The problem of incorrect data values and missing data values is pervasive in the databases used in large knowledge discovery exercises. It is in particular a problem with data collected by humans, such as application forms, crime reports, ecological field data, medical records, and surveys. Even data collection via sensors is subject to occasional malfunction or failure. These data problems tend to grow, not lessen, as databases grow in size. In addition, these data problems may be amplified when data are combined from different sources, due to differences in the set of variables collected, failed matches and non-overlapping time periods.

Unfortunately, almost all tools used for knowledge discovery assume that datasets are complete and that data values have been accurately recorded. While standard data quality control procedures are capable of identifying clearly implausible data values, such as a reported age of 150 years, they almost always fail to identify the 9-year-old grandparent and a myriad of other implausible combinations. Further, eliminating observations with extreme, but possible values on a variable, may drop the most interesting cases. Such observations should only be identified as outliers if their extreme values are not suggested by the pattern of values on other variables.

Standard procedures generally drop observations with a missing value on any of the variables being used in the analysis. Even if data values are missing completely at random dropping these observations can result in a loss in efficiency. Such efficiency losses may be substantial in any analysis when a large number of variables is being used even if the percent of missing data on any particular variable is small. When the presence of a missing value on one variable is related to the values taken by other variables, any estimates developed from using only the complete observations may also be extremely biased.

A number of procedures have been proposed for dealing with missing values. Some, such as assigning the mean of the observed values to an observation with a missing value, are clearly *ad hoc* with bad statistical properties. Others procedures may have good statistical properties but depend critically on assumptions about functional relationships and error distributions. While these assumptions can be checked, the modeling effort required when faced with a large number of variables quickly become unfeasible from either a labor or computational perspective.

This project will develop an automated approach that first sets to missing variable values that are identified as gross outliers in a multivariate sense and then imputes all the missing values. The imputed values will be derived using a procedure having at its core the non-parametric Classification and Regression Tree (CART) algorithm widely used in KDI applications. The CART algorithm already embodies a computationally quick, albeit statistically inefficient, approach to dealing with missing values for possible predictor variables.

A CART tree will be grown for each variable in the database, yielding a set of predicted values for each variable that can be substituted for corresponding missing values. An interesting variant of the procedure is to iterate it in a manner similar to the EM algorithm to take advantage of the greater informational content of past predictions for the missing values. Many of the algorithms advanced in the statistics literature on missing values which are designed to exploit spatial or temporal correlation or to maintain the original variability of the data can be incorporated into our procedure at the level of a CART terminal node.

Although the procedure proposed is computationally infeasible for very large databases on a single processor machine, it can be adapted to run efficiently on a machine with parallel processors since a CART tree can be grown on each processor, thereby removing computational limitations. Much of the proposed work is devoted to that objective. Extensive testing of the properties of different variants of procedure on real and simulated datasets is planned.

TABLE OF CONTENTS

For font size and page formatting specifications, see GPG section II.C.

Section		Total No. of Pages in Section	Page No.* (Optional)*
Cover Sheet (NSF Form 1207 - Submit Page 2 with original proposal only)			
A	Project Summary (not to exceed 1 page)	1	
B	Table of Contents (NSF Form 1359)	1	
C	Project Description (including Results from Prior NSF Support) (not to exceed 15 pages) (Exceed only if allowed by a specific program announcement/solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)	19	
	☐ Please check if Results from Prior NSF Support already have been reported to NSF via the NSF FastLane System, and list the Award Number for that Project	NSF Award No.	
D	References Cited	4	
E	Biographical Sketches (Not to exceed 2 pages each)	8	
F	Summary Budget (NSF Form 1030, including up to 3 pages of budget justification)	30	
G	Current and Pending Support (NSF Form 1239)	4	
H	Facilities, Equipment and Other Resources (NSF Form 1363)	2	
I	Special Information/Supplementary Documentation		
J	Appendix (List below.) (Include only if allowed by a specific program announcement/ solicitation or if approved in advance by the appropriate NSF Assistant Director or designee)		
	Appendix Items:		

*Proposers may select any numbering mechanism for the proposal, however, the entire proposal must be paginated. Complete both columns only if the proposal is numbered consecutively.

BROAD SCALE OUTLIER DETECTION AND MISSING VALUE IMPUTATION: CLEANING AND MENDING DATABASES

I. Introduction

Before knowledge discovery begins on any large database, incorrect values have to be detected and corrected in some fashion and missing values have to be filled in. This problem infects data across all disciplines. Social science and medical databases are riddled with missing data, as are most scientific and environmental databases. Likewise commercial databases built around sales and credit records contain numerous recording errors and many scattered holes. As databases get larger, the outlier and missing values problem, if anything, become more pressing. Combining data from different sources tends to further amplify the problem.

The difficulty for knowledge discovery exercises is that almost all of the major knowledge discovery algorithms are designed for use with these databases to predict future phenomena or to discover important patterns which implicitly assume that the data is clean and that none of it is missing. As one recent book on data mining (Berry and Linoff, 1997) put it: "In the real world, data records are often missing fields, or in the relational parlance, contain 'nulls.' Some data mining tools deal gracefully with missing values, most do not." SPPS, a major vendor of statistical software, has recently produced a white paper entitled: "Missing Data: The Hidden Problem." Most data mining tools do better with respect to inputting data with incorrect values in that they produce results, however, the old adage, garbage in—garbage out, applies.

There has been surprisingly little work on the topic of cleaning and mending large databases for use with knowledge discovery techniques relative to the thousands of papers on the various techniques themselves. One of the reasons for this is the general issue of incorrect and missing values typically treated as one of data quality control (Redman, 1992) rather than as an analysis issue. While it is clearly the case that an implementation of an effective quality control program can greatly reduce the amount of incorrect and missing data, it is unlikely that such a program will ever reduce these two problems to the point where the data analyst will not have to think about them. Another reason is that dealing with incorrect and missing values is often thought of as a data preprocessing issue rather than as a formal analysis issue. Cabena *et al.* (1998) in IBM's Redbook Series volume on data mining note: "During data preprocessing, two of the most common issues are noisy data and missing values." Brachman and Anand (1996) in their review of the data-preprocessing phase observe: "Another truism about real-world KDD is that the client's data virtually always has problems. Data may have been collected in an ad hoc manner, unfilled fields in records will invariably be found, mistakes in data entry may have been made, etc. As a result, a KDD process cannot succeed without a serious effort to 'clean' the data."

