Performance Evaluation of Mutation / Non-Mutation Based Classification With Missing Data

N.C. Vinod
Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India
Dr. M. Punithavalli
Research Supervisor, Manonmaniam Sundaranar University, Tirunelveli, Tamil Nadu, India

***Abstract*—A common problem encountered by many data mining techniques is the missing data. A missing data is defined as an attribute or feature in a dataset which has no associated data value. Correct treatment of these data is crucial, as they have a negative impact on the interpretation and result of data mining processes. Missing value handling techniques can be grouped into four categories, namely, complete case analysis, Imputation methods, maximum likelihood methods and machine learning methods. Out of these imputation methods are the widely used solution for handling missing values. However, there are situations when imputation methods might not work correctly. This study studies and analyzes the performance of two algorithms, one imputation based and another without imputation based classification on missing data.**

Keywords-Missing Values, Imputation, Non-imputation, Classification with missing data.

1. INTRODUCTION

The process of extracting hidden knowledge from databases is called data mining. The data mining methods have great potential in predicting future trends and behaviours, allowing businesses to make proactive, knowledge-driven decisions. It is a multidisciplinary field, drawing work from areas including database technology, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, knowledge-based systems, knowledge acquisition, information retrieval, high-performance computing and data visualization [26]. The era of information explosion is currently encountering a sudden upsurge in both size and number of databases. This increase far exceeds the ability of humans to analyze and extract knowledge from these databases.

A common problem encountered by many database and data mining applications is missing data. A missing data is defined as an attribute / feature in a dataset which has no associated data value stored to it. Correct treatment of missing data is essential as they have a negative impact, if untreated, on the interpretation result. The situation when the missing rate is more than 15%, is a clear indication that the application in question should implement some sophisticated tools to handle them correctly. Several studies have focused on developing algorithms that deal with missing data ([22], [24]). But most of these have some drawbacks when applied to classification tasks. Classification is the process of dividing a dataset into mutually exclusive groups such that the members of each group are as “close” as possible to one another, and different groups are as “far” as possible from one another, where distance is measured with respect to specific variable(s) that are used for prediction. Mishandling missing values during classification may produce erroneous classification results ([1], [5], [11], [25]).

Generally, the methods that deal with missing values can be grouped into four different categories, which are based on the technique used to solve the problem. The first is ‘Complete case analysis’. These algorithms ignore the records with missing values and the analysis is based on the complete data and can only work well with large sized database [24]). The demerit of this approach is that the number of data discarded has a direct impact on the efficiency of classification. The second approach is ‘Imputation method’ [32]. Imputation is a class of procedures that aims to fill in the missing values with estimated ones. The process uses known relationship that exists among the complete values in the dataset to estimate the missing data. After filling the missing data with the estimated value, the classifier learns the modified data set. The third approach is ‘Maximum likelihood method’ [27]. In this method, the input data distribution is modeled by using either Expectation-Maximization (EM) algorithm or by using the variants of EM to handle the missing values. This is followed by the classification task performed by means of the Bayes rule [17]. The fourth approach is the ‘machine learning methods’, which deals with missing values without any imputation and uses techniques such as decision trees [30] and fuzzy neural network [15]. Most of the methods in this category require complete input data matrix for processing and requires considerable efforts.

Out of this imputation methods have been more widely used. Methods like mean and mode imputation [2], hot deck imputation [24], prediction models [32], artificial neural network imputation [13], recurrent neural

network imputation [18], auto-associative imputation [20], k-nearest neighbour imputation [19] and Self-Organizing Map imputation [23] have reported to be successful in handling missing values.

Recently, Garcia-Laencina *et al*. [16] proposed a solution to handle missing data using the popular k-nearest neighbour (KNN) imputation process. This procedure uses a feature-weighted distance metric based on mutual information (MI) to estimate the missing value to improve classification task. MI is defined as a measure of dependence between random variables and has been used in the past as relevance measure in several selection algorithms ([21], [31]). This model is referred to as GL-Model in this paper. The primary objective of this model is to provide a solution to missing data problem by focusing imputation solutions based on the K Nearest Neighbours (KNN) algorithm ([4], [5], [33]). The KNN algorithm is one of the most popular approaches for solving incomplete data problems. This algorithm selects 'K' closest observations (neighbours) from an incomplete dataset, using a distance metric. The selected observations present known values on the features to be imputed. The weighted average of these values is calculated and is used as an estimate for each incomplete feature value. Next, to improve the classification performance, a novel KNN imputation procedure that uses the feature-weighted distance metric is used. The input attribute relevance to Mutual Information (MI) for classification is considered according to the distance metric.

