EPiC Series in Computing

Volume 41, 2016, Pages 161–173

GCAI 2016. 2nd Global Conference on Artificial Intelligence

Empirical Investigation of Learning-Based Imputation Policies

Hara Skouteli[1] and Loizos Michael[2]

[1] Open University of Cyprus, harikleia.skouteli@st.ouc.ac.cy
[2] Open University of Cyprus, loizos@ouc.ac.cy

Abstract

Certain approaches for missing-data imputation propose the use of learning techniques to identify regularities and relations between attributes, which are subsequently used to impute some of the missing data. Prior theoretical results suggest that the soundness and completeness of such learning-based techniques can be improved by applying rules anew on the imputed data, as long as one is careful in choosing which rules to apply at each stage. This work presents an empirical investigation of three natural learning-based imputation policies: training rules once and applying them repeatedly; training new rules at each iteration; continuing the training of previous rules at each iteration. We examine how the three policies fare across different settings. In line with the predictions of the theory, we find that an iterative learn-predict approach is preferable.

1 Introduction

Missing values imputation is an actual yet challenging issue and there are a variety of methods to handle it, including statistical analysis and machine learning techniques. The selection of the most appropriate solution is determined by factors such as i) possible relations between attributes ii) attribute type: continuous, categorical or mixed data iii) the categories of missing data types. In all cases the solution of the problem is based on two main axes, the first is discovering rules or correlations between attributes by using available data and the second axis is data recovery by using the discovered rules.

However, studying literature one can easily detect that most of the cases make use of the learn-predict iteration model. More precisely, while there are solutions on i) how to modify existing learning algorithms with aim to cope the difficulty of learning from incomplete data ii) various data recovery techniques and policies of how to use them; there has not yet given place,in our opinion, to the investigation on how these two axes should interact at each iteration of the process in order to deliver more sound and complete dataset.

Additionally the reasons that led us to study the various imputation approaches are the following, firstly once the solution involves learning automatically inherits all the problems that concern learning research area (discovery of incomplete and/or incorrect learning rules) and secondly if we assume that our original data were only incomplete, after the first imputation step they will be noisy too. Thus the question that rise is: Is there a natural way to chain the process of learning and recovery in order to maximize soundness and completeness?

This study aims to give an answer to the above question through an empirical investigation of three natural learning-based imputation policies: training rules once and applying them repeatedly; training new rules at each iteration; continuing the training of previous rules at each iteration using recovered data. More over we examine how those policies respond with different type of learning algorithms, percentages of missing attributes and dataset sizes.

2 Imputing Values Through Learning

Learning from incomplete or missing data has been studied under different perception of missing data types [16]:, Missing Completely At Random *(MCAR)*, Missing At Random *(MAR)* and Missing Not At Random *(MNAR)*. In case of reasoning with uncertain knowledge, widely used approaches can be found in the statistics literature [4]. The idea is to impute data based on their expected values given the observed data. In case of MCAR and MAR, Expectation Maximization (EM), a two step approach, is commonly used [2].

In more sophisticated Multiple Imputation (MI) Frameworks, multiple imputation procedure replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute. These multiply imputed data sets are then analyzed by using standard procedures for complete data and by combining the results of these analyses [17].

Statistical approaches are used to discover value estimations when the data are assumed to be independent, rather than a structure of learning sets. In some cases of high level of incompleteness such solutions can lead to a bias system. The objective of this study is to answer the question: "Is there a natural bases imputation policy that could take into account all data and the relation between attributes as well in order to lead us to a more complete and less sound dataset?"

Two machine learning based approaches are demonstrated [7] as alternatives to standard statistical data completion methods: i) Autoclass method, to automatically discover clusters in data (based on Bayesian classification theory) and ii) the C4.5 method, a supervised learning algorithm for decision tree induction. Besides Modern Knowledge Systems use probabilistic Bayesian networks augmented by decision theory to allow making decision about appropriate actions. In a Bayesian framework the Data Augmentation (DA) algorithm is the natural analogue of the EM algorithm. Another approach of MI has also been studied under noise-free deterministic rule learning setting in the Probably Approximately Correct (PAC) learning framework. It has been shown [21] that the PAC learning semantics can be extended to deal with arbitrarily missing information, and that certain PAC learning algorithms can be easily modified to cope with such experiences [9]. Other studies[23] present the benefits of simultaneously learning multiple predictors (rules) from a common dataset and [22] the benefits of rule chaining. To this end a study of Michael [10] presents a theoretical framework for the simultaneous learning and predict approaches as well as the possible ways and benefits of chaining the predictions of each approach.

