A Wrapper Feature Selection Approach to Classification with Missing Data

Cao Truong Tran, Mengjie Zhang, Peter Andreae, and Bing Xue

School of Engineering and Computer Science,
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{cao.truong.tran,mengjie.zhang,peter.andreae,bing.xue}@ecs.vuw.ac.nz

Abstract. Many industrial and real-world datasets suffer from an unavoidable problem of missing values. The problem of missing data has been addressed extensively in the statistical analysis literature, and also, but to a lesser extent in the classification literature. The ability to deal with missing data is an essential requirement for classification because inadequate treatment of missing data may lead to large errors on classification. Feature selection has been successfully used to improve classification, but it has been applied mainly to complete data. This paper develops a wrapper feature selection approach to classification with missing data and investigates the impact of this approach. Empirical results on 10 datasets with missing values using C4.5 for an evaluation and particle swarm optimisation as a search technique in feature selection show that a wrapper feature selection for missing data not only can help to improve accuracy of the classifier, but also can help to reduce the complexity of the learned classification model.

Keywords: Missing data, feature selection, classification, C4.5, particle swarm optimisation

1 Introduction

Classification is one of the most important tasks in machine learning and data mining [14]. The input space plays a crucial role in most classification algorithms. Many classification algorithms such as decision trees and rule-based classifiers are not able to achieve adequate predictive performance when the input contains many features that are not necessary for predicting the desired output. Feature selection which finds a sufficient feature subset from original features is one approach to the problem [18].

Missing values are a common problem in many datasets [21], [27]. For example, 45% of the datasets in the UCI repository [1], which is one of the most popular data repository for benchmarking machine learning tasks, contain missing values [11]. Missing data causes a number of serious problems [2]. One of the most serious problems is non-applicability of data analysis methods because the majority of existing data analysis methods require complete data. Therefore, these data analysis methods cannot work directly with original data containing missing values. Furthermore, missing data may lead to biased results because of differences between missing and complete data.

In statistical analysis field, the problem of missing data has been tackled extensively [12], [21], [26], [27] and also, but with less effort, in the classification literature. There are two main approaches to classification with missing data. One approach is to use imputation methods that fill missing values by plausible values before using classifiers. The other approach is to use classifiers that are able to classify missing data. Although the two approaches are able to handle missing data, they often result in large errors on classification [10]. Therefore, further approaches to improving classification accuracy of missing data should be investigated.

Feature selection is the process of finding a subset of the original features that is sufficient to solve the classification problem. Feature selection has been widely used to improve classification for complete data [7], [18], [19]. In feature selection, two main ways of evaluating feature subsets are the wrapper approach and the filter approach (nonwrapper) [18]. The wrapper approach uses the performance of a classifier to evaluate feature subsets. In contrast, instead of using a particular classifier, the filter approach uses a measure such as information gain (IG) and information gain ratio (IGR) [23] to evaluate the feature subset. In [9], a filter approach to feature selection for missing data was proposed, and the experimental results showed that the filter feature selection method for missing data can increase the precision of the prediction models. However, a wrapper approach to feature selection for missing data has not been investigated. Therefore, whether a wrapper approach to feature selection can improve classification with missing data is still an open issue.

1.1 Research Goals

The overall goal of this paper is to develop a wrapper approach to feature selection on classification with missing data and investigate the impact of this approach. To achieve this goal, three different ways are used to classify missing data. Firstly, missing data is classified by using a classifier that is able to classify directly missing data . Secondly, missing values are filled with plausible values by using imputation methods before using a classifier. Thirdly, a wrapper feature selection method is used to select a feature subset from missing data before using classifiers. Results from the three processes are compared to answer the following questions:

1. Whether feature selection for missing data can improve classification accuracy and achieve dimensionality reduction compared to without using feature selection; and
2. Whether feature selection for missing data can improve classification accuracy and achieve dimensionality reduction compared to using imputation methods.

1.2 Organisation

The rest of the paper is organised as follows. Section 2 discusses related work. Section 3 outlines the method and experiment design. Section 4 presents empirical results and analysis. Section 5 draws conclusions and presents future work.

2 Related Work

This section discusses related work including classification with missing data, imputation methods, feature selection, C4.5 for classification with missing data and Particle Swarm Optimisation-based feature selection.

2.1 Classification with Missing Data

There are three major approaches to classification with missing data including deletion approach, imputation approach and machine learning approach [11].

Deletion approach eliminates all instances containing missing values before using classifiers. This approach provides complete data that can be classified by any classifiers, but instances containing missing values are not included in the classification process [11].

Imputation approach uses imputation methods that fill missing values with plausible values before using classifiers. By using imputation methods, missing data is transferred to complete data that can be then classified by any classifiers. Moreover, most imputation methods help improve classification accuracy when compared to classification without using imputation methods. Therefore, using imputation methods is a major approach to classification with missing data [10].

Machine learning approach builds classifiers that are able to classify directly missing data without using imputation methods. For example, C4.5 [25], CART [8] and CN2 [5] can deal with missing values in any feature for both training set and test set.

2.2 Imputation Methods

The purpose of imputation methods is to fill missing values with plausible values. By using imputation methods, missing data is transformed into complete data that can be then analysed by any data analysis methods. Therefore, using imputation methods is a popular approach to handling missing data [21], [26], [27]. This section presents three popular imputation methods which are used in this paper.

