

Chapter 1

Introduction to Decision Trees

1.1 Data Mining and Knowledge Discovery

Data mining, the science and technology of exploring data in order to discover previously unknown patterns, is a part of the overall process of knowledge discovery in databases (KDD). In today's computer-driven world, these databases contain massive quantities of information. The accessibility and abundance of this information makes data mining a matter of considerable importance and necessity.

Most data mining techniques are based on inductive learning (see [Mitchell (1997)]), where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that the trained model is applicable to future, unseen examples. Strictly speaking, any form of inference in which the conclusions are not deductively implied by the premises can be thought of as induction.

Traditionally, data collection was regarded as one of the most important stages in data analysis. An analyst (e.g., a statistician) would use the available domain knowledge to select the variables that were to be collected. The number of variables selected was usually small and the collection of their values could be done manually (e.g., utilizing hand-written records or oral interviews). In the case of computer-aided analysis, the analyst had to enter the collected data into a statistical computer package or an electronic spreadsheet. Due to the high cost of data collection, people learned to make decisions based on limited information.

Since the dawn of the Information Age, accumulating data has become easier and storing it inexpensive. It has been estimated that the amount of stored information doubles every twenty months [Frawley *et al.* (1991)].

Unfortunately, as the amount of machine-readable information increases, the ability to understand and make use of it does not keep pace with its growth.

Data mining emerged as a means of coping with this exponential growth of information and data. The term describes the process of sifting through large databases in search of interesting patterns and relationships. In practise, data mining provides tools by which large quantities of data can be automatically analyzed. While some researchers consider the term “data mining” as misleading and prefer the term “knowledge mining” [Klosgen and Zytkow (2002)], the former term seems to be the most commonly used, with 59 million entries on the Internet as opposed to 52 million for knowledge mining.

Data mining can be considered as a central step in the overall KDD process. Indeed, due to the centrality of data mining in the KDD process, there are some researchers and practitioners that regard “data mining” and the complete KDD process as synonymous.

There are various definitions of KDD. For instance [Fayyad *et al.* (1996)] define it as “the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data”. [Friedman (1997a)] considers the KDD process as an automatic exploratory data analysis of large databases. [Hand (1998)] views it as a secondary data analysis of large databases. The term “secondary” emphasizes the fact that the primary purpose of the database was not data analysis.

A key element characterizing the KDD process is the way it is divided into phases with leading researchers such as [Brachman and Anand (1994)], [Fayyad *et al.* (1996)], [Maimon and Last (2000)] and [Reinartz (2002)] proposing different methods. Each method has its advantages and disadvantages. In this book, we adopt a hybridization of these proposals and break the KDD process into eight phases. Note that the process is iterative and moving back to previous phases may be required.

- (1) Developing an understanding of the application domain, the relevant prior knowledge and the goals of the end-user.
- (2) Selecting a dataset on which discovery is to be performed.
- (3) Data Preprocessing: This stage includes operations for dimension reduction (such as feature selection and sampling); data cleansing (such as handling missing values, removal of noise or outliers); and data transformation (such as discretization of numerical attributes and attribute extraction).

- (4) Choosing the appropriate data mining task such as classification, regression, clustering and summarization.
- (5) Choosing the data mining algorithm. This stage includes selecting the specific method to be used for searching patterns.
- (6) Employing the data mining algorithm.
- (7) Evaluating and interpreting the mined patterns.
- (8) The last stage, deployment, may involve using the knowledge directly; incorporating the knowledge into another system for further action; or simply documenting the discovered knowledge.

1.2 Taxonomy of Data Mining Methods

It is useful to distinguish between two main types of data mining: verification-oriented (the system verifies the user's hypothesis) and discovery-oriented (the system finds new rules and patterns autonomously) [Fayyad *et al.* (1996)]. Figure 1.1 illustrates this taxonomy. Each type has its own methodology.

Discovery methods, which automatically identify patterns in the data, involve both prediction and description methods. Description methods focus on understanding the way the underlying data operates while prediction-oriented methods aim to build a behavioral model for obtaining new and unseen samples and for predicting values of one or more variables related to the sample. Some prediction-oriented methods, however, can also help provide an understanding of the data.

