Toward the Integration of Exploration and Modeling in a Planning Framework

Robert St. Amant and Paul R. Cohen

Computer Science Technical Report 94-41

Experimental Knowledge Systems Laboratory
Computer Science Department, Box 34610
Lederle Graduate Research Center
University of Massachusetts
Amherst, MA 01003-4610

Abstract

Statistical operations are often facilitated by other operations. We can facilitate modeling operations by testing their input for irregularities and removing problems wherever possible. A planning representation is well-suited to this task. We describe the representation used in Igor, a system for exploratory data analysis, and its integration with two modeling systems, Pearl's IC and Cohen's FBD. We show that introducing outliers into the inputs of the algorithms can influence their performance. We demonstrate that a planning representation offers a flexible way of integrating outlier detection and removal into the modeling process, taking account of specific characteristics of the modeling operations involved.

This work is supported by ARPA/Rome Laboratory under contract #'s F30602-91-C-0076 and F30602-93-C-0100. To appear in *Proceedings of AAAI-94 Workshop on Knowledge Discovery in Databases.*

1 Introduction

The techniques of exploratory data analysis (EDA) rely on two general strategies in exploring data: one generates simplifying descriptions of data, the other extends and refines surface descriptions of data. EDA techniques simplify data by constructing partial descriptions and models that capture particular characteristics of the data. They make descriptions more effective by looking beyond surface descriptions at what is left unexplained. Exploratory strategies generate increasingly detailed, complementary descriptions of data.

Causal modeling is one approach to describing data. Like many other statistical procedures, causal modeling algorithms often make strong assumptions about properties of their input data. We can facilitate modeling operations by testing their input for irregularities (e.g., nonlinearity, outliers) and removing problems wherever possible. In general, we apply transformations to the input of an operation to ensure that it corresponds to the requirements of the operation and to improve the quality of its results.

Igor is a knowledge-based system designed for exploratory statistical analysis of complex systems and environments [1]. Igor uses a script-based planning representation to guide application of statistical operations. In Igor models of data are built incrementally and opportunistically, relying on information acquired during the exploration and modeling process.

In this paper we describe a limited integration of exploration and modeling in Igor. We focus on a specific exploration operation, the detection of outliers in linear relationships, and two modeling algorithms, Pearl's IC [11] and Cohen's FBD [3]. We demonstrate that a planning representation offers a promising way of integrating exploration into the modeling process. Though the integration is limited in terms of the range of exploration and modeling operations performed, it shows that the combination can be profitable: modeling operations can provide context for exploratory actions; exploration can test assumptions made by the modeling algorithm.

2 Motivation

A central element of some well-known causal modeling algorithms is the notion of conditional independence [11]. If X, Y, and Z are disjoint sets of variables, then X and Y are said to be conditionally independent given Z, iff

$$I(X,Z,Y) \text{ iff } P(x,y|z) = P(x|z)P(y|z).$$

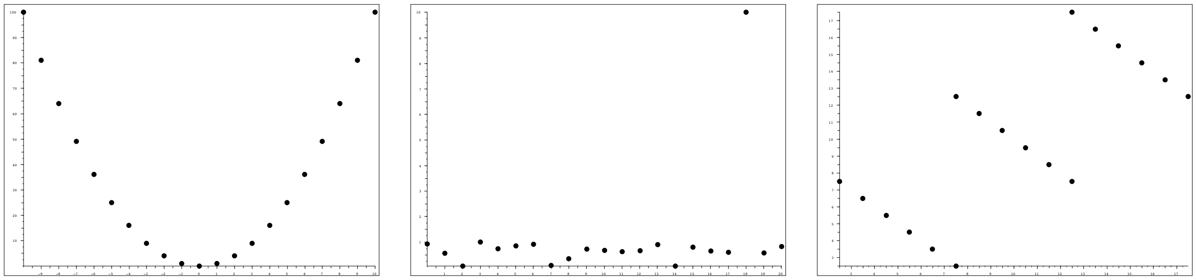
Informally, if holding Z constant renders X and Y independent, then there can be no direct influence between X and Y.

