Pages 3 and 7 are meant to be seen in color.

MineSet: An Integrated System for Data Mining

Cliff Brunk James Kelly Ron Kohavi
Data Mining and Visualization
Silicon Graphics, Inc.
2011 N. Shoreline Blvd
Mountain View, CA 94043-1389
{brunk,jkelly,ronnyk}@engr.sgi.com

May 1997

Abstract

We describe MineSet, Silicon Graphics' interactive system for data mining. MineSet integrates three powerful technologies: database access, analytical data mining, and data visualization. It supports the knowledge discovery process from data access and preparation through iterative analysis and visualization to deployment. MineSet provides highly integrated facilities for accessing data from commercial databases or files, mining that data, and visualizing both the data and the models generated from the data by the mining algorithms. MineSet is based on a client-server architecture that scales to large databases. The database access component provides a rich set of operators that can be used to preprocess and transform the stored data into forms appropriate for visualization and analytical mining. The 3D visualization capabilities allow direct data visualization for exploratory analysis, including tools for displaying high-dimensional data taking advantage of geographical and hierarchical information. The analytical mining algorithms help the user identify potentially interesting models of the data, which can be viewed using visualization tools specialized for the learned models. Third party vendors can interface to the MineSet tools for model deployment or for integration with other packages.

1 Introduction

Recently there has been a significant increase in the amount of data stored online. For instance, SpaceCrunch (Taylor & Schwartz 1996), the worlds largest chemical database, stores information on over 150 billion compounds; the 2nd Palomar Observatory Sky Survey which generated 3 terabytes of image data containing an estimated 2 billion sky objects (Fayyad, Haussler & Stolorz 1996). Similar examples arise in business where retailers track inventory, telecommunications companies track phone calls, and credit card companies track transactions. The creation of these large databases has brought about a need for tools that can help identify, distill, summarize, and explain the knowledge contained therein.

We describe MineSet 2.0, an interactive tool for exploring large databases. MineSet integrates three powerful technologies: database access, analytical data mining, and data visualization. MineSet provides highly integrated facilities for accessing data from commercial databases or files, mining that data, and visualizing both the data and the models generated from the data by the mining algorithms. The database access component provides a rich set of operators that can be used to preprocess and transform the stored data into a form appropriate for mining. The mining algorithms help the user identify potentially interesting models of the data, while the visualization tools display the data in various ways and help the user better understand the learned models.

The goal of MineSet development is two fold. First, MineSet provides a state-of-the-art general purpose stand-alone data access, analysis, and visualization tool. Second, MineSet provides the technological foundation needed for third party developers to create cutting-edge domain specific applications requiring this technology. The current development direction is to extend the set of analytical and visual data mining techniques while simultaneously increasing the level of integration between these techniques.

This paper provides an overview of MineSet and illustrates how it integrates the several technologies to provide an interactive tool that can help in the knowledge discovery process (Fayyad, Piatetsky-Shapiro & Smyth 1996*b*). We begin by describing the MineSet system architecture. Then we focus on MineSet's visual mining tools and the analytical data mining engine. Next we describe how the visualization techniques are coupled with the analytical techniques to visualize models of the data. We conclude the paper with a discussion of the other facilities in MineSet designed specifically to help manage the knowledge discovery process and a brief description of future directions.

2 System Architecture

MineSet employs a three tiered architecture (Figure 1). The first tier is the client, which includes Tool Manager and the visualization tools. Tool Manager is the graphical interface through which the user interacts with MineSet. The visualization tools are used to display data and models of data generated by the mining algorithms. After invoking a visual tool with Tool Manager, the user can interact directly with that tool and send information from it to other tools via Tool Manager. The second tier is the server, which includes Data Mover and the analytical mining engines. Data Mover is the database access and data transformation component of MineSet. It extracts data from the source, transforms it, and orchestrates