The concept of Mutation might not work in situation like follows. Consider a set of diagnostic data of patients and normal people collected from different hospitals distributed geographically. The missing rate in this type of situation is very high and techniques that are based on feature deletion and imputation strategy are not suitable. The reason behind this is that the missing values may distribute different from the existing ones and these techniques either bring in bias or report inaccurate results. For example, the diagnosis data of patients and normal people have different distributions. To solve this problem, Qu *et al*. [29], proposed a two stage model that avoids mutation and deletion. In phase I, the dataset is divided into disjoint subsets based on the attributes with missing values. In phase II, each subset is used to train appropriate classification algorithms respectively in parallel. This model is referred as QU-Model in this paper.

The main objective of this paper is to compare the performance of the GL-Model and QU-Model with different datasets, whose missing mechanism is Missing At Random (MAR). MAR is the probability of the observed missing pattern, given the observed and unobserved data, does not depend on the values of the unobserved data. This mechanism is common in practice and is generally considered as the default type of missing data. The paper is organized as follows. Section II explains the GL-Model, while Section III explains QU-Model. Section IV presents the experimental results and performs the comparison. Section V presents the summary of the work.

II. GL-MODEL

The KNN method selects 'K' patterns from the full dataset in such a way that they minimize a distance measure. After finding the k nearest neighbours, a replacement value to substitute the missing attribute must be estimated. The method used for estimation depends on the type of data; the mode can be used for discrete data and the mean for continuous data. An improved alternative is to weight the contribution of each neighbour according to their distance to the incomplete pattern whose values will be imputed, giving greater contribution to close neighbours [21]. An advantage over mean/mode imputation and simple hot deck method (in fact, KNNimpute with K = 1) is that the replacement values are only influenced by the most similar cases rather than by all cases or the most similar one, respectively. The main drawback of this approach is that whenever KNNimpute looks for the most similar patterns, the algorithm looks for through all training data set (in the complete data portion), which implies a high computational cost. One important step in KNNImpute is the selection of distance metric. The distance metric used in the GL-model is the heterogeneous Euclidean-overlap metric (HEOM) [5]. HEOM is described as follows: The distance between two input vectors x_a and x_b is denoted as $d(x_a, x_b)$ and is calculated using Equation (1).

$$d(x_a, x_b) = \sqrt{\sum_{j=1}^{n} d_j(x_{aj}, x_{bj})^2} \qquad (1)$$

where $dj(x_{aj}, x_{bj})$ is the distance between x_a, x_b on its jth attribute and is given by Equation (2).

$$d_j(x_{aj}, x_{bj}) = \begin{cases} 1 & (1-m_{aj})(1-m_{bj}) = 0 \\ d_0(x_{aj}, x_{bj}) & x_j \text{ is qualitative} \\ d_N(x_{aj}, x_{bj}) & x_j \text{ is quantitative} \end{cases} \qquad (2)$$

When the input values are unknown, then it is treated missing data and a distance value of 1 is returned in these situations. The overlap function d_0 is assigned a value 0 if the qualitative features are same other wise it is assigned a value 1 (Equation 3) and d_N is the range normalized difference distance function and is given by Equation (4).

$$d_0(x_{aj}, x_{bj}) = \begin{cases} 0 \text{ if } x_{aj} = x_{bj} \\ 1 \text{ if } x_{aj} \neq x_{bj} \end{cases} \quad (3)$$

$$d_N(x_{aj}, x_{bj}) = \frac{|x_{aj} - x_{bj}|}{\max(x_j) - \min(x_j)} \quad (4)$$

where $\max(x_j)$ and $\min(x_j)$ are the maximum and minimum values observed in the N training instances for the jth attribute.