Conditional independence is defined on probabilities, or probability distributions. The partial correlation statistic provides a straightforward way to operationalize the test:

$$\mathit{Partial}(X,Y|Z) < \mathit{Threshold} \rightarrow I(X,Z,Y).$$

If the partial correlation of X and Y with Z held constant falls below a threshold, then X and Y are conditionally independent. Thus some implementations of causal modeling algorithms take a covariance matrix as input and base their conditional independence inferences on the partial correlations derived from the matrix.

Recall that the correlation coefficient (and by extension the partial correlation coefficient) measures the degree of *linear* association between variables. Consider the three cases in Figure 1. In Figure 1(a) the correlation of Y and X is zero, but clearly after an appropriate transformation



(a) Quadratic

(b) Outlier

(c) Partitions

Figure 1: Examples

the correlation would be perfect. In Figure 1(b) the correlation between Y and X is moderate to high, but would be near zero after filtering the single outlier. Figure 1(c) shows a strong relationship between Y and X, but the positive correlation would change to a negative correlation after partitioning on a third variable.

Each of these cases demonstrates that the correlation statistic may not capture the desired relationship between two variables. If a modeling algorithm uses partial correlations to calculate conditional independence, then the validity of its results depends on properties of the input variables–in particular, that these sorts of irregularities are not present in the data, either obscuring existing relationships or inducing spurious relationships.

This discussion should not be taken as criticism of the modeling algorithms; these considerations lie outside their domain. It is rather recognition of the limits of their applicability. By integrating data exploration with modeling operations we extend these limits.

3 Igor

Elsewhere [1] we draw an analogy between the process of EDA and planning. Briefly:

- Exploratory strategies are plans consisting of sequences of statistical operations; these operations are actions that transform data relationships.
- As in planning, primitive exploratory operations can be combined in different ways for different effects. For example, in considering a relationship between two variables, it makes a difference whether we remove outliers before or after applying a transformation to the relationship.
- Conversely, abstract statistical operations often decompose naturally into more primitive operations, just as in hierarchical plan decomposition. For example, the abstract operation of fitting a robust line to a relationship may expand to partitioning the relationship, calculating medians, and combining the results.
- Selection of the most effective exploratory strategy is akin to selection of an appropriate plan to satisfy a given goal. We must often evaluate different paths to find the most effective one.
- Just as plans fail and require repair, an exploratory operation may require iteration for adequate results. Retrying an operation is analogous to retrying an action as a part of plan failure recovery. Selecting a different, more promising strategy corresponds to replanning.

Igor uses a script-based planning representation, based on the RESUN signal interpretation system [2], to guide application of statistical operations. In Igor, sequences of simple operations are combined for complex effect. Results are derived incrementally and opportunistically, based on constructing and revising plans according to information acquired during the process.

```
(define-plan generate-standardized-residuals-plan
  :goal        (standardized-residuals
                    ?y-variable ?x-variable ?residuals)
  :input       (y-variable x-variable)
  :internal    (slope intercept)
  :output      (residuals)
  :grammar     (:sequence
                 simple-linear-regression
                 generate-residuals
                 standardize))
```

Figure 2: Example Script

The data structures manipulated at the lowest level in the planning representation are frames. A variable is a simple frame; a linear relationship between two variables is a slightly more complex hierarchy of frames; an annotated causal model is a highly interconnected hierarchy of frames. We call frames and hierarchies of all types structures.

The primitive operations provided by the representation are called actions. An action is a data transformation or decomposition of an input structure to an output structure. A log transform is a simple example of an action; it applies a log function to each element in a sequence and collects the results. More complex transformations include smoothing, outlier removal, and fit operations. Each action has an associated goal form and may be triggered by the establishment of a matching goal.

Actions are combined in scripts. A script is a sequence of subgoals whose associated actions transform one structure into a more concise, better parametrized, more descriptive structure. Scripts, like actions, have associated goal forms, and thus may be combined hierarchically to satisfy the goals of other scripts. Combination of subgoals in a script is governed by the specification of the script. A script specification defines how its subgoals must be satisfied in order for the top level goal to be satisfied. Specification constructs allow sequential combination of subgoals, iteration over sets of subgoals, conditionalization on tests of variable values, and activation of subgoals in parallel.

