Guest Editors' Introduction

Data Mining: From Serendipity to Science

Serendipity refers to making fortunate discoveries quite by accident, so transitioning to a science might seem like an inherently difficult task. But that is just what modern data mining hopes to accomplish.

Naren Ramakrishnan
Virginia Tech

Ananth Y. Grama
Purdue University

In his novel, *The Last Voyage of Somebody the Sailor*, John Barth writes "you don't reach Serendip by plotting a course for it. You have to set out in good faith for elsewhere and lose your bearings ... serendipitously." This is perhaps an apt description of the discovery process carried out to query large-scale data repositories. Specifically, if we knew what to look for, the process of discovery would be trivial and the destination, unexciting.

The idea of unsupervised learning from basic facts (axioms) or from data has fascinated researchers for decades. Knowledge discovery engines try to extract general inferences from facts or training data. Statistical methods take a more structured approach, attempting to quantify data by known and intuitively understood models. The problem of gleaning knowledge from existing data sources poses a significant paradigm shift from these traditional approaches.

The size, noise, diversity, dimensionality, and distributed nature of typical data sets make even formal problem specification difficult. Moreover, you typically do not have control over data generation. This lack of control opens up a Pandora's box filled with issues such as overfitting, limited coverage, and missing/incorrect data with high dimensionality.

Once specified, solution techniques must deal with complexity, scalability (to meaningful data sizes), and presentation. This entire process is where data mining makes its transition from serendipity to science.

EMERGING SCIENCE

With the Web's emergence as a large distributed data repository and the realization that huge online databases can be tapped for significant commercial gain, interest in data-mining techniques has virtually exploded. As the field evolves from its roots in artificial intelligence (AI), statistics, and algorithmics, it is gaining a unique character of its own.

Researchers have explored core mining techniques such as clustering, classification, associations, and time series analysis. While making significant progress on techniques and their application, they have also uncovered new challenges.

Deriving qualitative assessments from quantitative data—inferring that people will use alternate gas stations if the price of gas is 10 percent higher, for example—remains a challenge. Since most data-mining techniques are heuristic, bounded-error approximation techniques and approximate algorithms will eventually play a significant role. The coupling between data mining and presentation (visualization) will tighten. Applications in scientific domains will play a critical role in furthering computational simulation as a key design technology.

This extremely wide scope of data-mining applications falls into various data-mining domains.

DATA-MINING DOMAINS

Goals common to all data-mining applications are the detection, interpretation, and prediction of qualitative or quantitative patterns in data. To characterize and evaluate patterns, data-mining algorithms employ a wide variety of models from machine learning, statistics, experimental algorithmics, AI, and databases. These techniques also draw from mathematical approaches such as approximation theory and dynamical systems.

The applications driving the development of these algorithms also influence the basis, assumptions, and methodological issues underlying them and their application. For example, developments in molecular biology have led to improved algorithms for sequence analysis and for mining categorical data.

Perspectives

Five recurrent perspectives—induction, compression, querying, approximation, and search—underlie most research in data mining.

Induction. The most common perspective, induction—proceeding from the specific to the general—has its roots in AI and machine learning. It answers questions like "given 10 specific examples of good travel destinations, what are the characteristics of a favorable tourist attraction?"

Thus, induction is typically implemented as a search

0018-9162/99/$10.00 © 1999 IEEE

through the space of possible hypotheses. Such searches usually employ some special characteristic or aspect to arrive at a good generalization—"tropical islands are favorable," for example. Systems such as Progol (not Prolog), FOIL (First Order Inductive Learning), and Golem view induction as reversing the deduction process in first-order logic inference.

Compression. Of course, several general concepts can apply to one set of data, so mining techniques typically look for the most succinct or easily described pattern. This principle, known as Occam's Razor, effectively equates mining to compression, where the learned patterns are in some sense "smaller to describe" than exhaustively enumerating the original data itself.

The emergence of computational learning theory in the 1980s and the feasibility of models such as MDL (the Minimum Description Length principle) provided a solid theoretical foundation to this perspective. Several commercial data-mining systems employ this view of data mining as compression to determine the effectiveness of mined patterns: If a pattern mined from 10 data points is itself 16 "features" long, then mining might provide no tangible benefit.

Querying. This unique perspective comes from the database systems community. Since most business data resides in industrial databases and warehouses, commercial companies view mining as a sophisticated form of database querying. Research based on this perspective seeks to enhance the expressiveness of query languages like SQL to allow queries like "Find all the customers with deviant transactions."

Other database perspectives concentrate on enhancing the underlying data model. (The relational model is good for abstracting and querying data. Is it also a good model for mining?) Or they offer metaquery languages ("Find me a pattern that connects something about writers' backgrounds and the characters in their novels"). Still others concentrate on developing interactive techniques for exploring databases.

