Taming the Giants and the Monsters: Mining Large Databases for Nuggets of Knowledge

Usama Fayyad
Microsoft Research
One Microsoft Way
Redmond, WA 98052, USA
http://www.research.microsoft.com/~fayyad
fayyad@microsoft.com

SUGGESTED Lead intro paragraph (a bit of realistic fiction by Usama):

[Phoenix, AZ, 3/4/97, 9:30PM, office of COO, TLI Communications Corp]

"But how do we find those guys?" asked Underwood, "there's some 10 billion records on 100 million customers in the main database." Edwards turned his head slowly, "Well I thought we had all those fields about every call they make, we have all this info on the customers, can't you ask the database to figure out which lines are connected to fax machines?" He was losing his patience with his Data Division manager. "What good is all this data to TLI otherwise?"

Underwood had to reiterate: "These databases are for billing purposes, doing this kind of stuff is just not possible..." He was interrupted, Elderman could no longer contain himself: "But those MPI guys over in Florida are already doing this! I hear they are even basing future network expansion plans on this info. The information is in there, I just know it!" he said. "What would it take to get it out?" he wondered aloud.

Although fictional, scenarios like this one (perhaps where stronger language is employed) are unfortunately becoming commonplace in organizations large and small. Large data sets are now a fact of life for most organizations; in fact, we have gathered so much data that asking even the simplest questions has become a big challenge. Consider a simple applications that determines if two rows in a data table are likely to be the same, given that it is acceptable for "a few fields" to differ. While this "find-similar" problem appears simple, and one could think of several ways to achieve it, executing it on a massive data store is far from straightforward. Large data stores are now a fact of life for most organizations.

A *gigabyte* is a quantity of information; it represents about 10^9 bytes of stored information. The word derives from the Latin *giga*, meaning "giant." The next unit up is the *terabyte*, from the Greek *teras*, meaning "monster", represents 10^{12} bytes. Quite appropriately, in certain database circles, the terabyte is also referred to as the "terrorbyte": a term I first heard used by Jim Gray.

The modern information revolution is creating huge data stores which, instead of offering increased productivity and new opportunities, are threatening to drown us in a flood of useless information. Those who dare to tap their databases for the simplest browsing operation are usually blasted by an explosion of irrelevant and unimportant facts. Even people who do not "own" large databases face the data overload phenomenon when they surf the Web.

The biggest challenge now facing the digital community is how to sift through these data dumps to find useful information. Meeting this challenge is the primary goal of the emerging discipline of data mining and knowledge discovery in databases (KDD).[1]

Source of the Flood

The simplest daily task like making a call, using a credit card, or buying groceries usually leaves a computer record. As of 1996, the biggest data warehouse in the world--the Wal-Mart system built on NCR Teradata--reportedly contained 11 terabytes ($11x10^{12}$ bytes) of data. In science, the data glut problem has been known many years now and is only getting worse. The satellites of NASA's Earth Observing System are projected to generate more than a terabyte of data *per day* when they become operational in 1998.

While these massive databases hold potential treasures of scientific, business, and social knowledge, their sheer sizes make traditional, interactive approaches to analyzing the data and extracting knowledge practically impossible. Business questions such as: “What products need to be improved?” or “What customers are likely to switch to the competition?” cannot be answered with a standard query, nor can typical scientific and government agency questions. [2]

These facts aside, perhaps the most significant reason for the growing need for data mining and KDD systems is the success of database systems themselves and their proloferation in all sorts of organizations. Until a few years ago, only the largest organizations could afford to build and maintain large databases. Today, even the smallest business can quickly build up large databases--by capturing transactions from cash registers, recording customer interaction events in excruciating detail, and using Web servers that produce gigabytes of log data after short operational periods. The result is an explosion of database “owners.”

But unlike large organizations that can afford to maintain dedicated analysis departments, this new generation of data owners have to do the analysis themselves. As I’ll explain, these data owners are not finding tools suitable for use by nonanalysts. To make the situation worse, in a competitive environment (be it in business, science, or government), ignoring data assets can quickly lead to major losses. A telling example is that of MCI, which exploited a next-generation, flexible billing and information service to create marketing programs the competition couldn’t match.

