IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

"CLASSIFICATION PROBLEM IN DATA MINING - BY USING DECISION TREES"

VIKAS CHAHAR
Associate Professor
VIMT Rohtak, Haryana (India)
vikas.chahar@gmail.com

Abstract

The aim of this paper is to present the classification problem in data mining using decision trees. Simply stated, data mining refers to extracting or "mining" knowle dge from large amounts of data. Data mining known by different names as – knowledge mining, knowledge ex traction, data/pattern analysis, data archaeology, data dredging, knowledge discovery in databases (KDD). Data Mining, or Knowledge Discovery in Databases (KDD) as it is also known, is the nontrivial extraction of implicit, previously unknown, and potentially useful information from data. Classification is an important problem in data mining. Given a database D= {t1,t2,…., tn} and a set of classes C= {Cl,..…, Cm}, the Class ification Problem is to define a mapping f: D – → C where each it is assigned to one class. It means that given a database of records, each with a class label, a classifier generates a concise and meaningful description for each class that can be used to classify subsequent records. Actually classifier divides the database into equivalence classes that is each class contains same type of records.

Key words: classification problem, data mining, decision trees, Knowledge Discovery in Databases (KDD)

1. INTRODUCTION

Generally, **data mining** (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information – information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

HOW DATA MINING WORKS

The process of data mining consists of three stages:

1. The Initial exploration
2. Model building or pattern identification with validation or verification.
3. Deployment (the application of model to new Data in order to generation predictions).

Stage 1: Exploration. This stage usually starts with data preparation which may involve cleaning data, data transformations and selecting subsets of records. Also, in case of data sets with large number of variables ("fields") some preliminary feature selection operations are performed to bring the number of variables to a manageable range (depending on the statistical methods, which are being considered). Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere between a simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods (see Exploratory Data Analysis [EDA] in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

Stage 2: Model building and validation. This stage involves considering various models and choosing the best one based on their predictive performance (i.e. explaining the variability in question and producing stable results across samples). This may sound like a simple operation, but in fact, it sometimes involves a very elaborate process. There are a variety of techniques developed to achieve that goal – many of which are based on so-called "competitive evaluation of models," that is, applying different models to the same data set and then comparing their performance to choose

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

the best. These techniques – which are often considered the core of predictive data mining – include: Bagging (Voting, Averaging), Boosting, Stacking (Stacked Generalizations), and Meta-Learning.

Stage 3: Deployment. That final stage involves using the model selected as best in the previous stage and applying it to new data in order to generate predictions or estimates of the expected outcome.

II. WHAT CAN BE DISCOVERED?

The kinds of patterns that can be discovered depend upon the data mining tasks employed. By and large, there are two types of data mining tasks: descriptive data mining tasks that describe the general properties of the existing data, and predictive data mining tasks that attempt to do predictions based on inference on available data. The data mining functionalities and the variety of knowledge they discover are briefly given below

- Characterization
- Discrimination
- Association analysis
- **Classification**
- Prediction
- Clustering
- Outlier analysis

The aim of this paper is to present the classification problem in data mining using decision trees.

Classification analysis is the organization of data in given classes. Also known as supervised classification, the classification uses given class labels to order to objects in the data collection. Classification approaches normally use a training set where all objects are already associated with known class labels.

In other words, classification is the process of dividing a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to specific variable(s) you are trying to predict. For example, a typical classification problem is to divide a database of companies into groups that are as homogeneous as possible with respect to a creditworthiness variable with values "Good" and "Bad".

So, in a classification problem, you have a number of cases (examples) and wish to predict which of several classes each case belongs to. Each case consists of multiple attributes, each of which takes on one of several possible values. The attributes consist of multiple predictor attributes (independent variables) and one target attribute (dependent variable). Each of the target attribute's possible values is a class to be predicted on the basis of that case's predictor attribute values.

III. METHODOLOGY

The aim of Classification problems to identify the characteristics that indicate the group to which each case belongs. This pattern can be used both to understand the existing data and to predict how new instances will behave. For example, you may want to predict whether individuals can be classified as likely to respond to a direct mail solicitation, vulnerable to switching over to a competing long distance phone service, or a good candidate for a surgical procedure.

