Mirage: A Tool for Interactive Pattern Recognition from Multimedia Data

Tin Kam Ho

Bell Laboratories, Lucent Technologies

Abstract. Many data mining queries in astronomy involve the identification of objects that are similar or discernible in different aspects such as spectral shapes and features, light curves, morphology, positions, or other derived attributes. Analyses must go beyond conventional clustering algorithms that stop at computing a single proximity structure according to a specific criterion. We describe Mirage, a software tool designed for interactive exploration of the correlation of multiple partitional or hierarchical cluster structures arising in different contexts. The tool shows points, point classes, traversals of proximity structures in one, two, or higher dimensional projections, in linked views of various types or over an image background. It supports highly flexible layouts of plots, simple clustering procedures, intuitive graphical querying, and includes a command interpreter for further extensions. Mirage has found uses in many scientific and engineering contexts.

1. Introduction

An important class of questions in astronomical data analysis involve comparing objects simultaneously from different perspectives: positions in a specific coordinate system, image features, spectra, time variability, or other derived measures such as kinematic properties. Are the objects similar in spectral features in one band also similar in another band? Do objects with similar morphological features show similar patterns in their light curves? Are there unexpected correlations between the parameters suggesting a systematic error? Questions like these are routinely asked to confirm a classification, to discover new regularities, or to validate a data processing design. Common to all these questions is a need to correlate data from different representations. Mirage is a software tool designed to facilitate exploration of such correlations.

2. Analysis of Point Proximity

Mirage takes as input a data matrix, where a row represents an object and a column represents an attribute of the object. There are no limits on the number of rows or columns other than those imposed by machine capacity. One can define feature vectors that consist of arbitrary subsets of the attributes. A feature vector represents a set of measurements that are on a common scale and can be interpreted as a group, such as a spectrum. Instances of the same feature

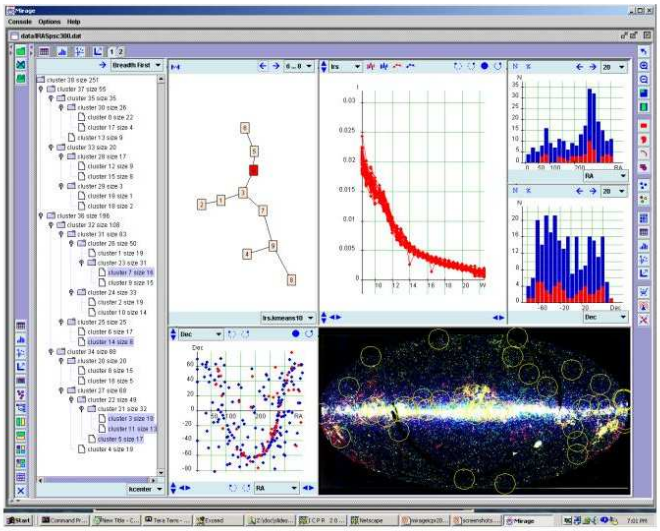


Figure 1. A screenshot showing a variety of views. A cluster is selected on the graph and tracked in other views. (IRAS data and image for demo are courtesy of Infrared Processing and Analysis Center, Caltech/JPL. IPAC is NASA's Infrared Astrophysics Data Center.)

vector are expected to be comparable by the same metric for cluster analysis. Mirage takes the results of a cluster analysis in the form of a data structure summarizing information about inter-point proximity. These data structures are correlated with simple one, two, or multi-dimensional projections of the data via linked displays that include histograms, scatter plots, parallel coordinate plots, tables, trees, and graphs (Figure 1). Users can highlight or color points from an arbitrary display and see the highlights or colors echoed in other displays. In addition, points can have attributes that describe their positions on an image background that is also linked to the displays. A page layout tool allows one to tile a view pane arbitrarily and open an arbitrary display in any tile. The tool keeps all displays neatly organized in a set of pages.

With this setup, one can track points that are in close proximity in different feature spaces that can be as simple as a single attribute or a 100-dimensional space of flux measurements. A user can append all available descriptions of the objects to the data matrix as columns. Some columns can be position angles, color estimates, flux within specific wavebands, where other groups of columns can be a spectrum, a set of morphological features extracted from an image, or luminosity estimates at various times. The selection operations and echoing mechanisms enable many kinds of proximity related queries in data mining.

3. Cluster Analysis from Multiple Perspectives

The design of Mirage is motivated by the need to study correlations of proximity relationships between points from different perspectives. In images, one or two dimensional projections (histograms or scatter plots), points falling in

a neighborhood are easy to find by visual inspection. The parallel coordinate plot can be used to find points that are close in projections on the individual dimensions. But detailed studies of proximity relationships in high-dimensional spaces require the aid of automatic cluster analysis.