How can you understand/visualize the contents of a database containing millions of records, each record having hundreds or even thousands of fields? Humans are not well-suited to reasoning about problems involving voluminous data or high-dimensional spaces. Consider a store that carries 1000 items and serves 10,000 customers a day. An interesting question--such as what items tend to be purchased together during one month--requires analysis of millions of data points in thousands of dimensions, a task humans are simply incapable of performing. Dimensions and data volumes are exploding even further when one considers that patterns over weeks, months, years, or seasons are being examined. Similar problems face advertisers and marketers when they select targets for a mailing campaign, by investors who select where to put their money, by government agencies that monitor fraud, by engineers in manufacturing plants when they have to track down failures or mysterious bottlenecks, and even by scientists who have to analyze large amounts of data.

Database systems have been developed to efficiently retrieve, update, and maintain these data stores. The query language for interacting with data stores is, of course, SQL. SQL, however, was designed for a different purpose than to aid humans in exploring, understanding, or analyzing data. A SQL query represents an exact logical description of a target data set. Unfortunately, the information needed for decision making is not easy to specify in a query language like SQL.

Say you want to detect fraudulent transactions in a credit-card database. Writing a query that distinguishes fraudulent records from legitimate ones is extremely difficult, given that the nature of fraud constantly changes. (In contrast, a data mining approach would be to take cases that are known to be fraudulent and others known to be “clean” and let the computer build a model that distinguishes between the two). Other examples of decision-support questions include: “Get me records that look like these but not like those others,” “Tell me how the data clusters,” or “Give me a summary of the contents of this large database.” All these queries are not easily expressed in SQL. While the data has been growing from underneath us, we managed to develop systems to store it without realizing that the interface for access, usage, and analysis is, essentially, broken.

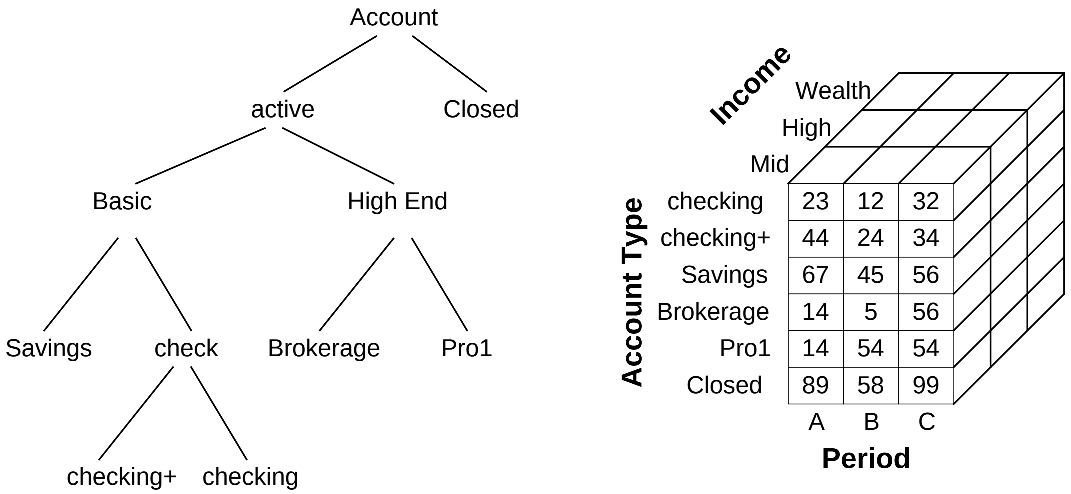
The Current Environment

The natural place to look for data analysis solutions is the statistical community, which has been concerned with problems of data analysis for over a hundred years. The problem sof data mining are fundamentally statistical in nature: inferring patterns or models from data. Unfortunately, statistical tools have little to offer the untrained user; statistical packages are usually designed for statisticians. Furthermore, most such tools are not integrated with database environments.

Even if one were to overcome these initial hurdles, the last one is sure to cause problems: the challenge of scalability. A typical statistical package assumes small data sets and low dimensionality. For example, suppose you want to do some simple database segmentation by running the popular *k*-means clustering algorithm: a basic simple method for clustering data. Let’s say you have managed to lay your hands on an implementation in some statistical library. The first operation the routine will execute is “load data.” In most settings, this operation will also be the last as the process hits its memory limits and comes crashing down.