Approximation. This view of mining starts with an accurate (exact) model of the data and deliberately introduces approximations in the hope of finding some hidden structure in the data.

Such approximations might involve dropping higher-order terms in a harmonic expansion or collapsing two or more nearby entities into one—viewing three connected nodes as one in a graph, for instance.

One technique that has found extensive use in document retrieval is called Latent Semantic Indexing. This technique, patented by Bellcore, uses linear algebraic matrix transformations and approximations to identify hidden structures in word usage, thus enabling searches that go beyond simple keyword matching. Related techniques have also been used in Karhunen-Loeve expansions for signal processing and principal-component analysis in statistics.

Search. This perspective relates to induction, but focuses on efficiency. Our favorite example is the widely popular work on association rules at IBM Almaden that uses the forward-pruning nature of patterns (frequent itemsets) to restrict the space of possible patterns.

In mining very large databases, reducing complexity and overhead from out-of-core computations is key.

Other viewpoints

Besides the taxonomy we've just presented, there are other ways to categorize data-mining techniques. Techniques fall into categories based on

- their induced representations (decision trees, rules, correlations, deviations, trends, or associations);
- the data they operate on (continuous, time series, discrete, labeled, or nominal); or
- application domains (finance, economic models, biology, Web log mining, or semistructured models for abstracting from Web pages).

Patterns, in turn, can be characterized based on accuracy, precision, expressiveness, interpretability, parsimony, "surprisingness," "interestingness," or actionability (by the business enterprise). For example, a pattern that translates into sound organizational decisions is better than one that is accurate and interesting but provides no tangible commercial benefit. A classic example is the Automated Mathematician program, which purportedly mined the pattern "All numbers greater than one can be expressed as the sum of 1s."

IN THIS ISSUE

The five articles in this issue cover a gamut of topics that include algorithmics, query languages, mining Web hyperlinks, and full-fledged integrated systems.

Venkatesh Ganti and colleagues present a survey of association, clustering, and classification algorithms. It is an excellent starting point for new researchers as well as a good overview for current researchers in data mining. Two key issues are reducing complexity and reducing the overhead incurred by out-of-core computations.

Jiawei Han and colleagues present an integrated approach to database mining and querying that uses a taxonomy of constraints to guide the process. This strategy controls complexity by incorporating domain-specific restrictions into the data-mining process and also provides the miner with a declarative high-level interface. The authors envision that such techniques will receive widespread acceptance for online mining of large information warehouses.

Typical data analysis requires considerable user input to guide the discovery/analysis process. Joseph Hellerstein and colleagues describe the Control project, which uses techniques for tightening the loop in the data

Promising Research: Is the Old *New* Again?

The most promising data-mining research centers on the following issues.

Secondary, tertiary, and distributed storage

An old maxim in the database community is that algorithms and systems for secondary storage are qualitatively different from those that operate on main memory. The issues that dominate "traditional" algorithm development are unlike those that have driven the design of specialized strategies for databases such as indices, query processors, and locking mechanisms.

We feel this trend will manifest itself in algorithms optimized for mining from secondary, tertiary, and distributed storage. (Secondary storage includes devices like hard disks, for example. Tertiary storage involves physical media retrieved by robotic arms. Distributed storage involves data that resides in several storage locations.) Specifically, these algorithms will incorporate new relational primitives for mining (similar to the Cube operator[1] for data summarization). They will also help curb the "curse of dimensionality," which refers to the difficulties in mining data when you have only a few data points to determine many dimensions (characteristics of interest). These algorithms will also work with well-publicized test data suites (akin to OLAP benchmarks), provide more efficient data updates (for incremental mining), and push mining functionality *into* database systems.

Other efforts involve revisiting the logical models of databases; the goal is enhanced representations for mining that include relational and constraint-based descriptions.

Privacy issues and techniques to ensure confidentiality during mining will underscore developments in the business and corporate sectors.

Sampling and anytime techniques

Sampling pervades many aspects of databases. For example, query optimization, data layout organization, and physical database tuning depend on sampling-derived estimates.

Sampling can also help develop an abstraction of the database to be mined; it is most useful for patterns that are supposed to hold *universally* in a schema (such as in determining whether "All transactions involve both a cosmetic and a food product").[2] Sampling also relates to approximation, as described later.

The *anytime query-answering technique*[3] provides an interruptible query-processing algorithm, wherein an answer to the query is available every time (anytime) and the quality of the answer monotonically improves with time. Such anytime techniques will find more visibility in a data-mining system where there are limited resources and where "good-enough" answers are acceptable. Many popular data-mining techniques, such as those based on association rules, can be elegantly recast as anytime algorithms.

