Knowledge Discovery in Databases creates the context for developing the tools needed to control the flood of data facing organizations that depend on ever-growing databases of business, manufacturing, scientific, and personal information.

The KDD Process for Extracting Useful Knowledge from Volumes of Data

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth

As we march into the age of digital information, the problem of data overload looms ominously ahead. Our ability to analyze and understand massive datasets lags far behind our ability to gather and store the data. A new generation of computational techniques and tools is required to support the extraction of useful knowledge from the rapidly growing volumes of data. These techniques and tools are the subject of the emerging field of knowledge discovery in databases (KDD) and data mining.

Large databases of digital information are ubiquitous. Data from the neighborhood store's checkout register, your bank's credit card authorization device, records in your doctor's office, patterns in your telephone calls, and many more applications generate streams of digital records archived in huge databases, sometimes in so-called data warehouses.

Current hardware and database technology allow efficient and inexpensive reliable data storage and access. However, whether the context is business, medicine, science, or government, the datasets themselves (in raw form) are of little direct value. What is of value is the knowledge that can be inferred from the data and put to use. For example, the marketing database of a consumer

TERRY WIDENER

goods company may yield knowledge of correlations between sales of certain items and certain demographic groupings. This knowledge can be used to introduce new targeted marketing campaigns with predictable financial return relative to unfocused campaigns. Databases are often a dormant potential resource that, tapped, can yield substantial benefits.

This article gives an overview of the emerging field of KDD and data mining, including links with related fields, a definition of the knowledge discovery process, dissection of basic data mining algorithms, and an analysis of the challenges facing practitioners.

The value of storing volumes of data depends on our ability to extract useful reports, spot interesting events and trends, support decisions and policy based on statistical analysis and inference, and exploit the data to achieve business, operational, or scientific goals.

Impractical Manual Data Analysis

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 analyze current trends and changes in health-care data on a quarterly basis. The specialists then provide a report detailing the analysis to the sponsoring health-care organization; the report is then used as 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 geologic objects of interest, such as impact craters.

For these (and many other) applications, such manual probing of a dataset is slow, expensive, and highly subjective. In fact, such manual data analysis is becoming impractical in many domains as data volumes grow exponentially. Databases are increasing in size in two ways: the number N of records, or objects, in the database, and the number d of fields, or attributes, per object. Databases containing on the order of $N = 10^9$ objects are increasingly common in, for example, the astronomical sciences. The number d of fields can easily be on the order of 10^2 or even 10^3 in medical diagnostic applications. Who could be expected to digest billions of records, each with tens or hundreds of fields? Yet the true value of such data lies in the users' ability to extract useful reports, spot interesting events and trends, support decisions and policy based on statistical analysis and inference, and exploit the data to achieve business, operational, or scientific goals.

When the scale of data manipulation, exploration, and inference grows beyond human capacities, people look to computer technology to automate the bookkeeping. The problem of knowledge extraction from large databases involves many steps, ranging from data manipulation and retrieval to fundamental mathematical and statistical inference, search, and reasoning. Researchers and practitioners interested in these problems have been meeting since the first KDD Workshop in 1989. Although the problem of extracting knowledge from data (or observations) is not new, automation in the context of large databases opens up many new unsolved problems.