A good illustration of the nature of the problem comes in a paper by John Schmitz (1996) presented at the National Research Council's Massive Data Sets Workshop. Schmitz works for Information Resources, a marketing research firm with annual revenues of 350 million dollars, which processes and sells huge amounts of retail store and product level marketing data each week to firms who need to make immediate pricing and supply decisions. Schmitz notes: "The

data [from the stores] is reasonably clean. We spend an enormous amount of effort on QA of data. We [still] do have to clean up a lot of the data. Five to 15 percent of it is missing in any one week. We will infer data for stores which simply don't get their data tapes to us in time." Similar stories with respect to a variety of well-known datasets are easily found. Witte *et al.* (1997) look at the nature of a sizeable number of instances where two or more clinical chemistry tests, that should have obtained comparable results, differed by a large enough amount to alter patient care. For surveys, missing values are often the primary issues; although erroneously entered values are also not uncommon. Income is often the variable with a high percent of missing values. For instance, in the 1988 Survey of Consumer Finances almost 28% of the sample did not report an adjusted gross income, while 20% typically do not report income in the Census Bureau's monthly Current Population Survey. For administrative records, data is often missing for the main variables of interest to secondary data analysis. For instance, in the Fatal Accident Reporting System for vehicles, blood alcohol levels and seat belt usage are frequently not reported.

There is, however, no reason why some of the knowledge discovery approaches developed cannot also be used to deal with multivariate outliers and missing values at the data preprocessing stage. While the general situation is perhaps best described by Riley's comment: "Missing values have been largely ignored in the pattern recognition literature," initial efforts at outlier detection and missing value imputation using KDI approaches do appear to confirm their potential usefulness (*e.g.*, Carson, 1984; Pemmaraju and Mitra, 1993; John, 1995; Wong and Gedeon, 1995; Gupta and Lam, 1996; Guyon, Matic, and Vapnik, 1996, Lakshminarayan, *et al.*, 1996; and Teague and Thomas, 1996). Furthermore, these approaches have achieved considerable success in the related area of interpolation in images and high frequency time series.

What is needed for KDI techniques to perform up to their potential is an effective method for cleaning and mending databases, particularly those which are very large in size with respect to either the number of variables and/or the number of observations. Since databases once constructed are used for multiple purposes with many different techniques, the procedure used to clean and mend the database should be suitable for a wide range of purposes and techniques (Rubin, 1996). No such procedure currently exists. We propose to develop such a procedure that will be universally acceptable for its accuracy and robustness.

II. Current Practices

A very large number of procedures have been proposed for dealing with outliers and missing values. A few of these are in common use. Many of these are *ad hoc* and have little foundation in statistical theory. Formal analysis of these procedures -suggests that most have very bad properties. More appropriate procedures are usually available from the statistics literature (*e.g.* Huber, 1981; Rubin, 1987; Barnett and Lewis, 1994; Gelman, *et al.*, 1995), but are infrequently used. The reason for this lack of usage is straightforward. These procedures are: (a) labor intensive to implement correctly, (b) considerably more computationally intensive and (c) require a greater degree of statistical sophistication than more commonly used approaches. We take up outliers first and then missing values because the usual treatment of outliers, not correctable by recourse to original records or current verification, is to set them to missing values.

Outliers

Outliers may be of two types—simply erroneous values or an eccentric individual instance in the database whose characteristics do not resemble those of the other instances in the data. Either way, outliers are important to detect and study. The frequently implemented approach to dealing with potential outliers is to check whether a variable being used contains "illegal" values. For instance, if sex is coded "F" for female, "M" for male, and "blank" for missing, are there observations which use values other than these three? Finding observations that do have other values raises the general issue with the obviously incorrect values: having identified them what do you do with them? There are two general options: correct them, perhaps by going back to the original data source or making a new determination, or set them to missing values.

Having eliminated the "illegal" values, the analyst at the preprocessing stage often looks at extreme (but legal) values. This method of detection consists of trying to find gross errors through lists or one-dimensional plots of individual variables. This approach has many difficulties. Among them are the increased likelihood that "correct" extreme values, which may represent the most interesting cases in the database, are set to missing while many of the more troubling observations which are incorrect, but not in a univariate outlier sense, are retained. An uncharacteristic instance may have each of its recorded variables well within the mainstream of the variable values for the database, but may still be highly unusual. An often-told story related to the U.S. Census is the occurrence of 9 year old grandmothers and other similar phenomena. Being 9 years old is not unusual, neither is being a grandmother, but the two together indicate either a gross error or a highly unusual youngster.

This inability to detect the multivariate outliers is clearly acknowledged in some general texts on data mining (*e.g.*, Kennedy, *et al.*, 1997). What tends to happen in practice is that the data analyst uses substantial subject area knowledge to look at the major multivariate relationships that might be problematic. This is most often done by looking at commonly used two-dimensional plots. The number of bivariate relationships that must be examined grows rapidly as the number of variables in the databases grows and examination of two dimensional relationships are unlikely to be adequate for identifying erroneous observations when even moderately complex phenomena and heterogeneous databases are involved. Further, many of the causes of erroneous data values do not produce extreme values in a univariate sense. For instance, in an environmental database that one of us has worked with, a number of errors consisted of the same reading being entered into the record repeatedly. Since the original reading was not an error, one-dimensional plots would not reveal these erroneous repetitions

If data values judged to be likely to be erroneous are not corrected or set to missing, then the procedures used for the knowledge discovery exercise should be designed to be relatively insensitive to such observations. The implementation of loss functions that are robust/resistant to distributional assumptions and multivariate outliers is now common in mainstream statistical analysis (Rousseeuw and Leroy, 1987; Staudte and Sheather, 1990; Barnett and Lewis, 1994; Venables and Ripley, 1997). Such loss functions are being implemented with increasing frequency in standard knowledge discovery techniques such as neural networks (*e.g.*, Martin, 1995; Liano, 1996; Tsai, Chung, and Chang, 1996; Wang, Jiang, and Yu, 1996). This trend is not surprising

given the much greater sensitivity that techniques able to achieve good local approximation of a function to local outliers usually exhibit.

Missing Values

Cabena, *et al.* (1998) aptly characterize this issue when they say: "To deal with missing values, data analysts use different techniques, none of which is ideal." The most commonly used method because of its easy implementation is to simply delete all observations that have a missing value in any of the variables being used. IBM's Intelligent Miner, for instance, has an option in its data processing module to "discard records with missing values."

This practice is problematic for two reasons. First, the amount of data discarded may be large and this can seriously deplete the dataset used in the analysis. For instance, in a medical database, where one might have hundreds of variables measured on an individual, this would implies deleting that individual from the analysis if even one of her measurements was missing. Second, unless the variable value is missing complete at random, which is typically not the case, the parameters of the model estimated will generally be biased.

If the cases with missing values on variables being used in the analysis are not deleted then effectively one must assign a value to the missing values in some fashion. The simplest approach for filling in the missing value is to assign some summary statistic such as the mean based on the cases with observed values. This approach eliminates the loss of data but is not appropriate unless the data is missing completely at random. For instance, in the Fatal Accident Reporting System blood alcohol level example noted above, this test is often not conducted for occupants of a vehicle not at fault in the accident where there is no other evidence suggesting the use of alcohol. Thus imputation of the mean blood alcohol level from the cases where this test was conducted to cases where this test was not conducted would like grossly overstate the role of alcohol in fatal accidents.

