Experimental analysis of methods for imputation of missing values in databases

Alireza Farhangfar [a] , Lukasz Kurgan [b] , Witold Pedrycz [c]
[a] IEEE Student Member (farhang@ece.ualberta.ca)
[b] IEEE Member (lkurgan@ece.ualberta.ca)
[c] IEEE Fellow (pedrycz@ee.ualberta.ca)
Electrical and Computer Engineering Department
University of Alberta, Edmonton, AB, Canada, T6G 2V4

ABSTRACT

A very important issue faced by researchers and practitioners who use industrial and research databases is incompleteness of data, usually in terms of missing or erroneous values. While some of data analysis algorithms can work with incomplete data, a large portion of them require complete data. Therefore, different strategies, such as deletion of incomplete examples, and imputation (filling) of missing values through variety of statistical and machine learning (ML) procedures, are developed to preprocess the incomplete data. This study concentrates on performing experimental analysis of several algorithms for imputation of missing values, which range from simple statistical algorithms like mean and hot deck imputation to imputation algorithms that work based on application of inductive ML algorithms. Three major families of ML algorithms, such as probabilistic algorithms (e.g. Naïve Bayes), decision tree algorithms (e.g. C4.5), and decision rule algorithms (e.g. CLIP4), are used to implement the ML based imputation algorithms. The analysis is carried out using a comprehensive range of databases, for which missing values were introduced randomly. The goal of this paper is to provide general guidelines on selection of suitable data imputation algorithms based on characteristics of the data. The guidelines are developed by performing a comprehensive experimental comparison of performance of different data imputation algorithms.

Keywords: Incompleteness, missing values, imputation, preprocessing, machine learning

INTRODUCTION

Most of the real world databases are characterized by an unavoidable problem of incompleteness, in terms of missing or erroneous values. A variety of different reasons result in introduction of incompleteness in the data. Examples include manual data entry procedures, incorrect measurements, equipment errors, and many others. Existence of errors, and in particular missing values, makes it often difficult to generate useful knowledge from data, since many of data analysis algorithms can work only with complete data. Therefore different strategies to work with data that contains missing values, and to impute, or another words fill in, missing values in the data are developed [9].

1. Methods for dealing with missing values

In general two groups of algorithms used to preprocess databases that contain missing values can be distinguished. First group concerns unsupervised algorithms that do not use target class values. Second group are supervised algorithms that use target class values, and which are most commonly implemented by using supervised ML algorithms [11].

The unsupervised algorithms for handling missing data range from very simple methods like Mean imputation to statistical methods based on parameter estimation. Several simple algorithms are characterized and described by Little and Rubin [10]. They include methods where:

- incomplete examples are ignored and discarded,
- parameter estimation in the presence of missing data and imputation procedures are performed.

The approach of deleting examples that have missing values, or ignoring attributes with missing values, is the most trivial way of handling missing values. However, this is a practical and computationally low cost solution that can be

applied to databases that have small amount of missing or erroneous values. In case of applying it to databases with considerable amount of missing values, significant amount of useful information will be deleted. This may result in severe decrease of quality of data, and even inability to use the data for analysis purposes. The second approach works based on parameter estimation. The data is described based on models and their parameters, which are estimated by maximum likelihood or maximum a posteriori procedures (MAP) that use variants of the Expectation-Maximization (EM) algorithm [6]. The unsupervised statistical methods also include Hot Deck and Mean imputation, which are explained later. Another unsupervised method, in which a missing value is considered as a new meaningful value for each attribute often leads to problems during data analysis [17]. For example, the result of comparison between two examples that have missing values in the same attribute is not clear.

The supervised algorithms usually use ML algorithms for preprocessing of databases that contain missing values. The ML algorithms are used to generate data model and perform classification task with the data that contains missing information. The results of the classification are used to impute missing values. Several kinds of ML algorithms can be used, such as decision trees, probabilistic, and decision rule [3], but the underlying methodology remains the same.

2. The objective

Different studies have been conducted in the subject of imputation of missing values [9], however none of them were comprehensive enough to give an overall view of the best method for a given database type. In this study we use five different imputation algorithms to impute artificially created missing information. The experiments are used to perform comparison between different imputation algorithms, and most importantly develop strategies for selection of the best performing imputation algorithm based on a set of characteristics that are derived for a given database.

The selected imputation algorithms include two unsupervised algorithms: Mean and Hot Deck, and three supervised imputation algorithms. The supervised imputation algorithms include those that work based on the C4.5 ML algorithms, which is a decision tree algorithm, the CLIP4 ML algorithm, which is a decision rule algorithm, and the Naïve-Bayes ML algorithm, which is a probabilistic algorithm. Database characteristics are obtained using certain number of measures, which are similar to those in [2], and [12]. They include the following:

- Number of examples,
- Number of attributes,
- Proportion of Boolean attributes,
- Number of classes.