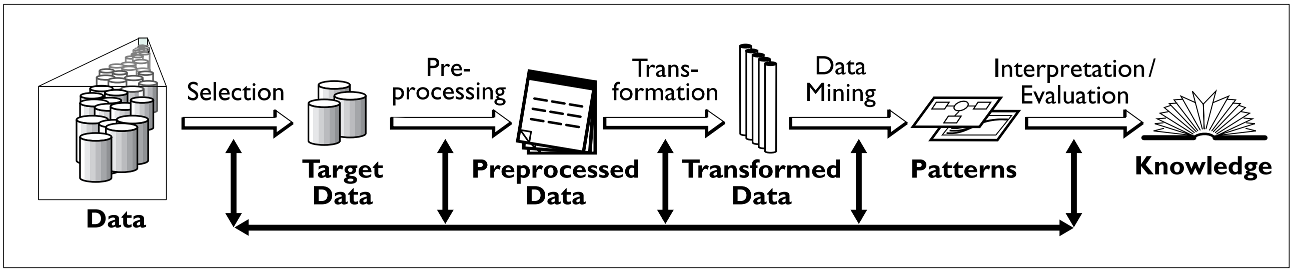
Definitions

Finding useful patterns in data is known by different names (including data mining) in different communities (e.g., knowledge extraction, information discovery, information harvesting, data archeology, and data pattern processing). The term “data mining” is used most by statisticians, database researchers, and more recently by the MIS and business communities. Here we use the term “KDD” to refer to the overall process of discovering useful knowledge from data. Data mining is a particular step in this process—application of specific algorithms for extracting patterns (models) from data. 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 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 leading to discovery of meaningless patterns.

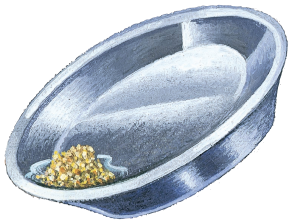
KDD has evolved, and continues to evolve, from the intersection of research in such fields as databases, machine learning, pattern recognition, statistics, artificial intelligence and reasoning with uncertainty, knowledge acquisition for expert systems, data visualization, machine discovery [7], scientific discovery, information retrieval, and high-performance computing. KDD software systems incorporate theories, algorithms, and methods from all of these fields.

Database theories and tools provide the necessary infrastructure to store, access, and manipulate data. Data warehousing, a recently popularized term, refers to the current business trend of collecting and cleaning transactional data to make them available for online analysis and decision support. A popular approach for analysis of data warehouses is called online analytical processing (OLAP).[1] OLAP tools focus on providing multidimensional data analysis, which is superior to SQL (a standard data manipulation language) in computing summaries and breakdowns along many dimensions. While current OLAP tools target interactive data analysis, we expect they will also include more automated discovery components in the near future.

Figure 1. Overview of the steps constituting the KDD process



Fields concerned with inferring models from data—including statistical pattern recognition, applied statistics, machine learning, and neural networks—were the impetus for much early KDD work. KDD largely relies on methods from these fields to find patterns from data in the data mining step of the KDD process. A natural question is: How is KDD different from these other fields? KDD focuses on the overall process of knowledge discovery from data, including how the data is stored and accessed, how algorithms can be scaled to massive datasets and still run efficiently, how results can be interpreted and visualized, and how the overall human-machine interaction can be modeled and supported. KDD places a special emphasis on finding understandable patterns that can be interpreted as useful or interesting knowledge. Scaling and robustness properties of modeling algorithms for large noisy datasets are also of fundamental interest.



Statistics has much in common with KDD. Inference of knowledge from data has a fundamental statistical component (see [2] and the article by Glymour on statistical inference in this special section for more detailed discussions of the relationship between KDD and statistics). Statistics provides a language and framework for quantifying the uncertainty resulting 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 over the fact that if one searches long enough in any dataset (even randomly generated data), one can find patterns that appear to be statistically significant but in fact are not. This issue is of fundamental importance to KDD. There has been substantial progress in understanding such issues in statistics in recent years, much directly relevant to KDD. Thus, data mining is a legitimate activity as long as one understands how to do it correctly. KDD can also be viewed as encompassing a broader view of modeling than statistics, aiming to provide tools to automate (to the degree possible) the entire process of data analysis, including the statistician's art of hypothesis selection.