3 Preliminaries and Notation

Given a set A of attributes x_i, with $i \in \{1, ..., n\}$, define $dom[i]$ the domain of attribute x_i, and by $dom[A]$ the cross product $dom[1] \times ... \times dom[n]$ of all attribute domains. A complete record R is defined by $R \in dom[A]$, and the value of x_i in R is denoted as $R[i]$. In case of missing value of x_i define R^* the partially observation of R, and with $R^*[i]$ the value of the attribute x_i in R^*. The imputed version of R^* is denoted as R^+. The number of missing attributes in

the R^* is denoted as n^* and the number of records in D is denoted as r. The complete dataset is denoted as D, and the dataset with missing values as D^*.

The number of impute iterations of learning-predict process is denoted with *epochs*, and the imputed version of a dataset D^* after j iterations where $0 \leq j \leq epochs$ is denoted as D_j^+.

The set of predictors of the jth epoch for all attributes is denoted by P_j where $0 \leq j \leq epochs$, and the set of predictors for each attribute i of jth epoch is denoted by P_j^i where $0 \leq i \leq n$.

We denote with PoL_j the policy layer which defines the order used in epoch j to apply the predictors of P_j, and a Policy as $\mathbb{P}$, where $\mathbb{P} =< PoL_1, PoL_2, ..., PoL_{epochs} >$.

We denote depth d the number of layers within a policy $\mathbb{P}$. If $d = 1$ the $\mathbb{P}$ is flat, if $d = 0$ the $\mathbb{P}$ is empty and chained if $d > 1$. If imputation solution requires the creation of a new set of predictors in each *epoch*, then $d = epochs$. Furthermore, based on the imputation approach, PoL_{epochs} may shares the same predictors with $PoL_{epochs-1}$ in a different order. Alternatively, we denote D_{epochs}^+ the final imputed version of D^* after applying PoL_{epochs} over $D_{epochs-1}^+$. Note that the given R^* may not have the required completeness to attain the missing value of $x_i \in R$, in such a case $R[i] =^*$.

To this end we consider a dataset as more complete than another when it has less missing values and as more sound than another, when it has less incorrect values. Thus, for better evaluation of each approach we calculate the correctly recovered, incorrectly recovered and unable to recover values.

4 Why Learning and Recovery Processes Should not Be Decoupled

Machine learning approaches base their success to the ability of the learning algorithms to cope with the incomplete data. Also they aim to discover knowledge in the form of rules or functions which can be used accordingly in order to predict values of missing attributes. Those rules represent the discovered correlation between the attributes of the dataset. Hence, in order to predict a missing value, all the attributes involved in the rule must have a known value.

In cases that not all necessary values are available and the process is unable to predict the missing value, there are three approaches to handle the problem: i) to use/find another rule ii) predict the missing values of the involved attributes and try to predict the specific value again (creating this way a chain between attribute rules) iii) to combine the two previous processes.

Although little attention has been devoted to the impact of combining the processes of learning and predict, Michael [10] presents the possible variations of such an iterative approach and how it better exploits the recovered data in order to achieve further imputation of incomplete datasets.

Therefore, the two major imputation policies of simultaneous learning and prediction described in the study of Michael [10] are designed, tested and compared with the basic approach of learning and iteratively predict. Yet some, could argue that the success of an imputation approach does not depend only on the combination of the discovered rules and the way to apply them but also on other parameters like i) the BLA and its ability to cope with missing values, ii) the *epochs* (iterations) needed to impute the data, iii) the degree of correlation between attributes, iv) the percentage of missing attributes and the reason of their missingness.

Algorithm 1: FLTC imputation policy

Input: BLA← base learning algorithm; D^* ← incomplete dataset; e← epochs; n←number of attributes
Output: D^+ ← the imputed dataset
$D_0 \leftarrow D^*$;
i←0;
P_{i+1}← DiscoverPredictors(BLA,D_i,n);
PoL_{i+1}← FindPolicyLayer(D_i,P_{i+1});
while $i < e$ **do**
  D_{i+1}← ImputeData(D_i,P_{i+1},PoL_{i+1});
  i←$i+1$;
return D_{i+1};

5 Imputation Policies

Consider a medical incomplete dataset D^* with multiple rows and columns corresponding to real-valued patient characteristics $x_i \in A$ with $i \leq n$ and a base learning algorithm BLA modified to cope with incompleteness. Each approach aims to create a policy $\mathbb{P}$ of predictor layers $< PoL_1, PoL_2, ..., PoL_d >$ where d is the depth of the chained layers and PoL_i the ordered set of predictors of each epoch.

