Data mining knowledge representation

1 What Defines a Data Mining Task?

- Task relevant data: where and how to retrieve the data to be used for mining
- Background knowledge: Concept hierarchies
- Interestingness measures: informal and formal selection techniques to be applied to the output knowledge
- Representing input data and output knowledge: the structures used to represent the input of the output of the data mining techniques
- Visualization techniques: needed to best view and document the results of the whole process

2 Task relevant data

- Database or data warehouse name: where to find the data
- Database tables or data warehouse cubes
- Condition for data selection, relevant attributes or dimensions and data grouping criteria: all this is used in the SQL query to retrieve the data

3 Background knowledge: Concept hierarchies

The concept hierarchies are induced by a *partial order*[1] over the values of a given attribute. Depending on the type of the ordering relation we distinguish several types of concept hierarchies.

3.1 Schema hierarchy

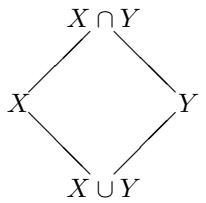
- Relating concept generality. The ordering reflects the generality of the attribute values, e.g. $street < city < state < country$.

3.2 Set-grouping hierarchy

- The ordering relation is the *subset* relation ($\subseteq$). Applies to set values.
- Example:

  $\{13, ..., 39\} = young$; $\{13, ..., 19\} = teenage$;
  $\{13, ..., 19\} \subseteq \{13, ..., 39\} \Rightarrow teenage < young$.

- Theory:
  - *power set*: the set of all subsets of a set, X.
  - *lattice* $(2^X, \subseteq)$, $sup(X, Y) = X \cap Y$, $inf(X, Y) = X \cup Y$.



  - top element $\top = \{\}$ (empty set), bottom element $\bot = X$.

[1] Consider a set A and an ordering relation R. R is a *full order* if for any $x, y \in A$, xRy exists. R is a *partial order* if for any $x \in A$, there exists $y \in A$, such that either xRy or yRx exists.

3.3 Operation-derived hierarchy

Produced by applying an operation (encoding, decoding, information extraction). For example:

markovz@cs.ccsu.edu

instantiates the hierarcy $user\_name < department < university < usa\_univeristy$.

3.4 Rule-based hierarchy

Using rules to define the partial order, for example:

if *antecedent* then *consequent*

defines the order $antecedent < consequent$.

4 Interestingness measures

Criteria to evaluate *hypotheses* (knowledge extracted from data when applying data mining techniques). This issue will be discussed in more detail in Lecture notes - Chapter 9: "Evaluating what's been learned".

4.1 Bayesian evaluation

- E - data
- $H = \{H_1, H_2, ..., H_n\}$ - hypotheses
- $H_{best} = argmax_i\ P(H_i|E)$
- Bayes theorem:

$$P(H_i|E) = \frac{P(H_i)P(E|H_i)}{\sum_{i=1}^{n} P(H_i)P(E|H_i)}$$

4.2 Simplicity

Occam's Razor

Consider for example, association rule length, decision tree size, number and length of classification rules. The intuition suggests that the best hypotesis is the simplest (shortest) one. This is the so called *Occam's Razor Principle* also expressed as a mathematical theorem (Occam's Razor Theorem). Here is an example of applying this principle to grammars:

- Data:
  $E = \{0, 000, 00000, 0000000, 000000000\}$
- Hypotheses:
  $G_1 : S \rightarrow 0|000|00000|0000000|000000000$
  $G_2 : S \rightarrow 00S|0$
- Best hypothesis: G_2 (fewer and simpler rules)

However, as simplicity is a subjective measure we need formal criteria to define it.

Formal criteria for simplicity

- Bayesian approach: need of large volume of experimental results (statistics) to define prior probabilities.
- Algorithmic (Kolmogorov) complexity of an object (bit string): the length of the shortest program of Universal Turing Machine, that generates the string. Problems: computational complexity.
- Information-based approches: Minimum Description Length Principle (MDL). Most often used in practice.

