Applied Intelligence 11, 259–275 (1999)
© 1999 Kluwer Academic Publishers. Manufactured in The Netherlands.

Imputation of Missing Data in Industrial Databases

KAMAKSHI LAKSHMINARAYAN, STEVEN A. HARP AND TARIQ SAMAD
Honeywell Technology Center, 3660 Technology Drive, Minneapolis, MN 55418
laksh004@tc.umm.edu
sharp@htc.honeywell.com
samad@htc.honeywell.com

Abstract. A limiting factor for the application of IDA methods in many domains is the incompleteness of data repositories. Many records have fields that are not filled in, especially, when data entry is manual. In addition, a significant fraction of the entries can be erroneous and there may be no alternative but to discard these records. But every cell in a database is not an independent datum. Statistical relationships will constrain and, often determine, missing values. *Data imputation*, the filling in of missing values for partially missing data, can thus be an invaluable first step in many IDA projects. New imputation methods that can handle the large-scale problems and large-scale sparsity of industrial databases are needed. To illustrate the incomplete database problem, we analyze one database with instrumentation maintenance and test records for an industrial process. Despite regulatory requirements for process data collection, this database is less than 50% complete. Next, we discuss possible solutions to the missing data problem. Several approaches to imputation are noted and classified into two categories: data-driven and model-based. We then describe two machine-learning-based approaches that we have worked with. These build upon well-known algorithms: AutoClass and C4.5. Several experiments are designed, all using the maintenance database as a common test-bed but with various data splits and algorithmic variations. Results are generally positive with up to 80% accuracies of imputation. We conclude the paper by outlining some considerations in selecting imputation methods, and by discussing applications of data imputation for intelligent data analysis.

Keywords: missing data, industrial databases, multiple imputation, machine learning

1. Introduction

Like many businesses involved in the manufacture and service of complex equipment, Honeywell and its customers compile and maintain vast amounts of data. This data is usually incomplete—i.e., many records have fields that are not filled in. Such omissions often arise when entries must be done manually. The incompleteness of the data makes the use of common charting, data visualization and reporting tools difficult. Analysts interested in using the data for parameter estimation or statistical inference are handicapped since most common data analytic packages either analyze only complete cases, or handle missing data entries using idiosyncratic strategies such as case deletion. *Data imputation*, or the filling in of missing values for partially missing data can be used to complete such incomplete databases, and often forms an invaluable first step in IDA.

To motivate and illustrate the missing data problem, we describe a database consisting of instrumentation maintenance and test records for an industrial process. Such data is regularly collected for processes such as oil refineries, chemical factories, pharmaceutical plants, and pulp and paper mills. Despite regulatory requirements for data collection, the database we discuss is less than 50% complete.

We have experimented with some novel techniques for data completion using data from the instrumentation maintenance database described above. One of these techniques uses a decision-tree based supervised learning method. The other approach is a type of *hot-deck imputation*, a common statistical procedure for completing incomplete databases. We describe our

Roku EX1019
IPR2024-01055
U.S. Patent No. 8,756,622

experiments and their results, and draw conclusions. In addition to describing our empirical work, we review the literature on existing data completion methods.

This paper elaborates our preliminary work on data completion methods for industrial databases [1].

1.1. Structure of This Paper

The organization of this paper is as follows. The first section describes the magnitude of the missing data problem in the type of industrial databases maintained by Honeywell and its customers. The second section summarizes existing methods from statistical literature for dealing with missing data, and talks in detail about imputation procedures and common reasons for missing data imputation. The next section introduces the two techniques we have explored as potential data completion approaches. The first of these two techniques involves modeling the data by supervised induction of a decision tree-based classifier, C4.5 [2]. The classifier learns the function mapping between the predictor variables' values, and the target attribute. This classifier can then be used to predict the most likely choices for missing entries of the target variable. The second technique involves a commonly used statistical imputation procedure known as hot deck imputation. Here the data is divided into subsets or clusters, each of which is homogeneous with respect to values of some designated attributes. Missing attribute values for a certain case or record are filled in by sampling from the cluster to which that record belongs. The next section presents and discusses empirical results from the application of these two procedures for filling in missing data, and factors affecting the choice of one imputation method over another. The last section discusses potential applications of the procedures we are developing.

2. The Missing Data Problem

Honeywell and its customers routinely compile maintenance information for plant and building equipment installed in various locations. Entry in these databases is carried out by field personnel, and for various reasons is plagued by a high proportion of missing values in data fields. In addition, the entered data is sometimes erroneous, or is in a non-standard format and frequently has spelling errors.

The magnitude of the problem for a typical industrial process maintenance database studied by one of Honeywell's business units is shown in Fig. 1. This database contains maintenance information for control devices. Values for 82 variables or features are recorded in this database. Some of these variables are categorical and others are continuous. The variables measure properties of the devices such as the manufacturer and model of a device, as well as states of a device at various

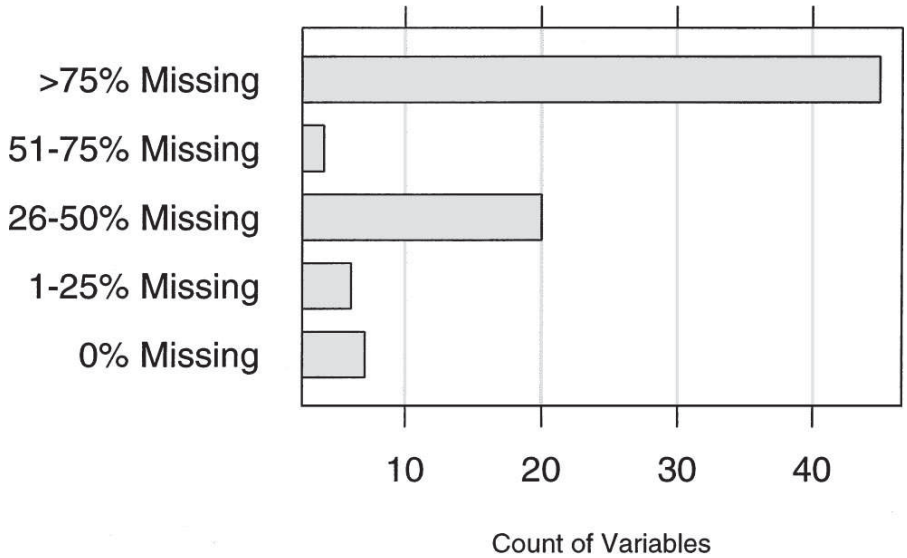

Figure 1. Number of variables in each of 5 categories of severity of missing cases (%).

times such as calibration error and pass/fail results of testing. Of the 4383 records in this database, none of the records were complete and only 33 variables out of 82 have more than 50% of the records complete. Other databases dealing with equipment health maintenance have similar magnitudes of missing data. This level of incompleteness of these databases seriously limits their usefulness to both analysts and field personnel.

While some of data fields in the database described above are empty due to structural zeros (e.g., the field corresponding to 'Type of Controller' for a measuring device such as a voltmeter), a large fraction of the missing data may be attributed to the economics and mechanics of data entry. In other words, the magnitude of the missing data problem presented above is typical of databases where data entry is manual rather than automated. One obvious solution would therefore involve the elimination of manual data entry. In the case of equipment maintenance databases such as the one described above, a partial elimination of manual data entry could be accomplished by bar-coding of all equipment or by implanting them with electronic transponders. However, some important information such as equipment environment, the person performing the maintenance of the equipment, results of manual tests performed, the function of the equipment,[1] etc. will still have to be manually entered. Further, the equipment which Honeywell and its customers[2] deal with is commonly located in process intensive areas, such as petrochemical refineries, and paper and pharmaceutical manufacturing plants. Equipment installed in these locations is exposed to high levels of corrosion, [3, p. 106], and the bar-codes on such equipment will become rapidly unreadable.

