Chapter 1

Introduction

1.1 Context and Motivations

1.1.1 The KDD Process

Problems of automated learning and knowledge acquisition been there for quite some time but have increasingly come to the fore during the recent years due to the emergence of very large amounts of data, the dramatic cost decrease of mass storage devices, the emergence, and quick growth of data base management systems, the advances in computer technology such as faster computers and parallel architectures, the continuous developments in automatic learning techniques and the possible presence of uncertainty in data (noise, outliers, missing information). However, the 'knowledge acquisition bottleneck' [37] seems to remain one of the key problems in the design of intelligent and knowledge-based systems. However, formalizing human expert knowledge for a certain problem such as medicine etc is a very complex and time consuming task. Thus, these type of expert knowledge based systems are not providing enough satisfactory results along with their tedious approach. Thus, using readily available data for the extraction of knowledge had gained popularity because of its ease of use and interesting results. Various approaches have been studied where the expert knowledge is extracted from available data and have work reasonably.

This development has seen the emergence of the field of knowledge discovery in databases (KDD), which has attracted diverse research in recent years. The vast amount of progress

in data acquisition and storage technology, along with tools and techniques developed to analyze large amounts of data, this field has grown to a new research discipline. KDD is an amalgamation of statistics, machine learning, artificial intelligence, data bases and other areas. According to a widely accepted definition, KDD refers to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable structure in data [120],[37]. The central step within the overall KDD process is data mining, the application of computational techniques to the task of finding patterns and models in data.

The main object is to generate learning models from empirical data by the task of automated learning. The induction of models aid in the accurate prediction of future observations, finding patterns of interest or describing the interactions between various features of a given problem. A typical model induction process involves the tasks of data acquisition, data preparation, generation of models, model interpretation and validation/application. The typical KDD process has much in common with the process of inductive reasoning, except for the fact that the former can be (and indeed often is) circular in the sense that the data mining results will retract on the acquisition, selection, and preparation of the data, possibly initiating a repeated pass with modified data, analysis tools, or queries. A typical KDD process may comprise the following steps:

- Data integration and cleaning (combination of multiple sources)
- Data selection
- Data transformation (into a form suitable for the analysis)
- Data mining
- Evaluation of patterns
- Knowledge presentation

To explain this above process in more detail as shown in figure 1.1 from [118], this overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

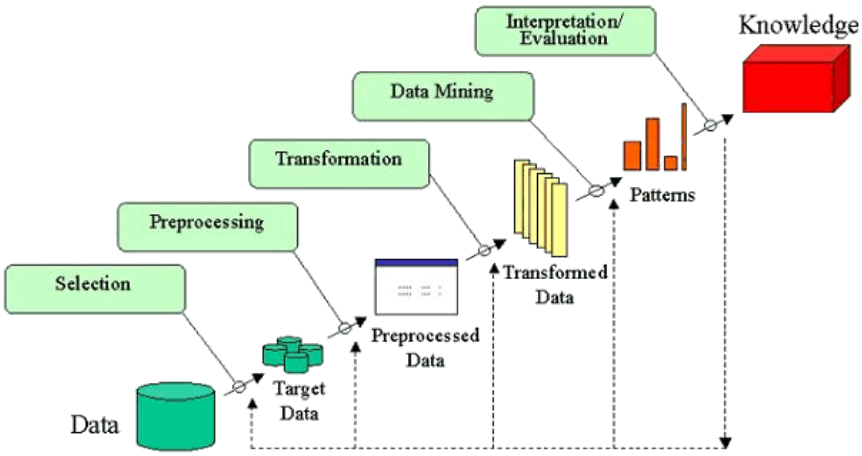

Figure 1.1: The KDD process and phases.

- Creating a target data set: selecting a data set, or focusing on a subset of variables, or data samples, on which discovery is to be performed based on the understanding of the application domain.

- Data cleaning and preprocessing that include the removal of noise or outliers, collecting necessary information to model or account for noise, strategies for handling missing data fields and depending upon the data discretization of continuous features.