Classification – A Two-Step Process

Data classification is a two-step process. **In the first step**, a model is build describing a predetermined set of data classes or concepts. The model is constructed by analyzing database temples described by attributes. Each temple 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 temples 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 temples making up the training set are referred to as training samples and are randomly selected from the sample population. Typically, the learned model is represented in the form of classification rules, decision trees, or mathematical formulae. 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 holdout 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

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

given test set is the percentage of test set samples that are correctly classified by the model. For each test sample, the known class label is compared with the learned model's class prediction for that sample. 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. Classification and prediction have numerous applications including credit approval, medical diagnosis, performance prediction, and selective marketing.

Criteria for comparing classification methods

Classification method can be compared and evaluated according to the following criteria:

1. Predictive accuracy – this refers to the ability of the model to correctly predict the class label of new or previously unseen data.
2. Speed – This refers to the computation costs involved in generating and using the model.
3. Robustness – this is the ability of the model to make correct predictions given noisy data with missing values.
4. Scalability - This refers to the ability to construct the model efficiently given large amounts of data.
5. Interpretability – this refers to the level of understanding and insight that is provided by the model.

Classification has been successfully applied to several areas like medical diagnosis weather prediction, credit approval, customer segmentation and fraud detection. Many different techniques have been proposed for classification– ***Bayesian classification, neural networks, genetic algorithms, decision trees, rule induction etc.***

IV. DECISION TREE CLASSIFIERS

Among these proposals, **decision tree classifiers** have found the widest applicability in large scale data mining environments. As in data mining applications, very large training sets with several million examples are common; a decision tree classifier scales well and can handle training data of this magnitude. The ability to classify large training data can also improve the classification accuracy. So given our goal of classifying large data sets, we focus mainly on decision tree classifiers.

Decision trees are tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a data set. Specific decision tree methods include classification and regression trees (CART), ID3, C4.5, SLIQ etc. For the goal of classifying large does sets-decision trees are mainly focused. They are relatively fast compared to other classification methods. A decision tree can be converted into simple and easy to understand classification rules. They can also be converted into SQL queries for accessing databases.

TYPES OF DECISION TREES

Decision tree has two other names:

Regression tree: Decision trees used to predict continuous variables are called regression trees. That is they approximate real-valued functions instead of being used for classification tasks (e.g. estimate the price of a house or a patient's length of stay in a hospital).

Classification tree: Decision trees which are used to predict categorical variables are called classification trees because they place instances in categories or classes. For example if the Y is a categorical variable like, sex (male or female), the result of a game (loses or wins).

REQUIREMENTS FOR DECISION TREES

Decision tree induction is a typical inductive approach to learn knowledge on classification. The key requirements to do mining with decision trees etc:

- **Attribute-value description**: object or case must be expressible in terms of a fixed collection of properties or attributes.
- **Predefined classes**: The categories to which cases are to be assigned must have been established beforehand (supervised data).
- **Discrete classes**: A case does or does not belong to a particular class, and there must be for more cases than classes.
- **Sufficient data**: Usually hundreds or even thousands of training cases.
- **"Logical" classification model** : Classifier that can be only expressed s decision trees or set of production rules.

CHARACTERISTICS OF DECISION TREES

Hierarchical nature of decision trees: The hierarchical nature of classification trees is one

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

of their most basic features. The relationship of a leaf to the tree on which it grows can be described by the hierarchy of splits of branches (starting from the trunk) leading to the last branch from which the leaf hangs.

Flexibility of decision trees: Another distinctive characteristic of classification trees is their flexibility. The ability of classification trees to perform univariate splits, examining the effects of predictors one at a time has implications for the variety of types of predictors that can be analyzed. Decision trees can be computed for categorical predictors, continuous predictors, or any mix of the two types of predictors when univariate splits is used.

STRENGTHS AND WEAKNESSES OF DECISION TREE METHODS ARE

Strengths –

- Decision trees are able to generate understandable rules
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

Weaknesses –

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.

- Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

- Decision trees do not treat well non-rectangular regions. Most decision-tree algorithms only examine a single field at a time. This leads to rectangular classification boxes that may not correspond well with the actual distribution of records in the decision space.

V. DECISION TREE CLASSIFICATION

Most decision tree classifiers perform classification in two phases. Tree Building and Tree Pruning.

- **Tree Building**: An initial decisions tree is grown in this phase by repeatedly partitioning the training data. The training set is split into two or more partitions using an attribute. This process is repeated recursively until all the examples in each partition belong to one class. Figure gives an overview of the process –

Make Tree (Training Data T)

Partition T;

Partition (Data S)

If (all points in S are in the same class) then return;

Evaluate splits for each attribute A

Use best split found to partition S into S1 and S2;

Partition (S1);

Partition (S2);

Fig. Tree Building Algorithm

In the building phase, for every partition a new node is added to the decision tree; initially, the tree has a single root node for the entire data set. For a set of records in a partition S, a test criterion C for further partitioning the set into S1… Sm is first determined. New nodes for S1… Sm are created and these are added to the decision tree as children of the node for S. Also, the node for S is labeled with test C, and partitions S1… Sm are then recursively partitioned. A partition in which all the records have identical class labels is not partitioned further, and a leaf corresponding to it is labeled with the class.

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

The building phase constructs a perfect tree that accurately classifies every record from the training set. It is an iterative process which involves splitting the data into progressively smaller subsets. Each iteration considers the data in only one node. The first iteration considers the root node that contains all the data. Subsequent iterations work on derivative nodes that will contain subsets of the data.

The algorithm begins by analyzing the data to find the independent variable that when used as a splitting rule will result in nodes that are most different from each other with respect to the dependent variable. There are several alternative ways to measure this difference. Some implementations have only one measure built in; others let the user choose which measure to use. Entropy, mutual info, gain ratio, gini, and chi-squared. Regardless of the measurement used, all methods require a cross-tabulation between the dependent variable and each of the independent variables.

One important characteristic of the tree splitting algorithm is that it is greedy. Greedy algorithms make decisions locally rather than globally. When deciding on a split at a particular node, a greedy algorithm does not look forward in the tree to see if another decision would produce a better overall result.

Once a node is split, the same process is performed on the new nodes, each of which contains a subset of the data in the parent node. The variables are analyzed and the best split is chosen. This process is repeated until only nodes where no splits should be made remain.

- **Tree Pruning**: The tree built in the first phase completely classifies the training data set. This implies that branches are created in the tree even for spurious “noise” data and statistical fluctuations. These branches can lead to errors when classifying test data. Tree proving is aimed at removing these branches from the decision tree for selecting the sub-tree with the least estimated error rate.

There are two main approaches to estimating the error rate:

One using the original training dataset and the other using an independent dataset for error estimation. Cross-validation belongs to the first category. Multiple samples are taken from the training data and a tree is grown for each sample. These multiple trees are then used to estimate the error rates of the sub trees of the original tree. Although this approach selects compact trees with high accuracy, it is inapplicable for large data sets, where building even one decision tree is expensive. Alternative approaches that use only a single decision tree often lead to large decision trees.

The **second** class of methods divides the training data into two parts where one part is used to build the tree and the other for pruning the tree. The data used for pruning should be selected such that it captures the “true” data distribution, which brings up a potential problem with this method. How large should the test sample be and how should it be selected? Moreover, using portions of the data only for pruning reduces the number of training examples available for the tree-growing phase, which can lead to reduced accuracy.

There are two common approaches to tree pruning –

(a) Pre-pruning approach

In this approach, a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training samples at a given node). Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset samples or the probability distribution of those samples.

When constructing a tree, measures such as statistical significance, information gain, and so on, can be used to assess the goodness of a split. If partitioning the samples at a node would result in a split that falls below a pre-specified threshold, then further partitioning of the subset is halted. There are difficulties, however, in choosing an appropriate threshold. High thresholds could result in oversimplified trees, while low thresholds could result in very little simplification.

