Using Entropy to Impute Missing Data in a Classification Task

Thomas Delavallade and Thanh Ha Dang

***Abstract*— In real applications, part of the data is usually missing. But most techniques of data analysis and data mining can only deal with complete data. In this paper, a new taxonomy of imputation methods is proposed. Within this taxonomy a new technique, based on entropy measures is introduced. Its behaviour is studied through an empirical comparative analysis.**

I. Introduction

Most data mining algorithms are highly dependent on the quality of input data. Besides, in real applications, available data often contain corrupted, inconsistent or missing values. In order to come to valid conclusions, it is thus fundamental to tackle this issue. We have decided to focus on classification problems. Among the machine learning tools, classifiers are wide-spread and their performances can worsen when data are incomplete. Some can even not handle missing values.

When dealing with missing data, three strategies are possible:

1) Use an algorithm that can intrinsically deal with missing values, or modify it in that purpose. This is for instance what Quinlan did in his C4.5 decision tree algorithm [1].
2) Remove all observations containing missing values in order to create a new complete database[1]. This solution is quite popular because of its simplicity. Yet it leads to big information losses. Besides basic statistic estimations, like the mean and variance, are biased.
3) Impute missing values in order to complete the original database without information loss. Since the first strategy is not always applicable and the second has some unacceptable weaknesses, this last strategy is the one we study in this paper.

Values can be missing for many different reasons: sensor default, human mistake or simply because the value does not exist. Causes of the missing data pattern, namely the missing data mechanism is quite important since a given imputation method may not be well suited for all cases, depending on its properties. Little and Rubin [2], and since then all researchers in this field, distinguish three mechanisms: Missing Completely At Random (MCAR) when the probability of being missing is a constant, Missing At Random (MAR) when this probability depends on the observed but not on the missing data. Otherwise the mechanism is said to be Not Ignorable (NI) or Not Missing At Random (NMAR). Subsequently we will only consider the MCAR mechanism. It is the most straightforward to deal with from a theoretical point of view. Maybe because of this, it serves as a reference for empirical comparisons of imputation methods [3], [4]. Before introducing our model in section III, we give an overview of the most popular imputation techniques in section II, so as to underline the characteristics of our model. In section IV we analyse the empirical behaviours of various techniques including ours. Lastly we point out some research directions.

Thomas Delavallade is with THALES Land & Joint Systems and Université Pierre et Marie Curie - Paris6, CNRS UMR 7606, DAPA, LIP6, 104 avenue du président Kennedy 75016 Paris, France (phone: +33 144278751; email: Thomas.Delavallade@lip6.fr).

Thanh Ha Dang is with the Université Pierre et Marie Curie - Paris6, CNRS UMR 7606, DAPA, LIP6, 104 avenue du président Kennedy 75016 Paris, France (phone: +33 144278726; email: ThanhHa.Dang@lip6.fr).

[1]Symmetrically variables with missing values can be removed. Because of the potential importance of a given variable, this solution is rarely used.

II. Imputation techniques

Missing data imputation has been investigated for decades by statisticians in order to analyse surveys. It is now an important research area in data mining, as a preprocessing step. Describing every method from both fields would be fastidious. Therefore we prefer to present the main differences among the various families of methods, so that a synthetic taxonomy can be sketched. Only the methods tested in the empirical comparison of section IV will be detailed and situated in this taxonomy.

Hu *et al.* [5] introduced a first bicriterion dichotomy among the different imputation techniques. Both criteria can be interpreted as binary questions. The first one is: *is the technique deterministic or stochastic?* while the second is: *is there an underlying prediction model?* Following these ideas we have decided to raise new discriminatory questions, adding new criteria to distinguish the various techniques in a more precise way.

Trying to find a substitution value for a missing one, corresponding to a given observation of a given variable, various options are possible depending on how you answer the following questions.

