1 From Data Mining to Knowledge Discovery: An Overview

Usama M. Fayyad
Jet Propulsion Laboratory
California Institute of Technology

Gregory Piatetsky-Shapiro
GTE Laboratories

Padhraic Smyth
Jet Propulsion Laboratory
California Institute of Technology

We are drowning in information,
but starving for knowledge
– *John Naisbett*

Abstract

The explosive growth of many business, government, and scientific databases has far outpaced our ability to interpret and digest this data, creating a need for a new generation of tools and techniques for automated and intelligent database analysis. These tools and techniques are the subject of the rapidly emerging field of knowledge discovery in databases (KDD) and are the subject of this book. This chapter presents an overview of the state of the art in this field. We first clarify our view of the relation between knowledge discovery and data mining. We begin with a definition of the KDD process and basic data mining methods. We proceed to cover application issues in KDD including guidelines for selecting an application and current challenges facing practitioners in the field. The discussion relates methods and problems to applicable chapters in the book, with the goal of providing a unifying vision of the common overall goals shared by the chapters.

1.1 What Is this Book About?

In the last decade, we have seen an explosive growth in our capabilities to both generate and collect data. Advances in scientific data collection (e.g. from remote sensors or from space satellites), the widespread

introduction of bar codes for almost all commercial products, and the computerization of many business (e.g. credit card purchases) and government transactions (e.g. tax returns) have generated a flood of data. Advances in data storage technology, such as faster, higher capacity, and cheaper storage devices (e.g. magnetic disks, CD-ROMS), better database management systems, and data warehousing technology, have allowed us to transform this data deluge into "mountains" of stored data.

Representative examples are easy to find. In the business world, one of the largest databases in the world was created by Wal-Mart (a U.S. retailer) which handles over 20 million transactions a day (Babcock 1994). Most health-care transactions in the U.S. are being stored in computers, yielding multi-gigabyte databases, which many large companies are beginning to analyze in order to control costs and improve quality (e.g. see Matheus, Piatetsky-Shapiro, & McNeill, this volume). Mobil Oil Corporation, is developing a data warehouse capable of storing over 100 terabytes of data related to oil exploration (Harrison 1993).

There are huge scientific databases as well. The human genome database project (Fasman, Cuticchia, & Kingsbury 1994) has collected gigabytes of data on the human genetic code and much more is expected. A database housing a sky object catalog from a major astronomy sky survey (e.g. see Fayyad, Djorgovski, & Weir, this volume) consists of billions of entries with raw image data sizes measured in terabytes. The NASA Earth Observing System (EOS) of orbiting satellites and other spaceborne instruments is projected to generate on the order of 50 gigabytes of remotely sensed image data per *hour* when operational in the late 1990s and early in the next century (Way & Smith 1991).

Such volumes of data clearly overwhelm the traditional manual methods of data analysis such as spreadsheets and ad-hoc queries. Those methods can create informative reports from data, but cannot analyze the contents of those reports to focus on important knowledge. A significant need exists for a new generation of techniques and tools with the ability to *intelligently* and *automatically* assist humans in analyzing the mountains of data for nuggets of useful knowledge. These techniques and tools are the subject of the emerging field of knowledge discovery in databases (KDD).

The interest in KDD has been increasing, as evidenced by the number of recent workshops (Piatetsky-Shapiro 1991, Piatetsky-Shapiro 1993, Ziarko 1994, Fayyad & Uthurusamy 1994), which culminated in the

First International Conference on K
ing (Fayyad & Uthurusamy 1995).
have been devoted to the topic, i
Piatetsky-Shapiro 1992, Parsaye &
1993, Piatetsky-Shapiro et al 1994,
ious publications and special issue
applications which have been repor
government, medicine, and science
together the most recent relevant r
tradition of the first *Knowledge Dis*
Shapiro and Frawley 1991).

The chapter begins by discussin
data mining and the choice of title
the distinction between the terms
and explain how they fit together.
data mining refers to a class of meth
comprising the overall KDD process
in Section 1.2. The typical steps in
and discussed in Section 1.3. We th
methods in the context of the overa
general issues involved in data min
data mining methods. Having de
some of the methods, we turn our
issues of KDD in Section 1.6. Se
a preview of the rest of the chap
relate the discussion of particular
chapters with the goal of provid
overall goals shared by the chapte

1.1.1 About this Book's Tit

Historically the notion of finding
edge) in raw data has been give
discovery in databases, data min
discovery, information harvestin
processing. The term *knowledge*
was coined in 1989 to refer to the
data, and to emphasize the "hig
mining methods. The term *dat*

First International Conference on Knowledge Discovery and Data Mining (Fayyad & Uthurusamy 1995). A growing number of publications have been devoted to the topic, including (Inmon & Osterfelt 1991, Piatetsky-Shapiro 1992, Parsaye & Chignell 1993, Cercone & Tsuchiya 1993, Piatetsky-Shapiro et al 1994, Piatetsky-Shapiro 1995). These various publications and special issues document some of the many KDD applications which have been reported across diverse fields in business, government, medicine, and science (see Section 1.6). This book brings together the most recent relevant research in the field, continuing in the tradition of the first *Knowledge Discovery in Databases* book (Piatetsky-Shapiro and Frawley 1991).

The chapter begins by discussing the historical context of KDD and data mining and the choice of title for this book. We begin by explaining the distinction between the terms *data mining* and *knowledge discovery*, and explain how they fit together. The basic view we adopt is one where data mining refers to a class of methods that are used in some of the steps comprising the overall KDD process. We then provide a definition of KDD in Section 1.2. The typical steps involved in the KDD process are outlined and discussed in Section 1.3. We then focus in particular on data mining methods in the context of the overall KDD process. Section 1.4 covers the general issues involved in data mining while Section 1.5 discusses specific data mining methods. Having defined the basic terms and introduced some of the methods, we turn our attention to the practical application issues of KDD in Section 1.6. Section 1.7 concludes the chapter with a preview of the rest of the chapters in this volume. Throughout, we relate the discussion of particular methods and techniques to applicable chapters with the goal of providing a unifying vision of the common overall goals shared by the chapters constituting this book.

1.1.1 About this Book's Title

Historically the notion of finding useful patterns (or nuggets of knowledge) in raw data has been given various names, including knowledge discovery in databases, data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing. The term *knowledge discovery in databases*, or KDD for short, was coined in 1989 to refer to the broad process of finding knowledge in data, and to emphasize the "high-level" application of particular *data mining* methods. The term *data mining* has been commonly used by

statisticians, data analysts and the MIS (Management Information Systems) community, while KDD has been mostly used by artificial intelligence and machine learning researchers.

In this overview chapter we adopt the view that KDD refers to the overall *process* of discovering useful knowledge from data while *data mining* refers to the application of algorithms for extracting patterns from data without the additional steps of the KDD process (such as incorporating appropriate prior knowledge and proper interpretation of the results). These additional steps are essential to ensure that useful information (knowledge) is derived from the data. Blind application of data mining methods (rightly criticised as "fishing" or "dredging," and sometimes a "mining," in the statistical literature) can be a dangerous activity in that invalid patterns can be discovered without proper interpretation.

Thus, the overall *process* of finding and interpreting patterns from data is referred to as the KDD *process*, typically interactive and iterative, involving the repeated application of specific *data mining* methods or algorithms and the interpretation of the patterns generated by these algorithms. In sections to follow we will provide a more detailed definition of the overall KDD process and a more detailed look at specific data mining methods.

In combining the two terms "data mining" and "knowledge discovery" in the title of the book, we are attempting to build bridges between the statistical, database, and machine learning communities and appeal to a wider audience of information systems developers. The dual nature of the title reflects the contents of the book and the direction of the field, namely a focus on both types of issues: (i) the overall knowledge discovery process which includes preprocessing and postprocessing of data as well as interpretation of the discovered patterns as knowledge, and (ii) particular data mining methods and algorithms aimed solely at extracting patterns from raw data.