Within the database community, however, some help has been arriving for the power user. The popular trend of data warehousing has transformed the primary purpose of the database system from reliable storage to decision support. To respond to the new needs of data access for decision support, OLAP technology, as first advocated by E. F. Codd,[4] has recently become popular.

The current emphasis of OLAP systems is on supporting query-driven exploration of the data warehouse. Part of this goal entails precomputing aggregates along data “dimensions” in the multidimensional data store (see Figure 1). Because the number of possible aggregates is exponential in the number of “dimensions,” much of the work in OLAP systems involves deciding which aggregates to precompute and how to derive other aggregates (or estimate them reliably) from the precomputed projections. Note that the multidimensional store may not necessarily be instantiated in memory, as in principle it could be derived dynamically from the underlying relational database. For efficiency purposes, some OLAP systems employ “lazy” strategies in precomputing summaries and incrementally build up a cache of aggregates. This approach has produced a variety of new technology acronyms: MOLAP (multidimensional OLAP), ROLAP (relational OLAP), HOLAP (hybrid OLAP), and most recently, DOLAP (desktop OLAP). However, despite the alphabet soup, the fundamental problems of dealing with many dimensions remain unaddressed.



	A	B	C
checking	23	12	32
checking+	44	24	34
Savings	67	45	56
Brokerage	14	5	56
Pro1	14	54	54
Closed	89	58	99

Figure 1. An Illustration of the OLAP Cube

While OLAP systems enable the execution of queries that would usually be expensive, and while they provide a more convenient environment for manipulating and visualizing data, they are dependent on human-driven exploration and on the analyst to formulate and test new hypothesis in the data. Exploration is guided by queries issued by the user that specify level in dimension hierarchy (for example, “drill down” operations to move into a lower level of detail). Inference or modeling is relegated completely to the analyst who is expected to “find” patterns of interest via visualization in low-dimensional projections (or summaries) of data. While this helps a bit, it does not really make much progress towards giving the analyst a handle on the analysis problem. As I explain later, simple projections and aggregations are far from constituting sufficient help.

The KDD Process

KDD--which combines methods from database theory, statistics and pattern recognition, AI, data visualizations, high-performance computing, and other areas--involves the process of finding useful and interesting structure in data. Structure denotes patterns or models; a pattern is classically defined as a parsimonious description of a subset of the data.

The KDD process (see Figure 2) involves many steps, including data selection, data cleaning, data transformation, and reduction. A critical step in this process is data mining--the enumeration of patterns (and/or models; I use the terms interchangeably here) over the data. Given a data set, it is possible to enumerate many more patterns than there are data items (often infinitely many patterns/models). The challenge in data mining is to decide which patterns to search for and enumerate. Most such patterns are likely to be uninteresting, superfluous, and perhaps even incorrect--products of random chance or simple statements of the extremely obvious. Hence, an important step in the KDD process is the evaluation phase, in which “interesting” patterns are identified after data mining. In fact, the final product of the KDD process is patterns of interest--what we call “knowledge.” The evaluation steps include methods for visualizing patterns and data to help the end user understand the data better.

