Methods for Covering Missing Data in XCS

John H. Holmes

Jennifer A. Sager

Warren B. Bilker

Center for Clinical Epidemiology and Biostatistics
University of Pennsylvania School of Medicine
Philadelphia, PA 19104 USA
jholmes@cceb.med.upenn.edu
wbilker@cceb.upenn.edu

Department of Computer Science
University of New Mexico
Albuquerque, NM 87131, USA
sagerj@cs.unm.edu

Abstract. Missing data pose a potential threat to learning and classification in that they may compromise the ability of a system to develop robust, generalized models of the environment in which they operate. This investigation reports on the effects of three approaches to covering these data using an XCS-style learning classifier system. Using fabricated datasets representing a wide range of missing value densities, it was found that missing data do not appear to adversely affect LCS learning and classification performance. Furthermore, three types of missing value covering were found to exhibit similar efficiency on these data, with respect to learning rate and classification accuracy.

1 Introduction

Learning Classifier Systems (LCS) are used for a number of functions, including agent control and data mining. All of the environments in which LCS operate are potentially plagued by the problem of incomplete, or *missing*, data. Missing data arise from a number of different scenarios. In databases, fields may have values that are missing because they weren't collected, or they were lost or corrupted in some way during processing. In real-time autonomous agent environments, data may be missing due to the malfunctioning of a sensor. In any case, missing data can cause substantial inaccuracies due to their frequency, their distribution, or their association with variables that are important to learning and classification. As a result, missing data has attracted substantial attention in the data mining and machine learning communities [1, 2, 10, 11, 15, 16].

Although one study [12] has investigated the use of a genetic algorithm in analyzing clinical data with missing values, and one other [9] has investigated their use in spectral estimation, the effects of missing data on LCS learning and classification performance have been described only by Holmes and Bilker [8]. They found that missing data adversely affect learning and classification performance in a stimulus-response LCS based on the NEWBOOLE [3] paradigm, and this effect is positively correlated with increasing fractions of missing data. However, no work to date has investigated the effects of missing data on XCS-type LCS, nor on the use of imputation for dealing with these data.

This paper reports on an investigation into the effects of missing data on the classification performance of an XCS-type LCS, EpiXCS, when it is applied to a simulated database with controllable numbers of missing values. Thus, this investigation focuses on the use of LCS in a simulated data mining task, rather than one in agent-based environments. However, the results of this investigation are applicable to a variety of settings wherever missing data are present in the environment.

1.1 Covering in XCS

In the XCS paradigm, the taxon (left-hand side) of an input case that is not matched by any classifier in the population will be *covered*. That is, a copy of the input case will be created and inserted into the population after adding "wild-card" or "don't care" values for specific variables at some probability. Covering occurs perhaps most frequently during the early training phase, although it can occur later, especially if the training set is large. The traditional approach to covering could incur substantial danger of overgeneralization, however, particularly if there are large numbers of missing values in the input case that is being covered.

1.2 Types of missing data

The values of fields in a database can be considered as "responses" to a query, such that for a field such as gender, the value for any given record (or row) in the database reflects a response to the question "What is the gender of [what or whom is represented by the record]?" within the response domain {MALE, FEMALE}. Responses can be *actual*, that is, valid responses within the domain, or they can be *missing*, such that a response value does not exist for that field in the database. Note the important distinction between missing data and erroneous data: missing data are not responsive, while erroneous data are responsive, but not within the response domain.

Missing responses, or more generally, missing data, are typically categorized into one of three types, depending on the pattern of the response [5, 13] on a given field, *x*, and the other fields, *y*, in the database. The first type of missing data is characterized by responses to *x* that are statistically independent of responses to *x* or *y*. That is, the probability of a missing value for x is independent of the value of *x*, as well as of the values of the variables *y*. This type of missing data is referred to as *missing completely at random* (MCAR). An example of MCAR data would be where the value for gender is randomly missing for some cases, but the "missingness" of gender for any particular case is unrelated to the value of *y*, as well as the true, but unknown, value of *x* itself.

