2015 IEEE Symposium Series on Computational Intelligence

Classification Uncertainty of Multiple Imputed Data

Tuomo Alasalmi, Heli Koskimäki, Jaakko Suutala and Juha Röning
Data Analysis and Inference Group
Faculty of Information Technology and Electrical Engineering
University of Oulu
Finland
Email: {tuomoala, hejunno, jaska, jjr}@ee.oulu.fi

***Abstract*—Every classification model contains uncertainty. This uncertainty can be distributed evenly or into certain areas of feature space. In regular classification tasks, the uncertainty can be estimated from posterior probabilities. On the other hand, if the data set contains missing values, not all classifiers can be used directly. Imputing missing values solves this problem but it suppresses variation in the data leading to underestimation of uncertainty and can also bias the results. Multiple imputation, where several copies of the data set are created, solves these problems but the classical approach for uncertainty estimation does not generalize to this case. Thus in this paper we propose a novel algorithm to estimate classification uncertainty with multiple imputed data. We show that the algorithm performs as well as the benchmark algorithm with a classifier that supports classification with missing values. It also supports the use of any classifier, even if it does not support classification with missing values, as long as it supports the estimation of posterior probabilities.**

I. Introduction

Sometimes the uncertainty of a classification result can be as important as the classification result itself. This might be the case with e.g. decision support systems in medical domain where human lives can be on the line. In this case, the extra information about the uncertainty of the classification result could be very valuable. In other applications, the uncertainty could be used to reject samples with too high uncertainty or direct them to manual inspection. Uncertainty estimate could also be used to analyse the possible sources of the uncertainty. Based on this information, the classification model could possibly be fine tuned or more data could be collected about the uncertain samples to reduce uncertainty and improve accuracy.

Nature of the classification problem itself and the classification algorithm are at the core when uncertainty is quantified. In this article, focus will be on real-world data sets with, in this case artificially introduced, missing data. This is a case where multiple imputation is of great value [1] and the missing values in explanatory variables can add to the uncertainty. Thus a way is also needed to quantify the amount of uncertainty that is caused by some value being missing or any combination of missing values.

Classification algorithms have traditionally been developed using complete data sets and most require values for all variables to be present to work. There are a few exceptions, such as Naive Bayes (NB) classifier, that can handle missing values in data but they are rather exceptions than the norm. Classifiers that have proven to perform best generally are Random Forest and Support Vector Machine (SVM) [2], both of which cannot handle missing values, at least natively. Many real world data sets are, however, cursed with missing data. There are many reason why this is the case. For example many medical data sets contain patient records that do not all contain the same variables simply because it is not realistic to expect every patient has been measured for every possible blood test etc. Many of the incomplete variables could, however, add value to the final model if the modeling tools were able to use that information.

There are three main approaches to handling missing data. The most common approach is to simply ignore any data samples containing missing values. This is called complete case analysis or listwise deletion. Sometimes all available data is used for each analysis separately. This is called available-case analysis or pairwise deletion [1]. Another approach is to impute the missing values with something sensible or to use model-based methods to model the distribution of the variables to make analysing the whole data set possible [3]. Both deletion and imputation ignore the uncertainty that is caused by the unknown true values of the missing data and deletion does not make it possible to evaluate test samples that are incomplete. Sometimes deletion or imputation can be made behind the scenes during data preparation or even made implicitly by statistical software. Whether deletion or imputation was used, the modeller might be totally unaware that the data set has had missing values in the first place as well as how the missing values were handled. If single imputation had been used, this tends to bias the analysis results and underestimate the amount of variance in the imputed variables. Multiple imputation, where two or more possible values are imputed for each missing value, solves these problems [1], [4] but machine learning algorithms cannot directly handle multiple imputed data sets.

It is straightforward to estimate the uncertainty, i.e. standard error, of statistical estimands with multiple imputed data by using simple rules, called Rubin's rules [1], [4]. These rules take into account the between-imputation variance and within-imputation variance when estimating the standard error of the statistical estimand in question. On the other hand, for classification result in the case of complete or singly imputed data, the estimated posterior probability of the predicted class, when available, can be used as an uncertainty measure directly or through a derived measure. Using single imputed data can, however, bias the results and it will most certainly suppress the inherent variability in the data set therefore ignoring the uncertainty about the true values of the missing values.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.32

Multiple imputation solves the problems of single imputation. Nevertheless, to the best of our knowledge, uncertainty estimation of classification results when multiple imputed data is used has not been studied. In this article we propose an algorithm to estimate the uncertainty in this case.

This article is divided as follows. Background of the topic is presented in Section II. The proposed algorithm is then derived in Section III and some practical experiments with the algorithm are introduced in Section IV. The results obtained in the experiments are then presented in Section V and discussed in Section VI. Section VII concludes the article.

II. Background

Missing data is common problem with many real-world data sets [5]–[7]. As most classification algorithms can directly only use complete data, the missing data points need to be handled somehow. As stated in the introduction, just ignoring missing values is common practise and this can even be hidden from the analyst by removing the missing values either in the preprocessing stage or in the analysis stage by the default missing data handling method in the statistical software at hand. For example in R, in the package e1071, the default action in training an SVM model is listwise deletion if missing values are encountered. Pretending there is no problem of missing data might be convenient but it will throw away valuable data at best and severely bias the results at worst. Some data sets might even be completely useless if missing data is ignored and enough data points are missing. This becomes apparent when looking at Table I. If all features are required to be present for the analysis, only a very small percentage of the values can be missing or the data set becomes useless.

