Preliminary System Design for an EDA Assistant

Robert St. Amant and Paul R. Cohen
Computer Science Dept., LGRC
University of Massachusetts
Box 34610
Amherst, MA 01003-4610
stamant@cs.umass.edu, cohen@cs.umass.edu

1 Understanding Complex Data

Data analysis plays a central role in our attempts to understand the behavior of complex systems. While research in both statistics and artificial intelligence has addressed issues in the automation of later stages of analysis, such as theory generation, model selection, and experiment design [7], less attention has been given to initial exploration of data. Deriving structure from data is nevertheless a necessary first step.

Exploratory data analysis (EDA) [8] provides a wide range of statistical tools for looking at data. Human analysts find it straightforward to select and apply these tools effectively. The difficulty in automating exploration is one of *control.* At any point we can apply a large number of complex functional transformations or clustering operations to the data; to each result we can apply the same set of operations. The problem grows explosively.

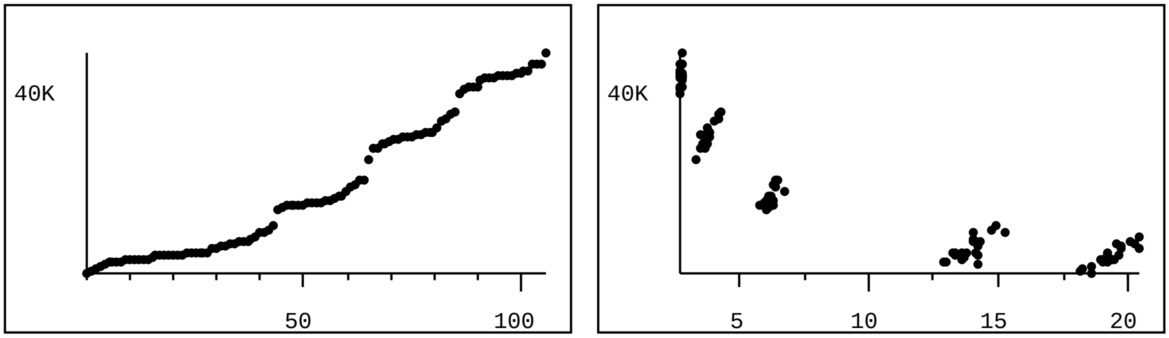
We have developed a blackboard-based design of an automated system that acts as an EDA assistant to a human analyst. The system maintains a sharp boundary between data-directed and goal-directed exploration. Data-directed mechanisms extract simple observations and suggestive indications from the data. EDA operations are then applied in goal-directed fashion to generate deeper descriptions of the data. Control rules guide the EDA operations, relying on intermediate results for their decisions.

This abstract gives an overview of the design, which is partly implemented. We emphasize that the work is incomplete.

2 Exploratory Data Analysis

We can best illustrate the EDA approach with a simple example. Much of our research deals with the behavior of AI planners in demanding simulation environments. One such system is TransSim, a transportation planner/simulator [6]. In an early experiment we examined the relationship between the costs of two resources, port cost (P) and ship cost (S), measured over the duration of a trial. Figure 1a shows the sorted values of S for the 107 trials of the experiment, Figure 1b the relationship between P and S (denoted $\langle P, S \rangle$.)

We begin with summary statistics for the variable S (Figure 1a): the mean is about 31, the median 30, the interquartile range 9.5, and there is a slight skew toward lower values. More significantly, there are three clear gaps that separate the data into four clusters. Our preliminary partial description of S comprises the statistics and our observations about the clustering.



a. Observation (x) vs S (y)

b. P (x) vs S (y)

Figure 1: Examples

When we turn to the relationship $\langle P, S\rangle$ (Figure 1b), we see a different pattern: the values fall into *five* clusters. The distinct separation in $\langle P, S\rangle$ values, as well as the observation that one of the S clusters is twice as large as the others, leads us to return to our description of S. We establish an alternative description of S as containing five clusters, consistent with the clusters in $\langle P, S\rangle$.

Continuing with our analysis of $\langle P, S\rangle$, we see that values in the first cluster (which we denote ps_1) can be fit by a straight line. In fact, this is true for all five clusters, though it is a different line in each case. Once we settle on the description of each cluster as a line, we can add the observation that the slopes of the lines decrease as the clusters move toward the right.