Mean imputation fills missing values in each feature with the average of complete values in the same feature. This method maintains the mean of each feature, but it under-represents the variability in the data because all missing values in each feature are filled with the same value [11].

KNN-based imputation finds the K most similar instances of each instance containing missing values, and then fills missing values of the instance with the average of the values in the K most similar instances. KNN-based imputation is often better than mean imputation [3]. However, this method is often computationally intensive due to having to search through all instances to find the K most similar instances of each instance containing missing values [11].

Expectation Maximization-based imputation uses the Expectation Maximization(EM) algorithm to estimate a maximum likelihood variance-covariance matrix and vector of means that are then used to impute missing values [21], [27]. This method is an iterative procedure that includes two main steps at each iteration: an E-step and an M-step. The E-step is used to estimate the means, variances and covariances from complete values and the current best guess of missing values. The M-step is used to estimate new regression equations for each attribute predicted by all others, after that the new regression equations are then used to update the best guess for missing values during the E-step of next iteration. EM-based imputation has been proven to be one of the most powerful imputation methods [12].

2.3 Feature Selection

Feature selection is the process of finding a subset of the original features that is sufficient to solve the classification problem. Feature selection can remove redundant features; hence, it helps to improve classification accuracy. Furthermore, feature selection results in dimensionality reduction, so it makes the learning and execution processes faster. Moreover, models constructed using a smaller set of selected features are often easier to interpret [19].

The two main components of a feature selection method are a search technique and an evaluation criterion. The search procedure is used to generate candidate feature subsets that are then examined by the evaluation procedure to determine their goodness. The quality of the final selected features depends strongly on both the search technique and the evaluation criterion [7].

Many search techniques have been applied to feature selection including conventional methods and evolutionary techniques. For example, sequential forward selection and sequential backward selection are two traditional search techniques used in feature selection [15]. Recently, evolutionary computation techniques such as Genetic Algorithms and Particle Swarm Optimisation (PSO) have been applied to feature selection [4], [20], [24], [28], [30].

The two main ways of evaluating selected features are the wrapper approach and the filter approach (nonwrapper) [7]. In the wrapper approach, the performance of a classifier is used to evaluate the subset and hence guide the search.

Because every evaluation requires training a classifier and then testing its performance, the search process using a wrapper approach is typically computationally intensive. In the filter approach, instead of using a particular classifier in the evaluation function, the selected features are evaluated by a measure such as information gain and information gain ratio [23]. Because no classification algorithm is involved in the evaluation of selected features, the search process of the filter approach is expected to be more efficient and the results are expected to be more general. However, wrapper approaches often achieve better classification performance than filter approaches [18].

Feature selection has been mainly applied to complete data. A filter approach to feature selection for regression with missing data was proposed in [9], where mutual information was modified to evaluate feature subsets containing missing values. The experimental results showed that the filter approach to feature selection for missing data help improve the performance of the prediction models. However, a wrapper approach to feature selection for missing data has not been investigated.

2.4 C4.5 for Classification with Missing Data

In a wrapper feature selection algorithm for missing data, a classifier that is able to classify missing data is required to evaluate feature subsets. In this paper, the C4.5 algorithm that can classify directly missing data is used to evaluate feature subsets [25].

C4.5 can handle missing values in any feature for both training set and test set. C4.5 uses a probabilistic approach to handling missing values in both the training set and test set, but the way of handling missing values in the training stage is different from the testing stage. In the training stage, each value of each feature is assigned a weight: if a feature value is known, then the weight is assigned one; otherwise, the weight of any other values for that feature is the frequency of that values. In the testing stage, if a test case is unknown, from the current node, it finds all the available branches and decides the class label by using the most probable value [25].

2.5 PSO-based Feature Selection

Particle swarm optimisation (PSO) is a swarm intelligence algorithm proposed by Kennedy and Eberhart in 1995 [16], [17]. PSO is inspired by the movement of organisms such as a bird flocking or fish schooling. In order to optimize a problem, PSO builds a population of candidate solutions encoded as particles in the search space, and moves these particles around in the search space based on information of the particles' position and velocity. The movement of each particle is guided not only by its local best known position but also by the global best known position in the search space. When improved positions are discovered, these will be used to guide the movements of the swarm. This is expected to move the swarm toward the best solution. PSO does not require making assumptions about the problem being optimized and has ability to search very large spaces of candidate solutions. Therefore, PSO is able to be used for optimization problems that are partially noisy, irregular, change over time, etc. However, as like the majority of evolutionary computation algorithms, PSO does not ensure an optimal solution is ever found.

PSO has recently been applied to feature selection problems [29]. In PSO-based feature selection, PSO is used as a search technique to find feature subsets. If n is the total number of original features in the dataset, then the dimensionality of the search space is n. Each particle in the swarm is often a vector of n real numbers. The value of particle i in the d^{th} dimension, x_{id}, is usually in interval [0, 1]. To determine whether a feature will be selected or not, a threshold $0 < \theta < 1$ is required to compare with the real numbers in the position vector. If $x_{id} > \theta$, then the feature {d} will be selected; otherwise, feature {d} will be not selected.

Many PSO based feature selection algorithms have been proposed for both wrapper approaches and filter approaches. PSO has been proven to have the potential to address feature selection problems [4], [20], [28], [30]. However, the performance of PSO for feature selection on missing data has not been investigated.

