Interaction with a Mixed-Initiative System for Exploratory Data Analysis

Robert St. Amant
Department of Computer Science
North Carolina State University
Box 8206
Raleigh, NC 27695-8206
stamant@csc.ncsu.edu

Paul R. Cohen
Computer Science Department, LGRC
University of Massachusetts
Box 34610
Amherst, MA 01003-4610
cohen@cs.umass.edu

ABSTRACT

Exploratory data analysis (EDA) plays an increasingly important role in statistical analysis. EDA is difficult, however, even with the help of modern statistical software. We have developed an assistant for data exploration, based on AI planning techniques, that addresses some of the strategic shortcomings of conventional software. This paper illustrates the behavior of the system, gives a high level description of its design, and discusses its experimental evaluation.

Keywords

Artificial intelligence, planning, data exploration

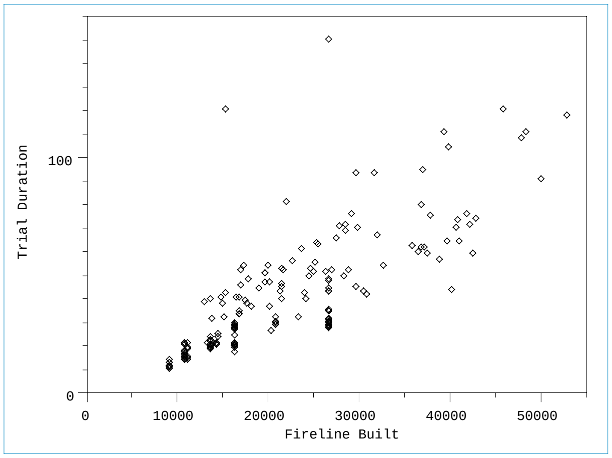
INTRODUCTION

Exploratory data analysis (EDA) has come to play an increasingly important role in statistical analysis. Modern computer-based statistics packages contain a rich set of operations, suitable for almost any EDA application. These systems are nevertheless limited; they are almost completely lacking in strategic ability. Imagine a statistical system with both a set of basic exploratory operations and a set of strategies for applying them. A user might say, "Generate a linear fit for this bivariate relationship." The system then generates a least-squares or perhaps a resistant fit, checks the residuals for indications (e.g. curvature, outliers, unequal variance), performs appropriate transformations, iteratively refits the data if necessary, and reports all interesting results. The user might say, "There are clusters in this relationship," prompting the system to search for potential relationships between the clusters, to examine the behavior of the data internal to each cluster, to search other variables and relationships for similar behavior, and to present its findings. Further, the same system might initially suggest one of these lines of analysis, based on its own evaluation of the data. This system, instead of being a repository of statistical tools and techniques, comes closer to acting as an automated statistical assistant.

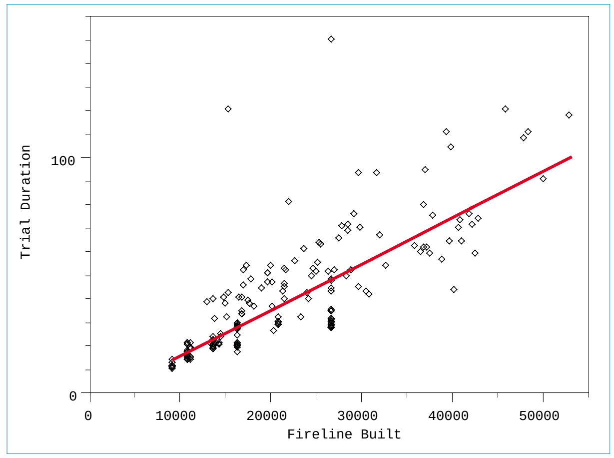
Two properties let us call a system an assistant rather than a sophisticated toolkit. First, an assistant is at least partly autonomous. We can give an assistant general instructions and let it make its own decisions about how to carry them out. Second, an assistant responds to guidance as it works. An automated system will inevitably make mistakes from time to time, so its reasoning process (past decisions as well as current ones) must be available to the user for approval or modification. A responsiveness to the guidance provided by human knowledge of context has been termed "accommodation" [14]. An accommodating system takes advantage of human knowledge to augment its own necessarily limited view of the world. The combination of autonomy with accommodation lets the human data analyst shift some of the routine or search-intensive aspects of exploration to an automated system, without giving up the ability to review and guide the entire process.

