ELSEVIER

Knowledge-Based Systems 10 (1998) 265–273

Knowledge-Based SYSTEMS

Interaction with a mixed-initiative system for exploratory data analysis

Robert St. Amant[a,*], Paul R. Cohen[b]

[a]Department of Computer Science, North Carolina State University, P.O. Box 8206, Raleigh, NC 27695-8206, USA
[b]Computer Science Department, LGRC, University of Massachusetts, P.O. Box 34610, Amherst, MA 01003-4610, USA

Received 16 June 1997; accepted 5 August 1997

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 describes the design and behavior of the system and discusses an experimental evaluation that demonstrates the effectiveness of our approach. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Artificial intelligence; Planning; Data exploration

1. 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' [15]. 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.

* Corresponding author. Tel.: +1 919 515 7938; fax: +1 919 515 7896; e-mail: stamant@csc.ncsu.edu

PII S0950-7051(97)00038-5

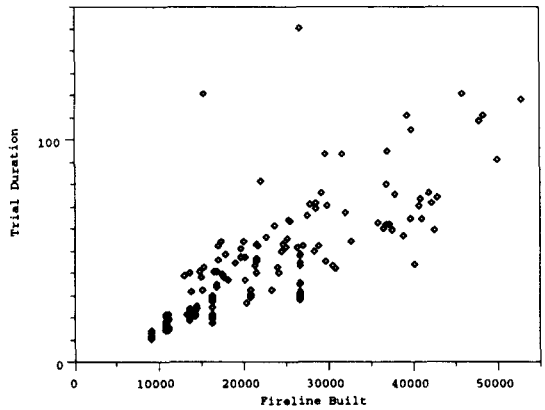

Fig. 1. Initial relationship.

The paper ends with an account of an experimental evaluation of the system.

2. Exploring data

We can best understand AIDE's behavior with an example, which we take from an experiment with AIDE, a simulation of forest fires and fire-fighting agents in Yellowstone National Park [5]. 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, nonuniform 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 AIDE 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). Fig. 1 shows a scatter plot of these two variables.

The dialog below shows how the user interacts with AIDE in exploring the dataset, beginning with the analysis of this relationship. For conciseness, the exchange is presented as a natural conversation; the dialog actually takes place within a statistical interface, through menu item selections, active object selection, typed input and output, 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 Fig. 2.) 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), it was decided to fit a line. The active alternative is to explore clusters. (2) With the goal of fitting a line, the system chose a resistant fit, without consultation. 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 Fig. 3.) 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: The relationship (FirelineBuilt, Duration) is linear, with high confidence, despite indications of outliers and clustering. 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 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 Fig. 4.)
USER: Explore clusters.
AIDE: (Isolates clusters, as shown in Fig. 5.)

The analysis eventually determines that the vertical clusters are an effect of not having to replan; that is, observations fall into clusters only when the variable #Replans has 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

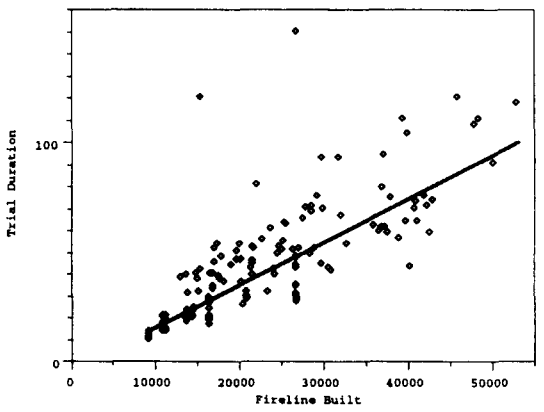

Fig. 2. Resistant fit.

