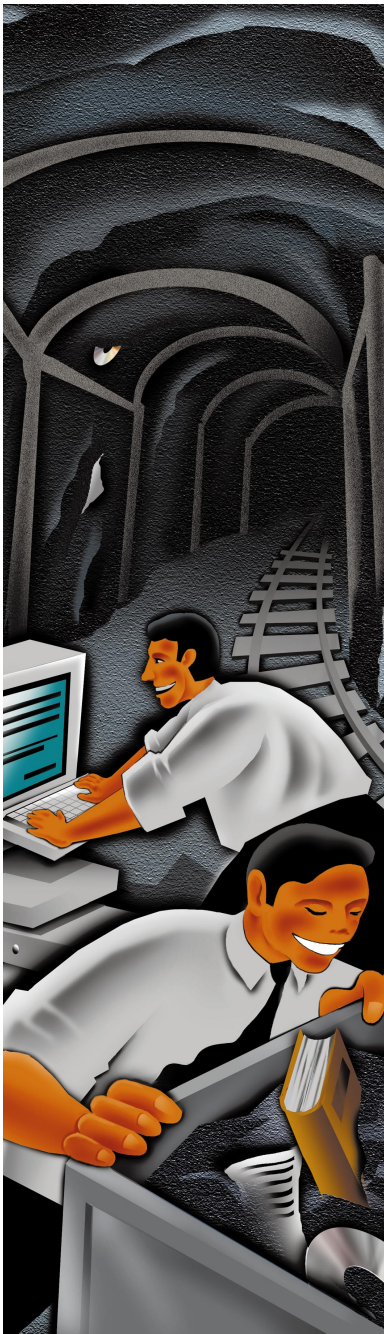


Albert Y. Zomaya
The University of Western Australia

Tarek El-Ghazawi
George Mason University

Ophir Frieder
Illinois Institute of Technology



Parallel and Distributed Computing for Data Mining

Because of advances in computing, Internet, and sensor technologies, recent years have seen a massive increase in data. For example, traditional remote-sensing instruments have provided measurements at a few spectral bands, but today's hyperspectral instruments can deliver observations at hundreds of spectral bands. Also, the NASA Earth Observing System (EOS), when it becomes fully operational, will deliver close to a terabyte of remote sensing data per day, and NASA will have petabytes of archived data in the next few years.

Similar scenarios will occur in other areas: we will see large numbers of radiological images generated in hospitals and immense product and customer databases as the Internet and e-commerce continue to expand.¹ Exploring useful information from such data will require efficient parallel algorithms running on high-performance computing systems with powerful parallel I/O capabilities. Without such systems and techniques, invaluable data and information will remain undiscovered. We also need to quantify the time it takes to retrieve pertinent information, for time-critical applications such as credit card fraud detection.²⁻⁹

Terms you should know

Generally, *data mining* is the discovery of valuable information from large data volumes, using computationally efficient techniques. The data-mining literature contains many other definitions as well.^{1,4,5,10-13} These definitions vary somewhat, depending on the application and whether data has been organized into a database. To this end, some people have defined data mining as the discovery of something new—or unknown patterns—from the facts in a database. Others have defined it as the automatic or semiautomatic exploration and analysis of voluminous data to discover meaningful patterns and rules.

Starting points

ACM Special Interest Group on Knowledge Discovery in Data and Data Mining:

www.acm.org/sigkdd

Comm. ACM Special Issue on Data Mining (Vol. 39, No. 11, Nov. 1996):

www.research.microsoft.com/research/datamine/acm-contents.htm

The Data Mine:

www.cs.bham.ac.uk/~anp/TheDataMine.html

Data Mining and Knowledge Discovery Journal:

www.research.microsoft.com/datamine

Database Research Groups:

www.informatik.uni-trier.de/~ley/db/groups.html

The Machine Learning network Online Information Service (projects related to knowledge discovery in databases, data mining, machine learning, case-based reasoning, and knowledge acquisition):

www.ml原因.org/community/projects-index.html

NSF Data Mining Workshop:

www.isi.edu/nsf

On-Line Software for Clustering and Multivariate Analysis:

www.pitt.edu/~csna/software.html

Quest Data Mining Home Page:

www.almaden.ibm.com/cs/quest

URLs for Data Mining:

lib.stat.cmu.edu/~bill/DMLIST.html

Usama Fayyad et al., *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., 1996; MIT Press, Cambridge, Mass., 1996;

www.research.microsoft.com/~fayyad/advances-kdd/fayap.html

Knowledge discovery from databases is a related, widely used process. Some people view data mining as only one step in the KDD process, which includes application understanding, target data set creation, corrupt data removal or correction, data reduction, data mining, and interpretation of mined patterns.