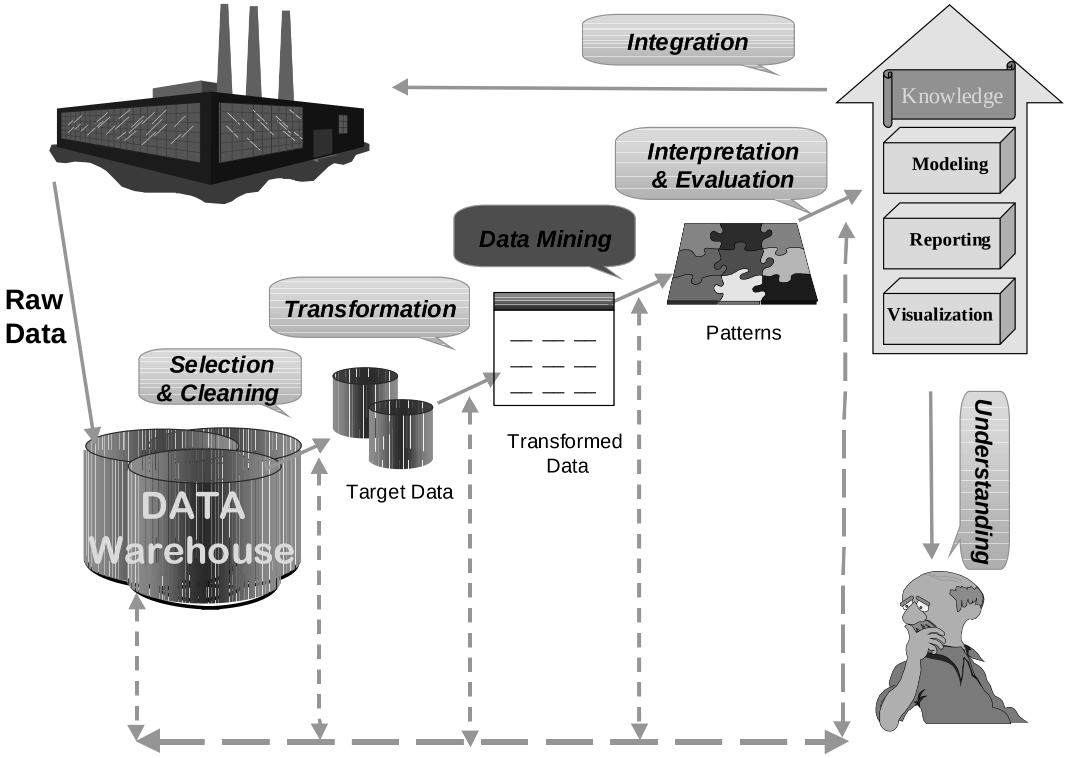


Figure 2. The KDD Process

Data mining is a central and crucial step among the many steps of the KDD process. However, historically, the term was used in statistics literature to refer to the (frowned upon) practice of “fishing” around in data without having *a priori* hypothesis to verify. Such practice may indeed lead one to find statistically significant patterns *even in completely random data*. KDD systems must guard against the dangers of such “mines” in data mining and protect the user from them. However, with the risk of mines also comes the promise of treasures. The road from raw data to “knowledge” is a long and arduous one. The raw data generated by some process (manufacturing, telephone switch, customer billing) is usually not easy to retrieve. The first step is to collect and organize such data in a data warehouse that provides a unified logical view of diverse aspects of business or organization. In building a warehouse, an organization is forced to address basic issues of data cleaning and consolidation, such as mapping many fields with differing names into one (for example, one business database reportedly had 27 different spellings of “K-Mart”), deriving fields that might be missing from some tables, establishing the correct keys, and defining an indexing scheme and table layout strategy that would satisfy anticipated lookup queries. Because of the volumes involved, access to data in flexible nonpredefined ways will be slow and impractical.

An explanation of each step of the KDD process follows. While Figure 2 separates the steps conceptually, in practice these steps may be combined and interleaved in many ways.

Selection and cleaning. Assuming that you understand the analysis problem, goals, constraints, and bottlenecks involved, the KDD process starts with selecting the target data--the subset of the data warehouse relevant to the task. Selection is usually accompanied by cleaning the data. While some cleaning may have been done at the warehousing stage, this step involves operations related explicitly to the specific data mining task, such as deciding how to treat missing field values, summarizing multiple rows into single entries ("roll-ups"), detecting known noisy cases or outliers, and enforcing other specialized constraints. An important part of this task is how to identify different transactions as related to the same entity. Such identification may be done by matching similar records and bringing all relevant information about the same entity together to facilitate the next step.

Transformation. This step involves transforming the data into a form suitable for the data mining method. Transformations can include dimensionality reduction steps (replacing several fields with a derived field) or data reduction steps (mapping multiple records for a single entity and "rolling-up" derived quantities into the resulting single record). It also includes adding new fields that the user may judge relevant to the problem at hand. The result is a fairly large data set, and typically involves unifying the data into a single database relation (table) for the next step.

Data mining. In the data mining phase, an algorithm is applied to find structures of interest (patterns or models) in the data table. A pattern could be a predictive model, data summary, data segmentation, or a model of dependencies (links) in data. The type of patterns generated depends on the data mining task. From a finite data set, a very large (possibly infinite) number of possible patterns and models can be extracted. Hence, exhaustive or optimal search methods are usually impractical and heuristic search techniques are required.

