Appears in KDD-97

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

Abstract

MineSet^{TM}, Silicon Graphics' interactive system for data mining, 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 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 containing geographical and hierarchical information. The analytical mining algorithms help 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 and for integration with other packages.

Introduction

This paper provides an overview of MineSet 2.0 and illustrates how it integrates three foundational technologies to provide an interactive tool that can help in the knowledge discovery process.

For business users, MineSet provides a state-of-the-art general purpose stand-alone data access, analysis, and visualization tool. For third party developers, MineSet provides the technological foundation needed 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.

System Architecture

MineSet employs a three tiered architecture. 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 moving it from one MineSet component to another. The mining tools are used to generate models that 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 PC or 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.

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

Copyright © 1997, American Association for Artificial Intelligence (www.aaai.org). All right reserved.

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 1 left) displays scatterplots with up to eight dimensions: three axes, entity color, entity size, entity rotation, and two independent attributes shown through animation. The Splat Visualizer extends the scatterplots when there are tens of thousands of records; it blurs the points using Gaussian smoothing. The Map Visualizer (Figure 1 right) 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 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: mapping attributes to graphical attributes (color, height, shape); manipulating the scene using thumb-wheels and mouse movements for rotation, translation, and zoom; data slicing and animating by manpulating sliders for two additional independent variables as shown in Figure 1 (left); drilling-down by pointing to elements in the scene; and searching and filtering the data. These capabilities allow users to visualize large high-dimensional databases without the information overload usually associated with such tasks.

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

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), and option decision trees (Kohavi & Kunz 1997). 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 optimize the standard defaults.

MineSet's analytical mining components are based on $\mathcal{MLC}$++ (Kohavi, Sommerfield, & Dougherty 1996), the Maching Learning library in C++. $\mathcal{MLC}$++ source code is freely available for research purposes, providing openness and promoting data mining research. 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. This openness also allows for easy model deployment, as nothing is hidden about the mining algorithms and nothing is proprietary about the generated models.

MineSet also uses the underlying Machine Learning and Statistics technology to aid users in other operations of the KDD process. The two most important examples are automatic binning (discretization) and feature selection. Automatic binning creates bins of (possibly) unequal width based on entropy minimization. For example, binning age with respect to salary for the UCI census file 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 aids users in finding the relevant attributes for their task. MineSet identifies the important attributes using conditional entropy minimization based on 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 that factor.

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. A 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. The Evidence Visualizer (Figure 2) shows a simple-Bayes model. The real power of the 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 Decision Tree Visualizer (Figure 3 left) shows a decision tree. 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 detail and additional levels are introduced into the scene. The Rule Visualizer (Figure 3 right) shows a set of association rules.

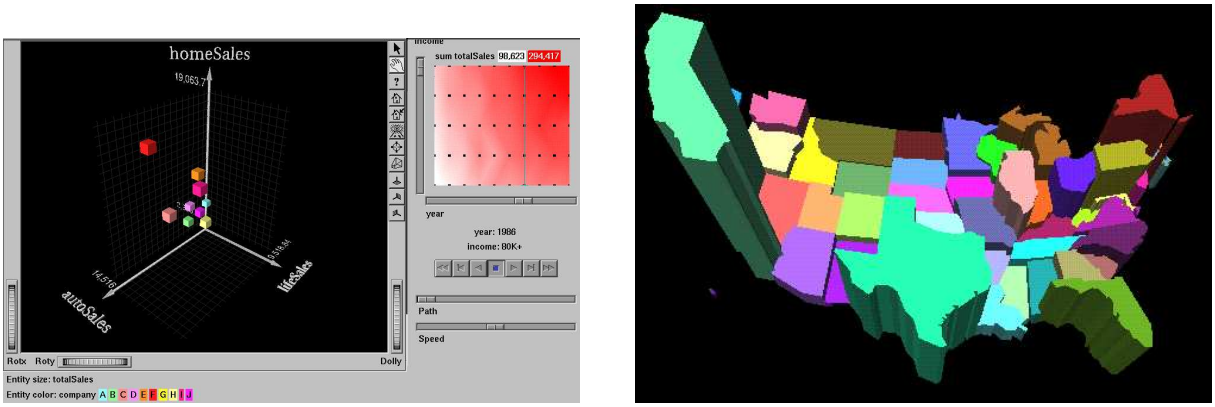

Figure 1: Left: A Scatterplot for 10 insurance companies and their policy sales. The cube size represents the total revenue. The upper right portion shows a 2D density plot for the two independent attributes for data slicing. The center right portion shows the VCR controls for animations. Right: A map of the US with height representing the US population in 1990.