We have developed an assistant for intelligent data exploration called Aide. Aide is a knowledge-based planning system that incrementally explores a dataset, guided by user directives and its own evaluation of the data. Its plan library contains a set of strategies for generating and interpreting indications in data, building appropriate descriptions of data, and combining results in a coherent whole. The system is mixed-initiative, autonomously pursuing high- and low-level goals while still allowing the user to inform or override its decisions.

This paper begins with an example of an exploratory session, which describes the capabilities we expect of an automated assistant—capabilities that Aide provides. We then discuss the issues in Aide's mixed-initiative design. The paper ends with a brief account of an experimental evaluation of the system.



(a) Initial relationship



(b) Resistant fit

Figure 1: Patterns in FirelineBuilt and Trial Duration (with continuation on next page)

EXPLORING DATA

We can best understand AIDE's behavior with an example, which we take from an experiment with PHOENIX, a simulation of forest fires and fire-fighting agents in Yellowstone National Park [6]. The experiment involved setting a fire at a fixed location and specified time, then observing the behavior of the fireboss (the planner) and the bulldozers (the agents that put out the fire). Variability between trials is due to randomly changing wind speed and direction, non-uniform terrain and elevation, and the varying amounts of time agents take in executing primitive tasks. In this experiment forty variables were collected over the course of some 340 PHOENIX trials, including measurements of the wind speed, the outcome (success or failure), the type of plan used, and the number of times the system needed to replan. We became interested in the relationship between the time it takes the planner to put out a fire (Duration) and the amount of fireline built during the trial (FirelineBuilt). Figure 1a shows a scatter plot of these two variables.

The dialog below shows how the user interacts with AIDE in exploring the dataset. For conciseness, the exchange is presented as a natural conversation; the dialog actually takes place within a statistical interface, through menu selections, active object selection, typed input, and graphical displays. The information exchanged on the part of both conversants, however, is accurately rendered.

AIDE: (FirelineBuilt, Duration) has these indications: high correlation ($r = 0.93$); outliers in both FirelineBuilt and Duration; clustering. Suggested courses of action are (1) generating a linear fit and (2) exploring clusters.

USER: Why the linear fit?

AIDE: A linear fit can be appropriate for relationships with high correlation.

USER: Why the given ordering of actions?

AIDE: When there is a high correlation and no evidence of extreme outliers, we fit a line before trying other possibilities.

USER: Okay.

AIDE: (Generates linear fit, as shown in Figure 1b.) A three-group resistant line is appropriate; its intercept is -0.96, its slope 0.0017. There are outliers in the residuals, as well as evidence of clustering.

USER: Show recent decisions.

AIDE: (1) With the goal of exploring (FirelineBuilt, Duration), we decided to fit a line. The active alternative is to explore clusters. (2) With the goal of fitting a line, I decided without consultation on a resistant fit. Alternative is a regression fit.

USER: Back up a step.

AIDE: In fitting a line to the relationship, the possibilities are a resistant fit, which is in progress, or a regression fit, which has not started.

USER: Execute the regression fit.

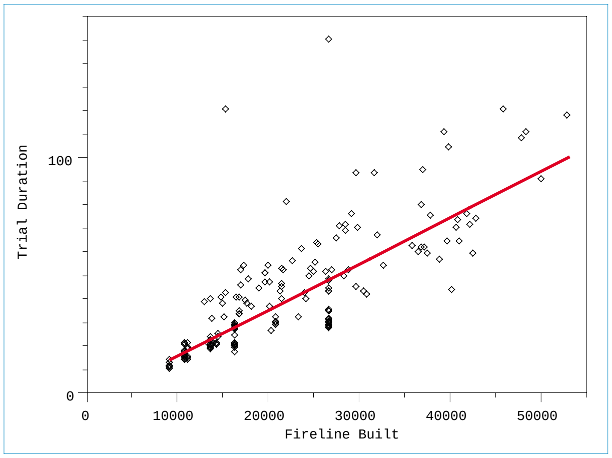
AIDE: (Generates regression fit, as shown in Figure 2c.) The regression line has an intercept of -2.73 and a slope of 0.0019; $r^2 = 0.57$. There are outliers in the residuals, as well as evidence of clustering. The slope of the regression line is 1.13 times the slope of the resistant line.