- Data reduction and projection which represent finding useful features to represent the data depending on the goal of the task, using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representations for the data.

- Choosing the data mining task consisting of deciding whether the goal of the KDD process is classification, regression, clustering, etc. Then choosing the data mining algorithm(s), selecting method(s) to be used for searching for patterns in the data, deciding which models and parameters may be appropriate. Then matching a particular data mining method with the overall criteria of the KDD process.

- Applying the data mining task and searching for patterns of interest in a particular

representational form or a set of such representations as classification rules or trees, regression, clustering, and so forth.

- Interpreting mined patterns and consolidating discovered knowledge.

A common distinction of performance tasks in machine learning is supervised learning (e.g. classification and regression), unsupervised learning (e.g. clustering) and reinforcement learning (explained in later chapters).

1.1.2 The Data Mining Task

The general purpose of data mining is to process the information from the enormous stock of data. It concerns with developing methods and algorithms to extract useful information from data and support future decision making, be it in the form of a classification or a prediction problem, searching for causalities or modeling dependencies between data, extracting trends from temporal data or identifying abnormal deviations in data. Databases are usually expensive to create and maintain, and for a small additional investment in mining them, highly profitable information may be discovered hidden in the data.

The data-mining component of the KDD process often involves repeated iterative application of particular data-mining methods e.g decision trees [75] that are the main focus of the paper and are presented in chapter 2. The knowledge discovery goals are defined by the intended use of the system. Fayyad et al [120] distinguish two types of goals namely verification discovery. With verification, the system is limited to verifying the userŠs hypothesis. With discovery, the system autonomously finds new patterns. The discovery is further subdivided into prediction, where the system finds patterns for predicting the future behavior of some entities, and description, where the system finds patterns for presentation to a user in a human-understandable form. In this thesis, we are primarily concerned with discovery-oriented data mining. The goals of prediction and description can be achieved using a variety of particular data-mining methods like the following defined in [120]:

- Classification is learning a function that maps (classifies) a data item into one of several predefined classes.

- Regression is learning a function that maps a data item to a real-valued prediction variable.
- Clustering is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data.
- Summarization involves methods for finding a compact description for a subset of data.
- Dependency modeling consists of finding a model that describes significant dependencies between variables.
- Change and deviation detection focuses on discovering the most significant changes in the data from previously measured or normative values

This thesis focuses on improving the accuracy and quality of classification methods which can be supervised or unsupervised. These notions are discussed in detail in the coming chapters.

1.1.3 Data Preprocessing and Transformation

One of the most complex tasks of the KDD process is concerned with preprocessing and transformation of data, where usually 80 percent of the time is spent. The result of these steps should be data in the form suitable for data mining or to enhance the performance and quality of the data mining task or technique. The necessary preprocessing and transformation operations include the merging data, creating new attributes, excluding unnecessary attributes and discretization of data.

Many machine learning algorithms are known to produce better models by discretizing continuous attributes [121, 112]. Most of the methods that perform automatic learning require discrete attributes in order to operate in an efficient manner or to operate at all. However, a very large proportion of real data sets include continuous variables. One solution to this problem is to partition numeric variables into a number of sub-ranges and treat each such sub-range as a category. This process of partitioning continuous variables into categories is usually termed discretization. Unfortunately, the number of ways to discretize a continuous attribute is infinite. Discretization is a potential time-consuming bottleneck,

since the number of possible discretizations is exponential in the number of interval threshold candidates within the domain [112].

The goal of discretization is to find a set of cut points to partition the range into a small number of intervals that have good class coherence, which is usually measured by an evaluation function [108]. Thus, a discretization method searches for the best correlation between the class and attribute values maximizing the separability of classes in discrete intervals. In addition it tries to minimize the number of intervals without significant loss of class attribute mutual dependence. Discretization is usually a preprocessing task and these intervals can be found automatically, depending on the data or be a priori defined. It then needs to find the width, or the boundaries, of the intervals given the range of values of a continuous attribute.