1.1.2 Links Between KDD and Related Fields

KDD is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization. KDD systems typically draw upon methods, algorithms, and techniques from these diverse fields. The unifying goal is extracting knowledge from data in the context of large databases.

In the fields of machine learning and pattern recognition, overlap with KDD lies in the study of theories and algorithms for systems which extract patterns and models from data (mainly data mining methods). KDD focuses on the extension of these theories and algorithms to the problem of finding special patterns (ones that may be interpreted as *useful or interesting knowledge*, see the next section) in large sets of real-world data. KDD also has much in common with statistics, particularly exploratory data analysis (EDA). KDD systems often embed particular statistical procedures for modeling data and handling noise within an overall knowledge discovery framework.

Machine discovery which targets the discovery of empirical laws from observation and experimentation (Shrager & Langley 1990), and *causal modeling* for the inference of causal models from data (Spirtes, Glymour, & Scheines 1993) are related research areas. Kloesgen & Zytkow (this volume) provide a glossary of terms common to KDD and machine discovery.

Another related area is *data warehousing*, which refers to the recently popular MIS trend for collecting and cleaning transactional data and making them available for on-line retrieval. A popular approach for analysis of data warehouses has been called OLAP (*on-line analytical processing*), after a new set of principles proposed by Codd (1993). OLAP tools focus on providing multi-dimensional data analysis, which is superior to SQL (standard query language) in computing summaries and breakdowns along many dimensions. We view both knowledge discovery and OLAP as related facets of a new generation of intelligent information extraction and management tools.

1.1.3 A Simple Illustrative Example

In the discussion of KDD and data mining methods in this chapter, we shall make use of a simple example to make some of the notions more concrete. Figure 1.1 shows a simple two-dimensional artificial data set consisting of 23 cases. Each point on the graph represents a person who has been given a loan by a particular bank at some time in the past. The horizontal axis represents the income of the person, the vertical axis represents the total personal debt of the person (mortgage, car payments, etc.). The data has been classified into 2 classes: the x's represent persons who have defaulted on their loans, the o's represent persons whose loans are in good status with the bank. Thus, this simple

artificial data set could represent a historical data set which may contain useful knowledge from the point of view of the bank making the loans. Note that in actual KDD applications there are typically many more dimensions (up to several hundreds) and many more data points (many thousands or even millions). The purpose here is to illustrate basic ideas on a small problem in 2-dimensional space.

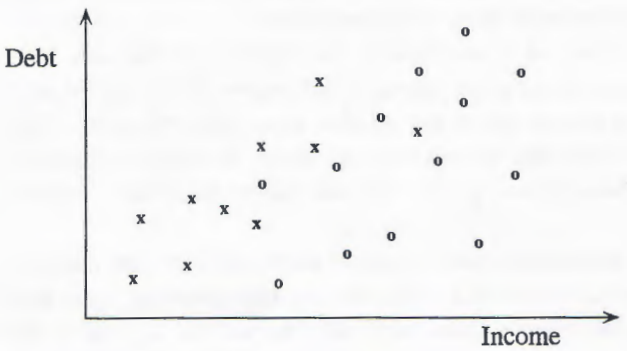

Figure 1.1
A simple data set with 2 classes used for illustrative purposes.

1.2 A Definition of Knowledge Discovery in Databases

To reflect the recent developments and growth in KDD, we have revised the definition of KDD given in (Frawley, Piatetsky-Shapiro, & Matheus 1991). We first start with a general statement of this definition in words:

> *Knowledge discovery in databases* is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

Let us examine these terms in more detail.

- *Data* is a set of facts F (e.g., cases in a database). In our simple example of Figure 1.1, F is the collection of 23 cases with three fields each containing the values of *debt*, *income*, and *loan status*.

- *Pattern* is an expression E in a language L describing facts in a subset F_E of F. E is called a pattern if it is simpler (in some sense, see below) than the enumeration of all facts in F_E. For example, the pattern: "If income $<$ \$$t$, then person has defaulted on the loan" would be one such pattern for an appropriate choice of t. This pattern is illustrated graphically in Figure 1.2.

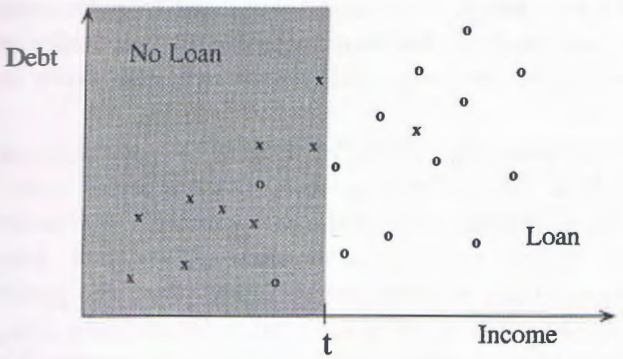

Figure 1.2
Using a single threshold on the income variable to try to classify the loan data set.

- *Process:* Usually in KDD process is a multi-step process, which involves data preparation, search for patterns, knowledge evaluation, and refinement involving iteration after modification. The process is assumed to be non-trivial—that is, to have some degree of search autonomy. For example, computing the mean income of persons in the loan example, while producing a useful result, does not qualify as discovery.
- *Validity:* The discovered patterns should be valid on new data with some degree of certainty. A measure of certainty is a function C mapping expressions in L to a partially or totally ordered measurement space M_C. An expression E in L about a subset $F_E \subset F$ can be assigned a certainty measure $c = C(E, F)$. For example, if the boundary for the pattern shown in Figure 1.2 is moved to the right, its certainty measure would drop since more good loans would be admitted into the shaded region (no loan).
- *Novel:* The patterns are novel (at least to the system). Novelty can be measured with respect to changes in data (by comparing

current values to previous or expected values) or knowledge (how a new finding is related to old ones). In general, we assume this can be measured by a function $N(E, F)$, which can be a boolean function or a measure of degree of novelty or unexpectedness.

- *Potentially Useful:* The patterns should potentially lead to some useful actions, as measured by some utility function. Such a function U maps expressions in L to a partially or totally ordered measure space M_U: hence, $u = U(E, F)$. For example, in the loan data set this function could be the expected increase in profits to the bank (in dollars) associated with the decision rule shown in Figure 1.2.
- *Ultimately Understandable:* A goal of KDD is to make patterns understandable to humans in order to facilitate a better understanding of the underlying data. While this is difficult to measure precisely, one frequent substitute is the simplicity measure. Several measures of simplicity exist, and they range from the purely syntactic (e.g., the size of a pattern in bits) to the semantic (e.g., easy for humans to comprehend in some setting). We assume this is measured, if possible, by a function S mapping expressions E in L to a partially or totally ordered measure space M_S: hence, $s = S(E, F)$.

An important notion, called *interestingness*, is usually taken as an overall measure of pattern value, combining validity, novelty, usefulness, and simplicity. Some KDD systems have an explicit interestingness function $i = I(E, F, C, N, U, S)$ which maps expressions in L to a measure space M_I. Other systems define interestingness indirectly via an ordering of the discovered patterns.

Given the notions listed above, we may state our definition of knowledge as viewed from the narrow perspective of KDD as used in this book. This is by no means an attempt to define "knowledge" in the philosophical or even the popular view. The purpose of this definition is specify what an algorithm used in a KDD process may consider knowledge.

- *Knowledge:* A pattern $E \in L$ is called knowledge if for some user-specified threshold $i \in M_I$, $I(E, F, C, N, U, S) > i$.

Note that this definition of knowledge is by no means absolute. As a matter of fact, it is purely user-oriented, and determined by whatever

functions and thresholds the user chooses. For example, one instantiation of this definition is to select some thresholds $c \in M_C$, $s \in M_S$, and $u \in M_u$, and calling a pattern E knowledge if and only if

$$C(E,F) > c \text{ and } S(E,F) > s \text{ and } U(S,F) > u.$$

By appropriate settings of thresholds, one can emphasize accurate predictors or useful (by some cost measure) patterns over others. Clearly, there is an infinite space of how the mapping I can be defined. Such decisions are left to the user and the specifics of the domain.