Most of the discovery-oriented techniques are based on inductive learning [Mitchell (1997)], where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that the trained model is applicable to future unseen examples. Strictly speaking, any form of inference in which the conclusions are not deductively implied by the premises can be thought of as induction.

Verification methods, on the other hand, evaluate a hypothesis proposed by an external source (like an expert etc.). These methods include the most common methods of traditional statistics, like the goodness-of-fit test, the t-test of means, and analysis of variance. These methods are less associated with data mining than their discovery-oriented counterparts because most data mining problems are concerned with selecting a hypothesis (out of a set of hypotheses) rather than testing a known one. The focus of tra-

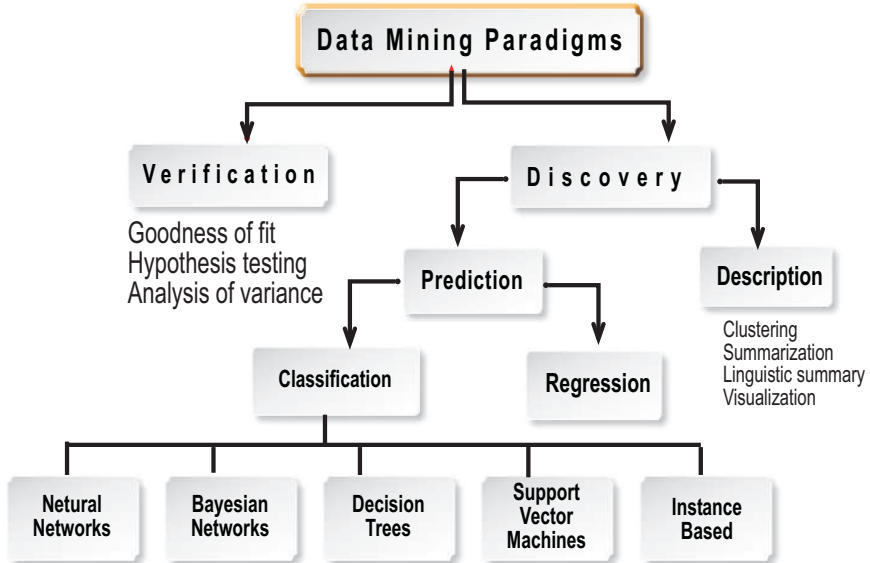


Fig. 1.1 Taxonomy of data mining Methods.

ditional statistical methods is usually on model estimation as opposed to one of the main objectives of data mining: model identification [Elder and Pregibon (1996)].

1.3 Supervised Methods

1.3.1 Overview

In the machine learning community, prediction methods are commonly referred to as supervised learning. Supervised learning stands opposed to unsupervised learning which refers to modeling the distribution of instances in a typical, high-dimensional input space.

According to [Kohavi and Provost (1998)], the term “unsupervised learning” refers to “learning techniques that group instances without a prespecified dependent attribute”. Thus the term “unsupervised learning” covers only a portion of the description methods presented in Figure 1.1. For instance the term covers clustering methods but not visualization methods.

Supervised methods are methods that attempt to discover the relation-

ship between input attributes (sometimes called independent variables) and a target attribute (sometimes referred to as a dependent variable). The relationship that is discovered is represented in a structure referred to as a *Model*. Usually models describe and explain phenomena, which are hidden in the dataset, and which can be used for predicting the value of the target attribute when the values of the input attributes are known. The supervised methods can be implemented in a variety of domains such as marketing, finance and manufacturing.

It is useful to distinguish between two main supervised models: *Classification Models (Classifiers)* and *Regression Models*. Regression models map the input space into a real-valued domain. For instance, a regressor can predict the demand for a certain product given its characteristics. On the other hand, classifiers map the input space into predefined classes. For instance, classifiers can be used to classify mortgage consumers as good (full mortgage pay back the on time) and bad (delayed pay back). Among the many alternatives for representing classifiers, there are, for example, support vector machines, decision trees, probabilistic summaries, algebraic function, etc.

This book deals mainly in classification problems. Along with regression and probability estimation, classification is one of the most studied approaches, possibly one with the greatest practical relevance. The potential benefits of progress in classification are immense since the technique has great impact on other areas, both within data mining and in its applications.