1) *Do you deal in the observation space or variable space?* You can either use information from other observations of the same variable or from other variables for the same observations.
2) When considering the variable workspace, *which prediction model do you want to use?* Using information from other variables means you believe the correlation structure may help you in predicting the missing value. In this case, different models can be applied: bayesian, regression or classification for instance.
3) *Do you favour an iterative solution?* To avoid a vicious circle when using prediction models, one might think that the prediction techniques available for imputation are those able to handle directly missing values. This is not mandatory. Indeed it is possible to make a first imputation with a basic technique and then refine the

1-4244-1210-2/07/$25.00 ©2007 IEEE.

imputed values thanks to any prediction method. In such situations, it is easy to understand that the process can be repeated several times to get finer and finer imputed values, until a stopping criterion is met.

4) *Do you use local or global information?* In both workspaces you may favour observations or variables close to the missing data, assuming the information conveyed is more relevant.
5) *Do you use class information?* In supervised classification you may like to benefit from the information contained in the class variable. In such a case, the method is said to be *supervised*. In unsupervised settings a similar question would be: *do you use cluster information?* In this case, any clustering method can be used to get this information.
6) *Is your technique stochastic or deterministic?* This is the first discriminatory question of Hu *et al.*

The taxonomy implied by these questions is a useful tool not only to discriminate different techniques based on basic characteristics, but also to define new methods. For instance one popular technique is the k nearest neighbours [3], [4]. It operates in the observation workspace, uses local information but does not take into account the class variable. It is deterministic and not iterative. To impute a value that is missing for a given observation of a given variable it looks for the k closest observations and computes the weighted mean of the values taken by the variable of interest, if it is numeric, for these k observations. For nominal variables, a majority vote is performed. Obvious difficulties lie in the choices of k and the distance metric.

To compute the distance between observations containing missing values for some variables, we can simply ignore these variables and assess the distance on a reduced dimension. But this solution implies that all distances that have to be compared are not computed on the same dimensions. To circumvent this problem, inspired by the third question mentioned above, we have developed an iterative version of the k nearest neighbours. A first basic assignment is carried out, so that we have a complete database from which we compute ordinary distances in order to reestimate the originally missing values. We repeat the process until values do not change much between two iterations. In section IV, this technique is denoted *kppvI*.

In addition to *kppvI*, working in the observation space, we have the *mean*, *median* and *mode* which replace a missing value with the mean, median and mode of the variable of interest. Instead of computing these statistics on the whole variable, we can take only the observations of the same class as the observations of interest. We will denote these methods by *Cmean*, *Cmedian* and *Cmode*. In order to test non deterministic methods we also use the stochastic version of the mean methods: *meanS* and *CmeanS*. The difference with their deterministic counterparts is that a random number is drawn around the mean, according to a normal distribution with a standard deviation estimated from the data. One advantage of these methods is that the estimation of the variance is increased compared to the biased underestimation made by the deterministic versions. All these methods are detailed in [2].

To serve as a baseline, we use a *random* technique which simply substitutes a missing value with a random one, drawn uniformly between the minimum and maximum values of the variable of interest, if it is numeric. For nominal variables, we draw uniformly one of its modalities.

In the variable workspace category, we use classification techniques for nominal variables and both regression and classification for numeric ones. Classifiers can be easily used with a nominal variable, which serves temporarily as the class variable, while the others are used to predict it. We tested C4.5, nearest neighbour and naïve Bayes classifiers, as they are implemented in Weka 3.4.7[6]: J48, IB1 and NB.

For numeric variables, a discretization step is needed before processing classification[2]. Only J48 has been tested, with three discretization methods: *EW* (equal width intervals), *EF* (equal frequency intervals) and *ID3* (discretization method of Quinlan's ID3, based on entropy [1]).

A more obvious way of predicting a missing value of a continuous variable is to consider this task as a regression problem. The variable to be imputed becomes the dependent variable, while the others are the independent ones. In this category we have implemented the Local Least Square (*LLS*) method of Kim *et al.* [7]. First of all missing values are imputed with a basic method like the *mean*, then the k closest neighbours of the variable of interest are selected and used as the independent variables in a linear regression model. This step is repeated until the imputed values do not change much. This technique is thus iterative and local. The difficulties are the same as the ones mentioned for the k nearest neighbours.

