

Generalization and Decision Tree Induction: Efficient Classification in Data Mining

by

M. Kamber, L. Winstone, W. Gong, S. Cheng, J. Han

Presented by: Rubeena Shahnaz
EE 380L Data Mining

Purpose

- Presents a data classification method that addresses the efficiency and scalability issues concerning data mining in large databases.
- The classification method integrates a decision tree classifier with attribute-oriented induction and relevance analysis.

Decision Tree Classifiers

- Classification rules can be expressed as a decision tree, which is a flow chart like structure.
- Each leaf of the decision tree represents a class, other nodes represent attribute-based decision or tests.
- Each outgoing branch corresponds to a possible outcome of the test.

To classify an unlabeled data sample, the classifier tests the attribute values of the sample against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for the sample.

Problems with Decision Tree Classifiers

- Most of the classification algorithms using decision trees have the restriction that the training tuples should reside in main memory.
- So for large training sets tree construction becomes inefficient due to swapping of the training samples in and out of main and cache memories.

Three Steps Toward Achieving Efficiency and Scalability

- Attribute-oriented induction,
- Relevance analysis,
- Multi-level mining.

Attribute-oriented Induction

A concept tree ascending technique is used which replaces attribute values by generalized concepts from the corresponding attribute concept hierarchies.

Advantages:

- This allows the user to view the data at more meaningful abstractions compared to the primitive level.
- Classification tree becomes more understandable, smaller, easier to interpret.

Attribute-oriented Induction (cont'd.)

- The data is more compact than the original data (i.e. data is compressed).

NOTE: Identical data tuples are merged at the time of generalization and count information is gathered which registers the number of tuples in the original training set that the generalized tuple represents.

- Also does attribute removal, which makes the data more compact. An attribute is removed if it has large number of distinct values and no higher level concept for it.

Generalized data is stored in a multi dimensional data cube for fast accessing.

Attribute-oriented Induction (cont'd.)

- To What Level Each Attribute should be Generalized?

Answer:

Intermediate level: Trade off between too high a level and too low a level.

- Intermediate level can be specified by a domain expert or by a threshold that defines the desired number of distinct values for each attribute (like 3-6 distinct values etc.).
- In some cases predefined concept hierarchy may not work well. Then level-adjustment is needed.

Relevance Analysis

This step identifies attributes that are either irrelevant or redundant.

- Information-theoretic asymmetric measure of relevance known as uncertainty coefficient is used for this scheme.

Uncertainty coefficient,

$$U(A) = \frac{I(p_1, p_2, \dots, p_m) - E(A)}{I(p_1, p_2, \dots, p_m)}.$$

Relevance Analysis (cont'd.)

- $U(A)=0$ means statistical independence between the classifying attribute and attribute A . $U(A) =1$ means strongest degree of relevance between the two attributes.
- User may retain either the n (user defined) most relevant attributes or all the attributes whose uncertainty coefficient is greater than the pre-specified uncertainty threshold.

Multi-level Mining

- This combines the decision tree with the knowledge in concept hierarchies.
- Once a decision tree has been derived, the concept hierarchies can be used to generalize or specialize individual nodes in the tree.
- Allows attribute rolling-up or drilling down and reclassification of the data for newly specified abstraction level.

Two Proposed Algorithms for Multi-level Decision Tree Induction

- 1) MedGen - Directly applies a decision tree algorithm to the data that is generalized to an intermediate level.
- 2) MedGenAdjust – Allows for dynamic adjustment between different levels of abstraction during the tree building process.

MedGen Algorithm

Steps are:

- 1) Data collection, 2) Generalization (controlled by a set of thresholds), 3) Relevance analysis and 4) Decision tree construction.

Decision Tree Construction:

- a) Select the attribute, which gives maximum information gain as the decision or the test attribute and partition the current set of objects accordingly.
- b) For each subset created repeat step 4(a) to further classify until either
 - 1) all or substantial portion (not less than the classifying threshold) of the objects are in one class, 2) no more attributes can be used for further classification, or 3) the percentage of objects in the subclass (wrt the total # of the training samples) is below the exception threshold.

MedGenAdjust Algorithm

- It is same as MedGen, only the decision tree induction step is replaced by level-adjusted decision tree generation.
- This allows generalization and/or specialization of the abstraction level of the individual decision nodes in the tree.

The level-adjustment process consists of

- 1) Node-merge,
- 2) Node-split,
- 3) Split-node-merge.

Conclusions

- Brings a lot of ideas together.
- There are some flexibilities in choosing the substeps of the algorithm.
- Requires some predefined quantities (thresholds).
- The paper is easily readable and understandable.

NOTE: See Figure 2, 3 and Table 1, 2 in the paper.