(b) Post-pruning approach

This approach removes branches from a “fully grown” tree. A tree node is pruned by removing its branches. The cost complexity pruning algorithm is an example of the post-pruning approach. The lowest un-pruned node becomes a leaf and is labeled by the most frequent class among its former branches. For each non-leaf

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): 2231 –5268
www.ijcsms.com

node in the tree, the algorithm calculates the expected error rate that would occur if the sub tree at that node were pruned. Next, the expected error rate occurring if the node were not; pruned is calculated using the error rates for each branch, combined by weighting according to the proportion of observations along each branch. If pruning the node leads to a greater expected error rate, then the sub tree is kept. Otherwise, it is pruned. After generating a set of progressively pruned trees, an independent test set is used to estimate the accuracy of each tree. The decision tree that minimizes the expected error rate is preferred.

Post-pruning requires more computation than pre-pruning, yet generally leads to a more reliable tree. Studies have shown that post-pruning will result in smaller and more accurate trees by up to 25%.

TESTING A TREE

Prior to integrating any decision tree into your business as a predictor, you must test and validate the model using an independent dataset. Once accuracy has been measured on an independent dataset and is determined to be acceptable, the tree (or its rules) is ready to be used as a predictor. Be sure to retest the tree periodically to insure that it maintains the desired accuracy.

UNDERSTANDING THE OUTPUT

One of the inherent benefits of a decision tree model is its ability to be understood by a broad user community. Presentation of a decision tree model in a graphical format along with the ability to interactively explore it has become standard features supported by many decision tree vendors.

Decision tree output is often presented as a set of rules which are more concise and, particularly when the tree is large, are often easier to understand. In other words, the knowledge represented in decision trees can be extracted and represented in the form of classification IF-THEN rules. One rule is created for each path from the root to a leaf node. Each attribute-value pair along a given path forms a conjunction in the rule antecedent (“IF” part). The leaf node holds the class prediction, forming the rule consequent (“THEN” part).

Decision trees have obvious value as both predictive and descriptive models. We have seen that prediction can be done on a case-by-case basis by navigating the tree. More often, prediction is accomplished by processing multiple new cases through the tree or rule set automatically and generating an output file with the predicted value or class appended to the record for each case. Many implementations offer the option of exporting the rules to be used externally or embedded in other applications.

The distinctive output from a decision tree algorithm makes it easy to recognize its descriptive or exploratory value. In an exploratory mode the user is interested in outputs that facilitate insight about relationships between independent and dependent variables.

VI. CONCLUSION

Data mining makes it possible to analyze routine business transactions and glean a significant amount of information about individuals buying habits and preferences. Security is an important issue with any data collection that is shared and/or is intended to be used for strategic decision-making. In addition, when data is collected for customer profiling, user behavior understanding, correlating personal data with other information, etc., large amounts of sensitive and private information about individuals or companies is gathered and stored. This becomes controversial given the confidential nature of some of this data and the potential illegal access to the information. Moreover, data mining could disclose new implicit knowledge about individuals or groups that could be against privacy policies.

REFERENCES

[1] Herb Edelstein, “data mining: exploiting the hidden trends in your data”, www. D62mag.com/970ledel.htm.

[2] “recon: introduction to data mining”, www.occ. uniconsabruck. de/ fachgeb/win fachinfo/datamol. Level.

[3] Mitchell, Tom, M. Machine Learning, McGraw-Hill, 1997.

[4] M. Mehta, R. Agrawal, and I. Rissamen, SLIQ: A fast scalable classifier for data mining. In Proc. 1996 Int. Conf. Extending Database Technology (EDBI 96), Avignon, France, March, 1996.

IJCSMS International Journal of Computer Science & Management Studies, Vol. 11, Issue 01, May 2011 ISSN (Online): **2231 –5268**
www.ijcsms.com

[5] John, Shafer, Rakesh Agarwal and Manish Mehta. "SPRINT: A Scalable Parallel Classifier for data mining". In Proceedings of the 22nd International Conference on Very Large Data Bases, Mumbai (Bombay), India, September 1996.

[6] R. Rastogi and K.