The main criticisms that can be made to the prediction techniques concern the hypotheses that are made: dependencies and independencies between the variables, linearity for the regression *etc.* These are common criticisms, but there is something deeper we would like to point out. The underlying goal pursued when making use of these techniques is to find substitution values as close as possible to an hypothetic "true" value. This becomes clearer when looking at the evaluation measures of such techniques that you can find in the literature. Very often it is the error between the predicted and the original value, known thanks to the experimental design [7]. Statisticians may also be interested in methods that preserve data distribution [5], which explains why they prefer, from a theoretical point of view, non deterministic techniques [2]. Remind that we have decided to place ourselves in the context of supervised classification. Our major concern is to achieve the best performances with a given classifier and not the distribution of the data or the closeness to true values that are not known anyway in real applications. The new method we describe in the next section has been specifically developed to match this goal.

[2]The classifiers will predict an interval. To come back to a numeric value we draw a random number in this interval according to a normal distribution, its mean and standard deviation being estimated from the observations belonging to this interval.

III. A New Imputation Technique Based on Entropy

The new technique we want to develop has to fulfil the following requirement: *given a classification algorithm and an incomplete database, find substitution values for the missing ones that enable to get the best classification performances*. Of course these performances have to be assessed on an independent dataset to avoid overfitting.

In classification, the value of a nominal variable has to be predicted thanks to the values of other variables. The link between a predictive variable (or attribute) and the class has thus to be considered, the stronger the link the better the attribute[3]. One way of measuring this link in classification problems, is to estimate how well an attribute can discriminate the various classes. Coming from information theory, entropy measures are well suited to quantify the discrimination power of an attribute. That is why they are used for feature selection, especially in decision tree induction since Quinlan's ID3 algorithm. Our assumption is that missing values deteriorate the discrimination power of attributes. So we intend to process imputation, attribute by attribute[4], on the basis of entropy computations, in order to improve this power.

Let $\xi = \{e_1, ..., e_n\}$ be the incomplete database, with n observations and let A be a symbolic attribute with k modalities $v_1, ..., v_k$. Shannon's entropy of ξ is defined by:

$$I(\xi) = -\sum_{i=1}^{C} P(c_i) \log P(c_i)$$

where C is the number of classes and c_i denotes the i^{th} class. The entropy of ξ conditioned on v_j of A is:

$$I(\xi|A = v_j) = -\sum_{i=1}^{C} P(c_i|v_j) \log P(c_i|v_j)$$

The entropy of ξ conditioned on A is then the weighted mean of the entropies of ξ conditioned on all modalities of A:

$$I(\xi|A) = \sum_{j=1}^{k} P(v_j) I(\xi|A = v_j)$$

Finally the discrimination power of an attribute is defined by means of the information gain:

$$G(\xi, A) = I(\xi) - I(\xi|A)$$

We will note A^m and A^o the sets of missing and observed values of A, and S the set of imputation solutions. As there are k modalities with which we can impute each missing value of A, the cardinal of S is finite: $|S| = k^{|A^m|}$.

The main idea is to replace missing values of A with the values that maximize $G(\xi, A)$ or equivalently with those that minimize $I(\xi|A)$ because $I(\xi)$ is independent of A:

$$s_{optimum} = \arg\min_{s \in S} I(\xi|s(A))$$

where $s(A)$ represents the attribute A on which the imputation s has been performed. It can be demonstrated that, in the optimal solution, all examples from the same class are imputed with the same value. This property means that the complexity of our algorithm can decrease down to $O(k^C)$ when $C < |A^m|$. We now investigate the suitable algorithms for the implementation of this principle.

Our first algorithm is exhaustive. It consists in evaluating all possible imputations and choose the one which minimizes the previous equation. This exactly fits our requirements. Yet the complexity is exponential in the number of cases or in the number of classes. So when these numbers are high, this algorithm is rather costly.

To overcome these difficulties we propose two approximate solutions. The simpler one proceeds in two steps: all missing values v of A are treated separately:

1) For each modality v_j, compute the entropy of ξ conditioned on A^o to which the potential substitution value v_j for v is added:
$$I(\xi|A^o \ \& \ v = v_j)$$
2) Choose the smallest conditional entropy and set v to the corresponding value.