The obtained results are used to develop guidelines for selection of best imputation algorithm for a database described by a given set of characteristics. Such guidelines can be used by a Meta learning procedure that is used to oversee a more general data preprocessing process.

The remaining of the paper is organized as follows. Sections 3 and 4 provide details on imputation methods that are used in this paper. Section 5 briefly discusses different mechanisms that lead to introduction of missing values in databases. Sections 6 and 7 describe experiments that apply the imputation methods to seven selected databases. First, missing values are introduced in the databases. Next, imputation of missing data is performed and the imputed values are compared with original values, and accuracy of the imputation is computed. The results for supervised imputation methods are also expressed in terms of sensitivity, specificity, and predictive accuracy of the underlying ML algorithms [3]. The results are presented in terms of graphs showing performance of imputation algorithms against specific characteristics of input databases to enable easy and efficient analysis of the results. Finally, the results are analyzed and the discussion and conclusions are presented in sections 8 and 9.

METHODS FOR IMPUTATION OF MISSING VALUES IN DATABASES

Data imputation methods used in this paper include two unsupervised imputation algorithm: Mean imputation and Hot Deck imputation, and three supervised ML-based imputation algorithms. First, the unsupervised imputation algorithms are described. Next, both supervised imputation methods and the underlying ML algorithms are briefly introduced. Finally, mechanisms that lead to introduction of missing values in the data are described.

3. Unsupervised imputation algorithms

3.1. Mean imputation

In this method, mean of values of an attribute that contains missing data is used to fill in the missing values. In case of a categorical attribute, the mode, which is the most frequent value, is used instead of mean. The algorithm imputes missing values for each attribute separately.

3.2. Hot-Deck imputation

In this method, for each example that contains missing values, the most similar example is found, and the missing values are imputed from that example. If the most similar example also contains missing information for the same attributes as the missing information in the original example, then it is discarded and another closest example is found. The procedure is repeated until all missing values are successfully imputed or entire database is searched.

There are several ways to find the most similar example to the example with missing values [13]. The distance function that is used to measure similarity between different examples assumes distance of 0 between two attributes if both have the same numerical or nominal values, otherwise the distance is 1. The distance of 1 is also assumed for an attribute, for which any of the two examples has a missing value. For example, we consider a database described by 4 attributes that has two examples with the same value for the first attribute, different values for the rest of the attributes, and a missing value for the fourth attribute in one of the examples. In this case, the distance between these two examples is 2. This distance function is suitable only in case of data that contains discrete attributes. The distance for continuous attributes must use other formula, such as Euclidian or Manhattan distance.

4. Supervised imputation algorithms

Another way of imputing missing values is by using ML algorithms. The imputation is performed by performing multiple classification tasks using a ML algorithm. Each classification task is performed in two steps. First, during the learning step the ML algorithm generates the model using learning data. The data model is used to classify examples into a set of predefined classes, which in case of missing value imputation are just all distinct values of an attribute that has missing values. Second, during the testing step, the generated model is used to impute missing data for the testing data, which was not used during learning. The detailed procedure to impute missing values using ML algorithms follows.

First the attributes that contain missing values are determined. Each such attribute is treated as target (class) attribute in turn, which means that classification task is performed as many times as the number of attributes that contain missing values. Next, the data is divided into training and testing parts. All examples that have a non-missing value in the attribute that is selected as the target attribute are placed in the training set. The remaining examples, i.e. those that have missing information in the target attribute, are placed in the testing set. Next, the ML algorithm is used to generate data model using the training data. The model is applied to the testing data, and classification task is performed to predict values of the target attribute. The predicted values are imputed for the missing values. For each classification, i.e. imputation performed for each attribute that contains missing values, the results in terms of sensitivity, specificity and accuracy of the classification, are recorded. Next, another attribute that contains missing values is selected and the process repeats until all attributes are considered. Finally, the average value of sensitivity, specificity and accuracy across all attributes is computed.

Figure 1 is used to illustrate the above procedure. A database, which contains missing values in attributes 1 and 2, is shown in Figure 1a. First, the attribute 1 is assumed to be the target attribute. The database is divided into training and testing dataset, as shown in Figures 1b and 1c. The training dataset is used to develop a data model, which is applied on the testing data to predict the values of the target attribute. The Figure 1d shows that the attribute 1 will use predicted values to impute corresponding missing values. Next, the same procedure will be repeated for the attribute 2.

