Classification Methods

Aijun An
York University, Canada

INTRODUCTION

Generally speaking, classification is the action of assigning an object to a category according to the characteristics of the object. In data mining, classification refers to the task of analyzing a set of pre-classified data objects to learn a model (or a function) that can be used to classify an unseen data object into one of several predefined classes. A data object, referred to as an example, is described by a set of attributes or variables. One of the attributes describes the class that an example belongs to and is thus called the class attribute or class variable. Other attributes are often called independent or predictor attributes (or variables). The set of examples used to learn the classification model is called the training data set. Tasks related to classification include regression, which builds a model from training data to predict numerical values, and clustering, which groups examples to form categories. Classification belongs to the category of supervised learning, distinguished from unsupervised learning. In supervised learning, the training data consists of pairs of input data (typically vectors), and desired outputs, while in unsupervised learning there is no a priori output.

Classification has various applications, such as learning from a patient database to diagnose a disease based on the symptoms of a patient, analyzing credit card transactions to identify fraudulent transactions, automatic recognition of letters or digits based on handwriting samples, and distinguishing highly active compounds from inactive ones based on the structures of compounds for drug discovery.

BACKGROUND

Classification has been studied in statistics and machine learning. In statistics, classification is also referred to as discrimination. Early work on classification focused on discriminant analysis, which constructs a set of discriminant functions, such as linear functions of the predictor variables, based on a set of training examples to discriminate among the groups defined by the class variable. Modern studies explore more flexible classes of models, such as providing an estimate of the join distribution of the features within each class (e.g. Bayesian classification), classifying an example based on distances in the feature space (e.g. the k-nearest neighbor method), and constructing a classification tree that classifies examples based on tests on one or more predictor variables (i.e., classification tree analysis).

In the field of machine learning, attention has more focused on generating classification expressions that are easily understood by humans. The most popular machine learning technique is decision tree learning, which learns the same tree structure as classification trees but uses different criteria during the learning process. The technique was developed in parallel with the classification tree analysis in statistics. Other machine learning techniques include classification rule learning, neural networks, Bayesian classification, instance-based learning, genetic algorithms, the rough set approach and support vector machines. These techniques mimic human reasoning in different aspects to provide insight into the learning process.

The data mining community inherits the classification techniques developed in statistics and machine learning, and applies them to various real world problems. Most statistical and machine learning algorithms are memory-based, in which the whole training data set is loaded into the main memory before learning starts. In data mining, much effort has been spent on scaling up the classification algorithms to deal with large data sets. There is also a new classification technique, called association-based classification, which is based on association rule learning.

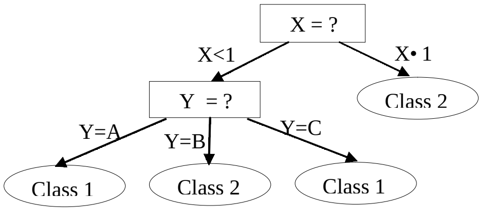
MAIN THRUST

Major classification techniques are described below. The techniques differ in the learning mechanism and in the representation of the learned model.

Decision Tree Learning

Decision tree learning is one of the most popular classification algorithms. It induces a decision tree from data. A decision tree is a tree structured prediction model where each internal node denotes a test on an attribute, each outgoing branch represents an outcome of the test, and each leaf node is labeled with a class or

Figure 1. A decision tree with tests on attributes X and Y



class distribution. A simple decision tree is shown in Figure 1. With a decision tree, an object is classified by following a path from the root to a leaf, taking the edges corresponding to the values of the attributes in the object.

A typical decision tree learning algorithm adopts a top-down recursive divide-and conquer strategy to construct a decision tree. Starting from a root node representing the whole training data, the data is split into two or more subsets based on the values of an attribute chosen according to a splitting criterion. For each subset a child node is created and the subset is associated with the child. The process is then separately repeated on the data in each of the child nodes, and so on, until a termination criterion is satisfied. Many decision tree learning algorithms exist. They differ mainly in attribute-selection criteria, such as information gain, gain ratio (Quinlan, 1993), gini index (Breiman, Friedman, Olshen, & Stone, 1984), etc., termination criteria and post-pruning strategies. Post-pruning is a technique that removes some branches of the tree after the tree is constructed to prevent the tree from over-fitting the training data. Representative decision algorithms include CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993). There are also studies on fast and scalable construction of decision trees. Representative algorithms of such kind include RainForest (Gehrke, Ramakrishnan, & Ganti, 1998) and SPRINT (Shafer, Agrawal, & Mehta., 1996).

