Applied Data Mining

Rockhurst University

Frequent pattern analysis (Association rules)

Section 1

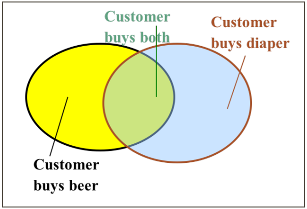
Frequent pattern analysis

What is it?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
    - What products were often purchased together?? Beer and diapers?!
    - What are the subsequent purchases after buying a PC?
    - What kinds of DNA are sensitive to this new drug?
    - Can we automatically classify web documents?.

Setup

What are things that occur together?



Data:

ID	Items
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk

Basic concepts: frequent patterns

- **itemset**: A set of one or more items
- **k-itemset**: $X = x_1, \ldots, x_k$
- **(absolute) support**, or, **support count**, σ of X: Frequency or occurrence of an itemset X (e.g. σ(Bread, Milk, Diaper) $= 2$)
- **(relative) support**, s, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X) (e.g. s(Milk, Bread, Diaper) $= \frac{\sigma(Milk, Diaper)}{T} = 2/5$)
- **Frequent itemset**: An itemset X that has support above a minimum threshold
- **Minsup**: The minimum threshold that defines a frequent itemset.

Basic concepts: Association rules

A rule states that when we see X, we expect to also see Y. It is written as $X \rightarrow Y$ For example, $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

Goal: to find rules $X \rightarrow Y$ with a minimum level of support and confidence.

- **support**: s, probability that a transaction contains X and Y - $P(X \cup Y)$
- **confidence**: c, conditional probability that a transaction having X also contains Y - $P(Y|X)$

Example: $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

Support $= s(Milk, Bread, Diaper) = 2/5$

Confidence $= \frac{\sigma(Milk, Diaper, Beer)}{\sigma(Milk, Diaper)} = 2/3$

Rules with high confidence are **strong** rules.

Evaluating rules

Lets consider the rule A → B in order to compute some metrics.

$$Support = \frac{Number\ \ of\ \ transactions\ \ with\ \ both\ \ A\ \ and\ \ B}{Total\ \ number\ \ of\ \ transactions} = P\left(A \cap B\right)$$

$$Confidence = \frac{Number\ \ of\ \ transactions\ \ with\ \ both\ \ A\ \ and\ \ B}{Total\ \ number\ \ of\ \ transactions\ \ with\ \ A} = \frac{P(A \cap B)}{P(A)}$$

$$ExpectedConfidence = \frac{Number\ \ of\ \ transactions\ \ with\ \ B}{Total\ \ number\ \ of\ \ transactions} = P\left(B\right)$$

$$Lift = \frac{Confidence}{Expected\ \ Confidence} = \frac{P(A \cap B)}{P(A).P(B)}$$

Lift is the factor by which, the co-occurence of A and B exceeds the expected probability of A and B co-occuring, had they been independent. So, higher the lift, higher the chance of A and B occurring together.

http://r-statistics.co/Association-Mining-With-R.html

Process

If we have T transactions, we want to find patterns that are *frequent* and meet some minimum support and confidence.

How do we find them? We could use the brute force approach:

- Find all possible association rules
- Compute the support and confidence for each rule
- Prune rules that fail the minsup and minconf thresholds

Downward closure property

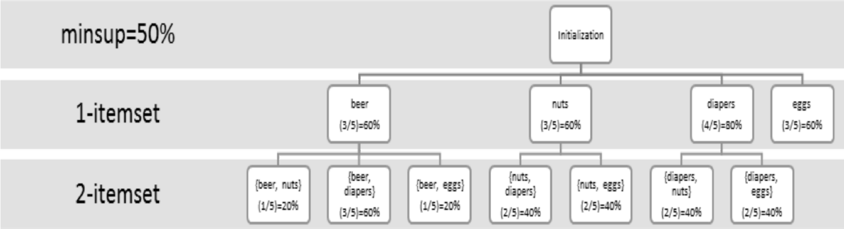
Sounds very simple: Downward closure says any subset of a frequentitemset must be frequent.

Remember a frequent itemset is one that has support above a minimum threshold (minsup).

If {Milk, Diapers, Beer} is frequent, so is {Beer, Diapers} because every transaction having {Milk, Diapers, Beer} also contains {Diapers, Beer}

In a picture

What are things that occur together?



Notice that no 3-itemset candidates are generated because only one 2-itemset beer, diapers is frequent.

Method 1: Apriori

Initially, scan data once to get frequent 1-itemset Generate length (k + 1) candidate itemsets from length k frequent itemsets Test the candidates against data. Prune candidate itemsets based on minimum support threshold Terminate when no frequent or candidate set can be generated. NOTE: Requires data to be in a horizontal format. Remember: these are the data in a horizontal format:

ID	**Items**
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk

Apriori example: First scan

If minsup = .5, let's see what qualifes as frequent.
First step: Generate candidate list for 1-itemset:

Itemset	**Support**
{Beer}	3 (60%)
{Nuts}	3 (60%)
{Diapers}	4 (80%)
{Coffee}	2 (40%)
{Eggs}	3 (60%)
{Milk}	2 (40%)

Second step: prune to meet the minsup = .5 requirement:

Itemset	**Support**
{Beer}	3 (60%)
{Nuts}	3 (60%)
{Diapers}	4 (80%)
{Eggs}	3 (60%)

Apriori example: second scan

First step: Generate candidate list for 2-itemset. Notice that we only consider sets with Beer, Nuts, Diapers, and Eggs because Milk and Cofee aren't considered frequent.

Itemset	Support	Itemset	Support
{Beer,Nuts}	1 (20%)	{Nuts, Diapers}	2 (40%)
{Beer, Diapers}	3 (60%)	{Nuts, Eggs}	2 (40%)
{Beer, Eggs}	1 (20%)	{Diapers, Eggs}	2 (40%)

Second step: prune to meet the minsup = .5 requirement:

Itemset	Support
{Beer, Diapers}	3 (60%)

And we are done. Note this approach is scalable to very large datasets but is computationally intensive. It can be very slow because we have to keep comparing the candidate itemsets against the database until no frequent and/or candidate itemsets can be generated.

ECLAT example

Requires data to be in a different format:

Beer	Nuts	Diapers	Coffee	Eggs	Milk
10	10	10	20	30	40
20	40	20	50	40	50
30	50	30		50	
		50			

Support of a 1-itemset is the tidset Support of a k-itemset is the intersection of corresponding itemsets

ECLAT method

Support for {Beer, Diapers} is counted by matching the tidsets of beer and diapers.

Beer $\in \{10; 20; 30\}$. Diapers $\in \{10; 20; 30; 50\}$. So Beer and Diapers $\in \{10; 20; 30\}$ Method:

- Generate 1-itemset Candidate and Count Support at the Same Time
- Prune candidates based on minsup threshold
- Repeat Steps 1 and 2 until no more candidates can be generated or no frequent itemset is found.