> *Data Mining* is a step in the KDD process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, produces a particular enumeration of patterns E_j over F (see Sections 1.4 and 1.5 for more details)

Note that the space of patterns is often infinite, and the enumeration of patterns involves some form of search in this space. The computational efficiency constraints place severe limits on the subspace that can be explored by the algorithm.

> *KDD Process* is the process of using data mining methods (algorithms) to extract (identify) what is deemed knowledge according to the specifications of measures and thresholds, using the database F along with any required preprocessing, subsampling, and transformations of F.

The data mining component of the KDD process is mainly concerned with means by which patterns are extracted and enumerated from the data. Knowledge discovery involves the *evaluation* and possibly *interpretation* of the patterns to make the decision of what constitutes knowledge and what does not. It also includes the choice of encoding schemes, preprocessing, sampling, and projections of the data prior to the data mining step.

1.3 The KDD Process

The KDD process is interactive and iterative, involving numerous steps with many decisions being made by the user. Brachman & Anand (this

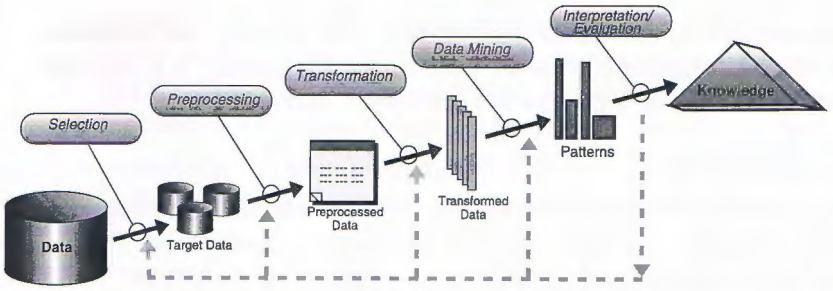

Figure 1.3
An overview of the steps comprising the KDD process.

volume) give a practical view of the KDD process emphasizing the interactive nature of the process. Here we broadly outline some of its basic steps:

1. Developing an understanding of the application domain, the relevant prior knowledge, and the goals of the end-user.
2. 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.
3. Data cleaning and preprocessing: basic operations such as the removal of noise or outliers if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, accounting for time sequence information and known changes.
4. Data reduction and projection: 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.
5. Choosing the data mining task: deciding whether the goal of the KDD process is classification, regression, clustering, etc. The various possible *tasks* of a data mining algorithm are described in more detail in Section 1.4.1.
6. Choosing the data mining algorithm(s): selecting method(s) to be used for searching for patterns in the data. This includes deciding which models and parameters may be appropriate (e.g. models

for categorical data are di
reals) and matching a pa
overall criteria of the KD
more interested in unders
capabilities—see Section 1
7. Data mining: searching f
representational form or a s
rules or trees, regression, c
for details). The user can s
by correctly performing th
8. Interpreting mined patter
for further iteration.
9. Consolidating discovered k
into the performance syste
ing it to interested parties.
solving potential conflicts
knowledge.

The KDD process can involv
loops between any two steps. Th
potential multitude of iterations
Most previous work on KDD ha
However, the other steps are of c
ful application of KDD in pract
Anand, this volume, for a more

Having defined the basic notio
now focus on the data mining co
most attention in the literature.

1.4 An Overview of Data

The data mining component of th
iterative application of particular
this section is to present a unifie
data mining methods in current
models loosely throughout this
as instantiation of a model, e.g.
$f(x) = \alpha x^2 + \beta x$ is considered a

for categorical data are different than models on vectors over the reals) and matching a particular data mining method with the overall criteria of the KDD process (e.g. the end-user may be more interested in understanding the model than its predictive capabilities—see Section 1.4.2).

7. Data mining: searching for patterns of interest in a particular representational form or a set of such representations: classification rules or trees, regression, clustering, and so forth (see Section 1.5 for details). The user can significantly aid the data mining method by correctly performing the preceding steps.
8. Interpreting mined patterns, possible return to any of steps 1–7 for further iteration.
9. Consolidating discovered knowledge: incorporating this knowledge into the performance system, or simply documenting it and reporting it to interested parties. This also includes checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

The KDD process can involve significant iteration and may contain loops between any two steps. The basic flow of steps (although not the potential multitude of iterations and loops) are illustrated in Figure 1.3. Most previous work on KDD has focused on step 7—the data mining. However, the other steps are of considerable importance for the successful application of KDD in practice. See the chapter by Brachman & Anand, this volume, for a more elaborate account of this aspect.

Having defined the basic notions and introduced the KDD process, we now focus on the data mining component, which has by far received the most attention in the literature.

1.4 An Overview of Data Mining Methods

The data mining component of the KDD process often involves repeated iterative application of particular data mining methods. The objective of this section is to present a unified overview of some of the most popular data mining methods in current use. We use the terms *patterns* and *models* loosely throughout this chapter: a pattern can be thought of as instantiation of a model, e.g., $f(x) = 3x^2 + x$ is a pattern whereas $f(x) = \alpha x^2 + \beta x$ is considered a model.

Data mining involves fitting models to, or determining patterns from, observed data. The fitted models play the role of inferred knowledge: whether or not the models reflect *useful* or *interesting* knowledge is part of the overall, interactive KDD process where subjective human judgment is usually required. There are two primary mathematical formalisms used in model fitting: the *statistical* approach allows for nondeterministic effects in the model (for example, $f(x) = \alpha x + e$ where e could be a Gaussian random variable), whereas a *logical* model is purely deterministic ($f(x) = \alpha x$) and does not admit the possibility of uncertainty in the modeling process. We will focus primarily on the statistical/probabilistic approach to data mining: this tends to be the most widely-used basis for practical data mining applications given the typical uncertainty about the exact nature of real-world data-generating processes. See the chapter by Elder & Pregibon (this volume) for a perspective from the field of statistics.

Most data mining methods are based on concepts from machine learning, pattern recognition and statistics: classification, clustering, graphical models, and so forth. The array of different algorithms for solving each of these problems can often be quite bewildering to both the experienced data analyst and the novice. In this section we offer a brief overview of data mining methods and in particular try to convey the notion that most (if not all) methods can be viewed as extensions or hybrids of a few basic techniques and principles.

The section begins by discussing the primary tasks of data mining and then shows that the data mining methods to address these tasks consist of three primary algorithmic components: *model representation*, *model evaluation*, and *search*. The section concludes by discussing particular data mining algorithms within this framework.

1.4.1 The Primary Tasks of Data Mining

The two "high-level" primary goals of data mining in practice tend to be *prediction* and *description*. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. Description focuses on finding human-interpretable patterns describing the data. The relative importance of prediction and description for particular data mining applications can vary considerably. However, in the context of KDD, description tends to be more important than prediction. This is in contrast to pattern recognition

and machine learning applications (such as speech recognition) where prediction is often the primary goal (see Lehmann 1990, for a discussion from a statistical perspective).

The goals of prediction and description are achieved by using the following primary data mining tasks.

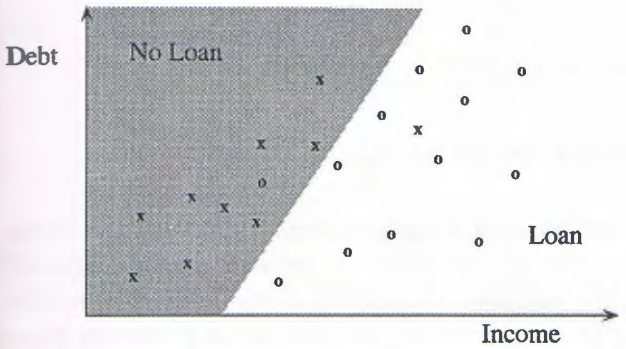

Figure 1.4
A simple linear classification boundary for the loan data set: shaded region denotes class "no loan."

