Descubrimiento de Conocimiento a partir de Datos

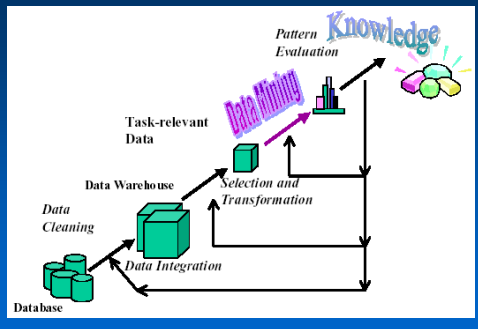
Prof. Dra. Silvia Schiaffino
ISISTAN

sschia@exa.unicen.edu.ar

Agenda

- Introducción al KDD
- Etapas
- Pre-procesamiento de datos
- Operaciones de Data Mining
  - Reglas de Asociación
  - Clasificación y Predicción
  - Clustering
- Web Mining
- Social Mining

Proceso de KDD



Introduction

- Association Rules is an artificial intelligence technique widely used in Data Mining
- Data Mining is the process of finding interesting **trends** or **patterns** in large datasets in order to guide future decisions.
- Data Mining is the discovery of knowledge and useful information from the large amounts of data stored in databases.

Examples

Students taking courses *X* tend to take course *Y*

Clients purchasing products *X* tend to purchase product *Y*

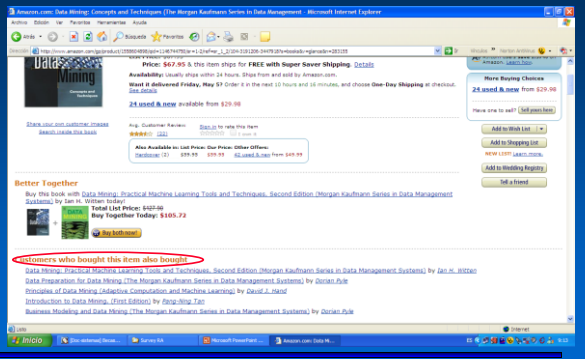
Papers referring to papers *X* tend to refer to paper *Y*

Some real problems

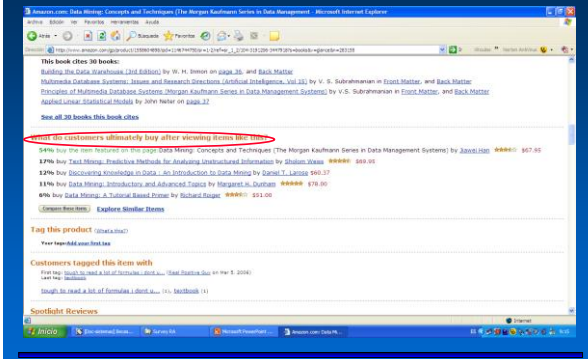
- Associations between products bought in a store
  - milk and bread
  - beer and diapers
- Deciding on product discounts and sales
- Placing goods in stands to maximize profits
- Personal recommendation page at Amazon (books, DVDs)
- Sites or Web pages a user visits in the same session

Association Rules: describing association relationships among the items in the set of relevant data.

Example: Amazon



Example: Amazon



Market Basket Analysis

A **market basket** is a collection of items bought by a single customer in a single customer transaction

Problem: Identify sets of items that are purchased together

Attempt to identify rules of the form

{pen} ➔ {ink}

Meaning: If a pen is bought in a transaction, it is likely that ink will also be bought

Association Rules

General form: L➔R, where L and R are sets of items

L is the antecedent of the rule and R its consequent

Support: The support for L➔R is the percentage of transactions containing all items from L and from R

Confidence: The confidence for L➔R is the percentage of transactions that contain R, from among the transactions that contain L

Support - Confidence

$$\text{Support}(L \rightarrow R) = \text{Prob}(L \cup R) = \text{Support}(L \cup R)$$

$$\text{Confidence}(L \rightarrow R) = \text{Prob}(R / L) = \frac{\text{Support}(L \cup R)}{\text{Support}(L)}$$

Understanding the measures

- If a rule has low support then it might have arisen by chance. There is not enough evidence to draw a conclusion.
- If a rule L➔R has low confidence then it is likely that there is no relationship between buying L and buying R