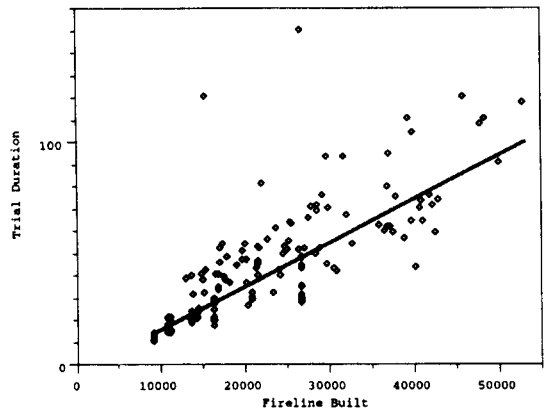

Fig. 3. Regression fit

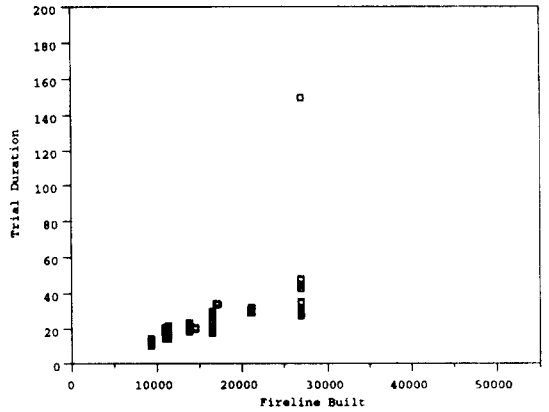

Fig. 5. Vertical clusters isolated.

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 Fig. 6.

The exploration continues, but this short dialog characterizes the general behavior of the system and its interaction with the user. AIDE's analysis is comparable to an exploration of the same data originally carried out by hand [4].

3. Mixed-initiative assistance

AIDE's design exploits a striking similarity between interactive data exploration and planning [22,23], especially partial hierarchical planning [8]. 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 a single plan 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

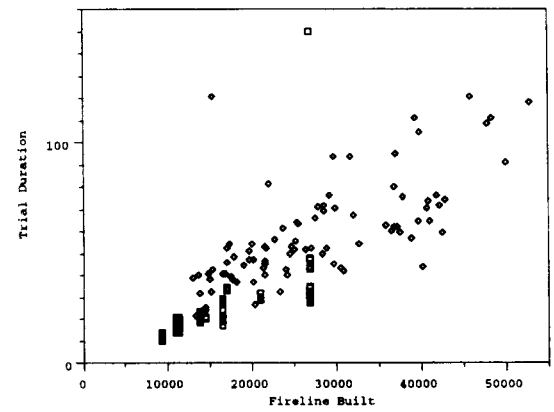

Fig. 4. Vertical clusters.

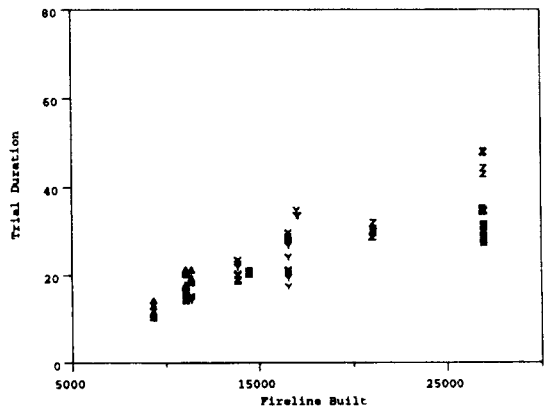

Fig. 6. Vertical clusters colored by wind speed.

procedural knowledge, three defining characteristics of planning [14]. 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. These plans 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 us 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. 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.

AIDE is designed to support mixed-initiative interaction. In a mixed-initiative system, the user and the machine both contribute to a problem solution – formulation, development, analysis, repair – without the need for a constant exchange of explicit instructions [3]. 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 interpret these criteria in 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.

4. Evaluation

Evaluation focused on a simple hypothesis:

Hypothesis 1. Exploration is more effective with AIDE than without.

We also considered two subsidiary issues related to the human element in exploration. One danger in increasing the autonomy of a system is that its less predictable behavior may cause users to mistrust its results. In contrast, an ideal assistant would cause users to be more confident in results generated with its help. Phrased as a hypothesis,

Hypothesis 2. AIDE's assistance improves user confidence in results.

In our evaluation this turned out not to be the case, with one suggestive exception. Fortunately, however, neither did the converse hold: AIDE's participation did not reduce confidence.

Much of the literature in statistical expert systems (and collaborative systems in general) holds that an important factor in successful exploration is the contextual knowledge, or knowledge of what the data mean, that a user brings to bear [12,15,24]. Our experiment also attempted to quantify this factor and determine its influence on exploration. In other words, we wished to show that

Hypothesis 3. The presence of contextual knowledge influences the effectiveness of exploration.