A second type of missing data occurs when the probability of a response to *x* is dependent on the response to *y* (or, more simply, the value of *y*). Data such as these are *missing at random* (MAR). An example of MAR data would be where the value for gender is missing when the value of y is at a certain value, or more specifically, if the probability of a missing value for gender is highest when another field, such as race, is equal to Asian. In this case, the missing values for gender are MAR. While the probability of a missing value for gender is essentially random, there is an implicit dependency on race which lessens the degree of randomness of response to the gender field. Thus, it can be seen that MAR data are qualitatively less desirable, and potentially more problematic than MCAR, in analyses and possibly classification.

The last type of missing data is *not missing at random* (NMAR), and these pose the greatest threat to data analysis. NMAR data are found where the probability of a response to *x* is dependent on the value of *x* or a set of data which have not been measured. An example of NMAR data would be where the probability of a missing value for gender is highest when gender is male. NMAR data are not ignorable in a statistical analysis, due to the possibility of extreme bias that may be introduced by them.

In traditional statistical analyses, MCAR and MAR data may be ignorable, depending on the type of analysis to be performed. NMAR data, however, are not ignorable, and must be dealt with using a variety of procedures loosely grouped under the rubric of *imputation*, which calls for the replacement of missing data with statistically plausible values created by means of one of numerous algorithmic approaches [13]. This is the only viable option when there is a large fraction of missing data. However, for cases where the fraction of missing data is small, it may be reasonable to omit cases with missing data only for MCAR. For MAR or NMAR data, omitting these cases will result in uncorrected bias, so even where the fraction of missing data is small, imputation should be considered in analysis, and it is reasonable to assume that it should be considered in using LCS for classification.

This paper is the first in a series to report on the effects of various covering mechanisms to handle missing data. Since MCAR data is arguably the most common, this first paper focuses on covering this type of missing data.

2 Methods

2.1 Data

Generation of the baseline datasets. This investigation used datasets that were created with the DataGen [14] simulation dataset generator. This software facilitates the creation of datasets for use in testing data mining algorithms and is freely available on the Web. Twenty-five baseline datasets were created, each containing 500 records consisting of 10 dichotomously coded (as 0/1) predictor variables and one dichotomously coded (as 0/1) class variable. No missing values were incorporated into the baseline datasets, and although each dataset contained the same number of variables, each was unique in that significant differences existed between the datasets with respect to the distribution of the predictor variables (p>>0.05) and in the association of each predictor with the class variable (p>>0.05).

The baseline datasets were created in such a way as to incorporate noise at a rate of 20%; thus, over the 5,000 variable-record pairs in each dataset, there were 1,000 variables that conflicted or contradicted a putative association with the class variable. This was done to ensure that the dataset was sufficiently difficult in terms of learning and classification. In addition to incorporating noise, the user of DataGen has the capability of specifying the number of expected conjuncts per rule; the higher the number, the more complex the relationships between the predictor variables and the class. For this investigation, the maximum number of conjuncts per rule was set at six. After examining the resulting conjuncts, one of the 10 predictor variables was found to be prevalent in most, or all, of the rules. This variable was used as a candidate missing value, and thus corresponds to *x* that is discussed in Section 1.1. The baseline datasets are described in the Appendix.

Generation of datasets with missing values. From the baseline datasets, separate versions were created to simulate 30 increasing proportions, or *densities*, of missing data, ranging from 2.5% to 75%, in 2.5% intervals. The density of missing data was determined as a proportion of the possible variable-record pairs that result from multiplying the number of possible candidate variables by the number of records (500). In each of the datasets, only one variable was replaced with a missing value. The actual number of records that contained a missing value changed, depending on the missing value density. For example, at 5% density, there were a total of 25 (500*0.05) records with missing values, all in variable *x*. In summary, separate datasets were created at 30 missing value densities for each of the 25 baseline datasets, for a total of 750 datasets; with the addition of the 25 baseline datasets, there were 775 datasets in all. Each of these datasets provided separate pools of data from which training and testing cases were drawn, as described below.

Creation of training and testing sets. Once created, the 775 datasets were partitioned recursively into training and testing sets by randomly selecting records without replacement at a sampling fraction of 0.50. Thus, each training and testing set contaned 250 mutually exclusive records. Care was taken to sample the records so as to preserve the original class distribution, which was 50% positive and 50% negative cases.