TABLE I. Expected percentage of complete cases with different number of features and percentages of missing values.

	NUMBER OF FEATURES				
MISSING	5	10	20	40	80
1 %	95.1 %	90.4 %	81.8 %	66.9 %	44.8 %
2 %	90.4 %	81.7 %	66.8 %	44.6 %	19.9 %
5 %	77.4 %	59.9 %	35.8 %	12.9 %	1.7 %
10 %	59.0 %	34.9 %	12.2 %	1.5 %	
20 %	32.8 %	10.7 %	1.2 %		
40 %	7.8 %	0.6 %			
60 %	1.0 %				

There are three distinct mechanisms that describe the probability of each data point being missing. If all data points have exactly the same probability of being missing, the data are described as being missing completely at random (MCAR). With MCAR, factors unrelated to the data itself are the causes of missing data. If the probability of a data point being missing can be explained by observed data, then the data are said to be missing at random (MAR). If neither of these conditions hold, the data is missing not at random (MNAR). A thorough explanation of missing data mechanisms and their implications can be found in e.g. [1]. In most cases, assuming MAR is reasonable. In practice this means that missing values can be estimated with information available from the observed variables [1] but this is not always the case and the choice of the missing data handling method should take into consideration the mechanism behind the missing values [3].

A. Handling missing data

García-Laencina et al. [5] and Aste et al. [3] review missing data handling methods in pattern classification tasks. One method is to delete cases where some of the data is missing but this approach is not recommended unless the amount of missing data is small and the overall amount of data is large. Another method is missing data imputation which will be discussed in more detail in the next section. A third class of methods are model-based procedures in which the joint distribution of the variables are modelled and then used by the classifier. The final class of methods are machine learning approaches which are able to deal with missing data while avoiding explicitly imputing the missing values.

One of the best known model-based algorithm that is used to handle data with missing values is the expectation-maximization (EM) algorithm and it has been successfully used with many problems with incomplete data. EM can be used to e.g. train Gaussian mixture models [8].

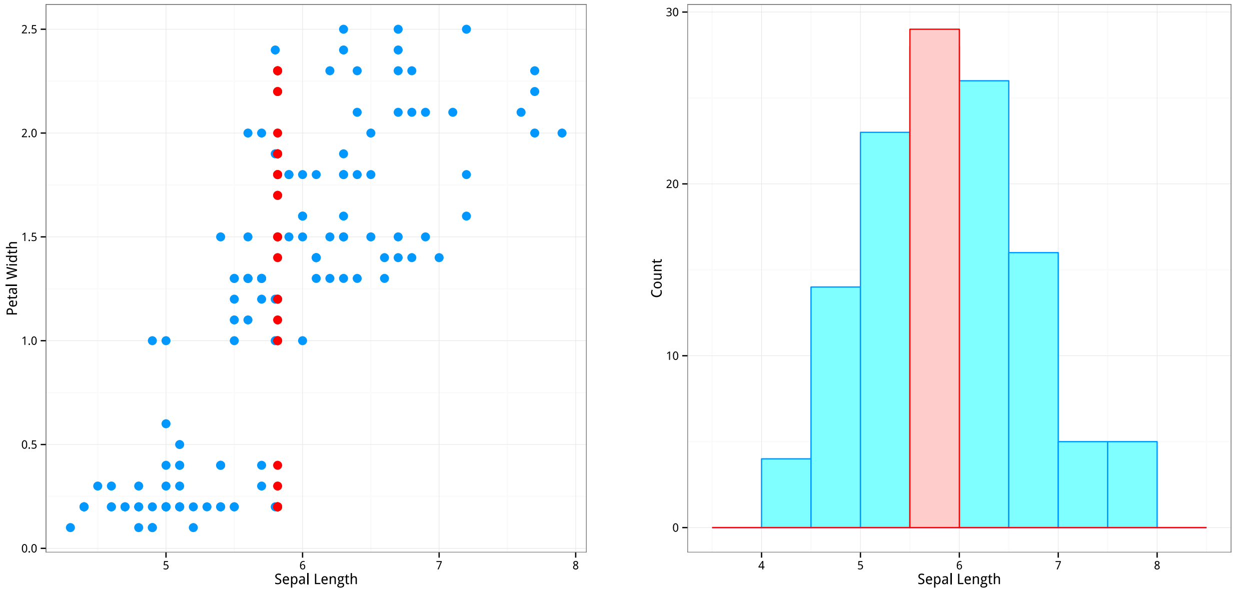
In the machine learning approaches, the classifiers themselves are able to deal with missing values. Naive Bayes classifier, for example, does this simply by ignoring the missing features. In ensemble methods, such as neural network ensembles, several classifiers are created that each use a different combination of complete features. Some decision trees can handle missing values natively, too. For example, ID3 creates an additional edge for the unknown values. Fuzzy approaches, where missing values are represented by an interval of possible values, have also been used. Also, Support Vector Machines have been extended to handle missing values by harnessing the EM algorithm [5].

