A Comparison of Three 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

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.

The effects of missing data on LCS learning and classification performance have been described previously by only one study, which found that missing data adversely affect learning and classification performance in a stimulus-response LCS based on the NEWBOOLE paradigm, and that this effect is positively correlated with increasing fractions of missing data. However, no work to date has investigated the effects of missing data on learning or classification accuracy in XCS, nor on the use of covering as a type 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. This paper is the first in a series to report on the effects of various covering mechanisms to handle missing data. Since data that are "missing completely

at random" (MCAR) are arguably the most common, this first paper focuses on covering this type of missing data.

Twenty-five simulation 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 ($p>>0.05$). 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. 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. 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.

Missing values in input (training or testing) data were handled, or *covered*, during creation of the Match Sets ([M]) by one of three mechanisms. 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]. A second approach uses the mean or mode of the missing variable as a value for covering. The *random assignment* approach replaces the missing value with one randomly selected within the range for the variable with the missing value. Using each of these approaches separately, EpiXCS was trained over 2,500 iterations and the learning rate calculated. After the completion of training the final learning state of the system was evaluated using every case in the testing set.

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. 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, non-significant differences in the mean values for these metrics were noted between the three covering methods.

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 imputing missing data directly, rather than relying solely on covering for this purpose.