Another solution, more costly yet closer to what we want, can be achieved through an iterative process. A first imputation is performed, like the one just mentioned for the non-iterative process. For the following iterations, each substitution value is reestimated in the same way. The only difference being that all previously imputed values are considered to be truly observed. The iterations stop when the conditional entropy do not decrease significantly anymore. Examples of such computations can be found in [8]. The optimum found by both algorithms is only guaranteed to be local, but not global. They have yet the advantage of being less computationally demanding than the original one, the number of entropy computations being of the order of $\Theta(\min\{|A^m|, C\} \times k \times L)$ where L is the number of iterations. In our experiments of section IV, both algorithms were used, but we present only the results of the iterative one, which are slightly better.

It can be shown that our method tends to favour, for a missing value of an example of class c_i, the modality v_j which maximizes $P(c_i|v_j)$ in the completed database. The demonstration consists in computing the conditional entropy of ξ without considering the missing values, then the conditional entropy considering there are d examples with missing values in A, from the class c_i, all replaced by the same v_j. Under the hypothesis that the number of observations with the modality v_j for attribute A and from class c_i is large enough, results can be simplified and the proposition can be proved. This property may be used to realize the initial imputation in the iterative algorithm. Though the algorithms described herebefore are devoted to symbolic variables, it is straightforward to apply them to continuous variables. The process is the same as the one described in section II for classification applied on continuous variables. Features are first discretized with one of the three mentioned techniques.

[3]This remark is valid as long as attributes are considered independently.

[4]This implies our technique works in the observation space. It takes into account all observations. So it acts globally.

Then the algorithm based on entropy that we have just detailed is used to find a discrete imputed value. Finally this value which represents an interval of the original feature space, is replaced with a numeric value uniformly drawn around the mean of the interval.

IV. Empirical Analysis

Now that we have presented the strengths and weaknesses of several imputation techniques, it is important to analyse empirically their behaviours on real datasets. This will enable us to characterize them more finely and to compare our new technique with existing ones.

A. Protocol

Contrary to classic imputation techniques, the one we developed, based on entropy, focuses on the improvement of performances of a classifier. With this goal in mind, evaluating imputation techniques consists in evaluating the performances of a classifier learnt on an incomplete database filled with substitution values according to those techniques. Following the first comparative works in the field of imputation for classification [3], [9], we have decided to use the global accuracy to measure the goodness of a classifier, as a proxy for imputation goodness, though it is a controversial measure. Indeed it does not take into account the fact that class distributions may be unbalanced, and that class errors may not have the same costs. However this was the only way of comparing our findings with previous studies.

In order to control every empirical parameter we only deal with complete databases from which values are deleted according to the MCAR generation mechanism described in section I. To avoid biases induced by the experimental process, we have decided to set up a strict protocol. It is also a way of enabling reproducible experiments.

Each database is first split into ten pairs of independent training and test sets through a stratified cross-validation process. Then some data are removed from the ten training sets, according to the MCAR mechanism, so that a given missing ratio is reached. We have used five different missing ratios in order to study the behaviours of the different techniques with more or less missing data: 10%, 20%, 30%, 40% and 50%. This means that each data from the complete database has a corresponding chance of being missing. All attributes and all observations contain missing data. So for each database we build 50 training sets and 10 test sets (none of them contain missing values). For each technique to be tested, missing values are imputed in the training sets, and classifiers are learnt on each of these sets and evaluated on the corresponding test sets. We have tried three different classifiers: C4.5, nearest neighbour and naïve Bayes[5]. Then for each missing data ratio, the performances are computed as the mean accuracies on the ten pairs of datasets.

It is very close to what Batista and Monard did [3] (same protocol, but only the most relevant attributes contain missing data), but different from Zou *et al.* protocol [10] in which there is no cross-validation (problem with the robustness of the induced classifiers). We also depart from the protocol of Acuña and Rodriguez [4], and Grzymala-Busse and Hu [9] in which missing values are inserted in the whole database and imputation is processed on this incomplete database before performances are assessed through cross-validation (problem with supervised imputation techniques which use the class variable to find the substitution values, while these methods should not be used on test sets for which the class variable must be hidden to avoid overfitting).