If we were to plot the central locations of the clusters ps_1 through ps_5 (e.g., the coordinates P_{median} and S_{median} for each cluster) we would see that these five summary points fall along a smooth curve. Further exploration shows that the curve is of the form $S_{median} = c/P_{median}$. When we perform this transformation it straightens the curve, leaving no clear pattern in the residuals.

To summarize, we begin with *initial descriptions* of the data, such as the observation of gaps between adjacent values. From these we generate *indications* [5], or suggestive characteristics: the data fall into clusters. Based on these indications we apply specific EDA *procedures*: we break the data down and analyze the clusters individually. These procedures may involve *iterative refinement*, as with the alternative descriptions of clusters in S. When we find that a particular description, a straight line, applies to one cluster of $\langle P, S\rangle$, we try to *generalize* that description to all clusters in the relationship. We extend the generalization by exploring features shared by the clusters—features derived from the generalization—such as the slopes of the lines. The result is a coherent, structured description of the data.

3 System Design

Exploration begins with *initial descriptions* of data. We generate summary statistics for single variables, correlations for bivariate relationships, single-linkage clustering for multivariate relationships, and so on. The results are all atomic: numbers and symbols, plus a few special, non-decomposable structures. These different types of description divide the data and derived results naturally into these types:

- Variables: Sequences of values, such as S and P, are variables. We can also generate derived variables, as we did by aggregating line slopes. (See Figure 2a.)
- Relationships between variables: A bivariate relationship such as $\langle P, S\rangle$ is a relationship between variables. Multivariate relationships and datasets are also examples. Notably, the

$\boxed{S} \rightarrow \langle 30.0, 35.1, \ldots \rangle$

a. Variable

$\boxed{\langle P, S \rangle} \rightarrow \left\langle \boxed{P}, \boxed{S} \right\rangle$

b. Relationship

$\boxed{\langle P, S \rangle_{cl}} \rightarrow \left\langle \boxed{ps_1}, \ldots, \boxed{ps_5} \right\rangle$

c. Composite Description

Figure 2: Data Structures

individual clusters ps_1 through ps_5 resulting from the partitioning of $\langle P, S \rangle$ all have the same form as the original relationship: they are derived relationships between variables. (See Figure 2b.)

- Relationships between relationships: The clusters ps_1 through ps_5 are related to one another, by their derivation from the same parent data. We represent this grouping of clusters by an explicit data structure, a *structural description* object. We similarly break down the $\langle P_{median}, S_{median} \rangle$ relationship into fit and residuals by a *functional description.* These data structures, relationships between relationships, are *composite* structural and functional descriptions of data. (See Figure 2c.)

The blackboard is divided into spaces, one for each of these data types. Initial descriptions are generated by data-driven knowledge sources, or KSs, associated with each space. When a structure, such as S, is added to the `variable` space on the blackboard, the description KSs (DKSs) associated with the space (thus with the type `variable`) automatically generate descriptions for it. The DKSs incrementally add information to the structures on the blackboard.

Indication KSs (IKSs) monitor the initial descriptions, looking for unusual properties. As with descriptions, these calculations are also carried out automatically. Indications often rely on heuristically set thresholds for their activation. For example, the IKS that detects the separations in S and $\langle P, S \rangle$ looks for outliers in the distance array produced by the clustering description. Large, outlying distances, as determined by a fourth-spread test [3], indicate gaps between clusters. Other indications include presence of outliers, high correlation, excessive skew, and curvature.

While the blackboard gives one view (by data type) of the structures produced by exploration, a different view is also possible: the interrelated structures form a hierarchical semantic network. The data and exploration results on the blackboard are related by semantic links running through and and between the spaces. The variable S on the `variable` space, for example, is related to $\langle P, S \rangle$ on the `relationship` space by a `component-of` link. The clusters ps_1 through ps_5 are related by `subset/superset` links to $\langle P, S \rangle$ on the same space. Lines and clusters on the `relationship between relationships` space are linked with their source relationships by `functional-description` and `structural-description`.

Specific EDA procedures, triggered by indications, add new structures to the hierarchical network, which simultaneously become visible on the blackboard. A high correlation indication in a bivariate relationship, for example, may cause a resistant line structure to be generated. A curvature indication in a sequence of residuals causes a heuristic straightening transformation to be applied. A gap indication, described earlier, causes the generation of clusters corresponding to the gap positions.