3 Method and Experiment Design

This section shows detailed experiment design including the method, datasets, C4.5 algorithm, imputation methods and PSO parameter settings for feature selection.

3.1 The Method

The main objective of this study is to empirically evaluate the impact of a wrapper feature selection method on classification with missing data. To achieve this, three experimental setups are designed, as shown in Fig.1, Fig.2 and Fig.3, respectively. The Fig.1 shows classification with missing data by using a classifier that is able to classify missing data. The Fig.2 shows classification with missing data by using an imputation method before applying a classifier. The Fig.3 shows classification with missing data by using a feature selection algorithm before applying a classifier that is able to classify missing data.

In the three experimental setups, in case of complete data, firstly, missing values are introduced into complete data to generate missing data. Next, missing data is divided into training missing data and testing missing data. In the first setup, as shown in Fig.1, the training missing data is directly put into a classifier to build a classification model that is then used to classify testing missing data. In the second setup, as shown in Fig.2, both training missing data and testing missing data are put into an imputation method to generate imputed training data and imputed testing data, and then, the imputed training data is put into a classifier to build a classification model that is then used to classify the imputed testing data. In the third setup, as shown in Fig.3, training missing data is used by a feature selection procedure to choose a suitable feature subset that is then used to build a data transformation. The data transformation is then used to transform the training missing data and the testing missing data into transformed training missing data and transformed testing missing data, respectively. The transformed training missing data is then put into a classifier to build a classification model that is then used to classify the transformed testing missing data.

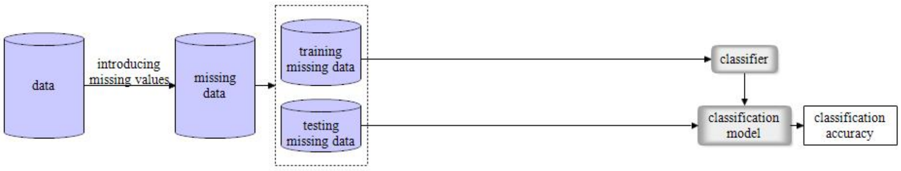


Fig. 1. Classification with missing data by using a classifier able to classify missing data

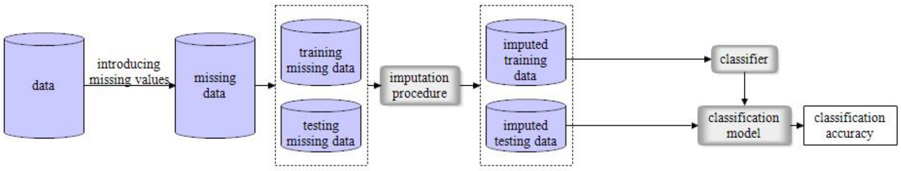


Fig. 2. Classification with missing data by using an imputation method before applying a classifier

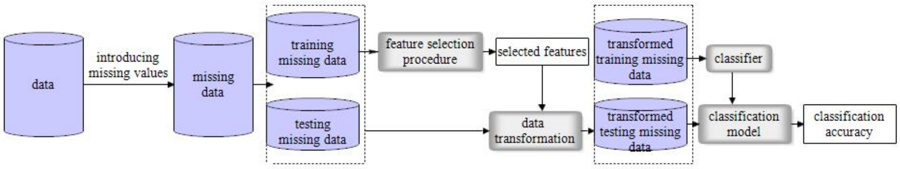


Fig. 3. Classification with missing data by using a feature selection method before using a classifier able to classify missing data.

3.2 Datasets

The experiments used 10 benchmark datasets selected from the UCI machine learning repository [1]. Table 1 summarises the main characteristics of each dataset including the number of instances, the number of features, the number of classes and the percentage of instances in the datasets which have at least one missing value.

Table 1. The datasets used in the experiments.

Dataset	#Instances	#Features	#Classes	Missing Inst (%)
Cleveland	303	13	5	1.98
Hepatitis	155	19	2	48.39
Marketing	8993	13	9	23.54
Ozone	2536	73	2	27.12
Wisconsin	699	9	2	2.29
Climate	540	20	2	0
Ionosphere	351	34	2	0
Parkinsons	197	23	2	0
Robot	463	90	5	0
Sonar	208	60	2	0

The first five datasets have missing values in a “natural”way. There is not any information related to the randomness of missing values in the datasets, so we assume they are distributed in a missing at random (MAR) way [21], [22].

To test the performance of the wrapper feature selection method for datasets with different levels of missing values, the missing completely at random mechanism (MCAR) [21] was used to introduce missing values into the last five complete datasets. Six levels of missing values: 5%, 10%, 20%, 30%, 40% and 50% were used to put into the datasets. For each dataset and each level of missing values, perform 30 times: choose randomly 50% features of the dataset, and then put the level of missing values into the chosen features. Therefore, for each level of missing values on one dataset, 30 artificial missing datasets were generated. Hence, from one complete dataset, 180 (30 × 6) artificial missing datasets were generated and a total of 900 (180 × 5) artificial missing datasets were used in the experiments.

Since none of the datasets in the experiments comes with a specific test set and the number of examples in some datasets is relatively small, a ten-fold cross-validation approach was used to evaluate the performance of induced classification models. With the first five datasets containing natural missing values, a ten-fold cross-validation approach was performed 30 times for each dataset. With the last five datasets, for each level of missing values on one dataset, ten-fold cross-validation was performed on the 30 missing datasets. As a result, for each dataset in the first five datasets and each level of missing values on one dataset in the last five datasets, 300 couples of training set and testing set were generated.