First Learn, Then Chain (FLTC): Within this approach proceed as follows: First call the BLA on the list D^* to create one predictor P_1^i for each $x_i \in A$. Then a policy $\mathbb{P}$ will be created by ordering the initial set of predictors P_1 within each layer PoL_d where $d \leq epochs$ and the policy layers within the policy $\mathbb{P}$. Proceed independently from learning process to predict missing values in D_1^* using $\mathbb{P}$ in order to produce the final D_{epochs}^+. Algorithm 1, henceforth denoted by $FLTC$, will be considered as the basic approach for the fair comparison of the simultaneous parallel learning and prediction variations.

Simultaneous Parallel Learning and Prediction *(SLAP)*: Within this approach learning and prediction do not proceed independently, but but together at each epoch. First layer PoL_1 of predictors will be created by using the dataset D^* and will include one predictor P_1^i for each attribute $x_i \in A$, after takes place the prediction phase where the D_1^+ will be generated using D^* and PoL_1. Subsequently a new learning phase will take place to create PoL_2 of predictors by using D_1^+ and after predicting missing values to lead to D_2^+ and so on. This process eventually creates a chained policy of layers $\mathbb{P} =< PoL_1, PoL_2, ..., PoL_d >$ together with the imputed version D_d^+ of D^*, where $d = epochs$ the depth of the chained layers.

There are two ways to continually learning from an incomplete dataset; the first (Algorithm 2), is to discover predictors all over again in each learning phase, henceforth denoted by $SLAP - NP$, and the second (Algorithm 3) is to update predictors at each learning phase and henceforth denoted by $SLAP - UP$. Each learning phase uses the latest imputed version of the dataset D_{d-1}^+. Thus, theoretically in both cases PoL_d predictors will be an improved version of those in PoL_{d-1} where in this approach $d \leq epochs$.

In all cases the predictors are applied in parallel for each missing $x_i \in R$. Each predictor P^i proceeds independently of the others to predict the missing value of x_i in R. After recovering all missing values, all parallel versions of R will be merged to create the final version R^+. Repeat

for each R in the dataset.

Algorithm 2: SLAP-NP imputation policy

Input: BLA← base learning algorithm; D^* ← incomplete dataset; e← epochs; n←number of attributes
Output: D^+ ← the imputed dataset
$D_0 \leftarrow D^*$;
i←0;
while $i < e$ **do**
  P_{i+1}← DiscoverPredictors(BLA,D_i,n);
  PoL_{i+1}← FindPolicyLayer(D_i,P_{i+1});
  D_{i+1}← ImputeData(D_i,P_{i+1},PoL_{i+1});
  i←$i+1$;
return D_{i+1};

Algorithm 3: SLAP-UP imputation policy

Input: BLA← base learning algorithm; D^* ← incomplete dataset; e← epochs; n←number of attributes
Output: D^+ ← the imputed dataset
$D_0 \leftarrow D^*$;
i←0;
while $i < e$ **do**
  P_{i+1}← DiscoverPredictors(BLA,P_i,D_i,n);
  PoL_{i+1}← FindPolicyLayer(D_i,P_{i+1});
  D_{i+1}← ImputeData(D_i,P_{i+1},PoL_{i+1});
  i←$i+1$;
return D_{i+1};

6 Base Learning Algorithms

A BLA is designed to better perform under complete datasets. It has been shown that the PAC learning semantics can be extended to deal with arbitrarily missing information [21] , and that PAC learning algorithms can easily modified to cope with such experiences [9]. In our experiments we tested two algorithms modified to cope with incompleteness, the Winnow2 [8] and a Back Propagation NN algorithm [18].

Winnow2: For each attribute x_i, where $i \in \{1, ..., n\}$, generate a set of weights $W_i =< w_1, ..., w_{n-1} >$ and use the prediction function $f(x_i, Wi, R^*)$ which, based on the W_i of each attribute and the available values of a record predicts the missing value of x_i. Thus W_i could be considered as the predictor of x_i and the set $W =< W_1, .., W_n >$ as the predictors of the

learning phase.

$$f(i, Wi, R) = \begin{cases} 1, & \text{If } \sum_{j=1, j\neq i}^{n} w_j x_j > \theta \\ 0, & \text{otherwise} \end{cases}$$

Where θ is a real number denoted as *threshold*.

Modifications in the learning phase: Attributes with missing values in R do not contribute to weight sum of x_i.