3. Methods for Dealing with Incomplete Data

Incomplete data is an unavoidable problem in dealing with most of the real world data sources, and various methods for dealing with such data have been developed, particularly in the context of missing data in sample surveys. Some of these, such as the multiple imputation methods discussed by Little and Rubin, are based on statistical principles and pay attention to the added uncertainty of inferences obtained from incomplete data [4]. Others, like complete case analysis, are idiosyncratic but practical and therefore widely used. Vach discusses the gap between theory and practise in dealing with missing values, and ways to bridge that gap [5]. In this section, we will present a brief overview of existing methods for dealing with missing data, and comment on their suitability for the equipment maintenance databases such as that described in the previous section. These methods have been roughly grouped following the taxonomy by Little and Rubin [6]:

1. *Ignoring and Discarding Incomplete Records and Attributes.* Complete case analysis, or ignoring records with any missing values, is practical when the number of incomplete records is very small compared to the total number of records, and when analysis of only complete records will not lead to serious biases in inferences obtained from the data. Ignoring incomplete records is usually not an option for industrial databases such as that summarized in Fig. 1. Discarding attributes with missing entries during analysis is not a viable solution when we are interested in making inferences about these attributes. Both complete case and complete variable analysis are wasteful procedures since they usually decrease the information content of the data. Other methods where a missing value is treated as an additional category for each attribute also causes problems during analysis [5]. For example, when two records that have the same field empty are compared for equality it is not always clear whether the result of comparison for that attribute should be true or missing.
2. *Parameter Estimation in the Presence of Missing Data.* Here a model for the complete data is defined, assumptions regarding the missing data mechanism made, and the parameters of the model are estimated by maximum likelihood or maximum aposteriori (MAP) procedures. The missing data mechanism is usually classified as,

   - Missing completely at random (MCAR), when the probability of a record having a missing value for an attribute does not depend on either the observed data or the missing data. Data which is missing due to structural reasons cannot be regarded as MCAR.
   - Missing at random (MAR), when the probability of a record having a missing value for an attribute could depend on the observed data, but not on the value of the missing data itself. Data which is incomplete only due to structural reasons are MAR.
   - Not missing at random (NMAR), when the probability of a record having a missing value for

an attribute could depend on the value of the attribute. Examples include a sensor not detecting temperatures below a certain threshold, people not filling in yearly income in surveys if the income exceeds a certain value.

Maximum likelihood procedures that use variants of the Expectation-Maximization (EM) algorithm [7] can handle parameter estimation in the presence of missing data. Markov chain Monte Carlo methods such as Data Augmentation can estimate posterior distributions for incomplete data [8].

3. *Imputation Procedures.* Imputation is the filling in of missing values for partially missing data. Single imputation is the filling in of a single value for each missing observation. It carries with it the disadvantage that imputing a single value does not capture the sample variability of the imputed value, or the uncertainty associated with the model used for imputation. Multiple imputation procedures suggest multiple, usually likelihood ordered, choices for each missing value [9]. They are computationally more expensive compared to single imputation procedures, but are not associated with the two drawbacks mentioned above. Simultaneous imputation is the filling in of missing values for multiple variables in a record at the same time. We discuss reasons for data completion in greater detail in the next section.

The choice between the different approaches listed above largely depends upon the nature and quantity of the available data, the intended use/users of the data, and the extent and the underlying mechanism of the missing data problem. As mentioned earlier, the first approach of eliminating incomplete records is feasible only in situations where these records constitute an ignorable percentage of the total data, and no significant bias is introduced by their elimination. Model based approaches involving parameter estimation are more suitable for users who are familiar with the missing data mechanism and have the necessary expertise and tools for analyzing the incomplete data. Standard statistical software either deal only with complete data, or treat missing variable values in an idiosyncratic manner. For example, filling in the mean or mode of the observed values seems to be a common practise. Imputation methods are therefore suitable for shared data sets, where all users or analysts may not have the resources for dealing with incomplete data [6, 10].

Imputation of missing data is the focus of this paper. We concentrate on completion procedures for discrete valued variables, such as *manufacturer of a device*, and *function of device* (for instance, sensors, valves, transmitters). We ignore the imputation of missing values for continuous variables since the equipment maintenance databases we deal with tend to have complete entries for them. This is because the measurement of these variables (such as, the deviation from calibration or the linearity properties of a given device) is largely automated and does not require manual data entry by field personnel. For example, deviation from calibration is measured by portable calibrators, which are then used for for re-calibrating the equipment to perform within acceptable limits.

In the next section we talk about reasons for missing data imputation in industrial databases, and review some commonly used techniques for imputation.

4. Imputation of Missing Data

Imputation procedures were first introduced and developed in the context of non-response in surveys, and most of the available literature refers to applications in that domain. (See for example, [9, 11, 12]. The reasons underlying missing value imputation for survey data had to do with the fact that this data was usually shared and prone to have many users, most of whom with access only to methods for analyzing complete data. Hence, it was desirable to have the data completed at the source by persons aware of mechanisms behind the missing data.

In the case of the equipment maintenance databases we deal with, facilitating the use of complete data methods on legacy data bases is an important reason for data completion. These methods include various data visualization packages, data summarization and reporting tools, and modeling tools such as neural nets. For example, all common visualization packages either display complete records alone, and this may lead loss of important information in the data (Fig. 2), or the missing values are mapped to an arbitrary fixed value on the display, for example -1 or 9999, leading to distortion of the scales used in the display.

Notable exceptions to this trend in visualization software are the MANET system [13], and, the work by Swayne which has been implemented in the XGobi software [14, 15]. The MANET software represents the missing data as missing, thereby preserving correlations between the occurrence of missing values in

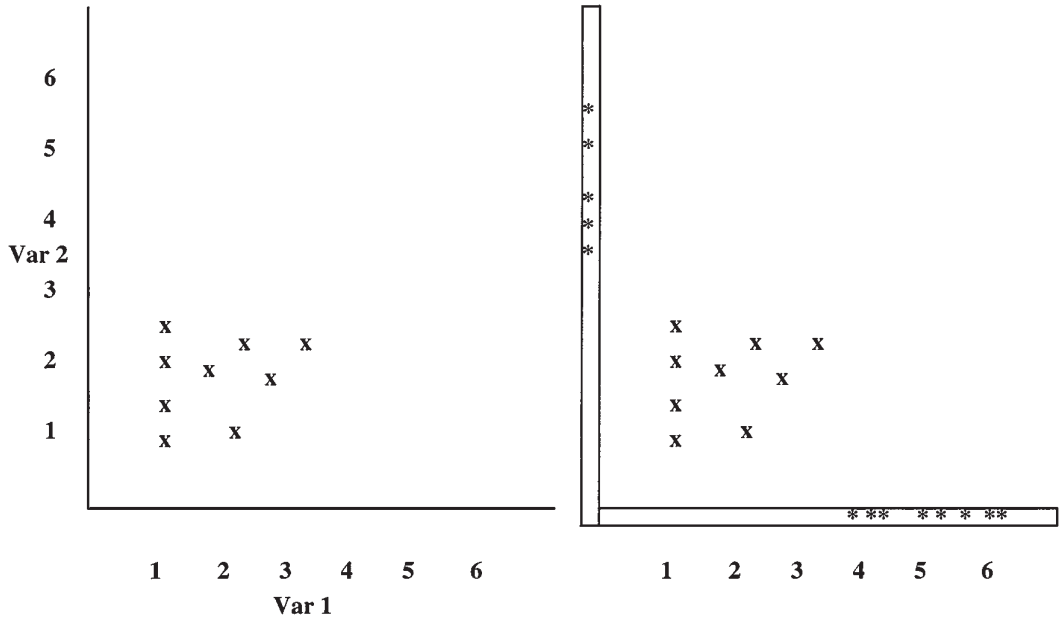

Figure 2. Scatter plots illustrating loss of information by viewing complete records alone. The right plot uses MANET's convention for showing missing values in scatter plots.

one variable and the values of its covariates. This is shown in Fig. 2. In this figure, the plot on the left shows cases where both variable 1 and 2 are recorded. The plot on the right shows cases missing variable 2 below the x-axis at their variable 1 values and, cases missing variable 1 near the y-axis at their variable 2 values. We see from this plot that high values in variable 1 are correlated with missing values in variable 2 and vice versa.