In the example script in Figure 2, the :sequence directive orders the goals simple-linear-regression, generate-residuals, and standardize. The result of this script is a sequence containing the standardized residuals from a linear regression between two variables.

Scripts and actions control procedural execution in the representation, managed flexibly by goal establishment. These constructs still do not provide the degree of opportunism and context-specific control we associate with exploration, however. For this we rely on two mechanisms that depend on context, monitoring and focusing.

A strictly goal-driven system can find it difficult to take new structures under consideration during the search process. A monitor is a goal, active in parallel with the execution of a script, and matching scripts, which test intermediate results produced. Monitors evaluate the ’interestingness’ of results, taking context information into account, to initiate new directions in the exploration.

Focusing heuristics guide and constrain the exploration process based on local context information. Focusing heuristics are activated whenever there is a choice between which goals to pursue and which scripts to apply; they evaluate the precedence of active goals and the relevance of matching scripts when deciding which scripts should be activated and which ignored. We use focusing heuristics to evaluate the cost of pursuing particular search paths. A focusing heuristic is free to prune the goals or scripts it takes as input, temporarily or permanently. As with monitors, a focusing heuristic may take advantage of domain-specific knowledge in its processing.

4 Integration and Evaluation

We begin by describing the datasets we used for the evaluation. For each algorithm we then describe

- the modeling algorithm itself,
- outlier detection techniques suited to its operation,
- the plan form of the integrated operations,
- how the integration affects performance.

We generated 60 sets of linear structural equations of varying size. For each set of equations we generated a dataset of 50 tuples. Exogenous variables were sampled from normal distributions. Endogenous variables were derived according to the structural equations. Error was added to each variable, sampled from independent normal distributions. By following these procedures we ensured that the datasets we generated accurately reflected the structural equations.

We then perturbed each dataset by adding a single outlier to the 50 tuples. Here an outlier is a tuple in which each element has been independently sampled from a normal distribution. We generated outliers at three standard deviation settings: 2.0, 3.0, and 4.0. As this value increases, we say that the perturbation level of the dataset increases. In the real world such outliers might plausibly be attributed to measurement error or anomalous experimental conditions.

Our evaluation considered two algorithms, Pearl's IC and Cohen's FBD. We ran the same procedure on both algorithms. We first applied the algorithm to each original dataset to generate a *nominal model*. We then generated models for each of the perturbation levels and evaluated the *outlier-data models* with respect to the nominal model. Note that we are not evaluating the algorithms with respect to some true model, but rather only how they are affected by the outliers.

We then integrated the modeling algorithm with specific outlier detection operations in Igor. We evaluated the results, the *filtered-data models*, with respect to the nominal and outlier-data models.

4.1 IC

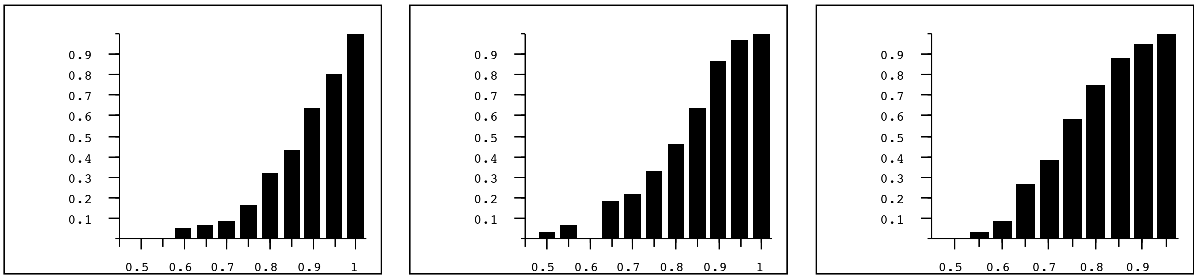
The IC algorithm generates a causal model in the form of a DAG in which nodes represent variables, edges causal influences. Links are identified as *marked unidirectional*, indicating genuine causation, *unmarked unidirectional*, indicating potential causation, *bidirectional*, indicating spurious association, and *undirected*, indicating an undetermined relationship. Two models are said to be identical if they have the same links (edges without regard to direction) and the same uncoupled head-to-head nodes (converging arrows emanating from non-adjacent nodes, such as $a \rightarrow c \leftarrow b$.) A much more complete description is given in [11].