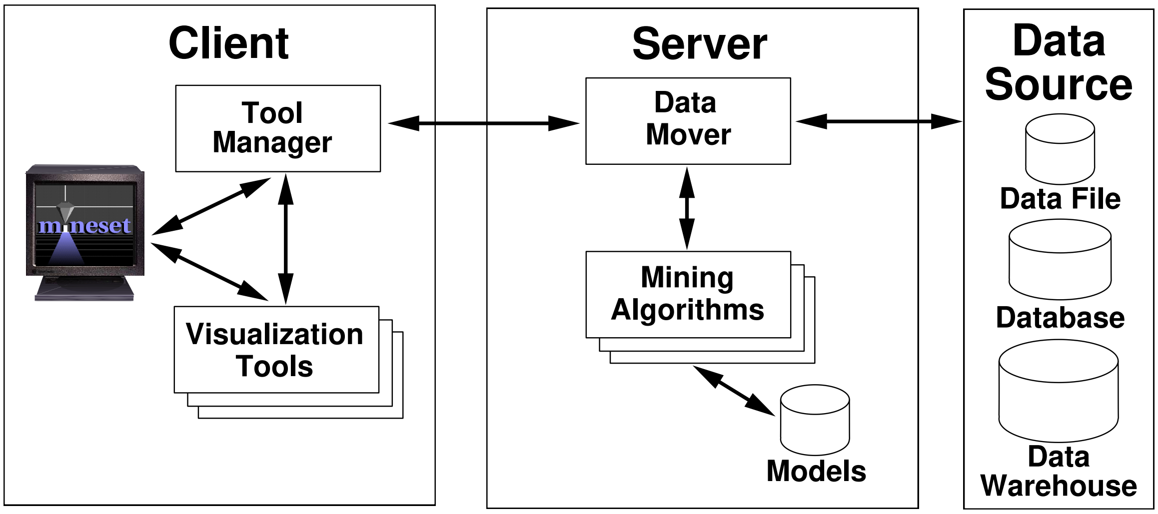

Figure 1: MineSet Architecture

moving it from one MineSet component to another. The mining tools are used to generate models of the transformed data, which can be applied to new data or visualized. The third tier is the data source, which includes the storage subsystem that maintains the users data. It can be either a file or a commercial database such as Oracle, Sybase, or Informix. The tiers are not tied to specific machines: all three can reside on a single hardware platform or three separate machines.

This architecture provides the flexibility needed to scale with the size of the problem. Mining large databases requires powerful hardware: a substantial amount of memory, a number of fast CPUs, and high I/O bandwidth. Desktop machines are not powerful enough to support mining data beyond tens of megabytes. The MineSet architecture allows mining to be done on a powerful server machine, while the client can reside on a smaller workstation. When working with small databases or during pilot projects, the architecture has the flexibility to allow the client, server and data source all to reside on a single machine. Any OpenGL/X-windows enabled workstation or PC can be used to interact with MineSet.

3 Visual Mining

MineSet provides a rich set of visualization tools that enable users to interactively explore data and quickly discover new patterns, trends, and relationships. The human perception system can identify anomalies and patterns much faster in a representative landscape than in a spreadsheet. The tools utilize 2D and 3D landscapes that take advantage of a person's ability to navigate in space, track movement and compare objects of different sizes, colors, and shapes. In addition to visualization and navigation, the tools contain sophisticated filtering and search facilities that allow users to quickly reduce the landscape to items of interest.

The Statistics Visualizer displays basic statistics in histograms and boxplots. The Scatter Visualizer (Figure 2 left) displays scatterplots with up to eight dimensions: three axes,