To classify an unlabeled pattern x, the distances from x to the labeled instances are computed, its K nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of x. According to the voting KNN rule, x is assigned to the class represented by a majority of its K nearest neighbours [9]. In the standard KNN algorithm, the K neighbours are implicitly assumed to have equal weight in decision, regardless of their distances to the pattern to be classified. Some approaches have been proposed based on assigning different weights to the K neighbours according to their distances to x, with closer instances having greater weights. In GL-model, the weighted KNN algorithm is referred as KNNclassify. Following the distance-weighted rule proposed by Dudani [12], KNNclassify is implemented using the same weighting procedure described in [7]. Thus, a weight a_k is assigned to each nearest neighbour v_k of x, with k = 1, 2, . . . , K. The nearest neighbour receives a weight of 1, the furthest neighbour a weight of 0, and the remaining neighbours are scaled linearly between 0 and 1. An unlabeled pattern is assigned to the class producing the highest summed weight among its reference neighbours.

In the GL-model, the approach for missing data imputation and classification is a modified KNNImpute. The KNNImpute method is modified because, the conventional method's learning process is not oriented to provide an appropriate imputed dataset for solving classification task. The modified method uses an effective procedure where the neighbourhood is selected by considering the input attribute relevance for classification. For each incomplete pattern, its selected K neighbours are used to provide imputed values which can make the classifier design easier, and thus, the classification accuracy is increased. This approach uses a feature-weighted distance metric based on MI, which is a good indicator of dependence between random variables.

Feature selection algorithms, in general, assign binary weights to features, that is, a weight equal to 1 for selected relevant attributes; a value of 0 for irrelevant features. This method has the advantage of reducing input dimensionality and computation complexity. In GL-model, the feature-weighted procedure assigns one weight per feature according to the MI estimate between each feature and the target class variable. Details of MI estimation can be found in [21]. For classification, the MI measures the amount of information contained in an input feature for predicting the target class variable [10]. A high MI between an input feature and the target means that this feature is relevant, regardless of the classification algorithm. Otherwise, when the shared information between both variables is small, the input feature is irrelevant for the classification task.

In GL-model, the MI concept to weight the input features distances in (1) according to their relevance for classification is sued. The method assigns a weight λ_j (Equation 5) to each jth input feature according to the amount of information that this attribute contains about the target class variable. The scaling factors λ_j is computed heuristically, as given by Weinberger *et al*. [34]. The higher λ_j the more relevant X_j is for classification.

$$\lambda_j = \frac{I(X_j, C)}{\sum_{f=1}^{n} I(X_j, C)} \quad (5)$$

According to the MI concept, the feature-weighted distance metric (Equation) between two input vectors x_a and x_b is computed using Equation (6).

$$d_j(x_a, x_b) = \sqrt{\sum_{j=1}^{n} \lambda_j d_j(x_{aj}, x_{bj})^2} \quad (6)$$

where d_j is the distance defined in Equation (2). When this MI-weighted distance metric is implemented in KNNImpute, the modified MI-KNNImpute is formed. The advantage of this method is that it selects the K-nearest cases by considering the input attribute relevance to the target class, thus adding useful information about classification task during imputation stage and providing missing data estimation oriented to solve the classification task.

The MI between an attribute and target class variable is estimated by the Parzen window method [21]. The MI-KNN classify assigns a weight β_k to the k-th nearest neighbour using Equation (7).

$$\beta_k(x) = \frac{d_I(v_k, x) - d_I(v_k, x)}{d_I(v_k, x) - d_I(v_1, x)} = 1 \text{ when } d_I(v_k, x) = d_I(v_k, x) \quad (7)$$

where $d_I(.)$ is the distance measure based on MI concept defined by (6) and V_x is the set of K nearest neighbours of x arranged in increasing order of distance. During classification, x is assigned to the class for which the weights β_k of the representatives among the K_C nearest neighbours sum to the largest value. This is performed by considering the relevancy between the input features of the target class variable.