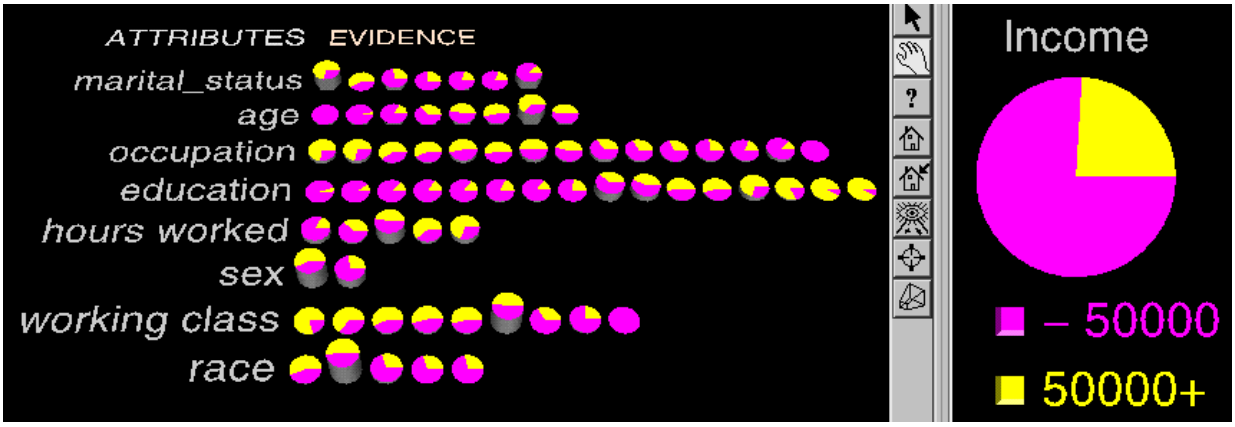

Figure 2: An Evidence Visualizer view for census data on adults working in the US. The pie in the right pane shows the prior probability distribution for adults making over and under $50,000. The pies in the left pane show each atttribute and the amount of evidence (pie slice size) each value or range 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). Each pie's height is proportional to the number of records in the corresponding value or range.

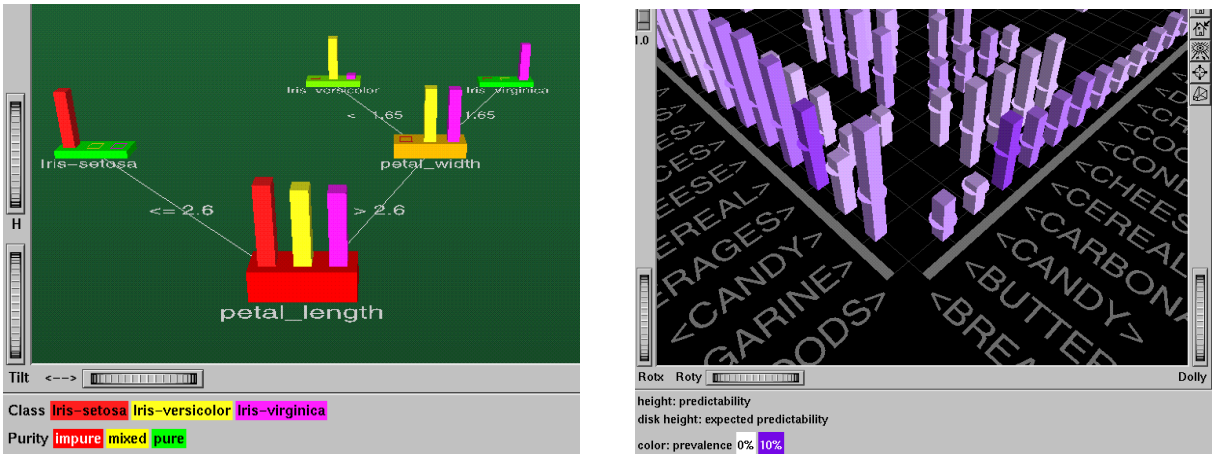

Figure 3: Left: A Tree Visualizer view for the iris dataset; the bars represent the class distribution, the base height represents the number of records, and the base color ranges from red (impure) to green (pure). Right figure: a Rule Visualizer view of supermarket items. The height of each bar represents how well the attribute value on the left predicts the attribute value on the right. The disks represent the probability of the right side alone. The colors represent the frequency of both the left hand side and the right hand side occuring together.

Managing the KDD Process

In addition to visualization and analytical mining, MineSet supports the broader knowledge discovery process (Fayyad, Piatetsky-Shapiro, & Smyth 1996). This section focuses on the role that Tool Manager and Data Mover play in helping execute and manage this process.

Tool Manager provides a consistent graphical interface to all of the tools. Selecting data sources, specifying data transformations, setting visualization and mining parameters are all done via a point-and-click interface. The Tool Manager also provides a graphic history for managing all the operations performed by the user and allows updating them in the future. While many data mining systems focus primarily on either the analytic tools or the visual tools, MineSet combines state-of-the-art analytic and visual tools with extensive data transformation capabilities.

The key data transformation facilities 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 a fast machine.

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

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 by summing purchases, thus giving a total purchase amount for each customer. In addition to summing, MineSet supports minimum, maximum, and average aggregation operations.

Transpose creates new attributes from multiple records for the same entity, allowing multiple transaction records to be transformed into a single customer "profile" that is 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.

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.

Summary

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

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 essential to effective mining and fast, thorough analysis.

Acknowledgments We would like to thank everyone on the MineSet team and Ross Bott who had the insight to initiate this project at the time when SGI was still a hardware-only company.

References

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

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

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

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

Kohavi, R., and Kunz, C. 1997. Option decision trees with majority votes. In Fisher, D., ed., *Machine Learning: Proceedings of the Fourteenth International Conference*. Morgan Kaufmann Publishers, Inc.

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

Kohavi, R.; Sommerfield, D.; and Dougherty, J. 1996. Data mining using $\mathcal{MLC}$++: A machine learning library in C++. In *Tools with Artificial Intelligence*, 234–245. IEEE Computer Society Press. `http://www.sgi.com/Technology/mlc`.

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

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

Srikand, R., and Agrawal, R. 1995. Mining generalized association rules. In *Proceedings of the 21st International Conference on Very Large Databases*.