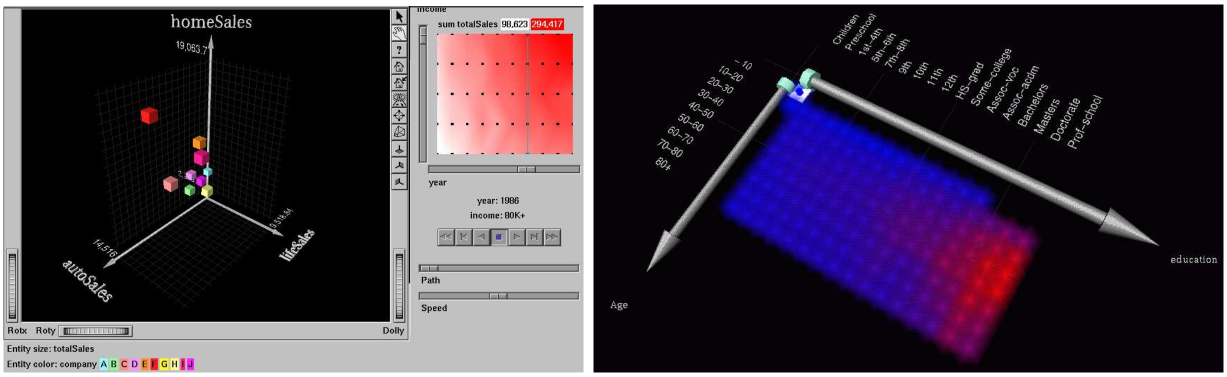

Figure 2: Left: A Scatterplot for 10 insurance companies and their policy sales. The cube size represents the total revenue. The upper right portion shows a 2D density plot for the two independent attributes for data slicing. The center right portion shows the VCR controls for animations. Right: A Splat plot of age bracket versus education in a census file, with the color indicating income (blue is low, red is high). Splat plots perform Gaussian smoothing and are designed for showing many records (about 150,000 in this plot).

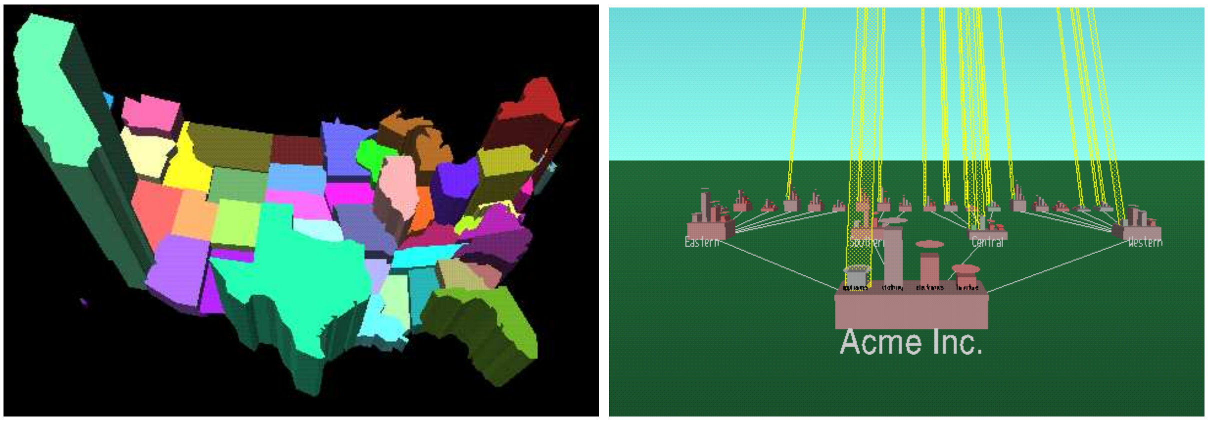

Figure 3: Left: A map of the US with height representing the US population in 1990. Right: Total revenue (bars) and targets (disks) for four major categories for a store; the hierarchy splits on major geography, then states, and finally cities; spotlights point to bars satisfying a user-defined condition.

entity color, entity size, entity rotation, and two independent attributes shown through animation (described below). The Splat Visualizer extends the scatterplots when there are more than tens of thousands of records; it blurs the points using Gaussian smoothing (Figure 2 right). The Map Visualizer (Figure 3 left) displays data with a spatial component; a polygonal map must be provided and two attributes can be mapped to the polygon's height and color; multiple maps may be linked together to show different attributes for the same spatial geography. The Tree Visualizer (Figure 3 right) displays data hierarchically; users can determine the hierarchy and map attributes to a histogram at each node. Additional capabilities shared by most visual tools include:

Attribute mapping Many attributes in the data can be mapped to graphical attributes in the scene. For example, attributes can be mapped to entity color, height, and shape.