Note: L➔R and R➔L will always have the same support, but may have different confidence

Different types of association rules

- Boolean vs. Quantitative Associations
  - Based on the type of values handled
  - Buys(X, "SQL Server") ∧ Buys(X, "DMBook") → Buys(X, "DBMiner") [0.2%, 60%]
  - Age(X, 30..39) ∧ Income(X, 4k..8K) → Buys (X, "PC") [1%, 75%]

Boolean Association Rules

Quantitative Association Rules

Different types of association rules

- Single Dimension vs. Multiple dimensional associations
  - Based on the dimension in data involved
  - One predicate then single dimension. More predicates then multi dimensions
  - Ex: Buys(X, Bread) → Buys (X, milk)
    - Age(X, 30-45) ∧ Income(X, 50k-70k) → Buys (X, Car)

Single-dimensional Association Rules

Multi-dimensional Association Rules

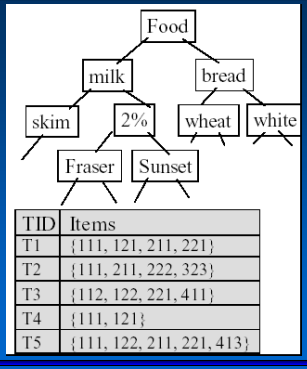
Different types of association rules

- Single level vs. multiple level associations
  - Based on the level of abstraction involved (Ex. brands of products)
  - Find association rules at different levels of abstraction
  - Ex: Buys(X, Bread) → Buys (X, milk)
    - Buys(X, White Bread) → Buys (X, Nido milk)

Single-level Association Rules

Multi-level Association Rules

Multi-level association rules



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

Different types of association rules

- Single occurrence vs. multiple occurrences
  - One item may occur more than once in the transaction
  - The presence of the item is not important but its frequency
  - Ex: Buys(X, Bread, 2) → Buys (X, milk, 1)

Single-occurrence-items Association Rules

Recurrent-items Association Rules

- Single vs. constraint-based
  - Constraints can be added on the rules to be discovered

Goal

Find association rules with high support and high confidence.

Support and confidence values are specified by the user (minsup and minconf thresholds).

Remember: Finding such a rule does not mean that there **must** be a relationship between the left and right sides. A person must evaluate such rules by hand.

Formal statement of the problem

$I = \{ i_1 , i_2 , \ldots , i_m \}$ is a set of items

D is a set of transactions T

Each transaction T is a set of items (subset of I)

Example

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

I = {A, B, C, D, E }

D = {1, 2, 3, 4, 5 } = {{A,C,D}, {B,C,E}, {A,B,C,E}, {B,E,}, {A,B,C,E}}

Itemsets

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

Let X be a set of items, $X \subseteq I$.

X is an **itemset**.

An itemset containing k items is called **k-itemset**.

e.g. {A,B} is a 2-itemset

The support of an itemset X is the percentage of transactions in D containing X:

Support(X) = $|\{T \in D / X \subseteq T\}| / |D|$ = Support ({A,B}) = 2/5 = 0.40

Problem decomposition

The problem can be decomposed into two sub-problems:

1. Find all sets of items (*itemsets*) that have support (number of transactions) greater than the minimum support (*large or frequent itemsets*).
2. Use the *large itemsets* to generate the desired rules.

   For each *large itemset* l, find all non-empty subsets, and for each subset a generate a rule a ➔ (l-a) if its confidence is greater than the minimum confidence.

Mining Algorithms

- Apriori y AprioriTid [Agrawal R. & R. Srikant (1994)];
- Opus [Webb G. I. (1996)];
- Direct Hasing and Pruning (DHP) [Adamo J.M.(2001)];
- Dynamic Set Counting (DIC) [Adamo J.M. (2001)];
- Charm [Zaki M. & C. Hsiao (2002)];
- FP-growth [J. Han, J. Pei & Y. Yin (1999)];
- Closet [Pei J., J. Han & R. Mao (2000)];

Mining Algorithms

- The different algorithms must always generate the same knowledge
- What makes them different?
  - The way in which data is loaded into memory
  - Processing time
  - Attribute types (Numerical, Categorical)
  - The way in which itemsets are generated
  - Data structures used

Apriori Algorithm

1. In the first pass, the support of each individual item is counted, and the *large* ones are determined
2. In each subsequent pass, the *large* itemsets determined in the previous pass are used to generate new itemsets called *candidate* itemsets.
3. The support of each *candidate* itemset is counted, and the *large* ones are determined.
4. This process continues until no new *large* itemsets are found.