In order to compare the various techniques tested, we need to rely on some methodology to ensure, given a small enough probability of error, that observed differences are not artifacts caused by the sampling procedure. If the Student paired t test is probably the most popular way of comparing two machine learning algorithms on a given dataset [11], Dietterich showed that it has a rather high type I error rate [12]. Moreover it cannot be applied to compare more than two algorithms which is what we want to do. In such cases, the corresponding parametric test is the repeated measures analysis of variance often called ANOVA. Yet its underlying assumptions, just like with the t test, are seldom verified: the scores to be compared must come from normal distributions, with the same variance (homoschedasticity). Therefore, inspired by Bradzil and Soares [13] and Demsar [14], we have chosen to use the Friedman test, a non parametric equivalent of the ANOVA[6].

Suppose we have k algorithms, tested on n databases. For each algorithm we can compute its mean rank μ_i. The null hypothesis states that $\mu_1 = \mu_2 = ... = \mu_k$. The test enables us to accept or reject this hypothesis for a given confidence level. When the null hypothesis is rejected we only know that there exists i, j such that $\mu_i \neq \mu_j$ but we neither know how many classifiers performances differ, nor which ones differ. So we need to perform post-hoc tests. As Demsar suggests, we use the Nemenyi test when we wish to compare all pairs of algorithms. It is the non parametric equivalent of the Tukey test. When one algorithm serves as a control against which all others are tested, we rely on the the modified Bonferroni-Dunn test procedure called Holland-Copenhaver step-up, hereafter denoted HC-SU. It has a higher power than the procedures depicted by Demsar [15].

B. Experimental Results on Symbolic Datasets

Our method has first been designed to handle symbolic data. So let us first have a look at its behaviour on symbolic data. We have tested with five distinct databases, all taken from the UCI repository[7], described in Table I. For each database, 5 replicates have been built with 5 different missing rates and each time 3 classifiers have been used to estimate the performances of 7 imputation techniques. So it would require 15 tables to present thoroughly our results. With numeric data this number raises up to 24 since there are

[5]We use their Weka implementation: J48, IB1 and NB.

[6]Friedman test consists in performing the ANOVA on the ranks of the algorithms, rather than directly on their performances.

[7]http://www.ics.uci.edu/~mlearn/MLRepository.html

TABLE I

SYMBOLIC DATABASES DESCRIPTION

Database	Features	Instances	Classes
Car Evaluation	6	1728	4
Congressional Voting Records	16	435	2
Tic Tac Toe	9	958	2
Zoo	16	100[1]	7
Promoter Gene Sequence	57	106	2

[1] There are in fact 101 instances, but we removed one of the two *frog* duplicates.

8 datasets. For this reason we will give only our most representative performances tables. The interested reader can get the full results on our webpage[8].

To get a global synthesis of the results, we present the outcomes of the statistical tests for each classifier in Table II. Note that since we introduced a new technique we wish to compare to existing imputation techniques, we have used the HC-SU post-hoc tests to assess the statistical significance of the observed differences when the family-wise Friedman test concludes there exists significant differences.

TABLE II

STATISTICAL COMPARISON BETWEEN SYMBOLIC IMPUTATION METHODS[1]

Imputation \ Classifiers	J48	IB1	NB[2]
entropy	2.94	2.84	4.3
mode	**4.82***	3.4	4.52
Cmode	2.26	2.36	3.64
classifier *J48*	4.54*	**4.8***	3.7
classifier *IB1*	4.36	4.34*	3.98
classifier *NB*	4.28	**4.72***	4.56
random	**4.8***	**5.54***	3.3

[1] This table contains the average ranks of each technique, taken over the 25 datasets (5 replicates per dataset).
[2] The Friedman test does not reject the null hypothesis. So post-hoc tests have not been performed.

Table II gathers for each classifier the average rank of each imputation technique. The lower the rank the better the technique. Bold figures indicate that the corresponding technique is statistically different from our *entropy* method at 95% confidence level. When the figure is followed by a star, the conclusion is the same but with less confidence: 90%.