3.3 Imputation Algorithms

The experiments used three imputation methods including mean imputation, KNN-based imputation, EM-based imputation. Mean imputation and KNN-based imputation were in-house implementations. For KNN-based imputation, the number of neighbors was set to 10. The experiments used WEKA’s [13] implementation for EM-based imputation by setting their parameters as the default values.

3.4 Classification Algorithm

The experiments used C4.5 that has ability to classify missing data. C4.5 was used to classify data and evaluate feature subsets in feature selection. The experiments used WEKA's [13] implementation for C4.5 by setting its parameters as the default values.

3.5 PSO Settings

The experiments used PSO as a search technique for feature selection. The parameters in the PSO based feature selection algorithm were selected according to common settings proposed by Clerc and Kennedy [6]. The detailed settings were shown as follows: $\omega = 0.729844$, $c_1 = c_2 = 1.49618$, population size was 50, and the maximum iteration was 100. The fully connected topology is used. The threshold θ was set 0.6 as suggested by [29] to determine whether a feature is selected or not. For each dataset in the first five datasets and each level of missing values on one dataset in the last five datasets had 300 couples of training set and test set, so PSO repeated 300 times on each dataset.

4 Results and Analysis

Table 2 and Table 3 present the average of classification accuracy along with standard deviation of the first five datasets and the last five datasets with six levels of missing values, respectively, by using C4.5 in different ways. With the first five datasets containing natural missing values, the average of classification accuracy were calculated on accuracy of 30 times performing ten-fold cross-validation on each dataset. With the last five datasets, for each dataset and each missing level, the averages of classification accuracy were calculated on accuracy of 30 generated missing datasets with the missing level.

Table 4 and Table 5 present the average of size of decision trees (the number of nodes in the trees) generated by using C4.5 in different ways of the first five datasets and the last five datasets with six levels of missing values, respectively.

In the four tables, C4.5 column indicates results from the first experimental setup in Fig.1; C4.5MI, C4.5KNNI and C4.5EMI columns indicate results from the second experimental setup in Fig.2 by using mean imputation, KNN-based imputation and EM-based imputation, respectively; C4.5FS column indicates results from the third experimental setup in Fig.3. In order to compare the classification performance of C4.5FS with other methods, t-tests at 95% confidence level have been conducted to compare the classification performance achieved by C4.5FS with all other methods. "T"columns in Table 2 and Table 3 indicate significant tests of the columns before them against C4.5FS, where "+"means C4.5FS was significantly more accurate, "="means not significantly different, and "-"means significantly less accurate.

Table 2. Classification accuracy comparison of C4.5FS with C4.5, C4.5MI, C4.5KNNI and C4.5EMI on datasets containing natural missing values. The T columns indicate significant tests of the columns before them against C4.5FS.

Dataset	**C4.5FS**	**C4.5**	**T**	**C4.5MI**	**T**	**C4.5KNNI**	**T**	**C4.5EMI**	**T**
Cleveland	**56.64**±1.37	55.07±1.88	+	53.89±1.31	+	54.08±1.53	+	53.77±1.71	+
Hepatitis	**79.62**±1.93	78.59±1.87	+	76.84±2.56	+	77.24±2.98	+	77.71±2.08	+
Marketing	**32.85**±0.46	30.80±0.41	+	29.99±0.38	+	30.0±0.39	+	29.98±0.40	+
Ozone	**96.55**±0.31	96.28±0.26	+	95.94±0.32	+	95.93±0.38	+	95.95±0.31	+
Wisconsin	94.62±0.55	94.73±0.46	=	94.47±0.47	=	**94.96**±0.48	-	**94.87**±0.48	-

Table 3. Classification accuracy comparison of C4.5FS with C4.5, C4.5MI, C4.5KNNI and C4.5EMI using several missing rates. The T columns indicate significant tests of the columns before them against C4.5FS.