III. QU-MODEL

The QU-Model proposes a two-stage algorithm for situations where imputation methods are not applicable. In the first stage, the original dataset is segmented into disjoint sets according to the information gain of the features with missing values (Equation 8). In the second stage, the data in each set obtained from stage 1 is used to train the classifier.

$$\text{Gain}(S, F) = E(S) - \sum_{v \in \{F_N, F_A\}} \frac{|S_v|}{|S|} E(S_v) \qquad (8)$$

where, S is the original dataset, F is the feature set with missing values, F_N and F_A represents the subsets of F depending on whether the value of F is missing (N) or available (A). E(S) is the information entropy of S (Equation 9) and n is number of categories and p_i is the probability that the samples in dataset S belong to the ith class.

$$E(S) = -\sum_{i=`}^{n} p_i \log_2(p_i) \qquad (9)$$

The method used for training during stage I is given in Figure 1.

1. *Given a training dataset D, calculate the information gain of each feature with missing data.*
2. *Select the feature F with the highest information gain. Split D into two subsets DN and DA based on the values of F (missing or available). In other words, DA consists of samples where the values of F are available and DN contains the rest samples. Since the values of F are all missing in DN, F is removed and the dimension of DN is reduced by one. A binary tree T is built in which D is the root node and the child nodes are DA and DN.*
3. *For datasets DA and DN, repeat step 1 and step 2 until there is no missing values in each child node.*

Figure 1 : Training – Stage I

After Stage 1, the dimensions of most subsets are lower than that of the original dataset *D*, which may help improve the performance of the classifiers used in Phase II. In stage 2, the original dataset *D* is split into several subsets *D*i. A separate classifier *C*i is trained on each subset respectively with the classification algorithm given by Libsvm [8]. When the number of features with missing values is high, a minimum missing rate is applied to avoid generating too many subsets (the number of samples in each subset needs to be maintained at a reasonable level for classification).

Testing is performed as below: In Stage 1, for each unknown sample X to be classified find the corresponding subset *D*i in the tree T generated above. In Stage 2, Apply *C*i (trained on *D*i) to assign a class label to *X*.

IV. EXPERIMENTAL RESULTS

The experiments were conducted with the objective of testing the GL and QU models in their efficiency in solving classification task with incomplete and complete data. To compare the results, the KNNImute approach, GL-model and QU-model was tested with the C4.5 classifier. The efficiency of the algorithm was analyzed using the classification accuracy and time taken to classify. The real world datasets used are Abalone, Credit Approval and Annealing datasets and were obtained from UCI repository [28]. The 10-fold cross validation was used [6] was used in all experiments. Abalone [3] is a multi-class dataset and for the sake of clarity, a new two-class dataset was created by selecting a subset of samples and merging different classes. The resulting dataset was a relatively balanced one containing 689 class 1 samples and 709 class 2 samples. There was a feature 'sex' in the dataset with three possible values: 'M', 'F' and 'I' and all 'sex' values marked with 'I' were regarded as missing data. As a result, the incomplete dataset had 578 samples with missing values. Credit card dataset (http://sede.neurotech.com. br:443/PAKDD2009) contain customer information and credit levels (2-class problem). A subset of the original data was selected based on shops (ID_Shop) and then the values of the feature 'Months_In_Job' were discarded for some selected shops to create an incomplete dataset. This was to simulate the real world scenario where certain data from specific shops were missing. Finally, there were 10384 samples and 3171 of them had missing values. The annealing dataset [14] is a multi-class dataset having 798 instances having 38 attributes. In the dataset, the present of the character '-' in any of the attribute denotes missing values,

resulting in an incomplete dataset. A subset of the original data was selected based on family type (family), totaling to 70 cases of missing data.

Table I shows the classification accuracy for the three datasets selected. The experiments were conducted with subset selection and full dataset.

Table I. Classification Accuracy

Dataset	C4.5	GL Model	QU Model
Abalone	78.36	83.12	87.34
CreditApproval(subset)	81.44	72.81	81.96
CreditApproval (full dataset)	80.23	84.36	80.55
Annealing (subset)	67.47	71.79	84.33
Annealing (full dataset)	65.12	83.41	81.65

While taking the execution time into consideration, the QU-algorithm was fast in producing classification results than GL-Algorithm. But it could be seen that the C4.5 algorithm without missing value handler was the quickest among all the three. The reason is that the additional algorithm requires a negligible time to handle the incomplete data during classification. But this difference is very minimal (on average 0.09% with GL-Model and 0.04% with QU-Model) and the performance of accuracy obtained by the GL and QU Models outweighs this fact.