USER: Make a note: I am confident that the relationship (FirelineBuilt, Duration) is linear, despite indications of outliers and clustering. I prefer the regression fit.

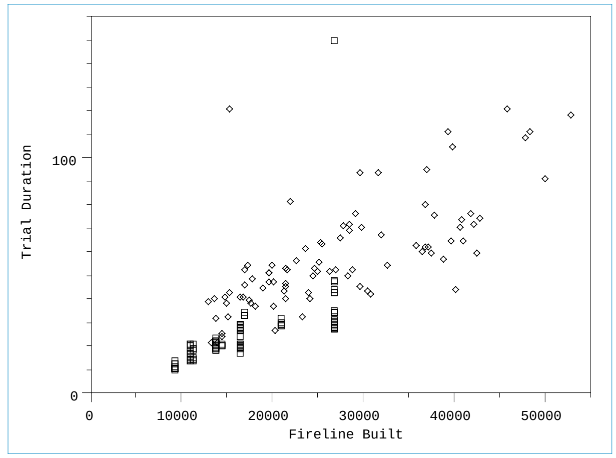
AIDE: (Accepts note.)

USER: Okay.

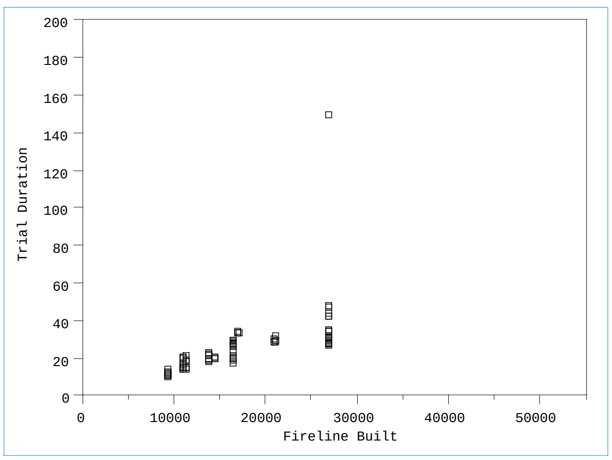
AIDE: (Considers linear fit phase to be complete.) The possibilities remaining to explore (FirelineBuilt, Duration) are (1) the linear fit, which is in progress, and (2) cluster exploration, which has



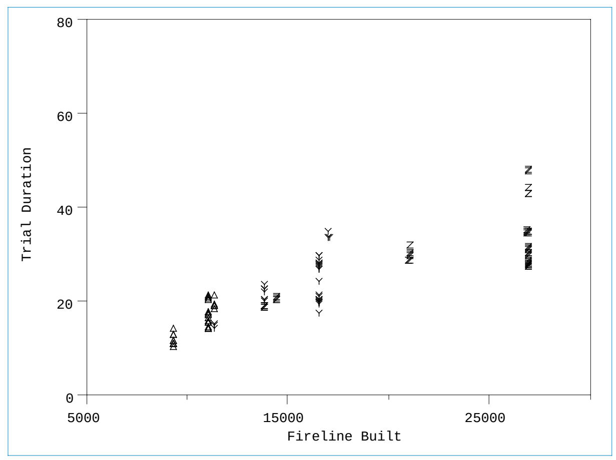
(c) Regression fit



(d) Vertical clusters



(e) Vertical clusters isolated



(f) Vertical clusters colored by wind speed

Figure 2: Patterns in FirelineBuilt and Trial Duration (continued)

not started.

User: Okay.

Aide: (Begins cluster exploration.) There are several different ways (FirelineBuilt, Duration) can be broken into clusters. (Displays identifiers for potential clustering criteria: single-linkage clusters in FirelineBuilt, regions of constant values in FirelineBuilt, etc. Also displays similarity between resulting clusters.)

User: Show constant FirelineBuilt clusters.

Aide: (Displays clusters, as shown in Figure 2d.)

User: Explore clusters.

Aide: (Isolates clusters, as shown in Figure 2e.) ...