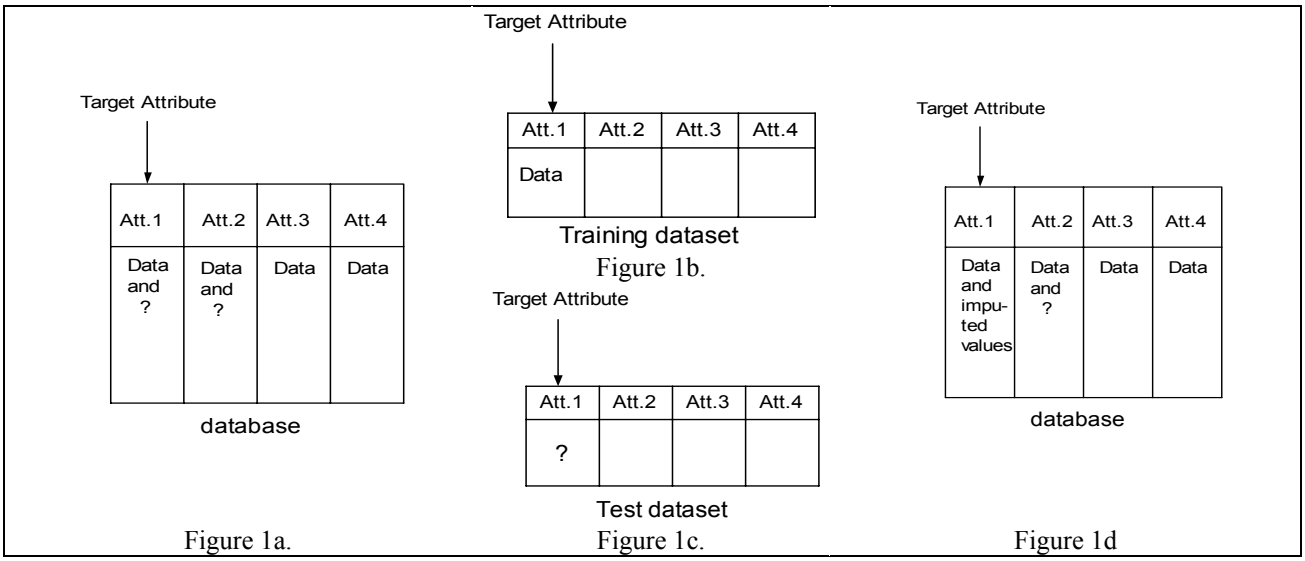

Figure 1. Supervised imputation process using a ML algorithm

The paper uses three different ML algorithms to perform the classification task. The description of the algorithms follows.

4.1. CLIP4

CLIP4 is a rule-based algorithm that works in three phases [4] [5]. During the first phase a decision tree is grown and pruned to divide the data into subsets. During the second phase the set covering method is used to generate production rules. Finally, during the third phase goodness of each of the generated rules is evaluated, and only the best rules are kept while the remaining (weaker) rules are discarded. A specific feature of CLIP4 is use of the integer programming model to perform crucial operations, such as splitting the data into subsets during the first phase, selecting the data subsets that generate the least overlapping and the most general rules, and generating the rules from the data subsets in the second phase. The CLIP4 generates data model that consists of production rules, which use inequalities in all selectors, i.e. IF NUMBER_OF_WHEELS $\neq 4$ AND ENGINE $\neq$ yes THEN CLASS=bicycle. It works only with discrete data.

4.2. Naïve-Bayes

Naïve-Bayes is a classification technique based on computing probabilities [7]. It analyzes relationship between each independent variable and the dependent variable to derive a conditional probability for each relationship. When a new example is analyzed, a prediction is made by combining the effects of the independent variables on the dependent variable, i.e. the outcome that is predicted. Naïve-Bayes requires only one pass through the training set to generate a classification model, which makes it very efficient. The Naïve Bayes generates data model that consists of set of conditional probabilities, and works only with discrete data.

4.3. C4.5

C4.5 is a decision tree algorithm [14] [15]. It uses an entropy based measure [16], which is called gain ratio, as a splitting criterion to generate decision trees. Each tree level is generated by dividing the data at a given node into a number of subsets, which are represented by branches. For each division, gain ratio is used to select the best attribute, which values are used to divide the data into subsets. Each subset contains data that takes on one of the values of the selected attribute. C4.5 generates data model that consists of a decision tree, which can be translated into a set of production rules that use equalities in all selectors. It can work with both discrete and continuous data.

5. Mechanisms leading to introduction of missing values

In general, three different mechanisms, which lead to introduction of missing values can be distinguished [9]:

- Missing completely at random (MCAR), when the distribution of an example having a missing value for an attribute does not depend on either the observed data or the missing data. For example, a student's final grade is missing, and this does not depend on his midterm grade or his final grade.
- Missing at random (MAR), when the distribution of an example having a missing value for an attribute depends on the data, but does not depend on the missing data. For example, student's final mark is missing, and this does depend on midterm grade, but not on the final grade.
- Not missing at random (NMAR), when the distribution of an example having a missing value for an attribute depends on the missing values. For example, student's final grade is missing, and this does depend on the final grade in terms that grades in a special range are always missing.

This paper consider only MCAR model and uses it to introduce missing values when performing experiments.

EXPERIMENTS AND RESULTS

The experiments were performed using seven different datasets, and the five missing data imputation algorithms. The selected seven datasets originally do not contain missing values. The missing data were introduced artificially, using the MCAR model, into each of the datasets. As a result missing values were introduced into all attributes, including class attribute. The missing data was artificially generated to enable verification of the quality of imputation, which was preformed by comparing the imputed values with the original values.

