From Data Mining to Knowledge Discovery in Databases

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth

■ Data mining and knowledge discovery in databases have been attracting a significant amount of research, industry, and media attention of late. What is all the excitement about? This article provides an overview of this emerging field, clarifying how data mining and knowledge discovery in databases are related both to each other and to related fields, such as machine learning, statistics, and databases. The article mentions particular real-world applications, specific data-mining techniques, challenges involved in real-world applications of knowledge discovery, and current and future research directions in the field.

Across a wide variety of fields, data are being collected and accumulated at a dramatic pace. There is an urgent need for a new generation of computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data. These theories and tools are the subject of the emerging field of knowledge discovery in databases (KDD).

At an abstract level, the KDD field is concerned with the development of methods and techniques for making sense of data. The basic problem addressed by the KDD process is one of mapping low-level data (which are typically too voluminous to understand and digest easily) into other forms that might be more compact (for example, a short report), more abstract (for example, a descriptive approximation or model of the process that generated the data), or more useful (for example, a predictive model for estimating the value of future cases). At the core of the process is the application of specific data-mining methods for pattern discovery and extraction.[1]

This article begins by discussing the historical context of KDD and data mining and their intersection with other related fields. A brief summary of recent KDD real-world applications is provided. Definitions of KDD and data mining are provided, and the general multistep KDD process is outlined. This multistep process has the application of data-mining algorithms as one particular step in the process. The data-mining step is discussed in more detail in the context of specific data-mining algorithms and their application. Real-world practical application issues are also outlined. Finally, the article enumerates challenges for future research and development and in particular discusses potential opportunities for AI technology in KDD systems.

Why Do We Need KDD?

The traditional method of turning data into knowledge relies on manual analysis and interpretation. For example, in the health-care industry, it is common for specialists to periodically analyze current trends and changes in health-care data, say, on a quarterly basis. The specialists then provide a report detailing the analysis to the sponsoring health-care organization; this report becomes the basis for future decision making and planning for health-care management. In a totally different type of application, planetary geologists sift through remotely sensed images of planets and asteroids, carefully locating and cataloging such geologic objects of interest as impact craters. Be it science, marketing, finance, health care, retail, or any other field, the classical approach to data analysis relies fundamentally on one or more analysts becoming

intimately familiar with the data and serving as an interface between the data and the users and products.

For these (and many other) applications, this form of manual probing of a data set is slow, expensive, and highly subjective. In fact, as data volumes grow dramatically, this type of manual data analysis is becoming completely impractical in many domains. Databases are increasing in size in two ways: (1) the number N of records or objects in the database and (2) the number d of fields or attributes to an object. Databases containing on the order of $N = 10^9$ objects are becoming increasingly common, for example, in the astronomical sciences. Similarly, the number of fields d can easily be on the order of 10^2 or even 10^3, for example, in medical diagnostic applications. Who could be expected to digest millions of records, each having tens or hundreds of fields? We believe that this job is certainly not one for humans; hence, analysis work needs to be automated, at least partially.

There is an urgent need for a new generation of computational theories and tools to assist humans in extracting useful information (knowledge) from the rapidly growing volumes of digital data.

The need to scale up human analysis capabilities to handling the large number of bytes that we can collect is both economic and scientific. Businesses use data to gain competitive advantage, increase efficiency, and provide more valuable services to customers. Data we capture about our environment are the basic evidence we use to build theories and models of the universe we live in. Because computers have enabled humans to gather more data than we can digest, it is only natural to turn to computational techniques to help us unearth meaningful patterns and structures from the massive volumes of data. Hence, KDD is an attempt to address a problem that the digital information era made a fact of life for all of us: data overload.

Data Mining and Knowledge Discovery in the Real World

A large degree of the current interest in KDD is the result of the media interest surrounding successful KDD applications, for example, the focus articles within the last two years in *Business Week*, *Newsweek*, *Byte*, *PC Week*, and other large-circulation periodicals. Unfortunately, it is not always easy to separate fact from media hype. Nonetheless, several well-documented examples of successful systems can rightly be referred to as KDD applications and have been deployed in operational use on large-scale real-world problems in science and in business.

In science, one of the primary application areas is astronomy. Here, a notable success was achieved by SKICAT, a system used by astronomers to perform image analysis, classification, and cataloging of sky objects from sky-survey images (Fayyad, Djorgovski, and Weir 1996). In its first application, the system was used to process the 3 terabytes (10^{12} bytes) of image data resulting from the Second Palomar Observatory Sky Survey, where it is estimated that on the order of 10^9 sky objects are detectable. SKICAT can outperform humans and traditional computational techniques in classifying faint sky objects. See Fayyad, Haussler, and Stolorz (1996) for a survey of scientific applications.

In business, main KDD application areas includes marketing, finance (especially investment), fraud detection, manufacturing, telecommunications, and Internet agents.

Marketing: In marketing, the primary application is database marketing systems, which analyze customer databases to identify different customer groups and forecast their behavior. *Business Week* (Berry 1994) estimated that over half of all retailers are using or planning to use database marketing, and those who do use it have good results; for example, American Express reports a 10- to 15-percent increase in credit-card use. Another notable marketing application is market-basket analysis (Agrawal et al. 1996) systems, which find patterns such as, "If customer bought X, he/she is also likely to buy Y and Z." Such patterns are valuable to retailers.

Investment: Numerous companies use data mining for investment, but most do not describe their systems. One exception is LBS Capital Management. Its system uses expert systems, neural nets, and genetic algorithms to manage portfolios totaling $600 million; since its start in 1993, the system has outperformed the broad stock market (Hall, Mani, and Barr 1996).

Fraud detection: HNC Falcon and Nestor PRISM systems are used for monitoring credit-card fraud, watching over millions of accounts. The FAIS system (Senator et al. 1995), from the U.S. Treasury Financial Crimes Enforcement Network, is used to identify financial transactions that might indicate money-laundering activity.

Manufacturing: The CASSIOPEE troubleshooting system, developed as part of a joint venture between General Electric and SNECMA, was applied by three major European airlines to diagnose and predict problems for the Boeing 737. To derive families of faults, clustering methods are used. CASSIOPEE received the European first prize for innova-

tive applications (Manago and Auriol 1996).

Telecommunications: The telecommunications alarm-sequence analyzer (TASA) was built in cooperation with a manufacturer of telecommunications equipment and three telephone networks (Mannila, Toivonen, and Verkamo 1995). The system uses a novel framework for locating frequently occurring alarm episodes from the alarm stream and presenting them as rules. Large sets of discovered rules can be explored with flexible information-retrieval tools supporting interactivity and iteration. In this way, TASA offers pruning, grouping, and ordering tools to refine the results of a basic brute-force search for rules.

Data cleaning: The MERGE-PURGE system was applied to the identification of duplicate welfare claims (Hernandez and Stolfo 1995). It was used successfully on data from the Welfare Department of the State of Washington.

In other areas, a well-publicized system is IBM's ADVANCED SCOUT, a specialized data-mining system that helps National Basketball Association (NBA) coaches organize and interpret data from NBA games (U.S. News 1995). ADVANCED SCOUT was used by several of the NBA teams in 1996, including the Seattle Supersonics, which reached the NBA finals.

Finally, a novel and increasingly important type of discovery is one based on the use of intelligent agents to navigate through an information-rich environment. Although the idea of active triggers has long been analyzed in the database field, really successful applications of this idea appeared only with the advent of the Internet. These systems ask the user to specify a profile of interest and search for related information among a wide variety of public-domain and proprietary sources. For example, FIREFLY is a personal music-recommendation agent: It asks a user his/her opinion of several music pieces and then suggests other music that the user might like (<http://www.ffly.com/>). CRAYON (http://crayon.net/>) allows users to create their own free newspaper (supported by ads); NEWSHOUND (<http://www.sjmercury.com/hound/>) from the *San Jose Mercury News* and FARCAST (<http://www.farcast.com/> automatically search information from a wide variety of sources, including newspapers and wire services, and e-mail relevant documents directly to the user.