Another problematic issue in data mining is that most problems of interest in data mining involve data with a large number of dimensions. The reduction of dimensionality can lead to an increased capability of extracting knowledge from the data by means of visualization, and to new possibilities in designing efficient and possibly more effective classification schemes [78]. The objective of dimensionality reduction is keeping only the most important dimensions, i.e. the ones that hold the most information for the task at hand. This is achieved by projecting some dimensions onto others. These steps will improve significantly our ability to visualize the data (by mapping them in two or three dimensions), and improve on the time complexity or inquiries into data as the need for scanning the unnecessary dimensions of vast data is no longer required. There are other advantages of using a dimensionality reduction procedure as it may help reveal the underlying dynamics governing a data set. An example is a data set in form of a swiss roll discussed in chapter 9. Where a complex three dimensional structure can easily be represented as a simple two dimensional form. A more general example could be related to text mining. A typical approach to this problem would be to represent the text as a vector of counts of the words appearing in the text. The large dimensionality of this type of data is a huge problem but an effective dimensionality reduction technique may reveal that there are only a few significant features that can be identified.

There are many approaches to dimensionality reduction based on a variety of assumptions

and underlying methodologies. One particular type of approach is based on the observation that high dimensional data is often much simpler than the original dimensionality. Thus, for example many attributes of that data set could be highly correlated in that they could represent a particular measurement or aspect of an object. The features of such a data set contain much overlapping information and thus, their simplification would require a technique that could identify the governing or underlying properties or representation of the attributes. These approaches use the notion of a manifold which is the data set lies along a low-dimensional manifold embedded in a high-dimensional space, where the low dimensional space reflects the underlying parameters and high-dimensional space is the feature space [78]. Attempting to uncover this manifold structure in a data set is referred to as manifold learning.

1.1.4 Automatic Learning

As discussed above the advantages of a data-driven approach are obvious and thus, automatic learning from data aims at building a model of the system from the data which is already available, to predict the future behavior and situations of the system. By model, we mean some relationships between the variables used to describe the system. The techniques are called automatic due to their capacity of doing the job of learning from past experience without much help or dependence on a human being, thus computer based. Thus, automatic learning is an effective and practical technique for discovering relations and extracting knowledge in cases where the mathematical model of the problem may be too expensive to get, or not available at all.

The interest of automatic learning is obvious when it is impossible to analytically model a system as it is often easy to gather data from its direct observation [97]. For example, a handwriting recognition system is very difficult to put into equation, notably because of the high variability of handwriting. However, it is easy to ask several people to write some letters or words and hence constitute a database for applying automatic learning techniques.

1.1.5 Decision Trees

There are many alternatives to represent classifiers. The decision tree [75] is probably the most widely used approach for this purpose. Originally it has been studied in the fields of decision theory and statistics. However, it was found to be effective in other disciplines such as data mining, machine learning, and pattern recognition. Decision trees are also implemented in many real-world applications. Here we introduce decision trees but they are discussed in detail in the next chapters.

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 root that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is called internal node or test nodes. All other nodes are called leaves (also known as terminal nodes or decision nodes). In the decision tree each internal node splits the instance space into two or more subspaces according to a certain discrete function of the input attributes values. In the simplest and most frequent case each test considers a single attribute, such that the instance space is partitioned according to the attributeŠs value. In the case of numeric attributes the condition refers to a range.

Each leaf is assigned to one class representing the most appropriate target value. Alternatively the leaf may hold a probability vector indicating the probability of the target value having a certain value. 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.

1.1.6 Fuzzy Reasoning

Fuzzy reasoning aims at representing the vagueness present in every day life in the mathematical representation of logic. Fuzzy sets [32] are a generalization of the conventional set theory and have been introduced by Zadeh in 1965 [73]. They adopt a more natural approach to the reasoning used by a computer. Whereas a crisp set has sharp boundaries (0 or 1), the transition between full membership (1) and non membership (0) is rather gradual in a fuzzy set. So that a model can assign a degree of certainty to an answer rather than answering in yes or no e.g. 'is that person tall?' instead of asking question like 'is that person 6 feet tall'?

