Association Rules

Isabelle Bloch

LIP6, Sorbonne Université - LTCI, Télécom Paris

isabelle.bloch@sorbonne-universite.fr, isabelle.bloch@telecom-paris.fr

Objectives

- Data mining.
- Knowledge discovery.
- Automatic construction of rules from examples.
- Frequent patterns.
- Typical example: market basket.

Example

	Items
1	novel, newspaper
2	novel, film, comics, contemporary music
3	newspaper, film, comics, classical music
4	novel, newspaper, film, comics
5	novel, newspaper, film, classical music

Examples of rules:

- {film } $\Rightarrow$ {comics}
- {newspaper, novel} $\Rightarrow$ {film, classical music}
- {comics, novel} $\Rightarrow$ {newspaper}

Interpretation :

- $\Rightarrow$ means co-occurrence (not causality...)
- $X \Rightarrow Y$ = if attributes of X are present in an example, then so are attributes of Y.

Rule induction:

- Derivation of a set of rules to classify examples.
- Creation of independent rules.
- Rules may not cover all possible cases.
- Rules may be conflicting.

Definitions

- Itemset $=$ collection of items
- k-itemset $=$ itemset that contains k items
- Support count σ $=$ number of occurrences of an itemset
- Support s $=$ Fraction of transactions that contain an itemset
- Frequent itemset $=$ itemset whose support is greater than or equal to a *minsup* threshold

Example

- itemset $\{$newspaper, novel, film$\}$
- $\sigma(\{\text{newspaper, novel, film}\}) = 2$
- $s(\{\text{newspaper, novel, film}\}) = 2/5$

Association rule

Expression of the form $X \Rightarrow Y$ (X and Y: itemsets)

- Support of a rule:

$$S(X \Rightarrow Y) = \frac{\sigma(X, Y)}{|T|}$$

  ($|T|$ total number of records)
  Measures the relative frequency of co-occurrences of X and Y.

- Confidence in a rule:

$$C(X \Rightarrow Y) = \frac{\sigma(X, Y)}{\sigma(X)}$$

  Measures how often items in Y appear in records containing X.

Example: $\{\text{newspaper, film}\} \Rightarrow \{\text{comics}\}$

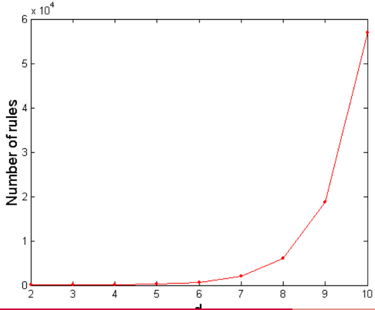
$$S = \frac{\sigma(\{\text{newspaper, film, comics}\})}{|T|} = \frac{2}{5} = 0.4$$

$$C = \frac{\sigma(\{\text{newspaper, film, comics }\})}{\sigma(\{\text{newspaper, film}\})} = \frac{2}{3} = 0.67$$

Brute force method:

1. List all possible association rules.
2. Compute S and C for each rule.
3. Prune rules for which $S < minsup$ or $C < minconf$ (two preset thresholds).

But intractable in practice...



d items
2^d itemsets
$R = \sum_{i=1}^{d-1} C_d^i (\sum_{j=1}^{d-i} C_{d-i}^j)$
possible association rules
$d = 6 \Rightarrow R = 602$

Example from the same itemset (X, Y):

- $\{$newspaper,film$\} \Rightarrow \{$comics$\}$ $(S = 0.4, C = 0.67)$
- $\{$newspaper,comics$\} \Rightarrow \{$film$\}$ $(S = 0.4, C = 1.0)$
- $\{$film,comics$\} \Rightarrow \{$newspaper$\}$ $(S = 0.4, C = 0.67)$
- $\{$comics$\} \Rightarrow \{$newspaper,film$\}$ $(S = 0.4, C = 0.67)$
- $\{$film$\} \Rightarrow \{$newspaper,comics$\}$ $(S = 0.4, C = 0.5)$
- $\{$newspaper$\} \Rightarrow \{$film,comics$\}$ $(S = 0.4, C = 0.5)$

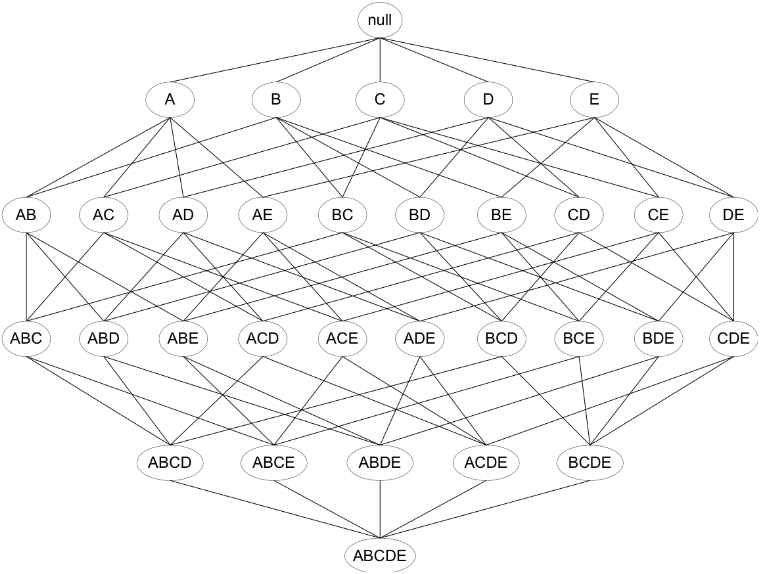
Same S and different C.

Algorithm based on frequent items

1. Frequent itemset generation, with $S \geq minsup$.
2. Rule generation, from binary partition of each frequent itemset, and with $C \geq minconf$.