A k-means procedure is provided that can be applied to an arbitrary feature space that the user defines. The results are returned as a minimum spanning tree (MST) connecting the centers of the resultant clusters. The MST is then embedded in a two-dimensional display by a graph layout algorithm and is linked with other views. Paths for leaf-to-leaf traversal are provided. User can walk along any path and see the corresponding points highlighted in other views. These paths represent the dominant directions of variations in the corresponding feature space, and the entire MST outlines a skeleton of the data distribution in that space. One can also compute a hierarchical cluster structure from an arbitrary feature space, and inspect the cluster tree in several standard traversals (e.g. depth-first, breadth-first). The walk is echoed in all other views.

Here cluster analysis is used as a way to compress the data and summarize their proximity relationships. One can initiate the k-means procedure with different numbers of desired clusters for a multi-resolution study. Computation of clusters is restricted to the chosen feature space where the metric (e.g. the Euclidean distance) is meaningful because the group of attributes are on a common scale. Thus one avoids the difficulty of finding a global metric that takes into account all attributes that can be on mixed scales. Cluster structures computed from each feature space can be correlated by walking on one structure and checking the echo on other structures. Histograms are treated as a special cluster structure where the clusters are points in each bin. One can walk along the array of bins and see the corresponding points highlighted in other views. This provides a convenient way for sensitivity analysis between spaces of arbitrary dimensions.

Since the data matrix can have arbitrary columns, it is easy to use Mirage along with other dimensionality reduction techniques such as projection to principal components or non-metric multidimensional scaling. One can simply compute the projected coordinates by an external procedure and merge them with the data set as new columns. The separation of computation and graphics simplifies the code and also allows for arbitrary coupling with those techniques.

4. Software Features

Mirage is written in Java with heavy use of the Swing library. The software is organized around an interpreter that takes commands from either the graphical interface or as textual input from a prompt. Commands can also be packaged in a script to be loaded at run time, or executed off-line for a preparatory analysis. A few features are emphasized in the design:

1. allow different treatments for different subspace projections,
2. facilitate traversals of partitional or hierarchical cluster structures,
3. provide easy-to-use user interfaces and data formats, and
4. support potential extensions by new commands.

Each graphical display includes a canvas and a control panel. Attributes, vectors, or cluster structures can be selected via the list boxes on the control

panel. The control panel can be removed once all display-specific choices are made, and re-inserted when needed. Split panes with movable dividers are used extensively to provide maximum flexibility in screen utilization.

Communication of selections between displays is done via passing points represented as Java objects to a console, and forcing a repaint. The repaint manager figures out which display is visible and calls the display-specific update method which looks up the selections from the console. Color tags are stored in a field of the point objects and used when a color plot is requested. By default all plots are monochrome for highlights to show up easily and to minimize visual clutter.

5. Usage Scenarios

In addition to proximity queries in a data mining context, Mirage can also be used in several other scenarios as follows.

Analysis of simulations. Simulations involving a parameter sweep can be studied in Mirage by tracking samples with neighboring input parameter values in the space of output features. One can also gain insights on the inverse model by finding points close in the output space and tracking the corresponding input values via the broadcasting mechanism.

Comparison of observation and theoretical prediction. One can construct a data matrix by appending the observations to the predictions, with a flag distinguishing between the two sets. Each set can then be selected using the flag and colored differently for comparison in all views.

Examination and verification of existing classification. The user can code the classes numerically and then open a histogram on the class code, and walk on the bins to examine in other views the attributes of points belonging to each class. Alternatively each class can be painted a different color for simultaneous comparison.

We have used Mirage in a large variety of contexts involving both observation or simulation data. Examples include analysis of optical fiber designs, testing for robustness of optical devices, monitoring network traffic, making diagnosis of wireless communication systems, inspecting spectral classes in the IRAS point source catalog, and checking for systematics in the data pipeline of the Deep Lens Survey. These exemplify many potential usages in astronomy driven by science or engineering goals.

Acknowledgments. I thank L. Cowsar, J.A. Tyson, D. Wittman, V. Margoniner, and A. Becker for motivations of the project, suggestions of many features, and trials of the early prototypes, A. Jain and G. Nagy for perspectives in pattern recognition, and W. Cleveland and D. Temple Lang for discussions on statistical graphics.

References

Inselberg, A., & Dimsdale, B. 1994, SIAM J. of Appl. Math., 54, 559.

Ho, T. K. 2002, Proc. 16th Int'l. Conf. on Pattern Recognition, II.4p.