- *Classification* is learning a function that maps (classifies) a data item into one of several predefined classes (Hand 1981; Weiss & Kulikowski 1991; McLachlan 1992). Examples of classification methods used as part of knowledge discovery applications include classifying trends in financial markets (Apte & Hong, this volume) and automated identification of objects of interest in large image databases (Fayyad, Djorgovski, & Weir, this volume). Figure 1.4 shows a simple partitioning of the loan data into two class regions—note that it is not possible to separate the classes perfectly using a linear decision boundary. The bank might wish to use the classification regions to automatically decide whether future loan applicants will be given a loan or not.
- *Regression* is learning a function which maps a data item to a real-valued prediction variable. Regression applications are many, e.g., predicting the amount of biomass present in a forest given

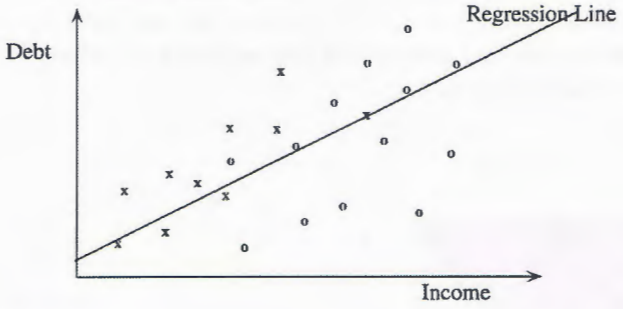

Figure 1.5
A simple linear regression for the loan data set.

remotely-sensed microwave measurements, estimating the probability that a patient will die given the results of a set of diagnostic tests, predicting consumer demand for a new product as a function of advertising expenditure, and time series prediction where the input variables can be time-lagged versions of the prediction variable. Figure 1.5 shows the result of simple linear regression where "total debt" is fitted as a linear function of "income": the fit is poor since there is only a weak correlation between the two variables.

- *Clustering* is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data (Titterington, Smith & Makov 1985: Jain & Dubes 1988). The categories may be mutually exclusive and exhaustive, or consist of a richer representation such as hierarchical or overlapping categories. Examples of clustering applications in a knowledge discovery context include discovering homogeneous sub-populations for consumers in marketing databases and identification of sub-categories of spectra from infra-red sky measurements (Cheeseman & Stutz, this volume). Figure 1.6 shows a possible clustering of the loan data set into 3 clusters: note that the clusters overlap allowing data points to belong to more than one cluster. The original class labels (denoted by x's and o's in the previous figures) have been replaced by +'s to indicate that the class membership is no longer assumed

known. Closely related to clustering is the task of *probability density estimation* which consists of techniques for estimating from data the joint multi-variate probability density function of all of the variables/fields in the database (Silverman 1986).

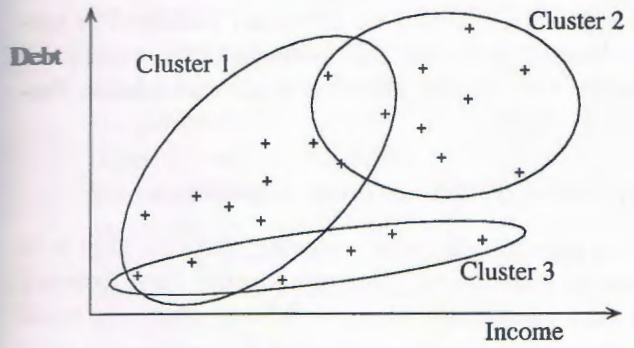

Figure 1.6
A simple clustering of the loan data set into 3 clusters. Note that original labels are replaced by +'s.

- *Summarization* involves methods for finding a compact description for a subset of data. A simple example would be tabulating the mean and standard deviations for all fields. More sophisticated methods involve the derivation of summary rules (Agrawal et al., this volume), multivariate visualization techniques, and the discovery of functional relationships between variables (Zembowicz & Zytkow, this volume). Summarization techniques are often applied to interactive exploratory data analysis and automated report generation.
- *Dependency Modeling* consists of finding a model which describes significant *dependencies* between variables. Dependency models exist at two levels: the *structural* level of the model specifies (often in graphical form) which variables are locally dependent on each other, whereas the *quantitative* level of the model specifies the strengths of the dependencies using some numerical scale. For example, probabilistic dependency networks use conditional independence to specify the structural aspect of the model and proba-

bilities or correlations to specify the strengths of the dependencies (Heckerman, this volume; Glymour et al., 1987). Probabilistic dependency networks are increasingly finding applications in areas as diverse as the development of probabilistic medical expert systems from databases, information retrieval, and modeling of the human genome.

- *Change and Deviation Detection* focuses on discovering the most significant changes in the data from previously measured or normative values (Berndt & Clifford, this volume; Guyon et al., this volume; Kloesgen, this volume; Matheus et al., this volume; Basseville & Nikiforov 1993).

1.4.2 The Components of Data Mining Algorithms

Having outlined the primary tasks of data mining, the next step is to construct algorithms to solve them. One can identify three primary components in any data mining algorithm: *model representation*, *model evaluation*, and *search*. This reductionist view is not necessarily complete or fully encompassing: rather, it is a convenient way to express the key concepts of data mining algorithms in a relatively unified and compact manner (Cheeseman (1990) outlines a similar structure).

- *Model Representation* is the language L for describing discoverable patterns. If the representation is too limited, then no amount of training time or examples will produce an accurate model for the data. For example, a decision tree representation, using univariate (single-field) node-splits, partitions the input space into hyperplanes which are parallel to the attribute axes. Such a decision-tree method cannot discover from data the formula $x = y$ no matter how much training data it is given. Thus, it is important that a data analyst fully comprehend the representational assumptions which may be inherent to a particular method. It is equally important that an algorithm designer clearly state which representational assumptions are being made by a particular algorithm. Note that more powerful representational power for models increases the danger of overfitting the training data resulting in reduced prediction accuracy on unseen data. In addition the search becomes much more complex and interpretation of the model is typically more difficult.

- *Model Evaluation* estimates how well a particular pattern (a model and its parameters) meet the criteria of the KDD process. Evaluation of predictive accuracy (validity) is based on cross validation. Evaluation of descriptive quality involves predictive accuracy, novelty, utility, and understandability of the fitted model. Both logical and statistical criteria can be used for model evaluation. For example, the maximum likelihood principle chooses the parameters for the model which yield the best fit to the training data.
- *Search Method* consists of two components: *Parameter Search* and *Model Search*. In *parameter search* the algorithm must search for the parameters which optimize the model evaluation criteria given observed data and a fixed model representation. For relatively simple problems there is no search: the optimal parameter estimates can be obtained in closed form. Typically, for more general models, a closed form solution is not available: greedy iterative methods are commonly used, e.g., the gradient descent method of backpropagation for neural networks. *Model Search* occurs as a loop over the parameter search method: the model representation is changed so that a family of models are considered. For each specific model representation, the parameter search method is instantiated to evaluate the quality of that particular model. Implementations of model search methods tend to use heuristic search techniques since the size of the space of possible models often prohibits exhaustive search and closed form solutions are not easily obtainable.

1.5 A Discussion of Popular Data Mining Methods

There exist a wide variety of data mining methods: here we only focus on a subset of popular techniques. Each method is discussed in the context of model representation, model evaluation, and search.

1.5.1 Decision Trees and Rules

Decision trees and rules that use univariate splits have a simple representational form, making the inferred model relatively easy to comprehend by the user. However, the restriction to a particular tree or rule representation can significantly restrict the functional form (and thus the

approximation power) of the model. For example, Figure 1.2 illustrates the effect of a threshold "split" applied to the income variable for loan data set: it is clear that using such simple threshold splits (parallel to the feature axes) severely limit the type of classification boundaries which can be induced. If one enlarges the model space to allow more general expressions (such as multivariate hyperplanes at arbitrary angles), then the model is more powerful for prediction but may be much more difficult to comprehend. There are a large number of decision tree and rule induction algorithms described in the machine learning and applied statistics literature (Breiman et al 1984; Quinlan 1992).