Even if some classifiers can handle incomplete data, imputation can offer some benefits. It allows one to use any desired classifier and if multiple imputation is used, the inherent uncertainty about the true values of the missing data points will be captured as variation between the imputations. Also, comparison of classifier performance is easy when the missing data is handled separately from the classification problem. In this work we will concentrate on imputation as the missing data handling method.

B. Missing data imputation

If the classifier we would like to use cannot handle missing values and we want to keep all the available data for analysis, we need to replace the missing values by some estimate. This process is called missing value imputation. There are several ways how to choose the imputed value. To demonstrate the effect that the chosen imputation method can have on the characteristics of the data, we will use an example where 20 % of the data is removed completely at random from the Iris data set [9].

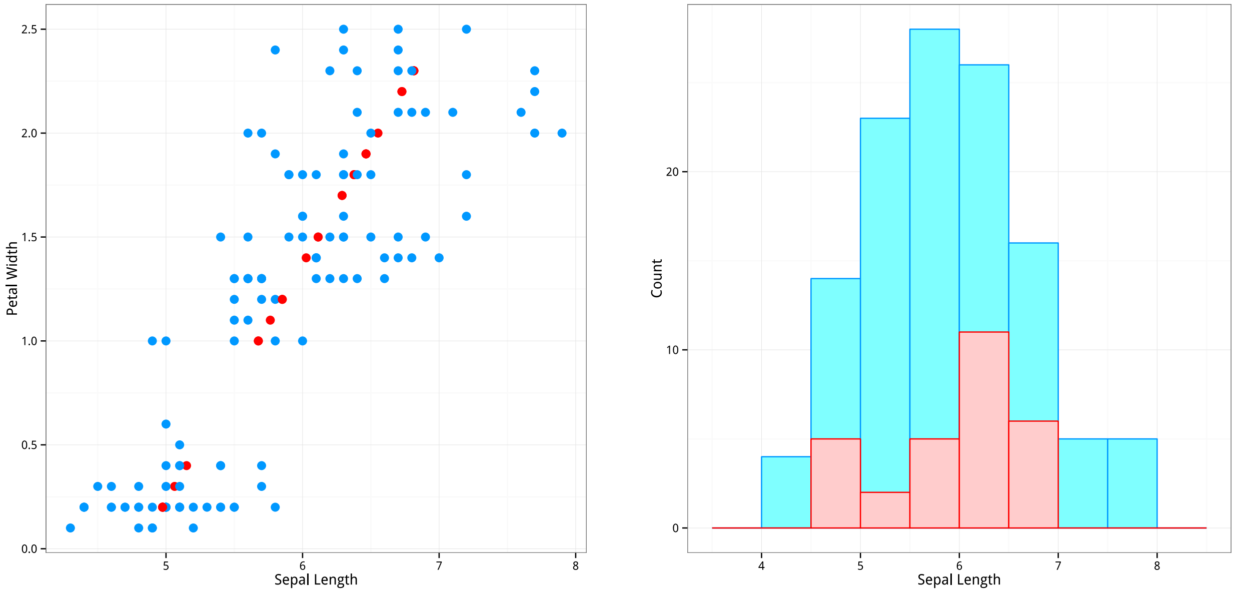
Probably the simplest way to fill in missing values is by mean imputation, i.e. replacing any missing values with a mean of that particular variable. For categorical data, the mode can be used. Figure 1 shows how mean imputation behaves when it is used with the example data. Mean imputation will obviously underestimate the amount of variance in the data. This can be clearly seen from the scatter plot and the histogram of the complete data and imputed data when they are plotted together.



(a) Scatter plot of the features.

(b) Histogram of Sepal Length.

Fig. 1. Mean imputation. Blue indicates observed values and red indicates imputed values.



(a) Scatter plot of the features.

(b) Histogram of Sepal Length.

Fig. 2. Regression imputation. Blue indicates observed values and red indicates imputed values.

Mean imputation will also disturb inter-variable relations and bias any estimates other than the mean itself. Even the mean estimate will be biased by mean imputation if the data are not MCAR [1].

A step forward is regression imputation, where information from other observed variables is used to predict the missing value. A regression model is built from the observed data which is then used to calculate the replacement value for missing data points. The values produced are the most likely values *according to the model* but variation is again suppressed and therefore it is unlikely that the predicted values correctly represent the distribution of the true values had they been observed. An example where the missing values were imputed with regression imputation can be seen in Figure 2. Using regression imputation also affects correlation; the imputed values have a correlation of 1 as they are located on a line thus increasing the combined correlation of the observed and imputed values. If the data are MCAR, mean estimates will be unbiased with regression imputation. An illustrative way to think about regression imputation is, however, that if the prediction model did indeed produce perfect estimates of the missing values, no information was really missing in the first place [1].

In stochastic regression imputation, adding noise to regression imputation restores the lost variability in the imputed data points. A regression model is first fit to the complete data, residual variance is estimated, and finally the imputed values are drawn according to these parameters. As can be seen from Figure 3, adding noise to the imputed values really opens up the distribution of the imputed values and the values seem like they could actually have come from the original distribution even though they are not. Stochastic regression preserves the relationship and correlation of the variables but it does not deal with the inherent uncertainty of the imputed values. Regression imputation and stochastic regression imputation can also produce values that are impossible, such as negative values for variables that are in reality strictly positive.