By searching for related behavior in other variables, the analysis eventually determines that the clusters are an effect of not having to replan; that is, observations fall into clusters in FirelineBuilt only in cases where the variable #Replans takes on a value of 0. A better view of the general pattern of clustering is given by each cluster's central location, its median FirelineBuilt and Duration value. Once the clusters have been reduced to a set of representative values, they can be described in turn. These points also follow a linear pattern, with a slope slightly less than that of the line fitting the entire partition. Associating each cluster with a unique identifier leads to the additional finding that the discrete variables WindSpeed and PlanType together strongly predict cluster membership; a scatter plot of the clusters, colored by wind speed, is shown in Figure 2f.

The exploration continues, but this should be enough to give the flavor of the interaction. (The analysis and Aide's participation are described in more detail elsewhere [5, 17].)

MIXED-INITIATIVE ASSISTANCE

Aide's design exploits a striking similarity between interactive data exploration and planning [18, 19], especially partial hierarchical planning [9]. Briefly, a partial hierarchical planner has these properties:

A plan library: A great deal of procedural knowledge is not generated from scratch when required, but rather retrieved from memory of past experience. A partial hierarchical planner maintains a library of general-purpose and specific plans.

Hierarchical plans: Plans in the library are not necessarily elaborated down to the level of primitive operators; they often specify behavior in terms of subgoals. A plan to build a house, for example, might contain two high-level goals: "Lay the foundation" and "Erect the walls."

Explicit control: A plan may establish an explicit procedural specification for the way its component subgoals are to be satisfied, or actions to be executed. Control may be sequential, conditional, iterative, or some more specialized combination. For example, the house-building plan would probably specify, "*First* lay the foundation, and *then* erect the walls."

Interleaved generation and execution: The planner may execute a plan that has not yet been completely elaborated to the operator level; for example, one might want to complete the foundation before deciding where to put walls.

Meta-level reasoning: When more than one plan can potentially satisfy a single goal, this results in a *focus point*, at which the planner must choose among the applicable plans to continue with the exploration. As plans execute, a network of such focus points is created. The planner may opportunistically revisit and modify focus point decisions in order to follow the most promising path to a solution.

This kind of planning is well-suited to exploration. EDA makes use of abstraction, problem decomposition, and procedural knowledge, three defining characteristics of planning [13]. Further, EDA is reactive. One cannot anticipate every pattern that might possibly appear in the data; rather, the analysis is driven by the data, which argues for integrating the generation and execution of procedures. In addition, exploratory procedures often need explicit control. Some common EDA techniques, like resistant line generation, smoothing, and lowess, are iterative, while other techniques need sequencing, conditionals, mapping, and other kinds of control. Finally, exploration is constructive. An exploratory result is not simply a graph or a statistical summary, but also includes a set of supporting decisions, which give the context for results to be interpreted appropriately.

AIDE's library contains about a hundred plans, at different levels of detail. At the primitive level, these plans call a set of heavily parameterized operations for computing reductions of multidimensional arrays to scalar values (e.g. summary statistics), transformations (e.g. power transforms), and decompositions (e.g., isolation of clusters). Primitive operations for special-purpose modeling procedures are also available. Higher-level plans based on these primitives are intended to capture elements of common statistical practice, such as the examination of residuals after fitting a function to a relationship, the search for refinements and predictive factors when observing clustering, the reduction of complex patterns to simpler ones, and so forth. The plans lack a human-level knowledge of subject-matter context—what the data actually mean—but they are sensitive to the procedural context in which they are applied.

AIDE plans as follows. When a dataset or relationship is presented to the system, a goal is established for its exploration. The planner searches through its library for an appropriate plan and expands it, that is, establishes a set of new subgoals to be satisfied. These subgoals are satisfied in turn by plans from the library. Goals can also be satisfied directly by primitive actions, which execute code directly rather than establishing new subgoals. This process is more complex than it might initially appear: often, several plans in the library can satisfy a single goal, and there may be an unlimited number of ways to bind a given plan's internal variables to different values. For each decision, or focus point, the planner relies on a set of control rules to decide which plan or variable binding to select. As planning continues, the planner may sometimes backtrack to one of these focus points to make a different selection. The process continues until the goal at the top level has been satisfied. Thus we cast exploration as a problem of constructing and navigating through a network of decisions, represented by focus points. Execution of each primitive action generates one or more new results; the network combines all the findings.

