

CHAPTER-16

16 . CLASSIFICATION AND PREDICTION

16.1 Introduction

16.2 Example

16.3 Issues Regarding Classification and Prediction

16.4 Comparing Classification Methods

16.5 Review Questions

16.6 References

16 . CLASSIFICATION AND PREDICTION

16.1 Introduction

Databases are rich with hidden information that can be used for making intelligent business decisions. Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Whereas classification predicts categorical labels, prediction models continuous-valued functions. For example, a classification model may be built to predict the expenditures of potential customers on computer equipment given their income and occupation. Many classification and prediction methods have been proposed by researchers in machine learning, expert systems, statistics, and neurobiology. Most algorithms are memory resident, typically assuming a small data size. Recent database mining research has built on such work, developing scalable classification and prediction techniques capable of handling large disk-resident data. These techniques often consider parallel and distributed processing.

The basic techniques for data classification such as decision tree induction, Bayesian classification and Bayesian belief networks, and neural networks. The integration of data warehousing technology with classification is also discussed, as well as association-based classification. Other approaches to classification, such as k-nearest neighbour classifiers, case-based reasoning, genetic algorithms, rough sets, and fuzzy logic techniques are introduced. Methods for prediction, including linear, non-linear and generalized linear regression models, are briefly discussed. Where applicable, you will learn of modifications, extensions, and optimizations to these techniques for their application to data classification and prediction for large databases,

Data classification is a two-step process. In the first step, a model is built describing a predetermined set of data classes or concepts. The data classification process:

(a) Learning : Training data are analyzed by a classification algorithm. Here, the class label attribute is `credit_rating`, and the learned model or classifier is represented in the form of classification rules.

(b) Classification: Test data are used to estimate the accuracy of the classification rules—if the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples constructed by analyzing database tuples described by attributes. Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute. In the context of classification, data tuples are also referred to as samples, examples, or objects. The data tuples analyzed to build the model collectively form the training data set. The individual tuples making up the training set are referred to as training samples and are randomly selected from the sample population. Since the class label of each training sample is provided, this step is also known as supervised learning (i.e., the learning of the model is “supervised” in that it is told to which class each training sample belongs). It contrasts with unsupervised learning (or clustering), in which the class label of each training sample is not known, and the number or set of classes to be learned may not be known in advance.

Typically, the learned model is represented in the form of classification rules, decision trees, or mathematical formulae. For example, given a database of customer credit information, classification rules can be learned to identify customers as having either excellent or fair credit ratings. The rules can be used to categorize future data samples, as well as provide a better understanding of the database contents.

In the second step, the model is used for classification. First, the predictive accuracy of the model [or classifier] is estimated. The hold method is a simple technique that uses a test set of class-labeled samples. These samples are randomly

selected and are independent of the training samples. The accuracy of a model on a given test set is the percentage of test set samples that are correctly classified by the model. For each test set sample, the known class label is compared with the learned model's class prediction for that sample. Note that if the accuracy of the model were estimated based on the training data set, this estimate could be optimistic since the learned model tends to overfit the data (that is, it may have incorporated some particular anomalies of the training data that are not present in the overall sample population) therefore, a test set is used.

If the accuracy of the model is considered acceptable, the model can be used to classify future data tuples or objects for which the class label is not known. (Such data are also referred to in the machine learning literature as "unknown" or "previously unseen" data.) For example, the classification rules learned from the analysis of data from existing customers can be used to predict the credit rating of new or future (i.e., previously unseen) customers.

Prediction can be viewed as the construction and use of a model to assess the class of an unlabeled sample, or to assess the value or value ranges of an attribute that a given sample is likely to have. In this view, classification and regression are the two major types of prediction problems, where classification is used to predict discrete or nominal values, while regression is used to predict continuous or ordered values. In our view, however, we refer to the use of prediction to predict class labels as classification, accurate use of prediction to predict continuous values (e.g., using regression techniques) as prediction. This view is commonly accepted in data mining.

Classification and prediction have numerous applications including credit approval, medical diagnosis, performance prediction, and selective marketing.

16.2 Example:

Suppose that we have a database of customers on the ABCCompany mailing list. The mailing list is used to send out promotional literature describing new products and upcoming price discounts. The database describes attributes of the customers, such as their name, age, income, occupation, and credit rating. The customers can be classified as to whether or not they have purchased a computer at ABC company.

Suppose that new customers are added to the database and that you would like to notify these customers of an upcoming computer sale. To send out promotional literature to every new customer in the database can be quite costly. A more cost efficient method would be to target only those new customers who are likely to purchase a new computer. A classification model can be constructed and used for this purpose.