To a large extent they are based on likelihood-based model evaluation methods with varying degrees of sophistication in terms of penalizing model complexity. Greedy search methods, which involve growing and pruning rule and tree structures, are typically employed to explore the super-exponential space of possible models. Trees and rules are primarily used for predictive modeling, both for classification (Apte & Hong, this volume; Fayyad, Djorgovski, & Weir, this volume) and regression, although they can also be applied to summary descriptive modeling (Agrawal et al., this volume).

1.5.2 Nonlinear Regression and Classification Methods

These methods consist of a family of techniques for prediction which fit linear and non-linear combinations of basis functions (sigmoids, splines, polynomials) to combinations of the input variables. Examples include feedforward neural networks, adaptive spline methods, projection pursuit regression, and so forth (see Friedman (1989), Cheng & Titterington (1994), and Elder & Pregibon (this volume) for more detailed discussions). Consider neural networks, for example. Figure 1.7 illustrates the type of non-linear decision boundary which a neural network might find for the loan data set. In terms of model evaluation, while networks of the appropriate size can universally approximate any smooth function to any desired degree of accuracy, relatively little is known about the representation properties of *fixed* size networks estimated from *finite* data sets. In terms of model evaluation, the standard squared error and cross entropy loss functions used to train neural networks can be viewed as log-likelihood functions for regression and classification respectively (Geman, Bienenstock & Doursat 1992; Ripley 1994). Backpropagation is a parameter search method which performs gradient descent in param-

eter (weight) space to find a local maximum of the likelihood function starting from random initial conditions. Nonlinear regression methods, though powerful in representational power, can be very difficult to interpret. For example, while the classification boundaries of Figure 1.7 may be more accurate than the simple threshold boundary of Figure 1.2, the threshold boundary has the advantage that the model can be expressed as a simple rule of the form "if income is greater than threshold t then loan will have good status" to some degree of certainty.

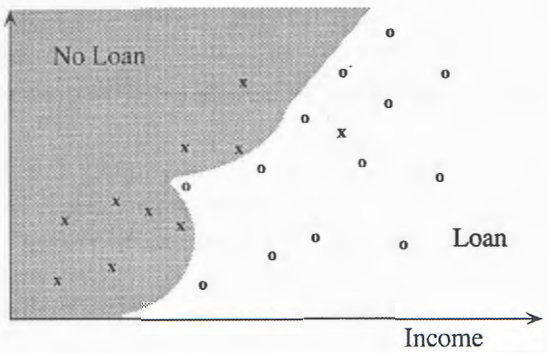

Figure 1.7
An example of classification boundaries learned by a non-linear classifier (such as a neural network) for the loan data set.

1.5.3 Example-based Methods

The representation is simple: use representative examples from the database to approximate a model, i.e., predictions on new examples are derived from the properties of "similar" examples in the model whose prediction is known. Techniques include nearest-neighbor classification and regression algorithms (Dasarathy 1991) and case-based reasoning systems (Kolodner, 1993). Figure 1.8 illustrates the use of a nearest neighbor classifier for the loan data set: the class at any new point in the 2-dimensional space is the same as the class of the closest point in the original training data set.

A potential disadvantage of example-based methods (compared with tree-based methods for example) is that a well-defined distance metric

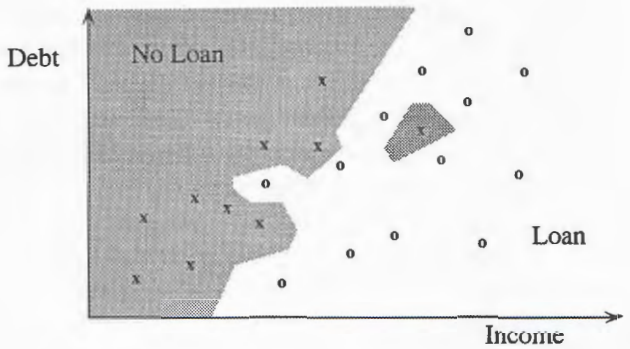

Figure 1.8
Classification boundaries for a nearest neighbor classifier for the loan data set.

for evaluating the distance between data points is required. For the loan data in Figure 1.8 this would not be a problem since income and debt are measured in the same units: but if one wished to include a variable such as the duration of the loan, then it would require more effort to define a sensible metric between the variables. Model evaluation is usually based on cross-validation estimates (Weiss & Kulikowski, 1991) of a prediction error: "parameters" of the model to be estimated can include the number of neighbors to use for prediction and the distance metric itself. Like non-linear regression methods, example-based methods are often asymptotically quite powerful in terms of approximation properties, but conversely can be difficult to interpret since the model is implicit in the data and not explicitly formulated. Related techniques include kernel density estimation (Silverman 1986) and mixture modeling (Titterington, Smith, & Makov 1985).

1.5.4 Probabilistic Graphical Dependency Models

Graphical models specify the probabilistic dependencies which underlie a particular model using a graph structure (Pearl 1988; Whittaker, 1990). In its simplest form, the model specifies which variables are directly dependent on each other. Typically these models are used with categorical or discrete-valued variables, but extensions to special cases, such as Gaussian densities, for real-valued variables are also possible.

Within the artificial intelligence and statistical communities these models were initially developed within the framework of probabilistic expert systems: the structure of the model and the parameters (the conditional probabilities attached to the links of the graph) were elicited from experts. More recently there has been significant work in both the AI and statistical communities on methods whereby both the structure and parameters of graphical models can be learned from databases directly (Buntine, this volume; Heckerman, this volume). Model evaluation criteria are typically Bayesian in form and parameter estimation can be a mixture of closed form estimates and iterative methods depending on whether a variable is directly observed or hidden. Model search can consist of greedy hill-climbing methods over various graph structures. Prior knowledge, such as a partial ordering of the variables based on causal relations, can be quite useful in terms of reducing the model search space. Although still primarily at the research phase, graphical model induction methods are of particular interest to KDD since the graphical form of the model lends itself easily to human interpretation.

1.5.5 Relational Learning Models

While decision-trees and rules have a representation restricted to propositional logic, relational learning (also known as inductive logic programming) uses the more flexible pattern language of first-order logic. A relational learner can easily find formulas such as X=Y. Most research so far on model evaluation methods for relational learning are logical in nature. The extra representational power of relational models comes at the price of significant computational demands in terms of search. See Dzeroski (this volume) for a more detailed discussion.

Given the broad spectrum of data mining methods and algorithms, our brief overview is inevitably limited in scope: there are many data mining techniques, particularly specialized methods for particular types of data and domains, which were not mentioned specifically in the discussion. We believe the general discussion on data mining tasks and components has general relevance to a variety of methods. For example, consider time series prediction: traditionally this has been cast as a predictive regression task (autoregressive models and so forth). Recently, more general models have been developed for time series applications such as non-linear basis function, example-based, and kernel methods. Furthermore, there has been significant interest in *descriptive* graphi-

cal and local data modeling of time series rather than purely *predictive* modeling (Weigend & Gershenfeld 1993). Thus, although different algorithms and applications may appear quite different on the surface, it is not uncommon to find that they share many common components. Understanding data mining and model induction at this component level clarifies the task of any data mining algorithm and makes it easier for the user to understand its overall contribution and applicability to the KDD process.

We would like to remind the reader that our discussion and overview of data mining methods has been both cursory and brief. There are two important points we would like to make clear:

1. *Automated Search:* Our brief overview has focused mainly on automated methods for extracting patterns and/or models from data. While this is consistent with the definition we gave earlier, it does not necessarily represent what other communities might refer to as *data mining*. For example, some use the term to designate any manual search of the data, or search assisted by queries to a DBMS or humans visualizing patterns in data as data mining. In other communities, it is used to refer to the automated correlation of data from transactions or the automated generation of transaction reports. We choose to focus only on methods that contain certain degrees of search autonomy.
2. *Beware the Hype:* The state-of-the-art in automated methods in data mining is still in a fairly early stage of development. There are no established criteria for deciding which methods to use in which circumstances, and many of the approaches are based on crude heuristic approximations to avoid the expensive search required to find optimal or even good solutions. hence, the reader should be careful when confronted with overstated claims about the great ability of a system to mine useful information from large (or even small) databases.