The XGobi system trivially imputes the missing data, for example filling in missing values for an attribute with constant value 20% below its minimum value. In addition, it displays previously completed data sets while keeping track of which values were originally missing and this raises the need for missing data imputation prior to visualization. Unlike visualization tools, data summarization tools (for example, spread sheets) can easily work around missing data, and usually report the frequency and distribution of missing data as part of the summary. In the context of the Honeywell's equipment maintenance data, sample summaries of interest would be:

> Voltmeters of model $abcde$ from manufacturer $M1$ have the lowest average deviation from linearity, while voltmeters of model $vwxyz$ from manufacturer $M2$ have the highest.
>
> The catalytic cracker unit of the plant reports the highest number of failures for temperature sensors, and the sulfur unit temperature sensors have the lowest number of failures this month. The failure information for the alkylation unit has not been recorded.

However, the main purpose of producing reports is to capture the information in the data. If the existing information in covariates can be used to reliably fill in missing data, the summaries generated can be more complete and hence more useful.

A second important reason for imputing missing values is to facilitate parameter estimation and inference from incomplete data. For example, we may wish to test the hypothesis that the mean failure rate of flow meters from manufacturer A is significantly different from the mean failure rate of flow meters from other manufacturers. If the manufacturer is not always recorded, we may be able to use information from other covariates such as model numbers to fill in the missing values with measurable uncertainty. Parameter estimation in the presence of missing data can be done by using complete data methods *after* filling in missing values. A second approach is to use algorithms which fill in missing information and perform parameter estimation such as the Expectation-Maximization algorithm and we

discuss these algorithms elsewhere in the paper. Other than facilitating the use of complete data methods and parameter estimation, data imputation methods address the need for mandatory record completion required (by regulatory agencies) of certain Honeywell customer industries, such as those dealing with the manufacture of pharmaceuticals and petrochemicals. Other applications of data completion methods are discussed later in the paper.

4.1. A Framework for Imputation Procedures

Traditional data completion procedures can be broadly divided into those that are model based, and those based on quasi-randomization inference, [6, 16]. Model based procedures assume that the population quantities of interest are outcomes of random variables, indexed by unknown population parameters. This model can then be used for making inferences about population characteristics from the observed data using maximum likelihood or Bayesian approaches. In the case of incomplete data, the missing data mechanism—whether ignorable, missing at random, missing completely at random, or not missing at random—interacts with the statistical model for the population. Therefore, estimates of model parameters and imputed values for missing data will have to take this into account. Quasi-randomization procedures (or data driven procedures) on the other hand, assume that the population values are fixed, i.e., they are not governed by unknown parameters, and therefore, are not the outcomes of random variables. Further, the distribution determining sample selection is taken as known and whether a sampled unit responds or not is modeled as an outcome of a response random variable with an assumed distribution. We next summarize various available imputation procedures and comment on their important features [6]. contains an analytic discussion of these procedures.

1. Imputation procedures based on quasi-randomization modes of inference.
   (a) *Mean Imputation.* In this method, the missing value for a given attribute in a record is filled in with the mean of all the reported values for that attribute. If the filled in mean is not conditioned on the values of other variables in the record, this procedure is referred to as imputing unconditional means. It carries with it the disadvantage that the variance of the imputed variable and its covariance with other variables are systematically underestimated. Imputing conditional means was introduced by Buck [17]. This method imputes a mean which depends on the values of the recorded variables for the incomplete record. For categorical variables, an alternative to mean imputation is imputation of the mode (the most frequently observed) value for missing values.
   (b) *Hot Deck Imputation.* Here a missing attribute value for a record is filled in with a value from an estimated distribution for the missing value from the current data sample. The main idea behind hot-deck is that for every case for which a variable value is missing, we fill in a value from another *similar* case for which that variable value is available.

      Hot-deck imputation is typically implemented as a two-stage process. The first stage is the classification stage where the data is divided into disjoint, homogeneous clusters. In the second stage, for each incomplete case, the complete cases in its cluster or class are used for filling in missing values. One way of doing this is to choose the most frequently occurring value for that variable within the cluster. Division of data into homogeneous clusters can be done using prior knowledge as is typically done in sample surveys. For example, if one is trying to fill in missing values for reported income, one can divide up the data into clusters based on occupation if that information is available. One of the data completion methods described later in this paper is a hot-deck procedure which uses an automated clustering algorithm to find homogeneous clusters in the data.
   (c) *Cold Deck Imputation.* Similar to hot deck imputation, a cold deck imputation is the filling in of a missing value for a given attribute from a source *other than* than the current data sample. For example, values based on previous data samples could be used in a cold deck procedure.
   (d) *Substitution.* This method is typically used in sample surveys where one unit (case or record) does not respond to the questionnaire. In this situation, this case is replaced by another which was originally excluded from the sample. This method is not applicable in the case of the industrial databases we are dealing with.

2. Model based approaches. As mentioned above these approaches regard the data to be generated by a

model governed by unknown parameters. The process which generates the missing data interacts with the model likelihood function. For the purposes of this paper we regard the missing data mechanism to be *ignorable*, i.e., the missing data are missing at random or MAR as defined earlier, *and*, the data model parameters and the parameters of the missing data mechanism are distributed independently of each other [18]. More complicated interactions between the data generating process and the data censoring mechanism are discussed in [6].

(a) *Regression and Stochastic Regression Imputation.* In *regression imputation* the missing values are predicted by a regression of the unobserved variable values against observed ones for that record or case. In *stochastic regression imputation* a noise term is added to the imputed value. When missing values are not confined to one or two variables, we will need multiple regression equations for each predicted variable, depending on which attributes are observed. This may be well worth the effort by the data collector if the data set is to be shared by multiple users. Machine learning methods such as CART, (Classification & Regression Trees [19], can handle input vectors or cases where the same attributes may not be observed for each case. CART is closely allied to conditional mean imputation which in turn can be considered a special case of regression imputation, and the regression tree produced by CART is really a set of regression equations for each predicted (imputed) variable.

The particular regression model chosen for imputation will depend on the type of variable being predicted. For example, missing values for continuous variables can be predicted using linear or polynomial regression models, whereas binary variable values can be predicted using logistic regression. In the case of categorical data, one can fit log-linear models to contingency tables and predict likelihood ordered choices for the missing value from complete cases. In the industrial databases discussed in this paper, the numbers of variables are large, and some variable subset selection procedure is necessary to produce contingency tables with reasonable cell counts. Also, even though the variable values being imputed are categorical, the databases contain both categorical and continuous variables. In such situations more computationally intense procedures such as the EM algorithm become necessary and this is discussed below.