In our evaluation we evaluated the effects of perturbation on a model M by means of

- links in M and the nominal model,
- uncoupled head-to-head nodes in M and the nominal model, and
- edges denoted genuine in M and the nominal model.

Running IC on the perturbed datasets[1] we produced models which we summarize in Figure 3. The histograms show the cumulative distribution of the ratio of correct links in a model with

[1] We fixed the separating set size at 1, the partial threshold at 0.1



(a) Perturbation 2.0 (b) Perturbation 3.0 (c) Perturbation 4.0

Figure 3: IC: Effect of perturbation on correct link ratio

respect to the nominal model. In other words, we counted the number of links shared between each model and its nominal model and divided by the total number of links in the nominal model, giving us one number per model. Thus each bin in the histogram measures the proportion of models with a correct link ratio equal to or lower than the value on the x-axis.

Ideally each distribution should be as right skewed as possible, with all cases achieving high correct link ratios. We see by the clustering around 1.0 on the x-axis that the outlier-data models do tend to be very similar to the nominal models. As the perturbation grows greater, however, the central location of the distribution decreases while spread increases–outlier-data models are found farther and farther away from their nominal models. The distributions for other measures behave similarly.

We can redress these problems by identifying the outliers in the data. The detection procedure treats variables pairwise. It runs a linear regression and examines the residuals, looking for those cases which exert undue leverage on the linear relationship between the variables. Such leverage is indicated by outliers in the standardized residuals. The normal test counts an element (x, y) as an outlier in the relationship (X, Y) if its standardized residual falls outside $[-2.7, +2.7]$. The fourth-spread test [7] (similar to a quantile test) counts an an element (x, y) as an outlier if its residual falls more than $3/2d_F$ below or above the fourth spread boundaries, where d_F measures the spread itself. These tests give similar results for our data.

The combination of outlier removal and model construction is managed by incorporating the call to the IC algorithm in a script, generate-model-plan-A. We make the input exploration and transformation phase explicit by satisfying the goal explore-input before proceeding with model construction. The goal explore-input is not specific to outlier detection, so that other exploration operations also apply. Note that the goal representation allows us to use either algorithm, IC or FBD, in the model building phase, the selection managed by a simple focusing heuristic.

```
(define-plan generate-model-plan-A
  :goal     (top-level-build-model
               ?domain ?variables ?relationships ?context)
  :input    (domain variables context)
  :output   (relationships)
  :grammar  (:sequence
              explore-input
              build-model))

(define-plan explore-remove-outliers-plan
```

Condition	Pert	Correct% (N)	W/C	C-ratio	G-Same	G-Diff
Outlier	2.0	0.87 (31.97)	0.22	0.73	0.40	0.60
Filtered		0.80 (28.78)	0.29	0.57	0.42	0.80
Outlier	3.0	0.81 (29.37)	0.29	0.59	0.31	0.66
Filtered		0.81 (29.47)	0.26	0.60	0.48	0.64
Outlier	4.0	0.75 (27.63)	0.38	0.44	0.19	1.40
Filtered		0.82 (29.55)	0.24	0.62	0.46	0.72

Table 1: IC: Degradation by perturbation level

```
:goal     (explore-input ?variables ?transformed-variables)
:input    (x-variable y-variable)
:internal (relationships)
:output   (outliers)
:grammar  (:sequence
            generate-relationships
            (:in-parallel (relationships) mark-outliers)
            transpose-variables
            remove-marked-tuples
            transpose-tuples))
```