These are just a few of the numerous such systems that use KDD techniques to automatically produce useful information from large masses of raw data. See Piatetsky-Shapiro et al. (1996) for an overview of issues in developing industrial KDD applications.

Data Mining and KDD

Historically, the notion of finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction, information discovery, information harvesting, data archaeology, and data pattern processing. The term *data mining* has mostly been used by statisticians, data analysts, and the management information systems (MIS) communities. It has also gained popularity in the database field. The phrase *knowledge discovery in databases* was coined at the first KDD workshop in 1989 (Piatetsky-Shapiro 1991) to emphasize that knowledge is the end product of a data-driven discovery. It has been popularized in the AI and machine-learning fields.

In our view, KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. *Data mining* is the application of specific algorithms for extracting patterns from data. The distinction between the KDD process and the data-mining step (within the process) is a central point of this article. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, are essential to ensure that useful knowledge is derived from the data. Blind application of data-mining methods (rightly criticized as data dredging in the statistical literature) can be a dangerous activity, easily leading to the discovery of meaningless and invalid patterns.

The basic problem addressed by the KDD process is one of mapping low-level data into other forms that might be more compact, more abstract, or more useful.

The Interdisciplinary Nature of KDD

KDD has evolved, and continues to evolve, from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, AI, knowledge acquisition for expert systems, data visualization, and high-performance computing. The unifying goal is extracting high-level knowledge from low-level data in the context of large data sets.

The data-mining component of KDD currently relies heavily on known techniques from machine learning, pattern recognition, and statistics to find patterns from data in the data-mining step of the KDD process. A natural question is, How is KDD different from pattern recognition or machine learning (and related fields)? The answer is that these fields provide some of the data-mining methods that are used in the data-mining step of the KDD process. KDD focuses on the overall process of knowledge discovery from data, including how the data are stored and accessed, how algorithms can be scaled to massive data sets

and still run efficiently, how results can be interpreted and visualized, and how the overall man-machine interaction can usefully be modeled and supported. The KDD process can be viewed as a multidisciplinary activity that encompasses techniques beyond the scope of any one particular discipline such as machine learning. In this context, there are clear opportunities for other fields of AI (besides machine learning) to contribute to KDD. KDD places a special emphasis on finding understandable patterns that can be interpreted as useful or interesting knowledge. Thus, for example, neural networks, although a powerful modeling tool, are relatively difficult to understand compared to decision trees. KDD also emphasizes scaling and robustness properties of modeling algorithms for large noisy data sets.

Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data.

Related AI research fields include machine discovery, which targets the discovery of empirical laws from observation and experimentation (Shrager and Langley 1990) (see Kloesgen and Zytkow [1996] for a glossary of terms common to KDD and machine discovery), and causal modeling for the inference of causal models from data (Spirtes, Glymour, and Scheines 1993). Statistics in particular has much in common with KDD (see Elder and Pregibon [1996] and Glymour et al. [1996] for a more detailed discussion of this synergy). Knowledge discovery from data is fundamentally a statistical endeavor. Statistics provides a language and framework for quantifying the uncertainty that results when one tries to infer general patterns from a particular sample of an overall population. As mentioned earlier, the term *data mining* has had negative connotations in statistics since the 1960s when computer-based data analysis techniques were first introduced. The concern arose because if one searches long enough in any data set (even randomly generated data), one can find patterns that appear to be statistically significant but, in fact, are not. Clearly, this issue is of fundamental importance to KDD. Substantial progress has been made in recent years in understanding such issues in statistics. Much of this work is of direct relevance to KDD. Thus, data mining is a legitimate activity as long as one understands how to do it correctly; data mining carried out poorly (without regard to the statistical aspects of the problem) is to be avoided. KDD can also be viewed as encompassing a broader view of modeling than statistics. KDD aims to provide tools to automate (to the degree possible) the entire process of data analysis and the statistician's "art" of hypothesis selection.

A driving force behind KDD is the database field (the second D in KDD). Indeed, the problem of effective data manipulation when data cannot fit in the main memory is of fundamental importance to KDD. Database techniques for gaining efficient data access, grouping and ordering operations when accessing data, and optimizing queries constitute the basics for scaling algorithms to larger data sets. Most data-mining algorithms from statistics, pattern recognition, and machine learning assume data are in the main memory and pay no attention to how the algorithm breaks down if only limited views of the data are possible.

A related field evolving from databases is *data warehousing,* which refers to the popular business trend of collecting and cleaning transactional data to make them available for online analysis and decision support. Data warehousing helps set the stage for KDD in two important ways: (1) data cleaning and (2) data access.

Data cleaning: As organizations are forced to think about a unified logical view of the wide variety of data and databases they possess, they have to address the issues of mapping data to a single naming convention, uniformly representing and handling missing data, and handling noise and errors when possible.

Data access: Uniform and well-defined methods must be created for accessing the data and providing access paths to data that were historically difficult to get to (for example, stored offline).

Once organizations and individuals have solved the problem of how to store and access their data, the natural next step is the question, What else do we do with all the data? This is where opportunities for KDD naturally arise.

A popular approach for analysis of data warehouses is called *online analytical processing* (OLAP), named for a set of principles proposed by Codd (1993). OLAP tools focus on providing multidimensional data analysis, which is superior to SQL in computing summaries and breakdowns along many dimensions. OLAP tools are targeted toward simplifying and supporting interactive data analysis, but the goal of KDD tools is to automate as much of the process as possible. Thus, KDD is a step beyond what is currently supported by most standard database systems.

Basic Definitions

KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimate-

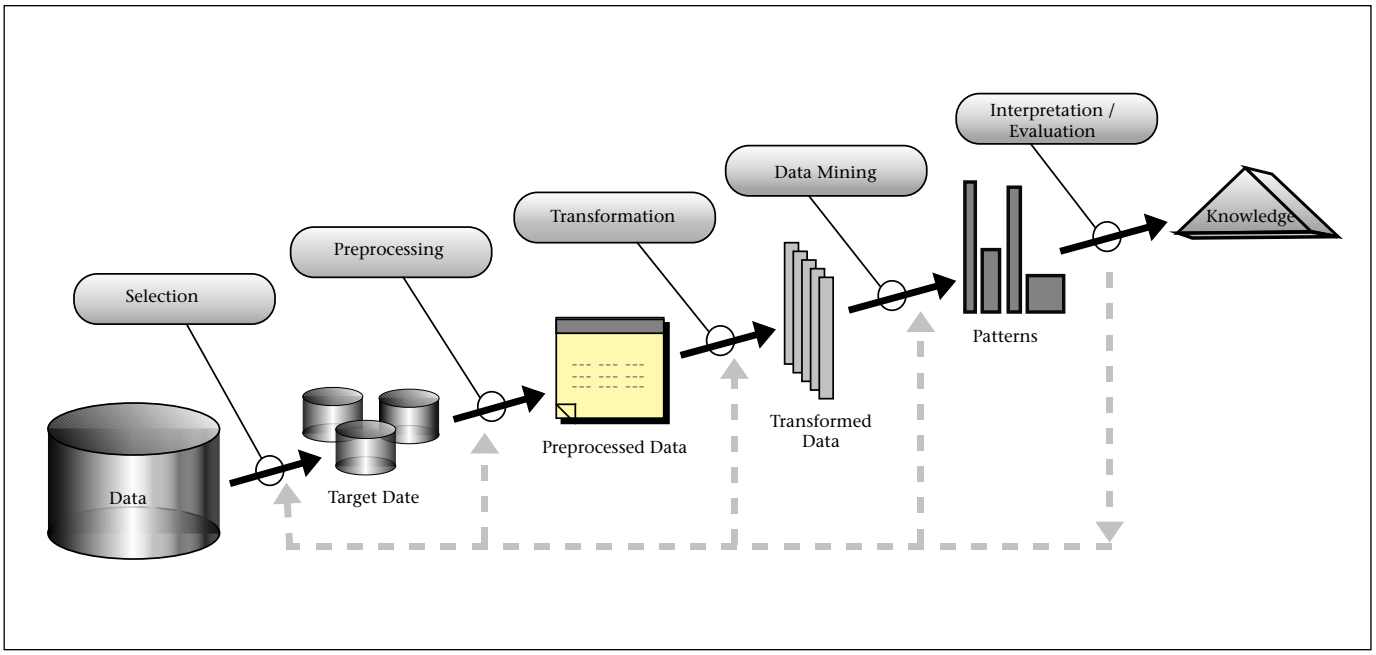

Figure 1. An Overview of the Steps That Compose the KDD Process.

ly understandable patterns in data (Fayyad, Piatetsky-Shapiro, and Smyth 1996).