Still computationally expensive!

Lattice of itemsets



Source: Tan, Steinbach, Karpatne, Kumar. Introduction to Data Mining

	Items
1	novel, newspaper
2	novel, film, comics, contemporary music
3	newspaper, film, comics, classical music
4	novel, newspaper, film, comics
5	novel, newspaper, film, classical music

$\longleftarrow \quad w \quad \longrightarrow$

Number of potential candidates: $M = 2^d$.
For each candidate itemset: scan the database ($N = |T|$) to compute the support.
Complexity in $O(NwM)$ ...

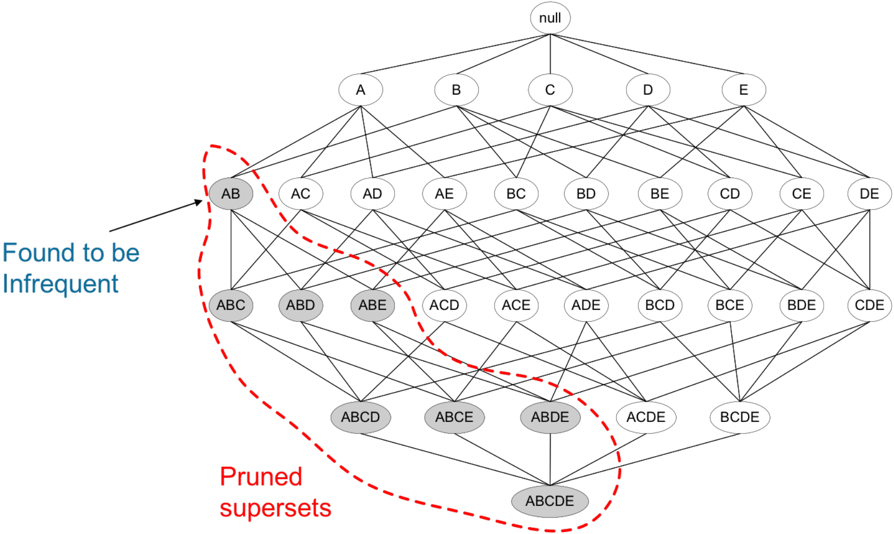
How to reduce the complexity?

- Reduce the number of candidates M
  - using pruning
  - example: A Priori Algorithm
- Reduce the number of records N
- Reduce the number of comparisons NM using efficient data structures (e.g. hash tables, frequent pattern tree) that avoid testing every candidate against every record.

A Priori Algorithm

A priori principle: If an itemset is frequent, then all of its subsets must also be frequent.
Results from the monotony of the support measure:

$$X \subseteq Y \Rightarrow S(X) \geq S(Y)$$



Source: Tan, Steinbach, Karpatne, Kumar. Introduction to Data Mining

A Priori Algorithm (Rakesh Agrawal and Ramakrishnan Sikrant, 1994)

1. Generate the set of frequent items F_1, $k = 1$
2. $k = k + 1$
3. Generate the set F_k of itemsets of cardinality k in F_{k-1}
4. Compute support and prune F_k to keep only the frequent itemsets
5. Return to step 2

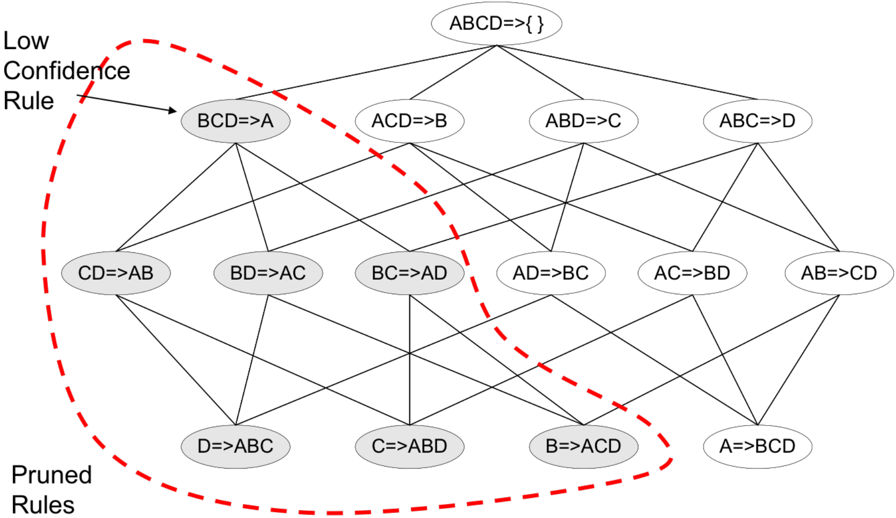
Example: apply the algorithm to the previous example.

Computing the association rules

1. Frequent itemset L.
2. Compute all non-empty subsets $L' \subset L$ (partition $L', L \setminus L'$ of the itemset).
3. Generate the rule $L' \Rightarrow L \setminus L'$ if it has a confidence higher than *confmin*.
4. If $C(L' \Rightarrow L \setminus L') < confmin$, Use the monotony property of C among rules generated by the same itemset to eliminate rules $L'' \Rightarrow L \setminus L''$ with $L'' \subset L'$ (i.e. $L \setminus L' \subset L \setminus L''$).

Example: $C(WXY \Rightarrow Z) \geq C(WX \Rightarrow YZ) \geq C(W \Rightarrow XYZ)$

Lattice of rules



Source: Tan, Steinbach, Karpatne, Kumar. Introduction to Data Mining

Conclusion

- Non-supervised rule generation.
- Easy interpretation.
- Many algorithms.
- Many extensions (measures for association rules...).
- Extensions to non-binary data:
  - continuous: discretization
  - categorial: new item for each attribute-value pair
  - sequential (in time)
  - ...