1.4 Classification Trees

In data mining, a decision tree is a predictive model which can be used to represent both classifiers and regression models. In operations research, on the other hand, decision trees refer to a hierarchical model of decisions and their consequences. The decision maker employs decision trees to identify the strategy most likely to reach her goal.

When a decision tree is used for classification tasks, it is more appropriately referred to as a classification tree. When it is used for regression tasks, it is called regression tree.

In this book we concentrate mainly on classification trees. Classification trees are used to classify an object or an instance (such as insurant) to a predefined set of classes (such as risky/non-risky) based on their attributes

values (such as age or gender). Classification trees are frequently used in applied fields such as finance, marketing, engineering and medicine. The classification tree is useful as an exploratory technique. However it does not attempt to replace existing traditional statistical methods and there are many other techniques that can be used classify or predict the membership of instances to a predefined set of classes, such as artificial neural networks or support vector machines.

Figure 1.2 presents a typical decision tree classifier. This decision tree is used to facilitate the underwriting process of mortgage applications of a certain bank. As part of this process the applicant fills in an application form that include the following data: number of dependents (DEPEND), loan-to-value ratio (LTV), marital status (MARST), payment-to-income ratio (PAYINC), interest rate (RATE), years at current address (YRSADD), and years at current job (YRSJOB).

Based on the above information, the underwriter will decide if the application should be approved for a mortgage. More specifically, this decision tree classifies mortgage applications into one of the following two classes:

- Approved (denoted as “A”) The application should be approved.
- Denied (denoted as “D”) The application should be denied.
- Manual underwriting (denoted as “M”) An underwriter should manually examine the application and decide if it should be approved (in some cases after requesting additional information from the applicant). The decision tree is based on the fields that appear in the mortgage applications forms.

The above example illustrates how a decision tree can be used to represent a classification model. In fact it can be seen as an expert system, which partially automates the underwriting process and which was built manually by a knowledge engineer after interrogating an experienced underwriter in the company. This sort of expert interrogation is called knowledge elicitation namely obtaining knowledge from a human expert (or human experts) for use by an intelligent system. Knowledge elicitation is usually difficult because it is not easy to find an available expert who is able, has the time and is willing to provide the knowledge engineer with the information he needs to create a reliable expert system. In fact, the difficulty inherent in the process is one of the main reasons why companies avoid intelligent systems. This phenomenon is known as the knowledge elicitation bottleneck.

A decision tree can be also used to analyze the payment ethics of customers who received a mortgage. In this case there are two classes:

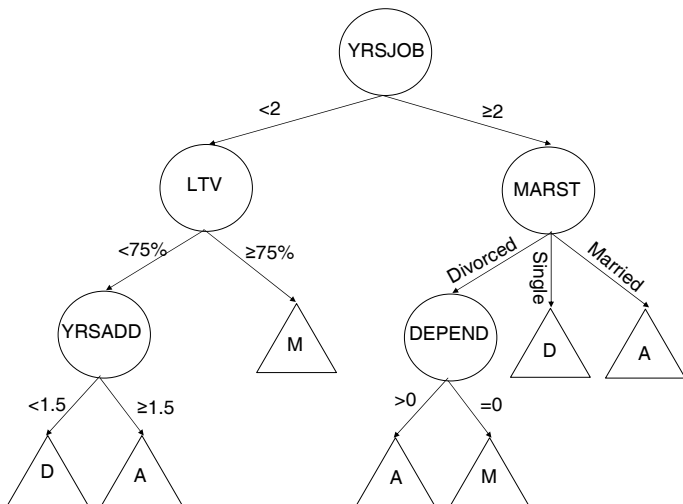


Fig. 1.2 Underwriting Decision Tree.

- Paid (denoted as “P”) - the recipient has fully paid off his or her mortgage.
- Not Paid (denoted as “N”) - the recipient has not fully paid off his or her mortgage.

This new decision tree can be used to improve the underwriting decision model presented in Figure 9.1. It shows that there are relatively many customers pass the underwriting process but that they have not yet fully paid back the loan. Note that as opposed to the decision tree presented in Figure 9.1, this decision tree is constructed according to data that was accumulated in the database. Thus, there is no need to manually elicit knowledge. In fact the tree can be grown automatically. Such a kind of knowledge acquisition is referred to as knowledge discovery from databases.