The KDD Process

Here we present our (necessarily subjective) perspective of a unifying process-centric framework for KDD. The goal is to provide an overview of the variety of activ-

[1] See *Providing OLAP to User Analysts: An IT Mandate* by E.F. Codd and Associates (1993).

ities in this multidisciplinary field and how they fit together. We define the KDD process [4] as:

The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

Throughout this article, the term *pattern* goes beyond its traditional sense to include models or structure in data. In this definition, data comprises a set of facts (e.g., cases in a database), and pattern is an expression in some language describing a subset of the data (or a model applicable to that subset). The term *process* implies there are many steps involving data preparation, search for patterns, knowledge evaluation, and refinement—all repeated in multiple iterations. The process is assumed to be *nontrivial* in that it goes beyond computing closed-form quantities; that is, it must involve search for structure, models, patterns, or parameters. The discovered patterns should be *valid* for 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* for the user or task. Finally, the patterns should be *understandable*—if not immediately, then after some postprocessing.

This definition implies we can define quantitative measures for evaluating extracted patterns. In many cases, it is possible to define measures of certainty (e.g., estimated classification accuracy) or utility (e.g., gain, perhaps in dollars saved due to better predictions or speed-up in a system's response time). Such notions as novelty and understandability are much more subjective. In certain contexts, understandability can be estimated through simplicity (e.g., number of bits needed to describe a pattern). An important notion, called *interestingness*, is usually taken as an overall measure of pattern value, combining validity, novelty, usefulness, and simplicity. Interestingness functions can be explicitly defined or can be manifested implicitly through an ordering placed by the KDD system on the discovered patterns or models.

In practice, a large portion of the applications effort can go into properly formulating the problem (asking the right question) rather than optimizing the algorithmic details of a particular data mining method.

Data mining is a step in the KDD process consisting of an enumeration of patterns (or models) over the data, subject to some acceptable computational-efficiency limitations. Since the patterns enumerable over any finite dataset are potentially infinite, and because the enumeration of patterns involves some form of search in a large space, computational constraints place severe limits on the subspace that can be explored by a data mining algorithm.

The KDD process is outlined in Figure 1. (We did not show all the possible arrows to indicate that loops can, and do, occur between any two steps in the process; also not shown is the system's performance element, which uses knowledge to make decisions or take actions.) The KDD process is interactive and iterative (with many decisions made by the user), involving numerous steps, summarized as:

1. Learning the application domain: includes relevant prior knowledge and the goals of the application
2. Creating a target dataset: includes selecting a dataset or focusing on a subset of variables or data samples on which discovery is to be performed
3. Data cleaning and preprocessing: includes basic operations, such as removing noise or outliers 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, as well as deciding DBMS issues, such as data types, schema, and mapping of missing and unknown values
4. Data reduction and projection: includes finding useful features to represent the data, depending on the goal of the task, and 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 function of data mining: includes deciding the purpose of the model derived by the data mining algorithm (e.g., summarization, classifica

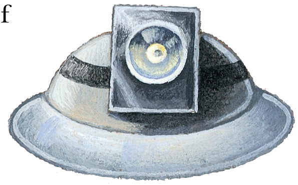
tion, regression, and clustering)

6. Choosing the data mining algorithm(s): includes selecting method(s) to be used for searching for patterns in the data, such as deciding which models and parameters may be appropriate (e.g., models for categorical data are different from models on vectors over reals) and matching a particular data mining method with the overall criteria of the KDD process (e.g., the user may be more interested in understanding the model than in its predictive capabilities)
7. Data mining: includes searching for patterns of interest in a particular representational form or a set of such representations, including classification rules or trees, regression, clustering, sequence modeling, dependency, and line analysis
8. Interpretation: includes interpreting the discovered patterns and possibly returning to any of the previous steps, as well as possible visualization of the extracted patterns, removing redundant or irrelevant patterns, and translating the useful ones into terms understandable by users
9. Using discovered knowledge: includes incorporating this knowledge into the performance system, taking actions based on the knowledge, or simply documenting it and reporting it to interested parties, as well as checking for and resolving potential conflicts with previously believed (or extracted) knowledge.

Most previous work on KDD focused primarily on the data mining step. However, the other steps are equally if not more important for the successful application of KDD in practice. We now focus on the data mining component, which has received by far the most attention in the literature.