Let's return to our notions of autonomy and accommodation, to see how they are supported by this process. As might be expected, autonomy is provided by the focus point network and the library of plans. At any point in the exploration AIDE considers one decision to be its current focus of attention. The system acts autonomously by making this decision without consulting the user. Because plans reflect common statistical practice, this behavior often has the effect of anticipating the user's actions. Not all decisions are handled this way; for some types of more difficult decisions, the default behavior is to stop and ask the user how to proceed. In these cases AIDE will present its own advice as well, but it will not proceed without an acknowledgment or an explicit directive from the user.

As to accommodation, AIDE can be used as a conventional menu-based statistics package. Menu choices let the user load a dataset, compose variables into relationships, compute summary statistics, generate linear models, partition data, run statistical tests, and so forth.

These menu operations are tied internally to the focus point network, so that if the user tells AIDE to run a regression of y on x, the system will search through the network to find a decision point associated with selecting relationships, find or create an appropriate branch for (x, y), and then incrementally select a sequence of plans that run the regression, explore the residuals, evaluate the results, and so forth. All this remains invisible to the user, who sees only the result. At this point the user can select another relationship or statistical procedure and proceed. The important aspect of this interaction is that the user is not constrained to consider only those decisions AIDE considers relevant, but can pursue his or her own goals in the exploration.

Beyond providing access to conventional statistical operations, AIDE gives the user an explicit representation, through the focus point network, of the decision-making process. This kind of explicit, structured justification for decisions and results is an important aspect of exploration [12]. Further, the network lets the user explicitly direct AIDE's actions. When AIDE reaches a focus point, its decision at that point can be reviewed and potentially modified by the user. In fact, all decisions made by the user or the system are available for review and possible revision. The explicit network of decisions provides the basis for our metaphor of exploration as navigation.

AIDE's partial hierarchical design is an effective framework for mixed-initiative planning. James Allen has identified three distinguishing characteristics of mixed-initiative planning, by analogy to dialog behavior. Mixed-initiative planners support flexible, opportunistic control of initiative, the ability to change focus of attention, and mechanisms for maintaining shared, implicit knowledge [1]. We can adapt these criteria to our design as follows. The plan library provides the planner its representation of shared knowledge about reasonable courses of action, while the network of focus points gives the current state of the exploration. Changing focus of attention means deciding which focus point should next be under consideration to extend the exploration. Flexible control of initiative is a matter of deciding whether the system or the user should make the next move in selecting a new focus point or making a decision at the current one.

EVALUATION

Evaluation focused on a simple hypothesis:

Exploration is more effective
with AIDE than without.

The experiment involved testing subjects under two conditions. In the USER+AIDE condition, subjects explored a dataset with AIDE's help, while in the USER-ALONE condition, subjects explored a dataset in a similar statistical computing environment, but without active interaction with AIDE. AIDE's effectiveness was then determined by measuring differences in performance between the two conditions.

Several factors can potentially counfound an experiment like this: different subjects may have different facility with EDA techniques; user interaction may be different under the two conditions; the datasets to be explored may contain different types of structure and patterns; and the order in which conditions are presented to subjects may make a difference.

To control for these and other effects, we set up the experiment as follows.

All subjects explored the same two datasets, one in the USER+AIDE condition and the other in the USER-ALONE condition. The interface was identical in both cases, lacking only AIDE functionality in the USER-ALONE condition. The datasets contained artificial data, generated by similar but not identical means, to provide equivalent problems to be solved in both conditions. The dataset/condition assignment was randomized, as was the order in which the datasets were explored. Because of the time and effort involved in overseeing individual trials, which lasted on the order of four hours per subject, the experiment was limited to eight subjects.