2.2 EpiXCS

System description. EpiXCS is an XCS version of EpiCS [7], a stimulus-response LCS employing the NEWBOOLE model [3]. It was developed to apply the XCS paradigm to the unique challenges of classification and knowledge discovery in epidemiologic data. While using the XCS kernel implemented in C by Lanzi (by personal communication), EpiXCS implements several additional variables that tailor the XCS paradigm to the demands of epidemiologic data and users who are not familiar with learning classifier systems. The distinctive features of EpiXCS include a graphical knowledge discovery workbench for parameterization and rule visualization, facilities for handling missing input data, multi-threaded and batch processing, and a methodology for determining risk as a classification metric. EpiXCS uses a variety of test characteristic-based metrics, such as area under the receiver operating characteristic curve and positive and negative predictive values as a means for driving the performance and reinforcement components. Binary, categorical, ordinal, and real data formats are all acceptable, even in the same dataset.

Missing value handling in EpiXCS. Missing values in input (training or testing) data are handled, or *covered*, during creation of the Match Sets ([M]) by one of three mechanisms. Each of these assumes that missing values are specifically encoded as such. It doesn't matter to EpiXCS which codes are used to indicate missing values as long as they do not conflict with the natural coding for a given variable. That is, if a dichotomous variable is normally coded 0 or 1, then a different value must be used to indicate a missing value for that variable, such as 9 or "*".

The first missing value handling mechanism is *Wild-to-Wild*, in which any classifiers in the population that match on the *specific* variables of an input case are added to [M]. The variables of population classifiers that correspond to missing values in the input case are considered matches as well. Thus, an input case consisting of six variables, 001990 (where 9's are missing values), will match (among others) 00#110, ##1010, or 001110 (where #s are "don't cares" or "wilds") in the population. Thus, if these three classifiers exist in the population at that time step, no covering will need to occur. However, if these classifiers do not exist in the population, there are no matches, and the input case will be added to the classifier population as 001##0, where the missing values have now been replaced by the "don't care" symbol. The "Wild-to-Wild" approach is perhaps the most intuitively obvious way of covering cases with missing data.

A second approach uses the mean or mode of the missing variable as a value for covering. This *Population-Average* approach will cover missing data by replacement with the mean (for continuous variables) or the mode (for categorical variables). For example, if an input case consists of six variables:

```
2  2.0  999  38.0  19  4.54
```

where 999 is a missing value for the third variable, and the population mode for that variable at that time step is 394, this variable will be replaced with that value, so that the newly inserted classifier will be:

```
2  2.0  394  38.0  19  4.54
```

The *random assignment* approach replaces the missing value with one randomly selected within the range for the variable with the missing value. For example, if the range for the third variable in the preceding example is 45 to 400 based on the extant classifiers at a given time step, the missing value would be replaced with a value within this range. Categorical variables are preserved to the extent that missing data are not replaced with real values. A variant on this approach, which uses a random number selected within the range of the standard deviation is also available in EpiXCS, but not used in this investigation, which focuses on dichotomous, rather than continuous data.

2.3 Metrics and analytic issues

Several metrics were used to evaluate the learning and classification performance of EpiXCS in this investigation. First, the *area under the receiver operating characteristic curve* (AUC) was used to evaluate evolving classification accuracy during learning and accuracy on classifying novel data. The AUC is preferable to the traditional accuracy metric (usually expressed as percent correct), as it is not sensitive to imbalanced class distributions such as is found in the simulation data used in this investigation [6]. In addition, the AUC represents, as a single metric, the true positive and false positive rate, thereby taking into account the different types of error that can be measured in a two-choice decision problem.

Second, *learning rate* was evaluated by means of a metric, *l*, created specifically for this purpose. This metric was calculated as follows:

$$l = \left(\frac{\text{AUC}_{Shoulder}}{\text{Shoulder}} \right) 1000 \tag{1}$$

Shoulder is the iteration at which 95% of the maximum AUC obtained during training is first attained, and $\text{AUC}_{Shoulder}$ is the AUC obtained at the shoulder. Thus, the higher the value of *l*, the faster is the learning rate. As the first AUC is not measured until the 100th iteration, and the maximum AUC measurable is 1.0, the maximum value of *l* is 10.0. The minimum *l* is 0.0.