Data Mining

Data mining involves fitting models to or determining patterns from observed data. The fitted models play the role of inferred knowledge. Deciding whether or not the models reflect useful knowledge is a part of the overall interactive KDD process for which subjective human judgment is usually required. A wide variety and number of data mining algorithms are described in the literature—from the fields of statistics, pattern recognition, machine learning, and databases. Thus, an overview discussion can often consist of long lists of seemingly unrelated, and highly specific algorithms. Here we take a somewhat reductionist viewpoint. Most data mining algorithms can be viewed as compositions of a few basic techniques and principles. In particular, data mining algorithms consist largely of some specific mix of three components:



- **The model.** There are two relevant factors: the function of the model (e.g., classification and clustering) and the representational form of the model (e.g., a linear function of multiple variables and a Gaussian probability density function). A model contains parameters that are to be determined from the data.
- **The preference criterion.** A basis for preference of one model or set of parameters over another, depending on the given data. The criterion is usually some form of goodness-of-fit function of the model to the data, perhaps tempered by a smoothing term to avoid overfitting, or generating a model with too many degrees of freedom to be constrained by the given data.
- **The search algorithm.** The specification of an algorithm for finding particular models and parameters, given data, a model (or family of models), and a preference criterion.

A particular data mining algorithm is usually an instantiation of the model/preference/search components (e.g., a classification model based on a decision-tree representation, model preference based on data likelihood, determined by greedy search using a particular heuristic. Algorithms often differ largely in terms of the model representation (e.g., linear and hierarchical), and model preference or search methods are often similar across different algorithms. The literature on learning algorithms frequently does not state clearly the model representation, preference criterion, or search method used; these are often mixed up in a description of a particular algorithm. The reductionist view clarifies the independent contributions of each component.

Model Functions

The more common model functions in current data mining practice include:

- Classification: maps (or classifies) a data item into one of several predefined categorical classes.
- Regression: maps a data item to a real-value prediction variable.
- Clustering: maps a data item into one of several categorical classes (or clusters) in which the classes must

be determined from the data—unlike classification in which the classes are predefined. Clusters are defined by finding natural groupings of data items based on similarity metrics or probability density models.

- Summarization: provides a compact description for a subset of data. A simple example would be the mean and standard deviations for all fields. More sophisticated functions involve summary rules, multivariate visualization techniques, and functional relationships between variables. Summarization functions are often used in interactive exploratory data analysis and automated report generation.
- Dependency modeling: describes significant dependencies among variables. Dependency models exist at two levels: structured and quantitative. The structural level of the model specifies (often in graphical form) which variables are locally dependent; the quantitative level specifies the strengths of the dependencies using some numerical scale.
- Link analysis: determines relations between fields in the database (e.g., association rules [1] to describe which items are commonly purchased with other items in grocery stores). The focus is on deriving multi-field correlations satisfying support and confidence thresholds.
- Sequence analysis: models sequential patterns (e.g., in data with time dependence, such as time-series analysis). The goal is to model the states of the process generating the sequence or to extract and report deviation and trends over time.

Model Representation

Popular model representations include decision trees and rules, linear models, nonlinear models (e.g., neural networks), example-based methods (e.g., nearest-neighbor and case-based reasoning methods), probabilistic graphical dependency models (e.g., Bayesian networks [6]), and relational attribute models. Model representation determines both the flexibility of the model in representing the data and the interpretability of the model in human terms. Typically, the more complex models may fit the data better but may also be more difficult to understand and to fit reliably. While researchers tend to advocate complex models, practitioners involved in successful applications often use simpler models due to their robustness and interpretability [3, 5].

Model preference criteria determine how well a particular model and its parameters meet the criteria of the KDD process. Typically, there is an explicit quantitative criterion embedded in the search algorithm (e.g., the maximum likelihood criterion of finding the parameters that maximize the probability of the observed data). Also, an implicit criterion (reflecting the subjective bias of the analyst in terms of which models are initially chosen for consideration) is often used in the outer loops of the KDD process.