(b) *Likelihood Based Approaches.* The Expectation-Maximization (EM) algorithm can be used for filling in missing categorical variable values, when the data set is composed of both categorical and continuous valued variables. The E step provides the conditional (on the observed data and current parameter values), expectation of functions of the missing data which appears in the complete data log-likelihood function. The M step performs maximum likelihood estimation of parameters as if there were no missing data. The whole process is repeated as in the case of parameter estimation by likelihood maximization in the absence of missing data. The EM algorithm provides maximum likelihood estimates of $p(Y = c \mid X_1 = x1, \ldots, X_k = xk)$, the conditional probability of categorical variable Y taking on the value c, given the various values of its covariates, $X_1, \ldots, X_k$. These estimates help us achieve likelihood ordered choices for each missing value of Y for various outcomes of its covariates. When the available complete data is small, a strict likelihood based approach may be inappropriate. In such cases, if one has accurate prior knowledge about the data (for example, knowledge of the causal relationships among variables), a Bayesian approach will produce better results than a likelihood based method.

4.2. Machine Learning Methods for Missing Data Imputation

The previous section reviewed traditional data completion procedures in common use by statisticians and data analysts. This section will describe missing data imputation using two machine learning (ML) systems: Autoclass and C4.5. Autoclass is a clustering method used for uncovering intrinsic structure in the data, while C4.5 is a program for learning decision tree classifiers from a set of examples. The rest of this section will describe these two ML tools, discuss how they can be used for data completion, and attempt to place them in the framework presented above.

4.2.1. Autoclass. The first algorithm we explored as a possible data completion tool was Autoclass, a program

developed by Cheeseman et al. to automatically discover clusters in data [20, 21]. It is based on Bayesian classification theory and belongs to the general family of unsupervised learning or clustering algorithms [22]. The choice of Autoclass as a data completion tool was motivated by the fact it could be used to predict different attributes after a single learning session. This makes its use economical in terms of time, and is in direct contrast to supervised learning methods which need to be trained separately for each attribute to be predicted. One interesting feature of Autoclass is that it automatically searches for classes, and has limits preventing data over-fitting, since it trades predictive accuracy versus complexity of classes.

The input to Autoclass is separate files containing the training and test data and a file containing a description of the various fields/features in the data set. This description specifies the distribution of each feature: whether it is continuous or discrete valued and which fields contain missing values. The version of Autoclass that we have used models each continuous variable as being normally distributed, and each discrete variable as having an underlying multinomial distribution. The task of Autoclass is to divide the data set into number of classes, and determine probability distributions for each feature given each class. In order to avoid "over fitting" (in the extreme, assigning each case to its own class), a penalty is assessed for the complexity of the classification. The trade-off between prediction and complexity is accomplished by the fact that each parameter in the model brings its own multiplicative Bayesian prior to the model, thereby lowering the marginal likelihood of the model.

The output of Autoclass consists of the following:

- *The Optimal Classification for the Training Data.* Autoclass can automatically choose the number of classes or it can use a number suggested by the experimenter.
- *A Conditional Probability Distribution of Features Given Classes.* For discrete attributes, the description will specify the conditional probability of each feature value, given each class. In other words, for each class c and feature f, $P(f(x) \mid \text{class}(x) = c)$. For continuous attributes, the conditional probability distribution will be specified by giving the mean and variance of a Gaussian random variable. Autoclass models missing values for attributes as another discrete value, 'failure to observe'. In the case of discrete valued attributes, this would mean that there is yet another possible value which could be observed, namely 'failure to observe'. In the case of independent continuous valued attributes, the value observed could be 'failure to observe' with a probability p, (determined from the data), and a real number with probability $(1 - p)$.
- *A Ranking of the Variables According to Importance in Generating the Classification.* This gives a rough heuristic measure of the "influence" of each attribute used in the classification.
- *A Probabilistic Assignment of Each Case in the Training and Test Data Set to its Class.* That is, for each case, x, and class, c, $P(\text{class}(x) = c)$.

Autoclass does not make point estimates for values of variables or features for the data in a test set. Rather, for each case x, Autoclass provides a probability distribution over the set of classes $\mathcal{C}$: $P(\text{class}(x) = C)$. We may use the conditional probability of features given classes to infer the most likely value or values of missing values of attributes of a case, given its class membership. To illustrate, let us assume that Autoclass after learning on the training set, classifies a test case x, as belonging to C_1 with probability 0.8, and C_2 with a probability of 0.2, based on the non-missing attribute values of x. If the value of a discrete attribute a is missing for this case, we could pick the most probable value of a in C_1 for the training cases, as the predicted value of attribute a for x. Or, we could pick the n most frequent values of attribute a in C_1 for the training cases, as potential candidates for prediction, and in the context of our work, multiple imputation. Since class membership is probabilistic we chose the most likely class of each test case to predict its missing values. An alternative approach would be to predict a distribution over the value space for missing attributes, where the distribution is determined by the case's probabilistic membership in various classes. In the case of a continuous attribute, we could use the mean of the distribution for the class as our prediction. The application involved in this paper involves prediction of missing values for discrete attributes only.

Autoclass regards the data as arising from a mixture model and can therefore be considered a model based procedure when used for data completion. However, the overall strategy used for imputation, namely, dividing the data into homogeneous classes or clusters and sampling within the relevant cluster to fill in missing values, is reminiscent of hot-deck imputation procedures. Hence, one can consider the imputation we do with Autoclass as being both model-based and randomization based.

4.2.2. C4.5. The second machine learning approach to data completion we explored was C4.5, a supervised learning algorithm for decision tree induction developed by Quinlan [2]. C4.5 uses an information-based measure, usually gain ratio, as a splitting criterion in inducing its decision trees. A splitting criterion is a test, usually about the input attribute's value, which partitions the cases into disjoint sets. More details about C4.5's methodology for constructing decision trees can be found in [2]. C4.5 takes as input files containing training (pre-classified) and test cases, and a description of various attributes. Unlike Autoclass, C4.5 can be directly used to predict missing attribute values. This is done by using values of the target attribute (for discrete-valued attributes) which is to be predicted for test cases, as the classes used for training. The training data should therefore have the target attribute value specified. The C4.5 algorithm uses the impurity at each terminal node to determine the class distribution for each input. For example, if x of N cases are assigned to a particular terminal node during training or learning, and if a test case is assigned to the same node, the predicted class distribution for the test case is the actual distribution for the x training cases. We exploit this distribution for multiple imputation by choosing the k most frequent values for the test case target variable from the among the x training cases. C4.5 does not naturally handle continuous variables as target classes. The common way to get around this is to use intervals on the real line as classes for continuous variables. One disadvantage with C4.5 compared to Autoclass, is that each candidate attribute for prediction needs a separate training session.

C4.5 uses a probabilistic approach to handle missing values (these are not the variables designated as target classes) in the training and test data. Any case from the training set is assigned a weight w_i of having outcome O_i for the value of a particular attribute. If the outcome is known and has value O_i, then $w_i = 1$, and, all other outcomes are assigned a weight 0 for this case. If the outcome is missing, then the weight of any outcome O_j for that attribute is the relative frequency of that outcome, among all training cases whose outcomes for this attribute are known. The approach used for the test data is similar. Of course, the target variable/attribute cannot be missing in the case of training data.

C4.5 makes no assumptions regarding the parameters generating the data and hence is not a model-based procedure. It is used for data completion by treating missing value imputation of discrete valued attributes as a classification task. Other decision tree based algorithms such as CART can be considered as model based methods when used for filling in the missing values of continuous valued attributes. This was discussed earlier in the context of regression imputation.

5. Experiments and Results

We conducted several experiments with Autoclass and C4.5 to determine how well they predicted missing values in our experimental data set. For the purpose of the experiments described in this paper, we assumed that data was missing at random, and that the missing data mechanism was ignorable. We also assumed that there was no particular pattern missing, i.e., there was no dependence between the occurrence of missing values for different variables.

