Evol. Intel. (2016) 9:81–94
DOI 10.1007/s12065-016-0141-6

SPECIAL ISSUE

Improving performance for classification with incomplete data using wrapper-based feature selection

Cao Truong Tran[1] · Mengjie Zhang[1] · Peter Andreae[1] · Bing Xue[1]

Received: 29 April 2016 / Revised: 13 July 2016 / Accepted: 21 July 2016 / Published online: 9 August 2016

Abstract Missing values are an unavoidable problem of many real-world datasets. Inadequate treatment of missing values may result in large errors on classification; thus, dealing well with missing values is essential for classification. Feature selection has been well known for improving classification, but it has been seldom used for improving classification with incomplete datasets. Moreover, some classifiers such as C4.5 are able to directly classify incomplete datasets, but they often generate more complex classifiers with larger classification errors. The purpose of this paper is to propose a wrapper-based feature selection method to improve the ability of a classifier able to classify incomplete datasets. In order to achieve the purpose, the feature selection method evaluates feature subsets using a classifier able to classify incomplete datasets. Empirical results on 14 datasets using particle swarm optimisation for searching feature subsets and C4.5 for evaluating the feature subsets in the feature selection method show that the wrapper-based feature selection is not only able to improve classification accuracy of the classifier, but also able to reduce the size of trees generated by the classifier.

✉ Cao Truong Tran
cao.truong.tran@ecs.vuw.ac.nz

Mengjie Zhang
mengjie.zhang@ecs.vuw.ac.nz

Peter Andreae
peter.andreae@ecs.vuw.ac.nz

Bing Xue
bing.xue@ecs.vuw.ac.nz

[1] Evolutionary Computation Research Group, School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand

Keywords Missing data · Incomplete data · Missing values · Feature selection · Classification · C4.5 · Particle swarm optimisation

1 Introduction

Classification is a major research area in data mining. The input space is one of the most important aspects affecting classification accuracy. Two main problems of input space are missing data and redundant/irrelevant features [5, 20].

An incomplete dataset is a dataset containing some features which do not have values in some fields. Missing values are a common issue in many real-world datasets [16, 29, 35]. For example, in the UCI repository [1], which is a popular data collection for machine leaning methods, 45 % of the datasets suffer from the problem of missing values [15]. Causes of missing values are various. For instance, survey sheets in a social survey may lack some values because some respondents ignore to answer some questions in the survey; some results in an industrial experiment might be missing because of mechanical failures while gathering data; medical datasets often contain missing values since not all tests can be run on every patient [13].

Missing values cause a number of severe issues. One severe issue is non-applicability of data analysis methods. Although some data analysis methods can deal with missing values, many others require complete data. Therefore, these methods are not able to work directly with original data containing missing values. Moreover, missing values may cause biased results owing to differences between missing data and complete data [2].

The problem of missing values in datasets has been addressed extensively in statistical fields [16, 29, 34, 35].

However, the problem of missing values has been tackled with less effort in classification tasks [15]. There are two major approaches to addressing the problem of missing values in classification tasks. The first approach is to use imputation methods to fill missing values with plausible values before using classifiers. The second approach is to use classifiers such as C4.5 [33] which are able to directly classify incomplete datasets. Although the two approaches are able to cope with missing values to a certain level, they often lead to large classification errors [12]. Hence, further approaches to enhancing classification accuracy when faced with missing values should be investigated.

When the input space contains numerous redundant/irrelevant features, many classifiers such as decision tress cannot achieve adequate accuracy. Feature selection that chooses a sufficient feature subset from original features is a well known solution to the problem [17, 19, 25]. The purpose of feature selection is to eliminate redundant features and only keep important features, while retaining or improving accuracy of the classification tasks. Feature selection has been widely used for improving classification in complete datasets [17, 19, 25].

In feature selection, there are two main approaches to evaluating feature subsets: the filter approach and the wrapper approach. The filter approach uses measures such as information gain to evaluate the quality of feature subsets [30]. In contrast, a wrapper method builds a classifier to evaluate the quality of feature subsets. In recent work [11, 32], filter approaches based on mutual information have been expanded to evaluate feature subsets when datasets contain missing values. The experimental results show that a filter-based feature selection can help improve regression and classification tasks when faced with missing values. In [39], a wrapper-based feature selection method using particle swarm optimisation (PSO) for incomplete datasets was developed, and was able to improve classification accuracy and reduce the complexity of the learned classifier, but it still has limitation. Therefore, deeper research on the wrapper-based feature selection for incomplete datasets should be investigated.