The generation of a dataset followed roughly this procedure. Start with a directed acyclic graph of twenty nodes. Each node corresponds to a variable. Associate with each node a simple function of the arcs from its incoming variables; for example, if a node c has arcs from a and b, the function might be $c = a \times b - b + \epsilon$, where ϵ is normally-distributed noise. Nodes with no incoming arcs, or exogenous nodes, are associated with specific distributions. A row of the dataset is computed by sampling from each exogenous node's distribution, and "pushing" these values through the rest of the graph. By repeating this process many times, we can collect as many rows as we need. The two datasets for the experiment were generated from graphs almost identical in structure and with comparable distributions and functions attached to the nodes and arcs. In the experiment, subjects were instructed to identify the direct relationships in the data and to describe them (i.e., as linear relationships, clusters, power relationships, and so forth.)

Our measurements in the experiment mainly concerned accuracy. In each condition c a subject s makes some number of observations: $O_{cs} = o_{cs1}, \ldots, o_{csk}$. We can classify each observation as correct or incorrect. By "correctness" we mean that the subject has associated an appropriate description with one of the direct relationships in the model that generated the data. The first measure, $\overline{p}$, is the mean number of correct observa-

	$\overline{p}$		$k\overline{p}$		$\overline{i}$		$k\overline{i}$	
	AIDE	ALONE	AIDE	ALONE	AIDE	ALONE	AIDE	ALONE
Subject 1	0.29	0.34	4.0	5.5	0.538	0.455	7	5
Subject 2	0.39	0.29	3.5	3.5	0.667	0.417	6	5
Subject 3	0.50	0.21	3.0	1.5	0.875	0.285	7	2
Subject 4	0.56	0.37	10.0	7.0	0.632	0.579	12	11
Subject 5	0.44	0.29	4.0	2.0	0.556	0.500	5	3
Subject 6	0.34	0.50	4.5	5.5	0.571	0.583	8	7
Subject 7	0.50	0.07	3.0	1.0	0.500	0.429	3	6
Subject 8	0.59	0.36	6.5	1.5	0.667	0.500	8	2

Table 1: Average correct ($\overline{p}$, $\overline{i}$) and total correct ($k\overline{i}$, $k\overline{i}$) observations per subject

tions made,

$$\overline{p} = \frac{\sum_{i=1}^{k} \text{Correct?}(o_{csi})}{k},$$

where k is the number of observations the subject makes in a condition and *Correct?* is a function that returns 1 if an observation is correct, 0 otherwise. Informally, $\overline{p}$ for a given subject gives the probability that one of his or her observations is correct.

The $\overline{p}$ performance measure takes both correct and incorrect judgments into account, which may sometimes be deceptive. We would like to distinguish, for example, between a subject with a $\overline{p}$ of 0.5 for a large number of observations and another subject with the same $\overline{p}$ score for many fewer observations. Another measure, which we call $k\overline{p}$ for consistency, considers number of correct observations alone:

$$k\overline{p} = \sum_{i=1}^{k} \text{Correct?}(o_{csi}).$$

We'll also consider a refinement of these two measures. Performance contains two components: identifying a significant variable or relationship and correctly describing it. We thus considered two further measures, $\overline{i}$ and $k\overline{i}$, which are comparable to $\overline{p}$ and $k\overline{p}$ but call an observation "correct" simply if a direct relationship is identified, ignoring its descriptive form (linear, cluster, nonlinear, etc.)

Subject performance is shown in Table 1. A matched-pair, one-tailed t-test tells us that $\overline{p}$ and $k\overline{p}$ are significantly higher for subjects in the USER+AIDE condition: $t = 2.217$ and 1.808, with p-values around 0.03 and 0.05, respectively. (We use a one-tailed test because we are interested in whether performance in the USER+AIDE condition is better than in the USERALONE condition, rather than simply seeing a difference in either direction.) A similar result holds true for $\overline{i}$ and $k\overline{i}$.

This comparison tells us that AIDE contributes significantly to the correctness of a given user's observations, on average, and that AIDE contributes to a higher total number of correct observations as well. To put this in perspective, we can dismiss a few plausible but trivial explanations for better performance in the USER+AIDE condition. First, subjects entered roughly the same number of observations in both conditions, with a median difference of 0.5 between the two conditions. For all subjects, the mean number of observations in the USER+AIDE condition was 14.1, in the USER-ALONE condition 13.0. Improved performance thus depends not only on making more correct observations, but also on making fewer incorrect observations. Further, subjects directly examined about the same number of variables and relationships in both conditions: 73 for USER+AIDE, 66 for USERALONE on average per subject. The difference between conditions is not significant, so better performance is not due to subjects simply seeing more of the data in the USER+AIDE condition. It is also not the case that subjects in the USERALONE condition never happen upon the relationships and patterns suggested by AIDE in the USER+AIDE condition. Of all the correct suggestions AIDE made about each dataset, only one was not also tried by subjects in the USERALONE condition.