Interpretation and evaluation. This step evaluates the selected patterns for the purpose of deciding which ones are interesting, and interprets them in terms understandable by the user or acceptable by the application that will use them. Evaluation criteria include: validity, utility, novelty, understandability (simplicity), and interestingness.

Integration. Finally, the discovered "knowledge" is either used by people to increase their understanding or integrated with the process that has generated it (to perform prediction or control, for example). This is a crucial step and can be very challenging: what action to take in the real world based on a new discovery?

Data Mining Tasks

When data mining, you are essentially looking for interesting structures within the data. These structures take the form of patterns or models, as I described earlier. However, the act of finding patterns in data has been studied for a long time in other fields--such as statistics, pattern recognition, and data. So why is a new term "data mining" being used?

The goals that are uniquely addressed by data mining fall into four categories:

* **Scaling analysis to large databases**. What if data is too large to fit in main memory? Are there abstract primitive accesses to the data that can provide mining algorithms with the information to drive the search for patterns? How do we minimize--or sometimes even avoid--having to scan the large database in its entirety? These are questions addressed by researchers in data mining.

* **Automated search**. Instead of relying solely on the human analyst to enumerate and create hypotheses (or candidate patterns to look for), we can let the computer perform much of this tedious and data intensive work. Data mining is about making exploration of large databases easy, convenient, and practical for those who have data but not years of training in statistics or data analysis.

* **Understandable models.** Helping users understand and gain insight from data by focusing on the extraction of patterns that are easy to understand or can be turned into meaningful reports and summaries. In most KDD applications, making the extracted "knowledge" understandable to the end user is particularly important. Hence, using a nonlinear regression model (a neural net, for example) to predict a variable from data is insufficient. It is important to help the user trade complexity for understandability and gain insight from the derived models/patterns.

* **Finding patterns and models that are "interesting" to users**. Classical methodologies for scoring models have focused on notions of accuracy (how well does the model predict data?) and utility (how beneficial is the derived pattern--Dollars saved/gained, and so on?). While these classical measures are well understood and have been studied formally in statistics and decision analysis, the KDD community is also concerned with new measures, such as how to gauge the understandability of a model or the novelty of a pattern.

For example, according to R. Kohavi of Silicon Graphics (SGI) Data Mining Group, in a test of the SGI MineSet data mining tool on a database of demographics, a very interesting rule emerged: "With 99.7 percent certainty, all individuals who are husbands are also males." While a human may find this pattern trivial, to a computer, it looks just as interesting as a pattern that detects double-billing practices in a large database of healthcare billing records. Such fraudulent practices were actually uncovered by IBM's data mining group using data from the Australian government healthcare agency.

Data Mining Methods

We have so far covered an overview of the KDD process, motivated data mining and explained how it fits in with other relevant fields. We now turn out attention to describing what one does when one is mining data. Data mining activities fall into five broad categories:
*** Predictive modeling** targets predicting one or more fields in the data using the rest of the fields. When the variable being predicted is categorical (to approve/reject a loan, for example), the problem is called classification. When the variable is continuous (such as expected profit/loss), the problem is referred to as regression. Classification is a traditionally well-studied problem. The simplest approaches include standard techniques for linear regression, generalized linear regression, and discriminant analysis. Methods popular in data mining include decision trees, rules, neural networks (non-linear regression) , readial basis functions, and many others.
* **Database clustering (segmentation)** is finding the clusters in the data that consist of similar subsets of records. Unlike in predictive modeling, there is no target variable. Examples include segmenting a customer database into clusters of similar customers, which enables the design of a separate marketing strategy for each segment. There are many methods for clustering data. Popular approaches include K-Means algorithm, hierarchical agglomarative methods, and mixture modeling using the Estimation-Maximization (EM) algorithm for fitting probabilistic mixture models to data. A good description with applications are covered in the chapter by Cheeseman and Stutz [1].
* **Dependency modeling** targets estimation of the underlying joint probability density of the data. If you know what the density function is, you one can answer any question of interest about the data. Density estimation is a difficult problem and can be extremely difficult in high-dimensional data sets. A scalable method for clustering and density estimation is given in [7].
*** Data summarization** involves methods for finding a compact description for a subset of data. For example, rather than enumerating all customers who have two phone lines, the system may return a short description (most of them have fax machines). Unlike the previous tasks, which work on records (rows), summarization is usually used in report generation and exploratory data analysis and is concerned with finding relations among fields (columns). The exact dependencies usually reflect some external rules (such as the account-type field determining the interest rate field). Usually the weaker dependencies--the ones that are less than 100-percent correct--are more interesting. For example, age and profession fields may be predictors of income, which may in turn be predictive of the loan default or repayment. Furthermore, in market-basket analysis, you may find that whenever a shopper wine and flowers, they usually buy a greeting card. Such a pattern could suggest a new way of shelving the merchandise (placing items that are likely to trigger a purchase between these items). Scalable techniques for finding such association rules are described by Agrawal et al in [1].
* **Change and deviation analysis** focuses on discovering the most significant changes in data from previous or expected values or in identifying similar sequences of events. Piatetsky-Shapiro et al in [1] Methods for change and deviation analysis have to account for the sequence in which data appears, which distinguishes them from the other methods above.