In what follows, first the seven selected datasets are introduced and described. Each dataset is described by a set of characteristics. The selected datasets cover entire spectrum of values for each of the characteristics. Next, the imputation experiments are described and explained. Finally, the results of experiments are investigated to analyze possible links between the characteristics of input datasets and quality of imputation for specific algorithms.

6. Experimental setup

The experiments use seven datasets selected from the UCI ML repository [1]. The selected datasets include only discrete attributes, since both Naïve Bayes and CLIP4 ML algorithms, which are used to perform supervised imputation, cannot work with continuous attributes. The description of the selected datasets, ordered by the number of examples, is shown in Table 1. It includes all characteristics described in the first section.

Table 1. Description of the datasets used in the experimentation

name	# examples	# attributes	# classes	% of Boolean attributes	% examples in the majority class
Lenses	24	4	3	60	62
Hayes-roth	132	5	3	0	39
Tic-tac-toe	958	9	2	11.11	65
Car	1728	6	4	0	70
Kr-vs-kp	3196	36	2	97.3	52
LED	6000	7	10	87.5	10.75
Nursery	12960	8	5	11.11	33

As it is shown in Table 1, the selected seven datasets cover the entire spectrum of values for each of the characteristics:

- the size of datasets, expressed in terms of the number of examples ranges between 24 and almost 13K
- the number of attributes ranges between 4 and 36
- the number of classes ranges between 2 and 10
- the ratio of Boolean attributes ranges between 0 and 97%

In general, the datasets were selected to assure comprehensiveness of the results. The experiments introduce missing values in four different quantities, i.e. 5%, 10%, 20% and 50% of data was randomly turned into missing values. This assures that entire spectrum, in terms of amount of missing values, is covered.

The quality of imputation was evaluated by comparison of imputed values with the original values. The experiments report accuracy of the imputation. For the supervised imputation methods, the sensitivity and specificity of the imputation are also computed. These values are computed for each of the attributes in the data, and the average value is reported.

7. Experimental results

7.1. Design of database characteristics

Based on the experimental results, several changes were made in respect to the choice and design of the database characteristics initially considered and described in section 1. We note that these characteristics were designed for general data analysis purposes, not just for the missing data imputation task. While analysis of results in respect to some characteristics, such as number of attributes and number of examples, generated some interesting knowledge, the analysis for the remaining characteristics, i.e. number of classes and proportion of Boolean attributes, did not generate useful knowledge showing that their definitions need to be redesigned.

In general, ML algorithms depend not only on the number of classes, but more properly on the number of examples for each class. Therefore, in this study, "number of examples/ number of classes" characteristic is used instead of the "number of classes" characteristic. Similar reasoning applies to the "proportion of Boolean attributes" characteristic. Using a simple proportion does not accommodate for the characteristics of the remaining, non Boolean, portion of the data, which is important from the ML point of view. We note that ML algorithms can be sensitive to granularity of attributes expressed in terms of number of their distinct values combined with the number of classes defined in the data. For example, attributes with number of distinct values lower than number of classes cannot be successfully used to distinguish between all classes. This lead to defining a new characteristic "number of Boolean values / (number of values*number of classes)", which was used instead of the "proportion of Boolean attributes" characteristic. Also, a new "amount of missing values" characteristic was added. Therefore, the following new characteristics are used to describe the input databases in order to come up with guidelines to select the most suitable missing data imputation methods:

- Amount of missing values,
- Number of examples,
- Number of attributes,
- Number of Boolean values/(number of values*number of classes),
- Number of examples/number of classes.

Next section provides and analyzes comparison of imputation methods based on the new characteristics.

7.2. Comparison of the data imputation methods

The results section summarizes experiments that apply five missing values imputation methods on seven datasets, for which four different amounts of missing information were introduced. The results report accuracy of the imputation, and are analyzed from the perspective of each of the input data characteristics.

First, Figure 2 summarizes imputation accuracy of each method against the four amounts of missing values. The accuracies for each amount of the introduced missing values are averaged over the seven datasets. Figure 2 shows that the supervised imputation method based on the C4.5 ML algorithm has on average the best imputation accuracy throughout the entire considered spectrum of amounts of missing values. The supervised imputation method based on the Naïve Bayes ML algorithm has the mean imputation accuracy, which is very close to the accuracy of the imputation based on the C4.5 algorithm. The Mean imputation method has, on average, the mean imputation accuracy that places it on the third position, while the remaining methods are significantly worse. In general, we observe that the supervised imputation algorithms have better performance comparing to the unsupervised algorithms. Among the supervised algorithms, method based on the C4.5 ML algorithm, which is a decision tree algorithm, has the best mean imputation accuracy across the different amounts of missing values. Figure 2 shows that, in general, the imputation accuracy of all imputation methods declines with the increasing amount of missing information, which is a result of poorer quality of the underlying data.

