Requirements/Challenges in Data Mining (Con't)

- Data source issues:
  - ➔ Diversity of data types
    - Handling complex types of data
    - Mining information from heterogeneous databases and global information systems.
    - Is it possible to expect a DM system to perform well on all kinds of data? (distinct algorithms for distinct data sources)
  - ➔ Data glut
    - Are we collecting the right data with the right amount?
    - Distinguish between the data that is important and the data that is not.

Requirements/Challenges in Data Mining (Con't)

- Other issues
  - – Integration of the discovered knowledge with existing knowledge: A knowledge fusion problem.

Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

Basic Concepts

A transaction is a set of items: $T=\{i_a, i_b, \ldots i_t\}$

$T \subset I$, where I is the set of all possible items $\{i_1, i_2, \ldots i_n\}$

D, the task relevant data, is a set of transactions.

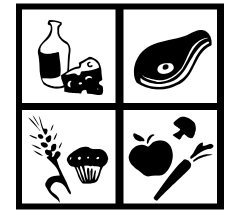
An association rule is of the form:
$P \Rightarrow Q$, where $P \subset I$, $Q \subset I$, and $P \cap Q = \emptyset$

Basic Concepts (con't)

P➔Q holds in *D* with <u>support</u> s
and
P➔Q has a <u>confidence</u> c in the transaction set *D*.

Support(P➔Q) = Probability($P \cup Q$)
Confidence(P➔Q)=Probability(Q / P)

Itemsets



A set of items is referred to as <u>itemset</u>.

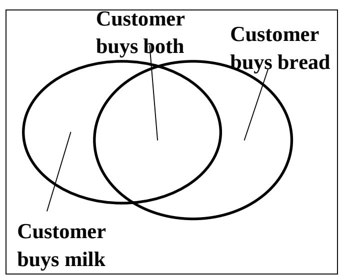
An itemset containing k items is called **k-itemset**.

An items set can also be seen as a conjunction of items (or a predicate)

Rule Measures: Support and Confidence

- ***Support of a rule*** $P \rightarrow Q$
  = Support of ($P \cup Q$) in ***D***

\- $s_D(P \rightarrow Q) = s_D(P \cup Q)$: percentage of transactions in *D* containing *P* and *Q*.
(#transactions containing *P* and *Q* divided by cardinality of *D*).



- ***Confidence of a rule*** $P \rightarrow Q$

\- $c_D(P \rightarrow Q) = s_D(P \cup Q) / s_D(P)$: percentage of transactions that contain both *P* and *Q* in the subset of transactions that contain already *P*.

Strong Rules

- Thresholds:
  - minimum support: *minsup*
  - minimum confidence: *minconf*

- **Frequent itemset** *P*
  - support of *P* larger than minimum support,
- **Strong rule** $P \rightarrow Q$ (c%)
  - ($P \cup Q$) frequent,
  - *c* is larger than minimum confidence.

Mining Association Rules

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $\{A\} \Rightarrow \{C\}$:

support = support($\{A, C\}$) = 50%
confidence = support($\{A, C\}$)/support($\{A\}$) = 66.6%

For rule $\{C\} \Rightarrow \{A\}$:

support = support($\{A, C\}$) = 50%
confidence = support($\{A, C\}$)/support($\{C\}$) = 100.0%

How do we Mine Association Rules?

- **Input**
  - A database of transactions
  - Each transaction is a list of items (Ex. purchased by a customer in a visit)
- Find all strong rules that associate the presence of one set of items with that of another set of items.
  - Example: *98% of people who purchase tires and auto accessories also get automotive services done*
  - There are no restrictions on the number of items in the head or body of the rule.

Mining Frequent Itemsets: the Key Step

➔Iteratively find the *frequent itemsets*, i.e. sets of items that have minimum support, with cardinality from 1 to k (k-itemsets)

➔Based on the ***Apriori principle***:

Any subset of a frequent itemset must also be frequent.

E.g., if $\{AB\}$ is a frequent itemset, both $\{A\}$ and $\{B\}$ must be frequent itemsets.

➔Use the frequent itemsets to generate association rules.

The Apriori Algorithm

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) **do begin**
  C_{k+1} = candidates generated from L_k;
  for each transaction t in database **do**
    increment the count of all candidates in C_{k+1} that are contained in t
  L_{k+1} = candidates in C_{k+1} with min_support
  end
return $\cup_k L_k$;

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

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

Generating Association Rules from Frequent Itemsets

- Only strong association rules are generated.
- Frequent itemsets satisfy minimum support threshold.
- Strong AR satisfy minimum confidence threshold.

- Confidence(P ➔ Q) = Prob(Q/P) = $\frac{\text{Support}(P \cup Q)}{\text{Support}(P)}$

For each frequent itemset, **f**, generate all non-empty subsets of **f**.
For every non-empty subset **s** of **f do**
output rule **s➔(f-s)** if support(**f**)/support(**s**) ≥ min_confidence
end

Data Mining

- Needing More than just Information Retrieval
- Elementary Concepts
- Patterns and Rules to be Discovered
- Requirements and Challenges
- Association Rule Mining
- Classification
- Clustering

What is Classification?

The goal of data classification is to organize and categorize data in distinct classes.

- ▶ A model is first created based on the data distribution.
- ▶ The model is then used to classify new data.
- ▶ Given the model, a class can be predicted for new data.