Applications of KDD

The best applications of KDD are in fields that involve information-rich, rapidly changing environments, require knowledge-based decisions that have high payoff, and have sufficient and accessible data**.** Perhaps the most pervasive business application is database marketing, which relies on the analysis of customer information to tailor offers to customer preferences. Data mining has also been applied successfully in fraud detection, manufacturing, banking, and telecommunications,[3] as well as in astronomy, remote sensing, and protein sequencing.[5] The less typical applications include IBM Advanced Scout, which analyzes basketball statistics to find patterns of play and is used by several NBA teams,[6] and a system that analyzes foster children records in Washington State to help design better social services. A noteworthy example is IBM's recent success in helping Mellon Bank achieve significantly higher accuracies and lower costs in a credit-card marketing campaign. What is especially interesting here is that the use of scalable and more general tools enabled the IBM team to outperform the bank's own statistics department, which traditionally managed advertising for the bank. They were able to save the bank significant mailing costs as well as manage to turn around the analysis in record time. Business successes are routinely reported on the web as well in trade magazines.

Challenges and Expectations

While early applications of KDD have yielded significant payoffs, the field is rife with problems that require a significant amount of research and investigation. The classical statistical problems of model inference from finite observations and managing uncertainty in the derived results remain fundamental to KDD. Because data mining emphasizes machine-driven exploration and the enumeration of many patterns, there are other challenges involving the search for patterns/models in a huge search space. Search algorithms can only explore a small part of the possible space and have to rely on heuristic methods.

A question of purpose. Many of the traditional methods for data analysis were designed with the assumption that the data was collected for the purpose of analysis. This subtle assumption and its implications are often overlooked. In typical data mining applications, it does not hold. Rather, the data is collected for a business purpose: accounting, billing, monitoring, and so on. Using this data for analysis purposes such as prediction and segmentation rules out many approaches that are designed to be applicable in specialized analysis situations.

A question of scale. Most algorithms developed in statistics and pattern recognition (and data analysis in general) assume that data can be read into the computer's memory and manipulated efficiently. New algorithms are needed to deal with large volumes of data that do not fit in memory. In addition, data mining algorithms enumerate many more patterns over data than is possible with traditional "manual" approaches. This fact introduces some very subtle dangers. For example, given data that is completely random, we can mathmatically show that given enough paterns enumerated from the data, with probability close to 1.0 you can find patterns that hold for the given data. Of course, this result is due purely to random chance as, by construction, the data cannot have any patterns in it. To complicate matters further, traditional techniques for assessing statistical significance all assume small data sets. With large sample sizes, everything that fits the sample becomes significant!

How does your data grow? Large databases grow over time--they do not represent a snapshot of a static, underlying data generating process. Hence, modeling the "entire" data set with one model is almost hopeless. Taking a random sample doesn't help. What is needed is the ability to identify various regimes in the data and model each regime with its local model, which is what scalable clustering methods and partitioning methods such as decision trees do.