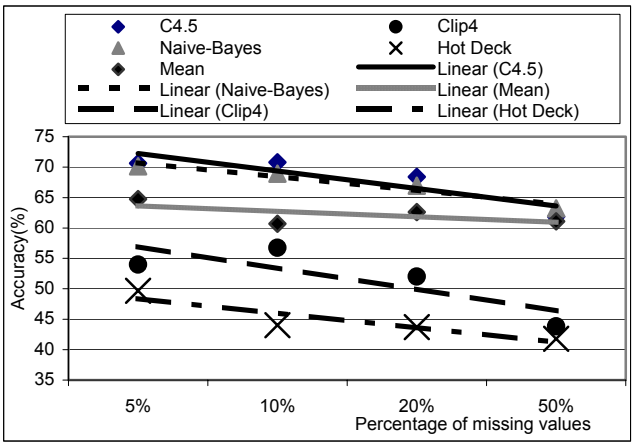

Figure 2. Accuracy against amount of missing values

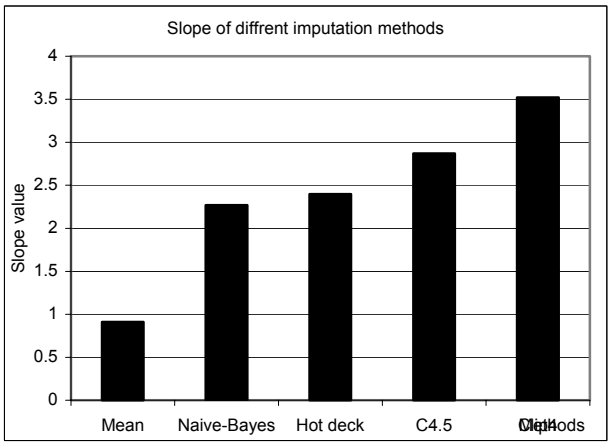

Figure 3. Slope of the accuracy trend for different amounts of missing values

Figure 3 shows the slope of the linear trend between the mean imputation accuracy and the different amounts of missing information. The slope shows the pace of performance degradation of each of the missing data imputation methods with the increasing amount of missing data. The higher the value of the slope the faster the quality of the method degrades. It can be observed that the Mean imputation method is the most stable imputation method. Although it has lower mean imputation accuracy for the considered amounts of missing information than the supervised methods based on the C4.5 and Naïve Bayes methods, its stability suggests that for higher amounts of missing values it may overrun the supervised imputation methods. We note that in general datasets contain small amount of missing information, but for some domains it is possible to have more then 50% of missing values. For example, a medical data describing patients with cystic fibrosis that contains over 65% of missing information was successfully used to find useful relationships about the disease [8].

To summarize, Figure 3 shows that the unsupervised imputation methods are more stable comparing to the supervised methods. The main reason is that the supervised methods must have a training dataset of proper quality to develop an accurate model that is used to impute the missing information. On the other hand, the unsupervised imputation methods are less sensitive to the amount of missing values.

Figures 4, 5, 6, and 7 compare different missing data imputation methods based on the normalized rank values. The normalized rank enables side by side comparison of the imputation methods, which is independent of the quality of the considered datasets. In order to compute the normalized rank value, the imputation accuracy of all methods is scaled to a common [0, 1] interval, with the lowest accuracy set to 0, and highest accuracy set to 1. The remaining imputation accuracy values are computed proportionally within the interval. For example, if the lowest accuracy for a given method would be 60% and the highest 90%, then 90% becomes 1, 60% becomes 0, and the scaled value for 80% accuracy would be 0.667.

Figure 4 shows the normalized rank values for the average imputation accuracy, across different amounts of missing values, for all imputation methods against the increasing number of examples in the datasets. The rank for both CLIP4 and Naïve Bayes based supervised imputation methods improves with the increasing size of the dataset. We note that in general the amount of input data is an important factor for ML algorithms. Having more data may help the ML algorithms to generate a better model, which consequently improves the quality of imputation. We also note that the supervised imputation method based on the C4.5 ML algorithm almost always performs the best. The quality of the imputation performed with the unsupervised imputation methods does not dependent on the size of the data. There is no clear trend in their performance for the increasing amount of input data.

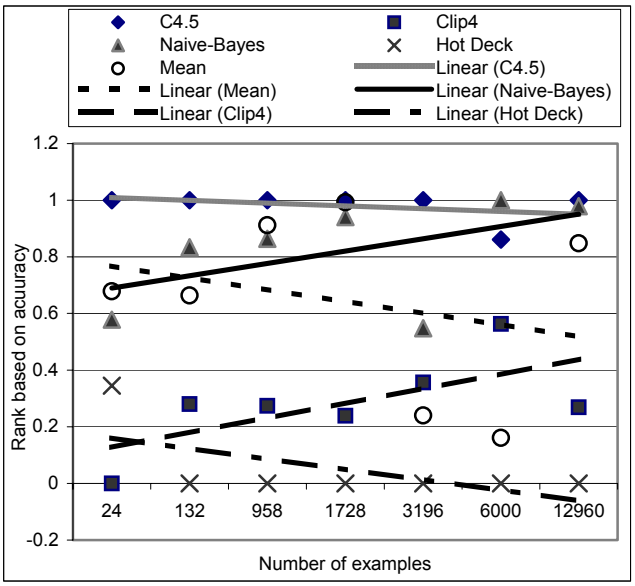

Figure 4. Normalized rank of the average imputation accuracy versus the number of examples