A finding in either direction would prove enlightening: perhaps a statistical assistant can compensate for a lack of contextual knowledge, giving better performance for situations in which a user knows little about the data under consideration; alternatively, it might be more effective in augmenting existing contextual knowledge. Our results were not conclusive, but were suggestive of the latter case.

4.1. Experimental design

The experiment involved testing subjects under two conditions. In the USER + AIDE condition, subjects explored a dataset with AIDE's help, while in the USERALONE 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 confound 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 USERALONE condition. The interface was identical in both cases, lacking only AIDE functionality in the USERALONE 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).

It is important to note that in no sense was AIDE tuned to the specific patterns in these datasets. The system developers had no role in building the data generator and producing the datasets, and there was no contact with the data before the experiment with AIDE began.

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, $\bar{p}$, is the mean number of correct observations made,

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

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, $\bar{p}$ for a given subject gives the probability that one of his or her observations is correct.

The $\bar{p}$ 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 $\bar{p}$ of 0.5 for a large number of observations and another subject with the same $\bar{p}$ score for many fewer observations. Another measure, which we call

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

	$\bar{p}$		$k\bar{p}$		$\bar{i}$		$k\bar{i}$	
	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

$k\bar{p}$ for consistency, considers number of correct observations alone:

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

We will 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, $\bar{i}$ and $k\bar{i}$, which are comparable to $\bar{p}$ and $k\bar{p}$ but call an observation 'correct' simply if a direct relationship is identified, ignoring its descriptive form (linear, cluster, nonlinear, etc.)

4.2. Results

Subject performance is shown in Table 1. A matched-pair, one-tailed t-test tells us that $\bar{p}$ and $k\bar{p}$ 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 $\bar{i}$ and $k\bar{i}$.

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 USERALONE 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 USERALONE condition. Of all the correct suggestions AIDE made about each dataset, only one was not also tried by subjects in the USERALONE condition.

Let us move to the second hypothesis. How does AIDE affect the confidence of subjects in the results they produce? If we combine confidence values (taking 'high' as 1, 'low' as 0) for all observations made by each subject, we arrive at a measure of the confidence of a subject during each condition. The mean confidence C_M of subjects in the USER + AIDE condition ($C_M = 0.599$) turns out to be not significantly different from that of subjects in the USERALONE condition ($C_M = 0.628$). This raises an obvious question of whether subjects have different confidence in observations that turn out to be correct than they do for incorrect observations. In fact, this is an important point: we are happy if a system makes subjects confident in their activities, but not if their results turn out to be consistently wrong. The results are shown in Table 2. When we break the dataset down into correct and incorrect observations, both for correct description ($\bar{p}$) and correct identification ($\bar{i}$), we find that confidence is higher for correct observations than for incorrect ones. The general pattern is the same for both measures of performance, with one suggestive exception. Confidence levels for correct observations are about the same in both conditions, but for incorrect observations (measured by $\bar{i}$) confidence levels are lower in the USER + AIDE condition. That is, for one interesting subset of cases subjects have more appropriate confidence levels in the USER + AIDE condition than in the USERALONE condition.

Table 2
C_M per subject, for correct and incorrect observations

	$C_M(\bar{p})$		$C_M(\bar{i})$	
	AIDE	ALONE	AIDE	ALONE
Correct means	0.68	0.69	0.66	0.66
Incorrect means	0.56	0.58	0.48	0.56

These results are equivocal. The good news is that AIDE does not reduce user confidence, a common effect when complex tasks are automated and thus a serious concern for intelligent assistants [21]. On the other hand, we cannot thereby conclude that AIDE has a positive, balancing effect on user confidence; one of the experimental subjects noted that his confidence assessment depended on factors independent of AIDE's autonomous activities.

The third issue concerns the effect of contextual knowledge on performance. An analysis of variance examined the interaction between the condition (USER + AIDE or USER-ALONE) and the presence or absence of contextual information for observations, in the form of meaningful names for variables. We can summarize the analysis by saying that contextual cues in the data were not strong enough to lead subjects directly to correct descriptions (as measured by $\bar{p}$), but nevertheless point in the right direction by drawing attention to those relationships worth pursuing (as shown by $\bar{i}$). This result is suggestive and intuitively plausible.