During experimentation, while the full dataset with missing data was considered and it could be seen that the GL-Model produced better classification accuracy. While a subset of the dataset was considered, in Credit Approval dataset, the sex attribute took only two nominal values, the mutation algorithm's performance degraded and QU-Model ranked better than GL-model. In all the cases, classifier without missing value handler produced poor result. Figure 2 shows the time complexity of the proposed algorithms.

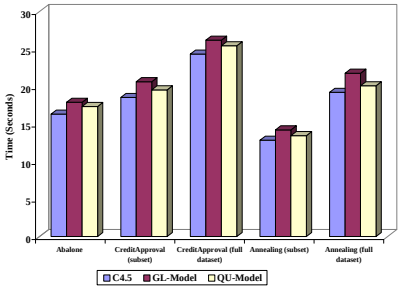

Figure 2 : Performance based on Classification Time

V. CONCLUSION

The problem of missing data in databases creates problems in almost all steps of data mining. In this paper, a method based on KNN combined with imputation and a method which uses a two stage approach without imputation are analyzed. The first method uses a feature-weighted distance metric combined with KNN classifier to handle incomplete data for classification. The second method first divides the dataset into disjoint subsets according to the attributes with missing values. Using these subsets the classification process is performed. Several experiments were conducted and the results prove that while both the models perform well with missing data, when the attribute value is binary, the QU-model performs better than GL-model. It would be worthwhile to pursue research into developing hybrid model that combines the GL and QU-models by applying the imputation method to the subsets in the first stage and then perform classification and to study and analyze their effect on classification.

REFERENCES

[1] Acuna, E. and Rodriguez, C. (2004) The treatment of missing values and its effect in the classifier accuracy, in: D. Banks, L. House, F.R. McMorris, P. Arabie, W. Gaul (Eds.), Classification, Clustering and Data Mining Applications, Springer, Berlin, Pp. 639–648.

[2] Allison, P.D. (2001) Missing data, Sage University Papers Series on Quantitative Applications in the Social Sciences, Thousand Oaks, California, USA.

[3] Asuncion, A. and Newman, D.J. (2007) UCI Machine Learning Repository, University of California.

[4] Batista, G.E. and Monard, M.C. (2002) A study of k-nearest neighbour as an imputation method, in: Second International Conference on Hybrid Intelligent Systems, vol. 87, Santiago, Chile, 2002, pp. 251–260.

[5] Batista, G.E. and Monard, M.C. (2003) An analysis of four missing data treatment methods for supervised learning, Applied Artificial Intelligence, Vol. 17, No.5–6, Pp. 519–533.

[6] Bishop, C.M. (1995) Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK.

[7] Brown, J.G. (2002) Using a multiple imputation technique to merge data sets, Applied Economics Letters, Vol. 9, No. 5, Pp.311–314.

[8] Chang, C. and Lin, C. (2001) LIBSVM: a library for support vector machines.

[9] Cover, T.M. and Hart, P.E. (1967) Nearest neighbor pattern classification, IEEE Transactions on Information Theory, Vol. 13, No.1, Pp. 21–27.

[10] Cover, T.M. and Thomas, J.A. (1991) Elements of Information Theory, Wiley-Interscience, New York.

[11] Duda, R.O., Hart, P.E. and Stork, D.G. (2000) Pattern Classification, Wiley-Interscience, New York.

[12] Dudani, S.A. (1976) The distance-weighted k-nearest-neighbor rule, IEEE Transactions on Systems, Man and Cybernetics, Vol. 6, No.4, 325–327.

[13] Francois, D., Rossi, F., Wertz, V. and Verleysen, M. (2007) Resampling methods for parameter-free and robust feature selection with mutual information, Neurocomputing, Vol. 70, Issue 7–9, Pp. 1276–1288

[14] Frank, A. and Asuncion, A. (2010) UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science.