Decision Rule Learning

Decision rules are a set of if-then rules. They are the most expressive and human readable representation of classification models (Mitchell, 1997). An example of decision rules is “if X<1 and Y=B, then the example belongs to Class 2”. This type of rules is referred to as propositional rules. Rules can be generated by translating a decision tree into a set of rules – one rule for each leaf node in the tree. A second way to generate rules is to learn rules directly from the training data. There is a variety of rule induction algorithms. The algorithms induce rules by searching in a hypothesis space for a hypothesis that best matches the training data. The algorithms differ in the search method (e.g. general-to-specific, specific-to-general, or two-way search), the search heuristics that control the search, and the pruning method used. The most widespread approach to rule induction is *sequential covering*, in which a greedy general-to-specific search is conducted to learn a disjunctive set of conjunctive rules. It is called sequential covering because it sequentially learns a set of rules that together cover the set of positive examples for a class. Algorithms belonging to this category include CN2 (Clark & Boswell, 1991), RIPPER (Cohen, 1995) and ELEM2 (An & Cercone, 1998).

Naive Bayesian Classifier

The naive Bayesian classifier is based on Bayes’ theorem. Suppose that there are *m* classes, $C_1, C_2, \ldots, C_m$. The classifier predicts an unseen example X as belonging to the class having the highest posterior probability conditioned on X. In other words, X is assigned to class C_i if and only if

$$P(C_i|X) > P(C_j|X) \text{ for } 1\leq j \leq m,\ j \neq i.$$

By Bayes’ theorem, we have

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{P(X)}.$$

As $P(X)$ is constant for all classes, only $P(X \mid C_i)P(C_i)$ needs to be maximized. Given a set of training data, $P(C_i)$ can be estimated by counting how often each class occurs in the training data. To reduce the computational expense in estimating $P(X|C_i)$ for all possible *X*s, the classifier makes a naïve assumption that the attributes used in describing *X* are conditionally independent of each other given the class of *X*. Thus, given the attribute values $(x_1, x_2, \ldots x_n)$ that describe X, we have

$$P(X \mid C_i) = \prod_{j=1}^{n} P(x_j \mid C_i).$$

The probabilities $P(x_1|C_i)$, $P(x_2|C_i)$, …, $P(x_n|C_i)$ can be estimated from the training data.

The naïve Bayesian classifier is simple to use and efficient to learn. It requires only one scan of the training data. Despite the fact that the independence assumption is often violated in practice, naïve Bayes often competes well with more sophisticated classifiers. Recent theoretical analysis has shown why the naive Bayesian classifier is so robust (Domingos & Pazzani, 1997; Rish, 2001).

Bayesian Belief Networks

A Bayesian belief network, also known as Bayesian network and belief network, is a directed acyclic graph whose nodes represent variables and whose arcs represent dependence relations among the variables. If there is an arc from node *A* to another node *B*, then we say that *A* is a parent of *B* and *B* is a descendent of *A*. Each variable is conditionally independent of its nondescendents in the graph, given its parents. The variables may correspond to actual attributes given in the data or to "hidden variables" believed to form a relationship. A variable in the network can be selected as the class attribute. The classification process can return a probability distribution for the class attribute based on the network structure and some conditional probabilities estimated from the training data, which predicts the probability of each class.

The Bayesian network provides an intermediate approach between the naïve Bayesian classification and the Bayesian classification without any independence assumptions. It describes dependencies among attributes, but allows conditional independence among subsets of attributes.

The training of a belief network depends on the senario. If the network structure is known and the variables are observable, training the network only consists of estimating some conditional probabilities from the training data, which is straightforward. If the network structure is given and some of the variables are hidden, a method of gradient decent can be used to train the network (Russell, Binder, Koller, & Kanazawa, 1995). Algorithms also exist for learning the netword structure from training data given observable variables (Buntime, 1994; Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1995).

The *k*-Nearest Neighbour Classifier

The *k*-nearest neighbour classifier classifies an unknown example to the most common class among its *k* nearest neighbors in the training data. It assumes all the examples correspond to points in a *n*-dimensional space. A neighbour is deemed nearest if it has the smallest distance, in the Euclidian sense, in the *n*-dimensional feature space. When *k* = 1, the unknown example is classified into the class of its closest neighbour in the training set. The *k*-nearest neighbour method stores all the training examples and postpones learning until a new example needs to be classified. This type of learning is called instance-based or lazy learning.