Finding Frequent Itemsets

Freq = {}
scan all transactions once and add to Freq the items that
  have support > minsup
k = 1
repeat
  foreach l_k in Freq with *k* items
    generate all itemsets l_{k+1} with *k+1* items,
      such that l_k is contained in l_{k+1}
    scan all transactions once and add to Freq the
      k+1-itemsets that have support > minsup
    k++
until no new frequent itemsets are found

Example

Minsup= 0.5

Database

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

L_1

Itemset	Support
{1}	2/4
{2}	¾
{3}	¾
{5}	¾

C_2

Itemset	Support
{1 2}	¼
{1 3}*	2/4
{1 5}	¼
{2 3}*	2/4
{2 5}*	¾
{3 5}*	2/4

C_3

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

L_3

Itemset	Support
{2 3 5}	2/4

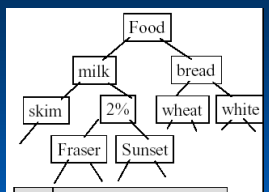
Deriving Association Rules

To generate rules, for every large itemset l, we find all non-empty subsets of l. For every such subset a, we output a rule of the form a➔(l-a) if the ratio of support(l) to support(a) is at least *minconf*.

We can improve the above procedure by generating the subsets of a large itemset in a recursive depth-first fashion.

Multi-level Association Rules

- Items often form a hierarchy
- Items at the lower level are expected to have lower support
- Rules regarding itemsets at appropriate levels could be quite useful
- Transaction database can be encoded based on dimensions and levels



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

Post-processing of rules

Numerous association rules are generated in a mining process

Some of the mined rules may be trivial facts...

"Pregnant" → "Female", Supp=20%, Conf=100%

... while some other rules may be redundant

"Drive fast" → "Had an accident", Supp=10%, Conf=40%

"Drive fast" and "Born in HK" → "Had an accident", Supp=9%, Conf=42%

Post-processing tasks

- Filtering out uninteresting rules
- Filtering out insignificant rules
- Eliminating redundant rules
- Summarizing the remaining rules

Filtering uninteresting rules

The study of "interestingness" of association rules aims at presenting those rules that are interesting to the user

Closely related to the study of "surprisingness" or "unexpectedness" of association rules

Interestingness: two approaches

- Objective measures (data-driven)
  - Mined rules are ranked by a pre-defined ranking system, or
  - Mined rules are filtered by a set of pre-defined pruning rules
- Subjective measures (user-driven)
  - Users are required to specify whether the mined rules are interesting...
    - But it is impossible to do so rule by rule
    - Hence rules are handled collectively

Objective Measures

- Lift (a.k.a interest and strength) [Brin97] [Dhar93]
- Conviction [Brin97]
- Gain [Fukuda96]
- Chi-squared value [Morimoto98]
- Laplace [Webb95]
- Gini [Morimoto98]
- Piatetsky-Shapiro [Piatetsky-Shapiro91]
- Entropy gain [Morimoto98]

Problem of support-confidence framework

P(X): probability that a transaction T from database D contains the itemset X

P(X,Y): probability that both X and Y are contained in T in D.

If X and Y are stochastically independent:

$$P(X)*P(Y)=P(X,Y)$$

Then for the confidence of the rule X ➔ Y follows:

$$conf(X \rightarrow Y)= P(Y/X) = P(Y)$$

Thus, as soon as the itemset Y occurs comparably often in the data the rule X ➔Y also has a high confidence value. This suggests a dependency of Y from X although in fact both itemsets are stochastically independent.