Dataset	**Missing rate (%)**	**C4.5FS**	**C4.5**	**T**	**C4.5MI**	**T**	**C4.5KNNI**	**T**	**C4.5EMI**	**T**
Climate	5	**91.34**±0.66	90.07±0.93	+	89.74±0.98	+	89.86±0.87	+	90.20±0.98	+
	10	**91.28**±0.80	90.39±0.95	+	89.42±0.97	+	89.86±1.13	+	89.63±1.30	+
	20	**91.23**±0.55	90.39±1.10	+	89.20±1.19	+	89.46±1.12	+	89.35±1.21	+
	30	**91.38**±0.63	90.94±1.09	+	89.30±1.20	+	89.30±1.15	+	89.29±0.95	+
	40	**91.41**±0.52	91.22±0.75	=	89.10±1.07	+	88.95±1.05	+	89.20±0.92	+
	50	**91.48**±0.13	91.08±0.91	+	89.30±1.30	+	89.23±1.92	+	89.43*1.20	+
Ionosphere	5	**90.87**±1.17	90.25±1.58	+	89.68±0.88	+	89.05±1.06	+	89.29±1.07	+
	10	**90.25**±1.67	89.36±1.70	+	89.08±1.50	+	88.47±1.52	+	89.49±1.35	=
	20	**90.30**±1.39	89.50±1.39	+	89.03±1.66	+	87.54±2.46	+	89.12±1.55	+
	30	**89.46**±1.62	**89.33**±1.67	=	88.64±1.62	+	88.18±2.23	+	88.10±1.70	+
	40	**88.60**±1.99	**88.59**±2.44	=	87.44±2.65	+	87.43±2.70	+	**88.40**±2.19	=
	50	**89.03**±2.24	**88.54**±2.64	=	86.36±2.60	+	86.36±2.60	+	87.56±2.30	+
Parkinsons	5	**87.23**±2.24	85.84±1.98	+	84.51±2.70	+	84.09±2.31	+	84.32±2.39	+
	10	**86.89**±1.72	85.05±2.34	+	84.36±2.41	+	84.10±2.62	+	84.52±2.43	+
	20	**86.51**±2.10	85.47±2.04	+	84.11±2.64	+	84.09±2.50	+	83.86±2.26	+
	30	**86.79**±1.87	84.96±2.50	+	83.90±1.91	+	83.28±2.35	+	83.16±2.03	+
	40	**86.66**±2.11	85.42±2.11	+	83.07±2.54	+	83.13±2.70	+	83.77±2.93	+
	50	**86.68**±2.08	85.03±2.09	+	83.19±3.41	+	83.06±3.41	+	83.15±2.00	+
Robot	5	**36.21**±1.91	32.72±2.16	+	31.97±1.91	+	31.82±2.11	+	32.53±1.93	+
	10	**35.12**±2.11	33.10±2.11	+	32.09±1.63	+	32.36±1.81	+	32.24±1.95	+
	20	**35.87**±1.75	32.54±1.96	+	33.54±2.01	+	33.54±2.02	+	33.39±2.08	+
	30	**35.44**±1.92	33.67±2.08	+	34.14±2.19	+	34.14±2.19	+	33.65±1.92	+
	40	**36.69**±2.61	35.18±2.01	+	34.60±2.04	+	34.60±2.04	+	**35.93**±1.90	=
	50	**38.39**±2.13	36.60±1.63	+	33.82±2.28	+	33.82±2.28	+	35.66±2.49	+
Sonar	5	**74.97**±3.04	72.96±2.63	+	72.65±3.00	+	**74.15**±2.72	=	72.68±2.77	+
	10	**74.11**±3.20	72.60±3.15	+	72.20±2.78	+	**72.79**±2.93	=	72.19±2.66	+
	20	**73.94**±3.48	73.94±3.34	+	71.58±2.82	+	71.44±2.77	+	**72.56**±2.76	=
	30	**72.23**±3.24	**72.74**±2.43	=	**70.94**±2.71	=	**70.94**±2.71	=	**71.22**±3.19	=
	40	**73.20**±4.17	**72.49**±3.85	=	69.31±3.82	+	69.31±3.82	+	**71.70**±2.72	=
	50	**73.71**±3.58	**72.85**±3.01	=	68.25±3.25	+	68.25±3.25	+	70.23±3.55	+

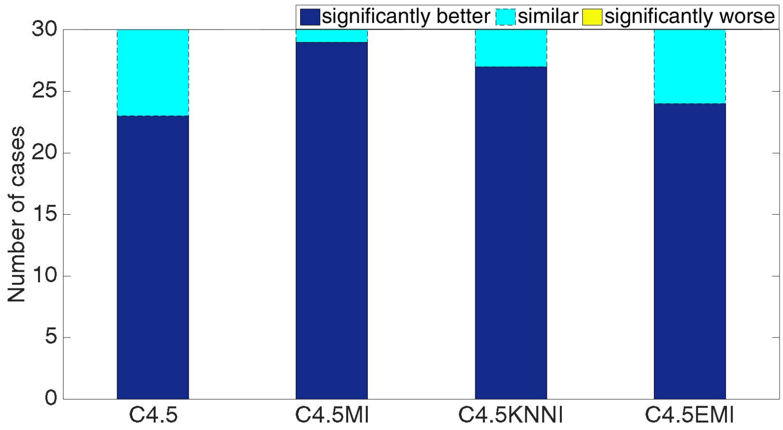


Fig. 4. Comparison of C4.5FS with C4.5, C4.5MI, C4.5KNNI and C4.5EMI

4.1 Classification Performance

It is clear from Table 2 that with the first five datasets containing natural missing values, C4.5FS achieves significantly better classification performance than other methods on the first four datasets, similar classification performance to C4.5 and C4.5MI on Wisconsin dataset and significantly worse classification performance to C4.5KNNI and C4.5EMI on Wisconsin dataset.

Fig.4 summarises the results from Table 3. It is clear from Fig.4 that with artificial missing datasets, C4.5FS often achieves significantly better or at least similar classification accuracy to the other methods. C4.5FS has more times achieving significantly better than C4.5MI and followed by C4.5KNNI, C4.5EMI and C4.5.

In summary, with both natural and artificial missing datasets, in most cases, feature selection for missing data can help improve classification accuracy of C4.5.