Scene manipulation Users can manipulate the scene using thumb-wheels and mouse movements. The standard rotate, translate, and zoom operations are provided. In addition, users can select entities in the scene to see their values.

Data Slicing Users can choose two independent attributes for slicing the data. Sliders for these attributes are shown near the main window (Figure 2 left). Users can move the sliders thus changing the scene to reflect the slice of data that matches the sliders' position. For example, a scatterplot showing three axes for channel, geography, and revenue can be augmented with two more axes, say month of year and buyer's income bracket as independent attributes. When points for the independent attributes are selected, the scatterplot changes.

Animation capabilities Users can select a path in the two-dimensional panel for the data slicing attributes. VCR controls shown in Figure 2 (left) allow animating between the different slices. MineSet provides interpolation facilities to make the animation smooth, thus allowing users to see emerging patterns and trends.

Drill down The entities in the scene can be selected to obtain more detailed information. For example, pointing to a state in a map selects all the records with that state. Selecting a cube in a scatterplot selects all records that contributed to that cube (*e.g.*, revenue for the channel, product group, income bracket, and time). The selected records can then be expanded or transferred to another tool (either visual or analytical).

Search/Filter Users can specify search criteria. Entities satisfying the criteria are filtered or highlighted as shown in Figure 3 (right).

These capabilities allow users to visualize large high-dimensional databases without the information overload usually associated with such tasks.

In this section we described the visual tools that can be used independently of the analytical mining algorithms. The next section describes the analytical mining tools, and Section 5 describes the specialized visual tools that help users understand the results of the those algorithms.

4 Analytical Data Mining Engines

MineSet provides a suite of analytical mining tools based on proven algorithms, such as classifier inducers (Mitchell 1997) and association generators (Srikand & Agrawal 1995). The classifiers include decision trees similar to C4.5 (Quinlan 1993), Simple/Naive-Bayes (Duda & Hart 1973, Domingos & Pazzani 1996) with wrapper-style feature selection (Kohavi & John (to appear), Langley & Sage 1994) and option decision trees kohavi-kunz-option-trees-short,buntine-learning-trees. Holdout and cross-validation can be used to estimate the future prediction accuracy of classifiers. MineSet provides reasonable default settings for all mining algorithms making them easy to use, while allowing experienced users to override the standard defaults.

MineSet's analytical mining components are based on $\mathcal{MLC}$++ (Kohavi, Sommerfield & Dougherty 1996), a stable and robust collection of data mining algorithms written in C++. The $\mathcal{MLC}$++ source code is freely available for research purposes, providing openness and promoting data mining research. Instead of re-implementing existing algorithms, researchers are able to use tested implementations and immediately begin exploring new ideas. In addition to promoting research, having an open mining architecture offers MineSet a strategic advantage because novel research ideas developed within the $\mathcal{MLC}$++ framework can be easily integrated into future releases of MineSet. In addition, this openness allows for easy model deployment, as nothing is hidden about the mining algorithms and nothing is proprietary about the generated models.

In addition to direct application of analytical mining algorithms, MineSet uses the underlying technology from to aid users in other operations of the KDD process. The two most important examples are automatic binning (discretization) and feature selection:

Automatic binning Users frequently discretize the values of an attribute into several bins. While uniform range size and uniform frequency bins are simple options, MineSet also provides automatic binning with respect to another variable (Kohavi & Sahami 1996, Fayyad & Irani 1993). The bins are created based on entropy minimization so that the distribution in each bin is different from other bins. For example, binning age with respect to salary for the UCI census file (Merz & Murphy 1996) yields thresholds of 21, 23, 24, 27, 30, 35, 41, 54, 61, 67. Note how bin ranges are unequal, with four small intervals between the ages 20 and 30, and bigger ranges above 30.