Recovery phase: For each attribute with a missing value calculate in parallel $f(i, Wi, R^{*=0})$, where missing values in R^* are replaced with 0 and $f(i, Wi, R^{*=1})$, where missing values in R^* are replaced with 1. If $f(i, Wi, R^{*=0}) = 1$ then predict 1 (because the available values are able to define the result), else if $f(i, Wi, R^{*=1}) = 0$ then predict 0 (because setting missing values equal to 1 is not enough to change the result), otherwise predict $*$.

$$P^i = \begin{cases} 1, & \text{if } f(i, W_i, R^{*=0}) = 1 \\ 0, & \text{if } f(i, W_i, R^{*=1}) = 0 \\ *, & \text{otherwise} \end{cases}$$

NN back propagation algorithm: For each attribute x_i select all records where x_i is not missing and split them into two datasets: the first includes the values of all attributes except x_i and denoted as *input* data, and the second and includes all the counterpart values of x_i and denoted as *target*. *Input* and *target* data will be used to train a two-layer feed-forward NN with *sigmoid* hidden *n-1* neurons and one *sigmoid* output neuron for predicting the missing values of x_i.

Thus, NN_i could be considered as a predictor of x_i and the set $NN =< NN_1, .., NN_n >$ as the predictors of the learning phase.

Modifications in the learning phase: Use the *trainbr* as a network training function, which updates the weight and bias values according to $Levenberg - Marquardt$ [5] optimization. The *trainbr* algorithm generally works best when the network inputs and targets are scaled, so that they fall approximately in the range $[-1, 1]$. In the case that data are binary we have to make a data mapping, values equal to 0 are set to -1 and missing values are set to 0, minimizing this way their impact of missing values in the training function.
Recovery phase: For each attribute x_i in D^+_{d-1} where d the current epoch, use NN_i to calculate in parallel all missing values in the dataset D^+_{d-1}. At the end of recovery phase concatenate all recovered versions to produce D^+_d.
If $P^i > \theta$ then consider prediction equal to 1 else if $P^i < -\theta$ then consider prediction equal to -1, otherwise as missing $*$.

7 Empirical Investigation

To provide and illustration of how each approach performs, a publicly available binary dataset has been selected. The LUCAS (LUng CAncer Simple set) dataset contains data generated artificially by Causal Bayesian Networks with binary variables. This dataset includes $n = 11$ attributes at each records and $r = 11000$ records. For more details, see

http://www.causality.inf.ethz.ch/data/LUCAS.html. The binary set selection came as a result of the limitation of the one of the $BLAs$ selected for comparing the policies. Despite the fact that one of the attributes is not correlated with the others and takes arbitrary values although we still try to predict it in order to test each approach under more realistic scenarios.
Data preparation phase: Initially an incomplete MCAR version D^* of D is created by copying the D and removing the appropriate number of values based on the experiment completely randomly from all attributes.
Learning phase: Select randomly the next R in order to train the predictors.
Recovery phase: Predict in parallel all the missing values of R^*. Select randomly an i where $i \leq n$ in order to predict the missing values of R^*.

Evaluation phase: For better conclusions we calculate the following: correctly recovered values cr; incorrectly recovered values fr; and unable to recover values nr by comparing initial D with its final imputed version D_epochs^+ after 20 epochs. In cases of using the NN as BLA the value of θ is 0.8.

7.1 Improving Predictors by Using Recovered Values

With FLTC the recovered values contribute to the further imputation of the dataset by supporting predictors, which previously abstain to predict due to incompleteness, to finally success in the current epoch. Graphs a. and d. of Figure 1 demonstrate the performance of FLTC approach for both Winnow and NN BLAs for each epoch after removing the 30% of D. The significant improvement in the dataset completeness after the first epoch, even for the FLTC approach, bears out the conclusion that all approaches continue to recover values even after the first epoch. However this phenomenon stops very early for the case of FLTC. Furthermore, SLAP-NP continues to recover missing values for more than 20 epochs (see Figure 1 b. and e.). The underlying argument in favor of SLAP-NP is that unlike FLTC, recovered data do not only contribute to the recovery phase but also contribute to the learning too, producing this way an enhancement set of predictors. To this end SLAP-UP (see Figure 1 c. and f.) has identical behavior to SLAP-NP but tends to be more conservative in its predictions.

Thus the main theoretical premise behind the success of the SLAP-NP is to recover enough missing values in order to support the enhancement of next epoch's predictors. SLAP-NP success against FLTC is based on its ability to discover new relations between attributes and efficiently cope with incomplete and noisy dataset too. Since, there is no guaranty that the new set of predictors will be an enhancement version of the previous one, further research in this area could include a solution which include previously discovered predictors in the current set, instead of discarding them.

In addition, there is an important difference between FLTC and SLAP approaches: repeating learn-predict process of SLAP-NP can improve completeness but also negatively affect soundness (see Figure 1). This is due to the fact that the D_d^+ may include mistaken predictions, thus any farther learning process using D_d^+ may lead to a new set of unsound predictions and predictors, generating an avalanche of mistaken predictions. A closer look at Figure 1 indicates that while epochs pass the average number of correctly recovered values falls and the incorrectly recovered values increase. This issue leads to the need of such solutions that will bound the effects of incorrect predictions. Under this perspective the SLAP-UP policy is a promising one.