Data Used. A subset consisting of 2117 records from the database described in Fig. 1 was used for experimentation. Fourteen attribute fields identified by the domain expert as the most interesting were chosen for the initial analysis. We will refer to this subset of 2117 records as data set A. The target feature chosen for prediction in the first round of experiments was the manufacturer of the device. There were a total of 52 manufacturers represented in the database, of which 30 were represented more than once. Of the 2117 records, only 257 (subset B) had the value for the manufacturer specified, the remaining records had this value missing. Of the 257 records with the manufacturer specified, in 22 cases, the manufacturer was a singleton: there was only one record with that manufacturer in the entire data set. Accordingly, we removed those records from B to get a data set (subset C) of 235 records. Table 1 summarizes the details of the data chosen for the experiments.

Experiment 1. C4.5 was used to learn and predict values for the target variable, manufacturer, on the subset C. Since C4.5 is a supervised classifier it could not be trained on cases where the manufacturer value was missing. A ten-way cross-validation was done to evaluate the accuracy of prediction. In other words, the set C was partitioned randomly into ten subsets of

Table 1. Subsets of data used in experiments.

Subset	Records	ML technique used
A	2117	Autoclass
B	257	—
C	235	Autoclass, C4.5

similar size. Nine of these were used as the training set, and the induced tree was used for predicting on the tenth subset (test set). This process was repeated until all ten subsets were used as test sets. The top choice of the value for the manufacturer variable as well as the top three choices were predicted. In other words, we carried out both single and multiple imputation tasks. The average error over all the test sets was for the single imputation task was 22.6%. For the multiple imputation task, average error over all test sets was 17.4%.

Experiment 2. A 10-way cross-validation similar to that with C4.5 in Experiment 1 was done for Autoclass with subset C. The manufacturer (both the top choice and the top three choices) was predicted for cases in the test set using the approach described in the previous section. The average error on all the test cases in the single choice prediction framework was 51.1%. For the multiple choice prediction task, the average error over all the test sets was 23.5%.

Experiment 3. Autoclass is an unsupervised learning algorithm. Therefore, no target variables need to be specified prior to learning. Further, we have seen that Autoclass can handle missing data during clustering. Hence data where the manufacturer variable value is missing, e.g., the subset (A–C) in Table 1, can be used as part of the training data. The intuition behind adding this subset to the training data is that this subset could contain information about the target variable in the observed values of other variables. Any clustering where this data is included could therefore be more informative about the distribution of the target variable values.

Experiment 3 tests the intuition described above. It is similar to Experiment 2 above in all details, except that the data set (A–C) is added to each training set. Prediction is made on the same subsets as in Experiment 2. The average error on test cases for the single choice prediction task was 41.7%. For the multiple choice prediction task the average error on all the test cases was 17.2%.

The t-test was used to compare the means of the 10-way cross validation results for single prediction and multiple prediction in this experiment and in Experiment 2 above. The results for the single target prediction task were significantly different (at the 2.5% level). The results for the multiple target prediction were only significant at the 10% level. The above results suggest that even in cases where the manufacturer variable was missing there was structural information in the data, which lead to better prediction for missing values.

Experiment 4. Autoclass was used to cluster the 2117 records in data set A. *The manufacturer variable was not used as part of the input feature space.* Autoclass uses the log posterior probability value for a classification to rank alternate models. The best classification produced by Autoclass had 10 classes. When the classification was compared to distribution of manufacturers for the data points in C, it was found that each manufacturer tended to cluster in a few classes. This is illustrated in Fig. 4.

In order to evaluate this clustering the 10-way cross-validation task on the 2117 records as in Experiment 3 was done. *However*, *the manufacturer variable was left out during clustering unlike in Experiment* 3. As a benchmark for performance on this task, a prediction based only on the relative frequency of the manufacturer in the data was made. The top three choices for the manufacturer of each case in C were picked using Autoclass results and directly by relative frequency from the data itself. If the manufacturer for a given data point in C, was one of the top three choices, a hit was scored for that method of prediction. Autoclass scored a hit 82.12% (error rate 17.88%) of the time, while prediction from relative frequency in the data scored a hit 50% of the time. If only the top choice instead of the top three was considered, then, Autoclass scored a hit 53.6% (error rate 46.4%) of the time, while prediction from relative frequency in the data scored a hit 32% (error rate 68%) of the time.

Combining Autoclass and C4.5. One way to combine unsupervised and supervised learning methods is to use the former for feature extraction, and use the extracted features as input to the supervised classifier. Autoclass as mentioned earlier assigns to each case a probabilistic classification. We used Autoclass to classify all the 2117 data points (subset A) after leaving out the manufacturer variable in the database. The best classification produced by Autoclass had ten classes. The class of each data point in C was given as input to C4.5, along with the usual information as in experiment 1 above. The results (error rate = 20.1%) indicate no statistically significant difference, between using the Autoclass class as an input feature, versus not using it as an input feature. (The t-test was used

to compare the means of the 10-way cross validation results for single target prediction in this task and in Experiment 1.) Autoclass ranks the input variables according to their contribution to the classification. When the two most highly ranked variables were removed from the input feature space of C4.5, and the same 10-way cross-validation was done the average error rate (21.3%) was again found to be not significantly different from Experiment 1. This suggests that Autoclass may be used for feature extraction prior to using a decision tree based algorithm to decrease the input feature space for the latter. One interesting effect is that when the Autoclass class of each data point is given as input to C4.5, all the 'best' trees grown have the Autoclass class as the root node.

Other results with Autoclass. In addition to a classification, there are interesting results from Autoclass. One such result is depicted in Fig. 3. Records belonging to (Auto)class or cluster 7, have a higher linearity error, (deviation from linearity is an input feature, and high deviations from linearity are undesirable for the devices in this database), as compared to records in other classes. Note that manufacturers (see Fig. 4) are not distributed uniformly across the classes. We now have an interesting exploratory lead: namely, we would like to test the hypothesis that the devices from manufacturers who fall into cluster 7, could have a significantly different linearity error compared to manufacturers not in that cluster.

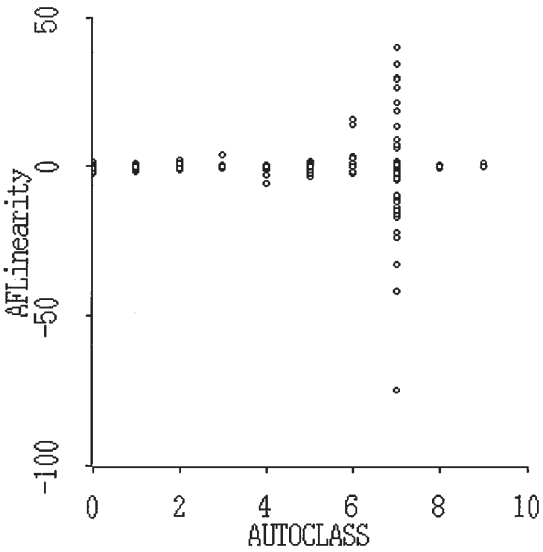

Figure 3. Distribution of linearity error against classes.

6. Discussion

6.1. Conclusions From Experimental Results

The experimental results are summarized in Figs. 5, and 6 and in Table 2. These results describe the imputation of the 'Manufacturer' of the device as a target variable. Other experiments using 'Function', (for example, flow meter, pressure transducer, etc.) 'Housing Material', (this refers to the metal casing a device is placed within for protection against corrosive environments; examples are steel, cast iron, etc.) as target variables produced similar results.