4.3. Explaining subject performance

Now, the simple fact of the performance difference between the USER + AIDE and USERALONE conditions is not entirely satisfying. We are really most interested in understanding why AIDE works. For a better understanding of AIDE's contribution, we divided subject actions into three types. Some operations are concerned with local decision-making: selecting a variable or constructing a relationship for display, examining indications, or asking the system for documentation of proposed actions. These are what we will call LocalOperations. They involve decision-making at a single focus point: assessing information about which variables and relationships it would be worthwhile to describe, or evaluating the applicability of different operations and procedures to describe a potential pattern. LocalOperations account for 40% of the operations in the USER + AIDE condition. NavigationOperations are such actions as initiating the exploration of a variable or relationship or going back after generating a result to select another relationship. In other words, these operations generate new focus points, or take the exploration from one focus point to another. Navigation is responsible for 44% of the operations in the USER + AIDE condition. Finally, ManipulationOperations are a specific type of navigation operation, involving selection of the reductions, transformations, and decompositions that make changes or additions to the data. Data manipulation accounts for only a small portion of the total number of operations. Table 3 gives a summary of the operations made in each condition for all subjects. Because the distributions are somewhat skewed, the table presents the median and interquartile range as well as the mean and standard deviation.

The difference between the USER + AIDE and USER-ALONE conditions is striking. While local decision-making is the most important factor in the USERALONE condition, navigation dominates in the USER + AIDE condition. We infer that the navigational facility, which relies on an explicit model of the data analysis process, above the level of individual operations, is a factor in improved performance in the USER + AIDE condition. Examining the relationships in more detail, we find that the relationship between NavigationOperations and LocalOperations is relatively strong ($r = 0.67$), as is the relationship between NavigationOperations and ManipulationOperations ($r = 0.54$). The variables ManipulationOperations and LocalOperations are weakly correlated to begin with ($r = 0.29$), and if we hold NavigationOperations constant the correlation drops to 0.12. Exploration of these relationships shows no unusual patterns. In relating these factors to our performance measurements, it appears that explicit data manipulation accounts for relatively little of the total effort a subject puts into the exploration, but is one of the strongest factors in determining performance. Navigation is the other important factor. A plausible explanation is that data manipulation operations are generally applied only when one perceives some kind of pattern. Data manipulation operations generally provide a more detailed view of a pattern, and thus a greater number of these operations leads to more accurate observations.

Though our understanding of these factors remains tentative, they 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

Table 3
Summary of operations, averaged over all subjects

Command	Condition	% Total	Mean	SD	Median	IQR
TotalOperations	USER + AIDE		331	158	361	297
	USERALONE		191	83	180	144
LocalOperations	USER + AIDE	38%	127	84	118	134
	USERALONE	73%	140	81	143	122
NavigationOperations	USER + AIDE	44%	146	73	137	14
	USERALONE	12%	22	5	21	9
ManipulationOperations	USER + AIDE	13%	44	37	28	65
	USERALONE	9%	17	21	9	24

and actions to another. This point was also emphasized by most of the subjects in their 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.

5. Discussion

The experimental results raise several further questions. First, though some effects were large enough to be easily seen in the small sample size (forced on us by time and resource constraints), one might hope for experiments on a larger scale to make more headway in investigating into user confidence and context. Second, our artificial datasets were constructed to reflect realistic, common patterns, and provided a necessary experimental control. As a follow-up question, one might ask how AIDE would perform on other datasets that contain patterns and relationships not present in the test data. Because AIDE's strategies were developed to handle patterns in a variety of datasets, we are confident in AIDE's robustness. A third point is also related to the use of artificial data: are AIDE's strategies up to the requirements of real world problems? In informal testing on real datasets, we found AIDE to be helpful for specific types of patterns. In general, these questions require further development and experimentation for clear answers.

Despite these caveats, we learned some important lessons in developing and evaluating AIDE. Our findings are largely consistent with other reviews of statistical expert system development [19,11,25,20,7,18], but we also identify a few fresh directions for research. We will begin with the research questions we addressed.