In multiple imputation, $m > 1$ complete data sets are created by replacing the missing values with plausible values. There are several possibilities for which model to use for the imputation and the model can be chosen individually for each variable. Each complete data set is then analysed individually. Results from each of the analyses can then be pooled to a final estimate using simple pooling rules called Rubin's rules [4]. The imputed data sets differ in the imputed values whereas the complete values are identical in all of the data sets. The variation in the imputed values indicates that there was uncertainty caused by the unobserved. The pooled statistical estimates are, however, unbiased and have correct statistical properties under appropriate conditions [1].

C. Classification with multiple imputed data

State-of-the-art classification algorithms, e.g. Support Vector Machines and Random Forests [2], require all variables to be present in both training and test samples to work. To the best of our knowledge, Belanche et al. [10] are the only ones who have used multiple imputed data sets in classification problems with such algorithms. They developed two different algorithms on how to combine classification results to form a final prediction when multiple imputed data sets were used. In one algorithm the training data set was imputed m times, merged into a single large data set, and then used to train a classifier, SVM in this case. Test data set was then concatenated with the stacked training data set, imputed m times, and extracted from the training samples for prediction. Each of the m now complete test data sets were used for prediction using the classifier that was trained in the previous step. Therefore, for each sample in the test data set, m predictions were produced and a majority vote was used to form the final prediction for each test sample. A diagram of the procedure is depicted in Figure 4. In the second algorithm, training data was again imputed m times and for each of the complete data sets, a classifier was trained. Test data set was then concatenated with each of the m imputed training data sets, imputed once (i.e. $m = 1$) and then used for prediction. Again, m predictions were produced and a majority vote was used to form the final prediction. The former algorithm, called 1MI, was determined to work generally better so we will use that algorithm in this work also.

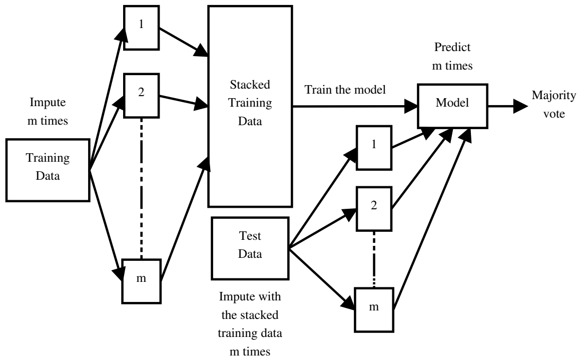


Fig. 4. Classification of multiple imputed data with the 1MI algorithm.

D. Uncertainty measures

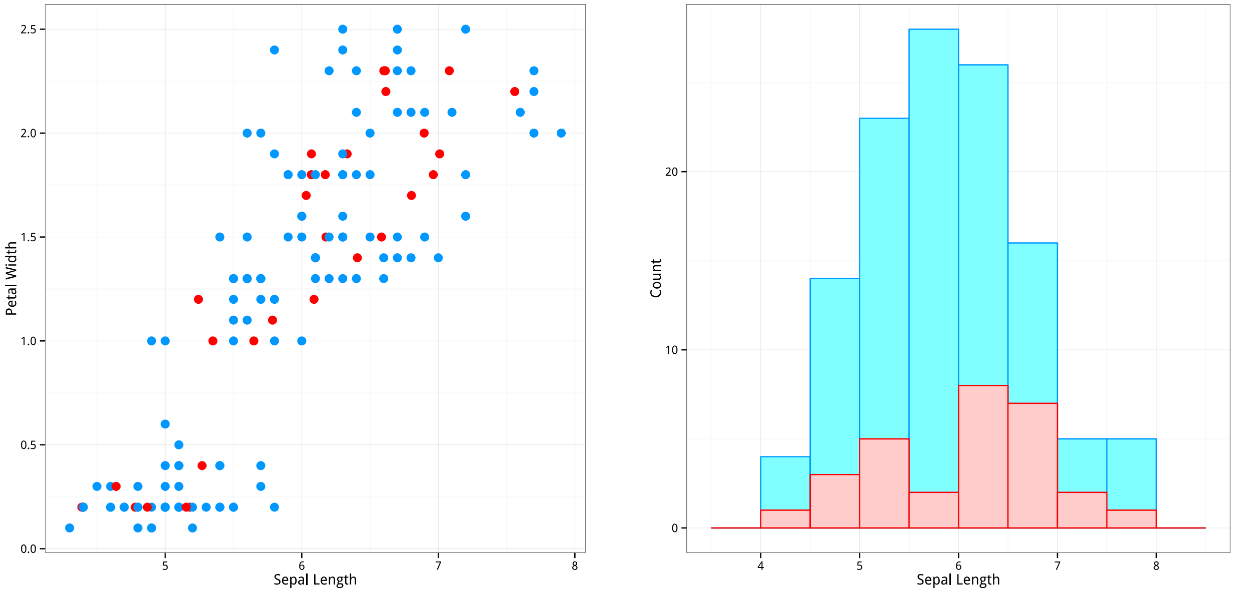
Classification uncertainty can be expressed using the posterior probability of the assigned class if the classifier used supports estimating that. Using the posterior probability directly gives an idea about the (un)certainty of the classification result. An uncertainty measure can also be derived from the posterior probabilities [11]. The measure is defined in (1) where p_i is the posterior probability of class i and n is the number of classes. This measure takes into account also the posterior probabilities of the other classes. For comparing with the uncertainty measure we will propose in this article, this uncertainty measure will be used.