Third, the ability of the trained EpiXCS system to classify previously unseen cases of similar genre to the training cases was assessed. This was done by comparing the AUCs obtained at testing across the range of missing value densities. In addition to classification accuracy, as measured by the AUC, it is important to assess the extent to which novel data is unclassifiable, and therefore doesn't factor in to the calculation of the AUC. A metric designed for this purpose, the *Indeterminant Rate* (IR), was used to quantify the proportion of testing cases that could not be classified on testing:

$$\text{Indetermin ant Rate} = \frac{\text{Number of testing cases not classifiab le}}{\text{Total number of testing cases}} \tag{2}$$

These metrics were used in a variety of statistical analyses. To evaluate the effects of missing data on learning performance, the *l*s were correlated by Spearman's rho (*r*) the nonparametric equivalent of Pearson's *r*. The nonparametric test was chosen because the independent variable in the correlation analyses, missing value density, is ordinal. The *l*s were compared, using the baseline dataset as the reference, across the range of missing value densities.

2.4 Experimental procedure

Training. EpiXCS was trained over 2,500 iterations, comprising a *training epoch*. At each iteration, the system was presented with a single training case. As training cases were drawn randomly from the training set with replacement, it could be assumed that the system would be exposed to all such cases with equal probability over the course of the 2,500 iterations of the training epoch. At the 0th and every 100th iteration thereafter, the learning ability of EpiXCS was evaluated by presenting the taxon of every case in the training set, in sequence, to the system for classification. As these iterations constituted a test of the training set, the reinforcement component and the genetic algorithm were disabled on these occasions. The decision advocated by EpiXCS for a given training case was compared to the known classification of the training case. The decision type was classified in one of four categories: true positive, true negative, false positive, and false negative, and tallied for each classifier. From the four decision classifications, the AUC and IR were calculated and written to a file for analysis.

Testing. After the completion of the designated number of iterations of the training epoch, EpiXCS entered the testing epoch, in which the final learning state of the system was evaluated using every case in the testing set, each presented only once in sequence. As in the interim evaluation phase, the reinforcement component and the genetic algorithm, were disabled during the testing phase. At the completion of the testing phase, the AUC and IR were calculated and written to a file for analysis, as was done during the interim evaluations. The entire cycle of training and testing comprised a single *trial*; a total of 20 trials were performed for this investigation for each of the 775 datasets.

Parameterization. EpiXCS was parameterized as described in Butz and Wilson [4], except that the population size was set to 500, which was found empirically to be optimal. Both action set and genetic algorithm subsumption were performed.

3 Results

3.1 Effects of missing data on learning performance

The learning rate of EpiXCS was remarkably stable across all missing value densities. No variance was noted in progressing from low to high densities, indicating that the system is not affected by even high proportions of missing input data during learning. In addition, relatively little variation was found between the three covering methods, as shown in Table 1. In addition, no correlation was found between *l* and missing value density.

Table 1. Learning rate (λ) for each covering method. Values averaged over the 20 runs, and then the 25 datasets at each density. Standard deviation represented in parentheses.

Mode	5.45 (2.77)
Random assignment	5.49 (2.73)
Wild-to-Wild	5.44 (2.69)

2.5 Effects of missing data on classification performance

The evaluation of the effect of missing data on classification performance focused on comparing the various test characteristics obtained on the testing set with missing value density, separately for each type of covering. These characteristics included Sensitivity, Specificity, Area Under the Receiver Operating Characteristic Curve, Indeterminant Rate, Positive Predictive Value, and Negative Predictive value. Virtually no effect of missing data density was observed on classification performance; the mean values for these metrics were virtually identical across the range of densities. Slight differences in the mean values for these metrics were noted between the three covering methods. These differences were not significant, and are shown in Table 2.

Table 2. Test characteristics indicating classification performance on testing data. Values averaged over the 20 runs, and then the 25 datasets at each density. Standard deviation represented in parentheses.