Automatic attribute selection With many datasets containing hundreds of attributes, it is important to aid users in finding the relevant ones for their task. MineSet identifies the important attributes using conditional entropy minimization, generated using oblivious decision trees (Kohavi & Li 1995). Thus, when a user needs to select the axes for a scatterplot in order to understand a given factor, he or she can ask the system to recommend some attributes that are important for explaining a given factor. For example, MineSet chooses relationship (husband, wife, *etc.*), education, and occupation as important factors for determining salary in the census file mentioned above. For determining gender, MineSet chooses occupation, gross-income, and education as important factors.[1].

[1]The relationship and marital-status attributes were removed to make the problem uninteresting. Interestingly, wife implies female only 99.87% of the time in this census file, indicating a minor quality problem.

5 Visualizing Models

Each analytical mining algorithm in MineSet is coupled with a visualization tool that aids users in understanding the learned models. Combining analytical mining tools with visual mining provides added value over that provided by each component alone. The decision tree classifier induced can be displayed using the Tree Visualizer. The simple Bayes model can be displayed and manipulated using the Evidence Visualizer. Similar visualization tool exists for viewing association rules.

The visual tools do not simply display the model; they provide facilities for searching, sorting, filtering, and drilling down, so that complex models can be viewed at several levels of detail. We now describe three examples in detail:

The Evidence Visualizer Figure 4 shows a simple-Bayes model. The prior probabilities are shown in the right pane and the conditional probabilities are shown in the left pane. The real power of the Evidence Visualizer comes from its ability to support "what if" analysis. By selecting pies corresponding to known values (or ranges) for attributes, users can see the affect they have on the predictions made by this classification model. The height of the pies correspond to the number of records, thus showing the reliability of the estimates. The pies representing nominal attributes can be ordered using a variety of criteria including: decreasing evidence level for a given class, record count, or alphabetically.

The Decision Tree Visualizer Figure 5 (left) shows a decision tree. Initially, only the first few levels of the tree are shown. Users can navigate over the decision tree using a fly-through paradigm. As they fly to specific areas of the tree, those subtree are dynamically expanded: nodes are moved apart to make room for more details and more levels are introduced into the scene. While users find the interface easy to navigate, the technology of skewing the space and dynamically expanding nodes is novel. Graphical attributes are used to show information about the decision tree, helping users identify interesting subtrees at lower levels. For instance, the node height is proportional to the number of records allowing users to see how records are distributed. The node base color denotes purity on a scale of red (impure) to green (pure); a green base indicating a strongly skewed distribution may be seen two or three levels down. Users can search for nodes meeting specified criteria, such as nodes with a large proportion of one class and a minimum number of records. Such nodes are visually highlighted, showing where they are in the tree and allowing users to quickly navigate to them.

The Rule Visualizer Figure 5 (right) shows a set of association rules. The prevalence (support), predictability, and expected predictability are mapped to bar color, bar height, and disk height respectively. Users can then filter to subset of rules or ranges of the prevalence and predictability.

6 Managing the KDD Process

In addition to the specific visualization and analytical mining, MineSet supports the broader knowledge discovery process as described in Fayyad, Piatetsky-Shapiro & Smyth (1996*a*,