The results shown in Fig. 5 indicate that for the single imputation task, the supervised learning algorithm, C4.5 performed better than Autoclass, which in turn performed better than the benchmark. For the multiple imputation tasks, (Fig. 6) C4.5 and Autoclass performed comparably well, and were both more accurate than the benchmark. The results also indicate that Autoclass uncovers useful, un-anticipated information in the data, (Fig. 3) and that it can be used as a feature extractor or pre-processor of data for other algorithms. One important advantage of Autoclass over C4.5 is that it can be readily used for simultaneous imputation of multiple target variables after a single training session. Decision tree algorithms have to be re-learned for each target variable. However, their high accuracy may make this a worthwhile task when single imputation is being done.

The procedure described in Experiment 4 illustrates how Autoclass can be used for hot deck imputation. Specifically, that experiment is an example of nearest-neighbor hot-deck imputation. Experiments 2 and 3 on the other hand do not fall under the hot-deck category since they use the imputed variable during classification. Most hot deck techniques distinguish between the variables used for classification and those imputed. Keeping these two sets of variables separate is a good idea, especially with an unsupervised classification algorithm since there is the danger of the imputed variable being used to define the clusters or classes. These classes would then not be very useful for classifying incomplete records. The imputed variable should however correlate highly with the classification variables.

6.2. Why Multiple Imputation?

The experiments described above treat data completion as a prediction problem, and the performance metric

Table 2. Comparison of results across experiments.

Expt number	Data	Method	Test error % Single choice	Test error % Multiple choice	Comments
1	C	C4.5	22.6	17.4	
2	C	AC	51.1	23.5	
3	A	AC	41.7	17.2	Improvement over Expt 2.
4	A	AC	46.4	17.88	Manuf. var. not used in training.
Benchmark	C	Mode	68	50	Rel. freq in data.

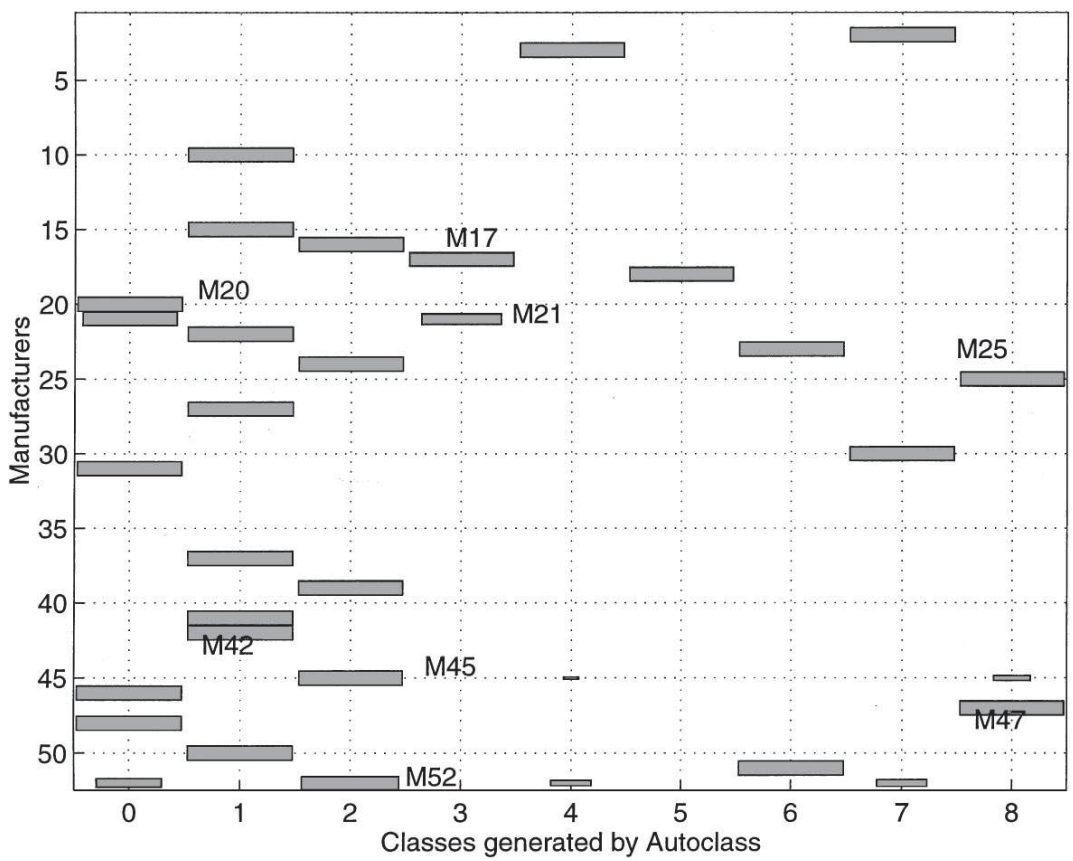

Figure 4. Each manufacturer is clustered in a few classes. The length of the boxes indicate relative proportion of each manufacturer in various clusters/classes.

used is accuracy in predicting missing values. However, the goals of missing data imputation usually extend beyond accurate record completion to further data analysis followed by statistical inference and parameter estimation from the completed records. In the industrial databases we are dealing with, one may for instance be interested in estimating the mean 'linearity error' for each manufacturer, or, we may be interested in predicting the 'Pass/Fail' results of each device as a function of its 'maintenance frequency' or 'location in the plant'. The performance of an imputation procedure therefore depends on how well it captures the

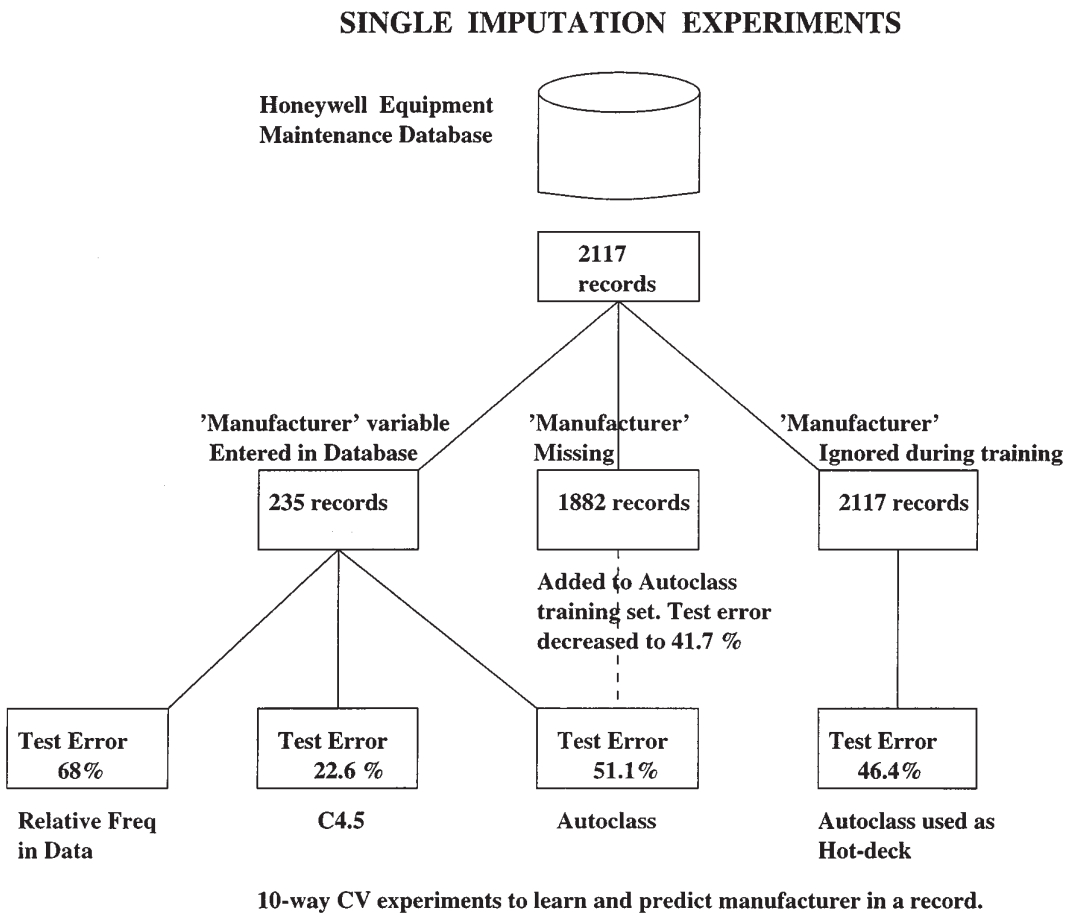

Figure 5. Summary of results from single imputation experiments.