	Mode	Random	Wild-to-Wild
Area under the curve	0.970 (0.021)	0.969 (0.023)	0.969 (0.022)
Sensitivity	0.967 (0.029)	0.966 (0.032)	0.966 (0.031)
Specificity	0.973 (0.026)	0.973 (0.027)	0.973 (0.027)
Positive predictive value	0.973 (0.026)	0.972 (0.026)	0.973 (0.026)
Negative predictive value	0.969 (0.027)	0.968 (0.029)	0.968 (0.027)
Indeterminant Rate	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Correlation of classification accuracy with missing value density. No correlation was found between classfication accuracy, using any of the above test characteristics, and missing value density.

3 Discussion

Table 1 clearly demonstrates that neither missing value density nor covering method affected learning rate, either positively or negatively. This indicates, at least on the simple MCAR datasets used in this investigation, that EpiXCS, and indeed XCS in general, is insensitive to even large amounts of missing data during the training phase in supervised learning environments. However, it is not yet clear what would happen when MAR, or particularly NMAR, data need to be covered. Nor is it clear that this level of learning performance would be seen in larger, more complicated datasets, consisting of large numbers of variables with mixtures of the three types of missing data.

Table 2 demonstrates a similar phenomenon: classification accuracy is essentially not affected by MCAR-type missing data, across a wide range of missing value densities. Some slight differences were observed in the test characteristics using the different covering methods, but it is not clear that this should dictate the use of one covering method over another. In fact, this study indicates that much more investigation is needed into the properties of the three covering methods, and in the face of a wide variety of contrived as well as real datasets.

Limitations of this study. While there is much in this investigation to suggest that EpiXCS is insensitive to missing data in terms of learning rate and classification accuracy, there are several ways to confirm these conclusions. First, only one variables in the data were used as candidates for missing data.. It would be very interesting to extend the patterns of missing data to sets of variables that included more than one each, such that *x* (and/or *y*, when extended to MAR and NMAR data) would have many variables contained within them. It should be noted, however, that doing so would substantially increase the complexity of the analysis, due to the possibility for interactions, so these would have to be handled carefully in creating the datasets.

Second, as noted previously, this study used small datasets. While these provide the basic groundwork for further investigation, much more needs to be done in extending this work to larger and real-world data.

4 Conclusions

This investigation is the first report into the effects of covering missing data on the learning and classification performance in an XCS-based learning classifier system. EpiXCS is insensitive to the missing data used in this study, but this is by no means the end of the story. Even in the face of the results presented here, researchers would be wise to exercise caution when employing LCS in any environment that may contain missing data.

A future task, in addition to researching the effects of covering a wider range of missing value densities and patterns in a variety of datasets and dataset sizes, is to study the effects of imputation on LCS performance. In a real sense, covering is a form of imputation, but it is highly non-traditional in the statistical and machine learning worlds, where missing data are imputed ("covered") even prior to exposure to the system. Thus, an interesting question remains: are standard methods of imputation better than the covering methods described here, or are they superfluous? Either way, the answer to this question has serious implications for the use of LCS in a variety of environments and domains, including maze learning and knowledge discovery, to name two.

References

1. Anand S.S., Bell D.A., Hughes J.G.: EDM: a general framework for data mining based on evidence theory. Data & Knowledge Engineering (1996) 18(3):189-223.

2. Bing, L., Ke, W., Lai-Fun, M., Xin-Zhi, Q.: Using decision tree induction for discovering holes in data. PRICAI'98: Topics in Artificial Intelligence. 5th Pacific Rim International Conference on Artificial Intelligence. Springer-Verlag, Berlin (1998), 182-93.

3. Bonelli, P., Parodi, A., Sen, S., Wilson, S.: NEWBOOLE: A fast GBML system, in: Porter, B. and Mooney, R. (eds.), Machine Learning: Proceedings of the Seventh International Conference. Morgan Kaufmann, San Mateo, CA (1990), 153-159.

4. Butz, M.V. and Wilson S. W. An algorithmic description of XCS. In Lanzi, P. L., Stolzmann, W., and S. W. Wilson (Eds.), Advances in Learning Classifier Systems. Third International Workshop (IWLCS-2000), Lecture Notes in Artificial Intelligence (LNAI-1996). Berlin: Springer-Verlag (2001).

