

Imputation of missing data using machine learning techniques

Kamakshi Lakshminarayan, Steven A. Harp, Robert Goldman and Tariq Samad

3660 Technology Drive
Honeywell Technology Center
Minneapolis, MN55418, USA
klakshmi@src.honeywell.com

Abstract

A serious problem in mining industrial data bases is that they are often incomplete, and a significant amount of data is missing, or erroneously entered. This paper explores the use of machine-learning based alternatives to standard statistical data completion (data imputation) methods, for dealing with missing data. We have approached the data completion problem using two well-known machine learning techniques. The first is an unsupervised clustering strategy which uses a Bayesian approach to cluster the data into classes. The classes so obtained are then used to predict multiple choices for the attribute of interest. The second technique involves modeling missing variables by supervised induction of a decision tree-based classifier. This predicts the most likely value for the attribute of interest. Empirical tests using extracts from industrial databases maintained by Honeywell customers have been done in order to compare the two techniques. These tests show both approaches are useful and have advantages and disadvantages. We argue that the choice between unsupervised and supervised classification techniques should be influenced by the motivation for solving the missing data problem, and discuss potential applications for the procedures we are developing.

Introduction

We have experimented with using various machine learning techniques for completing industrial maintenance databases. These databases are usually (in our experience, always) incomplete and contaminated with erroneous data. Tools for completing partial data based on past experience would be useful both as pre-processing for further analysis and to provide assistance to people performing data entry. We have experimented with Autoclass, a Bayesian unsupervised learning method and C4.5, a decision-tree based supervised learning method. We describe our experiments, describe the results and draw conclusions. Both offer viable imputation and may be used in combination.

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 data bases maintained by Honeywell and its divisions. The next section introduces the two machine learning techniques which we have explored as potential data completion approaches, and describes how they could be used for data completion. The first of these two techniques, Autoclass, is an unsupervised clustering strategy due to Cheeseman et al. (1988) which uses a Bayesian approach to cluster the data into classes. The classes so obtained are then used to predict multiple choices for the attribute of interest. The second technique involves modelling missing variables by supervised induction of a decision tree-based classifier, C4.5, due to Quinlan (1993). This predicts the most likely value for the attribute of interest. The next section then presents empirical results from applying these two machine learning techniques for predicting missing data. The last two sections discuss potential applications of the procedures we are developing, related work in statistics respectively.

The Missing data problem

Like many businesses involved in the manufacture and service of complex equipment, Honeywell and its customers compile vast amounts of maintenance data. For a number of reasons, this data is plagued with errors and lacunae. We discuss the type of data with which we are working in this section.

Honeywell and its customers routinely compile maintenance information for plant and building equipment installed in various locations. Entry in these data bases is carried out by field personnel, and for various reasons is plagued by a high proportion of missing 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 data base studied by one of Honeywell's business units, Honeywell Loveland Controls, is shown in Figure 1. This data base contains maintenance information from process control devices. Values for 82 variables or features are recorded in this data base. Some of these variables are symbolic and others are numeric. The variables measure properties

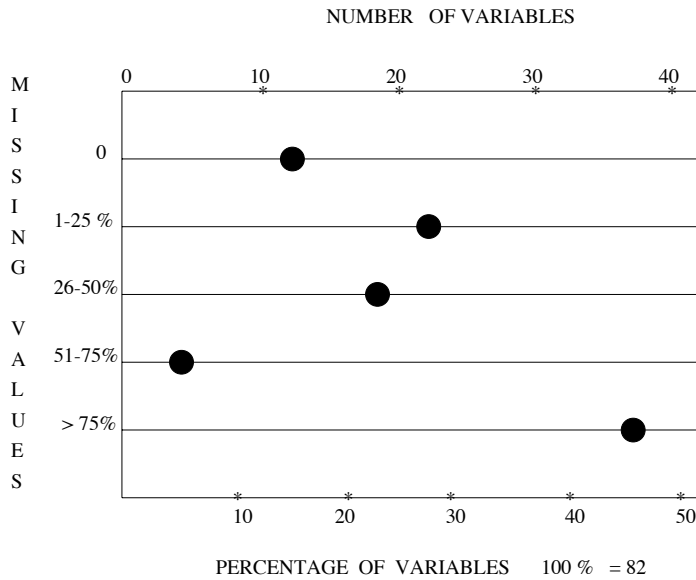


Figure 1: Distribution of variables by missing values. Total number of variables = 82. Total number of records = 4383. Number of complete records = 0.

of the devices such as the manufacturer and model of the device, as well as states of the devices at various times such as calibration error, and pass/fail results of testing. Of the 4383 records in this data base, none of the records were complete and only 41 variables out of 82 have more than 50% of the records complete. This level of incompleteness of the data base seriously limits its usefulness to both analysts and field personnel.

Machine learning techniques for missing data imputation

We experimented with two machine learning (ML) systems: Autoclass and C4.5. These are sophisticated systems for unsupervised and supervised machine learning, respectively. In this section we briefly discuss these two tools.