Interaction: closing the loop

We cannot overemphasize that data-mining processes are strongly interactive and repetitive. Strategies that intelligently close the loop (provide feedback to earlier parts of the system) in a data-mining system or systems that address both data exploration and analysis will become increasingly prevalent in specialized domains.

For example, systems that mine a pattern and also suggest parts of the data space to explore next could improve performance in the long run, while sacrificing some exploration time in the short term. The active and rule-based elements of a database system can help automate this aspect.[4]

Data approximations and approximate algorithms

We envision significant advances in bounded-error approximations, which could help alleviate some problems induced by extremely large data sets, heuristic results, and time- and memory-intensive algorithms. Approximation techniques let users execute expensive algorithms on reduced data sets or help develop approximate algorithms for desired operations.

They also allow tunable mining techniques that achieve results of the desired precision quickly given the appropriate user input. We find this important given the nature of most mining techniques.

Parallel and distributed computing

The sheer magnitude of many data-mining applications and the distributed nature of data require the use of parallel and distributed computing. Many parallel and distributed algorithms for data mining differ from traditional (numerically oriented) algorithms in their emphasis on data handling as opposed to computational power. Consequently, these parallel algorithms potentially perform significant excess computation over and above their serial counterparts to avoid data movement. In heterogeneous asynchronous dynamic environments with frequent failures and uncalibrated resources, conventional programming paradigms and performance evaluation techniques are inadequate, and new ones must be developed.

Large-scale parallel computers also provide large amounts of memory as well as significant aggregate memory bandwidth. This presents interesting trade-offs for out-of-core computations that typically minimize memory sweeps, often at the expense of increased computation.

References

1. J. Gray et al., "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," *J. Data Mining and Knowledge Discovery*, Vol. 1, No. 1, 1997, pp. 29-53.
2. J. Kivinen and H. Mannila, "The Use of Sampling in Knowledge Discovery," *Proc. 13th ACM Symp. Principles of Database Systems*, ACM Press, New York, 1994, pp. 77-85.
3. S.V. Vrbsky and J.W.S. Liu, "Approximate: A Query Processor That Produces Monotonically Improving Approximate Answers," *IEEE Trans. Knowledge and Data Eng.*, Dec. 1993, pp. 1,056-1,068.
4. J. Widom and S. Ceri, eds., *Active Database Systems*, Morgan Kaufmann Publishers, San Francisco, Calif., 1996.

analysis process. Specifically, making the discovery process visible to the user at all times makes it easier to guide or terminate the process after it achieves the desired results. The basic challenge is one of trading off the quality and accuracy of the mining process.

Soumen Chakrabarti and colleagues present the Clever system for mining the link structure of Web pages on the Internet. Clever was recently featured in *Scientific American* (June 1999). It models the real-life phenomenon underlying the way people connect Web pages and uses this information to form the abstraction for a data-mining system. This has important implications for online communities and for social and collaborative filtering techniques in e-commerce.

Finally, George Karypis and colleagues present the Chameleon system for automatically finding clusters in spatial data. This use of clustering is now prevalent in link-based analyses (as in fraudulent credit card transaction detection), semistructured data (for information integration and extracting schema), spatial databases, and problems envisaged in bio-informatics.

Putting this issue together has been a source of great pleasure and a learning experience. The overwhelming response to this issue from our research community is a testimony to the vitality and interest in this area.

The challenge of reducing serendipity to a science dates back to time immemorial. In one of the oldest known fairy tales, "The Three Princes of Serendip" (translated from Sanskrit), three young men from Persia set out to find the fabled silk islands of what now comprise Sri Lanka. They never found silk, but they did manage to find a land truly exotic and amazing. Their journey changed them all beyond cognition.

Our search for Serendip continues. ❖

Naren Ramakrishnan *is an assistant professor of computer science at Virginia Tech. His research interests include recommender systems, computational science, and data mining. Ramakrishnan has a PhD in computer sciences from Purdue University. He is a member of the IEEE, ACM, ACM SIGART, and the AAAI.*

Ananth Y. Grama *is an assistant professor of computer sciences at Purdue University. His research interests include parallel and distributed computing, large-scale simulations, data compression, analysis, and mining. Grama has a PhD in computer science from the University of Minnesota, Twin Cities. He is a member of Sigma Xi.*

Contact Naren Ramakrishnan at Virginia Tech, Dept. of Computer Science, Virginia Tech, VA 24061; naren@cs.vt.edu. Contact Ananth Y. Grama at Purdue Univ., Dept. of Computer Sciences, West Lafayette, IN 47907; ayg@cs.purdue.edu.