The need for fuzzy representation in practice stems from the fact that it helps to better model the real life scenarios which are most of the time imprecise or fuzzy. This inherent elasticity or vagueness can be taken into consideration e.g. while performing a classification task. Thus, instead of being totally wrong in certain situations, the system can give a vague answer by assigning a degree to it. Also for a human, certain instructions could appear as too precise and the precision may be quite useless, while vague directions can be interpreted and acted upon.

From some real-world applications, precise data are difficult or expensive, if not impossible, to obtain. The theory of fuzzy logic exploits both imprecise and precise kinds of information. Fuzzy theory comes with a natural way of expressing knowledge and the resulting models are more acceptable because they represent information similarly to the way human beings understand problems and because an answer in the form of a fuzzy class provides richer information comparatively with the yes-no type of answer [21].

In the context of automatic learning using fuzzy reasoning, there exists Several approaches in which fuzzy models (e.g. fuzzy rule bases) are learned from data in a fully automated way have already been developed [17]. These fuzzy models are used for classification or other data mining tasks and give a fuzzy output with a membership attached to each decision.

1.1.7 Performance of an Automatic Learning Technique

The performance of an automatic learning based data mining task is evaluated by its percentage of accurate predictions with very good generalization capabilities on new data. At the same time it has to be interpretable and comprehensible in order to gain the people confidence in the model output. A more simple model is favored over complex models due to its efficiency because large amounts of data have to be catered. Other important aspects are the robustness of the learning method in terms of unseen data and the ability to work without much human intervention. However, most of the time there is a trade-off between these performance criteria and we have to try and keep an optimal balance between them.

1.2 Objective of the thesis

Our objective in this thesis is to propose and develop automatic learning techniques and strategies that contribute to the data preparation, transformation and mining tasks of the KDD process. These technique have to be developed by keeping in mind the improvement of the performance and quality of the overall learning process. Thus, we try to improve on the prediction accuracy, interpretability, complexity and robustness of the automatic learning techniques and methods.

In the first part of the thesis we focus on the preprocessing task of the KDD process that deals with discretization of continuous features which is discussed in the previous section. Here, we propose and develop discretization point aggregation techniques that improve the quality of discretization. Usually, discretization and other types of statistical processes are applied to subsets of the population as the entire population is practically inaccessible. For this reason we argue that the discretization performed on a sample of the population is only an estimate of the entire population. Most of the existing discretization methods, partition the attribute range into two or several intervals using a single or a set of cut points. In this part, we introduce three variants of a resampling based technique (such as bootstrap) [9, 91] to generate a set of candidate discretization points and thus, improving the discretization quality by providing a better estimation towards the entire population. Later, we validate by comparing with other discretization techniques and with an optimal partitioning method on 10 benchmark data sets. Thus, the goal of this part is to observe whether this type of resampling can lead to better quality discretization points, which opens up a new paradigm to construction of soft decision trees.

The second part of our thesis concerns with automatic fuzzy partitioning for soft decision tree induction [17] and classification. Soft or fuzzy decision tree is an extension of the classical crisp tree induction such that fuzzy logic is embedded into the induction process with the effect of more accurate models and reduced variance, but still interpretable and autonomous. This is done by incorporating fuzziness into the decision tree model to cater for the vagueness of real life problems. We modify the above resampling based partitioning method to generate fuzzy partitions. In addition we propose, develop and validate another fuzzy partitioning method based on multiple fuzzy partitions that enhances the performance

of the decision tree.

In the third and final part, we concentrate on the data transformation task in the KDD process. We consider dimensionality reduction as an important preprocessing task for further enhancement in data mining techniques. Thus, we modify an existing manifold learning based dimensionality reduction technique and use it for classification purpose. Besides the obvious advantages of reducing variables or dimensionality such as lesser complexity, better visualization and interpretability, we see whether it can enhance the predictive power of decision tree classification.