Autoclass

The first algorithm we explored as a possible data completion tool was Autoclass, a program developed by Cheeseman et al.(1988) to automatically discover clusters in data. It is based on Bayesian classification theory (Hanson, Stutz & Cheeseman 1990) and belongs to the general family of unsupervised learning or clustering algorithms. 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 overfitting, since it trades predictive accuracy versus complexity of classes. Readers interested in more details about this algorithm are referred to Cheeseman & Stutz (1996).

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. Autoclass models each continuously distributed 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 “overfitting” (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. I.e., for each class c and feature f , $P(f(x)|\text{class}(x) = c)$. 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. I.e., for each case, x , and class, c , $P(\text{class}(x) = c)$.

Autoclass does not directly predict 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 . Another possibility is to pick the n most frequent values of attribute a in C_1 for the training cases, as potential candidates for prediction. 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.

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 (1993). 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 an input attribute's value, which partitions the cases into dis-joint sets. More details about C4.5's methodology for constructing decision trees can be found in Quinlan (1993). C4.5 takes as input a 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 the 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. C4.5 does not naturally handle continuous variables as target classes. One way to get around this would be 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 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 out-

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

Table 1: Various subsets of data used in experiments

comes 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.

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 ignored the mechanism of missing data. We also assumed that there was no particular pattern missing, i.e. there was no correlations between the occurrence of missing values for different variables.

Data Used. A subset consisting of 2117 records culled from the data base described in Figure 1, was chosen 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 chosen subset as data set A. The target feature chosen for prediction was the manufacturer of the device. There were a total of 52 manufacturers represented in the data base, 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 culled those records 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 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 average error over all the test sets was 22.6%.

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 was predicted for cases in the test set using the approach described in the previous section. The average error on all the

test cases was 48.7%.

Experiment 3. This 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 was again 48.7%. Considering the results of experiment 2 above, it appears that giving Autoclass the additional set of training cases was not helpful.

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.* The best (Autoclass uses the log posterior probability value for a classification to rank alternate models) classification produced by Autoclass had ten 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 Figure 3. In order to evaluate this clustering the leave one out cross-validation task was done. 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% (error rate 18%) of the time, while prediction from relative frequency in the data scored a hit 50% of the time.

The current version of C4.5 does not lend itself to multiple imputation since, it predicts the only best possible value for the target variable. We are working on extending the algorithm to allow C4.5 to do multiple imputation so that a comparison with Autoclass on the multiple prediction task is possible.

Experiment 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 data base. The classes 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% Table 2), indicate no statistically significant difference, (the t-test was used to compare the means of the 10-way cross validation results in experiment 1 and 5), between using the Autoclass class as an input feature, versus not using it as an input feature.

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 exper-

iment 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 which fall out of the Autoclass classification. One such result is depicted in Figure 2. It is seen here that records classified as belonging to class 7, have a higher linearity error, (deviation from linearity is an input feature, and high deviations from linearity are undesirable for the sensor devices in this data base), as compared to records in other classes. Since manufacturers (see Figure 3) are not distributed uniformly across the classes, in this case we can infer that devices from certain manufacturers are prone to a higher linearity error than others.

Conclusions

Applications of missing data imputation

There are several applications for the procedures we are developing. One application would be to 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. Unsupervised approaches such as Autoclass, approaches are particularly useful for this since:

1. It can be used to predict multiple choices for an attribute;
2. It can be used to predict multiple target variables, unlike decision tree-based algorithms which have to be re-learned for each target variable.

Autoclass performs poorly when predicting a single value for a target variable, although results from experiment 4 indicate that it has a high accuracy when predicting multiple choices for the same target variable.

A second application of missing value completion is to render existing data bases more useful to analysts. This would allow the generation of more comprehensive summaries and charts (under the completion assumptions) using their regular tools. A decision tree algorithm such as C4.5 or a combination of C4.5 and Autoclass (as in experiment 5) would be useful for such data completion due to its high accuracy when predicting single values for missing data.

A third application would be in the detection of erroneous data. Filled-in-fields of records can be compared to the best guesses of the completion procedure. Outliers can then be examined by analysts or other procedures.

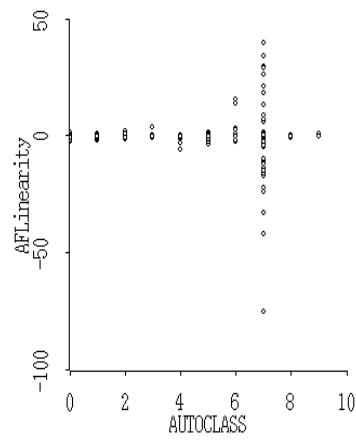


Figure 2: Distribution of linearity error against classes.