1.1 Research goals

The goal of this paper is to expand the wrapper-based feature selection method for classification with incomplete datasets in [39] by running the experiment with more datasets to make stronger conclusion about the effectiveness and the complexity of the wrapper-based feature selection for classification with incomplete datasets. We analyse the role of threshold in PSO-based feature selection and analyse the computation time of the wrapper method using PSO for feature selection with incomplete datasets. Finally, we attempt to identify why the wrapper-based feature selection can improve classification with incomplete datasets. The experimental results are used to address the following objectives:

1. How the threshold value in PSO-based feature selection affects the classification accuracy and the size of the learned classifier.
2. Whether the proposed wrapper-based feature selection method for incomplete datasets is able to enhance classification accuracy compared with using a classifier able to classify incomplete datasets without using feature selection or using imputation methods before using a classifier.
3. Whether the proposed wrapper-based feature selection method for incomplete datasets is able to reduce the complexity of the learned classifier compared with using a classifier able to classify incomplete datasets without using feature selection or using imputation methods before using a classifier.
4. How expensive the proposed wrapper-based feature selection method is for classification with missing values.
5. Why the proposed wrapper-based feature selection method for incomplete datasets is able to improve classification accuracy and reduce the complexity of the learned classifier.

1.2 Organisation

The rest of this paper is organised as follows. Related work is outlined in Sect. 2. After that, the method and experiment design are presented in Sect. 3. Empirical results and analysis are then shown in Sect. 4. Finally, conclusions and future work are presented in Sect. 5.

2 Related work

2.1 Classification with incomplete datasets

There are four major ways to deal with classification with missing values including removal approach, imputation approach, model-based approach and machine-learning approach [15].

Removal approach removes all instances containing missing values before using classifiers. The advantage of this approach is to provide complete data that is able to be classified by any classifiers. However, instances containing missing values are not classified by the learn classifier; hence, this method can be only used in the training process and when a dataset contains a small amount of missing values [13].

Imputation approach utilises imputation methods to fill missing fields with plausible values before using classifiers. For example, mean imputation replaces missing fields with the average of complete values in the same feature. The main advantage of using imputation methods is to provide complete data for classification. Therefore, both complete and incomplete instances are participated in the classification process. Moreover, most imputation methods are able to improve classification accuracy compared to without using imputation methods. Hence, imputation methods is a main method to deal with classification with incomplete datasets [12].

Model-based approach builds the data distribution model from input data. After that, the data distribution model is combined with the Bayesian decision theory [4] to classify both complete and incomplete instances. Although this approach is able to classify both complete and incomplete instances, it needs to make assumptions about the joint distribution of all features in the model [15].

Machine learning approach builds classifiers that can directly classify incomplete datasets without using imputation methods. For example, C4.5 [33] is able to address missing values in both the training set and test set by using a probabilistic approach. Some other classifiers can deal with incomplete datasets including CART [10] and CN2 [7].

2.2 Imputation methods

Imputation methods aim at filling missing fields with plausible values. As a result, an incomplete dataset is transformed to a complete dataset which is then classified by using any classifiers. Hence, using imputation method is one of the main approaches to classification with incomplete datasets [15, 36–38]. This section shows three popular imputation methods used in this paper: mean imputation, KNN-based imputation and EM-based imputation.

Mean imputation is the simplest imputation method that fills missing fields with the average of complete values in the same feature. The main advantage of mean imputation is to maintain the mean of each feature. However, mean imputation under-represents the variability in the data since all missing values in each feature are replaced with the same value [15].

KNN-based imputation performs two steps to impute missing fields in an incomplete instance. The first step is to find the K most similar with the incomplete instance. After that, missing fields of the incomplete instance is filled by the average of the complete values of the K instances. KNN-based imputation is usually better than mean imputation [3]. Nevertheless, the computation time of KNN-imputation is often expensive because this method has to search through all instances to find the K most similar instances [15].