Now, the simple fact of the performance difference is not entirely satisfying. We are really most interested in understanding why AIDE works. For a better view of AIDE's contribution, we divided subject actions in the USER+AIDE condition into three categories: Data-Manipulations, or the execution of a statistical test or procedure; Local-Decisions, or deciding between local alternatives, such as between a clustering description and a linear fit; and Navigation, or moving from one focus point decision to another. Perhaps surprisingly, we found that data manipulation was the smallest category of actions. Navigation was responsible for almost half (44%, on average, per subject) of the operations, with local decisions accounting for another 38%. Only 13% of the operations actually manipulated the data.

These summaries give us a rough idea of how subjects went about exploring a dataset. Much of the effort, in terms of the number of operations applied, involved examining the data from different angles and evaluating ways of building descriptions. Subjects showed a good deal of mobility, not just in moving from one data structure to the next, but also in moving from one point in the network of exploratory plans and actions to another. This point was also emphasized by most subjects in post mortem assessments: a common theme was the importance of being able to navigate through the exploration process. The summaries also show that data manipulation was secondary to other activities; we might think of navigation and local evaluation of decisions as setting the stage for data manipulation.

In examining how these factors are related, we find that Data-Manipulation and Local-Decisions are weakly correlated to begin with ($r = 0.29$), and if we hold Navigation constant the correlation drops to 0.12. We thus have a chain going from Data-Manipulation to Navigation to Local-Decisions—or vice versa; the ordering is ambiguous. Our tentative interpretation is that data manipulation operations generate new data, which can then be explored; thus the number of Data-Manipulation operations influences Navigation. Increased navigation leads in turn to the generation of new focus points, which requires more local decision-making, which means that Navigation influences Local-Decisions. This is only a tentative model of these relationships, because we have relatively few data points, but it is plausible and suggestive. We have planned further tests to examine the issue in more detail.

RELATED WORK

This work draws on a number of different sources. The clearest relationship is to early work in developing concepts of statistical strategy, or the formal descriptions of actions and decisions involved in applying statistical tools to a problem [11]. Gale and Pregibon's REX system, for example, implemented a strategy for linear regression [8]. Oldford and Peters implemented a complex strategy for collinearity analysis [15]. The goals of AIDE bear a resemblance to those of Lubinsky and Pregibon's TESS [14], which supports analysis by accommodating user knowledge of context in a search good descriptions of data.

AIDE has also been influenced by work examining human interaction with systems for data exploration. AIDE's representation of primitive EDA operations, for example, is very similar to that of IDES, the Interactive Data Exploration System, a component of SAGETOOLS [10, 16]. IDES and SAGETOOLS are the descendants of research in automatic presentation systems, which are intended to relieve users of the need for graphical design and display knowledge. AIDE concentrates on a different aspect of the same problem, on representing the process of exploration.

The knowledge discovery in databases literature also describes systems comparable to AIDE, for example Brachman et al.'s IMACS, which is aimed at the task of "data archaeology" [3]. Data archaeology is distinct from data mining, in which an autonomous statistical or machine learning algorithm searches a large database for implicit patterns. Data archaeology recognizes that results do not emerge in a single pass over the data, but rather evolve in an iterative process that requires constant human interaction. We hold this same view, in contrast to many who hope to build completely autonomous knowledge discovery systems.

Our goals for AIDE are similar to those adopted by other researchers interested in the use of planning in intelligent user interfaces. Bonar and Liffick, for example, use planning to support novice users without unduly constraining experts, and vice versa [2]. Planning is an integral part of many agent-based user interfaces [4, 7]. As an integration of reactive planning methods and user interface technology, AIDE is an example of the potential benefits of this approach.