1.6 Application Issues

In the business world, the most successful and widespread application of KDD is "Database Marketing," which is a method of analyzing customer databases, looking for patterns among existing customer preferences and

using those patterns for more targeted selection of future customers. Business Week, a popular business magazine in the United States, carried a cover story on Database Marketing (Berry 1994) that estimated that over 50% of all retailers are using or planning to use database marketing. The reason is simple—significant results can be obtained using this approach: e.g. a 15-20% percent increase in credit-card purchases reported by American Express (Berry 1994).

Another major business use of data mining methods is the analysis and selection of stocks and other financial instruments. There are already numerous investment companies (Barr and Mani 1994) which pick stocks using a variety of advanced data mining methods.

Several successful applications have been developed for analysis and reporting on change in data. These include Coverstory from IRI (Schmitz, Armstrong, & Little 1990), Spotlight from A.C. Nielsen (Anand & Kahn 1992) for supermarket sales data, and KEFIR from GTE, for health care databases (*Matheus, Piatetsky-Shapiro, & McNeil, this volume*)

Fraud detection and prevention is another area where KDD plays a role. While there have been many applications, published information is, for obvious reasons, not readily available. Here we mention just a few noteworthy examples. A system for detecting healthcare provider fraud in electronically submitted claims, has been developed at Travelers Insurance by Major and Riedinger (1992). The Internal Revenue Service has developed a pilot system for selecting tax returns for audits. Neural network based tools, such as Nestor FDS (Blanchard 1994) have been developed for detecting credit-card fraud and are reportedly watching millions of accounts.

A number of interesting and important scientific applications of KDD have also been developed. Example application areas in science include

- *Astronomy:* The SKICAT system from JPL/Caltech is used by astronomers to automatically identify stars and galaxies in a large-scale sky survey for cataloging and scientific analysis (see Fayyad, Djorgovski, & Weir, this volume).
- *Molecular Biology:* Systems have been developed for finding patterns in molecular structures (Conklin, Fortier, and Glasgow 1993) and in genetic data (Holder, Cook, and Djoko 1994).
- *Global Climate Change Modeling:* Spatio-temporal patterns such

as cyclones are automatically found from large simulated and observational datasets (Stolorz et al. 1994).

Other recent applications are described in (Fayyad & Uthurusamy 1994–1995, Piatetsky-Shapiro 1993).

1.6.1 Guidelines for Selecting a Potential KDD Application

The criteria for selecting applications can be divided into practical and technical. The practical criteria for KDD projects are similar to those for other application of advanced technology, while the technical ones are more specific to KDD.

Practical criteria include consideration of the **potential for significant impact of an application**. For business applications this could be measured by criteria such as greater revenue, lower costs, higher quality, or savings in time. For scientific applications the impact can be measured by the novelty and quality of the discovered knowledge and by increased access to data via automating manual analysis processes. Another important practical consideration is that **no good alternatives exist**: the solution is not easily obtainable by other standard means. Hence the ultimate user has a strong vested interest in insuring the success of the KDD venture. **Organizational support** is another consideration: there should be a champion for using new technology; e.g. a domain expert who can define a proper interestingness measure for that domain as well as participate in the KDD process. Finally, an important practical consideration is the **potential for privacy/legal issues**. This applies primarily to databases on people where one needs to guard against the discovered patterns raising legal or ethical issues of invasion of privacy.

Technical criteria include considerations such as the **availability of sufficient data (cases)**. The number of examples (cases) required for reliable inference of useful patterns from data varies a great deal with each particular application. In general, the more fields there are and the more complex are the patterns being sought, the more data are needed. However, strong prior knowledge (see below) can reduce the number of needed cases significantly. Another consideration is the **relevance of attributes.** It is important to have data attributes relevant to the discovery task: no amount of data will allow prediction based on attributes that do not capture the required information.

Furthermore, **low noise levels (few data errors)** is another consideration. High amounts of noise make it hard to identify patterns unless a large number of cases can mitigate random noise and help clarify the aggregate patterns. A related consideration is whether one can attach **confidence intervals** to extracted knowledge. In some applications, it is crucial to attach confidence intervals to predictions produced by the KDD system. This allows the user to calibrate actions appropriately.

Finally, and perhaps one of the most important considerations is **prior knowledge**. It is very useful to know something about the domain—what are the important fields, what are the likely relationships, what is the user utility function, what patterns are already known, and so forth. Prior knowledge can significantly reduce the search in the data mining step and all the other steps in the KDD process.

1.6.2 Privacy and Knowledge Discovery

When dealing with databases of personal information, governments and businesses have to be careful to adequately address the legal and ethical issues of invasion of privacy. Ignoring this issue can be dangerous, as Lotus found in 1990, when they were planning to introduce a CD-ROM with data on about 100 million American households. The stormy protest led to the withdrawal of that product (Rosenberg 1992).

Current discussion centers around guidelines for what constitutes a proper discovery. The Organization for Economic Cooperation and Development (OECD) guidelines for data privacy (O'Leary 1995), which have been adopted by most European Union countries, suggest that data about specific living individuals should not be analyzed without their consent. They also suggest that the data should only be collected for a specific purpose. Use for other purposes is possible only with the consent of the data subject or by authority of the law.

In the U.S. there is ongoing work on draft principles for fair information use related to the National Information Infrastructure (NII), commonly known as the "information superhighway." These principles permit the use of "transactional records," such as phone numbers called, credit card payments, etc., as long as such use is compatible with the original notice. The use of transactional records can be seen to also include discovery of patterns.

In many cases (e.g. medical research, socio-economic studies) the goal is to discover patterns about groups, not individuals. While group pat-

tern discovery appears not to violate the restrictions on personal data retrieval, an ingenious combination of several group patterns, especially in small datasets, may allow identification of specific personal information. Solutions which allow group pattern discovery while avoiding the potential invasion of privacy include removal or replacement of identifying fields, performing queries on random subsets of data, and combining individuals into groups and allowing only queries on groups. These and related issues are further discussed in (Piatetsky-Shapiro 1995b).

1.6.3 Research and Application Challenges for KDD

We outline some of the current primary research and application challenges for knowledge discovery. This list is by no means exhaustive. The goal is to give the reader a feel for the types of problems that KDD practitioners wrestle with. We point to chapters in this book that are of relevance to the challenges we list.

- *Larger databases.* Databases with hundreds of fields and tables, millions of records, and multi-gigabyte size are quite commonplace, and terabyte (10^{12} bytes) databases are beginning to appear. For example, Agrawal et al (this volume) present efficient algorithms for enumerating all association rules exceeding given confidence thresholds over large databases. Other possible solutions include sampling, approximation methods, and massively parallel processing (Holsheimer et al, this volume).
- *High dimensionality.* Not only is there often a very large number of records in the database, but there can also be a very large number of fields (attributes, variables) so that the dimensionality of the problem is high. A high dimensional data set creates problems in terms of increasing the size of the search space for model induction in a combinatorially explosive manner. In addition, it increases the chances that a data mining algorithm will find spurious patterns that are not valid in general. Approaches to this problem include methods to reduce the effective dimensionality of the problem and the use of prior knowledge to identify irrelevant variables.
- *Overfitting.* When the algorithm searches for the best parameters for one particular model using a limited set of data, it may overfit the data, resulting in poor performance of the model on test data. Possible solutions include cross-validation, regularization,

and other sophisticated statistical strategies (Elder & Pregibon, this volume).