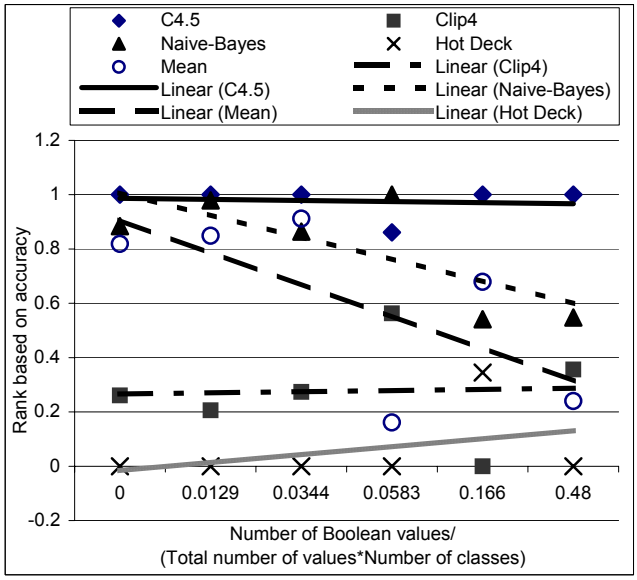

Figure 5. Normalized rank of the average imputation accuracy versus the Boolean characteristic

Figure 5 shows the normalized rank values for the average imputation accuracy, across different amounts of missing values, for all imputation methods against the *Boolean characteristic*, which is defined as "number of Boolean values / (total number of values*number of classes)". We observe that for the increasing values of this characteristic, performance of the supervised imputation method based on the Naïve Bayes algorithm gets worse comparing to the method based on the C4.5 ML algorithm. The same trend can be observed for the Mean imputation method. Other methods are not susceptible to this characteristic, and the imputation method based on the C4.5 ML algorithm has the best performance.

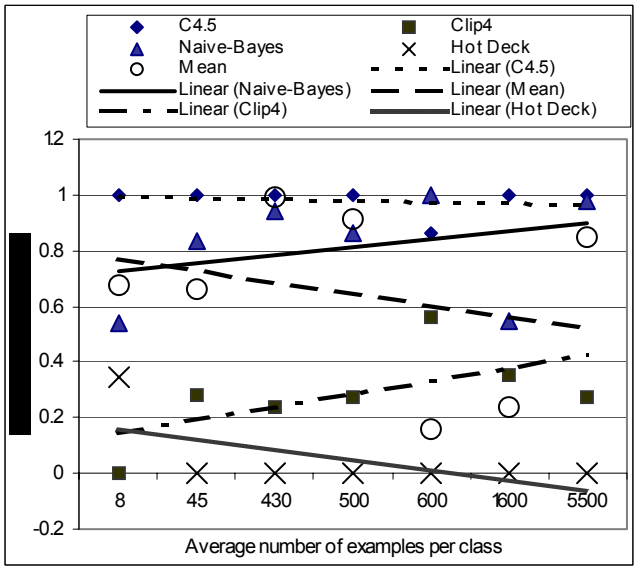

Figure 6. Normalized rank of the average imputation accuracy vs. the average number of examples / class

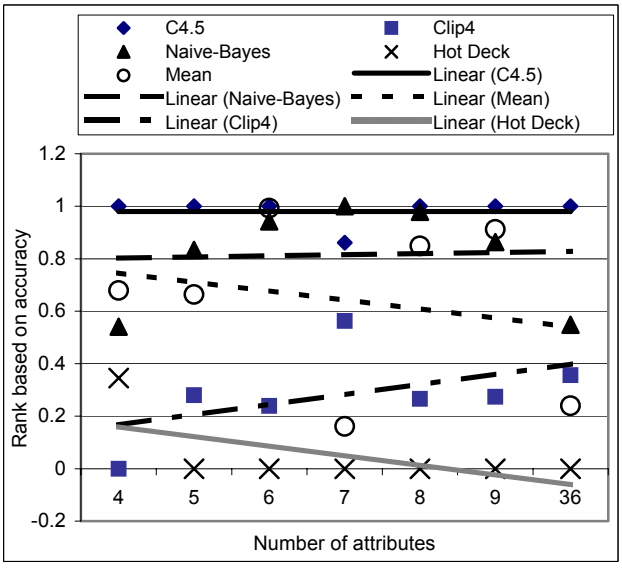

Figure 7. Normalized rank of the average imputation accuracy versus the number of attributes

Figure 6 shows the normalized rank values for the average imputation accuracy, across different amounts of missing values, for all imputation methods against the average number of examples per class, which is defined as "number of examples / number of classes". The Figure shows that the quality of imputation for the supervised imputation methods based on the CLIP4 and Naïve Bayes algorithms improves with the increasing value of the average number of examples per class. This can be attributed to the improved quality of the data model generated by the ML algorithms with the increasing number of examples that are used to generate it. As a result, the quality of the imputation that is performed using the generated data model improves. On the other hand, it can be observed, as expected, that the unsupervised imputation algorithms are not susceptible to this characteristic. We also note that the imputation method based on the C4.5 algorithm has the best average normalized rank.