$$U = 1 - \frac{\max_{i=1,\ldots,n}(p_i) - \sum_{i=1}^{n}(p_i)/n}{1 - 1/n} \in [0,1] \qquad (1)$$

For statistical analyses, standard error is used as a measure of uncertainty in the statistic at hand. It can be defined as "the standard deviation of the sampling distribution of some statistic" [12]. But why are we interested in standard errors? Because when dealing with missing data, there is a proven way to estimate the standard error of a statistic when multiple imputation is used. That way is to use pooling rules called Rubin's rules [1], [4] and we will use the same underlying principle later in defining an uncertainty measure in the case of classification with multiple imputed data.

Rubin's rules for combining statistical analysis results of repeated-imputation are defined in (2)-(5). Suppose we are estimating some statistic $\bar{Q}$ from a data set containing missing values. After using multiple imputation, we first analyse each imputed data set separately to obtain estimates of the statistic $\hat{Q}$ for each of imputed data sets. The final estimate can then be obtained as in (2). Here $\bar{Q}$ is calculated by taking the mean of the analyses results with different repeated-imputation data sets.

There are three sources of variation regarding the estimate $\bar{Q}$:



(a) Scatter plot of the features.

(b) Histogram of Sepal Length.

Fig. 3. Stochastic regression imputation. Blue indicates observed values and red indicates imputed values.

- Variance caused by the fact that only a sample of the population is observed and not the entire population. This is called within-variance.
- Variance caused by the missing values. This is called between-variance.
- Simulation variance caused by the fact that $\bar{Q}$ itself is estimated for finite m.

Within-variance $\bar{U}$, defined in (3), is the average of the sample variances within each imputed data set. To account for the uncertainty in the statistic caused by missing values, between-variance B is calculated as in (4). Because $\bar{Q}$ is itself an estimate using finite m, variance contribution of this needs to be taken into account, too. It has been shown in [4] that the variance of this factor is B/m and this is taken into account in (5) where the total variance T of the estimate is calculated as a combination of within-variance, between-variance, and simulation variance.

$$\bar{Q} = \frac{1}{m} \sum_{l=1}^{m} \hat{Q}_l \tag{2}$$

$$\bar{U} = \frac{1}{m} \sum_{l=1}^{m} \bar{U}_l \tag{3}$$

$$B = \frac{1}{m-1} \sum_{l=1}^{m} (\hat{Q}_l - \bar{Q})(\hat{Q}_l - \bar{Q})' \tag{4}$$

$$T = \bar{U} + B + \frac{B}{m} = \bar{U} + \left(1 + \frac{1}{m}\right) B \tag{5}$$

III. Classification Uncertainty of Multiple Imputed Data

In this section we will derive a novel uncertainty measure that can be used even in the presence of missing data and even if the chosen classifier does not support learning and prediction with incomplete samples. The only requirement is the ability to estimate posterior probabilities of the classification result.

Like with statistical estimands, the uncertainty of classification result can be of interest in many cases. When the data set contains missing values, multiple imputation is an excellent way to capture the uncertainty about the true values of the missing values. The uncertainty of the true nature of the missing values can be seen as variation between imputations, larger variation meaning higher uncertainty. In a classification task, this will be reflected both in the classification results between imputations and the posterior probabilities of the assigned classes. If the weight of the particular feature for the classification result is small, then even a high uncertainty about the true value will make a small difference to the end result and vice versa. This will be automatically captured in this approach.

Here we will derive an uncertainty measure T corresponding to the total uncertainty that is caused by missing variables to the classification result of a classifier and by the inherent uncertainty of the classification model itself. The idea behind this approach is the same as in Rubin [4], only adapted to classification. The magnitude of T is dependent on within-variance $\bar{U}$, i.e. uncertainty of the classification result within a single imputation according to (1), and between-variance B which corresponds to the variation of classification result between different imputed data sets. When T is minimized, i.e. when there is no uncertainty about the missing values, uncertainty of the underlying classification model will remain.

The within-variance for each imputed data set, $\bar{U}_l$, can be calculated as defined in (1). Because values of $\bar{U}_l$ will fall in the range $[0, 1]$, the combined within-variance $\bar{U}$, calculated according to (3), will also fall in the range $[0, 1]$. The unbiased estimate of the between-variance B for continuous statistical estimands is the mean squared error which can be interpreted as a $0/1$ loss in the case of discrete classes as defined in (6). The possible values that B can get fall in the range $[0, 1]$.

$$B = \frac{1}{m-1} \sum_{l=1}^{m} \begin{cases} 1 & \text{if } \hat{Q}_l \neq \bar{Q} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The total classification uncertainty T is a combination of within-variance, between-variance, and simulation variance and it can be calculated according to (5) like in [4] in the case of statistical estimands. The total uncertainty T can get values in the range $[0, \frac{2m+1}{m}]$ and can easily be scaled to the range $[0, 1]$ so that its value will not depend on m and for more intuitive interpretation.