The type of information sought often determines which data-mining technique to use. Basic data-mining techniques include clustering, association rule discovery, classification, sequential pattern discovery, and outlier detection. Other techniques exist, but we mention here only the main ones—those either more frequently used or more general. Furthermore, although data mining is a relatively new field, born of the recent influx of data collection, data-mining techniques themselves are not necessarily new and have roots in other disciplines, such as pattern recognition and statistics.

Clustering is the process of partitioning or grouping a given set of data points into distinct groups, or clusters, such that the similarity between the data points in one cluster is maximized and the similarity between data points in different clusters is minimized.⁷ You can use clus-

tering for a wide range of applications—from grouping companies with similar stock behavior or similar growth to identifying genes and proteins that have similar functions.

Given a number of transactions of item sets, *association rule discovery* finds all rules that correlate the presence of one set of items with that of another set of items.^{4,8,9,13} One familiar example is the discovery of items that sell together in a supermarket from mining the sales transactions at the point of sale. A management decision based on such findings could be to shelve these items close to one another.

Classification refers to assigning objects to predefined categories or classes. In a credit-evaluation scenario, classification can categorize applicants' credit ratings as good or poor, for acceptance or rejection.

Sequential pattern discovery determines strong sequential dependencies among different events.^{4,8,9,13} This process has many applications, from medical diagnosis to sales-transactions analysis to determine which customers are likely to buy a specific product in the near term.

Another application of data mining is the *detection of outliers* (or deviations).

Outlier detection finds data points that differ significantly from the majority of the data points in a given data set. Medical diagnosis and credit card detection are examples of outlier detection.

High-performance data mining, thus, refers to developing algorithms for data-mining techniques that lend themselves to parallelism and can map efficiently onto parallel computers.^{5,6,9} Such algorithms should consider the underlying architecture—such as distributed shared memory or a dedicated cluster of workstations—and the standard issues and challenges in parallel processing such as data locality, load balancing, interprocessor communication, and parallel I/O. Parallel I/O becomes very important in this case, because massive data analysis will require substantial data access to secondary storage and tape libraries. Issues of data organization,

data prefetching, and latency hiding by overlapping the mining tasks with the I/O will soon become even more critical, and many new research challenges and opportunities will materialize.

Meeting the goal

Our track on data mining seeks to increase knowledge about and promote the benefits of using parallel and distributed computing platforms to solve problems in data-mining applications. This and subsequent issues of *IEEE Concurrency* will feature a range of cross-disciplinary articles that explore the link between parallel and distributed computing and data mining. These articles will deal with

- algorithms and techniques (machine learning, statistical techniques, heuristics, multicriteria evaluation, data-mining benchmarks, and algorithm complexity);
- data and knowledge representation;
- text processing (information retrieval);
- data and dimensionality reduction;
- reprocessing and postprocessing;
- decomposition of large datasets;
- data visualization;

- interoperability and heterogeneous systems;
- advanced transaction models and management;
- performance and benchmarking;
- query processing;
- parallel and distributed databases;
- active and temporal databases; and
- legal, ethical, and social issues (privacy and commercialization).

We launch the track with two articles. In "Parallel and Distributed Association Mining: A Survey," Mohammed Zaki surveys parallel and distributed association rule mining algorithms. Zaki also highlights the open problems in this area, such as dealing with large databases, load balancing, and data skews.

In "Strategies for Parallel Data Mining," David Skillicorn argues that benchmarking and implementations are very expensive approaches for comparing parallel data-mining algorithms. He proposes a number of cost-effective alternative measures (counting computations, data accesses, and communication). These measures can provide a reasonably accurate picture of an application's performance. //

References

1. M.J.A. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*, John Wiley & Sons, New York, 1997.
2. K. Rastogi and K. Shim, "Tutorial on Scalable Algorithms for Mining Large Databases," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery & Data Mining*, ACM Press, New York, 1999; www.bell-labs.com/project/serendip/Talks/tutorial.ps.gz (current Nov. 1999).
3. C. Glymour et al., "Statistical Inference and Data Mining," *Comm. ACM*, Vol. 39, No. 11, Nov. 1996, pp. 35-41.
4. U. Fayyad and R. Uthurusamy, "Special Section: Data Mining and Knowledge Discovery in Databases: Introduction," *Comm. ACM*, Vol. 39, No. 11, Nov. 1996, pp. 24-26.

