Data Mining at CALD-CMU: Tools, Experiences and Research Directions

*C. Faloutsos, G. Gibson, T. Mitchell, A. Moore, S. Thrun**
Center for Automated Learning and Discovery (CALD)
Carnegie Mellon University

Abstract

We describe the data mining problems and solutions that we have encountered in the *Center for Automated Learning and Discovery (CALD)* at CMU. Specifically, we describe these settings and their operational characteristics, describe our proposed solutions, list the performance results, and finally outline future research directions.

1 Introduction

The Center for Automated Learning and Discovery (CALD) is a cross-disciplinary center at CMU, focusing on the research question "How can historical data be best used to improve future decisions?" Participants in CALD are drawn from diverse backgrounds, such as Computer Science (and specifically, Artificial Intelligence, Databases, Theory), Robotics, Statistics, Neurology, Philosophy, Engineering (Electrical, Civil, and Mechanical), Information Retrieval, and Language Processing. The Center involves industrial partners with challenging data-mining problems.

In this paper we describe some of these settings, recent research progress in our center, and finally list future research directions.

2 Some Real Applications

Next we describe two settings that we have encountered so far, and which seem typical in data mining environments:

- The AUTON project (`www.cs.cmu.edu/~AUTON`) is concerned with applying a combination of statistical machine learning techniques with optimal control theory to produce very autonomous controllers for complex manufacturing processes. This software

*In alphabetical order. E-addresses: {`christos,mitchell+,garth,awm,thrun+`}`@cs.cmu.edu`

system (using algorithms described in [2, 3]) has been deployed in a number of processes in a food manufacturing industry (bagging, packaging, cooling and cooking) with substantial economic savings. Current work is extending AUTON to give the learning controller an active learning capability in which it autonomously designs its own conservative experiments.

- AT&T sales dataset: the need is to store a multi Giga-byte matrix on-line, with customers for rows, days for columns, and amount spent in each cell of the matrix. In more detail, the specifications are as follows: The number of rows is very large: $\geq 10^5$; the number of columns is much smaller ($\approx 10^3$ columns); random access to each cell is essential. On the other hand, approximate answers are acceptable; Thus, what we want is a lossy compression method, carefully designed to allow fast (and as accurate as possible) reconstruction of any cell or cells. In the next section we describe our solution, based on a powerful technique from statistics and matrix algebra, the Singular Value Decomposition (SVD).

3 Recent Research Advances in CALD

This section describes several recent research developments within CALD. Specifically, the AD-tree method for fast manipulation of contingency tables; a probabilistic reasoning approach and its performance on a robotics setting; and a lossy compression method for large data matrices, which supports random access to arbitrary cells, with small error and high compression ratio. Each topic is discussed in detail below.

3.1 AD-tree

Problem Many data mining and machine learning algorithms need to do vast numbers of counting queries (e.g. "How many records have Color=Blue, Nationality=British, Smoker=False, Status=Married, Nose=Big?"). Similarly, many algorithms need to build and test huge numbers of contingency tables. A *contingency table* (also known as a "DataCube" [18] in the Database community) is defined by a set of attributes. A contingency table has one row for each possible set of values that the set of attributes may take. If an attribute called "Color" could take values { Red , Green , Blue } and if an attribute called "Smoker" could take values { Yes , No } then the contingency table for attribute-set { Color , Smoker } would be

Color	Smoker	Number Records Matching
Red	No	n_1
Red	Yes	n_2
Green	No	n_3
Green	Yes	n_4
Blue	No	n_5
Blue	Yes	n_6

where $\sum_i n_i =$ the total number of records.

Why do we wish to compute counts and contingency tables quickly? There are many applications in data mining. A database user may wish to bring up counts or contingency tables on-line while analyzing the dataset [10]. Interactive visualization tools similarly need to compute these statistics quickly. More importantly, many machine learning, statistics and data mining algorithms (e.g. Bayes Net builders, Feature Selectors, Rule Learners, Inductive Logic Program Learners, Decision Tree Learners) spend most of their effort on counting computations.