The *k*-nearest neighbour classifier is intuitive, easy to implement and effective in practice. It can construct a different approximation to the target function for each new example to be classified, which is advantageous when the target function is very complex, but can be discribed by a collection of less complex local approximations (Mitchell, 1997). However, its cost of classifying new examples can be high due to the fact that almost all the computation is done at the classification time. Some refinements to the k-nearest neighbor method include weighting the attributes in the distance computation and weighting the contribution of each of the *k* neighbors during classification according to their distance to the example to be classified.

Neural Networks

Neural networks, also referred to as *artificial neural networks*, are studied to simulate the human brain although brains are much more complex than any artificial neural network developed so far. A neural network is composed of a few layers of interconnected computing units (neurons or nodes). Each unit computes a simple function. The input of the units in one layer are the outputs of the units in the previous layer. Each connection between units is associated with a weight. Parallel computing can be performed among the units in each layer. The units in the first layer take input and are called the input units. The units in the last layer produces the output of the networks and are called the output units. When the network is in operation, a value is applied to each input unit, which then passes its given value to the connections leading out from it, and on each connection the value is multiplied by the weight associated with that connection. Each unit in the next layer then receives a value which is the sum of the values produced by the connections leading into it, and in each unit a simple computation is performed on the value - a sigmoid function is typical. This process is then repeated, with the results being passed through subsequent layers of nodes until the output nodes are reached. Neural networks can be used for both regression and classification. To model a classification function, we can use one output unit per class. An example can be classified into the class corresponding to the output unit with the largest output value.

Neural networks differ in the way in which the neurons are connected, in the way the neurons process their input, and in the propogation and learning methods used (Nurnberger, Pedrycz, & Kruse, 2002). Learning a neural network is usually restricted to modifying the weights based on the training data; the structure of the initial network is usually left unchanged during the learning process. A typical network structure is the *multilayer feed-forward neural network*, in which none of the connections cycles back to a unit of a previous layer. The most widely used method for training a feed-forward neural network is backpropagation (Rumelhart, Hinton, & Williams, 1986).

Support Vector Machines

The support vector machine (SVM) is a recently developed technique for multidimensional function approximation. The objective of support vector machines is to determine a classifier or regression function which minimizes the empirical risk (that is, the training set error) and the confidence interval (which corresponds to the generalization or test set error) (Vapnik, 1998).

Given a set of N linearly separable training examples $S = \{\mathbf{x}_i \in R^n \mid i = 1,2,..., N\}$, where each example belongs to one of the two classes, represented by $y_i \in \{+1,-1\}$, the SVM learning method seeks the optimal hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, as the decision surface, which separates the positive and negative examples with the largest margin. The decision function for classifying linearly separable data is:

$$f(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x} + b) ,$$

where **w** and b are found from the training set by solving a constrained quadratic optimization problem. The final decision function is

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \right).$$

The function depends on the training examples for which α_i is non-zero. These examples are called support vectors. Often the number of support vectors is only a small fraction of the original dataset. The basic SVM formulation can be extended to the nonlinear case by using nonlinear kernels that map the input space to a high dimensional feature space. In this high dimensional feature space, linear classification can be performed. The SVM classifier has become very popular due to its high performances in practical applications such as text classification and pattern recognition.

FUTURE TRENDS

Classification is a major data mining task. As data mining becomes more popular, classification techniques are increasingly applied to provide decision support in business, biomedicine, financial analysis, telecommunications and so on. For example, there are recent applications of classification techniques to identify fraudulent usage of credit cards based on credit card transaction databases; and various classification techniques have been explored to identify highly active compounds for drug discovery. To better solve application-specific problems, there has been a trend toward the development of more application-specific data mining systems (Han & Kamber, 2001).

Traditional classification algorithms assume that the whole training data can fit into the main memory. As automatic data collection becomes a daily practice in many businesses, large volumes of data that exceed the memory capacity become available to the learning systems. Scalable classification algorithms become essential. Although some scalable algorithms for decision tree learning have been proposed, there is still a need to develop scalable and efficient algorithms for other types of classification techniques, such as decision rule learning.

Previously, the study of classification techniques focused on exploring various learning mechanisms to improve the classification accuracy on unseen examples. However, recent study on imbalanced data sets has shown that classification accuracy is not an appropriate measure to evaluate the classification performance when the data set is extremely unbalanced, in which almost all the examples belong to one or more, larger classes and far fewer examples belong to a smaller, usually more interesting class. Since many real world data sets are unbalanced, there has been a trend toward adjusting existing classification algorithms to better identify examples in the rare class.