4.3 Minimum Description Length Principle (MDL)

- Bayes Theorem:

$$P(H_i|E) = \frac{P(H_i)P(E|H_i)}{\sum_{i=1}^{n} P(H_i)P(E|H_i)}$$

- Take a $-\log$ of both sides of Bayes (C is a constant):

$$-\log_2 P(H_i|E) = -\log_2 P(H_i) - \log_2 P(E|H_i) + C$$

- $I(A)$ – information in message A, $L(A)$ – min length of A in bits: $\log_2 P(A) = I(A) = L(A)$
- Then: $L(H_i|E) = L(H_i) + L(E|H_i) + C$
- MDL: The hypothesis must reduce the information needed to encode the data, i.e.

$$L(E) > L(H_i) + L(E|H_i)$$

- The best hypothesis must maximize *information compression*:

$$H_{best} = argmax_i \ (L(E) - L(H_i) - L(E|H_i))$$

4.4 Certainty

- Confidence of association "if A then B":

$$P(B|A) = \frac{\#\ of\ tuples\ containing\ both\ A\ and\ B}{\#\ of\ tupples\ containing\ A}$$

- Classification accuracy: Use a training set to generate the hypothesis, then test it on a separate test set.

$$Accuracy = \frac{\#\ of\ correct\ classifications}{\#\ of\ tuples\ in\ the\ test\ set}$$

- Utility (support) of association "if A then B":

$$P(A, B) = \frac{\#\ of\ tupples\ containing\ both\ A\ and\ B}{total\ \#\ of\ tupples}$$

5 Representing input data and output knowledge

5.1 Concepts (classes, categories, hypotheses): things to be mined/learned

- *Classification* mining/learning: predicting a discrete class, a kind of supervised learning, success is measured on new data for which class labels are known (test data).
- *Association* mining/learning: detecting associations between attributes, can be used to predict any attribute value and more than one attribute values, hence more rules can be generated, therefore we need constraints (minimum support and minimum confidence).
- *Clustering*: grouping similar instances into clusters, a kind of unsupervised learning, success is measured subjectively or by objective functions.
- *Numeric prediction*: predicting a numeric quantity, a kind of supervised learning, success is measured on test data.
- *Concept description*: output of the learning scheme

5.2 Instances (examples, tuples, transactions)

- Things to be classified, associated, or clustered.
- Individual, independent examples of the concept to be learned (target concept).
- Described by predetermined set of attributes.
- Input to the learning scheme: set of instances (dataset), represented as a single relation (table).
- Independence assumption: no relationships between attributes.
- Positive and negative examples for a concept, Closed World Assumption (CWA): $\{negative\} = \{all\} \backslash \{positive\}$.
- Relational (First Order Logic) descriptions:
  - Using variables (more compact representation). For example: $< a, b, b >$, $< a, c, c >$, $< b, a, a >$ can be represented as one relational tuple $< X, Y, Y >$.
  - Multiple relation concepts (FOIL, Inductive Logic Programming, see Lecture Notes - Chapter 11). Example:

    $grandfather(X, Z) \leftarrow father(X, Y) \wedge (father(Y, Z) \vee mother(Y, Z))$

5.3 Attributes (features)

- Predefined set of features to describe an instance.
- Nominal (categorical, enumerated, discrete) attributes:
  - Values are distinct symbols.
  - No relation among nominal values.

  - Only equality test can be performed.
  - Special case: boolean attributes, transforming nominal to boolean.
- Structured:
  - Partial order among nominal values
  - Example: concept hierarchy
- Numeric:
  - Continuous: full order (e.g. integer or real numbers).
  - Interval: partial order.

5.4 Output knowledge representation

- Association rules
- Decision trees
- Classification rules
- Rules with relations
- Prediction schemes:
  - Nearest neighbor
  - Bayesian classification
  - Neural networks
  - Regression
- Clusters:
  - Type of grouping: partitions/hierarchical
  - Grouping or describing: agglomerative/conceptual
  - Type of descriptions: statistical/structural

6 Visualization techniques: Why visualize data?

- Identifying problems:
  - Histograms for nominal attributes: is the distribution consistent with background knowledge?
  - Graphs for numeric values: detecting outliers.
- Visualization show dependencies
- Consulting domain experts
- If data are too much, take a sample