Recognition of the problems with imputation of missing values using only univariate information on the variable leads one to consider approaches where the value of various covariates play a role in the magnitude of the value imputed. The methods used differ widely and form a veritable Tower of Babel. One common approach is to define imputation classes based upon the belief that cases within an imputation class are more homogenous than those placed in another imputation class with respect to the variable of interest. For instance, in the 1990 Census, race-ethnicity data was missing for 30 million people; age was missing for 6 million and sex for 2.5 million. The Census Bureau fills in these missing values by defining imputations class and within those classes using what is known as a "hot deck" procedure to exploit possible spatial correlation. For instance, suppose that John Doe's age is missing. Census matches him on the non-missing characteristics defining the imputation classes, *e.g.*, race-ethnicity, sex, renter or owner and then tries to find the geographically, nearest individual to John Doe. Say this person is Jim Jones. Census then uses Jim Jones age as a fill-in for John Doe's age.

The other common approaches are based upon formally conditioning on available covariates (Little and Rubin, 1987). The simplest of these approaches is to regress the variable whose missing values are to be imputed on all of the other available covariates. The approach only works

if there are no missing values on the covariates being used. It requires the much weaker assumption a value is missing not completely at random but rather missing at random conditional on the observed covariates and functional form being used in the regression equation. To tackle the more typical situation where all of the variables being used have missing values, one typically makes a parametric distributional assumption concerning the joint distribution of the variables. Some variant of the EM algorithm is then used to iteratively impute the expected value of the missing values and then maximizes the joint likelihood function as if the imputed values were the true values until a specified convergence criteria is met (Demster, Laird, and Rubin, 1977). Several statistics packages (*e.g.*, BMDP) include a procedure for using the EM algorithm to impute missing values under the assumption of a joint normal distribution.

The difficulty with the joint normality assumption (or any other joint parametric distributional assumption) is that it is unlikely to be appropriate without a substantial modeling effort with respect to the relationship between the variables. The joint normality assumption, in particular, can lead to some very odd missing value imputations when it is strongly violated. This will almost always be the case when the variables of interest are of different types: continuous variables, continuous variables with a bounded range of support, count variables, unordered categorical variables, and ordered categorical variables. Consider, for instance, an HMO medical database consisting of personal and family data plus other variables such as the results of many different types of tests, the drugs previously prescribed, medical procedures performed, and information related to the primary care physician. Specifying a parametric joint distribution for all of these variables, *a priori* without an extensive modeling and specification effort, requires an incurable and unrealistic optimism, but this is what is necessary in order to believe in the validity of the filled-in missing values.

As a consequence, it is not surprising that much of the missing value literature in recent years has moved in the Bayesian direction. Gelman, *et al.*, 1995 provide a useful review. The Bayesian approach is a natural one because it formalizes the uncertainty about the joint distribution. Further, the EM algorithm whose first uses were with respect to missing values can be adapted to finding the mode of the marginal posterior density in a wide-variety of Bayesian applications (Tanner and Wong, 1987) where missing values are a natural feature of the estimation. A number of recent applications of the Bayesian approach such as Rubin, Stern, and Vehovar (1995) suggests that the Bayesian approach can perform quite well with respect to predicting missing values. The difficulty with the Bayesian approach is that while it represents an elegant way of naturally dealing with the missing value question, its effective implementation in a database with a very large number of variables of mixed type remains problematic with respect to both labor and computational time if anything approaching a short turn around in cleaning and mending a new database is required. The Bayesian approach may be the ideal one in situations with sufficient time for hand tuning and the receipt of new samples from the population on a regular basis.

There is one last direction in imputing missing values which should be briefly discussed and that is the concept of multiple imputation (Rubin, 1987). Almost all of the procedures which have been proposed to imputing missing values can be seen as filling in the missing value with some type of predicted value for the observations. A natural consequence of doing this is to reduce the variability of the data set with the missing values filled in relative to what would have been

observed if there were no missing values present. Effectively, using the prediction does not incorporate any information on the confidence in the prediction. One way of providing this information is to provide multiple imputations for the missing values. The different imputed values all effectively come from taking the predicted value and adding a random component to it. A number of different approaches to obtaining the different imputations have been put forth in the literature (Rubin, 1996). The multiple imputation concept can be profitably implemented with any method of imputing missing values.

III. Proposed Effort

We propose developing a tool that will provide accurate imputation of missing values and lists of all outliers for any database, small and immense. For large databases, the method we propose can be effectively parallelized, and transportable parallel code will be developed using the resources of the UC San Diego Supercomputing Center. We anticipate that the methods proposed will become universally accepted for filling in missing values and detecting outliers in virtually all standard databases where hand-tuned solutions are not feasible.

The building block of our method is a decision tree method called CART developed by four statisticians (Leo Breiman, Jerome Friedman, Richard Olshen and Charles Stone) at UC Berkeley and Stanford and described in a book published in 1984. Since then it has become widely known with thousands of references in the machine learning and statistical areas. It has been applied to an amazingly diverse set of problems and data--routing aircraft, controlling robot movements, diagnosing illnesses, and so on. CART will be used as the outlier detection and missing value imputation engine in the proposed procedure.

The properties of the CART methodology leading to its widespread use are that no distributional assumptions are made regarding the data. The data can consist of any mixture of numerical, categorical and nominal valued variables. Missing values are handled in a powerful way by using surrogate variables and trees are optimized in size for predictive accuracy. The prime function of CART is predictive—if one particular variable is designated, then CART will form a binary decision tree to predict the values of that variable on the basis of the other variables. In thousands of tests on all kinds of data, CART has proven to give reliable and robust predictions no matter what the distribution and type of data used.

Our approach is to use CART to form a prediction tree for every variable in the database in terms of the other variables. For instance, if the database consists of individual instances with each consisting of the values of 1000 variables, then 1000 prediction trees will be constructed—one for each variable. Outliers will be flagged if the predicted value differs too much from the actual value. The first missing value iteration will replace missing values in a variable with its tree-predicted value.

For instance, suppose a tree is grown to predict age from other variables in the Census database and that one of the other variables is a 0-1 response to the question "are you a grandparent of any children?" Then a positive response together with other variables might

predict age as 60 years with a dispersion band of -10 to +40 years. An actual age of 9 compared to a prediction of 60 would certainly be flagged for inspection.

What we will produce in our research is an automatic cleaning and mending procedure applicable to all standard databases. The procedure will produce a list of outliers for inspection by the analyst and we plan to implement graphical representations to further help the analyst understand the pattern of multivariate outliers in the database. The analyst will have the option of all or some of the designating outliers to be retained in the database while the rest are set to missing and filled in by the missing value procedure.

A rudimentary handcrafted version of the above approach has been applied to some environmental survey data and to some Census survey data. Various methods were used to artificially create missing values. One method was "completely at random"—*i.e.*, throw darts at the data. The other created more missingness with questions known to be sensitive *i.e.*, income. Yet another mechanism used a conditional creation of missingness. In all cases the CART tree prediction method produced more substantially more accurate imputations than either the EM algorithm assuming joint normality, or variations on the "hot deck" method. Some of the recent research is reported in Steinberg, Carson and Breiman (forthcoming).

Given the promise of these preliminary trials and the importance of the problem to data usage, we are proposing a research effort to resolve remaining issues. While we know the general direction, there are questions that need to be answered before we can be confident that we have a method that can produce a reliable list of outliers, robustly fill in missing values and is computationally feasible.