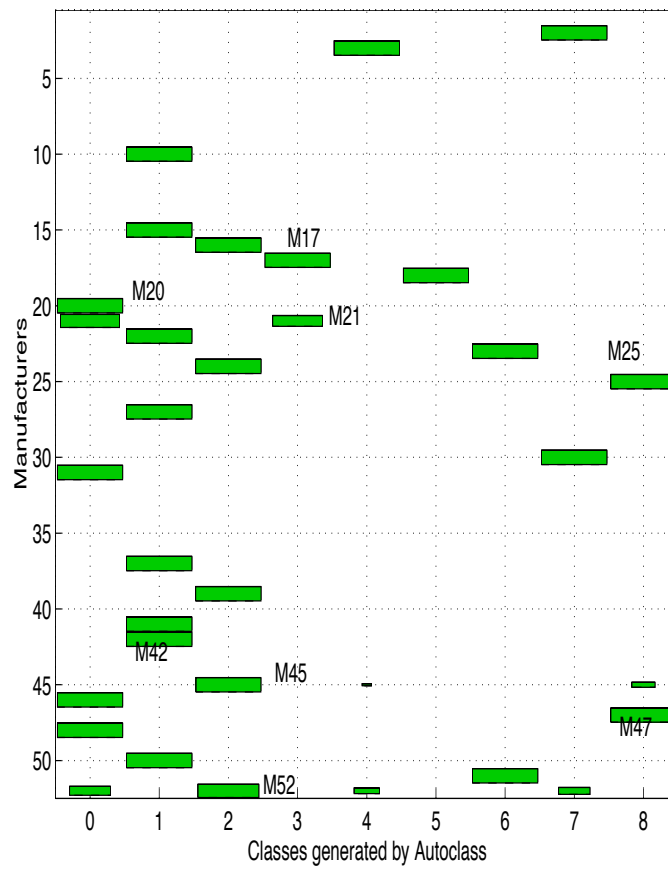


Figure 3: Each manufacturer is clustered in a few classes. The length of the boxes indicate relative proportion of each manufacturer in various classes.

Expt	Data	Method	Test error %	Comments
1	C	C4.5	22.6	Error not different from expt 2. Predict top 3 manufs. Predict top 3 manufs. AC used as additional input feature.
2	C	AC	48.7	
3	A	AC	48.7	
4	A	AC	18	
4	A	Rel. freq in data	50	
5	C	AC + C4.5	20.1	

Table 2: Comparison of results across experiments.

Related work

While the issue of missing data has been addressed in statistical research, most of this work has been directed toward statistical analysis of data with missing values. Some examples of this work are maximum-likelihood techniques, such as the EM algorithm, (Dempster, Laird & Rubin, (1977)). These techniques are helpful in parameter estimation in the presence of missing data, rather than imputation or the filling of missing values, (a.k.a. record completion). Statistical imputation, a less extensively researched field compared to statistical analysis with missing data, encompasses techniques such as mean imputation, regression imputation, hot-deck imputation etc. The former two have the disadvantage that they can be used only in cases where the data is continuous valued, and so cannot be used in cases where the missing data fields pertain to discrete valued attributes. Hot-deck imputation on the other hand can be used for numeric or symbolic valued features. In this method, an imputed value is selected from an estimated distribution for each missing value. This approach carries the same disadvantage as C4.5 in that each attribute needs to be handled individually unlike Autoclass. Currently, we are investigating this approach in order to compare its predictive accuracy with the ML-techniques described in this paper. See Little & Rubin(1987) for an overview of statistical analysis and imputation in databases with missing data.

Summary

We have demonstrated that ML-techniques could be used for missing data imputation in data bases. We have compared the performance of two such techniques, one a supervised classification algorithm, and the other an unsupervised clustering strategy. We also demonstrate how an unsupervised classifier could be used in combination with a supervised classifier. We discussed potential applications of such data imputation techniques and have argued that the choice of an unsupervised versus a supervised technique depends upon the motivation for solving the missing data problem.

References

Cheeseman, P., and Stutz, J. 1996. Bayesian Clas-

sification (Autoclass): Theory And Results In: *Advances in Knowledge Discovery and Data Mining*, Eds. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. Pub. Menlo Park, California: AAAI Press.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. 1988. Bayesian Classification. In Proceedings of American Association of Artificial Intelligence(AAAI), 607-611, San Mateo, California:Morgan Kaufmann Publishers Inc.

Dempster, A.P., Laird, N.M. and Rubin, D.B. 1977. Maximum Likelihood from incomplete data via the EM algorithm (with discussion), *Journal of Royal Statistical Society* B39:1-38.

Hanson, R., Stutz, J. and Cheeseman, P. 1990. Bayesian Classification Theory, Technical Report, FIA-90-12-7-01, NASA, Ames.

Little, R.J., & Rubin, D.B. 1987. *Statistical Analysis with Missing Data*, New York: John Wiley & Sons.

Quinlan, J.R., 1993. *C4.5 Programs For Machine Learning*, San Mateo, California: Morgan Kaufmann Publishers Inc.