IV. Experiments

The presented uncertainty measure was tested in a practical experiment where more and more data was incrementally removed completely at random from two publicly available data sets, the Iris data set [9] and the Wine data set [13]. A Naive Bayes classifier was chosen as the classifier because it supports calculating posterior probabilities and it can natively handle missing values. Therefore comparison of the uncertainty measure in (1) and the proposed uncertainty measure with the same classifier was possible. In addition, a Support Vector Machine classifier was tested with the proposed algorithm. Gaussian kernel and default parameter values were used.

For the imputations, the R package *mice* [14] was used. Because the purpose of the experiments was not to maximize the classification performance but instead demonstrate the proposed uncertainty measure, a good default imputation method was used. In practice this meant using Random Forests as the imputation method with the number of trees set to 10, which is the default. Using Random Forests can accomondate interactions between variables and nonlinearities inherently [15]. The number of imputed data sets, m, was chosen to be 10 which should be a decent default value capturing much of the variation between imputations while remaining computationally feasible.

Data was removed from the data sets in 10 % increments until the described multiple imputation procedure was unable to impute all the values. The probability for each value to be removed was the same for all variables, i.e. the final missing values were missing completely at random. The data set was split into training (70 % of the samples) and test (30 % of the samples) data sets. The removal rate was same for both the training and test data sets and for each variable, including the class label for the training data set. The 1MI algorithm, described in Section II, was used to handle the multiple imputed data. The only difference was the use of the class label of the training samples as part of the imputations. This information is indeed available for most of the training data set samples and should therefore be used as part of the imputation model.

For each data set and each step of data removal, the total uncertainty T was calculated according to (5) for every sample in the test data set after multiple imputation. In addition, uncertainty of Naive Bayes classifications on the same incomplete data samples were calculated with (1) for comparison. All experiments were simulated 1000 times with a different random seed on each simulation.

V. Results

For each amount of missing values, classification performance was assessed as a function of uncertainty. The classification results were ordered by their uncertainty in an ascending order and the cumulative number of samples that were above a threshold accuracy value were counted. The results when 50 % of the data was missing in the Iris data set can be seen in Figure 5. For the multiple imputed data, the proposed uncertainty measure was used, and for Naive Bayes on incomplete data, the uncertainty measure defined in (1) was used.

In this particular case, using the Naive Bayes classifier directly on the incomplete data led to a classification accuracy of 79.51 % whereas with the multiple imputed data, the accuracies were 78.61 % and 80.26 % for Naive Bayes and SVM classifiers, respectively.

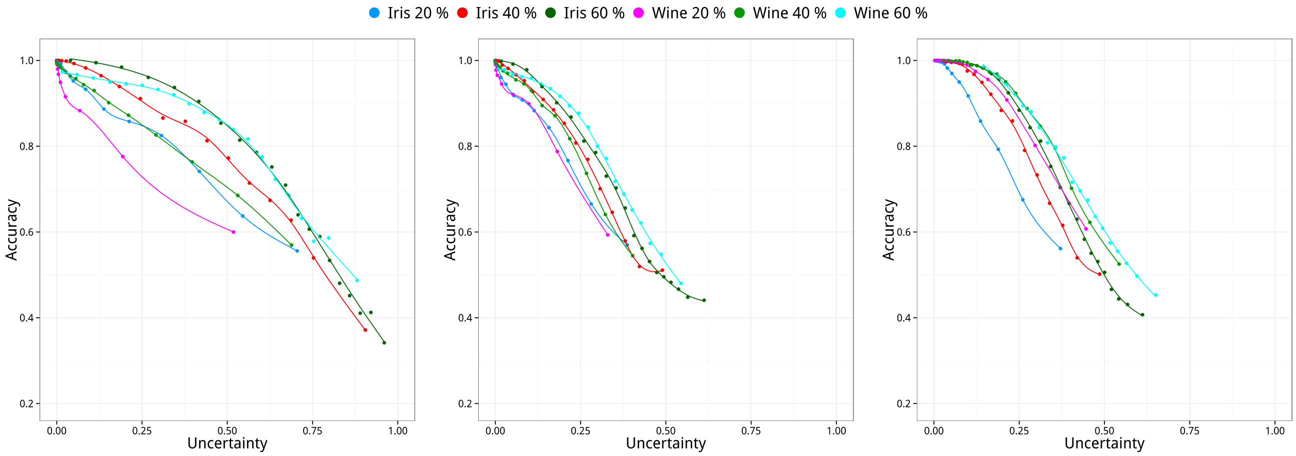
For a comparison of the proposed algorithm with directly using incomplete data, classification accuracy was plotted against uncertainty with both the Iris and the Wine data sets with different amounts of missing data. Naive Bayes classifier was used in both cases. The same procedure was repeated with Support Vector Machine classifier. The results are plotted in Figure 6.

For the same data sets and same amounts of missing values, average classification accuracies and average uncertainties were calculated. The results can be seen in Tables II and III for Naive Bayes classifier used directly on incomplete data as well for both Naive Bayes and Support Vector Machine classifiers using the proposed algorithm based on multiple imputation. Highest accuracy for each data set and each amount of missing values is in bold in the results.

The classification results were then sorted in an ascending order based on their uncertainty and the classification accuracies were calculated for both the lowest and highest uncertainty deciles of the samples. The results are presented in Tables IV and V again for the same data sets with different amounts of missing data and for all the same algorithms as above. For the lowest decile of uncertainties, i.e. the least uncertain of the samples, the highest classification accuracy is again in bold. For the highest decile, i.e. the most uncertain of the samples, the lowest accuracy is in bold in the results.