But our goal is clear—to produce a software program that is transportable and will provide cleaning and mending for even the largest databases. To scale up to the databases in the terabyte size, parallel computing will be utilized and a program written that will run on the San Diego Supercomputer Center's massively parallel machine, a Cray T3E with 256 processors. Even using this computer, problems of insufficient fast memory size will exist and must be addressed.

The payoff to academic and scientific research and to the private industrial sectors and government database usage of such a software product should be highly significant. Many months and years of highly skilled technical efforts are spent in devising generally unsatisfactory *ad hoc* methods to clean and mend specific databases. Further, the methods currently employed can often produce biased conclusions from analyses of the data.

The remainder of this proposal provides a fuller explanation of the issues introduced above. Section IV discusses the main issues related to the use of decision trees for detecting outliers and missing value imputation. Section V discusses the main computational issues. Section VI discusses how the performance of the procedure developed will be assessed. Section VII notes research results from prior NSF funding related to this project. Section VIII discusses plans for dissemination of results from this project. Section IX briefly notes institution commitments for equipment and space. Section X discusses annual performance goals. Section XI provides a brief overview of project management issues. Section XII contains the bibliographic references.

IV. Use of Decision Trees for Outlier Detection and Missing Value Imputation

Some Basics

There are two simple concepts underlying our approach:

(a) Make optimal use of the other values in an individual instance to identify outliers.

(b) Make optimal use of the other values in an individual instance to impute the missing values.

We take up missing values first because our understanding of this issue is more advanced.

One of the earliest sensible ideas for imputing missing values in a matrix of numerical values was to form a regression of each variable on all the other variables and use the regression prediction to fill in the missing values. This is close to what the EM algorithm produces using the joint normal assumption—since under this assumption, the best predictor of any variable based on the others is given by linear regression.

But linear regression can be a poor predictor for general data distributions and becomes more difficult if the data is a mixture of variable types, *i.e.*, categorical variables such as a variable taking color values {red, blue, ...} have to be recoded into a number of dummy variables. Prediction using linear regression is now even clumsier, and should be switched over to linear discriminant analysis—although it still makes restrictive normal assumptions.

To make this approach work on general databases, one needs a prediction algorithm that satisfies two requirements:

(i) Makes no distributional assumptions regarding the data, so differing data distributions will not impair effective prediction.

(ii) Can predict numerical, categorical, and ordered categorical variables using any mixture of such variables as predictor variables.

There has been a rapidly expanding interest in prediction methods over the last 15 years stimulated by work in the neural net and machine learning community. More recently, interest and research into data mining has developed rapidly. The two most prominent distribution-free prediction methods used in these areas are neural nets and decision trees.

Use of decision trees (CART) as the basic building block has significant advantages:

- It handles and predicts mixed variable types with aplomb.
- Trees are efficiently built, generally requiring orders of magnitude fewer cycles than neural nets.

- After thirteen years of widespread use, the performance of CART has been thoroughly tested and is known to be reliable and effectively bug-free.

- Neural nets need to be handcrafted to adapt the net to the data—otherwise results can be poor. CART runs in its basic version on every database.

- CART is highly robust with respect to outlying values and use of its LAD loss function makes it even more robust. Standard neural nets are not. Therefore, it will be difficult to use neural nets to flag outliers without specially constructing the network architecture and loss functions to serve this purpose.

Architecture of Imputation System

A prediction tree will be grown for each variable using all of the other variables. What the tree construction algorithm needs to be told is only the type of each variable—numerical, categorical or ordered categorical. For imputing missing data, the procedure is as follows:

(a) Build a separate tree for predicting each variable based on all other variables.

Note: CART has a built in method for utilizing missing values in the tree construction process, so taking (a) as a starting point presents no additional problems.

(b) Have each tree fill in the missing values in its variable by the predicted values.

(c) Use these imputed values to mend the entire database, leaving no missing values.

(d) Iterate (a), (b), and (c) so as to use the revised predictions to impute the missing values.

It is uncertain at this point as to how much iteration will be used. We anticipate that generally, at most, two or three will be necessary before the imputations stabilize. This is one of the open questions that will be resolved in our research. It is entirely possible that an adaptive approach will be created in which we track the changes from iteration to iteration and stop when the total change falls below a preset threshold.

There are several different approaches to obtain a prediction at the level of a CART terminal node. One can take the standard CART prediction of the mean or median value if a continuous variable or the most likely class if a categorical variable. It is also possible to draw randomly from the non-missing values in the CART terminal node to obtain multiple imputed values or to estimate the distribution of values in the terminal node and draw the imputations from it.

Outlier Detection

The beginning idea is that if the actual value of a variable differs too much from its predicted value, then it will be flagged as an outlier. The first issue that needs to be resolved is to quantify "differs too much." Two measures are possible. One is to look at the one-dimensional distribution of all differences between actual and predicted values for the variable and flag the value if it is extreme compared to the body of the distribution. The analyst would have the option of setting

the degree of extremity to flag, *e.g.*, with approximately normal data flag at 2.5 or 3.0 sigma units.

However, it is possible that the variable in question may be more predictable in some parts of the database than in others. For instance, if a tree is grown, the prediction differences may be larger in some nodes than in others. Thus, we may want to compare differences between actual and predicted only for data in the same node. To do this, a minimum terminal node size is necessary to give a reliable estimate of the actual-predicted differences for the node. A tree that has fewer terminal nodes than the optimally sized tree may then be necessary.

While this procedure will detect single values that are outliers in terms of the other records in the instance, it may not detect outliers of the form where a single value is erroneously recorded multiple times, either because of a transcription mistake or a stuck instrument. An example is in the EPA database on quarterly well water testing at toxic waste sites, where lengthy visual inspection detected a number of repeated values.

Since groundwater chemical concentrations tend to be seasonal, trees for each quarter based on the values for the other quarters and additional relevant variables, such as the concentrations of the other tested chemicals, would reveal somewhat out-of-bounds values for that particular well in quarters where the error occurred. The question is whether the somewhat out-of-bounds readings for the three quarters can be combined to form a single flag definitely going up. Part of our work will be to try to answer this question.

The program will output a list and visual representation of the outliers to the analyst. The analyst will have a choice of two options. One is that all detected outliers be treated as missing values and replaced by the imputed values. Another option is to designate a subset as "to be studied later" and the rest filled in by imputation.

More Accurate Imputations

We intended to devote considerable effort to obtaining more accurate imputations for missing values by exploring three main directions. The first direction is to implement recently proposed techniques shown to result in improved tree estimates. The second direction is to develop techniques that allow the presence of a missing predictor variable to be informative with respect to predicting a dependent variable. The third direction is to exploit potential information in a CART terminal node that had not already been used. Each of these directions is taken up in turn. Other possible directions we intend to pursue, such as the use of prior information on a variable's univariate distribution, are not elaborated on due to space constraints.