- *Assessing statistical significance.* A problem (related to overfitting) occurs when the system is searching over many possible models. For example, if a system tests N models at the 0.001 significance level, then on average, with purely random data, $N/1000$ of these models will be accepted as significant. This point is frequently missed by many initial attempts at KDD. One way to deal with this problem is to use methods which adjust the test statistic as a function of the search, e.g., Bonferroni adjustments for independent tests.
- *Changing data and knowledge.* Rapidly changing (non-stationary) data may make previously discovered patterns invalid. In addition, the variables measured in a given application database may be modified, deleted, or augmented with new measurements over time. Possible solutions include incremental methods for updating the patterns and treating change as an opportunity for discovery by using it to cue the search for patterns of change only (Matheus et al, this volume).
- *Missing and noisy data.* This problem is especially acute in business databases. U.S. census data reportedly has error rates of up to 20%. Important attributes may be missing if the database was not designed with discovery in mind. Possible solutions include more sophisticated statistical strategies to identify hidden variables and dependencies (Heckerman, this volume; Smyth et al., this volume).
- *Complex relationships between fields.* Hierarchically structured attributes or values, relations between attributes, and more sophisticated means for representing knowledge about the contents of a database will require algorithms that can effectively utilize such information. Historically, data mining algorithms have been developed for simple attribute-value records, although new techniques for deriving relations between variables are being developed (Dzeroski, this volume; Han and Fu, this volume).
- *Understandability of patterns.* In many applications it is important to make the discoveries more understandable by humans. Possible solutions include graphical representations (Buntine, this

volume; Heckerman, this volume), rule structuring with directed acyclic graphs (Gaines, this volume), natural language generation (Matheus et al., this volume), and techniques for visualization of data and knowledge. Rule refinement strategies (e.g. Major 1993; Kloesgen 1993) can be used to address a related problem: the discovered knowledge may be implicitly or explicitly redundant.

- *User interaction and prior knowledge.* Many current KDD methods and tools are not truly *interactive* and cannot easily incorporate prior knowledge about a problem except in simple ways. The use of domain knowledge is important in all of the steps of the KDD process as outlined in Section 1.3. Bayesian approaches (e.g. Cheeseman & Stutz, this volume) use prior probabilities over data and distributions as one form of encoding prior knowledge. Simoudis et al (this volume) make use of deductive databases to discover knowledge that is then used to guide the data mining search.
- *Integration with other systems.* A stand-alone discovery system may not be very useful. Typical integration issues include integration with a DBMS (e.g. via a query interface), integration with spreadsheets and visualization tools, and accommodating real-time sensor readings. Examples of integrated KDD systems are described by Simoudis et al (this volume) and Shen et al (this volume).

1.7 Organization of this Book

The chapters of this book span fundamental issues of knowledge discovery, classification and clustering, trend and deviation analysis, dependency derivation, integrated discovery systems, augmented database systems, and application case studies. The contributing authors include researchers and practitioners in academia, government laboratories, and private industry, indicating the breadth of interest in the field. We have organized the book into seven parts and an appendix.

Part I deals with fundamental issues in discovery. Brachman and Anand outline the state-of-the-practice of the KDD process. Buntine presents a unifying view of various data mining techniques under the broad area of graphical models. Elder and Pregibon provide the reader

a general statistical perspective on knowledge discovery and data mining.

Part II deals with specific techniques for data mining. Dzeroski presents an overview of recent developments relevant to KDD in inductive logic programming (ILP). Cheeseman and Stutz present a Bayesian approach to clustering and discuss the details of the AutoClass system. AutoClass attempts to infer the most likely number of classes in the data, and the most likely parameterization of the probability distributions chosen to model the data. Guyon, Matic, and Vapnik present a novel approach for discovering informative patterns within a supervised learning framework and describe the application of their techniques to "data cleaning" of large optical character recognition databases. Gaines discusses the use of exception directed acyclic graphs (EDAGs) for efficient representation of induced knowledge.

Part III presents methods for dealing with trend and deviation analysis. Berndt and Clifford show how to adapt dynamic time warping (a dynamic programming technique used in speech recognition) to finding patterns in time series data. Kloesgen describes Explora, a multistrategy discovery assistant, and examines the options for discovering different types of deviations and other patterns.

Part IV focuses on data mining techniques for deriving dependencies. Heckerman provides a survey of current research in the field of learning graphical models (also known as Bayesian networks) from data: graphical models provide an efficient framework for representing and reasoning with joint probability distributions over multiple variables. Agrawal, Mannila, Srikant, Toivonen and Verkamo introduce a variety of novel extensions of earlier work on deriving association rules from transaction data: empirical results demonstrate that the new algorithms are much more efficient than previous versions. Zembowicz and Zytkow show how to use contingency tables to discover different types of knowledge, including dependencies and taxonomies.

Part V focuses on integrated discovery systems which include multiple components, employ several data mining techniques, and generally address issues in solving some real-world problems. Simoudis, Livezey, and Kerber discuss how rule induction, deductive databases, and data visualization can be used cooperatively to create high quality, rule-based models by mining data stored in relational databases. Shen, Mitbander, Ong, and Zaniolo present a framework that uses metaqueries to integrate inductive learning methods with deductive database technologies

in the context of knowledge discovery from databases, and illustrate this with three case studies. Han and Fu show how to use attribute-oriented induction (which generalizes the relevant subset of data attribute-by-attribute) to find patterns of different types, including characteristic and classification rules.

Part VI includes two chapters on approaches for next generation database systems. Hsu and Knoblock show how learning can be used to generate rules for semantic query optimization. Holsheimer, Kersten, and Siebes present a parallel DBMS engine, called Data Surveyor, which has special features for optimizing various types of data mining.

Part VII presents several real and successful applications. Fayyad, Djorgovski, and Weir present SKICAT, a system which automatically detects and classifies sky objects from image data resulting from a major astronomical sky survey. The data mining techniques used in SKICAT enabled solving a difficult, scientifically significant problem, and resulted in a system that can outperform astronomers in its accuracy in classifying faint sky objects. It is now used to automatically catalog an estimated two billion objects. Matheus, Piatetsky-Shapiro, and McNeill present a framework for determining the interestingness of deviations from normative and previous values and show its implementation in the KEFIR system for the analysis of Healthcare data. Smyth, Burl, Fayyad, and Perona address the inconsistencies of human classifications in automating the catalog of a million small volcanoes in the 30,000 Venus images returned by the Magellan spacecraft. Apte and Hong show how to use minimal rule generation and contextual feature analysis techniques for extracting useful information from securities data to predict equity returns.

We conclude the book with an epilogue by Uthurusamy. The appendix provides a list of terms used in the KDD literature and their equivalents in other related fields. The goal of Appendix A by Kloesgen and Zytkow, is to provide the seeds for a common terminology in the rapidly growing KDD field. Appendix B by Piatetsky-Shapiro provides pointers to the many resources for Knowledge Discovery and Data Mining, including software, datasets, and publications that are available via the Internet and the World-Wide Web.

Acknowledgments

The authors would like to thank Evangelos Simoudis, R. Uthurusamy, and Chris Matheus for their comments and insights on an earlier draft of this chapter. GPS thanks Shri Goyal for his encouragement and support. Part of the writing of this chapter was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and was supported in part by ARPA and ONR under grant number N00014-92-J-1860.

References

Anand, T.; and Kahn, G. 1992. SPOTLIGHT: A Data Explanation System. In *Proceedings of the Eighth IEEE Conference on Applied AI,* 2–8. Washington, D.C.: IEEE Press.

Babcock, C. 1994. Parallel Processing Mines Retail Data. *ComputerWorld,* 6, September 26, 1994.

Barr, D.; and Mani, G. 1994. Using Neural Nets to Manage Investments. *AI Expert,* 16–21, February.

Basseville, M.; and Nikiforov, I. V. 1993. *Detection of Abrupt Changes: Theory and Application.* Englewood Cliffs, NJ: Prentice Hall.

Berry, J. 1994. Database Marketing. *Business Week,* 56–62, September 5.

Blanchard, D. 1994. News Watch. *AI Expert,* 7, December.

Breiman, L.; Friedman, J. H.; Olshen, R. A.; and Stone, C. J. 1984.*Classification and Regression Trees.* Belmont, Calif.: Wadsworth.