Here, *data* are a set of facts (for example, cases in a database), and *pattern* is an expression in some language describing a subset of the data or a model applicable to the subset. Hence, in our usage here, extracting a pattern also designates fitting a model to data; finding structure from data; or, in general, making any high-level description of a set of data. The term *process* implies that KDD comprises many steps, which involve data preparation, search for patterns, knowledge evaluation, and refinement, all repeated in multiple iterations. By *nontrivial*, we mean that some search or inference is involved; that is, it is not a straightforward computation of predefined quantities like computing the average value of a set of numbers.

The discovered patterns should be valid on new data with some degree of certainty. We also want patterns to be novel (at least to the system and preferably to the user) and potentially useful, that is, lead to some benefit to the user or task. Finally, the patterns should be understandable, if not immediately then after some postprocessing.

The previous discussion implies that we can define quantitative measures for evaluating extracted patterns. In many cases, it is possible to define measures of certainty (for example, estimated prediction accuracy on new data) or utility (for example, gain, perhaps in dollars saved because of better predictions or speedup in response time of a system). Notions such as novelty and understandability are much more subjective. In certain contexts, understandability can be estimated by simplicity (for example, the number of bits to describe a pattern). An important notion, called *interestingness* (for example, see Silberschatz and Tuzhilin [1995] and Piatetsky-Shapiro and Matheus [1994]), is usually taken as an overall measure of pattern value, combining validity, novelty, usefulness, and simplicity. Interestingness functions can be defined explicitly or can be manifested implicitly through an ordering placed by the KDD system on the discovered patterns or models.

Given these notions, we can consider a *pattern* to be knowledge if it exceeds some interestingness threshold, which is by no means an attempt to define knowledge in the philosophical or even the popular view. As a matter of fact, knowledge in this definition is purely user oriented and domain specific and is determined by whatever functions and thresholds the user chooses.

Data mining is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data. Note that the space of

patterns is often infinite, and the enumeration of patterns involves some form of search in this space. Practical computational constraints place severe limits on the subspace that can be explored by a data-mining algorithm.

The KDD process involves using the database along with any required selection, preprocessing, subsampling, and transformations of it; applying data-mining methods (algorithms) to enumerate patterns from it; and evaluating the products of data mining to identify the subset of the enumerated patterns deemed knowledge. The data-mining component of the KDD process is concerned with the algorithmic means by which patterns are extracted and enumerated from data. The overall KDD process (figure 1) includes the evaluation and possible interpretation of the mined patterns to determine which patterns can be considered new knowledge. The KDD process also includes all the additional steps described in the next section.

The notion of an overall user-driven process is not unique to KDD: analogous proposals have been put forward both in statistics (Hand 1994) and in machine learning (Brodley and Smyth 1996).

The KDD Process

The KDD process is interactive and iterative, involving numerous steps with many decisions made by the user. Brachman and Anand (1996) give a practical view of the KDD process, emphasizing the interactive nature of the process. Here, we broadly outline some of its basic steps:

First is developing an understanding of the application domain and the relevant prior knowledge and identifying the goal of the KDD process from the customer's viewpoint.

Second is 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.

Third is data cleaning and preprocessing. Basic operations include removing noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, and accounting for time-sequence information and known changes.

Fourth is data reduction and projection: finding useful features to represent the data depending on the goal of the task. With dimensionality reduction or transformation methods, the effective number of variables under consideration can be reduced, or invariant representations for the data can be found.

Fifth is matching the goals of the KDD process (step 1) to a particular data-mining method. For example, summarization, classification, regression, clustering, and so on, are described later as well as in Fayyad, Piatetsky-Shapiro, and Smyth (1996).

Sixth is exploratory analysis and model and hypothesis selection: choosing the data-mining algorithm(s) and selecting method(s) to be used for searching for data patterns. This process includes deciding which models and parameters might be appropriate (for example, models of categorical data are different than models of vectors over the reals) and matching a particular data-mining method with the overall criteria of the KDD process (for example, the end user might be more interested in understanding the model than its predictive capabilities).

Seventh is data mining: searching for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, and clustering. The user can significantly aid the data-mining method by correctly performing the preceding steps.

Eighth is interpreting mined patterns, possibly returning to any of steps 1 through 7 for further iteration. This step can also involve visualization of the extracted patterns and models or visualization of the data given the extracted models.

Ninth is acting on the discovered knowledge: using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parties. This process also includes checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

The KDD process can involve significant iteration and can contain loops between any two steps. The basic flow of steps (although not the potential multitude of iterations and loops) is illustrated in figure 1. Most previous work on KDD has focused on step 7, the data mining. However, the other steps are as important (and probably more so) for the successful application of KDD in practice. 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.

The Data-Mining Step of the KDD Process

The data-mining component of the KDD process often involves repeated iterative application of particular data-mining methods. This section presents an overview of the primary goals of data mining, a description of the methods used to address these goals, and a brief description of the data-mining algorithms that incorporate these methods.

The knowledge discovery goals are defined by the intended use of the system. We can distinguish two types of goals: (1) verification and (2) discovery. With *verification,* the system is limited to verifying the user's hypothesis. With *discovery,* the system autonomously finds new patterns. We further subdivide the discovery goal 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 article, we are primarily concerned with discovery-oriented data mining.

Data mining involves fitting models to, or determining patterns from, observed data. The fitted models play the role of inferred knowledge: Whether the models reflect useful or interesting knowledge is part of the overall, interactive KDD process where subjective human judgment is typically required. Two primary mathematical formalisms are used in model fitting: (1) statistical and (2) logical. The *statistical approach* allows for nondeterministic effects in the model, whereas a *logical model* is purely deterministic. We focus primarily on the statistical approach to data mining, which tends to be the most widely used basis for practical data-mining applications given the typical presence of uncertainty in real-world data-generating processes.

Most data-mining methods are based on tried and tested techniques from machine learning, pattern recognition, and statistics: classification, clustering, regression, and so on. The array of different algorithms under each of these headings can often be bewildering to both the novice and the experienced data analyst. It should be emphasized that of the many data-mining methods advertised in the literature, there are really only a few fundamental techniques. The actual underlying model representation being used by a particular method typically comes from a composition of a small number of well-known options: polynomials, splines, kernel and basis functions, threshold-Boolean functions, and so on. Thus, algorithms tend to differ primarily in the goodness-of-fit criterion used to evaluate model fit or in the search method used to find a good fit.