Another issue that has become more and more important in data mining is privacy protection. As data mining tools are applied to large databases of personal records, privacy concerns are rising. Privacy-preserving data mining is currently one of the hottest research topics in data mining and will remain so in the near future.

CONCLUSION

Classification is a form of data analysis that extracts a model from data to classify future data. It has been studied in parallel in statistics and machine learning, and is currently a major technique in data mining with a broad application spectrum. Since many application problems can be formulated as a classification problem and the volume of the available data has become overwhelming, developing scalable, efficient, domain-specific, and privacy-preserving classification algorithms is essential.

REFERENCES

An, A., & Cercone, N. (1998). ELEM2: A learning system for more accurate classifications. *Proceedings of the 12th Canadian Conference on Artificial Intelligence*, 426-441.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*, Wadsworth International Group.

Buntine, W.L. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, *2*, 159-225.

Castillo, E., Gutiérrez, J.M., & Hadi, A.S. (1997). Expert systems and probabilistic network models. New York: Springer-Verlag.

Clark P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the 5th European Working Session on Learning*, 151-163.

Cohen, W.W. (1995). Fast effective rule induction. Proceedings of the 11th International Conference on Machine Learning, 115-123, Morgan Kaufmann.

Cooper, G., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*, 309-347.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, *29*, 103-130.

Gehrke, J., Ramakrishnan, R., & Ganti, V. (1998). RainForest - A framework for fast decision tree construction of large datasets. *Proceedings of the 24th International Conference on Very Large Data Bases*.

Han, J., & Kamber, M. (2001). *Data mining — Concepts and techniques*. Morgan Kaufmann.

Heckerman, D., Geiger, D., & Chickering, D.M. (1995) Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197-243.

Mitchell, T.M. (1997). *Machine learning*. McGraw-Hill.

Nurnberger, A., Pedrycz, W., & Kruse, R. (2002). *Neural network approaches*. In Klosgen & Zytkow (Eds.), *Handbook of data mining and knowledge discovery*. Oxford University Press.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. Artificial Intelligence, *29*(3), 241-288.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Rish, I. (2001). An empirical study of the naive Bayes classifier. *Proceedings of IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.*

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533-536.

Russell, S., Binder, J., Koller, D., & Kanazawa, K. (1995). Local learning in probabilistic networks with hidden variables. *Proceedings of the 14th Joint International Conference on Artificial Intelligence*, *2*, 1146-1152.

Shafer, J., Agrawal, R., & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. *Proceedings of the 22th International Conference on Very Large Data Bases*.

Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley & Sons.

KEY TERMS

Backpropagation: A neural network training algorithm for feedforward networks where the errors at the output layer are propagated back to the previous layer to update connection weights in learning. If the previous layer is not the input layer, then the errors at this hidden layer are propagated back to the layer before.

Disjunctive Set of Conjunctive Rules: A conjunctive rule is a propositional rule whose antecedent consists of a conjunction of attribute-value pairs. A disjunctive set of conjunctive rules consists of a set of conjunctive rules with the same consequent. It is called disjunctive because the rules in the set can be combined into a single disjunctive rule whose antecedent consists of a disjunction of conjunctions.

Generic Algorithm: An algorithm for optimizing a binary string based on an evolutionary mechanism that uses replication, deletion, and mutation operators carried out over many generations.

Information Gain: Given a set E of classified examples and a partition $P = \{E_1, ..., E_n\}$ of E, the information gain is defined as

$$entropy(E) - \sum_{i=1}^{n} entropy(E_i) * \frac{|E_i|}{|E|},$$

where $|X|$ is the number of examples in X, and $entropy(X) = -\sum_{j=1}^{m} p_j \log_2(p_j)$ (assuming there are m classes in X and p_j denotes the probability of the jth class in X). Intuitively, the information gain measures the decrease of the weighted average impurity of the partitions $E_1, \ldots, E_n$, compared with the impurity of the complete set of examples E.

Machine Learning: The study of computer algorithms that develop new knowledge and improve its performance automatically through past experience.

Rough Set Data Analysis: A method for modeling uncertain information in data by forming lower and upper approximations of a class. It can be used to reduce the feature set and to generate decision rules.

Sigmod Function: A mathematical function defined by the formula

$$P(t) = \frac{1}{1+e^{-t}}$$

Its name is due to the sigmoid shape of its graph. This function is also called the standard logistic function.