Better Trees

Recent work on combining predictors (see Breiman (1998) for a discussion) has shown that growing multiple trees by perturbing the data set and aggregating these trees can increase prediction accuracy to the point at which, on several benchmark data sets, they achieve the lowest test set accuracy recorded to date as compared to over a dozen other classification methods, including different types of neural nets.

The simplest and most easily implemented of such procedures is called bagging (Breiman, 1996). In this method, new training sets are created by bootstrap sampling from the original training set, trees constructed using the bootstrap training sets, and the predictions of these trees averaged (regression) or voted (classification) to give an aggregated prediction. General experience is that almost all of the improvement is realized by combining 50 trees. One advantage of bagging is that it can be easily parallelized.

Other methods (arcing) of forming new training sets by sampling from the original set make the sampling adaptive and further lower misclassification rates. But arcing is confined to classification, is sensitive to outliers, and is intrinsically sequential. We will focus our research on bagging.

The question we will explore is whether use of this more accurate (but more compute intensive) prediction method will result in significantly better imputation of missing values. It is difficult to tell in advance what the effect will be—we will experiment on the artificially created missing data to see which methods most faithfully restore the original values and result in the least harm to subsequent analyses relative to having the complete data set.

There are other advantages in using the multiple bagged trees for imputation. If the imputed missing value is a categorical, then multiple trees can be used to give accurate estimates, not only of the most probable value, but also of the probabilities that the missing value was each one of the categories. For numerical missing values, the multiple trees can be used to give not only a predicted value based on the average, but also an estimate of the standard error of this prediction. In both situations, the information is useful in knowing the precision of the imputation. It would also be possible to use the imputed values from each of the bagged tree in a multiple imputation context.

Growing multiple bagged trees will also be useful in outlier detection. For instance, one of the problems in detection is to quantify how significant is the difference between actual and predicted value. Since the bagged trees provide a measure of the spread of the predicted values, one promising avenue is to flag the value as an outlier if it is more than "x" standard deviations from the predicted value. Breiman (1997) contains a study of estimation using multiple bagged trees.

Informative Missing Values

In many instances, the presence of a missing value on a predictor variable is informative as to the value of the dependent variable of interest. This seems to be particularly true in some types of application involving fraud detection. A simple way of giving this information to CART is to create a set of dummy variables that take on the value of 1 if the value of a particular variable is missing and 0 otherwise. Providing this set of indicator variables to CART has in some preliminary testing, resulted in some very substantial gains. In our missing value imputation procedure, this set of indicator variables could be created automatically and used in a computationally efficient manner.

Unexploited Information In Terminal Nodes

An early, and in many ways ingenious, approach to imputing missing values is the hot deck, where the last valid value is carried forward to be used to fill in a missing value in the next case if need be. This approach which is still used (*e.g.*, an option in SOLAS: FOR MISSING DATA) exploits that fact that in many instances there is strong spatial or temporal correlation between observations. All one needed to do in the days of key punch cards in order to use the hot deck was to sort the cases in spatial or temporal order. More sophisticated variants of this procedure define a set of *ad hoc* imputation classes and then use the hot deck approach usually with the cases spatially ordered. One way of viewing CART is that it produces the optimal set of imputation classes. If there is spatial or temporal information which has not been completely incorporated into the estimated CART tree, it is possible, instead of taking the mean value of the relevant CART terminal node as the imputed value, to take the value of spatially closest non-missing case falling into that CART terminal node as the imputed value.

V. Computational Issues: Outlier Detection and Missing Value Imputation

Modular Design

The procedure will be designed so that any outlier detection/missing value imputation engine can be plugged into it. Thus while we feel CART is the most appropriate engine, it would be possible to use another decision tree approach or something as bad as stepwise regression. This modular design has some drawbacks in terms of not fully optimizing CART for use on multiprocessor machines but doing so makes the project feasible in the allotted time. It also makes it easier to compare CART performance with that of competing methods and facilitates the use of the basic computational aspects of the procedure in other research projects.

The procedure to be developed from a computational perspective first accesses the relevant database and then efficiently divides the work between the available processors. In doing so it will have to deal with coordination issues related to outlier detection/missing value imputation between the different variables in the database. The procedure will also deal with shared memory issues across the processors. At the level of making an imputation for a missing value, a variety of options will be available for obtaining one or more imputations for the missing value. In some instances, these options will not be available for all imputation methods. For instance, drawing randomly from a terminal node is a natural thing to do for CART, other decision tree approaches and *ad hoc* imputation cells but not for a regression model where an error distribution is needed. Graphical displays of outlier and missing value patterns will also be developed for the procedure.

On-Line Requirements

All of the trees created can be compactly stored, requiring the order of M2log (N) bytes of memory storage where M is the number of variables per case and N the number of case. As new sets of records come in, they can be quickly run through all trees in the time of M*log(N) flops per case.

In the rudimentary version of the procedure put together in Steinberg, Carson, and Breiman (forthcoming), it took approximately 15 minutes on a single processor mid-range DEC Alpha to do two imputation iterations on a data set of approximately 30,000 observations and 25 variables with 20% of the data initially missing. It is interesting to contrast this with the first application of

CART to imputing missing values (Carson, 1984) where a single imputation round on a data set with approximately 1000 observations and 25 variables took almost a week on a VAX 11/750. It is not surprising that the approach was then seen as infeasible for uses with very large datasets even though the imputation results were quite good relative to other techniques then in use.

Division of Labor Across Processors

In the current implementation of CART all data to be used in the development of a classification or regression tree is held in RAM. Trees are developed for one variable at a time and one node at a time. To generate a node split, a pointer to the rows of the data is sorted on the values of one of the columns (a candidate splitter variable), and then each distinct value of the variable in that column is tested for splitting potential. The best split value (possibly subject to some constraints) for the column is saved. The sorting of the rows followed by evaluation of all values of a candidate splitter is repeated for each column of the data: the best column is selected to partition the data. Once the partition has been selected the subsets of the data defined by the partition are themselves searched for best partitions in turn. The process may be allowed to continue until a predefined depth of the tree has been reached, a preset minimum node size has been attained for every terminal node, or until further splitting is not possible.

There are several ways in which this work can be divided among multiple processors, not all of them equally feasible or efficient. First, when there are a large number of variables to treat as dependent or target variables, each analysis can be entirely independent of the others. Each variable to be examined for outliers or repaired with imputed values for missing values can potentially be analyzed without reference to any other tree. Thus, a natural way to parallelize the method is to allow each processor to develop the next tree scheduled to be grown in toto. Only when the results of the tree and its analysis are complete will it be necessary for any processor to communicate with the program supervisor. So long as the number of variables is substantially larger than the number of processors available (the usual situation for large databases), this division is natural and eliminates the need for most interprocess communication or coordination.

Another way to divide the computational labor is to allow more than one processor to work on a single tree. In this approach, there are several options. First, because each column must be searched separately, the columns could be divided between processors. Once each column had been searched and the best partition identified, the entire set of columns could be searched again for the next node needing to be processed. Again, processors would communicate their findings to the supervisor, which would determine the best split and so on. This is the simplest way for multiple processors to cooperate in growing a single tree.

