Classification and Regression Trees

Johannes Gehrke
Cornell University, USA

INTRODUCTION

It is the goal of classification and regression to build a data-mining model that can be used for prediction. To construct such a model, we are given a set of training records, each having several attributes. These attributes either can be numerical (e.g., age or salary) or categorical (e.g., profession or gender). There is one distinguished attribute—the dependent attribute; the other attributes are called predictor attributes. If the dependent attribute is categorical, the problem is a classification problem. If the dependent attribute is numerical, the problem is a regression problem. It is the goal of classification and regression to construct a data-mining model that predicts the (unknown) value for a record, where the value of the dependent attribute is unknown. (We call such a record an unlabeled record.) Classification and regression have a wide range of applications, including scientific experiments, medical diagnosis, fraud detection, credit approval, and target marketing (Hand, 1997).

Many classification and regression models have been proposed in the literature; among the more popular models are neural networks, genetic algorithms, Bayesian methods, linear and log-linear models and other statistical methods, decision tables, and tree-structured models, which is the focus of this article (Breiman, Friedman, Olshen & Stone, 1984). Tree-structured models, so-called decision trees, are easy to understand; they are non-parametric and, thus, do not rely on assumptions about the data distribution; and they have fast construction methods even for large training datasets (Lim, Loh & Shih, 2000). Most data-mining suites include tools for classification and regression tree construction (Goebel & Gruenwald, 1999).

BACKGROUND

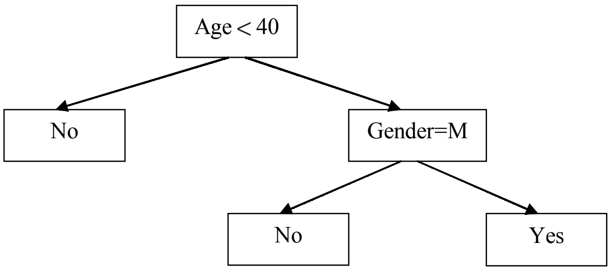
Let us start by introducing decision trees. For the ease of explanation, we are going to focus on binary decision trees. In binary decision trees, each internal node has two children nodes. Each internal node is associated with a predicate, called the splitting predicate, which involves only the predictor attributes. Each leaf node is associated with a unique value for the dependent attribute. A decision encodes a data-mining model as follows. For an unlabeled record, we start at the root node. If the record satisfies the predicate associated with the root node, we follow the tree to the left child of the root, and we go to the right child otherwise. We continue this pattern through a unique path from the root of the tree to a leaf node, where we predict the value of the dependent attribute associated with this leaf node. An example decision tree for a classification problem, a classification tree, is shown in Figure 1. Note that a decision tree automatically captures interactions between variables, but it only includes interactions that help in the prediction of the dependent attribute. For example, the rightmost leaf node in the example shown in Figure 1 is associated with the classification rule: “If (Age >= 40) and (Gender=male), then YES”; as classification rule that involves an interaction between the two predictor attributes age and salary.

Decision trees can be mined automatically from a training database of records, where the value of the dependent attribute is known: A decision tree construction algorithm selects which attribute(s) to involve in the splitting predicates, and the algorithm decides also on the shape and depth of the tree (Murthy, 1998).

MAIN THRUST

Let us discuss how decision trees are mined from a training database. A decision tree usually is constructed in two phases. In the first phase, the growth phase, an overly large and deep tree is constructed from the training data. In the second phase, the pruning phase, the final size of the tree is determined with the goal to minimize the expected misprediction error (Quinlan, 1993).

Figure 1. An example classification tree



There are two problems that make decision tree construction a hard problem. First, construction of the optimal tree for several measures of optimality is an NP-hard problem. Thus, all decision tree construction algorithms grow the tree top-down according to the following greedy heuristic: At the root node, the training database is examined, and a splitting predicate is selected. Then the training database is partitioned according to the splitting predicate, and the same method is applied recursively at each child node. The second problem is that the training database is only a sample from a much larger population of records. The decision tree has to perform well on records drawn from the population, not on the training database. (For the records in the training database, we already know the value of the dependent attribute.)

Three different algorithmic issues need to be addressed during the tree construction phase. The first issue is to devise a split selection algorithm, such that the resulting tree models the underlying dependency relationship between the predictor attributes and the dependent attribute well. During split selection, we have to make two decisions. First, we need to decide which attribute we will select as the splitting attribute. Second, given the splitting attribute, we have to decide on the actual splitting predicate. For a numerical attribute X, splitting predicates are usually of the form $X \leq c$, where c is a constant. For example, in the tree shown in Figure 1, the splitting predicate of the root node is of this form. For a categorical attribute X, splits are usually of the form X in C, where C is a set of values in the domain of X. For example, in the tree shown in Figure 1, the splitting predicate of the right child node of the root is of this form. There exist decision trees that have a larger class of possible splitting predicates; for example, there exist decision trees with linear combinations of numerical attribute values as splitting predicates $\sum a_i X_i + c \geq 0$, where i ranges over all attributes) (Loh & Shih, 1997). Such splits, also called oblique splits, result in shorter trees; however, the resulting trees are no longer easy to interpret.

The second issue is to devise a pruning algorithm that selects the tree of the right size. If the tree is too large, then the tree models the training database too closely instead of modeling the underlying population. One possible choice of pruning a tree is to hold out part of the training set as a test set and to use the test set to estimate the misprediction error of trees of different size. We then simply select the tree that minimizes the misprediction error.

The third issue is to devise an algorithm for intelligent management of the training database in case the training database is very large (Ramakrishnan & Gehrke, 2002). This issue has only received attention in the last decade, but there exist now many algorithms that can construct decision trees over extremely large, disk-resident training databases (Gehrke, Ramakrishnan & Ganti, 2000; Shafer, Agrawal & Mehta, 1996).

In most classification and regression scenarios, we also have costs associated with misclassifying a record, or with being far off in our prediction of a numerical dependent value. Existing decision tree algorithms can take costs into account, and they will bias the model toward minimizing the expected misprediction cost instead of the expected misclassification rate, or the expected difference between the predicted and true value of the dependent attribute.

FUTURE TRENDS

Recent developments have expanded the types of models that a decision tree can have in its leaf nodes. So far, we assumed that each leaf node just predicts a constant value for the dependent attribute. Recent work, however, has shown how to construct decision trees with linear models in the leaf nodes (Dobra & Gehrke, 2002). Another recent development in the general area of data mining is the use of ensembles of models, and decision trees are a popular model for use as a base model in ensemble learning (Caruana, Niculescu-Mizil, Crew & Ksikes, 2004). Another recent trend is the construction of data-mining models of high-speed data streams, and there have been adaptations of decision tree construction algorithms to such environments (Domingos & Hulten, 2002). A last recent trend is to take adversarial behavior into account (e.g., in classifying spam). In this case, an adversary who produces the records to be classified actively changes his or her behavior over time to outsmart a static classifier (Dalvi, Domingos, Mausam, Sanghai & Verma, 2004).

CONCLUSION

Decision trees are one of the most popular data-mining models. Decision trees are important, since they can result in powerful predictive models, while, at the same time, they allow users to get insight into the phenomenon that is being modeled.

REFERENCES

Breiman, L., Friedman, J.H., Olshen, R.A., & Stone, C.J. (1984). *Classification and regression trees.* Kluwer Academic Publishers.

Caruana, R., Niculescu-Mizil, A., Crew, R., & Ksikes, A. (2004). Ensemble selection from libraries of models. *Pro-*