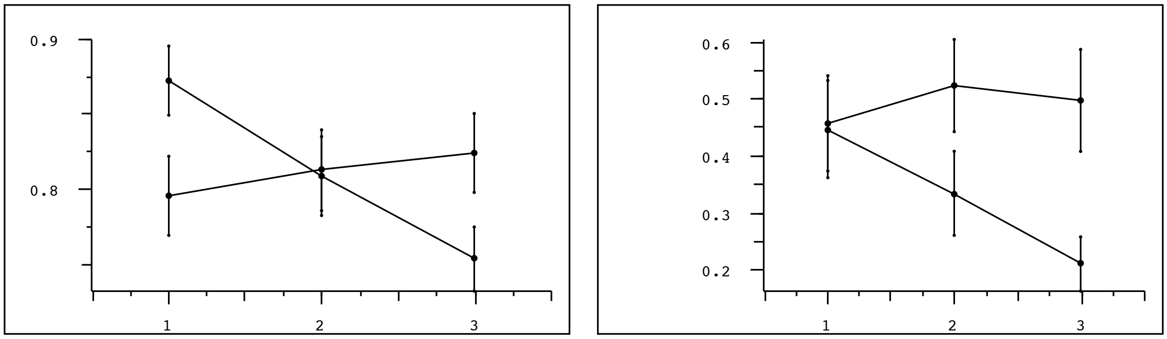
In this arrangement control is of the form *ExamineInput* → *ConstructModel*. Igor acts strictly as a preprocessor for the IC algorithm, in that outlier detection is managed by a test phase entirely before the modeling begins. In effect we consider exploration a simple extension of the modeling operation.

Summaries of our evaluation measures are shown in Table 1. Each pair of rows contains measurements of outlier-data and filtered-data models for each level of perturbation. Correct% is the mean percentage of correct links, per model, with the total number of correct links in parentheses. W/C is the ratio of correct links to incorrect links. C-ratio measures the overlap in uncoupled head-to-head nodes between a outlier-data and filtered-data model. G-Same and G-Diff measure the overlap and difference between genuine edges in the models.

In Figure 4 we see a more detailed comparison for Correct% (a) and G-Same (b). Along the x-axis we have increasing perturbation level, along the y-axis the appropriate measurement level. The lines correspond to the measures for outlier-data models and filtered-data models, with .90 confidence intervals. In both cases the downward sloping line belongs to the outlier-data models. There is a clear degradation for outlier-data models as perturbation increases, while the filtered-data models remain relatively insensitive. These results are roughly the same for the other measures as well.

At the initial perturbation level, the outlier-data models are unexpectedly closer to the nominal models than the filtered-data models, sometimes even outperforming the filtered-data models. We attribute this to the small size of the datasets: single outliers, both those we have introduced and those potentially already present in the data, can have a relatively large effect on the correlation between variables.[2] Thus removing small numbers of outliers may induce new relationships or

[2] This indicates that we should examine larger datasets as well. It does not lead us to dismiss these preliminary results. A larger sample size naturally dilutes the effect of a single outlier, but large single outliers or groups of smaller outliers can still influence statistical calculations. It also may be that larger samples are simply not available.



(a) correct link ratio

(b) genuine edge ratio

Figure 4: IC: Effect of perturbation

remove existing ones. As perturbation increases, however, outliers become easier to distinguish. Outlier detection improves all measurements with further increase in perturbation level.

4.2 FBD

The FBD algorithm, in addition to building qualitative causal descriptions of relationships, estimates the strength of relationships. FBD constructs a model iteratively: at each stage FBD considers a single variable and a set of candidate predictors for it. Filters are applied sequentially to prune the predictor set. When the filtering is completed, the algorithm selects another variable to predict. Here the essential aspects are that (1) FBD proceeds incrementally and (2) FBD's decisions are dependent upon the same assumptions made by linear regression.

We can measure how much effect perturbations have on a model M by measuring

- links in M and the nominal model,
- the difference in predictive power in M and the nominal model (ΔR^2),
- the difference in estimated correlations in M and the nominal model (Δr).

If we perform the data exploration/transformation process entirely before running FBD, as we did with the IC algorithm, we see improvement in most measures. We can better the results with a closer integration of model-building and exploration.

In the IC example we detect outliers with respect to linear relationships between pairs of variables. One of our considerations in choosing the detection mechanism was the nature of the partial correlation statistic. In general the notion of "outlier" depends on the operation to be performed. In the case of FBD we can detect outliers by examining the effect of removing presumed outliers on prediction parameters of the partially constructed model.