Proposed solution - Results The ADtree approach empirically gives us a *two to four order of magnitude speedup (40 to 2000-fold)* in doing counting. Analytically, we can show that, subject to certain assumptions, the costs become independent of the number of records and loglinear in the number of non-zero entries in the contingency table. The ADTree (all-dimensions tree) caches sufficient information to reconstruct any counting query. Tractably-sized data structures can be produced for large real-world datasets by (a) using a sparse tree structure that never allocates memory for counts of zero, (b) never allocating memory for counts that can be deduced from other counts, and (c) not bothering to expand the tree fully near its leaves.

In work so far we have shown how the ADTree can be used to accelerate Bayes net structure finding algorithms, rule learning algorithms, and feature selection algorithms on a number of large real-world datasets. For example, for arbitrary counting queries involving eight attribute-value pair on a medical database with 10,000 records and 100 attributes the average speed-up over direct counting was approximately 1000-fold. When building a Bayes Net (and learning rules) for a Census Dataset involving 17 very non-sparse attributes, the speedup was 50-fold.

Further results are given in [14], which also compares AD-trees with alternative representations such as kd-trees [5, 13], R-trees [9] and frequent sets [12]. In current work (funded by NSF and 3M corporation) we are using AD-trees to permit tractable feature-generation algorithms (which invent new attributes useful for prediction as complex functions of the original attributes). We are also actively seeking collaborations with people with large finance, medicine or manufacturing datasets to which we may attempt to apply ADtree-based learning. Further AD-tree information may be found (shortly) at `www.cs.cmu.edu/~AUTON`.

3.2 Probabilistic Reasoning

Problem Another CALD project focuses on state estimation and decision making in sensor-actuator systems. Systems equipped with sensors (such as robots) are inherently uncertain as to what is the case in the world. This uncertainty usually arises from perceptual limitations of sensors and from the dynamics of the world. Our research seeks fundamental ways to deal with this uncertainty and to make optimal decisions under uncertainty.

Proposed solution We recently have developed a family of probabilistic approaches for perception and decision making, which has been applied to a variety of difficult robotics problems. The key idea of these methods is that the system reasons probabilistically: Instead of just considering a single interpretation of what might be the case in the world, these methods consider an entire collection of interpretations, annotated by a numerical plausibility factor (a conditional probability As a result, these methods can represent uncertainty, and they provide robust and mathematically elegant ways for dealing with ambiguities, sensor noise, and dynamics.

Results These algorithms were applied to problems such as mobile robot localization, landmark detection and recognition, mapping of large-scale environments, and others [15]. In some of these domains, the probabilistic approach led to completely new insights, that made possible solutions for previously unsolved robotics problems. For example, our probabilistic algorithms has been demonstrated to enable robots to build maps of unprecedentedly large environments [17, 16]. Other algorithms were essential for a recent installation of a mobile robot in the "Deutsches Museum Bonn". Here, a mobile robot gave interactive tours to people in a densely populated museum. The robot navigated almost flawlessly at a *total distance of 18.6km* and at an *average speed of 36cm/sec*, entertaining more than 3,000 visitors (real visitors and Web users) [4]. The probabilistic algorithms were critical for position tracking and model acquisition.

3.3 Lossy Compression for Data Mining

Problem Ad hoc querying is difficult on very large datasets, since it is usually not possible to have the entire dataset on disk. While compression can be used to decrease the size of the dataset, compressed data is notoriously difficult to index or access.

We consider a very large dataset comprising multiple distinct time sequences. Our driving application was the AT&T sales dataset, described earlier. Each point in the sequence is a numerical value. Our goal is to compress such a dataset into a format that supports ad hoc querying, provided that a small error can be tolerated when the data is uncompressed.

Proposed method The idea behind our method [11] is to use the so-called Singular Value Decomposition (SVD) to approximate the data matrix; we went further to reduce the approximation error, by explicitly storing those data points that were 'outliers'. The resulting method, called 'SVDD' (for "SVD with Deltas") achieves all the specified goals.

