teach, School
doc2:	Student, School
doc3:	Teach, School, City, Game
doc4:	Baseball, Basketball
doc5:	Basketball, Player, Spectator
doc6:	Baseball, Coach, Game, Team
doc7:	Basketball, Team, City, Game

The model: rules

- An association rule is an implication of the form:

  $$X \to Y, \text{ where } X, Y \subset I, \text{ and } X \cap Y = \emptyset$$

- An itemset is a set of items.
  - E.g., X = {milk, bread, cereal} is an itemset.
- A *k*-itemset is an itemset with *k* items.
  - E.g., {milk, bread, cereal} is a 3-itemset

Rule strength measures

- Support: The rule holds with support *sup* in *T* (the transaction data set) if sup% of transactions contain $X \cup Y$.
  - $sup = \Pr(X \cup Y)$.
- Confidence: The rule holds in *T* with confidence *conf* if *conf*% of tranactions that contain *X* also contain *Y*.
  - $conf = \Pr(Y \mid X)$
- An association rule is a pattern that states when *X* occurs, *Y* occurs with certain probability.

Support and Confidence

- Support count: The support count of an itemset *X*, denoted by *X.count*, in a data set *T* is the number of transactions in *T* that contain *X*. Assume *T* has *n* transactions.
- Then,

$$support = \frac{(X \cup Y).count}{n}$$

$$confidence = \frac{(X \cup Y).count}{X.count}$$

An example

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

- Transaction data
- Assume:

    minsup = 30%
    minconf = 80%

- An example frequent *itemset*:

{Chicken, Clothes, Milk}   [sup = 3/7]

- Association rules from the itemset:

Clothes $\rightarrow$ Milk, Chicken   [sup = 3/7, conf = 3/3]

…   …

Clothes, Chicken $\rightarrow$ Milk,   [sup = 3/7, conf = 3/3]

Transaction data representation

- A simplistic view of shopping baskets,
- Some important information not considered. E.g,
  - the quantity of each item purchased and
  - the price paid.

Many mining algorithms

- There are a large number of them!!
- They use different strategies and data structures.
- Their resulting sets of rules are all the same.
  - Given a transaction data set *T*, and a minimum support and a minimum confident, the set of association rules existing in *T* is uniquely determined.
- Any algorithm should find the same set of rules although their computational efficiencies and memory requirements may be different.
- We study only one: the Apriori Algorithm

The Apriori algorithm

- **Probably the best known algorithm**
- **Two steps**:
  - Find all itemsets that have minimum support (*frequent itemsets*, also called large itemsets).
  - Use frequent itemsets to generate rules.

- E.g., a frequent itemset

  {Chicken, Clothes, Milk} [sup = 3/7]

  and one rule from the frequent itemset

  Clothes $\rightarrow$ Milk, Chicken [sup = 3/7, conf = 3/3]

Step 1 : generate frequent itemsets

- The candidate-gen function takes F_{k-1} and returns a superset (called the candidates) of the set of all frequent k-itemsets. It has two steps
  - *join* step: Generate all possible candidate itemsets C_k of length k
  - *prune* step: Remove those candidates in C_k that cannot be frequent.

Mining Frequent Itemsets

- Find the *frequent itemsets*: the sets of items that have minimum support
  - A subset of a frequent itemset must also be a frequent itemset
    - i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
  - Iteratively find frequent itemsets with cardinality from 1 to *k* (*k*-itemset)
- Use the frequent itemsets to generate association rules.

The Apriori Algorithm — Example

- Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

- Scan D

- C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

- L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

- C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

- Scan D

- C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

- L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

- C_3

itemset
{2 3 5}

- Scan D

- L_3

itemset	sup
{2 3 5}	2

Step 2: Generating rules from frequent itemsets

- Frequent itemsets ≠ association rules
- One more step is needed to generate association rules
- For each frequent itemset *X*,

  For each proper nonempty subset *A* of *X*,
  - Let *B* = X - *A*
  - A $\rightarrow$ B is an association rule if
    - Confidence(A $\rightarrow$ B) ≥ minconf,

      support(A $\rightarrow$ B) = support(A$\cup$B) = support(X)

      confidence(A $\rightarrow$ B) = support(A $\cup$ B) / support(A)

Generating rules: an example

- Suppose {2,3,5} is frequent, with sup=50%
  - Proper nonempty subsets: {2,3}, {2,5}, {3,5}, {2}, {3}, {5}, with sup=50%, 75%, 50%, 75%, 75%, 75% respectively
  - These generate these association rules:
    - 2,3 $\rightarrow$ 5 confidence=100%
    - 2,5 $\rightarrow$ 3, confidence=67%
    - 3,5 $\rightarrow$ 2, confidence=100%
    - 2 $\rightarrow$ 3,5, confidence=67%
    - 3 $\rightarrow$ 2,5, confidence=67%
    - 5 $\rightarrow$ 2,3, confidence=67%
    - All rules have support = 50%