The use of a decision tree is a very popular technique in data mining. In the opinion of many researchers, decision trees are popular due to their simplicity and transparency. Decision trees are self-explanatory; there is no need to be a data mining expert in order to follow a certain decision tree. Classification trees are usually represented graphically as hierarchical structures, making them easier to interpret than other techniques. If the classification tree becomes complicated (i.e. has many nodes) then its straightforward, graphical representation become useless. For complex

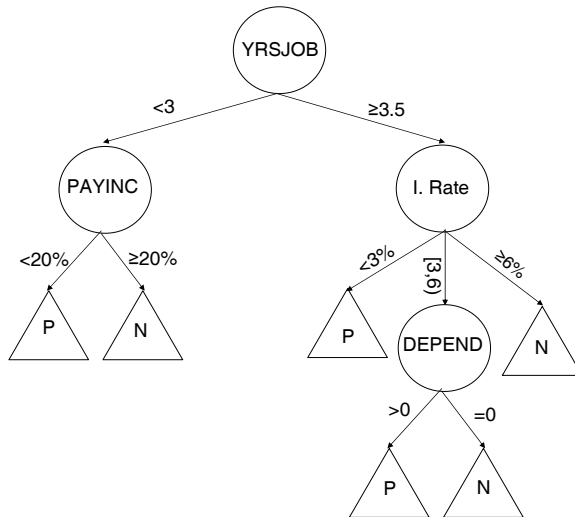


Fig. 1.3 Actual behavior of customer.

trees, other graphical procedures should be developed to simplify interpretation.

1.5 Characteristics of Classification Trees

A decision tree is a classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called a “root” that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is referred to as an “internal” or “test” node. All other nodes are called “leaves” (also known as “terminal” or “decision” nodes). In the decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attribute values. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attributes value. In the case of numeric attributes, the condition refers to a range.

Each leaf is assigned to one class representing the most appropriate tar-

get value. Alternatively, the leaf may hold a probability vector (affinity vector) indicating the probability of the target attribute having a certain value. Figure 1.4 describes another example of a decision tree that reasons whether or not a potential customer will respond to a direct mailing. Internal nodes are represented as circles, whereas leaves are denoted as triangles. Two or more branches may grow from each internal node (i.e. not a leaf). Each node corresponds with a certain characteristic and the branches correspond with a range of values. These ranges of values must give a partition of the set of values of the given characteristic.

Instances are classified by navigating them from the root of the tree down to a leaf, according to the outcome of the tests along the path. Specifically, we start with a root of a tree; we consider the characteristic that corresponds to a root; and we define to which branch the observed value of the given characteristic corresponds. Then we consider the node in which the given branch appears. We repeat the same operations for this node etc., until we reach a leaf.

Note that this decision tree incorporates both nominal and numeric attributes. Given this classifier, the analyst can predict the response of a potential customer (by sorting it down the tree), and understand the behavioral characteristics of the entire potential customer population regarding direct mailing. Each node is labeled with the attribute it tests, and its branches are labeled with its corresponding values.

In case of numeric attributes, decision trees can be geometrically interpreted as a collection of hyperplanes, each orthogonal to one of the axes.

1.5.1 *Tree Size*

Naturally, decision makers prefer a decision tree that is not complex since it is apt to be more comprehensible. Furthermore, according to [Breiman *et al.* (1984)], tree complexity has a crucial effect on its accuracy. Usually the tree complexity is measured by one of the following metrics: the total number of nodes, total number of leaves, tree depth and number of attributes used. Tree complexity is explicitly controlled by the stopping criteria and the pruning method that are employed.

1.5.2 *The hierarchical nature of decision trees*

Another characteristic of decision trees is their hierarchical nature. Imagine that you want to develop a medical system for diagnosing patients according