Interaction with data visualization. Another interesting prospect is the possibility of a new regime for visualizing massive data sets. Instead of the traditional human-driven techniques that rely on human analysis capabilities, we envision techniques that employ data mining algorithms that extract and simplify patterns, which makes it easier to visualize the data. For example, a clustering algorithm can break apart a large database into homogeneous segments that may be much easier to visualize. Finding such segments in the first place is difficult, and nearly impossible, for manual analysis.

Other challenges. In addition, most data mining algorithms, even the most efficient ones, cannot be expected to scan the entire data set (which can take days or weeks). This fact makes sampling strategies and approximation methods essential. Finally, there is a growing volume of nonstructured data--in the form of text, images, and everything related to the Web--that requires new data mining algorithms. Solutions to these challenging research problems will be needed to fully exploit the potential of mining the information bases of the next century.

While these challenges provide plenty of issues for researchers to engage in for a long time to come, the good news is that a new generation of data mining tools is emerging. While some of these tools are still a bit too complicated for use by business end-users, they are significantly simpler to use than the previous options offered by statictical analysis packages. The tools are being integrated with operational databases and in some cases can scale to very large databases. The payoff has been significant for those who have mined their growing databases. This will lead to an explosion in interest and with it the hype. The reader needs to be careful to realize that while successes are being achieved, data mining tools cannot find gold when it is not present. Not every database can be mined, regardless of what some might claim. A careful balance between realistic expectations and careful investigation of the mined goods is still a necessity. Despite all the traps, mines, and dangers, the rewards of a good data mining expedition can be quite gratifying.

References

1. Fayyad, U., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.
2. Way, J. and E. Smith. "The Evolution of Synthetic Aperture Radar Systems and their Progression to the EOS SAR." *IEEE Trans. on Geoscience and Remote Sensing* 29(6), 1991.
3. Brachman, R., T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro, and E. Simoudis. "Industrial Applications of Data Mining and Knowledge Discovery." *Communications of ACM* 39(11), November 1996.
4. Codd, E. F. "Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate." Publication of E. F. Codd and Associates, 1993.
5. Fayyad, U., D. Haussler, and P. Stolorz. "Mining Science Data." *Communications of the ACM* 39(11), November 1996.
6. Bhandari, I. et al. "Advanced Scout: Data Mining and Knowledge Discovery in NBA Data" (summary note). *Data Mining and Knowledge Discovery* 1(1), 1997.
7. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications." *Data Mining and Knowledge Discovery* 1(2), 1997.

About the Author:

Usama Fayyad is a Senior Researcher at Microsoft Research. His research interests include knowledge discovery in large databases, data mining, machine learning theory and applications, statistical pattern recognition, and clustering. After receiving the Ph.D. degree from the University of Michigan, Ann Arbor in 1991, he joined the Jet Propulsion Laboratory (JPL), California Institute of Technology (until 1996). At JPL, he headed the Machine Learning Systems Group 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 NASA Exceptional Achievement Medal. He was program co-chair of KDD-94 and KDD-95 (the First International Conference on Knowledge Discovery and Data Mining) and a general chair of KDD-96. Fayyad is an editor-in-chief of the *Data Mining and Knowledge Discovery* journal, and a co-editor of *Advances in Knowledge Discovery and Data Mining* book (AAAI/MIT Press, 1996).

Abstract:

Data stores are piling up at a menacing pace, leaving organizations to wonder what they are to do with their growing "write-only" stores. The problem is that the current interface to databases: SQL was not designed to support operations required for decision support application, data exploration, navigation, and visualization. We outline what data mining is, its role in the knowledge discovery for databases (KDD) process, and relate it to OLAP and data warehousing. This article is intended to provide an overview of this growing new field and outline the possibilities for what database systems could, and should, be providing their users. We also outline what is achievable versus what remains as a challenge for future research and development.

To learn more about this area, there are many web sites, reachable from:

- http://www.research.microsoft.com/datamine: Data Mining and Knowledge Discovery Journal.
- http://www.research.microsoft.com/datamine/acm-contents.html: Special issue of the Communications of the ACM, Nov. 96.
- http://www.kdnuggets.com: Knowledge Discovery Mine: moderated discussion list, archival of other resources, tools, and news items:
- http://www-aig.jpl.nasa.gov/kdd98 (and 95-97): International conferences on Knowledge Discovery and Data Mining.