We can notice that our method behaves fairly well. It is never statistically inferior to another one and it is superior to almost all of them at the 90% confidence level, at least for some classifier (IB1 or J48). Yet it is inferior to *Cmode* with all classifiers, although the observed difference is not significant. Among the three classifiers, the best performances are observed with IB1 and J48, while they are rather bad with NB. This is not surprising for J48 since it uses the same discrimination function as the one implemented in our algorithm. Regarding the bad results with NB, it has to be noted that all methods are judged equivalent by the Friedman test. Furthermore the best average ranking is quite high, which let us think that none of the tested method works always well with this classifier. Another surprising result concerns the good performances of a simple method like *random* imputation. It would be interesting to investigate this more deeply. From these remarks, it clearly appears that the goodness of an imputation method in our context, depends on the classifier to be used.

[8] http://www-poleia.lip6.fr/~dang/mdi
In this webpage we also provide all the incomplete datasets we have built to realize our experiments, so that anyone can reproduce them.

C. Experimental Results on Numeric Datasets

Numeric data are very common in real applications. To analyse more deeply our entropy-based technique it is important to figure out if it is still competitive when dealing with such data. In this perspective we have made comparisons on 8 datasets from the UCI repository. See Table III for a brief description of these databases.

TABLE III

NUMERIC DATABASES DESCRIPTION

Database	Features	Instances	Classes
Iris	4	150	3
Wine	13	178	3
Ionosphere	32[1]	351	2
Bupa	6	345	2
Pima Indians Diabetes	8	768	2
Breast Cancer	9	683[2]	2
Glass	9	214	2
Yeast	8	1484	10

[1] There are in fact 34 features, but we removed the first two features, following the suggestion in [4].
[2] There are in fact 699 instances but we removed the 16 observations containing missing values.

Just like with symbolic data, we give our synthetic results in Table IV under the form of average ranks for each imputation technique and each classifier. This time the ranks are averaged over 40 datasets (5 replicates for each of the 8 datasets). Significant differences are expressed in the same way as in Table II. HC-SU tests were applied, considering ID3 - *entropy* as the control, because it seemed to be the best, with regard to average ranks. In this table entropy-based and classification-based methods are prefixed with the name of one of the three discretization techniques mentioned in section II. Much more techniques have been tested, but only the bests of each family are presented. This is why for instance we have only the ID3 discretization with the J48 classifier.

Results from Table IV confirm the observations made on symbolic data. Our method cannot be said inferior to any of the methods tested, for both confidence levels. Furthermore the difference with all other methods, except *CmeanS*, is statistically significant at least once at the 90% confidence level. *CmeanS* can be seen as a stochastic equivalent, for numeric data, of *Cmode*. So both sets of experiments highlight the good potential of the two supervised techniques, including

TABLE IV

STATISTICAL COMPARISON BETWEEN NUMERIC IMPUTATION METHODS[1]

Imputation \ Classifiers	J48	IB1	NB
EW - *entropy*	5.8	5.625	**5.6375***
EF - *entropy*	4.6875	4.55	5.0875*
ID3 - *entropy*	4.5125	4.225	3.625
mean	5.3875	4.75	**7.025***
CmeanS	5.25	3.025	3.725
ID3 - classifier *J48*	5.6125	**6.7***	**5.5625***
5ppv	5.2375	6*	5.2875*
1ppvI	5.5375	**6.8125***	4.975*
LLS	5.3	4.7125	**6.6875***
random	**7.675***	**8.6***	**7.3875***

[1] This table contains the average rank of each technique, taken over the 40 datasets (5 replicates per dataset).

ours. Note that although not significant, the average rank of *entropy* is slightly better than the one of *CmeanS* with two classifiers (J48 and NB), while it was always *Cmode* which seemed to perform better on symbolic data.

Regarding the discretization preprocessing, it appears that ID3 method has always a better average rank than the other two. This is not so surprising since its goal is similar to ours: using also an entropy measure, it intends to build a partition that increases the discrimination power of an attribute.

About the other imputation methods, we can say that ID3 - classifier *J48* must be avoided. It is clearly dominated by the two top-ranking techniques. This remark is of course also valid for *random*. The contrary would have been embarrassing. We have used this method, only to get a reference which has to be outperformed.