On the basis of the evidence currently available (see Figure 1), it seems fair to suggest that SLAP-UP has the ability to carry previously discovered knowledge creating a stronger chain between the policy layers. Further evidences are extracted by Figure 2, illustrates the

percentages of final correctly and incorrectly recovered values on the initial version of D^* (30% MCAR missing values) using the predictors of the PoL_d created by each approach (SLAP-NP left and SLAP-UP right) after d recovery epochs of D^*. The data gathered in the tests suggest that SLAP-UP unlike SLAP-NP tends to successfully recover a larger amount of missing values of the initial D^* and follows a stable improving behavior as far as concerning its ability to impute, although SLAP-NP shows also an improving incline.

Figure 2 illustrates the percentages of final correctly and incorrectly recovered values on the initial version of D^* using the predictors PoL_d generated by NN SLAP-NP (left) and NN SLAP-UP (right) after d epochs, lends support to the claim that the rules created with SLAP-NP at each epoch using the D^+_{d-1} do not carry the knowledge of previous epochs (altough D^+_{d-1} is an imputed version of D^*) thus they abstain to satisfyingly impute the D^*. This is why the percentage of correctly recovered values has an unstable behavior. On the other hand each epoch of SLAP-UP generates predictors that are able to successfully recover missing values of the initial D^*, fact that indicates that SLAP-UP approach allows the knowledge transfer across the learning phases.

Additionally, evidences suggest (see Figure 1 and 3) that there are more ways to create stronger chains between the predictors. Figure 1 illustrates the performance of two different BLAs tested with the same dataset and approaches and bears out the fact that the choice of the BLA is able to alter approach's ability either to respond with an answer very soon (in this experiment see Winnow) or to behave more conservatively (in this experiment see NN). Moreover, the modifications of the BLA in order to overcome missingness (e.g. changing parameters to the NN BLA like the number of hidden layers, the threshold, the activation function or the size of the training set) can farther contribute to the behavior of each approach. Nevertheless, creating a tight chain between the predictors only guarantees that the recovery process will produce a less noisy dataset but costs in completeness. Furthermore, the difference between SLAP-NP and SLAP-UP is not as clear-cut and parameters like i) the selected BLA ii) the number of missing values and iii) the number of available records can significally affect each solutions. Further research in this area will be presented in the next sections.

7.2 How Dataset Size and Missingness Alter the Behavior of Each Approach

In this section we discuss how the number of available values in the dataset D^* affects the behavior of each approach. On the basis of the evidence currently presented on the previous section, it seems fair to suggest that BLA affects the degree of flexibility of the predictors; but what about the number of missing values? In order to clearly identify the impact of data availability, we performed several tests with the same settings as previously with changing only the size of the initial dataset (using data from the same dataset ensures fairer comparisons between use cases scenarios). Figure 3 illustrates the results of soundness and completeness of three different sizes of the D^* ($1D^*$,$2D^*$ and $3D^*$ with 500, 1000 and 11000 records respectively) for each approach with NN as BLA, while the number of MCAR missing values increases from 10% up to 60%.

A closer look at the results illustrated at Figure 3 indicates that data availability either because of the small number of available records (see Figure 3 a. d. g.) or increased number of missing values is able to alter the behavior of each approach. Although the performance of FLTC is not affected significantly by the data availability, however in case of SLAP-NP small datasets (see Figure 3 b.) tend to have the same performance even in cases of high percentages of missing values. In cases with large datasets the behavior of the SLAP-NP