A third way to use multiple processors would be to allow different processors to work on separate sub-trees. Once a node is split, the left and right child partitions can be searched separately, and thus, can be searched by different processors. Combination methods, in which processors sometimes divide the work by searching different columns of data and at other times by searching different sub-trees, are also possible.

It is plain the amount of inter-processor communication and coordination is minimized when the smallest unit of work assigned is the generation of an entire tree. The most complex

programming would be required when several processors would share the work in developing a single tree and each processor searches different parts of the tree. In the typical large database environment we see emerging, the number of variables will typically exceed the number of processors, and thus the simplest arrangements are the most desirable. At worst, a small number of processors may be idle while they wait for others to finish growing trees for variables that generate unusually large trees. There may be circumstances in which a large number of processors are available for the analysis of a single tree. In this circumstance the focus should be on methods in which the processors could share the development of a single tree. However, this circumstance by its nature is not characteristic of the large database world we anticipate.

Memory Matters

Given that the data CART will work with should reside in RAM, there are two important questions to address: (1) should each processor maintain its own copy of the data in its private RAM, and (2) how will we deal with the probably common situation in which the database being analyzed is far larger than available RAM? As we think it is important that as much RAM as possible be made available to the outlier detection-missing value process, it is likely that this is possible only when each processor accesses shared memory on a central server. With such an arrangement it is feasible to allow all processors to share some very large RAM, for example on the order of 32 Gigabytes. Even with the rapid decline in RAM costs, it is not reasonable to expect that supercomputer centers can routinely equip each processor with such amounts of RAM. With multiple processors working on a single image of the data in shared RAM, each processor will need to maintain only its own private pointers to the rows of the database (for sorting) and private copies of the trees it develops. Both types of private data are orders of magnitude smaller than the main database and impose no noticeable burden on the local RAM of an individual CPU. The pointers are needed only for the duration of a single column search, and the resulting trees can be written to disk with minimal performance penalty. The worst case scenario for multiple processors working on a shared database occurs when just one column of data remains to be analyzed, and, thus all but one of the processors would be idled while waiting for analysis of the final column to complete. This would occur for example if we had 256 processors working on 1001 columns of data; after each processor had completed 4 columns of data only one column would remain for analysis. As part of our research we will investigate ways to use the idled processors in this circumstance. One option would be to check the completed trees for prediction accuracy and to regrow poorly performing trees using a different tree growing method (*e.g.*, switch from the gini rule to the twoing or entropy rules). Any processor expected to be idled for more than a predetermined waiting time could conduct such refinement cycles.

Dealing with databases much larger than available RAM is easily handled for missing value imputation. Building on prior work in the combining of trees, our procedure would be as follows: (1) randomly divide the database to be processed into M segments, each of which can fit into the available RAM (2) develop a predictive model for each variable needing missing value imputation for each of the M segments, and (3) develop a final model by combining the separate models for each variable into a committee of experts, as is done in bagging, boosting, and arcing. Then use the combined tree model to impute missing values. Previous research supports our belief that such tree combination will yield very acceptable results. In particular, combining trees from separate subsamples of a very large database will give far better results than would be obtained from the

use of just one of the available trees. For example, with 32 Gig of RAM available, a terabyte database could be effectively analyzed with 20-60 trees developed per column of data. The exact number of trees required will depend on how much data is reserved for testing the CART predictive models. Since the CART trees are orders of magnitude faster to train than neural nets, the results will still be available more rapidly than could be obtained from a neural network, and without any special preprocessing or preparation of the data.

Application of the tree combination method to outlier detection is more difficult conceptually, although it is equally simple to implement. A major question we will need to address for this topic concern how to use the different trees: (a) take each segment of the data and let its own trees identify outliers—workable but potentially leading to inconsistent results (same value could be an outlier in one segment and not in another) or (b) wait until all K trees are grown. Develop an outlier rule for each tree (the rule might be quite complex and could potential be based on the entire tree sequence). Now drop all data down each tree and combine the judgements. One interesting criterion would be that a variable value must be declared an outlier on at least K trees in order to be branded as a multivariate outlier in the data set.

VI. Assessing Performance Characteristics of Procedure

An extensive effort will go into evaluating the performance characteristics of the procedure developed. The first set of these performance characteristics will be related to computation requirements in terms of time and memory given different size databases. Here we will be particularly concerned with how these resource requirements are influenced by the number of processors available and by different options with respect to how the CART trees are estimated.

The second set of performance characteristics related to how well the procedure does at detecting outliers. Here we will start with synthetic datasets where the outliers are created under the researcher's control. The objective here will be to create a design space that will adequately span a wide range of interesting and likely situations. An assessment will also be made as to how various (statistical) options available in the procedure influence the results. In the early stages of the project, these assessments will be diagnostic and will help eliminate options that do not work well and will hopefully suggest new options or ways of improving existing options.

The third set of performance characteristics related to how well the procedure does at imputing missing values. As in the outlier assessment above, we will start with synthetic datasets where the outliers are created under the researcher's control. The objective here will also be to create a design space that will adequately span a wide range of interesting and likely situations. A reasonable design space here is likely to be considerably larger than that for the outlier detection. We will concentrate on situations where there are interesting sample selection mechanisms (Heckman, 1979). These are among the more difficult situations for missing value imputation. At later stages, we will move to the use of large empirical databases to assess the performance of the procedure. For missing values, we will be particularly interested in a small number of publicly available data sets where Census has matched various variables to administrative records. The performance of the procedure developed under different contexts and different options will also be compared to several commonly used competitors.

VII. Prior NSF Support Related To This Proposal

NSF has supported research by Leo Breiman on nonparametric methods for classification and prediction for high dimensional data (DMS-9212419, Mathematical Sciences: computer intensive methodology in classification and regression, ($267,873).

The general theme that runs through all of Breiman's research is the investigation of methods for making accurate class and numerical predictions in complex databases. For instance, one recent avenue has been to improve the accuracy of tree-structured methods by resampling the training data and then combining alternative prediction/classification algorithms. These strategies can lead to dramatic improvements of performance, and have already been incorporated into two of the most notable data mining packages.

Publications Related to Breiman's Prior NSF Support

(1997, with J. Friedman), "Predicting Multiple Responses in Regression," with discussion, *Journal of the Royal Statistical Society*, Series B, 59, 3-54.

(1996), "The Heuristics of Instability in Model Selection," *Annals of Statistics*, 24, 2350-2381.

(1996), "Stacked Regressions," *Machine Learning*, 24, 41-64.

(1996), "Bagging Predictors," *Machine Learning*, 26, 123-140.

(1996), "Some Properties of Splitting Criteria," *Machine Learning*, 26, 41-47.

(1995), "Better Subset Selection Using the Non-Negative Garrote," *Technometrics*, 37, 373-384.

(1994, with A. Cutler), "Archetypal Analysis," *Technometrics*, 36, 338-347.

(1994), "Discussion of 'Neural Networks—Review from a Statistical Perspective, by Cheng and Titterington'," *Statistical Science*, 9, 2-54.