The diagnostic measure v_{ii} (also known as h_{ii}) can be used to detect outliers in the predictors in a regression. v_{ii} is the diagonal of the matrix V, where

$$V = X(X^TX)^{-1}X^T \text{ and } Y - \hat{Y} = (I - V)Y.$$

Each row of V corresponds to a tuple in the dataset. The matrix V is often called the hat-matrix, because it gives us a way of calculating $\hat{Y}$, the predicted value of the dependent variable, from Y, the actual value. A larger entry in the diagonal of V indicates a larger contribution of that

In any case, exploration is required to test our assumptions.

tuple to the predicted value of the dependent variable, and thus greater influence on the regression model. (A full discussion of v_{ii} is beyond the scope of this paper, but see [4].) We use v_{ii} to identify high leverage cases, those most likely to bias the predictors of a variable in a model.

Rather than running the test on the entire dataset before running FBD, as we did with IC, we can take advantage of FBD's incremental construction. We can apply the v_{ii} test most effectively by including only legal predictors of a given variable, as determined by the modeling filters, rather than all variables in the dataset. Thus during the predictor selection process the v_{ii} test checks the set of candidate predictors to see whether any cases might exert undue influence on the regression. If so, these cases are removed and FBD continues with its selection. The procedure is repeated for each predicted variable.

To do this we reimplement the top level control of the FBD algorithm in plan form. Two loops are involved, the outer loop over predicted variables, the inner loop over filters on each variable's potential predictors. The :iterate directive continues until its condition clause evaluates to nil. Focusing heuristics can control the order in which predicted variables and filters are selected, but currently they are selected in a fixed order. We manage outlier detection and removal by a defining a new filter and letting it execute with the others. When the inner loop completes, the variable is considered predicted and the next variable is selected. Incremental construction gives us a completed model when the plan is finished. Again the goal representation for filters and model building makes few assumptions about the particular modeling algorithm or heuristics involved.

```
(define-plan generate-model-plan-B
  :goal     (top-level-build-model
               ?domain ?variables ?relationships ?context)
  :input    (domain variables context)
  :internal (filter predicted-variable)
  :output      (relationships)
  :grammar     (:iterate ()
                         (setf predicted-variable
                               (select-predicted-variable))
                (:sequence
                 (:iterate ()
                           (setf filter
                                 (select-filter))
                  apply-filter)
                 incremental-build-model))
```

In this arrangement we see a different view of the process: we no longer have a monolithic outlier removal operation preceding an atomic modeling operation, but rather an incremental *Transform* → *Construct*...*Transform* → *Construct*.... The outlier detection function, implemented as a filter, merges seamlessly into the model construction process.

Application of outlier removal to FBD gives results similar to those for IC. The summarized results are shown in Table 2. The Condition column specifies whether the outlier-data model was used (Outlier), the model built using the pairwise detection algorithm (Preprocessed), or the model built interleaving construction with the v_{ii} test (Interleaved). W/C is again the ratio of wrong links to correct links. ΔR^2 measures the difference between the R^2 calculated for the model and the nominal model, per predicted variable. Δr is similarly the difference in correlation between the model and the nominal model. For both measures, lower values are better.

Figure 5 shows a more detailed comparison between the three conditions for the Correct% measure. Along the x-axis we have increasing perturbation level, along the y-axis the measurement

Condition	Pert	Correct% (N)	W/C	ΔR^2	Δr
Outlier	2.0	0.75 (13.12)	0.47	0.09	11.81
Preprocessed		0.72 (12.42)	0.53	0.09	12.88
Interleaved		0.74 (13.15)	0.47	0.08	11.24
Outlier	3.0	0.73 (12.98)	0.47	0.11	13.28
Preprocessed		0.76 (13.08)	0.45	0.09	15.72
Interleaved		0.78 (13.68)	0.38	0.09	11.53
Outlier	4.0	0.65 (11.58)	0.64	0.13	14.78
Preprocessed		0.76 (13.10)	0.44	0.09	13.33
Interleaved		0.78 (13.71)	0.40	0.08	13.09