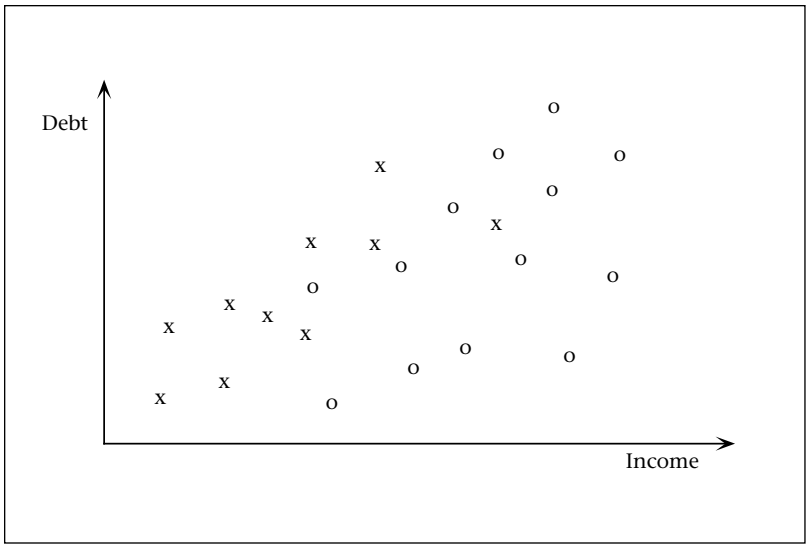

Figure 2. A Simple Data Set with Two Classes Used for Illustrative Purposes.

In our brief overview of data-mining methods, we try in particular to convey the notion that most (if not all) methods can be viewed as extensions or hybrids of a few basic techniques and principles. We first discuss the primary methods of data mining and then show that the data- mining methods can be viewed as consisting of three primary algorithmic components: (1) model representation, (2) model evaluation, and (3) search. In the discussion of KDD and data-mining methods, we use a simple example to make some of the notions more concrete. Figure 2 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, and so on). The data have been classified into two classes: (1) the x's represent persons who have defaulted on their loans and (2) 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 that can 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 (as many as several hundreds) and many more data points (many thousands or even millions).

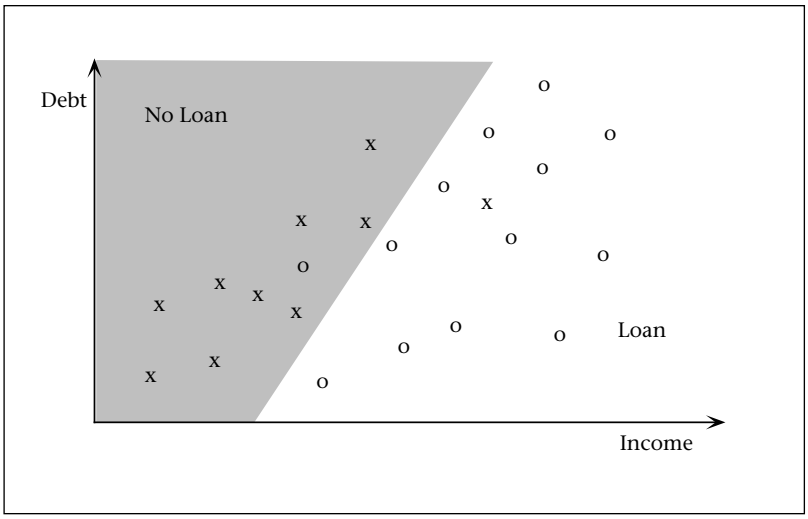

Figure 3. A Simple Linear Classification Boundary for the Loan Data Set.
The shaped region denotes class *no loan*.

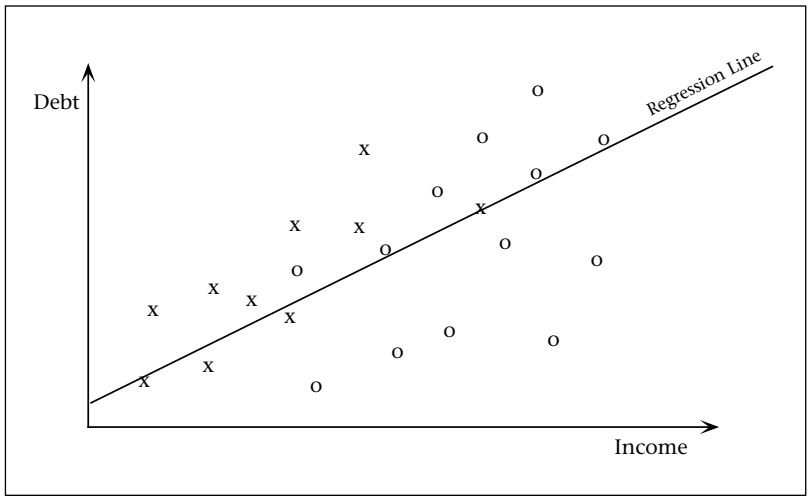

Figure 4. A Simple Linear Regression for the Loan Data Set.

The purpose here is to illustrate basic ideas on a small problem in two-dimensional space.

Data-Mining Methods

The two high-level primary goals of data mining in practice tend to be prediction and description. As stated earlier, prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest, and description focuses on finding human-interpretable patterns describing the data. Although the boundaries between prediction and description are not sharp (some of the predictive models can be descriptive, to the degree that they are understandable, and vice versa), the distinction is useful for understanding the overall discovery goal. The relative importance of prediction and description for particular data-mining applications can vary considerably. The goals of prediction and description can be achieved using a variety of particular data-mining methods.

Classification is learning a function that maps (classifies) a data item into one of several predefined classes (Weiss and Kulikowski 1991; Hand 1981). Examples of classification methods used as part of knowledge discovery applications include the classifying of trends in financial markets (Apte and Hong 1996) and the automated identification of objects of interest in large image databases (Fayyad, Djorgovski, and Weir 1996). Figure 3 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 want to use the classification regions to automatically decide whether future loan applicants will be given a loan or not.

Regression is learning a function that maps a data item to a real-valued prediction variable. Regression applications are many, for example, predicting the amount of biomass present in a forest given remotely sensed microwave measurements, estimating the probability that a patient will survive given the results of a set of diagnostic tests, predicting consumer demand for a new product as a function of advertising expenditure, and predicting time series where the input variables can be time-lagged versions of the prediction variable. Figure 4 shows the result of simple linear regression where total debt is fitted as a linear function of income: The fit is poor because only a weak correlation exists 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 (Jain and Dubes 1988; Titterington, Smith, and Makov 1985). The categories can 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 subpopulations for consumers in marketing databases and identifying subcategories of spectra from infrared sky measurements (Cheeseman and Stutz 1996). Figure 5 shows a possible clustering of the loan data set into three 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 a + 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 multivariate probability density function of all the variables or fields in the database (Silverman 1986).

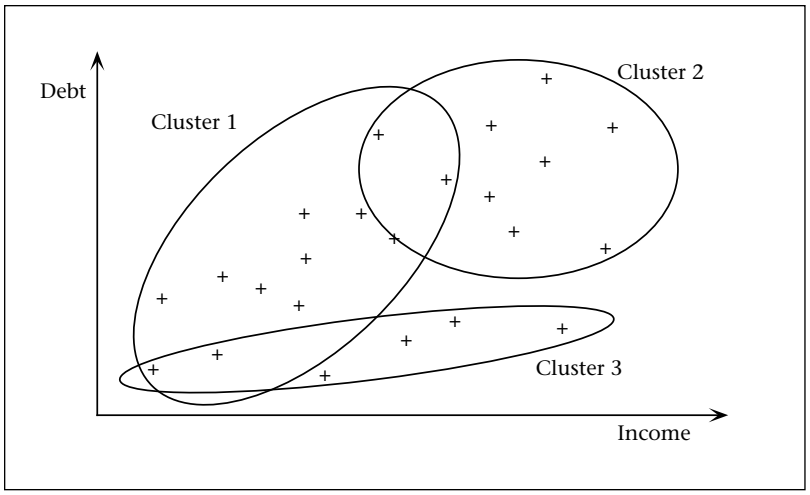

Figure 5. A Simple Clustering of the Loan Data Set into Three Clusters.
Note that original labels are replaced by a +.