1.3 Our Contributions

Our main contributions are highlighted in the following list:

- Proposition, development and empirical validation of a resampling based discretization technique that aggregates discretization points to form better quality discretizations in section 4.2.4. In addition, the introduction of a variant using this technique in section 4.2.5. To the best of our knowledge resampling has not been used before to enhance multi-interval discretization.

- An extension of the above concept and proposition of two other ways of using bootstrap based resampling for discretization in section 4.3 and 4.4 respectively. One of which enhances the quality of top-down discretizations only and the other one uses a different approach based on decision boundaries to define discretization points.

- A bias/variance study of various discretization techniques including our methods and using both decision tree and naive bayesian classifiers to demonstrate the effect of discretization in enhancing the performance of a classifier in chapter 5. This has also been attempted in [140].

- A fuzzy partitioning technique for existing fuzzy decision tree induction such as fuzzy ID3 [92]. This is an extension of the above mentioned resampling technique towards producing fuzzy splits instead of crisp ones presented in section 7.2.1.

- Another fuzzy partitioning technique based on multiple fuzzy intervals inspired by a bi-partitioning fuzzy approach in [21] in section 7.2.4.

- Validation of the fuzzy partitioning techniques by comparing with an optimal fuzzy partitioning algorithm to generate multiple fuzzy partitions. This is an extension of the Fisher's optimal algorithm for generating multiple partitions. We also use benchmark data sets for additional comparison in section 7.2.4.

- A bias/variance study of fuzzy decision tree classifiers in comparison with our methods and with ensemble learning techniques such as bagging and boosting in chapter 8. This type of study has been done in [97] for decision tree and then in [21] for fuzzy decision trees.

- An algorithm for determining the value of K in K-nearest neighbor graphs for supervised and semi-supervised learning schemes in section 9.2.2. Thus, enhancing ISOMAP [57], a dimensionality reduction based manifold technique.

- Using the above enhanced manifold learning technique as a pre-processing step for classification in decision trees and studying its effect on classifier performance in section 9.2.3.

1.4 Organization of the thesis

This thesis is organized into 3 main parts consisting of chapters 3 to 9. However, chapter 2 lays the preliminaries to supervised automatic learning by defining the framework, discussing some existing supervised learning methods, evaluation of these learning techniques and the concept of bias and variance in supervised learning. It also discusses various bias/variance error reduction techniques and thus enhancing the performance of supervised learning. It focuses on the use of resampling by bootstrap for variance reduction and lists existing studies done is this regard.

Part one of the thesis in concerned with discretization of continuous features. Chapter 3 lays the preliminaries of a discretization problem, discusses the types of discretizations and gives some definitions. It explains the concept of optimal partitioning and quality of an

obtained discretization. Finally, it describes the taxonomy and various existing discretization literature. Chapter 4 explains the motivation, concept and algorithm of our resampling based discretization techniques. Section 4.2 to 4.4 describes them in detail. Then in chapter 5, these techniques are thoroughly compared with various other existing and the optimal solution on 10 benchmark data sets.

Part two deals with fuzzy logic based learning through decision trees. Chapter 6 gives an introduction into fuzzy logic, fuzzy sets and fuzzy systems as a whole. It discusses the contributions of fuzzy theory and then describes fuzzy decision tree with an example and its components. Finally, it describes the state of the art and existing literature. Chapter 7 unveils our automatic fuzzy partitioning techniques used by fuzzy decision trees. Section 7.2 describes both the techniques in detail. Chapter 8 performs various experiments and analysis on various fuzzy decision tree methods compared with ours on existing data sets.

Part three of the thesis is based on semi-supervised learning using dimensionality reduction techniques such as manifold learning. It consists of chapter 9 which describes the concept of manifold learning and lays out the existing techniques in the literature. It discusses semi-supervised learning and then explains the concept behind our approach. Then it describes our algorithm concerning automatic setting of the value of K for KNN graphs and our overall classification technique using ISOMAP and decision trees. The end of the chapter performs certain comparisons with other classification schemes available and compares the performance.

Finally, chapter 10 concludes the thesis and gives perspectives and future directions regarding our work.