Search algorithms are of two types: parameter search, given a model, and model search over model space. Finding the best parameters is often reduced to an optimization problem (e.g., finding the global maximum of a nonlinear function in parameter space). Data mining algorithms tend to rely on relatively simple optimization techniques (e.g., gradient descent), although in principle more sophisticated optimization techniques are also used. Problems with local minima are common and dealt with in the usual manner (e.g., multiple random restarts and searching for multiple models). Search over model space is usually carried out in a greedy fashion.

Researchers and practitioners should ensure that the potential contributions of KDD are not overstated and that users understand the true nature of those contributions along with their limitations.

A brief review of specific popular data mining algorithms can be found in [4, 5]. An important point is that each technique typically suits some problems better than others. For example, decision-tree classifiers can be very useful for finding structure in high-dimensional spaces and are also useful in problems with mixed continuous and categorical data (since tree methods do not require distance metrics). However, classification trees with univariate threshold decision boundaries may not be suitable for problems where the true decision boundaries are nonlinear multivariate functions. Thus, there is no universally best data mining method; choosing a particular algorithm for a particular application is something of an art. In practice, a large portion of the applications 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.

The high-level goals of data mining tend to be predictive, descriptive, or a combination of predictive and descriptive. A purely predictive goal focuses on accuracy in predictive ability. A purely descriptive goal focuses on understanding the underlying data-generating process—a subtle but important distinction. In prediction, a user may not care whether the model reflects reality as long as it has predictive power (e.g., a model combining current financial indicators in some nonlinear manner to predict future dollar-to-deutsche-mark exchange rates). A descriptive model, on the other hand, is interpreted as a reflection of reality (e.g., a model relating economic and demographic variables to educational achievements used as the basis for social policy recommendations to cause change). In practice, most KDD applications demand some degree of both predictive and descriptive modeling.

Research Issues and Challenges

Current primary research and application challenges for KDD [4, 5] include:

- **Massive datasets and high dimensionality.** Multigigabyte databases with millions of records and large numbers of fields (attributes and variables) are commonplace. These datasets create combinatorially explosive search spaces for model induction and increase the chances that a data mining algorithm will find spurious patterns that are not generally valid. Possible solutions include very efficient algorithms, sampling, approximation methods, massively parallel processing, dimensionality reduction techniques, and incorporation of prior knowledge.
- **User interaction and prior knowledge.** An analyst is usually not a KDD expert but a person responsible for making sense of the data using available KDD techniques. Since the KDD process is by definition interactive and iterative, it is a challenge to provide a high-performance, rapid-response environment that also assists users in the proper selection and matching of appropriate tools and techniques to achieve their goals. There needs to be more emphasis on human-computer interaction and less emphasis on total automation—with the aim of supporting both expert and novice users. Many current KDD methods and tools are not truly interactive and do not easily incorporate prior knowledge about a problem except in simple ways. Use of domain knowledge is important in all steps of the KDD process. For example, Bayesian approaches use prior probabilities over data and distributions as one way of encoding prior knowledge (see [6] and Glymour's article on statistical inference in this special section). Others employ deductive database capabilities to discover knowledge that is then used to guide the data mining search.
- **Overfitting and assessing statistical significance.** When an 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. Proper assessment of statistical significance is often missed when the system searches many possible models. Simple methods to handle this problem include adjusting the test statistic as a function of the search (e.g., Bonferroni adjustments for independent tests) and randomization testing, although this area is largely unexplored.
- **Missing data.** This problem is especially acute in business databases. Important attributes may be missing if the database was not designed with discovery in mind. Missing data can result from operator error, actual system and measurement failures, or from a revision of the data collection process over time (e.g., new variables are measured, but they were considered unimportant a few months before). Possible solutions include more sophisticated statistical strategies to identify hidden variables and dependencies.
- **Understandability of patterns.** In many applications, it is important to make the discoveries more understandable by humans. Possible solutions include graphical representations, rule structuring, natural language generation, and techniques for visualization of data and knowledge. Rule refinement strategies can also help address a related problem: Discovered knowledge may be implicitly or explicitly redundant.
- **Managing changing data and knowledge.** Rapidly changing (nonstationary) 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.