Results Experiments on large, real world datasets (AT&T customer calling patterns) show that the proposed method achieves an average of *less than 5% error* in any data value after compressing to a *mere 2.5% of the original space* (i.e., a 40:1 compression ratio), with these numbers not very sensitive to dataset size. Experiments on aggregate queries achieved a *0.5% reconstruction error with under 2% space requirement.*

4 Future Directions

The primary goal of CALD research is to extend the state of the art in using historical data to improve future decisions. Our role is to invent new approaches that will become the basis for future commercial software. We will develop these new approaches by studying problems and data contributed by our industrial and government partners, and will make our results immediately available to CALD corporate members and partners. Thus, our partners will have access to new methods long before they become commercially available.

CALD research can be viewed either in terms of basic scientific issues to be addressed, or in terms of applications, as illustrated in Figure 1. The exact list of applications is being determined by the needs of our industrial and government partners. The list of basic research topics will be determined primarily by the needs of these application problems, and by faculty research interests and expertise.

The thesis underlying the CALD matrix research organization in Figure 1 is that the most important scientific issues will have significant impact across many different application areas. This allows CALD to spread the cost of this basic research over multiple problem domains and multiple funding sources. Examples of such basic scientific issues include:

- *Learning from mixed media data.* In many application domains, historical data will include a variety of types of media. For example, when learning to predict medical outcomes based on historical data, patient records often include a combination of symbolic data (e.g., gender), numerical data (e.g., temperature), images (e.g., x-rays and CAT scans), other sensor data (e.g., EKG), text (e.g., notes on the patient chart), and audio (dictations of physicians as they make hospital rounds). Unfortunately, current learning methods can make use of only a fraction of this patient record, because

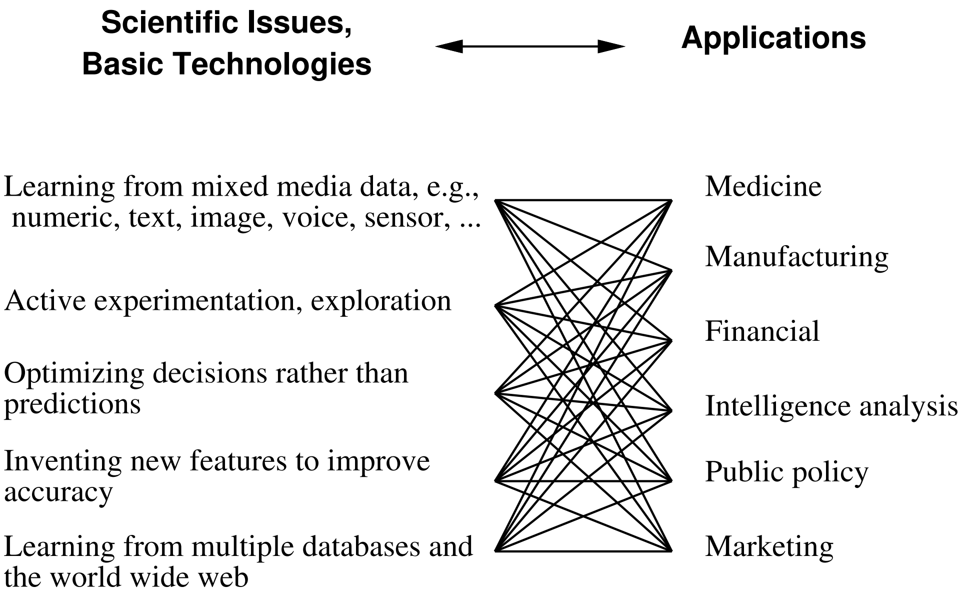

Figure 1: CALD research emphasizes fundamental scientific issues (left) with significant potential payoff across multiple application areas (right).

they are typically specific to a single type of media (e.g., decision tree learning methods for symbolic/numeric data, neural networks for image analysis, Bayesian methods for text classification). We need new learning methods capable of learning from the full range of data available in the historical record. If successful, this line of research will produce more accurate learning methods useful in a variety of applications, due to their ability to utilize the entire multiple-media historical record.