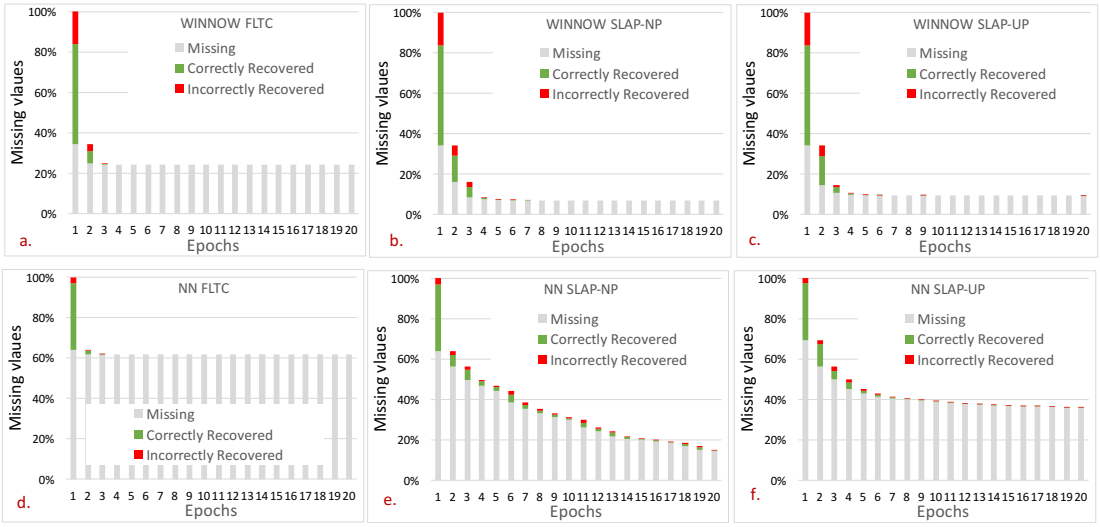


Figure 1: Each graph represents the performance of each approach using as dataset the D^* with 30% missing values. First row illustrates the performance of the Winnow and the second row the NN. First column illustrates the performance of FLTC for each BLA, second column the performance of SLAP-NP and third column the performance of SLAP-UP. Each bar of the graph represents the percentages of correctly, incorrectly and unable to predict values of the remaining missing values of previous epoch.

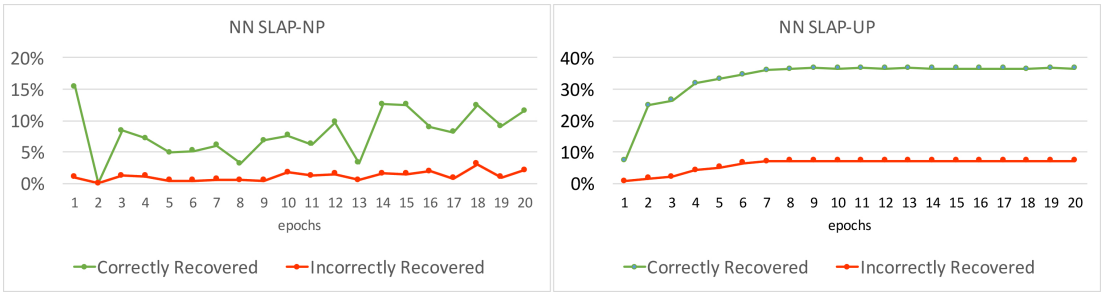
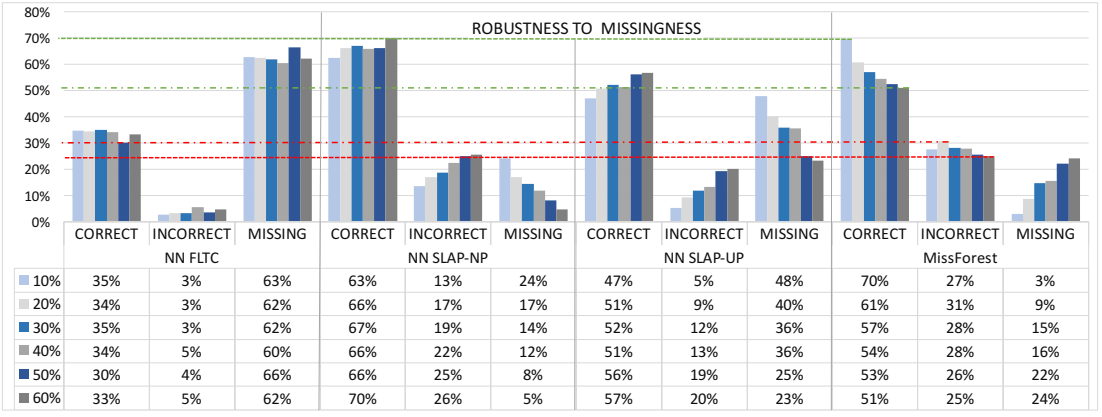


Figure 2: Percentages of final correctly and incorrectly recovered values on the initial version of D^* using the predictors of the PoL_d created by each approach (SLAP-NP left and SLAP-UP right) after d recovery epochs of D^*.

approach changes dramatically (see Figure 3 e. h.) by increasing completeness and sacrificing soundness. Moreover SLAP-UP approach follows the same behavior regardless of the dataset size. The explanation of such a behavior is that limited available data create high flexible predictors which can easily conclude to a prediction. An interesting observation is that datasets with larger number of records and high rate of missingness (see Figure 3 g. h. i.) strengthen the confidence of such predictors which lead to soaring errors that also contribute to the learning phase of the next epochs. Unfortunately even SLAP-UP is trapped to a chain of errors, but at least it continues to predict correctly in the same rate. On the other hand the SLAP-NP is only affected by the errors of the previous epochs and not by the chained predictors too.