actual distribution of the data, and the extent to which it preserves the existing uncertainty in the data. In this context, multiple imputation has an advantage over single imputation since it preserves the the uncertainty due to missing data. The statistical inferences obtained from combining results from multiply imputed data sets are therefore less biased than those obtained after a single imputation. Following Rubin and Schafer, the *efficiency* of a scalar estimator based on $m \geq 1$ imputations relative to one based on an infinite number of imputations can be defined as,

$$\left(1 + \frac{\gamma}{m}\right)^{-1}. \tag{1}$$

γ is the percentage of missing information relevant to that estimator [9, 18].[3] The maximum possible efficiency is therefore 100% and this occurs when no relevant information is missing or when there is an infinite number of imputations. Assuming that 50% of the relevant information in the data is missing, the efficiency of an estimator based on a singly imputed data set is 66.6%. If the estimator is based on data sets generated by 3 or 5 imputations the efficiency climbs to 85.7% and 90.9% respectively. Hence, even a small number of multiple imputations can increase the efficiency of an estimator based on imputed data.

6.3. *Choosing the Appropriate Imputation Methods*

Any technique used for repairing an incomplete database will be influenced by the type of data involved. The equipment maintenance databases we examined are characterized by the presence of both categorical and continuous variates, and, therefore, any technique

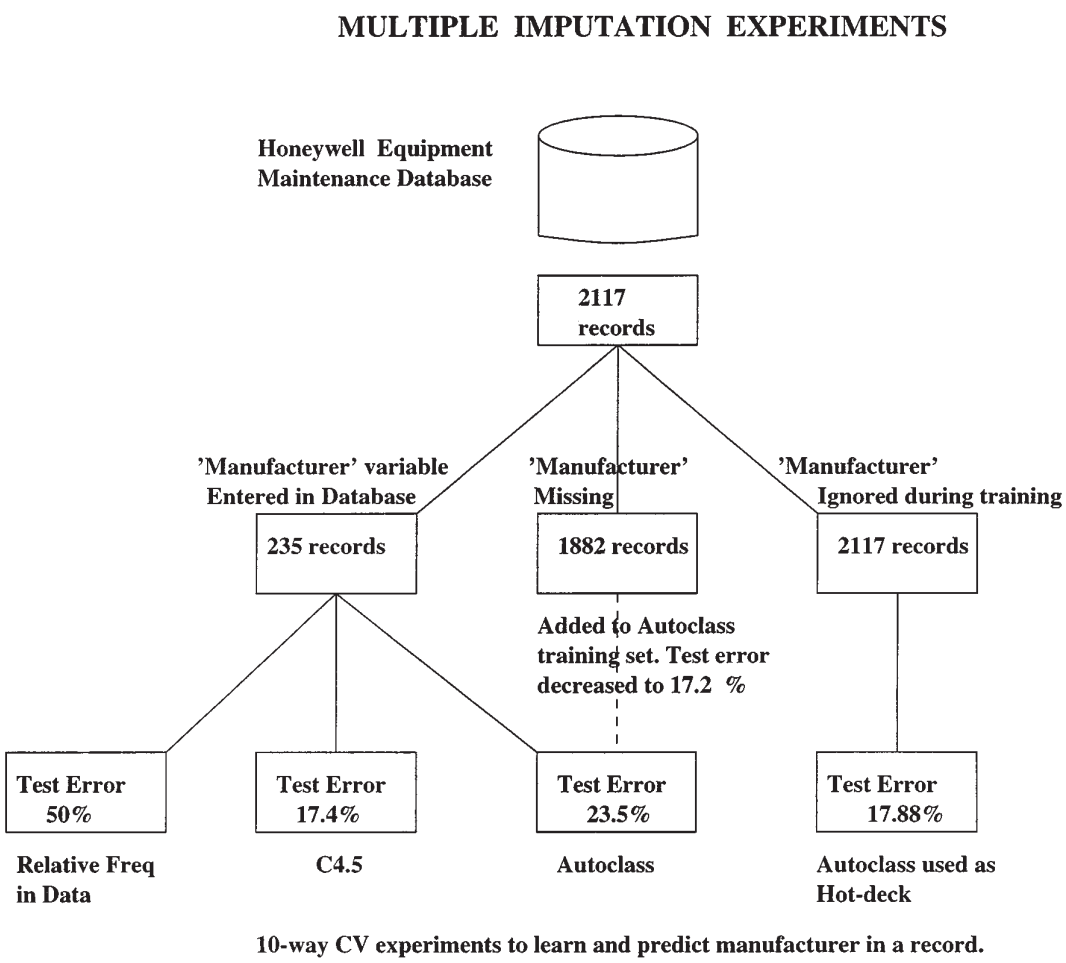

Figure 6. Summary of results from multiple imputation experiments.

for completing these databases should handle mixed data types. This hampers techniques which rely mainly on continuous distance metrics such as regression imputation. One can get around this problem and still attempt regression imputation by replacing each categorical variable by a set of dummy variables which number one less than the number of categories. If a categorical variable value is to be imputed, then, the regression equation would provide estimates of the probabilities falling into the categories represented by the dummy variables. However, we run into the danger of the regression equation predicting values outside the (0, 1) range for the probability estimates, and this limits the usefulness of this approach.

Many databases have high dimensionality and the covariates of a given variable have information relevant for imputation of its missing values. Certain techniques such as imputation of unconditional means (or mode values for categorical variables), do not take covariate information into account, and this could lead to incorrect choices for missing values. Figure 7 illustrates this kind of error which is referred to as internal inconsistency by Ford [12].

Model based procedures such as the EM algorithm can handle both categorical and continuous data types. They are usually reliable though prone to local minima and slow convergence when the proportion of missing data is as high as shown in Fig. 1. Hence, they may not be able to handle the large-scale sparsity typical of industrial databases. Model based imputation procedures make certain assumptions about the data which may or may not be reasonable when applied

	Var 1	Var 2	Var 3	Var 4
1	W	X	Y	??
2	W	X	Y	Z
3	W	X	Y	Z
4	A	B	C	D
5	A	B	C	D
6	A	B	C	D
7	A	B	C	D
8	A	B	C	D
9	W	X	??	??

Figure 7. Internal inconsistency in record completion. We would impute the value D for variable 4 in record 9 if side information from covariates were ignored, and the mode value for that variable was used for imputation.

to the databases commonly used for data mining. For example, a common assumption is that the observations (cases or records) are independent and identically distributed. This is a reasonable assumption in the case of survey data which is usually collected by careful experimental design. Honeywell's equipment maintenance data bases, however, are legacy systems collected for purposes other than statistical analyses, and are therefore quite unlike survey data. Nevertheless, one may accomplish imputation of acceptable accuracy even in the presence of strong modeling assumptions, and we have demonstrated this in the case of multiple imputation with Autoclass.