EDA procedures are implemented by script-like plans. As has been shown elsewhere [1, 4], AI planning structures provide a useful representation for exploration and modeling operations. As exploration plans execute they establish subgoals to be satisfied. Subgoals activate other plans in turn. Control is the one of the main issues in exploration—the desirability of finding new structure must be balanced against the possibility of the search space growing unreasonably large. Search is

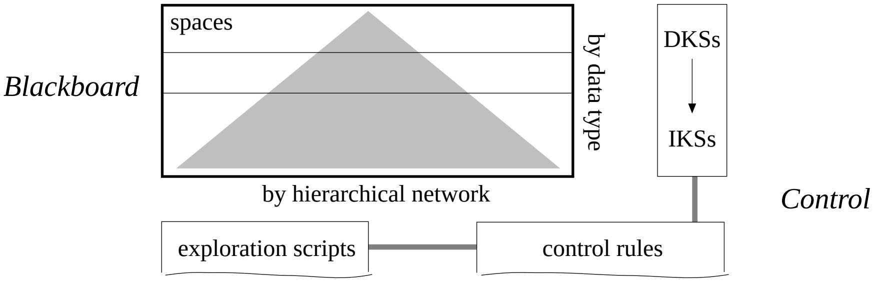


Figure 3: System Components

controlled here by focusing rules that control the matching of exploration plans with goals and the selection of structures to be explored. [2] While the initial stages of exploration involve data-driven calculation, the planning representation casts the later stages as goal-driven search.

Rules take advantage of the context provided by the hierarchical network to make their control decisions. They search through the network, with the purpose of

- selecting the structure to explore next, possibly suspending exploration of the current structure (e.g., we explore individual variables before relationships containing the variables);
- selecting appropriate plans, and suppressing inappropriate plans, for exploration of a structure (e.g., distinct clusters will often disturb a functional fit to a bivariate relationship);
- finding similarities between structures so that earlier results may be reused (e.g., successive line fits to clusters);
- reconsidering earlier decisions based on new information (e.g., reparameterizing a clustering script to increase consistency);
- incorporating user preferences into the analysis, by overriding default heuristics.

Control rules are invoked at explicit decision points to make the choices above. An example should give the flavor of the process. Suppose we are trying to characterize cluster ps_2 in the breakdown of $\langle P, S \rangle$, after having established a description of ps_1 as description $\langle$ *type* = `line`, F_1: `slope`, F_2: `intercept` $\rangle$. These rules become applicable:

IF structure S has an `is-a` sibling structure S_{sib},
and S_{sib} has associated description $D(F_1, F_2)$,
THEN or select plan apply-partial-description $D(F_1, F_2)$, S.

IF structure S_i has description D_i, and structure S_j has description D_j,
and S_i and S_j are `is-a` siblings with parent S_{par},
and D_i and D_j are *similar*,
THEN select plan generalize-descriptions D_i, D_j, S_{par}.

An overview of the system is shown in Figure 3. We have completed implementation of most of the required statistical procedures and data structures, as well as the planning structures that support the control rules. Our current work is aimed at integrating system components and extending the scope of the control rules.

References

[1] Robert St. Amant and Paul R. Cohen. Automated analysis of complex data. In *Proceedings of the Ninth Annual Goddard Conference on Space Applications of Artificial Intelligence*, 1994.

[2] Norman Carver and Victor Lesser. The evolution of blackboard control. *Expert Systems with Applications, special issue on the Blackboard Paradigm and its Applications*, 7(1), 1993.

[3] John D. Emerson and Michal A. Stoto. Transforming data. In David C. Hoaglin, Frederick Mosteller, and John W. Tukey, editors, *Understanding robust and exploratory data analysis.* Wiley, 1983.

[4] Amy L. Lansky and Andrew G. Philpot. Ai-based planning for data analysis tasks. *IEEE Expert*, 1993. forthcoming.

[5] Frederick Mosteller and John W. Tukey. *Data Analysis and Regression.* Addison-Wesley, 1977.

[6] Tim Oates and Paul R. Cohen. Toward a plan steering agent: Experiments with schedule maintenance. In Kristian Hammond, editor, *Second Internation Conference on Artificial Intelligence Planning Systems*, pages 134–139. AAAI Press, 1994.

[7] Jeff Shrager and Pat Langley. *Computational Models of Scientific Discovery and Theory Formation.* Morgan Kaufman, 1990.

[8] John W. Tukey. *Exploratory Data Analysis.* Addison-Wesley, 1977.