Cercone, N.; and Tsuchiya, M. 1993. Special Issue on Learning and Discovery in Databases, *IEEE Transactions on Knowledge and Data Engineering,* 5(6), Dec.

Cheeseman, P. 1990.On Finding the Most Probable Model. In *Computational Models of Scientific Discovery and Theory Formation,* ed. J. Shrager and P. Langley. San Francisco: Morgan Kaufmann, 73–95.

Cheng, B.; and Titterington, D. M. 1994. Neural Networks—a Review from a Statistical Perspective. *Statistical Science,* 9(1): 2–30.

Codd, E.F. 1993. *Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate.* E.F. Codd and Associates.

Conklin, D.; Fortier, S.; and Glasgow, J. 1993. Knowledge Discovery in Molecular Databases. *IEEE Transactions on Knowledge and Data Engineering,* 5(6): 985–987, Dec.

Dasarathy, B. V. 1991. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* Los Alamitos, Calif.: IEEE Computer Society Press.

Fasman, K. H.; Cuticchia, A. J.; and Kingsbury, D. T. 1994. The GDB (TM) Human Genome Database Anno 1994. *Nucl. Acid. R.,* 22(17): 3462–3469.

Fayyad, U. M.; and Uthurusamy, R. 1994. Proceedings of KDD-94: the AAAI-94 Workshop on Knowledge Discovery in Databases, AAAI Technical Report WS-94-03. Menlo Park, Calif.: The AAAI Press.

Fayyad, U. M.; and Uthurusamy, R. 1995. Proceedings, First International Conference on Knowledge Discovery and Data Mining. Menlo Park, Calif.: The AAAI Press.

Frawley, W. J.; Piatetsky-Shapiro, G.; and Matheus, C. J. 1991. Knowledge Discovery in Databases: An Overview. In *Knowledge Discovery in Databases,* ed. G. Piatetsky-Shapiro and B. Frawley. Cambridge, Mass: AAAI/MIT Press, 1–27.

Friedman, J. H. 1989. Multivariate Adaptive Regression Splines. *Annals of Statistics,* 19: 1–141.

Geman, S.; Bienenstock, E.; and Doursat, R. 1992. Neural Networks and the Bias/Variance Dilemma. *Neural Computation,* 4: 1–58.

Glymour, C.; Scheines, R.; Spirtes, P.; and Kelly, K. 1987. *Discovering Causal Structure.* New York: Academic Press.

Hand, D. J. 1981. *Discrimination and Classification.* Chichester, U.K.: John Wiley and Sons.

Harrison, D. 1993. Backing Up. *Network Computing,* October 15, 98–104.

Holder, L.; Cook, D.; and Djoko, S. 1994. Substructure Discovery in the SUBDUE system. In Proceedings of KDD-94: the AAAI-94 Workshop on Knowledge Discovery in Databases, 169–180. AAAI Technical Report WS-94-03. Menlo Park, Calif.: The AAAI Press.

Inmon, W. H.; and Osterfelt, S. 1991. *Understanding Data Pattern Processing: The Key to Competitive Advantage.* Wellesley, Mass.: QED Technical Publishing Group.

Jain, A. K.; and Dubes; R. C. 1988. *Algorithms for Clustering Data.* Englewood Cliffs, NJ: Prentice-Hall.

Kloesgen, W. 1993. Some Implementation Aspects of a Discovery System. In Proceedings of KDD-94: the AAAI-94 Workshop on Knowledge Discovery in Databases, 212–226. AAAI Technical Report WS-94-03. Menlo Park, Calif.: The AAAI Press.

Kolodner, J. 1993. *Case-Based Reasoning.* San Francisco: Morgan Kaufmann.

Lehmann, E. L. 1990. Model Specification: the Views of Fisher and Neyman, and Later Developments. *Statistical Science,* 5(2), 160–168.

Major, J. 1993. Selecting Among Rules Induced from a Hurricance Database. In Proceedings of KDD-94: the AAAI-94 Workshop on Knowledge Discovery in Databases, 28–44. AAAI Technical Report WS-94-03. Menlo Park, Calif.: The AAAI Press.

Major, J.; and Riedinger, D. 1992. EFD: A Hybrid Knowledge/Statistical-Based System for the Detection of Fraud. *International Journal of Intelligent Systems,* 7(7): 687–703.

Matheus, C.; Chan, P.; and Piatetsky-Shapiro, G. 1993. Systems for Knowledge Discovery. *IEEE Trans. on Knowledge and Data Engineering,* 5(6): 903–913.

McLachlan, G. 1992. *Discriminant Analysis and Statistical Pattern Recognition.* New York: Wiley.

O'Leary, D. 1995. Some Privacy Issues in Knowledge Discovery: OECD Personal Privacy Guidelines. *IEEE Expert.* Forthcoming.

Parsaye, K.; and Chignell, M. 1993. *Intelligent Database Tools & Applications.* New York: John Wiley.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems.* San Francisco: Morgan Kaufmann.

Piatetsky-Shapiro, G.; and Frawley, W. 1991. *Knowledge Discovery in Databases.* Menlo Park, Calif.: AAAI Press.

Piatetsky-Shapiro, G. 1991. Report on the AAAI-91 Workshop on Knowledge Discovery in Databases. *IEEE Expert,* 6(5): 74–76.

Piatetsky-Shapiro, G. 1992. Editor, Special issue on Knowledge Discovery in Databases and Knowledgebases. *International Journal of Intelligent Systems* 7:7, September.

Piatetsky-Shapiro, G.; Matheus, C.; Smyth, P.; and Uthurusamy, R. 1994. KDD-93: Progress and Challenges in Knowledge Discovery in Databases. *AI Magazine,* 15(3): 77–87.

Piatetsky-Shapiro, G. 1995b. Knowledge Discovery in Personal Data Versus Privacy: A Mini-Symposium. *IEEE Expert.* Forthcoming.

Quinlan, J. 1992. *C4.5: Programs for Machine Learning.* San Francisco: Morgan Kaufmann.

Ripley, B. D. 1994. Neural Networks and Related Methods for Classification. *Journal of the Royal. Stat. Society,* 56(3): 409-437.

Rosenberg, M. 1992. Protecting Privacy, Inside Risks Column. *Communications of ACM,* 35(4): 164.

Schmitz, J.; Armstrong, G.; and Little, J. D. C. 1990. CoverStory—Automated News Finding in Marketing. In *DSS Transactions*, ed. L. Volino, 46–54. Providence, R.I.: Institute of Management Sciences.

Shrager, J.; and Langley, P., eds. 1990. *Computational Models of Scientific Discovery and Theory Formation.* San Francisco, Calif.: Morgan Kaufmann.

Silverman, B. 1986. *Density Estimation for Statistics and Data Analysis.* New York: Chapman and Hall.

Spirtes, P.; Glymour, C.; and Scheines, R. 1993. *Causation, Prediction, and Search,* New York: Springer-Verlag.

Stolorz, P. et al. 1994. Data Analysis and Knowledge Discovery in Geophysical Databases. *Concurrent Supercomputing Consortium Annual Report,* California Institute of Technology, 12–14.

Titterington, D. M.; Smith, A. F. M.; and Makov, U. E. 1985. *Statistical Analysis of Finite Mixture Distributions.* Chichester, U.K.: John Wiley and Sons.

Way, J.; and Smith, E. A. 1991. The Evolution of Synthetic Aperture Radar Systems and their Progression to the EOS SAR. *IEEE Transactions on Geoscience and Remote Sensing,* 29(6): 962–985.

Whittaker, J. 1990. *Graphical Models in Applied Multivariate Statistics.* New York: Wiley.

Weigend, A.; and Gershenfeld, N., eds. 1993. *Predicting the Future and Understanding the Past.* Redwood City, Calif: Addison-Wesley.

Weiss, S. I.; and Kulikowski, C. 1991. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems.* San Francisco, Calif.: Morgan Kaufmann.

Ziarko, W. 1994. *Rough Sets, Fuzzy Sets and Knowledge Discovery.* Berlin: Springer Verlag.