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. 1996), multivariate visualization techniques, and the discovery of functional relationships between variables (Zembowicz and Zytkow 1996). Summarization techniques are often applied to interactive exploratory data analysis and automated report generation.

Dependency modeling consists of finding a model that describes significant dependencies between variables. Dependency models exist at two levels: (1) the *structural level* of the model specifies (often in graphic form) which variables are locally dependent on each other and (2) the *quantitative level* of the model specifies the strengths of the dependencies using some numeric scale. For example, probabilistic dependency networks use conditional independence to specify the structural aspect of the model and probabilities or correlations to specify the strengths of the dependencies (Glymour et al. 1987; Heckerman 1996). 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 and Clifford 1996; Guyon, Matic, and Vapnik 1996; Kloesgen 1996; Matheus, Piatetsky-Shapiro, and McNeill 1996; Basseville and Nikiforov 1993).

The Components of Data-Mining Algorithms

The next step is to construct specific algorithms to implement the general methods we outlined. One can identify three primary components in any data-mining algorithm: (1) model representation, (2) model evaluation, and (3) 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 used to describe discoverable patterns. If the representation is too limited, then no amount of training time or examples can produce an accurate model for the data. It is important that a data analyst fully comprehend the representational assumptions that might be inherent in 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 increased representational power for models increases the danger of overfitting the training data, resulting in reduced prediction accuracy on unseen data.

Model-evaluation criteria are quantitative

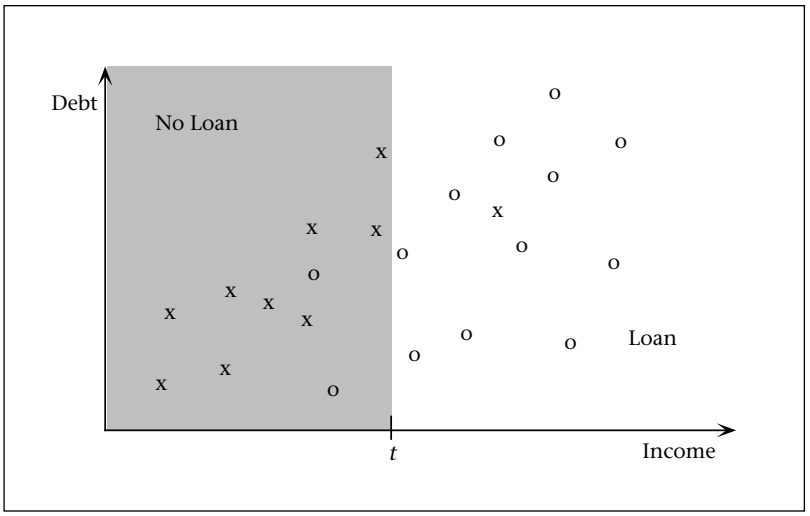

Figure 6. Using a Single Threshold on the Income Variable to Try to Classify the Loan Data Set.

statements (or *fit functions*) of how well a particular pattern (a model and its parameters) meets the goals of the KDD process. For example, predictive models are often judged by the empirical prediction accuracy on some test set. Descriptive models can be evaluated along the dimensions of predictive accuracy, novelty, utility, and understandability of the fitted model.

Search method consists of two components: (1) parameter search and (2) model search. Once the model representation (or family of representations) and the model-evaluation criteria are fixed, then the data-mining problem has been reduced to purely an optimization task: Find the parameters and models from the selected family that optimize the evaluation criteria. In parameter search, the algorithm must search for the parameters that optimize the model-evaluation criteria given observed data and a fixed model representation. Model search occurs as a loop over the parameter-search method: The model representation is changed so that a family of models is considered.

Some Data-Mining Methods

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

Decision Trees and Rules

Decision trees and rules that use univariate splits have a simple representational form, making the inferred model relatively easy for the user to comprehend. 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 6 illustrates the effect of a threshold split applied to the income variable for a loan data set: It is clear that using such simple threshold splits (parallel to the feature axes) severely limits the type of classification boundaries that 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 can be much more difficult to comprehend. A large number of decision tree and rule-induction algorithms are described in the machine-learning and applied statistics literature (Quinlan 1992; Breiman et al. 1984).

To a large extent, they depend 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 used to explore the superexponential space of possible models. Trees and rules are primarily used for predictive modeling, both for classification (Apte and Hong 1996; Fayyad, Djorgovski, and Weir 1996) and regression, although they can also be applied to summary descriptive modeling (Agrawal et al. 1996).

Nonlinear Regression and Classification Methods

These methods consist of a family of techniques for prediction that fit linear and nonlinear combinations of basis functions (sigmoids, splines, polynomials) to combinations of the input variables. Examples include feed-forward neural networks, adaptive spline methods, and projection pursuit regression (see Elder and Pregibon [1996], Cheng and Titterington [1994], and Friedman [1989] for more detailed discussions). Consider neural networks, for example. Figure 7 illustrates the type of nonlinear decision boundary that a neural network might find for the loan data set. In terms of model evaluation, although 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. Also, 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 (Ripley 1994; Geman, Bienenstock, and Doursat 1992). Back propagation is a parameter-search method that performs gradient descent in parameter (weight) space to find a local maximum of the likelihood function starting from random initial conditions. Nonlinear regression methods, although powerful in representational power, can be difficult to interpret.

For example, although the classification boundaries of figure 7 might be more accurate than the simple threshold boundary of figure 6, the threshold boundary has the advantage that the model can be expressed, to some degree of certainty, as a simple rule of the form "if income is greater than threshold, then loan will have good status."

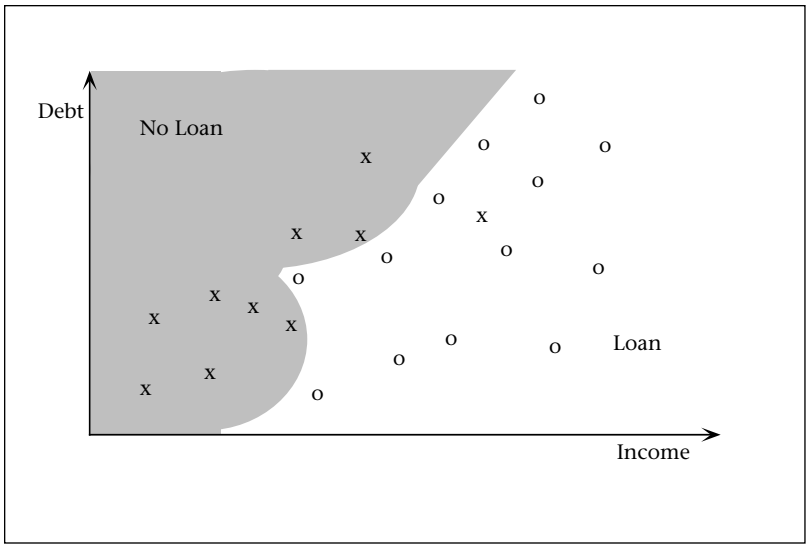

Figure 7. An Example of Classification Boundaries Learned by a Nonlinear Classifier (Such as a Neural Network) for the Loan Data Set.