5. Fengzhan, T., Hongwei, Z., Yuchang, L., Chunyi, S.: Incremental learning of Bayesian networks with hidden variables. Proceedings 2001 IEEE International Conference on Data Mining. IEEE Computing. Society, Los Alamitos, CA (2001), 651-2.

6. Holmes J.H.: Quantitative methods for evaluating learning classifier system performance In forced two-choice decision tasks. In: Wu, A. (ed.) Proceedings of the Second International Workshop on Learning Classifier Systems (IWLCS99). Morgan Kaufmann, San Francisco (1999), 250-257.

7. Holmes JH, Durbin DR, Winston FK: The Learning Classifier System: An evolutionary computation approach to knowledge discovery in epidemiologic surveillance. Artificial Intelligence in Medicine (2000) 19(1): 53-74.

8. Holmes JH and Bilker WB: The effect of missing data on learning classifier system classification and prediction performance. Advances in Learning Classifier Systems. Lecture Notes in Artificial Intelligence. Lanzi PL, Stolzmann W, and Wilson SW (eds.). Berlin, Springer Verlag, Vol. 2661: 46-60, 2003.

9. Jui-Chung. H., Bor-Sen, C., Wen-Sheng, H., Li-Mei, C.: Spectral estimation under nature missing data. 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings IEEE, Piscataway, NJ (2001), 3061-4.

10. Kryszkiewicz, M.: Association rules in incomplete databases. Methodologies for Knowledge Discovery and Data Mining. Third Pacific-Asia Conference, PAKDD-99. Springer-Verlag, Berlin (1999), 84-93.

11. Kryszkiewicz, M. and Rybinski, H.: Incomplete database issues for representative association rules. Foundations of Intelligent Systems. 11th International Symposium, ISMIS'99. Springer-Verlag, Berlin (1999), 583-91.

12. Laurikkala J., Juhola M., Lammi S., Viikki K.: Comparison of genetic algorithms and other classification methods in the diagnosis of female urinary incontinence. Methods of Information in Medicine (1999), 38(2):125-131.

13. Little R.J.A. and Rubin, D.B.: Statistical Analysis with Missing Data. John Wiley and Sons, New York, 1986.

14. Melli, Gabor: http://www.datasetgenerator.com/

15. Ng, V. and Lee J.: Quantitative association rules over incomplete data. SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, New York (1998), 2821-6.

16. Sarle W.S.: Prediction with missing inputs. [Conference Paper] Joint Conference on Intelligent Systems 1999 (JCIS'98). Association. for Intelligent. Machinery. (1998), 399-402.

Appendix

Variable-by-variable description of the 25 baseline datasets created by the DataGen generator. Cell values are modes for each predictor variable, for each dataset. The class distribution for each dataset was 50% Class 1 and 50% Class 2.

	Variable									
Dataset	1	2	3	4	5	6	7	8	9	10
1	1	1	0	1	1	1	0	0	0	1
2	1	1	0	0	1	1	0	0	1	1
3	0	0	0	1	1	0	0	0	0	0
4	1	1	0	0	1	0	0	1	0	0
5	1	0	1	0	0	0	1	1	1	0
6	0	0	0	1	0	1	0	0	1	1
7	0	0	0	0	1	0	0	0	0	0
8	1	0	1	0	1	0	0	1	0	1
9	1	0	1	1	1	0	0	1	1	0
10	0	1	1	1	1	1	1	1	0	0
11	1	0	1	0	1	0	0	1	1	0
12	0	0	1	1	0	1	1	1	0	1
13	1	0	0	0	0	0	1	1	0	1
14	1	1	1	1	1	1	0	1	1	1
15	0	1	0	1	0	0	0	1	0	0
16	1	1	0	0	1	1	0	1	1	0
17	1	1	1	0	0	0	1	1	1	1
18	0	1	1	0	0	0	0	1	1	0
19	1	1	1	1	1	1	0	0	0	1
20	0	1	1	1	0	0	0	1	0	0
21	0	1	1	0	1	1	1	1	1	1
22	0	0	0	1	1	1	1	1	0	0
23	0	0	1	1	1	1	1	1	1	1
24	1	0	1	0	1	1	1	1	1	0
25	0	0	0	1	0	0	0	0	0	0