Planning is a practical means of supporting the data analysis 'process'. Note the inclusion of the word 'process'. Data analysis is different from, for example, word processing and batch programming: the correctness of the end product cannot be checked without inspecting the path leading to it [13]. A great deal of work in statistical strategy takes this view. The most prominent example is probably Gale and Pregibon's REX system, which implemented a strategy for linear regression [6]. REX' actions were determined by the traversal of a decision tree; the tree provides an explicit representation of the sequential, coherent decision-making process. In contrast, conventional software for data analysis focuses on powerful individual operations, or a comfortable statistical programming environment, but provides little support for the structured organization of these operations and procedures.

AIDE supports the data analysis process more directly. Imagine in the course of exploring a dataset you decide to build a linear model of a set of variables. During the process you notice an unusual pattern of clusters in a subset of the data, and you suspend your modeling to follow this tangent. When you are finished, you return to the point at which you broke off, to continue with the model. By maintaining an explicit representation of the exploration *process*, in addition to its individual actions, AIDE can support this kind of navigation. AIDE furthermore helps to reorient the user in making such shifts in attention, by presenting the chain of decisions leading to a given point, displaying relevant data, making appropriate suggestions – in general, helping to restore context, as far as possible given the built-in limitation of the system's knowledge.

Shared control is a key aspect of effective assistance. AIDE takes the perspective that human involvement is an essential part of the exploratory process. A completely autonomous system can have little notion of the significance of its findings – but this is exactly the kind of knowledge that informs the selection of data, analysis methods, and evaluation techniques. This point has also been made in the knowledge discovery in databases literature, Brachman et al.'s IMACS, which is aimed at the task of 'data archaeology,' being a good example [2]. 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.

AIDE concentrates explicitly on this balance of autonomy and accommodation. As a mixed-initiative planner [1], AIDE *assists* in an exploration, rather than taking it over completely or waiting for instructions for each of its moves. This approach has several benefits, one of the most important being its flexibility. For example, a mixed-initiative system can potentially be acceptable to both novices and experts. A common problem faced by an intelligent assistant – in fact, by most user interfaces – is that providing comprehensive guidance and support for novice users can actively impede expert users. Conversely, building systems to support experts may entail an enormous learning curve for novices. In AIDE's mixed-initiative design, the system offers advice and analysis paths which may be helpful for novice users, but its decisions can be overridden at almost any point by an expert user. The interaction is not perfect for an expert user. For example, AIDE may consider decisions in a different order from the expert, who will have to guide the analysis at each step, occasionally rolling the analysis back to an earlier state if AIDE jumps ahead. Nevertheless, this kind of interaction has been made as easy as possible, for just such cases.

Maintaining context can be difficult problem in a mixed-initiative system. Sharing control is not without pitfalls. Whenever AIDE takes control of the analysis, it runs the risk of losing the user. This problem applies to many domains other than statistical analysis; in interaction with hypertext systems, for example, it is called the 'lost in

hyperspace' feeling [16].[1] This is a basic human–computer interaction concern: the system should provide the user with implicit answers to the questions: "Where am I?", "How did I get here?", "What can I do here?" and "Where can I go from here?" [17]. These are exactly the issues addressed by AIDE's navigation facilities.

Context is one of the central concerns of work in collaborative systems, where collaboration is a process in which two or more agents work together to achieve shared goals [9]. Loren Terveen has identified a set of issues that must be addressed by any system that collaborates in an intelligent way with its users [24]: 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 points, 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.

Acknowledgements

This research was supported by ARPA/Rome Laboratory under contract No. F30602-93-0100 and by the Department of the Army, Army Research Office, under contract No. DAAH04-95-1-0466. The US 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 US Government.

References

[1] J.F. Allen, Mixed initiative planning: Position paper (WWW document), Presented at the ARPA, Rome Labs Plannining Initiative, 1994, Available at: http://www.cs.rochester.edu/research/trains/mip/.

[2] R.J. Brachman, P.G. Selfridge, L.G. Terveen, B. Altman, A. Borgida, F. Halper, Th. Kirk, A. Lazar, D.L. McGuinnes, L. Alperin Resnick, Integrated support for data archeology, International Journal of Intelligent and Cooperative Information Systems, 1993.