Table 2: FBD: Degradation by perturbation level

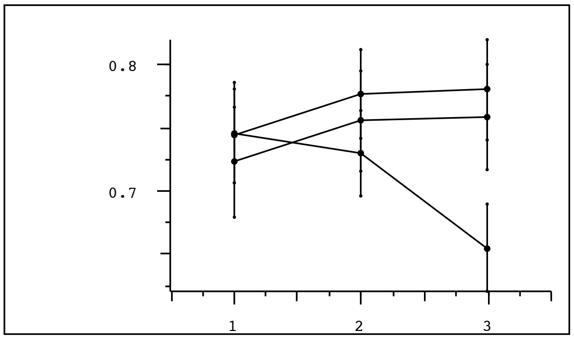


Figure 5: Effect of perturbations on link ratio

level. The lines correspond to the measures for outlier-data models, filtered-data (preprocessed) models, and filtered-data (interleaved) models, with .90 confidence intervals. Here again the downward sloping line belongs to the outlier-data models, and the approximately parallel lines to the filtered-data models. The highest performance is achieved in the interleaved case. In summary, we find that applying outlier detection as a preprocessing phase improves FBD's performance as perturbation increases. Incremental application improves results yet further, and reduces degradation for the low perturbation setting.

5 Conclusions

We have approached the integration of statistical modeling with transformation operations from the viewpoint of exploratory data analysis [5, 8, 12], with strong influences from work in the machine learning and knowledge discovery in databases literature [10]. We have shown preliminary evidence that the integration of exploration and model building can be profitable. We can automate parts of the data exploration phase necessarily associated with application of a modeling algorithm.

The purpose of this work is not to describe well-known statistical techniques, but rather to show how their application can be managed in the planning representation, and how control interleaves exploration and modeling. Lansky has noted that planning meshes well with iterative modeling because both processes are essentially constructive [9]. Igor's planning representation lets us tailor the combination of transformation operations with modeling operations in a way that takes advantage of the characteristics of the operations.

One issue we have not addressed is the selection of facilitation operations for particular modeling operations. Outlier detection/removal is just a single example. Realistically we will transform highly skewed variables, test whether relationships are approximately linear, ensure that cases in each variable are independent, and run a variety of domain-dependent tests as well. While we have mechanisms in place, focusing heuristics, to decide among potentially applicable actions, we have only begun to examine the knowledge required to make the *correct* decisions.

There are larger issues we will address in further work, in particular the notion of statistical strategies [6] in exploration. We hope to incorporate more complex interactions into the exploration and modeling process. Domain specific knowledge plays a strong role in the work of real statisticians; our work has left issues in representation so far unaddressed.

Acknowledgments

This research is supported by ARPA/Rome Laboratory under contract #'s F30602-91-C-0076 and F30602-93-C-0100. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

References

[1] Robert St. Amant and Paul R. Cohen. A planning representation for automated exploratory data analysis. In D. H. Fisher and Wray Buntine, editors, *Knowledge-Based Artificial Intelligence Systems in Aerospace and Industry Proc. SPIE 2244*, 1994.

[2] Norman Carver and Victor Lesser. A planner for the control of problem solving systems. *IEEE Transactions on Systems, Man, and Cybernetics, special issue on Planning, Scheduling, and Control*, 23(6), November 1993.

[3] Paul R. Cohen, Lisa Ballesteros, Dawn Gregory, and Robert St. Amant. Regression can build predictive causal models. 1994.

[4] R. Dennis Cook and Sanford Weisberg. *Residuals and Influence in Regression.* Chapman and Hall, 1982.

[5] A. P. Dempster. Purposes and limitations of data analysis. In G. E. P. Box, T. Leonard, and C.-F. Wu, editors, *Scientific Inference, Data Analysis, and Robustness.* Academic, 1983.

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

[7] David C. Hoaglin, Frederick Mosteller, and John W. Tukey. *Understanding robust and exploratory data analysis.* Wiley, 1983.

[8] David C. Hoaglin, Frederick Mosteller, and John W. Tukey. *Exploring Data Tables, Trends, and Shapes.* Wiley, 1985.

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

[10] Christopher J. Matheus, Philip K. Chan, and Gregory Piatetsky-Shapiro. Systems for knowledge discovery in databases. *IEEE TKDE special issue on Learning and Discovery in Knowledge-Based Databases*, 1993. forthcoming.

[11] J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of Principles of Knowledge Representation and Reasoning*, 1991.

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