Lift (Interest, Strength)

Lift directly addresses the problem presented before by expressing the deviation of the rule confidence from P(Y). In the case of stochastic independence lift=1 holds true. In contrast, a value higher than 1 means that the existence of X as part of a transaction "lifts" the probability for this transaction to also contain Y by factor lift. The opposite is true for lift values lower than one.

$$lift(X \rightarrow Y) = conf(X \rightarrow Y) / P(Y)$$

$$=conf(X \rightarrow Y) / sup(Y)$$

Conviction

P(¬Y): probability of a transaction T in D with Y ⊄ T

P(X,¬Y): probability of drawing a transaction out of D that contains X but not Y.

conviction(X ➔Y) now expresses in how far X and ¬Y are stochastically independent.

High values for conviction(X ➔Y) (up to ∞ where P(X, ¬Y)=0) express the conviction that this rule represents a causation.

conviction(X ➔Y)=P(X)P(¬Y) / P(X, ¬Y)

conviction(X ➔Y)=|D|-sup(Y) / |D| (1-conf(X ➔Y)

Subjective Measures

Interesting and uninteresting rules can be specified with templates [Klementinen et al. 94]

A rule template specifies what attributes to occur in the antecedent and consequent of a rule

e.g. Any rule of the form "Pregnant" & (any number of conditions) → "Female" is uninteresting

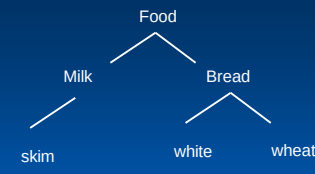
Templates

A template is an expression of the form:

$A_1,...,A_k \rightarrow A_{k+1},..,A_n$

where each A_i is either an attribute name, a class name, or an expression C+ and C*, which correspond to one or more and zero or more instances of the class C, respectively. A rule $B_1,...,B_h \rightarrow B_{h+1},...,B_m$ matches the pattern if the rule can be considered to be an instance of the pattern.

Taxonomies



Food ⊂ Milk ⊂ skim

Food ⊂ Bread ⊂ white

Food ⊂ Bread ⊂ wheat

Food, Milk, Bread: classes

skim, white, wheat: items

- Rules can involve different levels in a taxonomy or is-a hierarchy (generalized or multi-level association rules)

Item Constraints

Constraints are boolean expressions over the presence or absence of items in the rules. When taxonomies are present, the elements of the boolean expression can be ancestors or descendant items, as well as single items.

Constraints are embeded in the association rule mining algorithm. This is more efficient than post-processing

Eliminating redundant rules

Rule 1: drives alone ➔ not veteran (confidence 0.67)

Rule 2: drives alone, born in US ➔ not veteran (confidence 0.72)

If we know R1, then R2 is insignificant because it gives little extra information. Its slightly higher confidence is more likely due to chance than to true correlation. Thus, it should be pruned. R1 is more general and simple. General and simple rules are preferred.

Shah99 Rules

If there are two rules of the form A ➔ C and A, B ➔ C, an either both rules are positive or negative with similar strength, then A, B ➔ C is redundant

A positive rule A $\rightarrow^{+}$ B, is a rule where the presence of A is found to increase the probability of B's occurrence significantly. Formally, this means that for a user-defined coefficient P>1, P(B/A)>P*P(B).

A negative rule, denoted by A $\rightarrow^{-}$ B, where A and B are as before, is a rule where for a coefficient N>1, P(B) > N*P(B/A). In turn, A and B are independent, they must occur often together, i..e P(B)*P(A) > thr.

Two rules are of similar strength if for $1>\varepsilon>0$, $|\text{confidence(R1)-confidence(R2)}| < \varepsilon$

Shah99 Rules

If A ➔ C1 and A ➔ C1, C2, then A ➔ C1 is redundant

Consequent C1 and C2 is stronger than C1 in a logical sense.

Example:

white, US citizen ➔ speaks English well

White, US citizen ➔ speaks English well, moved in past 5 years

Tools and Software

- WEKA: http://www.cs.waikato.ac.nz/ml/weka/
- KNIME: http://www.knime.org/
- TANAGRA: http://eric.univ-lyon2.fr/~ricco/tanagra/
- RapidMiner: http://rapid-i.com/
- Orange: http://www.imcredel.com/open-source/orange
- Kettle: http://kettle.pentaho.com/
- Others, see: http://www.kdnuggets.com/software/suites.html