Expectation Maximization-based imputation is one of the most powerful imputation methods [29, 35]. This imputation uses the Expectation Maximization(EM) algorithm to calculate a maximum likelihood variance-covariance matrix and a mean vector which are then utilised to fill missing fields with plausible values. EM-based imputation is an iterative method including two main steps at each iteration: E-step and M-step. E-step is utilised to calculate variances, covariances and means from complete values and the current best values of missing fields. M-step is utilised to build new regression equations for each feature by using all other features, and then the new equations are used to update the best values for missing fields in the E-step of next iteration [16].

2.3 Feature selection

Feature selection is the process of searching for a feature subset from the original features which is adequate to perform the classification task. Feature selection is able to eliminate redundant features; thus, it assists to improve classification accuracy. Moreover, feature selection is able to reduce the complexity of the learned classifier; consequently, it makes the execution of the learned classifier faster. Furthermore, the classifiers constructed utilising a smaller number of features are often easier to interpret [26, 40].

A feature selection method consists of two main components: a search procedure and an evaluation measure. The search procedure is utilised to find feature subsets. After that, the evaluation measure is utilised to examine the goodness of the feature subsets. The quality of the feature selection method strongly depends on both the quality of the search procedure and the quality of the evaluation measure [9].

Search techniques in feature selection can be divided into conventional techniques and evolutionary techniques. For instance, two traditional search techniques are sequential backward selection and sequential forward selection [22]. Recently, evolutionary search techniques such as genetic algorithm, genetic programming and particle swarm optimisation (PSO) have been used widely to search for feature subsets in the feature selection method [6, 21, 28, 31, 42, 44].

Evaluation methods in feature selection can be divided into the wrapper methods and the filter methods [9]. A wrapper method uses a classifier to evaluate the feature subsets. A wrapper-based feature selection is often computationally intensive since every evaluation of feature subsets requires to train a classifier and then test its performance. In contrast, a filter method uses an evaluation measure such as information gain [30]. None of classifiers is participated in the evaluation of feature subsets; hence, feature subsets generated by using a filter-based feature selection is often more efficient and the results are often more general. However, wrapper-based feature selection methods usually achieve better classification than filter-based feature selection methods [25].

Feature selection has been mainly applied to complete data. A filter approach to feature selection for regression with incomplete datasets is proposed in [11], where nearest neighbors based mutual information estimator is extended to handle missing values. The experimental results on artificial as well as real-world datasets show that the method is able to select important features without the need for any imputation algorithm and help improve the performance of the prediction models. In [32], the mutual information criterion combined with rough sets is proposed to evaluate feature subsets in incomplete datasets. The experimental results on different datasets show that the proposed algorithm is more effective than existing algorithms for feature selection in incomplete datasets at most cases. In [39], a wrapper-based feature selection for incomplete datasets has been proved capable of improving classification accuracy and reducing the complexity of the learned classifier.

2.4 PSO for feature selection

Particle swarm optimisation (PSO) proposed by Kennedy and Eberhart in 1995 [23, 24] is a swarm intelligence algorithm. PSO is inspired by the movement of organisms such as a bird flocking. In order to optimise a problem, PSO makes a population of particles in the search space, and moves these particles around in the search space using the information of the particles' position and velocity. The movement of each particle uses both the personal best known position and the global best known position in the search space. When enhanced positions are found, this information will be utilised to guide the movements of the swarm toward the best solution. One advantage of PSO is that it does not require making assumptions about the problem being optimized. Furthermore, PSO is able to search very large spaces of candidate solutions. Consequently, PSO can be used to optimise problems which are partially noisy, irregular and change over time, etc. However, the same as other evolutionary algorithms, PSO cannot ensure to find an optimal solution.

PSO has been recently used as a search technique to find feature subsets form original features in feature selection problems [27, 40, 41, 43]. If the number of original features is n, then the search space dimensionality is n. Each particle in the swarm is usually presented by a vector of n real numbers. The value of the i^{th} particle in the d^{th} dimension, x_{id}, is often in an interval [0, 1]. In order to identify whether or not a feature will be chosen, the real value in the position vector is compared with a threshold $0 < \theta < 1$. If $x_{id} < \theta$, then the d^{th} feature will be not chosen; otherwise, the d^{th} feature will be chosen.

PSO has been used for both wrapper-based and filter-based feature selection. PSO has been proved capable of having the ability to deal well with feature selection problems [6, 21, 28, 42]. However, PSO-based feature selection for incomplete datasets has not been systematically investigated.