Similarity and Distance

- For many different problems we need to quantify how close two objects are.
- Examples:
  - For an item bought by a customer, find other similar items
  - Group together the customers of site so that similar customers are shown the same ad.
  - Group together web documents so that you can separate the ones that talk about politics and the ones that talk about sports.
  - Find all the near-duplicate mirrored web documents.
  - Find credit card transactions that are very different from previous transactions.
- To solve these problems we need a definition of similarity, or distance.
  - The definition depends on the type of data that we have

Similarity and Distance

Similarity

- Numerical measure of how alike two data objects are.
  - A function that maps pairs of objects to real values
  - Higher when objects are more alike.
- Often falls in the range [0,1], sometimes in [-1,1]

- Desirable properties for similarity
  1. s(p, q) = 1 (or maximum similarity) only if p = q. (Identity)
  2. s(p, q) = s(q, p)   for all p and q. (Symmetry)

Similarity between sets

- Consider the following documents

apple releases new ipod	apple releases new ipad	new apple pie recipe

- Which ones are more similar?

- How would you quantify their similarity?

Similarity: Intersection

- Number of words in common

apple releases new ipod	apple releases new ipad	new apple pie recipe

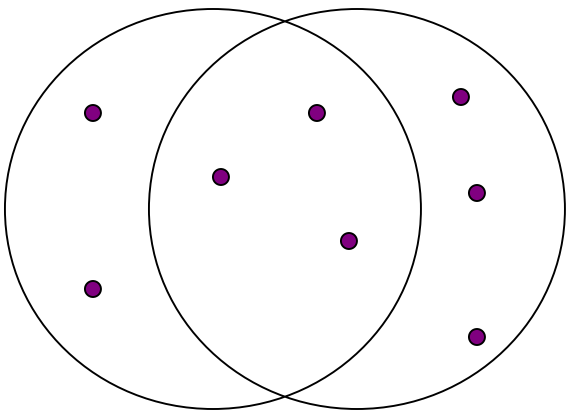
- Sim(D,D) = 3, Sim(D,D) = Sim(D,D) =2
- What about this document?

Vefa rereases new book with apple pie recipes

- Sim(D,D) = Sim(D,D) = 3

Jaccard Similarity

- The Jaccard similarity (Jaccard coefficient) of two sets S_1, S_2 is the size of their intersection divided by the size of their union.
  - JSim $(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.



3 in intersection.
8 in union.
Jaccard similarity = 3/8

  - Extreme behavior:
    - Jsim(X,Y) = 1, iff X = Y
    - Jsim(X,Y) = 0 iff X,Y have not elements in common
  - JSim is symmetric

Similarity: Intersection

- Number of words in common

apple releases new ipod	apple releases new ipad	new apple pie recipe	Vefa rereases new book with apple pie recipes

- JSim(D,D) = 3/5
- JSim(D,D) = JSim(D,D) = 2/6
- JSim(D,D) = JSim(D,D) = 3/9

Similarity between vectors

Documents (and sets in general) can also be represented as vectors

document	Apple	Microsoft	Obama	Election
D1	10	20	0	0
D2	30	60	0	0
D2	0	0	10	20

How do we measure the similarity of two vectors?

How well are the two vectors aligned?

Example

document	Apple	Microsoft	Obama	Election
D1	1/3	2/3	0	0
D2	1/3	2/3	0	0
D2	0	0	1/3	2/3

Documents D1, D2 are in the “same direction”
Document D3 is orthogonal to these two

Cosine Similarity

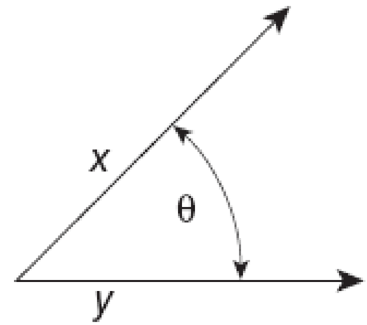


Figure 2.16. Geometric illustration of the cosine measure.

- Sim(X,Y) = cos(X,Y)
  - The cosine of the angle between X and Y

- If the vectors are aligned (correlated) angle is zero degrees and cos(X,Y)=1
- If the vectors are orthogonal (no common coordinates) angle is 90 degrees and cos(X,Y) = 0

- Cosine is commonly used for comparing documents, where we assume that the vectors are normalized by the document length.

Cosine Similarity - math

- If d_1 and d_2 are two vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1||\ ||d_2||\ ,$$

where $\bullet$ indicates vector dot product and $|| d ||$ is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**
d_2 = **1 0 0 0 0 0 0 1 0 2**

$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$

$||d_1|| = (3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5} = (42)^{0.5} = 6.481$

$||d_2|| = (1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5} = (6)^{0.5} = 2.245$

$\cos(d_1, d_2) = .3150$

Distance

- Numerical measure of how different two data objects are
  - A function that maps pairs of objects to real values
  - Lower when objects are more alike
- Minimum distance is 0, when comparing an object with itself.
- Upper limit varies

Distance Metric

- A distance function d is a distance metric if it is a function from pairs of objects to real numbers such that:
    1. $d(x,y) \geq 0$. (non-negativity)
    2. $d(x,y) = 0$ iff $x = y$. (identity)
    3. $d(x,y) = d(y,x)$. (symmetry)
    4. $d(x,y) \leq d(x,z) + d(z,y)$ (triangle inequality).