(1994), "Reflections After Reading Papers from NIPS," In *The Mathematics of Generalization*, D. Wolpert, ed. (Reading, MA: Addison-Wesley).

(1994), "Current tree research," In *The Mathematics of Generalization*, D. Wolpert, ed. (Reading, MA: Addison-Wesley).

(1993, with A. Cutler), "A Deterministic Algorithm for Global Optimization," *Mathematical Programming*, 58, 179-199.

(1993), "Hinging Hyperplanes for Regression, Classification and Noiseless Function Approximation," *IEEE Transactions on Information*, 39, 999-1013.

(1993), "Fitting Additive Models to Regression Data," *Journal of Statistical Computation and Data Analysis,* 15, 13-46.

VIII. Dissemination of Results

Results from the project will be disseminated in five main ways: (a) presentations at conferences, (b) publication in professional journals, (c) development of an ASA LearnStat course, (d) a web site, and (e) one or more alpha testers.

Conference Presentations

We intended to make a substantial number of conference presentations related to the work on this project. Conferences where presentations are likely to be made include the American Statistical Association, the annual Interface Conference, the annual Machine Learning conference, the annual Knowledge Discovery in Databases conference and the annual Neural Information Processing (NIPS) conference. We also feel that it is important to present the work and get feedback at a variety of conferences with audiences who have a great deal of experience with particular types of outlier and missing value problems. In this regard, we intend to present papers on the procedure at an American Association of Public Opinion Research conference, an American Marketing Association Advanced Research Techniques Forum, an Econometric Society meeting, an INFORMS conference, and at the proposed International Conference on Survey Nonresponse. We also intended to present the results at conferences focused on parallel computing issues such as the International Parallel Processing Symposium and the National Partnership in Advanced Computational Infrastructure (NPACI) Parallel Computing Institute. Presentations will also be made in seminars presented at universities and other institutions.

Publications

We intend to seek publication for many of the papers presented at conferences in order to reach a larger audience. We also intend to write one or more non-technical papers aimed at those responsible for assembling and maintaining large databases.

ASA LearnStat Course

A *LearnStat Course* will be developed for presentation at one or more of the annual American Statistical Association meetings. This has proven to be a successful format for introducing new procedures to a technically sophisticated audience.

Web Site

The project will maintain a web site. The web site will describe the objectives of the project, conference presentations and provide downloadable technical reports. In the literature review phase of the project we will create a set of web links to other outlier and missing value resources.

Alpha Testers

We will plan to entice a small number of other researchers to use the software and provide feedback on issues related to it ease of use and performance.

IX. Institutional Commitment to Equipment and Space

If this project is funded, we will apply for a grant of up to 20,000 processor hours from the San Diego Supercompter Center (SDSC). Preliminary indications are that this grant will be approved if this project is funded by NSF. Most of the processor hours are likely to be on SDSC's Cray T3E. This is a 256-processor machine rated at 150 GFLOPS. Each processor has 128MGB RAM and this memory can be linked across processors. There is an effective limit on the file size on this system of approximately 100GGB, although there is a considerably larger amount of disk storage (500GB) available on the system and a 200TB IBM HPSS high-speed tape system. We intend to port the code to one or more other computing platforms with multiple processors. At this time, the most likely multiprocessor machines are SDSC's new SUN Enterprise Server 10000 with 16 processors, SDSC's older SUN cluster system with 28 available processors or SDSC's DEC Alpha cluster. Breiman has access to multiprocessor SUN workstations, Carson has access to a multiprocessor HP machine, and Steinberg has access to a multiprocessor DEC machine.

If this project is funded, SDSC will provide space for the programmer to be hired as well as for Carson's graduate student researcher. SDSC will also provide temporary office space for Steinberg during periods when he is working on the project. Carson and Lear are already at SDSC.

X. Annual Performance Goals

Year One

We will conduct a review of: (a) the relevant literature on multivariate outlier detection and missing value imputation, (b) the options for handling these two problems which are available in the major commercial data mining packages, and (c) proposals in the academic literature related to tabular and visual displays of information related to outlier and missing value patterns in the database. A detailed specification of the desired features and available options for the proposed procedure will be developed. A detailed working schedule of programming tasks will be developed. The major objective is to get a basic version of the proposed procedure successfully working on a multi-processor machine. Results of work will be reported at professional conferences. The planned web site will be initiated.

Year Two

The performance of the basic version of the procedure will be analyzed and modifications made to more efficiently exploit capabilities of multi-processor machines. Basic procedure will be enhanced with additional features and options. Initial assessment of performance of variants of the procedure using synthetic data will be made. Results of work will be reported at professional conferences and papers will be prepared for submission to journals.

Year Three

Additional enhancements and refinements will be made to the procedure. Procedure will be ported to one or more additional multi-processor computing platforms. Extensive testing of properties of different variants of the procedure and competing procedures will be made using

synthetic and actual datasets. Results of work will be reported at professional conferences and additional papers will be prepared for submission to journals. Alpha testers of software will be located. LearnStat course on procedure will be offered at American Statistical Association meeting.

XI. Project Management

This is a multi-institution project that will be managed by the PI-Richard Carson, with UCSD/SDSC providing administrative support. Carson has considerable experience managing very large projects with multiple researchers at different locations and will work on all aspects of the project. He has served as PI for a number of large natural resource damage assessments for government agencies including the economic evaluation of the damages from the Exxon Valdez oil spill. For this project, most of the work and, in particular the programming effort, will occur at the San Diego Supercomputer Center (SDSC). Carson is a Senior Fellow at the Center, Robert Leary (General Atomics subcontract) is a Senior Staff Scientist at the Center and is in charge of overseeing the mathematical and statistical software libraries for the Center's Cray T3E and other multi-processor machines. Dan Steinberg will be in residence at SDSC during much of the project period and the main programmer at the project will be housed at SDSC as will Carson's graduate student researcher. Leary and Steinberg both have considerable expertise in managing large software projects. Carson founded the Social Science Computer Facility at UCSD and Breiman is director of the Statistical Computing Facility at UCB. Carson, Breiman, and Steinberg have a good working relationship and have collaborated in the past. UC Berkeley and San Diego State University are both members of National Partnership in Advanced Computational Infrastructure (NPACI) consortium headquartered at SDSC.

Project personnel at SDSC will meet weekly to discuss progress on a well-defined set of short term and longer term objectives. Leo Breiman and his graduate student researcher will take the lead on theoretical issues and new design features related to the proposed procedure. Trips between UCB and the San Diego Supercomputer Center are planned to coordinate efforts. Steinberg will take the lead on issues related to the conceptual design of the procedure and its programming implementation. Leary will take the lead on issues related to effective and transportable implementation of the procedure on multiprocessor machines. Carson will take the lead with respect to designing and implementing tests of the procedure.

XII. References

Barnett, V. and T. Lewis (1994), *Outliers in Statistical Data, 3rd ed.* (New York: Wiley).

Brachman, R.J. and T. Anand (1996), "The Process of Knowledge Discovery in Databases," in U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., *Advances in Knowledge Discovery and Data Mining* (Cambridge: MIT Press).