2.5 C4.5 for classification with missing data

In order to apply a wrapper-based feature selection for incomplete datasets, a classifier able to directly classify incomplete datasets is required to measure feature subsets. In this paper, the C4.5 algorithm [33] that can directly classify incomplete datasets is utilised to measure feature subsets.

C4.5 computes the information gain of an incomplete feature by computing the gain on the complete values and discounting it by the ratio of complete instances to all instances. C4.5 utilises a probabilistic approach to addressing missing values in both the training set and test set. C4.5 makes assumption that instances with the missing values are distributed probabilistically according to the relative frequency of known values. In the training process, each feature value is assigned a weight: the weight is assigned one if a feature value is known; otherwise, the weight of any other values for that feature is the frequency of those values. In the testing process, when a test feature is chosen, the cases with known values are divided into branches corresponding to these values. The cases with missing values are passed down all available branches, but with weight that corresponds to the relative frequency of the value assigned to a branch and it decides the class label by using the most probable value [33].

3 Method and experiment design

This section presents the detailed design of method and experiment including comparison method, datasets used in the experiment, C4.5 used as a classifier, imputation methods used to fill missing fields with plausible values and PSO parameter settings for searching feature subsets.

3.1 The method

This study is designed to empirically evaluate the effect of a wrapper-based feature selection method for incomplete datasets. In order to achieve this objective, a wrapper-based feature selection for incomplete datasets is proposed as shown in Fig. 1 and compared to other two common methods for dealing with classification with incomplete datasets as shown in Figs. 2 and 3. Figure 1 presents the proposed method for classification with incomplete datasets which uses a feature selection method to choose feature subsets from original features before using a classifier able to directly classify incomplete datasets. Figure 2 presents a common method for classification with incomplete datasets which uses a classifier able to directly classify incomplete datasets. Figure 3 presents another common method for classification with incomplete datasets which uses an imputation method to fill missing fields with plausible values before using a classifier.

In the three setups, an incomplete dataset is firstly divided into a training incomplete dataset and a testing incomplete dataset. In the proposed setup shown in Fig. 1, a feature selection method uses the training incomplete dataset as a training data to select a suitable feature subset. After that the feature subset is used to build a data transformation which is then used to transform the training incomplete dataset into training transformed incomplete dataset and the testing incomplete dataset into testing transformed incomplete dataset. The training transformed incomplete dataset is then used by a classifier to build a classification model which is then utilised to classify the testing transformed incomplete dataset. In the setup shown in Fig. 2, the training incomplete dataset is directly used by a classifier to build a classification model which is then utilised to classify the testing incomplete dataset. In the setup shown in Fig. 3, an imputation method is used to transfer the training incomplete dataset and the testing incomplete dataset into training imputed data and testing imputed data. After that the training imputed data is used by a classifier to build a classification model which is then utilised to classify the testing imputed data.

3.2 Datasets

Fourteen datasets, summarised in Table 1, are used in the experiments. These are taken from the UCI Repository of Machine Learning Databases [1]. Each dataset is presented in one row in Table 1 including the number of instances, the number of features, the number of classes, the proportion of instances containing at least one missing field and the proportion of features containing at least one missing field.

The first seven datasets suffer from missing values in a “natural”way. In the datasets, we do not know any information related to the randomness of missing values, so we make assumption that missing values in the datasets are distributed in a *missing at random* (MAR) way [29].

In order to test the performance of the proposed feature selection method with datasets containing different levels of missing values, the *missing completely at random* (MCAR) mechanism [29] was utilised to introduce missing values into the last seven complete datasets. Six different levels of missing values: 5, 10, 20, 30, 40 and 50 % were utilised to introduce missing values into the datasets. Although naturally incomplete datasets Ozone and Mammographic contain missing values in all features, the other five datasets contain missing values in some features.