Figure 3: Average final results of correctly, incorrectly and unable to predict values using three different subset of the D^* ($1D^*$,$2D^*$ and $3D^*$ with 500, 1000 and 11000 records respectively) for each approach with NN as BLA and MCAR missing values increase from 10% up to 60%.



	NN FLTC			NN SLAP-NP			NN SLAP-UP			MissForest		
	CORRECT	INCORRECT	MISSING	CORRECT	INCORRECT	MISSING	CORRECT	INCORRECT	MISSING	CORRECT	INCORRECT	MISSING
10%	35%	3%	63%	63%	13%	24%	47%	5%	48%	70%	27%	3%
20%	34%	3%	62%	66%	17%	17%	51%	9%	40%	61%	31%	9%
30%	35%	3%	62%	67%	19%	14%	52%	12%	36%	57%	28%	15%
40%	34%	5%	60%	66%	22%	12%	51%	13%	36%	54%	28%	16%
50%	30%	4%	66%	66%	25%	8%	56%	19%	25%	53%	26%	22%
60%	33%	5%	62%	70%	26%	5%	57%	20%	23%	51%	25%	24%

Figure 4: Average final results of correctly, incorrectly and unable to predict values for FLTC, SLAP-NP, SLAP-UP and MissForest approaches using the D^* with the missing values to increase from 10% up to 60%.

8 Related Work

The FLTC is a common approach, where learning phase is applied only over the subset of completed records [12]. However in cases where this is not feasible the more advanced approaches

take advantage of SLAP-NP capabilities. Such a case is [3] that uses the Farmer [11] as BLA and combines simultaneous learning of multiple predictive rules with differential scoring of evidence. In the same spirit of simultaneous learning and chaining of multiple predictive rules, [6] considers a set of partially observed assignments closely following the one of [9], and chains learned rules. [14] creates predictors in a hierarchical manner in order to escape the infeasibility of learning arbitrary concepts. Firstly they create relevant subconcepts of the target concept, and then the target concept itself, having this way a chaining list of predictors, with subsequent predictors being trained on the predictions of earlier ones.

Additionally, the problem of missing data could be considered as a special case of the classification problem where each attribute could potentially be considered as a label for the remaining set of attributes. Keeping this in mind we expanded our research to the area of classifying incomplete examples [19], where each label could be seen as an attribute in the dataset and the default classification functions as the predictors of the attributes. Furthermore the FLTC approach is similar to Binary Relevance method [13], where each classifier is independent from each other [24] and the dependency relations between classes are ignored. In order to overcome this issue, a study of Read [13] suggests the Classifier Chain Model (CC). CC involves $|L|$ binary classifiers linked along a chain, where each C_i classifies the set of attribute-labels $(x, l_1, ..., l_{i-1})$ and generates a BR problem which is associated with label $l_i \in L$, chaining this way the results of the previous classification step with the next one. Thus each step where the classification is expanded can be seen as a new iteration.This study [13] also proposes ECC which trains m CC classifiers.

Another, two step approach of chaining classifiers is suggested by this study [24] which, i) obtains a dependency structure for the class variables, and ii) based on the dependency structure, builds a classifier chain. Respectively to the concept of chaining predictors. A work of Royston [15] demonstrates (MICE) approach by chaining equations. In the same spirit MissForest [20] uses the random forests [1] instead the NN and follows the SLAP-NP approach to impute the dataset.

In order to have a better evaluation of the performance of each approach we also tested MissForest using again the D^*. For a fair comparison we set the same threshold θ in the recovery process as we did with NN and the results are illustrated in Figure 4. We can see that MissForest performs well and much better than FLTC concerning the completeness. However in the basis of the evidence currently available, it seems fair to suggest that NN SLAP-NP approach performs quite better for both soundness and completeness for all ranges of missing percentages. In comparison the NN SLAP-UP, the MissForest shows slightly better results concerning the completeness but the SLAP-UP outperforms regarding the soundness. While the number of missing values increases MissForest abstains to recover the values unlike with NN which exploits better the available data.

9 Conclusions and Future Work

This work presents an empirical investigation of three natural learning-based imputation policies: training rules once and applying them repeatedly; training new rules before applying them at each iteration; continuing the training of previous rules at each iteration. In line with the theory, we find that an iterative learn-predict approach is preferable. Simultaneous learning and predicting with multiple chained predictors can actually give better results considering soundness and completeness, without leading to a biased system.