VI. Discussion

With complete data, both the Iris data set and the Wine data set are almost completely separable. As the amount of missing values increases, classification accuracy starts to decline but remains very reasonable even with as much as 60 % of the data missing. Classification performance is almost identical when



(a) Naive Bayes with incomplete data

(b) Naive Bayes with multiple imputed data

(c) SVM with multiple imputed data

Fig. 6. Classification accuracy as a function of uncertainty.

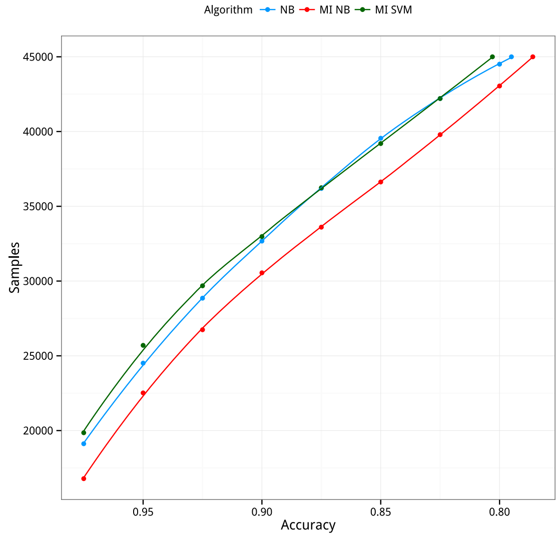


Fig. 5. Cumulative number of samples with the lowest uncertainty as a function of classification accuracy in the Iris data set when 50 % of the data was missing.

TABLE II. Classification accuracy and uncertainty of the Iris data set with Naive Bayes, Naive Bayes with multiple imputation, and SVM with multiple imputation.

	ACCURACY			AVG UNCERTAINTY		
MISSING	NB	MI NB	MI SVM	NB	MI NB	MI SVM
10 %	0.937	0.940	**0.948**	0.075	0.038	0.040
20 %	0.918	0.922	**0.934**	0.125	0.067	0.067
30 %	0.890	0.892	**0.909**	0.197	0.108	0.115
40 %	0.851	0.850	**0.869**	0.296	0.165	0.179
50 %	0.795	0.786	**0.803**	0.433	0.246	0.268
60 %	**0.708**	0.694	0.702	0.588	0.340	0.366
70 %	**0.568**	0.556	0.547	0.720	0.433	0.455

TABLE III. Classification accuracy and uncertainty of the Wine data set with Naive Bayes, Naive Bayes with multiple imputation, and SVM with multiple imputation.

	ACCURACY			AVG UNCERTAINTY		
MISSING	NB	MI NB	MI SVM	NB	MI NB	MI SVM
10 %	0.963	0.964	**0.971**	0.026	0.020	0.054
20 %	0.952	0.953	**0.961**	0.041	0.034	0.086
30 %	0.936	0.938	**0.943**	0.072	0.057	0.131
40 %	**0.915**	0.911	0.907	0.131	0.093	0.198
50 %	**0.877**	0.871	0.851	0.246	0.152	0.284
60 %	**0.818**	0.806	0.738	0.444	0.258	0.391
70 %	**0.667**	0.665	0.562	0.660	0.398	0.475

using the incomplete data directly and using multiple imputed data with the Naive Bayes classifier. Support Vector Machine seems to work marginally better when the amount of missing data is relatively small, even without tuning the classifier parameters, but falls behind in accuracy with these two data sets when the amount of missing values starts to increase more. This decline is more pronounced with the Wine data set and it is accompanied by increased average uncertainty compared to Naive Bayes classifier with the proposed algorithm. With the Iris data set the average uncertainties remain almost identical with increasing number of missing data and there is no dip in performance for either of the classifiers.

The two algorithms were compared by sorting the classification results by the uncertainty value and counting the number of most certain samples reaching a certain classification rate (Figure 5). A local polynomial regression smoother was then fitted to the data to make it easier to compare the results visually. The slope of the smoother for both algorithms is similar as it is also for both classifiers with the new algorithm. Only difference is in the height of the curve which is comparable to the overall accuracy achieved by that particular classifier-algorithm combination.

Because the two uncertainty measures differ slightly, they do not equally occupy the same value range and they are therefore not directly comparable value for value. With both uncertainty measures, however, it is very clear that low uncertainty values mean high classification accuracy and vice versa as is evident looking at Tables IV and V, which is exactly the feature we are interested in an uncertainty measure. A

TABLE IV. Classification accuracy of the lowest and highest uncertainty deciles of the Iris data set with Naive Bayes, Naive Bayes with multiple imputation, and SVM with multiple imputation.

	LEAST UNCERTAIN			MOST UNCERTAIN		
MISSING	**NB**	**MI NB**	**MI SVM**	**NB**	**MI NB**	**MI SVM**
10 %	**1.00**	**1.00**	**1.00**	**0.648**	0.675	0.650
20 %	**1.00**	**1.00**	0.9998	**0.597**	0.618	0.618
30 %	**1.00**	0.9998	0.9998	**0.541**	0.569	0.568
40 %	**1.00**	**1.00**	0.998	**0.456**	0.515	0.521
50 %	**1.00**	0.9996	0.996	**0.368**	0.487	0.473
60 %	**0.997**	0.995	0.987	**0.377**	0.444	0.419
70 %	**0.945**	0.936	0.873	**0.354**	0.383	0.370

TABLE V. Classification accuracy of the lowest and highest uncertainty deciles of the Wine data set with Naive Bayes, Naive Bayes with multiple imputation, and SVM with multiple imputation.