Example-Based Methods

The representation is simple: Use representative examples from the database to approximate a model; that is, 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 8 illustrates the use of a nearest-neighbor classifier for the loan data set: The class at any new point in the two-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) is that a well-defined distance metric for evaluating the distance between data points is required. For the loan data in figure 8, this would not be a problem because income and debt are measured in the same units. However, if one wished to include variables such as the duration of the loan, sex, and profession, then it would require more effort to define a sensible metric between the variables. Model evaluation is typically based on cross-validation estimates (Weiss and 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 nonlinear regression methods, example-based methods are often asymptotically powerful in terms of approximation properties but, conversely, can be difficult to interpret because the model is implicit in the data and not explicitly formulated. Related techniques include kernel-density

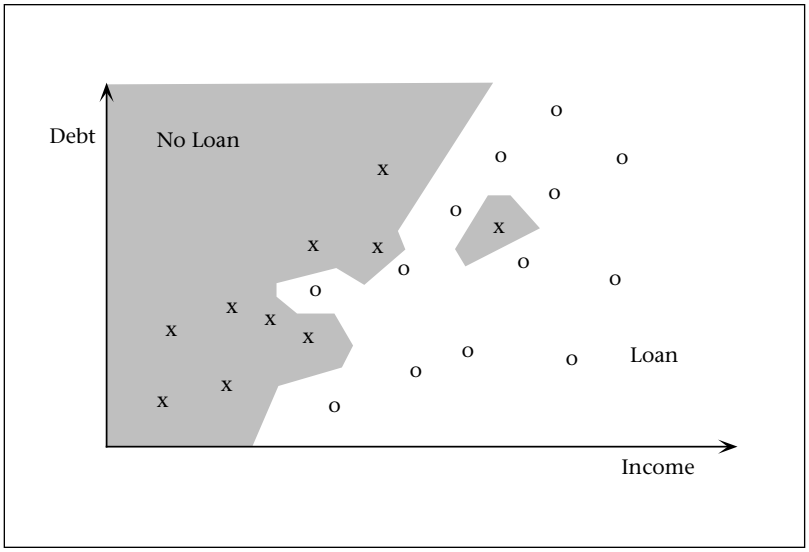

Figure 8. Classification Boundaries for a Nearest-Neighbor Classifier for the Loan Data Set.

estimation (Silverman 1986) and mixture modeling (Titterington, Smith, and Makov 1985).

Probabilistic Graphic Dependency Models

Graphic models specify probabilistic dependencies using a graph structure (Whittaker 1990; Pearl 1988). 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 AI 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. Recently, there has been significant work in both the AI and statistical communities on methods whereby both the structure and the parameters of graphic models can be learned directly from databases (Buntine 1996; Heckerman 1996). 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 useful in terms of reducing the model search space. Although still primarily in the research phase, graphic model induction methods are of particular interest to KDD because the graphic form of the model lends itself easily to human interpretation.

Understanding data mining and model induction at this component level clarifies the behavior of any data-mining algorithm and makes it easier for the user to understand its overall contribution and applicability to the KDD process.

Relational Learning Models

Although 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 to date on model-evaluation methods for relational learning is logical in nature. The extra representational power of relational models comes at the price of significant computational demands in terms of search. See Dzeroski (1996) for a more detailed discussion.

Discussion

Given the broad spectrum of data-mining methods and algorithms, our overview is inevitably limited in scope; many data-mining techniques, particularly specialized methods for particular types of data and domains, were not mentioned specifically. 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, which traditionally has been cast as a predictive regression task (autoregressive models, and so on). Recently, more general models have been developed for time-series applications, such as nonlinear basis functions, example-based models, and kernel methods. Furthermore, there has been significant interest in descriptive graphic and local data modeling of time series rather than purely predictive modeling (Weigend and Gershenfeld 1993). Thus, although different algorithms and applications might appear 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 behavior of any data-mining algorithm and makes it easier for the user to understand its overall contribution and applicability to the KDD process.

An important point is that each technique typically suits some problems better than others. For example, decision tree classifiers can be useful for finding structure in high-dimensional spaces and in problems with mixed continuous and categorical data (because tree methods do not require distance metrics). However, classification trees might not be suitable for problems where the true decision boundaries between classes are described by a second-order polynomial (for example). Thus, there is no universal data-mining method, and choosing a particular algorithm for a particular application is something of an art. In practice, a large portion of the application effort can go into properly formulating the problem (asking the right question) rather than into optimizing the algorithmic details of a particular data-mining method (Langley and Simon 1995; Hand 1994).

Because our discussion and overview of data-mining methods has been brief, we want to make two important points clear:

First, our overview of automated search focused mainly on automated methods for extracting patterns or models from data. Although this approach 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 database management system or to refer to humans visualizing patterns in data. 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.

Second, 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.

Application Issues

For a survey of KDD applications as well as detailed examples, see Piatetsky-Shapiro et al. (1996) for industrial applications and Fayyad, Haussler, and Stolorz (1996) for applications in science data analysis. Here, we examine criteria for selecting potential applications, which can be divided into practical and technical categories. The practical criteria for KDD projects are similar to those for other applications of advanced technology and include the potential impact of an application, the absence of simpler alternative solutions, and strong organizational support for using technology. For applications dealing with personal data, one should also consider the privacy and legal issues (Piatetsky-Shapiro 1995).

The technical criteria include considerations such as the availability of sufficient data (cases). In general, the more fields there are and the more complex the patterns being sought, the more data are needed. However, strong prior knowledge (see discussion later) can reduce the number of needed cases significantly. Another consideration is the relevance of attributes. It is important to have data attributes that are 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) are 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. Changing and time-oriented data, although making the application development more difficult, make it potentially much more useful because it is easier to retrain a system than a human. Finally, and perhaps one of the most important considerations, is prior knowledge. It is 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 on.

Research and Application Challenges

We outline some of the current primary research and application challenges for KDD. This list is by no means exhaustive and is intended to give the reader a feel for the types of problem that KDD practitioners wrestle with.

Larger databases: Databases with hundreds of fields and tables and millions of records and of a multigigabyte size are commonplace, and terabyte (10^{12} bytes) databases are beginning to appear. Methods for dealing with large data volumes include more efficient algorithms (Agrawal et al. 1996), sampling, approximation, and massively parallel processing (Holsheimer et al. 1996).

High dimensionality: Not only is there often a large number of records in the database, but there can also be a large number of fields (attributes, variables); so, 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 can model not only the general patterns in the data but also any noise specific to the data set, resulting in poor performance of the model on test data. Possible solutions include cross-validation, regularization, and other sophisticated statistical strategies.

Assessing of statistical significance: A problem (related to overfitting) occurs when the system is searching over many possible models. For example, if a system tests 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 that adjust the test statistic as a function of the search, for example, Bonferroni adjustments for independent tests or randomization testing.

Changing data and knowledge: Rapidly changing (nonstationary) data can make previously discovered patterns invalid. In addition, the variables measured in a given application database can 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, Piatetsky-Shapiro, and McNeill 1996). See also Agrawal and Psaila (1995) and Mannila, Toivonen, and Verkamo (1995).

Missing and noisy data: This problem is especially acute in business databases. U.S. census data reportedly have error rates as great as 20 percent in some fields. Important attributes can 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 1996; Smyth et al. 1996).

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 use 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 1996; Djoko, Cook, and Holder 1995).

Understandability of patterns: In many applications, it is important to make the discoveries more understandable by humans. Possible solutions include graphic representations (Buntine 1996; Heckerman 1996), rule structuring, natural language generation, and techniques for visualization of data and knowledge. Rule-refinement strategies (for example, Major and Mangano [1995]) can be used to address a related problem: The discovered knowledge might 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 the steps of the KDD process. Bayesian approaches (for example, Cheeseman [1990]) use prior probabilities over data and distributions as one form of encoding prior knowledge. Others employ deductive database capabilities to discover knowledge that is then used to guide the data-mining search (for example, Simoudis, Livezey, and Kerber [1995]).

Integration with other systems: A standalone discovery system might not be very useful. Typical integration issues include integration with a database management system (for example, through a query interface), integration with spreadsheets and visualization tools, and accommodating of real-time sensor readings. Examples of integrated KDD systems are described by Simoudis, Livezey, and Kerber (1995) and Stolorz, Nakamura, Mesrobiam, Muntz, Shek, Santos, Yi, Ng, Chien, Mechoso, and Farrara (1995).