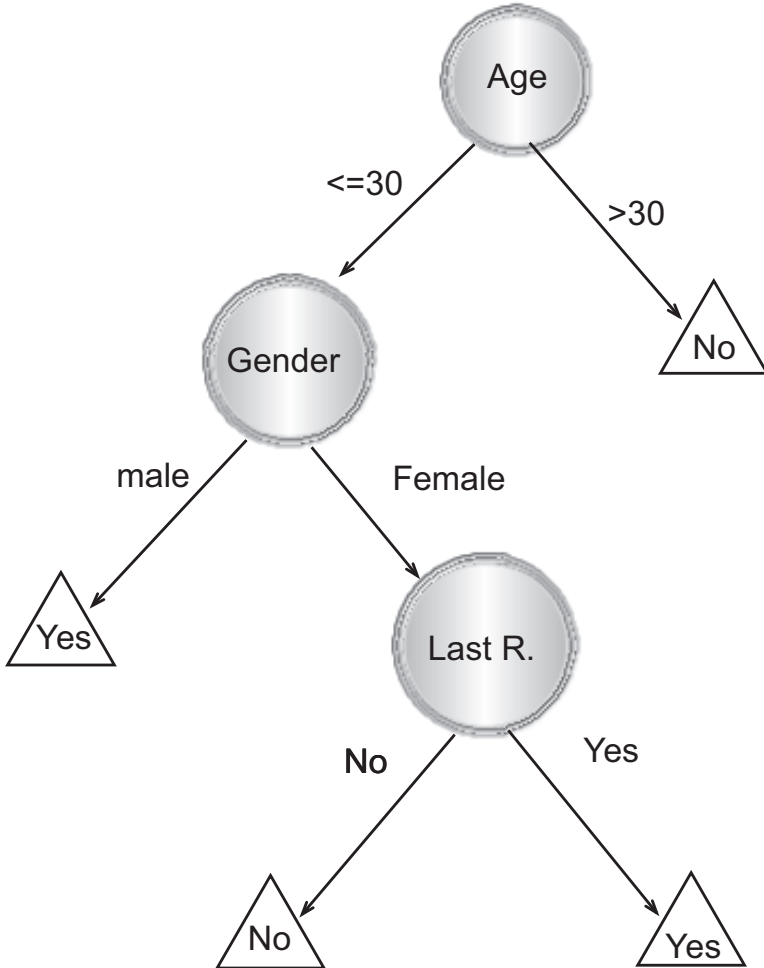


Fig. 1.4 Decision Tree Presenting Response to Direct Mailing.

to the results of several medical tests. Based on the result of one test, the physician can perform or order additional laboratory tests. Specifically, Figure 1.5 illustrates the diagnosis process, using decision trees, of patients that suffer from a certain respiratory problem. The decision tree employs the following attributes: CT finding (CTF); X-ray finding (XRF); chest pain type (CPT); and blood test finding (BTF). The physician will order an X-ray, if chest pain type is “1”. However, if chest pain type is “2”, then the physician will not order a X-ray but will order a blood test. Thus medical

tests are performed just when needed and the total cost of medical tests is reduced.

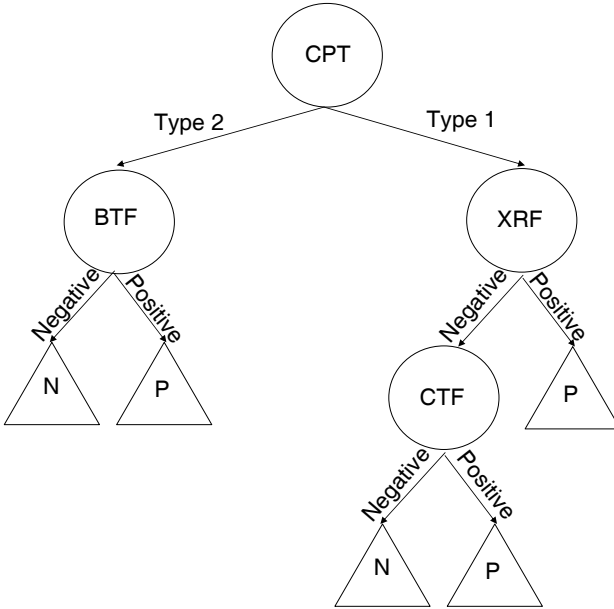


Fig. 1.5 Decision Tree For Medical Applications.

1.6 Relation to Rule Induction

Decision tree induction is closely related to rule induction. Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by conjoining the tests along the path to form the antecedent part, and taking the leaf's class prediction as the class value. For example, one of the paths in Figure 1.4 can be transformed into the rule: “If customer age is less than or equal to 30, and the gender of the customer is male — then the customer will respond to the mail”. The resulting rule set can then be simplified to improve its comprehensibility to a human user, and possibly its accuracy [Quinlan (1987)].