4.2 Size of the Learned Models

Table 4. Tree size of C4.5FS, C4.5, C4.5MI, C4.5KNNI and C4.5EMI on datasets containing natural missing values

Dataset	**C4.5FS**	**C4.5**	**C4.5MI**	**C4.5KNNI**	**C4.5EMI**
Cleveland	**32.7**	79.0	81.8	81.6	81.9
Hepatitis	**10.3**	17.3	19.8	21.3	18.6
Marketing	**304.8**	1367.1	1720.3	1665.2	1717.4
Ozone	**13.6**	24.8	29.4	30.7	30.1
Wisconsin	**15.8**	22.8	24.0	22.3	22.4

Table 5. Tree size of C4.5FS, C4.5, C4.5MI, C4.5KNNI and C4.5EMI with several missing rates

Dataset	**Missing amount (%)**	**C4.5FS**	**C4.5**	**C4.5MI**	**C4.5KNNI**	**C4.5EMI**
Climate	5	**10.4**	23.8	25.4	25.7	24.8
	10	**9.9**	21.1	27.4	26.5	25.9
	20	**7.2**	14.5	26.5	24.8	23.9
	30	**5.6**	10.2	28.0	25.3	22.7
	40	**4.8**	8.7	28.6	27.7	22.3
	50	**3.8**	6.5	25.7	25.0	22.0
Ionosphere	5	**17.5**	24.3	26.0	26.0	25.9
	10	**17.4**	24.2	25.4	25.6	25.3
	20	**17.3**	23.4	25.8	25.9	25.1
	30	**17.7**	22.8	26.8	26.5	25.6
	40	**18.5**	22.5	28.3	28.2	25.5
	50	**18.2**	20.7	27.7	27.7	25.6
Parkinsons	5	**15.0**	17.8	19.0	18.7	18.8
	10	**15.5**	17.9	18.7	19.2	18.5
	20	**15.4**	17.8	19.9	19.5	18.7
	30	**15.0**	17.0	19.6	19.5	18.8
	40	**13.9**	15.8	19.7	19.6	18.3
	50	**13.3**	14.6	19.3	19.2	18.7
Robot	5	**63.6**	71.3	118.4	106.6	100.9
	10	**69.4**	76.0	133.9	133.9	120.7
	20	**73.1**	86.4	131.7	131.7	129.2
	30	**74.6**	85.4	129.2	129.2	126.8
	40	**70.8**	79.4	128.4	128.4	125.0
	50	**63.7**	73.1	129.2	129.2	121.3
Sonar	5	**25.1**	27.7	28.0	27.5	27.9
	10	**25.4**	28.1	28.7	28.3	27.9
	20	**24.8**	28.4	29.5	29.5	27.6
	30	**23.2**	27.5	30.3	30.3	28.2
	40	**22.4**	26.7	30.6	30.7	28.7
	50	**22.7**	26.3	31.8	32.0	29.2

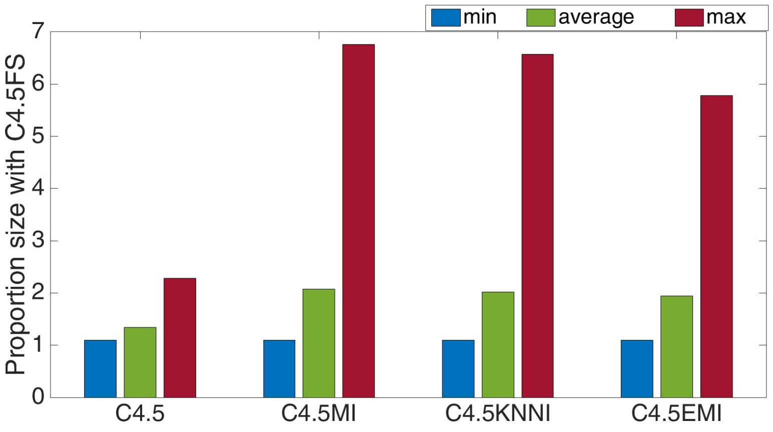


Fig. 5. Ratio tree size of C4.5, C4.5MI, C4.5KNNI and C4.5EMI with C4.5FS

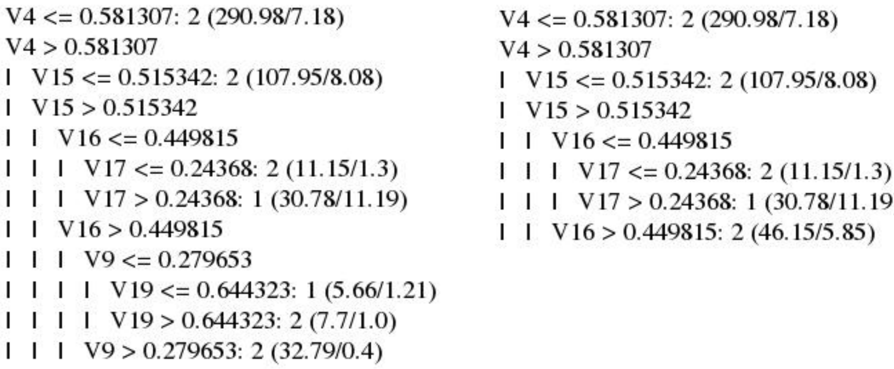
According to Table 4, with the first five datasets containing natural missing values, in all cases, C4.5FS generates smaller decision trees than other methods. For example, in Marketing dataset, sizes of decision trees generated by C4.5FS are nearly one fifth the sizes of decision trees generated by C4.5 and more than one fifth of sizes of decision trees generated by using imputation methods before using C4.5.