- *Active experimentation.* Most current data mining systems passively accept a predetermined data set. We need new computer methods that actively generate optimal experiments to obtain the additional needed information. For example, in modeling a manufacturing process it is relatively easy to capture data while the process runs under normal conditions. However, this data may lack information about how the process will perform under non-standard conditions. We require algorithms that will propose optimal experiments to collect the most informative data, taking into account precise models of the expected costs and benefits of the experiment.
- *Optimizing decisions rather than predictions.* The goal here is to use historical data to improve the choice of actions in addition to the more usual goal of predicting outcomes. For example, consider the problem of customer retention. Given historical data on customer purchases over time, one common data mining problem is to predict which customers are likely to remain loyal, and which are not. While this is useful, an even more useful task is to learn which actions can increase the probability of retaining these

customers. The point here is that we seek new algorithms that go beyond predicting the outcome of some time series, and instead learn which actions achieve the desired outcome. This problem raises difficult basic issues such as learning from biased data samples, and how to incorporate conjectures by human experts about the effectiveness of various intervention actions. If successful, this research will allow applying historical data much more directly to the decision-making problem at hand.

- *Inventing new features to improve prediction accuracy.* In many cases, the accuracy of predictions can be improved by inventing a more appropriate set of features to describe the available data. For example, consider the problem of detecting the imminent failure of a piece of equipment based on the time series of sensor data collected from the equipment. It is easy to generate millions of features that describe this time series by taking differences, sums, ratios, averages, etc of primitive sensor readings and previously defined features. Our conjecture is that given a sufficiently large and long-duration data set it should be feasible to automatically explore this large space of possible defined features in order to identify the small fraction of these features most useful for future learning. If successful, this work would lead to increased accuracy in many prediction problems, such as predicting equipment failure, customer attrition, credit repayment, medical outcomes, etc.
- *Storage issues* - RAID, striping, and beyond. Several data mining algorithms are very suitable for intelligent storage devices, such as those advocated by CMU's *Network Attached Secure Disks (NASD)* project [6, 7, 8]. A NASD device can do some computationally simple, but data-intensive processing, reducing the amount of data to be sent to the CPUs - this is appropriate for mining association rules [1] as well as for the training of neural networks. Specifically, there are two principle benefits from execution in intelligent storage devices:
  - Bandwidth reduction: Disk drives sustain 15 MB/s now and this data rate is growing at 40% per year. Network interfaces and client machines cannot cost-effectively move and consume multiple drives' bandwidth and have processing resources left over to filter that data, where simple filters reduce transfered data size by a factor of ten or more. Thus, device-embedded processing enables faster, more scalable data mining.
  - Inner loop computational parallelism: For computationally simple, data-intensive inner loops (where general purpose processors gain little from caches), execution in the drive provides computational parallelism in proportion to data capacity - i.e., 100 drives with 100 MHz processor optimized for data streaming is likely to be better for simple inner loops than 4 CPUs at 500 MHz.

CALD faculty research interests include many additional topics as well, such as problems in automatic data capture, visualization of large data sets, learning across multiple databases, protecting data privacy while utilizing historical data, methods for information filtering, and using the Internet as a large public data source to augment application-specific data sets.

5 Conclusions

In this paper we have provided a list of some typical settings, some recent research achievements and their performance benefits, and finally we outlined future research directions.

In our opinion, the most significant point is the need for *cross-disciplinary* collaboration: AD-trees are motivated by Bayesian Networks, but use or compete against database methods (R-trees etc); SVDD uses statistical methods to solve large-scale database problems; probabilistic reasoning led to successful robot navigation; advances is storage technology ("Network Attached Secure Devices") seem especially beneficial for data mining tasks. Further cross-fertilization of ideas and tools among the above areas, as well as Information Retrieval, Language Processing, Robotics and Neuro-physiology seems very promising, and probably necessary, to push the state of the art in Data Mining.