Berry, M.J. and G. Linoff (1997), *Data Mining Techniques for Marketing, Sales, and Customer Support* (New York: Wiley).

Breiman, L. (1996), "Bagging predictors," *Machine Learning*, 26, 123-140.

Breiman, L. (1997), "Out-of-Bag Estimation," Technical Report, Department of Statistics, University of California, Berkeley.

Breiman, L. (1998), "Randomizing Outputs to Increase Prediction Accuracy," Technical Report 518, Department of Statistics, University of California, Berkeley.

Breiman, L., J. Friedman, and R. Olshen, and C. Stone (1984), *Classification and Regression Trees* (Pacific Grove, CA: Wadsworth).

Cabena, P., P. Hadjinian, R. Stadler, J. Verhees, and A. Zanasi (1998), *Discovering Data Mining: From Concept to Implementation* (Upper Saddle River, NJ: Prentice-Hall).

Carson, R.T. (1984), "Compensating for Missing Data and Invalid Responses in Contingent Valuation Surveys," in *1984 Proceedings of the Survey Research Section of the American Statistical Association* (Washington: American Statistical Association).

Dempster, A.P., N.M. Laird, and D.B. Rubin (1977), "Maximum Likelihood From Incomplete Data Via the EM Algorithm," (with discussion), *Journal of the Royal Statistical Society*, B39, 1-38.

Gelman, A., J.B. Carlin, H.S. Stern, and D.B. Rubin (1995), *Bayesian Data Analysis* (London: Chapman and Hall).

Gupta, A. and M.S. Lam (1996), "Estimation of Missing Values Using Neural Networks," *Journal of Operational Research Society*, 47, 229-238.

Guyon, I. N. Matic, and V. Vapnik (1996), "Discovering Informative Patterns and Data Cleaning," in U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy,

eds., *Advances in Knowledge Discovery and Data Mining* (Cambridge: MIT Press).

Heckman, J. (1979), "Sample Selection Bias as Specification Error," *Econometrica,* 47: 153-161.

Huber, P. (1981), *Robust Statistics* (New York: Wiley).

John, G.H. (1995), "Robust Decision Trees: Removing Outliers from Databases," in U.M. Fayyad, and R. Uthurusamyu, eds., *KDD-95: Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (Menlo Park, CA: AAAI Press).

Kennedy, R.L., Y. Lee, B. Van Roy, C.D. Reed, and R. Lippman (1997), *Solving Data Mining Problems Through Pattern Recognition* (Upper Saddle River, NJ: Prentice-Hall).

Lakshminarayan, K. S.A. Harp, R. Goldman, and T. Samad (1996), "Imputation of Missing Data Using Machine Learning Techniques," in E. Simoudis, J. Han, and U. Fayyad, eds., *KDD-96 Proceedings: Second International Conference on Knowledge Discovery and Data Mining* (Menlo Park, CA: AAAI Press).

Liano, K. (1996), "Robust Error Measure for Supervised Neural Network Learning with Outliers," *IEEE Transactions on Neural Networks*, 7, 246-250.

Little, R.J.A. and D.B. Rubin (1987), *Statistical Analysis with Missing Data* (New York: Wiley).

Madow, W.G., H. Nisselson, I. Olkin, and D.B. Rubin, eds. (1983), *Incomplete Data in Sample Surveys, 3 vols.* (New York: Academic Press).

Martin, R.D. (1995), "Robust Neural Networks," in *Proceedings of the 1995 Conference on Computational Intelligence for Financial Engineering* (New York: IEEE).

Pemmaraju, S. and S. Mitra (1993), "Identification of Noise Outliers in Clustering by a Fuzzy Neural Network," in *Proceedings of IEEE Second International Fuzzy Systems Conference* (New York: IEEE).

Redman, T. (1992), *Data Quality: Management and Technology* (New York Bantam).

Ripley, B.D. (1996), *Pattern Recognition and Neural Networks* (New York: Cambridge Unversity Press).

Roosseeuw, P.J. and A.M. Leroy (1987), *Robust Regression and Outlier Detection* (New York: Wiley).

Rousseuw, P.J. and B.C. van Zomeren (1990), "Unmasking Multivariate Outliers and Leverage Points" with discussion, *Journal of the American Statistical Association*, 85, 633-651.

Rubin, D.B. (1987), *Multiple Imputation For Non-Response in Surveys* (New York: Wiley).

Rubin, D.B. (1996), "Multiple Imputation After 18+ Years," *Journal of the American Statistical Association*, 91, 473-490.

Rubin, D.B., H.S. Stern, and V. Vehovar (1995), "Handling 'Don't Know' Survey Responses: The Case of the Slovenia Plebiscite," *Journal of the American Statistical Association*, 90, 822-828.

Schmitz, J. (1996), "Marketing," in *Massive Data Sets Workshop: Proceedings of a Workshop* (Washington: National Academy Press).

Staudte, R.G. and S.J. Sheather (1990), *Robust Estimation and Testing* (New York: Wiley).

Steinberg, D., Carson, R.T. and L. Breiman (forthcoming), ""Broad Scale Missing Value Imputation with Iterative Binary Partitioning," in *1997 Symposium on the Interface: Computing Science and Statistics.*

Steinberg, D. and P. Colla (1995*), CART: Tree-Structured Non-Parametric Data Analysis* (San Diego: Salford Systems).

Tanner, M.A. and W.H. Wong (1987), "The Calculation of Posterior Distributions By Data Augmentation," (with discussion), *Journal of the American Statistical Association*, 82, 528-550.

Teague, A. and J. Thomas (1996), "Neural Networks as a Possible Means for Imputing Missing Census Data in the 2001 British Census of Population," in R. Banks, J. Fairgrieve, L. Gerranrd, and T. Orchard, eds., *Proceedings of the Second International Conference on Survey and Statistical Computing* (Chesham, UK; Association of Survey Computing).

Tsai, J.R., P.C. Chung, and C.I. Chang (1996), "Resisting the Influence of Outliers in Radical Basis Function Neural Networks," in S. Usui, Y. Tohkura, and S. Wilson, eds., *Proceedings of the 1996 IEEE Signal Processing Workshop,* Kyoto Japan (New York: IEEE).

Venables, W.N. and B.D. Ripley (1997), *Modern Applied Statistics With S-PLUS, 2nd ed.* (New York: Springer-Verlag).

Wang, J.H., J.H. Jiang, R.Q. Yu (1996), "Robust Backpropogation Algorithm as a Chemometric Tool to Prevent the Overfitting to Outliers," *Chemometrics and Intelligent Laboratory Systems*, 34, 109-115.

Witte, D.L., S.A. VanNess, D.S. Angstadt, and B.J. Pennell (1997), "Errors, Mistakes, Blunders, Outliers, or Unacceptable Results: How Many?," *Clinical Chemistry*, 43, 1352-1356.

Wong, P.M. and T.D. Gedeon (1995), "A New Method to Detect and Remove the Outliers in Noisy Data Using Neural Networks: Error Sign Testing," *Systems Research and Information Science*, 7, 55-65.