Fig.5 shows minimum, average and maximum of ratio of tree size of C4.5, C4.5MI, C4.5KNNI and C4.5EMI with C4.5FS from Table 5. The minimum of ratio of tree sizes of the other methods with C4.5FS show that C4.5FS generates smaller trees than other methods. On average, C4.5 generates about 30% bigger than those generated by C4.5FS, and the other three methods generate trees over twice bigger than those of C4.5FS. Especially, the maximum of ratio of tree sizes of the other methods with C4.5FS shows that sizes of decision trees generated by using imputation methods before using classifier in some cases are dramatically bigger than C4.5FS. The main reason is likely that imputation methods often generate further values for missing features; therefore, if the missing features are chosen to build decision trees, the further values make decision trees bigger.

In summary, with both natural and artificial missing data, in all cases, feature selection for missing data can help reduce complexity of the classification model generated by C4.5, especially compared to using imputation methods before using C4.5.

4.3 Analysis

To give a better picture of how C4.5FS can achieve better classification and smaller trees than the other methods, we looked carefully at the trees generated by C4.5 and C4.5FS on Climate dataset which has 20 features {V1,..,V20}. Climate dataset was chosen since the trees generated on the Climate dataset are not too big to analyse. Figs 6 and 7 show two typical pattern trees we observed.

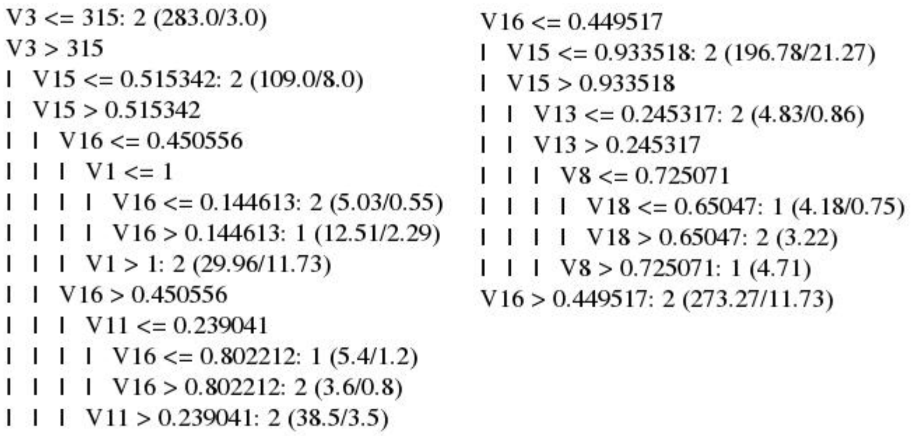


```
V4 <= 0.581307: 2 (290.98/7.18)
V4 > 0.581307
|   V15 <= 0.515342: 2 (107.95/8.08)
|   V15 > 0.515342
|   |   V16 <= 0.449815
|   |   |   V17 <= 0.24368: 2 (11.15/1.3)
|   |   |   V17 > 0.24368: 1 (30.78/11.19)
|   |   V16 > 0.449815
|   |   |   V9 <= 0.279653
|   |   |   |   V19 <= 0.644323: 1 (5.66/1.21)
|   |   |   |   V19 > 0.644323: 2 (7.7/1.0)
|   |   |   V9 > 0.279653: 2 (32.79/0.4)
```

```
V4 <= 0.581307: 2 (290.98/7.18)
V4 > 0.581307
|   V15 <= 0.515342: 2 (107.95/8.08)
|   V15 > 0.515342
|   |   V16 <= 0.449815
|   |   |   V17 <= 0.24368: 2 (11.15/1.3)
|   |   |   V17 > 0.24368: 1 (30.78/11.19)
|   |   V16 > 0.449815: 2 (46.15/5.85)
```

Fig. 6. Left tree generated by C4.5 and right tree generated by C4.5FS on Climate dataset with 20% missing values in features {V2, V3, V4, V12, V13, V14, V15, V17, V19}

Fig. 6 shows trees generated by C4.5 and C4.5FS on Climate dataset with 20% missing values in 9 features {V2, V3, V4, V12, V13, V14, V15, V17, V19}. After applying feature selection on the dataset, only seven features {V1, V4, V11, V15, V16, V17, V20} were chosen. The C4.5FS tree achieved slightly higher classification accuracy compared to the C4.5 tree, with 90.95% and 89.91%, respectively. Both of them had the same features in the top part of the trees. However, in the bottom part, the C4.5 tree had additional features which were not present in the C4.5FS tree because these features had been removed in the feature selection procedure. As a result, the C4.5FS achieved both better classification accuracy and a smaller tree than the C4.5.



```
V3 <= 315: 2 (283.0/3.0)
V3 > 315
|   V15 <= 0.515342: 2 (109.0/8.0)
|   V15 > 0.515342
|   |   V16 <= 0.450556
|   |   |   V1 <= 1
|   |   |   |   V16 <= 0.144613: 2 (5.03/0.55)
|   |   |   |   V16 > 0.144613: 1 (12.51/2.29)
|   |   |   V1 > 1: 2 (29.96/11.73)
|   |   V16 > 0.450556
|   |   |   V11 <= 0.239041
|   |   |   |   V16 <= 0.802212: 1 (5.4/1.2)
|   |   |   |   V16 > 0.802212: 2 (3.6/0.8)
|   |   |   V11 > 0.239041: 2 (38.5/3.5)
```

```
V16 <= 0.449517
|   V15 <= 0.933518: 2 (196.78/21.27)
|   V15 > 0.933518
|   |   V13 <= 0.245317: 2 (4.83/0.86)
|   |   V13 > 0.245317
|   |   |   V8 <= 0.725071
|   |   |   |   V18 <= 0.65047: 1 (4.18/0.75)
|   |   |   |   V18 > 0.65047: 2 (3.22)
|   |   |   V8 > 0.725071: 1 (4.71)
V16 > 0.449517: 2 (273.27/11.73)
```