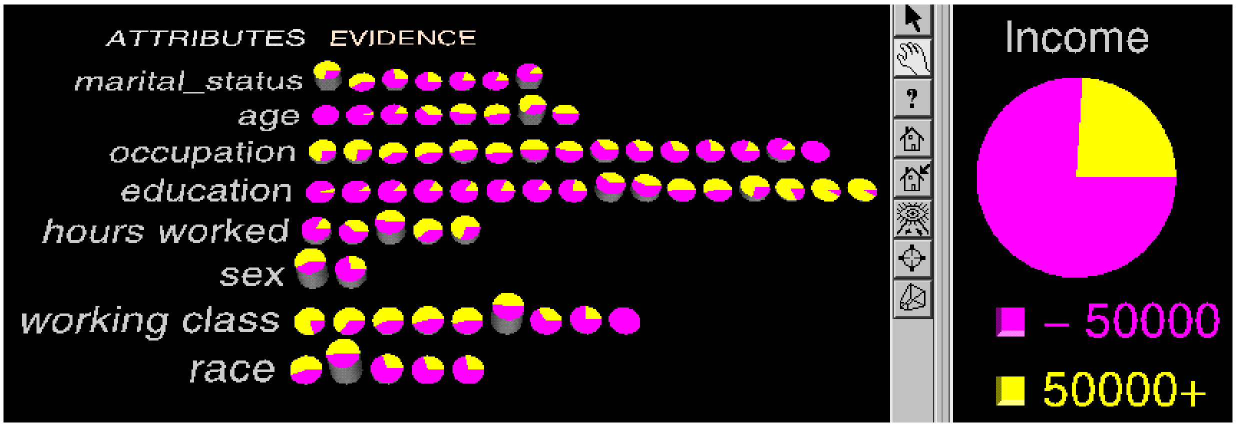

Figure 4: An Evidence Visualizer view for census data on adults working in the US. The pie in the right pane shows the prior probability distribution for adults making over and under \$50,000. The pies in the left pane show each atttribute and the amount of evidence (pie slice size) each value or range adds to each class label. For example, age is an important factor for making high salaries, with evidence slightly decreasing for the last pie (over 61 years old). Education is similarly important with income correlating highly with years of education. The height of each pie represents the number of records for that value or range. The tallest three pies for education correspond to high-school, some-college, and bachelors. Most people in this file are white and work between 38 and 41 hours a week.

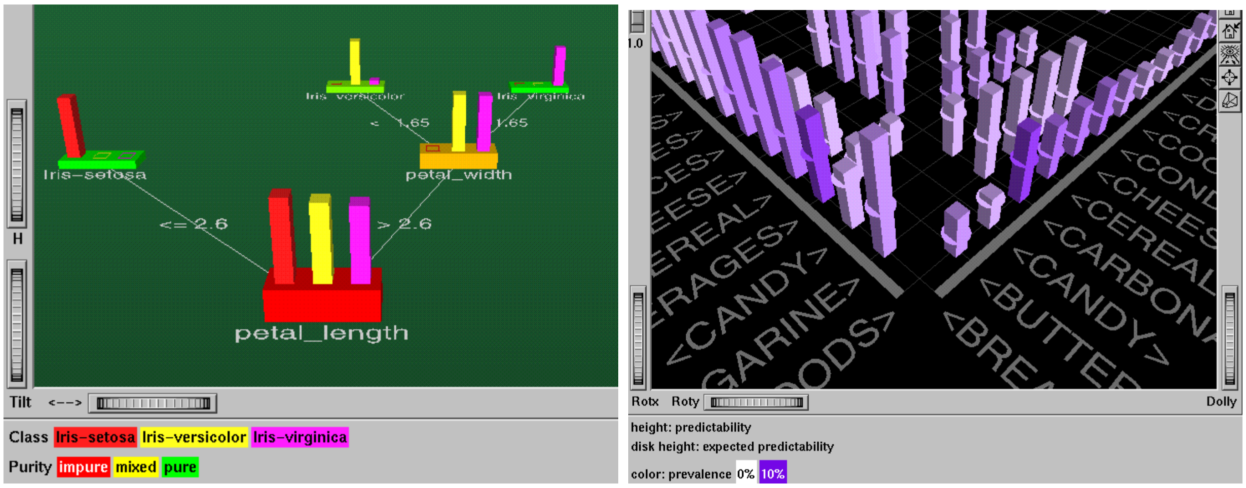

Figure 5: Left: A Tree Visualizer view for the iris dataset; the bars represent the class distribution, the base height represents the number of records, and the base color ranges from red (impure) to green (pure). Right figure: a Rule Visualizer view of supermarket items. The height of each bar represents how well the attribute value on the left predicts the attribute value on the right, *i.e.*, the conditional probability of the right hand side given the left hand side. The disks represent the probability of the right side alone. The colors represent the prevalence, or the frequency of both the left hand side and the right hand side occuring together.