Acknowledgements:

We would like to thank the remaining members of CALD for their feedback:

Avrim Blum (CS); Jaime Carbonell (CS); Howie Choset (Mech. Eng.); Greg Cooper (Medical Inf., U.Pitt.); Mark Derthick (Robotics); Bill Eddy (Statistics); Scott Fahlman (CS); Stephen Fienberg (Statistics); Alan Frieze (Math); Jim Garrett (Civil Eng.); Christopher Genovese (Statistics); Alex Hauptmann (CS); Brian Junker (Statistics); Jay Kadane (Statistics); Ravi Kannan (CS); Robert Kass (Statistics); John Lafferty (CS); Tai Sing Lee (CNBC); John Lehoczky (Statistics); Marsha C Lovett (CIL); Roy Maxion (CS); James McClelland (Psych); Mike Meyer (Statistics); Roni Rosenfeld (CS); Steve Roth (Robotics); Richard Scheines (Phil); Mark Schervish (Statistics); Teddy Seidenfeld (Statistics); Reid Simmons (CS); Peter Spirtes (Phil); Kannan Srinivasan (GSIA); Sarosh Talukdar (Elec. Comp. Eng.); Raul Valdes-Perez (CS); Manuela Veloso (CS); Isabella Verdinelli (Statistics); Pantelis Vlachos (Statistics); Alex Waibel (CS); Larry Wasserman (Statistics); Joel Welling (PSC); Yiming Yang (LTI).

References

[1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD*, pages 207–216, May 1993.

[2] C. G. Atkeson, A. W. Moore, and S. A. Schaal. Locally Weighted Learning. *AI Review*, 11:11–73, April 1997.

[3] C. G. Atkeson, A. W. Moore, and S. A. Schaal. Locally Weighted Learning for Control. *AI Review*, 11:75–113, April 1997.

[4] W. Burgard, D. Fox, G. Lakemeyer, D. Hähnel, D. Schulz, W. Steiner, S. Thrun, and A.B. Cremers. Real robots for the real world — the RHINO museum tour-guide project. (submitted for publication), October 1997.

[5] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Trans. on Mathematical Software*, 3(3):209–226, September 1977.

[6] Garth A. Gibson, David F. Nagle, Khalil Amiri, Fay W. Chang, Eugene M. Feinberg, Howard Gobioff, Chen Lee, Berend Ozceri, Erik Riedel, David Rochberg, and Jim Zelenka. File server scaling with network-attached secure disks. *ACM International Conference on Measurement and Modeling of Computer Systems (Sigmetrics'97)*, June 1997.

[7] Garth A. Gibson, J.S. Vitter, and J. Wilkes. Working group on storage i/o issues in large-scale computing. *Computing Surveys*, 28(4), December 1996.

[8] Garth A. Gibson and J. Wilkes. Self-managing network-attached storage, strategic directions in computing research: Working group on storage i/o issues in large-scale computing. *Computing Surveys*, 28A (online)(4), December 1996.

[9] A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proceedings of SIGMOD 84*, 1984.

[10] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. Implementing Data Cubes Efficiently. In *Proc. ACM SIGMOD*, pages 205–216, May 1996.

[11] Flip Korn, H.V. Jagadish, and Christos Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. *ACM SIGMOD*, pages 289–300, May 1997.

[12] Heikki Mannila and Hannu Toivonen. Multiple uses of frequent sets and condensed representations. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[13] A. W. Moore, J. Schneider, and K. Deng. Efficient Locally Weighted Polynomial Regression Predictions. In *Proceedings of the 1997 International Machine Learning Conference.* Morgan Kaufmann, 1997.

[14] Andrew W. Moore and Mary Soon Lee. Cached Sufficient Statistics for Efficient Machine Learning with Large Datasets. TR CMU-RI-TR-97-27: See also www.cs.cmu.edu/~awm/rl-papers/cache.ps, CMU Robotics Institute, July 1997.

[15] S. Thrun. A bayesian approach to landmark discovery in mobile robot navigation. *Machine Learning*, to appear.

[16] S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, to appear.

[17] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. A probabilistic approach for concurrent map acquisition and localization. Technical Report CMU-CS-97-183, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, October 1997.

[18] J.D. Ullman. *Database and Knowledge Base Systems*. Computer Science Press, 1988.

Contents