Machine learning methods such as CART, C4.5 and Autoclass (among others) handle mixed data naturally and are well suited for record completion for databases such as those shown in Fig. 1. These techniques take side information from covariates into account during record completion, are capable of multiple imputation, and are computationally fast. Some of them such as C4.5 do not make strong distributional assumptions about the data, whereas others, such as CART and Autoclass, do. Cobweb/3 [23], a program for conceptual clustering is another machine learning program which appears to be well-suited for filling in missing values.

Our future plans include a comparative evaluation of model based imputation procedures for mixed data, using the data and experimental framework described in this paper. The exact algorithms we are considering are the EM and Data Augmentation methods as described in [18]. Readers interested in software for these algorithms are referred to http://www.stat.psu.edu/j̃ls/.

7. Applications

There are several interesting applications for the procedures we are developing. One application would be a smart data entry system which would directly assist the field personnel gathering data by offering likely options at data entry time. For example, a user entering information about a particular device could be offered a list of manufacturers to choose from. The choices would be ordered by the likelihood of manufacturers given the already entered information. Such a system would reduce the incidence of missing and erroneous entries in the databases and also decrease the cost of data entry. This application is one situation where high multiple imputation accuracy is desirable.

A second application of missing value completion is to render existing databases more useful to analysts. This would allow the generation of more comprehensive summaries and charts (under the completion assumptions) using their regular tools. Most of the research in missing data imputation has been motivated by the need to allow sharing of large incomplete databases among multiple users who are interested in applying complete data methods to the data. See, for example, [10] for a discussion of this issue.

Data cleaning, the detection and rectification of errors, offers a third application. Filled-in-fields of records can be compared to the best guesses of the completion procedure. Outliers can then be examined by human analysts or other software.

8. Summary

Mining industrial databases is often hampered by missing and erroneous data entries. This paper discusses reasons for completing incomplete databases and summarizes well-known statistical approaches for missing data imputation. Novel techniques for data completion based on ML algorithms are presented and empirically evaluated in this paper. One of these techniques is a supervised learning algorithm while the other is an unsupervised clustering strategy. Experiments presented in this paper illustrate how the latter could be used for hot-deck imputation which is a well-known statistical approach for record completion. The paper also enumerates factors which may influence the choice of one data completion procedure over another, and concludes by discussing potential applications of such data imputation techniques.

Acknowledgment

This work was funded by Honeywell Technology Center. Thanks to Brian Parsonnet of Honeywell HiSpec Solutions for insights on the data used in this research.

Notes

1. The same equipment can be configured by the user to perform different functions. For example, a pressure transmitter can be used to measure flow.
2. Honeywell produces a wide variety of control as well as measurement devices.
3. Intuitively, one can think of the efficiency of an estimator for a parameter as the extent to which it captures all the information about that parameter present in the data.

References

1. K. Lakshminarayan, S.A. Harp, R. Goldman, and T. Samad, "Imputation of missing data using machine learning techniques," in *Proceedings: Second International Conference on Knowledge Discovery and Data Mining*, edited by Simoudis, Han and Fayyad, AAAI Press: Menlo Park, CA, pp. 140–145, 1996.
2. J.R. Quinlan, *C4.5 Programs For Machine Learning*, Morgan Kaufmann Publishers: San Mateo, California, 1993.
3. G.A. Greathouse, and C.J. Wessel, *Deterioration of Materials*, Reinhold Publishing Corporation, New York, 1954.
4. D.B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3, pp. 581–592, 1976.
5. W. Vach, "Missing values: statistical theory and computational practise," in *Computational Statistics*, edited by P. Dirschedl and R. Ostermann, Heidelberg: Physica-Verlag, pp. 345–354, 1994.
6. R.J. Little, and D.B. Rubin, *Statistical Analysis with Missing Data*, New York: John Wiley and Sons, 1987.
7. A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. Roy. Statist. Soci.*, vol. B39, pp. 1–38, 1977.
8. M.A. Tanner and W.H. Wong, "The calculation of posterior distributions by data augmentation (with discussion)," *Journal of the American Statistical Association*, vol. 82, pp. 528–550, 1987.
9. D.B. Rubin, *Multiple Imputation for Nonresponse in Surveys*, John Wiley and Sons: New York, 1987.
10. D.B. Rubin, "Multiple imputation after 18+ years," *Journal of the American Statistical Association*, 1996 Jun, vol. 91, no. 434, pp. 473–489, 1996.
11. D.B. Rubin, "Multiple imputation in sample surveys—A phenomenological Bayesian approach to non-response," in *Proceedings Survey Research Methodology Section, American Statistical Association*, 1978, pp. 20–28.
12. B.L. Ford, "An overview of hot-deck procedures," In *Incomplete Data in Sample Surveys, Volume 2, Theory and Bibliographies*, edited by G.W. Madow, I. Olkin, and D.B. Rubin, Academic Press: New York, pp. 185–207, 1983.
13. A. Unwin, G. Hawkins, H. Hofmann, and B. Siegl, "Interactive graphics for data sets with missing values—MANET," *Journal of Computational and Graphical Statistics* , vol. 5, no. 2, 1996.
14. D.F. Swayne and A. Buja, "Missing Data in Interactive High-Dimensional Data Visualization," *Computational Statistics*, vol. 13, no. 1, 1998.
15. D.F. Swayne, D. Cook, and A. Buja, "XGobi: interactive dynamic data visualization in the X window system," *Journal of Computational and Graphical Statistics*, vol. 7, no. 1, 1998.
16. H.L. Oh and F.J. Scheuren, "Weighting adjustments for unit nonresponse," in *Incomplete Data in Sample Surveys, Volume 2, Theory and Bibliographies*, edited by W.G. Madow, I. Olkin, and D.B. Rubin, Academic Press: New York, pp. 143–183, 1983.
17. S.F. Buck, "A method of estimation of missing values in multivariate data suitable for use with an electronic computer," *J. Roy. Statist. Soci.*, vol. B22, pp. 302–306, 1960.
18. J.L. Schafer, *Analysis of Incomplete Multivariate Data*, Chapman and Hall: London, 1997.
19. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Chapman and Hall, 1984.
20. P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "Bayesian classification," in *Proceedings of American Association of Artificial Intelligence (AAAI)*, Morgan Kaufmann Publishers: San Mateo, CA, 1988, pp. 607–611.
21. P. Cheeseman and J. Stutz, "Bayesian classification (autoclass): Theory and results," in *Advances in Knowledge Discovery and Data Mining*, edited by U.M. Fayyad. G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, AAAI Press: Menlo Park, CA, 1996.
22. R. Hanson, J. Stutz, and P. Cheeseman, "Bayesian classification theory," Technical Report, FIA-90-12-7-01, NASA, Ames Research Center, 1990.

23. K. McKusick and K. Thompson, "Cobweb/3: A portable implementation," Report FIA-90-6-18-2, NASA, Ames Research Center, 1990.

Kamakshi Lakshminarayan is a Principal Research Scientist in the Automated Reasoning group of the Honeywell Technology Center where she works on data mining and advanced data visualization. She holds a Ph.D. in Cognitive Science from the University of California, Irvine, and is currently working toward an M.D. from the University of Minnesota.

Steven Alex Harp is a Senior Principal Research Scientist in the Automated Reasoning group of the Honeywell Technology Center, where he works on simulation, data mining and decision support systems. He has received degrees in Statistics (University of Minnesota) and Psychology (Northwestern University).

Tariq Samad is a Principal Research Fellow at the Honeywell Technology Center. His research interests include intelligent control, intelligent systems, and complexity management. He has an undergraduate degree in Engineering and Applied Science from Yale University and Masters and doctoral degrees in Electrical and Computer Engineering from Carnegie Mellon University. He is editor-in-chief of IEEE Control Systems Magazine.