Finally, AIDE can also be viewed in some ways as a collaborative system, where collaboration is a process in which two or more agents work together to achieve shared goals. Loren Terveen has identified a set of issues that must be addressed by any system that collaborates in an intelligent way with its users [20]: reasoning about shared goals; planning, allocation, and coordination in achieving these goals; awareness of shared context; communication about goals, coordination, and evaluation of progress; and adaptation and learning. Of these, AIDE concentrates on planning and coordination. Other aspects of collaboration are not addressed in the current implementation, but are a part of our plans for future work.

ACKNOWLEDGMENTS

We are grateful for the helpful comments of an anonymous reviewer. This research is supported by ARPA/Rome Laboratory under contract #F30602-93-0100 and by the Dept. of the Army, Army Research Office, under contract #DAAH04-95-1-0466. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes not withstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the Advanced Research Projects Agency, Rome Laboratory, or the U.S. Government.

REFERENCES

[1] James F. Allen. Mixed initiative planning: Position paper. [WWW document]. Presented at the ARPA/Rome Labs Planning Initiative Workshop. URL http://www.cs.rochester.edu/research/trains/mip/, 1994.

[2] Jeffery Bonar and Blaise W. Liffick. Communicating with high-level plans. In Joseph W. Sullivan and Sherman W. Tyler, editors, *Intelligent User Interfaces*. ACM Press, 1991.

[3] Ronald J. Brachman, Peter G. Selfridge, Loren G. Terveen, Boris Altman, Alex Borgida, Fern Halper, Thomas Kirk, Alan Lazar, Deborah L. McGuinness, and Lori Alperin Resnick. Integrated support for data archaeology. *International Journal of Intelligent and Cooperative Information Systems*, 1993.

[4] David N. Chin. Intelligent interfaces as agents. In Joseph W. Sullivan and Sherman W. Tyler, editors, *Intelligent User Interfaces*. ACM Press, 1991.

[5] Paul R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 1995.

[6] Paul R. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32–48, Fall 1989.

[7] Oren Etzioni and Dan Weld. Intelligent agents on the internet: Fact, fiction, and forecast. *IEEE Expert*, 8(4):44–49, August 1993.

[8] W. A. Gale. REX review. In W. A. Gale, editor, *Artificial Intelligence and Statistics I*. Addison-Wesley Publishing Company, 1986.

[9] Michael P. Georgeff and Amy L. Lansky. Procedural knowledge. *Proceedings of the IEEE Special Issue on Knowledge Representation*, 74(10):1383–1398, 1986.

[10] J. Goldstein and S.F. Roth. Using aggregation and dynamic queries for exploring large data sets. In *Proceedings of the CHI'94 Conference*, pages 23–29. Association for Computing Machinery, 1994.

[11] D.J. Hand. Patterns in statistical strategy. In W.A. Gale, editor, *Artificial Intelligence and Statistics I*, pages 355–387. Addison-Wesley Publishing Company, 1986.

[12] Peter J. Huber. Languages for statistics and data analysis. In Peter Dirschedl and Ruediger Ostermann, editors, *Computational Statistics*. Springer-Verlag, 1994.

[13] Richard E. Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33:65–88, 1987.

[14] David Lubinsky and Daryl Pregibon. Data analysis as search. *Journal of Econometrics*, 38:247–268, 1988.

[15] R. Wayne Oldford and Stephen C. Peters. Implementation and study of statistical strategy. In W.A. Gale, editor, *Artificial Intelligence and Statistics I*, pages 335–349. Addison-Wesley Publishing Company, 1986.

[16] S.F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the CHI'94 Conference*, pages 112–117. Association for Computing Machinery, 1994.

[17] Robert St. Amant. *A Mixed-Initiative Planning Approach to Exploratory Data Analysis*. PhD thesis, University of Massachusetts, Amherst, 1996. Also available as technical report CMPSCI-96-33.

[18] Robert St. Amant and Paul R. Cohen. Control representation in an EDA assistant. In Douglas Fisher and Hans Lenz, editors, *Learning from Data: AI and Statistics V*, pages 353–362. Springer-Verlag, 1996.

[19] Robert St. Amant and Paul R. Cohen. A planner for exploratory data analysis. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems*, pages 205–212. AAAI Press, 1996.

[20] L. G. Terveen. Intelligent systems as cooperative systems. *International Journal of Intelligent Systems*, 3(2–4):217–250, 1993.