[3] M.H. Burstein, D.W McDermott, Issues in the devlopement of human–computer mixed initiative planning, in: B. Gorayska, J.L. Mey (Eds.), Cognitive Technology: In Search of a Human Interface, Elsevier Science, 1996, pp. 285–303.

[4] P.R. Cohen, Empirical Methods for Artificial Intelligence, MIT Press, 1995.

[5] P.R. Cohen, M.L. Greenberg, D.M. Hart, A.E. Howe, Trial by fire: Understanding the design requirements for agents in complex environments, AI Magazine 10 (3) (1989) 32–48.

[6] W.A. Gale, REX review, in: W.A. Gale (Ed.), Artificial Intelligence and Statistics I, Addison-Wesley, 1986.

[7] W.A. Gale, D.J. Hand, A.E. Kelly, Statistical applications of artificial intelligence, in: C.R. Rao (Ed.), Handbook of Statistics, vol. 9, Elsevier Science, 1993, pp. 535–576.

[8] M.P. Georgeff, A.L. Lansky, Procedural knowledge, IEEE Special Issue on Knowledge Representation 74 (10) (1986) 1383–1398.

[9] B. Grosz, Colaborative systems, AI Magazine 17 (2) (1996).

[10] D.J. Hand, A statistical knowledge enhancement system, Journal of the Royal Statistical Society A 150 (1987) 334–345.

[11] D.J. Hand, Patterns in statistical strategy, in: W.A. Gale (Ed.), Artificial Intelligence and Statistics I, Addison-Wesly, 1986, pp. 355–387.

[12] P.J. Huber, Data Analysis implications for command language design, in: K. Hopper, A.I. Newman (Eds.), Foundation for Human–Computer Communication, Elsevier Science, 1986.

[13] P.J. Huber, Languages for statistics and data analysis, in: P. Dirschedl, R. Ostermann (Eds.), Computational Statistics, Springer-Verlag, 1994.

[14] R.E. Korf, Planning as search: A quantitative approach, Artificial Intelligence 33 (1987) 65–88.

[15] D. Lubinsky, D. Pregibon, Data analysis as search, Journal of Econometrics 38 (1988) 247–268.

[16] J. Nielsen, Hypertext and Hypermedia, Academic Press, 1990.

[17] J. Nievergelt, J. Weydert, Sites, modes and trails: Telling the user of an interactive system where he is, what he can do, and how to get to places, in: R.M. Baecker, W.A.S. Buxton (Eds.), Readings in Human–Computer Interaction: A Multidisciplinary Approach, Morgan Kauffmann, 1987, pp. 438–441.

[18] C.M. O'Brien, Are there any lessons to be learnt from the building of glimpses?, in: D.J. Hand (Ed.), AI and Computing Power, Chapman and Hall, 1994, pp. 53–62.

[19] D. Pregibon, A DIY guide to statistical strategy, in: W.A. Gale (Ed.), Artificial Intelligence and Statistics I, Addison-Wesly, 1986, pp. 389–399.

[20] D. Pregibon, Incorporating statistical expertise into data analysis software, National Research Council, National Academy Press, 1991, pp. 51–62.

[21] W.B. Rouse, N.D. Geddes, R.E. Curry, An architecture for intelligent interfaces: Outline of an approach to supporting of complex systems, Human–Computer Interaction 3 (1987) 87–122.

[22] R. St. Amant, P.R. Cohen, Control Representation in an EDA assistant, in: D. Fisher, H. Lenz (Eds.), Learning from Data: AI and Statistics V, Springer-Verlag, 1996, pp. 353–362.

[23] R. St. Amant, P.R. Cohen, A Planner for exploratory data analysis, in: Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems, AAAI Press, 1996, pp. 205–212.

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

[25] J. Tukey, An Alphabet for statisticians' expert systems, in: W.A. Gale (Ed.), Artificial Intelligence and Statistics I, Addison-Wesly, 1986, pp. 401–409.

[1] Hand's knowledge enhancement system, KENS, let users browse through a network structure of statistical concepts containing over 200 nodes [10]. KENS is similar in some ways to hypertext systems, but users of KENS had no problems orienting themselves – a surprising and significant result.