Figure 7 shows the normalized rank values for the average imputation accuracy, across different amounts of missing values, for all imputation methods against the number of attributes. The rank of supervised imputation methods based on both CLIP4 and Naïve Bayes algorithms improves with the increasing number of attributes. This trend can be attributed to the increasing quality of the data models used to perform imputation, similarly as for the results described in the Figure 6. Again, we note that imputation method based on the C4.5 algorithm has the best average normalized rank.

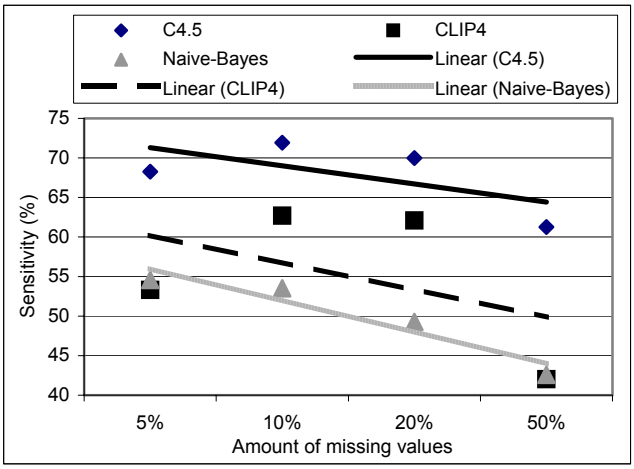

Figure 8. Average sensitivity versus the amount of missing values

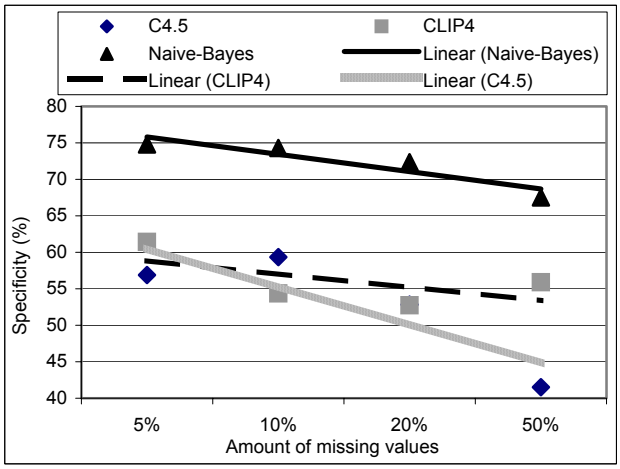

Figure 9. Average specificity versus the amount of missing values

The average sensitivity, over the seven input datasets, of the supervised imputation methods against the different amounts of the missing data is shown in Figure 8. Similarly, Figure 9 shows the average specificity of different imputation methods. In general, increasing the amount of missing values results in decline of both sensitivity and specificity for all supervised imputation methods. In case of the sensitivity, the slope of the trend line for the imputation method based on the Naïve-Bayes and CLIP4 ML algorithms is greater than the method based on the C4.5 algorithm. We also note that the imputation method based on the C4.5 algorithm achieves the best sensitivity values. On the other hand, the specificity of this method is worse than specificity of the method based on the Naïve Bayes algorithm. This shows that the imputation method based on the C4.5 is not universally better than imputation based on the Naïve Bayes algorithm, but rather they complement each other.

8. Discussion

The results shown in Figures 2 through 7 indicate that the supervised imputation method based on the C4.5 ML has the best overall performance. The results also indicate that the imputation method based on the Naïve Bayes ML algorithm is the second best. In general, it can be seen that the supervised imputation methods have better performance than the unsupervised imputation methods.

The analysis of stability of performance of the imputation methods with the increasing amount of missing values shows some interesting relationships. The Mean imputation is the most stable, which means that its performance degradation is the slowest compared to all other methods considered in this study. We expect that for the datasets with high amounts of missing values, unsupervised imputation algorithms may perform better than the supervised one. The rationale behind it is that supervised methods build data model which is used to perform imputation and which quality is dependent on the quality of underlying data, while unsupervised methods are more robust in terms of the quality of the underlying data.

Another important trend shows that increasing the number of attributes and number of examples results in increasing the quality of imputation for the supervised imputation methods. Comparison between the sensitivity and specificity of different supervised imputation methods shows that although the C4.5 based method has better sensitivity, the Naïve Bayes based method is superior in terms of specificity. This shows that these methods complement each other.