5. V. Kumar and M. Joshi, *Tutorial on High Performance Data Mining*, Dept. of Computer Science, Univ. of Minnesota, 1999; www-users.cs.umn.edu/~mjoshi/hpdmtdt/index.htm (current Nov. 1999).
6. M.J. Zaki, C. Ho, and R. Agrawal, "Parallel Classification on SMP Systems," *Proc. First Workshop on High-Performance Data Mining-Int'l Parallel Processing Symposium*, 1998.
7. K. Alsabti, S. Ranka, and V. Singh, "An Efficient K-means Clustering Algorithm," *Proc. First Workshop on High-Performance Data Mining*, 1998.
8. M.J. Zaki et al., "New Algorithms for Fast Discovery of Association Rules," *Third Int'l Conf. Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., 1997.
9. M.J. Zaki et al., "Parallel Data Mining for Association Rules on Shared-Memory Multiprocessors," *Proc. Supercomputing '96 (CD-ROM)*, IEEE Computer Society Press, Los Alamitos, Calif., 1996.
10. R. Kimball, *The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses*, John Wiley & Sons, New York, 1996.
11. C. Westphal and T. Blaxton, *Data Mining Solutions: Methods and Tools for Solving Real-World Problems*, John Wiley & Sons, New York, 1998.
12. S. Anahory and D. Murray, *Data Warehousing in the Real World: A Practical Guide for Building Decision Support Systems*, Addison-Wesley, Reading, Mass., 1997.
13. C.E. Brodley, T. Lane, and T. M. Stough, "Knowledge Discovery and Data Mining," *American Scientist*, Vol. 87, No. 1, Jan./Feb. 1999.

Albert Y. Zomaya is a professor in, and the associate head of, the Department of Electrical and Electronic Engineering at the University of Western Australia, where he also leads the Parallel Computing Research Laboratory. His research interests include parallel algorithms, scheduling, computational machine learning, scientific computing, adaptive computing systems, mobile computing, and data mining. He received his PhD in control engineering from Sheffield University, UK. He is an associate editor for the *IEEE Transactions Parallel and Distributed Systems*, *IEEE Transactions Systems, Man, and Cybernetics (parts A, B, and C)*, *Journal of Parallel Algorithms and Applications*, *Journal of Interconnection Networks*, the *International Journal of Parallel and Distributed Systems and Networks*, the *Future Generation Computer Systems Journal*, and the *International Journal of Foundations of Computer Science*. He is the founding editor of the *Wiley Book Series on Parallel and Distributed Com-*

puting. He is a board member of the International Federation of Automatic Control (IFAC) committee on Algorithms and Architectures for Real-Time Control and serves on the executive boards of the IEEE Technical Committee on Parallel Processing and the IEEE Task Force on Cluster Computing. He is the founding cochair of the Workshop on Biologically Inspired Solutions to Parallel Processing Problems. He is the stream chair of the 10th International Conference on Computing and Information (Kuwait, 2000) and program vice-chair of the International Conference on Parallel Processing (Toronto, 2000). He is a chartered engineer and a senior member of the IEEE and member of the ACM, Institute of Electrical Engineers (UK), and Sigma Xi. He received the 1997 Edgeworth David Medal from the Royal Society of New South Wales for outstanding contributions to Australian Science. Contact him at Electrical & Electronic Eng., Univ. of Western Australia, Nedlands, WA 6907, Australia; zomaya@ee.uwa.edu.au; www.ee.uwa.edu.au/~zomaya.

Tarek El-Ghazawi holds a joint appointment as associate professor in the Institute for Computational Sciences and Informatics and the Department of Electrical and Computer Engineering at George Mason University. His research interests include high-performance computing, experimental computer architecture, high-performance I/O systems, experimental performance evaluations, and computer vision. He received a BS in electronics and communications engineering from the University of Helwan, Egypt, and his MS and PhD in electrical and computer engineering, both from New Mexico State University. He is a senior member of the IEEE and a member of the ACM and Phi Kappa Phi. Contact him at the Inst. for Computational Sciences and Informatics, MS 5C3, George Mason Univ., 4400 University Dr., Fairfax, VA 22030-4444; tarek@gmu.edu; 129.174.45.49/~tarek.

Ophir Frieder is the Illinois Institute of Technology Research Institute Chaired Professor of Computer Science. He received the International Information Science Foundation Award from Japan and the NSF National Young Investigator Award. Recently, he received the Excellence through Teamwork Award for his contributions at Harris Information Systems Division and both the College of Engineering and the University-Wide Research Excellence Awards at Florida Tech. He is on the editorial boards of the *Journal of the American Society for Information Science* and the *Journal of Information Retrieval*. He is the author of *Information Retrieval: Algorithms and Heuristics* (Kluwer, 1998). He is a member of the ACM and Phi Beta Kappa and is a senior member of the IEEE. Contact him at Illinois Inst. of Technology, Dept. of Computer Science, 10 W. 31st St., Chicago, IL 60616; ophir@cs.iit.edu; www.cs.iit.edu/~ophir.