Concluding Remarks: The Potential Role of AI in KDD

In addition to machine learning, other AI fields can potentially contribute significantly to various aspects of the KDD process. We mention a few examples of these areas here:

Natural language presents significant opportunities for mining in free-form text, especially for automated annotation and indexing prior to classification of text corpora. Limited parsing capabilities can help substantially in the task of deciding what an article refers to. Hence, the spectrum from simple natural language processing all the way to language understanding can help substantially. Also, natural language processing can contribute significantly as an effective interface for stating hints to mining algorithms and visualizing and explaining knowledge derived by a KDD system.

Planning considers a complicated data analysis process. It involves conducting complicated data-access and data-transformation operations; applying preprocessing routines; and, in some cases, paying attention to resource and data-access constraints. Typically, data processing steps are expressed in terms of desired postconditions and preconditions for the application of certain routines, which lends itself easily to representation as a planning problem. In addition, planning ability can play an important role in automated agents (see next item) to collect data samples or conduct a search to obtain needed data sets.

Intelligent agents can be fired off to collect necessary information from a variety of

sources. In addition, information agents can be activated remotely over the network or can trigger on the occurrence of a certain event and start an analysis operation. Finally, agents can help navigate and model the World-Wide Web (Etzioni 1996), another area growing in importance.

Uncertainty in AI includes issues for managing uncertainty, proper inference mechanisms in the presence of uncertainty, and the reasoning about causality, all fundamental to KDD theory and practice. In fact, the KDD-96 conference had a joint session with the UAI-96 conference this year (Horvitz and Jensen 1996).

Knowledge representation includes *ontologies,* new concepts for representing, storing, and accessing knowledge. Also included are schemes for representing knowledge and allowing the use of prior human knowledge about the underlying process by the KDD system.

These potential contributions of AI are but a sampling; many others, including human-computer interaction, knowledge-acquisition techniques, and the study of mechanisms for reasoning, have the opportunity to contribute to KDD.

In conclusion, we presented some definitions of basic notions in the KDD field. Our primary aim was to clarify the relation between knowledge discovery and data mining. We provided an overview of the KDD process and basic data-mining methods. Given the broad spectrum of data-mining methods and algorithms, our overview is inevitably limited in scope: There are many data-mining techniques, particularly specialized methods for particular types of data and domain. Although various algorithms and applications might 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.

This article represents a step toward a common framework that we hope will ultimately provide a unifying vision of the common overall goals and methods used in KDD. We hope this will eventually lead to a better understanding of the variety of approaches in this multidisciplinary field and how they fit together.

Acknowledgments

We thank Sam Uthurusamy, Ron Brachman, and KDD-96 referees for their valuable suggestions and ideas.

Note

1. Throughout this article, we use the term *pattern* to designate a pattern found in data. We also refer to models. One can think of patterns as components of models, for example, a particular rule in a classification model or a linear component in a regression model.

References

Agrawal, R., and Psaila, G. 1995. Active Data Mining. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), 3–8. Menlo Park, Calif.: American Association for Artificial Intelligence.

Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, I. 1996. Fast Discovery of Association Rules. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 307–328. Menlo Park, Calif.: AAAI Press.

Apte, C., and Hong, S. J. 1996. Predicting Equity Returns from Securities Data with Minimal Rule Generation. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 514–560. Menlo Park, Calif.: AAAI Press.

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

Berndt, D., and Clifford, J. 1996. Finding Patterns in Time Series: A Dynamic Programming Approach. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 229–248. Menlo Park, Calif.: AAAI Press.

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

Brachman, R., and Anand, T. 1996. The Process of Knowledge Discovery in Databases: A Human-Centered Approach. In *Advances in Knowledge Discovery and Data Mining,* 37–58, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Menlo Park, Calif.: AAAI Press.

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

Brodley, C. E., and Smyth, P. 1996. Applying Classification Algorithms in Practice. *Statistics and Computing*. Forthcoming.

Buntine, W. 1996. Graphical Models for Discovering Knowledge. In *Advances in Knowledge Discovery and Data Mining*, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 59–82. Menlo Park, Calif.: AAAI Press.

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

Cheeseman, P., and Stutz, J. 1996. Bayesian Classification (AUTOCLASS): Theory and Results. In *Advances in Knowledge Discovery and Data Mining,* eds.

U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 73–95. Menlo Park, Calif.: AAAI Press.

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.

Dasarathy, B. V. 1991. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. Washington, D.C.: IEEE Computer Society.

Djoko, S.; Cook, D.; and Holder, L. 1995. Analyzing the Benefits of Domain Knowledge in Substructure Discovery. In Proceedings of KDD-95: First International Conference on Knowledge Discovery and Data Mining, 75–80. Menlo Park, Calif.: American Association for Artificial Intelligence.

Dzeroski, S. 1996. Inductive Logic Programming for Knowledge Discovery in Databases. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 59–82. Menlo Park, Calif.: AAAI Press.

Elder, J., and Pregibon, D. 1996. A Statistical Perspective on KDD. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 83–116. Menlo Park, Calif.: AAAI Press.

Etzioni, O. 1996. The World Wide Web: Quagmire or Gold Mine? *Communications of the ACM* (Special Issue on Data Mining). November 1996. Forthcoming.

Fayyad, U. M.; Djorgovski, S. G.; and Weir, N. 1996. From Digitized Images to On-Line Catalogs: Data Mining a Sky Survey. *AI Magazine* 17(2): 51–66.

Fayyad, U. M.; Haussler, D.; and Stolorz, Z. 1996. KDD for Science Data Analysis: Issues and Examples. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 50–56. Menlo Park, Calif.: American Association for Artificial Intelligence.

Fayyad, U. M.; Piatetsky-Shapiro, G.; and Smyth, P. 1996. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1–30. Menlo Park, Calif.: AAAI Press.

Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R. 1996. *Advances in Knowledge Discovery and Data Mining*. Menlo Park, Calif.: AAAI Press.

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.; Madigan, D.; Pregibon, D.; and Smyth, P. 1996. Statistics and Data Mining. *Communications of the ACM* (Special Issue on Data Mining). November 1996. Forthcoming.

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

Guyon, O.; Matic, N.; and Vapnik, N. 1996. Discovering Informative Patterns and Data Cleaning. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 181–204. Menlo Park, Calif.: AAAI Press.

Hall, J.; Mani, G.; and Barr, D. 1996. Applying Computational Intelligence to the Investment Process. In Proceedings of CIFER-96: Computational Intelligence in Financial Engineering. Washington, D.C.: IEEE Computer Society.

Hand, D. J. 1994. Deconstructing Statistical Questions. *Journal of the Royal Statistical Society A*. 157(3): 317–356.

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

Heckerman, D. 1996. Bayesian Networks for Knowledge Discovery. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 273–306. Menlo Park, Calif.: AAAI Press.

Hernandez, M., and Stolfo, S. 1995. The MERGE-PURGE Problem for Large Databases. In Proceedings of the 1995 ACM-SIGMOD Conference, 127–138. New York: Association for Computing Machinery.

Holsheimer, M.; Kersten, M. L.; Mannila, H.; and Toivonen, H. 1996. Data Surveyor: Searching the Nuggets in Parallel. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 447–471. Menlo Park, Calif.: AAAI Press.

Horvitz, E., and Jensen, F. 1996. *Proceedings of the Twelfth Conference of Uncertainty in Artificial Intelligence*. San Mateo, Calif.: Morgan Kaufmann.

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

Kloesgen, W. 1996. A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 249–271. Menlo Park, Calif.: AAAI Press.

Kloesgen, W., and Zytkow, J. 1996. Knowledge Discovery in Databases Terminology. In *Advances in Knowledge Discovery and Data Mining*, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 569–588. Menlo Park, Calif.: AAAI Press.