The results also show that the performance of the unsupervised imputation methods does not depend on the number of attributes, which conforms to the procedures they use.

We also note that although the execution time of the imputation algorithms was not measured, in general the unsupervised Mean imputation method was the fastest and scaled well with the increasing size of the input data. The second fastest was the supervised imputation that uses the C4.5 algorithm.

SUMMARY AND CONCLUSIONS

9. Summary and conclusions

Most of the real world databases have the shortcoming of containing missing values. This paper uses two well-known statistical approaches, i.e. Mean and Hot Deck methods, to impute missing data. In addition, supervised methods for data imputation, which are based on ML algorithms, were also used. These ML algorithms include C4.5, a decision tree based algorithm, CLIP4, a decision rule based algorithm, and Naïve Bayed, which is a probabilistic algorithm.

Experiments presented in this paper compare the performance of the imputation methods on seven databases. The database characteristics cover a wide range of values. The results of the experiments show the superiority of supervised imputation methods. Among the supervised methods, the decision tree based method has the best performance, while the Naïve Bayes based method is the second best.

We also note that the unsupervised methods are more stable with respect to increasing amount of missing information. Their performance decreases slower than the performance of the supervised methods. It can be expected that their performance may be better for databases with large amounts of missing values. The results also indicate that unsupervised imputation methods do not depend on the size of the input data, both in terms of the number of the attributes and the number of examples. On the other hand, the supervised imputation methods improve their performance with the increasing size of the input data.

10. Future work

This paper only considers missing data generated according to the MCAR mechanism. The next step is to consider other mechanisms, which include the MAR and NMAR methods. We also plan to include other imputation methods, such as parameter estimation methods, in order to improve comprehensiveness of the study. Also, increasing the range and improving granularity of the amount of missing data that is introduced into the original database is planed as the subject of the future work. New experiments will include 5%, 10%, 20%, 30%, 40%, 50%, 60%, and 70% of missing data. The expectation is that unsupervised algorithms will perform better in case of data that has high amounts of missing values. Finally, we plan to investigate using other, more complex distance definitions for computing the Hot Deck imputation in order to improve quality of imputation generated by this method.

REFERENCES

1. C.L. Blake, and C.J. Merz, *UCI Repository of Machine Learning Databases*, [http://www.ics.uci.edu/ ~mlearn/MLRepository.html], Irvine, CA: U. of California, Department of Information and Computer Science, 1998

2. P. Brazdil, J. Gama and R. Henery, Characterizing the Applicability of Classification Algorithms Using Meta Level Learning, In: F. Bergadano, and L. de Raedt (Eds.), *Machine Learning – ECML-94*, Springer Verlag, 1994
3. K.J. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers,1998
4. K. J. Cios, L.A. Kurgan, Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms, In: L.C. Jain, and J. Kacprzyk, (Eds.), *New Learning Paradigms in Soft Computing*, pp. 276-322, Physica-Verlag (Springer), 2001
5. K.J. Cios, and L.A. Kurgan, Hybrid Inductive Machine Learning Algorithm that Generates Inequality Rules, *Information Sciences*, Special Issue on Soft Computing Data Mining, in print, 2004
6. A.P. Dempster, N.M. Laird, and D.B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm (with discussion), *Journal of Royal Statistical Society*, vol.82, pp.528-550, 1978
7. R.O. Duda, and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, 1977
8. L.A. Kurgan, K.J. Cios, M. Sontag, and F.J. Accurso, Mining the Cystic Fibrosis Data, In: Zurada, J., and Kantardzic, M. (Eds.), *Novel Applications in Data Mining*, IEEE Press, in print, 2004
9. K. Lakshminarayan, S.A. Harp, and T. Samad, Imputation of Missing Data in Industrial Databases, *Applied Intelligence*, vol 11, pp. 259 – 275, 1999
10. R.J. Little, and D.B. Rubin, *Statistical Analysis with Missing Data,* John Wiley and Sons,1987
11. T.M., Mitchell, *Machine Learning*, McGraw-Hill, 1997
12. D. Michie, D.J. Spiegelhalterand, and C. Taylor, *Machine learning, Neural and statistical classification,* Ellis Horwood, 1994
13. D.B. Rubin, *Multiple Imputations for Nonresponse in Surveys*, John Wiley and Sons: New York, 1987
14. J.R. Quinlan, Induction of Decision Trees, *Machine Learning*, vol.1, pp.81-106, 1986
15. J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufman, 1992
16. C.E. Shannon, A Mathematical Theory of Communication, *Bell Systems Technical Journal*, vol.27, pp.379-423, 1948
17. W. Vach, Missing Values: Statistical Theory and Computational Practice, In: P. Dirschedl, and R. Ostermann, (Eds.), *Computational Statistics*, Physica-Verlag, pp. 345-354, 1994