- **Integration.** A standalone 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 accommodation of real-time sensor readings. Highly interactive human-computer environments as outlined by the KDD process permit both human-assisted computer discovery and computer-assisted human discovery. Development of tools for visualization, interpretation, and analysis of discovered patterns is of paramount importance. Such interactive environments can enable practical solutions to many real-world problems far more rapidly than humans or computers operating independently. There are a potential opportunity and a challenge to developing techniques to integrate the OLAP tools of the database community and the data mining tools of the machine learning and statistical communities.
- **Nonstandard, multimedia, and object-oriented data.** A significant trend is that databases contain not just numeric data but large quantities of nonstandard and multimedia data. Nonstandard data types include nonnumeric, nontextual, geometric, and graphical data, as well as nonstationary, temporal, spatial, and relational data, and a mixture of categorical and numeric fields in the data. Multimedia data include free-form multilingual text as well as digitized images, video, and speech and audio data. These data types are largely beyond the scope of current KDD technology.

Conclusions

Despite its rapid growth, the KDD field is still in its infancy. There are many challenges to overcome, but some successes have been achieved (see the articles by Brachman on business applications and by Fayyad on science applications in this special section). Because the potential payoffs of KDD applications are high, there has been a rush to offer products and services in the market. A great challenge facing the field is how to avoid the kind of false expectations plaguing other nascent (and related) technologies (e.g., artificial intelligence and neural networks). It is the responsibility of researchers and practitioners in this field to ensure that the potential contributions of KDD are not overstated and that users understand the true nature of the contributions along with their limitations.

Fundamental problems at the heart of the field remain unsolved. For example, the basic problems of statistical inference and discovery remain as difficult and challenging as they always have been. Capturing the art of analysis and the ability of the human brain to synthesize new knowledge from data is still unsurpassed by any machine. However, the volumes of data to be analyzed make machines a necessity. This niche for using machines as an aid to analysis and the hope that the massive datasets contain nuggets of valuable knowledge drive interest and research in the field. Bringing together a set of varied fields, KDD creates fertile ground for the growth of new tools for managing, analyzing, and eventually gaining the upper hand over the flood of data facing modern society. The fact that the field is driven by strong social and economic needs is the impetus to its continued growth. The reality check of real applications will act as a filter to sift the good theories and techniques from those less useful.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, I. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining,* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI/MIT Press, Cambridge, Mass., 1996.
2. Elder, J., and Pregibon, D. A statistical perspective on KDD. In *Advances in Knowledge Discovery and Data Mining,* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI/MIT Press, Cambridge, Mass., 1996.
3. Fayyad, U., and Uthurusamy, R., Eds. *Proceedings of KDD-95: The First International Conference on Knowledge Discovery and Data Mining.* AAAI Press, Menlo Park, Calif., 1995.
4. Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining,* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI/MIT Press, Cambridge, Mass., 1996.
5. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., Eds. *Advances in Knowledge Discovery and Data Mining*; AAAI/MIT Press, Cambridge, Mass., 1996.
6. Heckerman, D. Bayesian networks for knowledge discovery. *Advances in Knowledge Discovery and Data Mining,* U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI/MIT Press, Cambridge, Mass., 1996.
7. Langley, P., and Simon, H.A. Applications of machine learning and rule induction. *Commun. ACM 38,* 11 (Nov. 1995), 55–64.

Additional references for this article can be found at http://www.research.microsoft.com/research/datamine/CACM-DM-refs/.

USAMA FAYYAD is a senior researcher at Microsoft Research and a distinguished visiting scientist at the Jet Propulsion Laboratory of the California Institute of Technology. He can be reached at fayyad@microsoft.com.

GREGORY PIATETSKY-SHAPIRO is a principal member of the technical staff at GTE Laboratories. He can be reached at gps@gte.com.

PADHRAIC SMYTH is an assistant professor of computer science at the University of California, Irvine, and technical group leader at the Jet Propulsion Laboratory of the California Institute of Technology. He can be reached at smyth@ics.uci.edu.