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

Langley, P., and Simon, H. A. 1995. Applications of Machine Learning and Rule Induction. *Communications of the ACM* 38:55–64.

Major, J., and Mangano, J. 1995. Selecting among Rules Induced from a Hurricane Database. *Journal of Intelligent Information Systems* 4(1): 39–52.

Manago, M., and Auriol, M. 1996. Mining for OR. *ORMS Today* (Special Issue on Data Mining), February, 28–32.

Mannila, H.; Toivonen, H.; and Verkamo, A. I. 1995. Discovering Frequent Episodes in Sequences. In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), 210–215. Menlo Park, Calif.: American

Association for Artificial Intelligence.

Matheus, C.; Piatetsky-Shapiro, G.; and McNeill, D. 1996. Selecting and Reporting What Is Interesting: The KEFIR Application to Healthcare Data. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 495–516. Menlo Park, Calif.: AAAI Press.

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

Piatetsky-Shapiro, G. 1995. Knowledge Discovery in Personal Data versus Privacy—A Mini-Symposium. *IEEE Expert* 10(5).

Piatetsky-Shapiro, G. 1991. Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop. *AI Magazine* 11(5): 68–70.

Piatetsky-Shapiro, G., and Matheus, C. 1994. The Interestingness of Deviations. In Proceedings of KDD-94, eds. U. M. Fayyad and R. Uthurusamy. Technical Report WS-03. Menlo Park, Calif.: AAAI Press.

Piatetsky-Shapiro, G.; Brachman, R.; Khabaza, T.; Kloesgen, W.; and Simoudis, E., 1996. An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), eds. J. Han and E. Simoudis, 89–95. Menlo Park, Calif.: American Association for Artificial Intelligence.

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

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

Senator, T.; Goldberg, H. G.; Wooton, J.; Cottini, M. A.; Umarkhan, A. F.; Klinger, C. D.; Llamas, W. M.; Marrone, M. P.; and Wong, R. W. H. 1995. The Financial Crimes Enforcement Network AI System (FAIS): Identifying Potential Money Laundering from Reports of Large Cash Transactions. *AI Magazine* 16(4): 21–39.

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

Silberschatz, A., and Tuzhilin, A. 1995. On Subjective Measures of Interestingness in Knowledge Discovery. In Proceedings of KDD-95: First International Conference on Knowledge Discovery and Data Mining, 275–281. Menlo Park, Calif.: American Association for Artificial Intelligence.

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

Simoudis, E.; Livezey, B.; and Kerber, R. 1995. Using Recon for Data Cleaning. In Proceedings of KDD-95: First International Conference on Knowledge Discovery and Data Mining, 275–281. Menlo Park, Calif.: American Association for Artificial Intelligence.

Smyth, P.; Burl, M.; Fayyad, U.; and Perona, P. 1996. Modeling Subjective Uncertainty in Image Annotation. In *Advances in Knowledge Discovery and Data Mining,* 517–540. Menlo Park, Calif.: AAAI Press.

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

Stolorz, P.; Nakamura, H.; Mesrobian, E.; Muntz, R.; Shek, E.; Santos, J.; Yi, J.; Ng, K.; Chien, S.; Mechoso, C.; and Farrara, J. 1995. Fast Spatio-Temporal Data Mining of Large Geophysical Datasets. In Proceedings of KDD-95: First International Conference on Knowledge Discovery and Data Mining, 300–305. Menlo Park, Calif.: American Association for Artificial Intelligence.

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

U.S. News. 1995. Basketball's New High-Tech Guru: IBM Software Is Changing Coaches' Game Plans. *U.S. News and World Report,* 11 December.

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.

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

Zembowicz, R., and Zytkow, J. 1996. From Contingency Tables to Various Forms of Knowledge in Databases. In *Advances in Knowledge Discovery and Data Mining,* eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 329–351. Menlo Park, Calif.: AAAI Press.

Usama Fayyad is a senior researcher at Microsoft Research. He received his Ph.D. in 1991 from the University of Michigan at Ann Arbor. Prior to joining Microsoft in 1996, he headed the Machine Learning Systems Group at the Jet Propulsion Laboratory (JPL), California Institute of Technology, where he developed data-mining systems for automated science data analysis. He remains affiliated with JPL as a distinguished visiting scientist. Fayyad received the JPL 1993 Lew Allen Award for Excellence in Research and the 1994 National Aeronautics and Space Administration Exceptional Achievement Medal. His research interests include knowledge discovery in large databases, data mining, machine-learning theory and applications, statistical pattern recognition, and clustering. He was program cochair of KDD-94 and KDD-95 (the First International Conference on Knowledge Discovery and Data Mining). He is general chair of KDD-96, an editor in chief of the journal *Data Mining and Knowledge Discovery,* and coeditor of the 1996 AAAI Press book *Advances in Knowledge Discovery and Data Mining*.

Gregory Piatetsky-Shapiro is a principal member of the technical staff at GTE Laboratories and the principal investigator of the Knowledge Discovery in Databases (KDD) Project, which focuses on developing and deploying advanced KDD systems for business applications. Previously, he worked on applying intelligent front ends to heterogeneous databases. Piatetsky-Shapiro received several GTE awards, including GTE's highest technical achievement award for the KEFIR system for health-care data analysis. His research interests include intelligent database systems, dependency networks, and Internet resource discovery. Prior to GTE, he worked at Strategic Information developing financial database systems. Piatetsky-Shapiro received his M.S. in 1979 and his Ph.D. in 1984, both from New York University (NYU). His Ph.D. dissertation on self-organizing database systems received NYU awards as the best dissertation in computer science and in all natural sciences. Piatetsky-Shapiro organized and chaired the first three (1989, 1991, and 1993) KDD workshops and helped in developing them into successful conferences (KDD-95 and KDD-96). He has also been on the program committees of numerous other conferences and workshops on AI and databases. He edited and coedited several collections on KDD, including two books—*Knowledge Discovery in Databases* (AAAI Press, 1991) and *Advances in Knowledge Discovery in Databases* (AAAI Press, 1996)—and has many other publications in the areas of AI and databases. He is a coeditor in chief of the new *Data Mining and Knowledge Discovery* journal. Piatetsky-Shapiro founded and moderates the *KDD Nuggets* electronic newsletter (kdd@gte.com) and is the web master for Knowledge Discovery Mine (<http://info.gte.com/~kdd /index.html>).

Padhraic Smyth received a first-class-honors Bachelor of Engineering from the National University of Ireland in 1984 and an MSEE and a Ph.D. from the Electrical Engineering Department at the California Institute of Technology (Caltech) in 1985 and 1988, respectively. From 1988 to 1996, he was a technical group leader at the Jet Propulsion Laboratory (JPL). Since April 1996, he has been a faculty member in the Information and Computer Science Department at the University of California at Irvine. He is also currently a principal investigator at JPL (part-time) and is a consultant to private industry. Smyth received the Lew Allen Award for Excellence in Research at JPL in 1993 and has been awarded 14 National Aeronautics and Space Administration certificates for technical innovation since 1991. He was coeditor of the book *Advances in Knowledge Discovery and Data Mining* (AAAI Press, 1996). Smyth was a visiting lecturer in the Computational and Neural Systems and Electrical Engineering Departments at Caltech (1994) and regularly conducts tutorials on probabilistic learning algorithms at national conferences (including UAI-93, AAAI-94, CAIA-95, IJCAI-95). He is general chair of the Sixth International Workshop on AI and Statistics, to be held in 1997. Smyth's research interests include statistical pattern recognition, machine learning, decision theory, probabilistic reasoning, information theory, and the application of probability and statistics in AI. He has published 16 journal papers, 10 book chapters, and 60 conference papers on these topics.