[15] Gabrys, B. (2002) Neuro-fuzzy approach to processing inputs with missing values in pattern recognition problems, International Journal of Approximate Reasoning, Vol. 30, No.3, Pp. 149–179.

[16] Garcia-Laencina, P.J., Sancho-Gomez, J., Figueiras-Vidal, A.R.C and Verleysen, M. (2009) K nearest neighbours with mutual information for simultaneous classification and missing data imputation, Journal Neurocomputing, Elsevier Publications, Vol. 72. Issue 7-9, Pp. 1483-1493.

[17] Ghahramani, Z. and Jordan, M.I. (1994) Supervised learning from incomplete data via an EM approach, J.D. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in NIPS, Vol. 6, Morgan Kaufmann Publishers, Inc., Los Altos, CA, Pp. 120–127.

[18] Hammer, B. and Villmann, T. (2002) Generalized relevance learning vector quantization, Neural Networks, Vol. 15, Issues 8–9, Pp. 1059–1068.

[19] Hechenbichler, K. and Schliep, K. (2007) Weighted k-nearest-neighbor techniques and ordinal classification. Technical Report, Ludwig–Maximilians University Munich.

[20] Jerez, J.M. , Molina, I., Subirats, J.L. and Franco, L. (2006) Missing data imputation in breast cancer prognosis, in: BioMed'06: Proceedings of the 24th IASTED Interna- tional Conference on Biomedical Engineering, ACTA Press, Anaheim, CA, USA, Pp. 323–328.

[21] Kwak, N. and Choi, C.H. (2002) Input feature selection by mutual information based on Parzen window, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No.12, Pp. 1667–1671.

[22] Lall, U. and Sharma, A., (1996) A nearest-neighbor bootstrap for resampling hydrologic time series, Water Resource. Res., Vol.32, Pp.679–693.

[23] Little, R.J.A. (1999) Methods for handling missing values in clinical trials, Journal of Rheumatology, Vol. 26, No. 8, Pp. 1654–1656.

[24] Little, R.J.A. and Rubin, D.B. (2002) Statistical Analysis with Missing Data, 2nd Edition, John Wiley and Sons, New York.

[25] Markey, M.K. and Patel, A. (2004) Impact of missing data in training artificial neural networks for computer-aided diagnosis, in: International Conference on Machine Learning and Applications, Pp. 351–354.

[26] Martin, T. (2003) A day in the life of a Data Miner, Bulletin of the International Statistical Institute, 54th Session, Vol. LX, Invited Papers, August 2003, Berlin, Germany. Pp. 298-301.

[27] McLachlan, G.J. and Krishnan, T. (1997) The EM Algorithm and Extensions, Wiley, New York.

[28] Newman, D.J., Hettich, S., Blake, C.L. and Merz, C.J. (1998) UCI repository of machine learning databases.

[29] Qu, X., Yuan, B. and Liu, W. (2009) A Novel Two-Phase Method for the Classification of Incomplete Data, International Conference on Information Management, Innovation Management and Industrial Engineering, Pp. 452 – 455.

[30] Quinlan, J.R. (1993) C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA.

[31] Rossi, F., Lendasse, A., Francois, D., Wertz, V. and Verleysen, M. (2006) Mutual information for the selection of relevant variables in spectrometric nonlinear modeling, Chemometrics and Intelligent Laboratory Systems, Vol. 80, Pp. 215–226.

[32] Schafer, J.L. (1997) Analysis of Incomplete Multivariate Data, Chapman & Hall, Florida, USA.

[33] Troyanskaya, O., Cantor, M., Alter, O., Sherlock, G., Brown, P., Botstein, D., Tibshirani, R., Hastie, T. and Altman, R. (2001) Missing value estimation methods for DNA microarrays, Bioinformatics, Vol. 17, No.6, Pp. 520–525.

[34] Weinberger, K., Blitzer, J. and Saul, L. (2006) Distance metric learning for large margin nearest neighbor classification, in: Y. Weiss, B. Scholkopf, J. Platt (Eds.), Advances in NIPS 18, MIT Press, Cambridge, MA, Pp. 1473–1480.