Suppose instead that you would like to predict the number of major purchases that a customer will make at ABCCompany during a fiscal year. Since the predicted value

here is ordered, a prediction model can be constructed for this purpose.

16.3 Issues Regarding Classification and Prediction

This section describes issues regarding preprocessing the data for classification and prediction. Criteria for the comparison and evaluation of classification methods are also described.

Preparing the data for Classification and prediction

The following preprocessing steps may be applied to the data in order to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data cleaning: This refers to the preprocessing of data in order to remove or reduce noise (by applying smoothing techniques, for example) and the treatment of missing values [e.g by replacing a missing value with the most commonly occurring value for that attribute, or with most probable value based on statistics] although most classification algorithms have some mechanisms for handling noisy or missing data, this step can help reduce confusion during learning.

Relevance analysis: Many of the attributes in the data may be irrelevant to the

classification or prediction task. For example, data recording the day of the week on which a bank loan application was filed is unlikely to be relevant to the success of the application. Furthermore, other attributes may be redundant. Hence, relevance analysis may be performed on the data with the aim of removing any irrelevant or redundant attributes from the learning process. In machine learning, this step is known as feature selection. Including such attributes may otherwise slow down, and possibly mislead, the learning step.

Ideally, the time spent on relevance analysis, when added to the time spent on learning from the resulting “reduced” feature subset, should be less than the time that would have been spent on learning from the original set of features. Hence, such analysis can help improve classification efficiency and scalability.

Data transformation: The data can be generalized to higher-level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous-valued attributes. For example, numeric values for the attribute income may be generalized to discrete ranges such as low, medium, and high. Similarly, nominal-valued attributes, like street, can be generalized to higher-level concepts, like city. Since generalization compresses the original training data, fewer input/output operations may be involved during learning.

The data may also be normalized, particularly when neural networks or methods

Involving distance measurements are used in the learning step. Normalization involves scaling all values for a given attribute so that they fall within a small-specified range, such as -1.0 to 1.0 , or 0.0 to 1.0 . In methods that use distance measurements, for Example, this would prevent attributes with initially large ranges (like,say, income) From outweighing attributes with initially smaller ranges (such as binary attributes).

16.4 Comparing Classification Methods

Classification and prediction methods can be compared and evaluated according to

The following criteria:

- **Predictive accuracy:** This refers to the ability of the model to correctly predict the Class label of new or previously unseen data.
- **Speed:** This refers to the computation costs involved in generating and using the model.
- **Robustness:** This is the ability of the model to make correct predictions given noisy data or data with missing values.
- **Scalability:** This refers to the ability to construct the model efficiently given large amounts of data.

- **Interpretability:** This refers to the level of understanding and insight that is provided by the model.

The database research community's contributions to classification and prediction for data mining have emphasized the scalability aspect, particularly with respect to decision tree induction.

A decision tree is a flow chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an out-come of the test, and leaf nodes represent classes or class distributions. The top-most node in a tree is the root node.

In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node that holds the class prediction for that sample. Decision trees can easily be converted to classification rules.

We describe a basic algorithm for learning decision trees later. When decision trees are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data.

Algorithm: Generate_decision_tree. Generate a decision tree from the given training data.

Input: The training samples, samples, represented by discrete-valued attributes:
The set of candidate attributes, attribute-list.

Output: A decision tree.

Method:

- (1) create a node N
- (2) if samples are all of the same class, C then
- (3) return N as a leaf node labeled with the class C;
- (4) if attribute-list is empty then
- (5) return N as a leaf node labeled with the most common class in samples
- (6) select test-attribute, the attribute among attribute-list with the highest information gain;
- (7) label node N with test-attribute;
- (8) for each known value a_i , of test-attribute // partition the samples
- (9) grow a branch from node N for the condition test-attribute== a_i ;
- (10) let s_i ; is empty then
- (11) if s_i is empty then

(12)attach a leaf labeled with the most common class in samples;

(13)else attach the node returned by Generate_decision_tree(s_i, attribute-list test-attribute);

16.5 Review Questions

1 Explain the Issues Regarding Classification and Prediction

4 Compare Classification Methods

16.6 References

[1]. Data Mining Techniques, Arun k pujari 1st Edition

[2] .Data warehousing,Data Mining and OLAP, Alex Berson ,smith.j. Stephen

[3].Data Mining Concepts and Techniques ,Jiawei Han and MichelineKamber

[4]Data Mining Introductory and Advanced topics, Margaret H Dunham PEA

[5] The Data Warehouse lifecycle toolkit , Ralph Kimball Wiley student Edition