Comparing our findings with existing comparative studies in the literature, we can first say that only Batista and Monard [3] use a protocol equivalent to ours. They find that k nearest neighbours performs best. Nonetheless they did not test supervised techniques like *Cmean* or *CmeanS* which appear to be the best ones in our study. Their *kppv* version is also a bit different from ours since they build prototypes to reduce the dimension of the input space, and compute the distances between incomplete instances and the prototypes. In addition they insert missing values only on relevant attributes while our task is harder: there are missing values in all of them.

V. CONCLUSIONS

In this paper we have proposed a new imputation technique. Based on entropy it intends to find, for each incomplete attribute, the substitution values which lead to the highest discrimination power. Statistical tests have been undertaken to analyse objectively our empirical results. Our findings are more than promising. Our method has indeed turned out to be the top-ranked one on numeric data, even though the difference with the second one *CmeanS* is not statistically significant.

As the complexity of our method is greater, one may wonder if it is worth the extra cost. However the fact that it is top-ranked is a good incentive to process a deeper analysis of empirical results, in order to identify the types of problems on which it is worth using the *entropy* technique: on what kind of databases and in association with which classifier?

The two best techniques are supervised: they use the class information. Yet this is the case only with the MCAR mechanism. It would be interesting to test with the two other missing mecanisms. When imputing instances from the test set, the class information is not known. So supervised techniques cannot be used. In existing empirical protocols this is never the case, but in real application it happens very often that an instance we need to classify is incomplete. Therefore it would be important to test a new protocol with missing values in both training and test sets.

ACKNOWLEDGMENT

We would like to thank Bernadette Bouchon-Meunier, Christophe Marsala and Philippe Capet for their constant support and valuable methodological advices. We are also grateful to the reviewers for their useful comments.

REFERENCES

[1] J. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[2] R. Little and D. Rubin, *Statistical Analysis with Missing Data, 2nd edtion*. John Wiley and Sons, 2002.

[3] G. Batista and M. Monard, "An analysis of four missing data treatment methods fo supervised learning," *Applied Artificial Intelligence*, vol. 6, no. 3, pp. 309–327, 2003.

[4] E. Acuña and C. Rodriguez, "The treatment of missing values and its effect in the classifier accuracy," in *Classification, Clustering and Data Mining Applications*, D. Banks, L. House, F. McMorris, P. Arabie, and W. Gaul, Eds. Springer-Verlag, 2004, pp. 639–648.

[5] M. Hu, S. Salvucci, and M. Cohen, "Evaluation of some popular imputation algorithms," in *Proc. of the Section on Survey Research Methods*. American Statistical Association, 2000, pp. 309–313.

[6] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, 2nd edition*. San Francisco, USA: Morgan Kaufmann Publishers, Inc, 2005.

[7] H. Kim, G. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least square," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[8] T. Dang and T. Delavallade, "Utilisation de l'entropie pour substituer des valeurs manquantes symboliques dans un problème de classification supervisée," in *Proc. of the 4th journées internationales sur les Systèmes Intelligents: Théorie et Applications*, 2006, pp. 45–54.

[9] J. Grzymala-Busse and M. Hu, "A comparison of several approaches to missing attribute values in data mining," in *Proc. of RSCTC'00*. Springer-Verlag, 2001, pp. 378–385.

[10] Y. Zou, A. An, and H. Xiangji, "Evaluation and automatic selection of methods for handling missing data," in *Proc. of the IEEE International Conference on Granular Computing*, 2005, pp. 728–733.

[11] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[12] T. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, no. 10, pp. 1895–1924, 1998.

[13] P. Brazdil and C. Soares, "A comparison of ranking methods for classification algorithm selection," in *ECML*, ser. LNCS, vol. 1810. Springer-Verlag Berlin/Heidelberg, 2000, pp. 63–74.

[14] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, no. 7, pp. 1–30, 2006.

[15] Supattathum, Suchada, *et al.*, "Statistical power of modified Bonferroni methods," Paper presented at the Annual Meeting of the American Educational Research Association, April 1994.