Fig. 7. Left tree generated by C4.5 and right tree generated by C4.5FS on Climate dataset with 20% missing values in features {V1, V4, V5, V7, V9, V10, V13, V14, V16, V19}

Fig. 7 shows trees generated by C4.5 and C4.5FS on Climate dataset with 20% missing values in 10 features {V1, V4, V5, V7, V9, V10, V13, V14, V16, V19}. After applying feature selection on the dataset, only seven features {V7, V8, V9, V13, V15, V16, V18} were chosen. In C4.5, when computing the information gain of a feature containing missing values, it computes the gain on the complete values and discounts it by the ratio of complete instances to all instances [25]. In other words, missing values discount information gain of missing features. Therefore, C4.5 biases towards choosing complete features to build decision trees, but the bias of choosing complete features to build decision trees is not always good. For example, on Fig.7, while the first node of C4.5 tree is a complete feature V3, the first node of C4.5FS tree is a missing feature V16. However, the C4.5FS tree achieved both better classification accuracy (91.3% vs 90.1%) and smaller tree than the C4.5 tree . A possible reason could be that by removing less suitable features such as V3, feature selection helps to counteract the C4.5's bias towards choosing complete features to build decision trees.

In summary, feature selection is able to choose relevant features and remove irrelevant features. Therefore, feature selection helps to build better classifier.

5 Conclusions and Future Work

This paper presents research which has attempted to find the effect of a wrapper feature selection approach to classification with missing data. To undertake the research, three different experimental setups were designed: classification with missing data by using a classifier that is able to classify missing data, classification with missing data by using an imputation method before applying a classifier, and classification with missing data by using feature selection before using a classifier that is able to classify with missing data. The results from the three setups were compared on 10 datasets (five dataset containing natural missing values and five datasets with six levels of artificial missing values), using C4.5 for an evaluation and PSO as a search technique for feature selection. The empirical results showed that a wrapper feature selection approach to classification with missing data can help to improve classification performance of C4.5 and reduce the complexity of the learned classifier.

The experiment in this paper used C4.5 as a classifier because it can handle missing data. There are some other classifiers that are able to classify missing data such as CART [8] and CN2 [5]. Future work could repeat this investigation with CART and CN2. This paper used a wrapper-based approach to feature selection on classification problems with missing data. In [9], a filter-based approach to feature selection was applied to regression problems with missing data. The future work could explore the effectiveness of a filter-based approach to feature selection on classification problems with missing data.

References

1. A. Asuncion and D. Newman. UCI machine learning repository, 2007.
2. J. Barnard and X.-L. Meng. Applications of multiple imputation in medical studies: from aids to nhanes. *Statistical Methods in Medical Research*, 8:17–36, 1999.
3. G. E. Batista and M. C. Monard. A study of K-Nearest Neighbour as an Imputation Method. *HIS*, 87:251–260, 2002.
4. L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang. Improved binary pso for feature selection using gene expression data. *Computational Biology and Chemistry*, 32:29–38, 2008.
5. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine learning*, 3:261–283, 1989.
6. M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6:58–73, 2002.
7. M. Dash and H. Liu. Feature selection for classification. *Intelligent data analysis*, 1:131–156, 1997.
8. G. De'ath and K. E. Fabricius. Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, 81:3178–3192, 2000.

9. G. Doquire and M. Verleysen. Feature selection with missing data using mutual information estimators. *Neurocomputing*, 90:3–11, 2012.
10. A. Farhangfar, L. Kurgan, and J. Dy. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41:3692–3705, 2008.
11. P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.
12. J. W. Graham. Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60:549–576, 2009.
13. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11:10–18, 2009.
14. J. Han, M. Kamber, and J. Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.
15. A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19:153–158, 1997.
16. J. Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. 2010.
17. J. Kennedy, J. F. Kennedy, and R. C. Eberhart. *Swarm intelligence*. Morgan Kaufmann, 2001.
18. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97:273–324, 1997.
19. D. Koller and M. Sahami. Toward optimal feature selection. 1996.
20. S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert systems with applications*, 35:1817–1824, 2008.
21. R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2014.
22. J. Luengo, S. García, and F. Herrera. A study on the use of imputation methods for experimentation with radial basis function network classifiers handling missing attribute values: the good synergy between rbfns and eventcovering method. *Neural Networks*, 23:406–418, 2010.
23. D. J. MacKay. *Information theory, inference, and learning algorithms*, volume 7. Citeseer, 2003.
24. I.-S. Oh, J.-S. Lee, and B.-R. Moon. Hybrid genetic algorithms for feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26:1424–1437, 2004.
25. J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
26. J. L. Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.
27. J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological methods*, 7:147, 2002.
28. X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen. Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, (4):459–471, 2007.
29. B. Xue. *Particle Swarm Optimisation for Feature Selection in Classification*. Victoria University of Wellington, 2014.
30. B. Xue, M. Zhang, and W. N. Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *Cybernetics, IEEE Transactions on*, 43:1656–1671, 2013.