1996*b*). This section focuses on the role that Tool Manager and Data Mover play in helping execute and manage this process.

Tool Manager provides a graphical interface that allows the user to interact with all of the tools in a consistent manner. Selecting data sources, specifying data transformations, setting visualization and mining parameters are all done via a consistent point-and-click interface. Perhaps most importantly, Tool Manager provides a graphic history for managing all the operations performed on the data and allows updating them in the future.

While many data mining systems focus primarily on the analytic tools or visual tools, one of the strengths of MineSet is its extensive data transformation facilities. The key data transformation capabilities in MineSet are binning, aggregation, transposition, and expressions. These are specified in Tool Manager and executed by Data Mover on the server, in accordance with the architectural philosophy that CPU and I/O intensive operations should be done on the server.

Binning translates from raw data values to bins. MineSet provides a rich interface for binning linear attributes types (including dates).

Aggregation enables data to be summarized based on various groups that exist in the data. For example, it may be useful to summarize the data from a transaction database summing purchases, thus giving a total purchase amount for each customer. In addition to summing, other aggregation operations include minimum, maximum, and average.

Transpose creates new attributes from multiple records for the same entity, allowing transaction records to be transformed into customer "signatures" or "profiles" that are more amenable to mining. For example, if the original data contains transactional records with the customer name and product bought, transpose can create one record per customer, containing a new attribute for each possible product bought.

Expressions can be used to define new attributes using standard mathematical expressions and functions. For example, a total billings attribute can be defined as the sum of the billing constituents.

As an example of using MineSet's data transformation capabilities, consider the following simplified scenario based on a real interaction we had with a financial institution trying to understand causes of churn (*i.e.*, why customers leave the institution). The institution's data was stored as the daily transactions each customer made over the last eight quarters and the initial problem was to label each customer as *continuing* or *former* based on the definition that a former customer is one that did not have transactions totaling over $100 in the last quarter. The whole process was done using MineSet as follows: the transaction dates were *binned* by quarter/year; the amounts were then *aggregated* per quarter/year, yielding exactly eight records (one for each quarter) per customer; the data were then *transposed* so that each customer had eight extra attributes corresponding to the transaction volumes in each of the eight quarters. An *expression* was finally defined to label customers as former or continuing based on those eight attributes. This new attribute was then used as the label for the analytical mining algorithms.

In addition to the GUI, MineSet allows submitting mining operations through a command line interface. This capability allows developers to provide users with easy access to predefined operations that can be launched from other applications, including web browsers. MineSet also has a published API, allowing for tighter integration with other applications.

7 Future Directions

MineSet is evolving quickly and because our core competencies cannot cover all areas of data mining, we will interface with other products for those components. For example, there is no neural-network directly implemented in MineSet, nor do we have a database query builder. $\mathcal{MLC}$++ supports many analytical mining tools that we have not yet exposed in MineSet because we believe that either they are not easy to understand due to lack of visualization, or the technology has not yet matured. We believe that it is important to continue the parallel development of visual and analytical tools and tightly couple them. We are currently searching for ways to better display rule classifiers and utilize Bayesian networks.

MineSet uses a client-server architecture. The client side of MineSet can be viewed on any machine running an openGL X-server (*e.g.*, a Windows PC). The server side currently runs only on Silicon Graphics hardware; we are now working on a Windows NT port.

We are exploring ways of utilizing meta knowledge to facilitate model generation and visualization. Utilizing meta knowledge for both filtering and to help analytical algorithms will become more crucial as the field matures; "trivial" rules corresponding to functional dependencies in the real world should be automatically filterered or utilized in the analytical tools.

Silicon Graphics supports several University projects that can improve both the visual and the analytical mining in MineSet.