	LEAST UNCERTAIN			MOST UNCERTAIN		
MISSING	**NB**	**MI NB**	**MI SVM**	**NB**	**MI NB**	**MI SVM**
10 %	**1.00**	**1.00**	**1.00**	**0.735**	0.756	0.757
20 %	0.9998	**1.00**	**1.00**	**0.688**	0.691	0.705
30 %	0.9996	0.9996	**0.9998**	0.655	0.650	**0.642**
40 %	0.996	0.998	**0.9994**	0.627	0.593	**0.574**
50 %	0.988	0.993	**0.997**	0.564	0.546	**0.534**
60 %	0.970	**0.979**	0.975	0.537	0.514	**0.475**
70 %	0.927	**0.930**	0.769	0.444	0.425	**0.404**

very high accuracy is achieved in the least uncertain samples even as the average accuracy gets lower when more and more data is missing. This is more visually clear from Figure 6. As can be seen from the figures, the relationship between uncertainty and accuracy is not linear and it is not completely consistent between different data sets or different amounts of missing data within the same data set. When looking at the uncertainty values that each of the algorithms produce at a certain accuracy level with different amounts of missing data and on different data sets, our novel algorithm seems to outperform the benchmark algorithm. For example at 80 % accuracy, using incomplete data directly leads to uncertainty values that cover approximately 43 % of the total range of uncertainties in the data set. With our novel algorithm, this variation is cut into half when the same classifier is used and is around 26 % with the novel algorithm using the SVM classifier. This span of the uncertainty values is clearly lower with the novel algorithm suggesting it being more consistent across data sets and different amounts of missing data.

The proposed algorithm is computationally more complex than directly using incomplete data with a classifier that can handle missing values. The complexity comes from two sources. First, the imputation process adds complexity as the imputation models need to be constructed and the imputation is repeated $2m$ times, m times for the training data and m times for the test data. Second, the classification task is more complex as the amount of training data is higher when stacked training data is used and the classification is repeated m times.

VII. Conclusion

The main finding in our study is that the ability of the novel algorithm to estimate the uncertainty of a classification result works as well if not better as the benchmark algorithm. It also allows any classifier to be used even if the classifier does not support data modelling with missing values. Only requirement is that it supports estimating posterior probabilities. The new algorithm is computationally more complex, but with the cost of computational resources decreasing, this is a small price to pay for the upsides in tasks where uncertainty adds value to the bare classification results.

Acknowledgement

The authors would like to thank Infotech Oulu, Jenny and Antti Wihuri foundation, and Tauno Tönning foundation for financial support.

References

[1] S. van Buuren, *Flexible Imputation of Missing Data (Chapman & Hall/CRC Interdisciplinary Statistics)*. Chapman and Hall/CRC, 2012.

[2] E. Cernadas and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research*, vol. 15, pp. 3133–3181, 2014.

[3] M. Aste, M. Boninsegna, A. Freno, and E. Trentin, "Techniques for dealing with incomplete data: a tutorial and survey," *Pattern Analysis and Applications*, vol. 18, no. 1, pp. 1–29, 2014.

[4] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. Wiley, 1987.

[5] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, Sep. 2009.

[6] J. Luengo, S. García, and F. Herrera, *On the choice of the best imputation methods for missing values considering three groups of classification methods*. Springer-Verlag New York, Inc., Jun. 2011, vol. 32, no. 1.

[7] B. Twala, "An Empirical Comparison of Techniques for Handling Incomplete Data Using Decision Trees," *Applied Artificial Intelligence*, vol. 23, no. 5, pp. 373–405, 2009.

[8] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of Gaussians for Distance Estimation with Missing Data," *Neurocomputing*, vol. 131, pp. 32–42, May 2014.

[9] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[10] L. A. Belanche, V. Kobayashi, and T. Aluja, "Handling missing values in kernel methods with application to microbiology data," *Neurocomputing*, vol. 141, pp. 110–116, Oct. 2014.

[11] L. M. Gonçalves, C. C. Fonte, E. N. Júlio, and M. Caetano, "A method to incorporate uncertainty in the classification of remote sensing images," *International Journal of Remote Sensing*, vol. 30, no. 20, pp. 5489–5503, Sep. 2009.

[12] T. C. Urdan, *Statistics in Plain English*, 3rd ed. Taylor & Francis, 2010.

[13] M. Forina, "PARVUS - An extendible Package for Data Exploration, Classification and Correlation," 1991.

[14] S. van Buuren and K. Groothuis-Oudshoorn, "mice : Multivariate Imputation by Chained Equations in R," *Journal of Statistical Software*, vol. 45, no. 3, 2011.

[15] A. D. Shah, J. W. Bartlett, J. Carpenter, O. Nicholas, and H. Hemingway, "Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study," *American Journal of Epidemiology*, vol. 179, no. 6, pp. 764–774, 2014.