In case of SLAP approaches we show that updating the predictors (SLAP-UP) leads to a less complete dataset but more sound comparing with SLAP-NP approach which regenerates

the predictors in each epoch. We also tested how each approach is responding under different number of missing values. Results show that SLAP approach again is able to cope satisfactorily even under extreme cases. However data availability is able to alter the ability of each approach to predict unsound values. In order to have a better understanding we also tested all approaches under different number of records and percentages of missing values. The evidence gathered suggest that in cases with high rates of missing values (over 40%), datasets with big number of records actually prevent the SLAP approaches to perform regarding the soundness and in such cases a smaller portion of the dataset is more preferable.

We used two different types of BLA that have been modified to cope with the incompleteness in order to compare the three natural learning base imputation policies. Additionally we show experimentally that the choice of BLA may affects the soundness and completeness of the process but without minimizing the importance of simultaneous learning and recovering. In contrast the choice of BLA gives to the framework the flexibility to select between more complete and less sound results or more sound and less complete results.

In order to better evaluate our solutions we also performed a set of tests using missForest, a state of the art solutions, which belongs to the SLAP-NP approach. The results provide evidence that the NN BLA has better performance than missForest for both soundness and completeness and that SLAP-UP outperforms concerning soundness.

Further research in this area may involve i) optimizing the order of applying predictors in each iteration with aim to minimize the impact of noisy data ii) identifying the relation between the number of attributes, rates of missing values and dataset size iii) expanding the solution to be able to cope with continuous and mixed-type of attributes.

References

[1] Leo Breiman. Random forests. *Machine Learning*, 45(1):5 32, 2001.

[2] A. P. Dempster, N. M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[3] Janardhan R. Doppa, Mohammad S. Sorower, Mohammad Nasresfahani, Jed Irvine, Walker Orr, Thomas G. Dietterich, Xiaoli Fern, and Prasad Tadepalli. Learning rules from incomplete examples via implicit mention models. In *Proceedings of 20th Asian Conference of Machine Learning (JMLR)*, pages 1–16, 2011.

[4] Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006.

[5] T. Hill, P. Lewicki, and P. Lewicki. *Statistics: Methods and Applications : a Comprehensive Reference for Science, Industry, and Data Mining*. StatSoft, 2006.

[6] Brendan Juba. Implicit learning of common sence for reasoning. In *Proceedings of 23th International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 939–946, 2013.

[7] Kamakshi Lakshminarayan, Steven A. Harp, Robert P. Goldman, and Tariq Samad. Imputation of missing data using machine learning techniques. In Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad, editors, *KDD*, pages 140–145. AAAI Press, 1996.

[8] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[9] Loizos Michael. Partial observability and learnability. *Artificial Intelligence*, 174(11):639–669, 2010.

[10] Loizos Michael. Simultaneous learning and prediction. In *Proceedings of 14th International Conference on Principles of Knowledge Representation and Reasoning*, 2014.

[11] Siegfried Nijssen and Joost N. Kok. Efficient frequent query discovery in farmer. In *Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 2838, pages 350–362, 2003.

[12] Mostafizur M. Rahman and Darryl N. Davis. Fuzzy unordered rules induction algorithm used as missing value imputation methods for k-mean clustering on real cardiovascular data. In *Proceedings of The World Congress on Engineering*, volume I WCE 2012, pages 391–394, 2012.

[13] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333, 2011.

[14] Roland L. Rivest and Robert Sloan. A formal model of hierarchical concept learning. *Information and Computation*, 114(1):88–114, 1994.

[15] Patrick Royston and Ian R. White. Multiple imputation by chained equations (mice): Implementation in stata. *Statistical Software*, 45(4), 2011.

[16] Donald B. Rubin. Inference and missing data (with discussion). *Biometrical*, 63(3):581–592, Dec 1976.

[17] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. John Wiley Sons, New York, 1987.

[18] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall, Upper Saddle River, New Jersey, 1995.

[19] Dale Schuurmans and Russell Greiner. Learning to classify incomplete examples. In *Proceedings of Computational Learning Theory and Natural Learning Systems: Addressing Real World Tasks*, pages 87–105. MIT Press, 1993.

[20] Daniel J. Stekhoven and Peter Buhlmann. Missforest - nonparametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[21] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(1134–1142), 1984.

[22] Leslie G. Valiant. Robust logics. *Artificial Intelligence*, 117(2):231–253, 2000.

[23] Leslie G. Valiant. Knowledge infusion. In *Proceedings of 21th National Conference on Artificial Intelligence (AAAI'06)*, pages 1546–1551, 2006.

[24] Julio H. Zaragoza, L. Enrique Sucar, Eduardo F. Morales, Concha Bielza, and Pedro Larranaga. Bayesian chain classifiers for multidimensional classification. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, 2011.