8 Summary

MineSet provides an integrated analytical and visual data mining environment. It is primarily targeted at business users with an intuitive graphical user interface and good defaults for options, while providing professional data analysts and system integrators the ability to fine tune both the analytical and visualization tools either through the GUI or through the API.

MineSet is built using a scalable client-server architecture. The ability to run the visualizations on a client for good interactive performance while pushing the intensive computations needed for the data transformations and analytical mining to a server machine is crucial for mining large databases.

MineSet supports the knowledge discovery process from data access and preparation through iterative analysis and visualization to deployment. The tight integration between the different components is crucial to effective mining and quick turnaround time for analysis.

Acknowledgments We would like to thank everyone on the MineSet team: Barry Becker, Eric Eros, John Hawkes, Eben Haber, Andy Kar, Peter Rathmann, Gerald Sang'Udi, Dan Sommerfield, Mario Schkolnick, Joel Tesler, as well as Ross Bott who had the insight to initiate this project at the time when SGI was still a hardware-only company. We also wish to thank Eric Bauer, Jay Desouza, Ed Karrels, Alex Kozlov, Clay Kunz, and Tony Lindsey.

References

Domingos, P. & Pazzani, M. (1996), Beyond independence: conditions for the optimality of the simple Bayesian classifier, *in* L. Saitta, ed., 'Machine Learning: Proceedings of the Thirteenth International Conference', Morgan Kaufmann, pp. 105–112.

Duda, R. & Hart, P. (1973), *Pattern Classification and Scene Analysis*, Wiley.

Fayyad, U., Haussler, D. & Stolorz, P. (1996), 'Mining scientific data', *Communications of the ACM* **39**(11), 51–57.

Fayyad, U. M. & Irani, K. B. (1993), Multi-interval discretization of continuous-valued attributes for classification learning, *in* 'Proceedings of the 13th International Joint Conference on Artificial Intelligence', Morgan Kaufmann Publishers, Inc., pp. 1022–1027.

Fayyad, U. M., Piatetsky-Shapiro, G. & Smyth, P. (1996*a*), From data mining to knowledge discovery: An overview, *in* 'Advances in Knowledge Discovery and Data Mining', AAAI Press and the MIT Press, chapter 1, pp. 1–34.

Fayyad, U. M., Piatetsky-Shapiro, G. & Smyth, P. (1996*b*), 'The kdd process for extracting useful knowledge from volumes of data', *Communications of the ACM* **39**(11), 27–34.

Kohavi, R. & John, G. H. ((to appear)), 'Wrappers for feature subset selection', *Artificial Intelligence* .

Kohavi, R. & Li, C.-H. (1995), Oblivious decision trees, graphs, and top-down pruning, *in* C. S. Mellish, ed., 'Proceedings of the 14th International Joint Conference on Artificial Intelligence', Morgan Kaufmann, pp. 1071–1077.

Kohavi, R. & Sahami, M. (1996), Error-based and entropy-based discretization of continuous features, *in* 'Proceedings of the Second International Conference on Knowledge Discovery and Data Mining', pp. 114–119.

Kohavi, R., Sommerfield, D. & Dougherty, J. (1996), Data mining using $\mathcal{MLC}$++: A machine learning library in C++, *in* 'Tools with Artificial Intelligence', IEEE Computer Society Press, pp. 234–245. Received the best paper award.
`http://www.sgi.com/Technology/mlc`.

Langley, P. & Sage, S. (1994), Induction of selective Bayesian classifiers, *in* 'Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann, Seattle, WA, pp. 399–406.

Merz, C. J. & Murphy, P. M. (1996), UCI repository of machine learning databases.
`http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, California.

Srikand, R. & Agrawal, R. (1995), Mining generalized association rules, *in* 'Proceedings of the 21st International Conference on Very Large Databases'.

Taylor, H. & Schwartz, M. (1996), Spacecrunch project. `http://www.tripos.com/spacecrunch/ACS_news.html`.