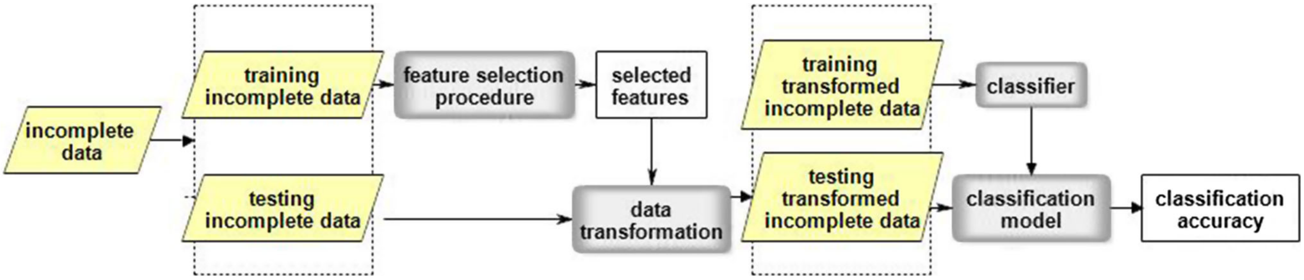

Fig. 1 Classification with incomplete datasets using a feature selection method before applying a classifier able to classify incomplete datasets

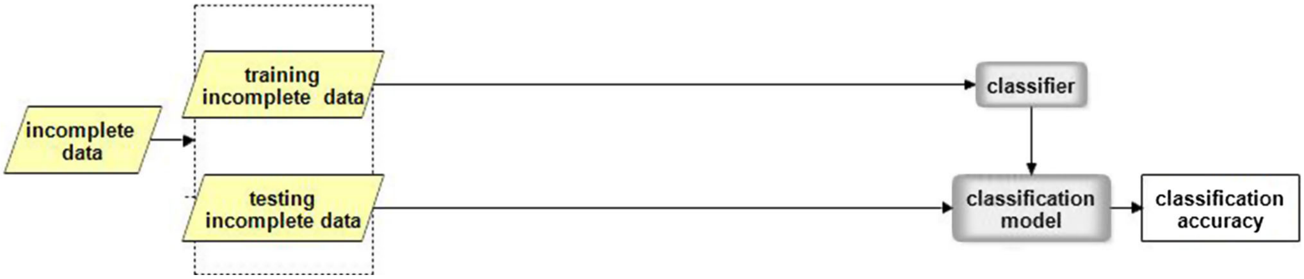

Fig. 2 Classification with incomplete datasets using a classifier able to classify incomplete datasets

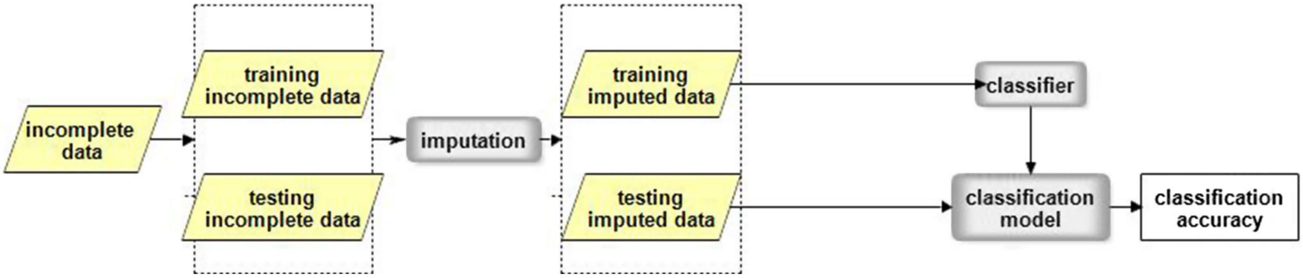

Fig. 3 Classification with incomplete datasets using an imputation method before using a classifier

Table 1 The datasets used in the experiments

Dataset	#Instances	#Features	#Classes	Missing Inst (%)	Missing features (%)
Breast	286	9	2	3.15	11.11
Cleveland	303	13	5	1.98	15.38
Hepatitis	155	19	2	48.39	78.94
Mammographic	961	5	2	13.63	100
Marketing	8993	13	9	23.54	69.23
Ozone	2536	73	2	27.12	100
Wisconsin	699	9	2	2.29	11.11
Climate	540	20	2	0	0
Ionosphere	351	34	2	0	0
Liver	345	7	2	0	0
Parkinsons	197	23	2	0	0
Robot	463	90	5	0	0
Sonar	208	60	2	0	0
Statlog	270	13	2	0	0

Therefore, with the last seven complete datasets, missing values were put randomly in 50 % random features in each dataset. With each dataset in the last seven datasets and each level of missing values in the six levels, repeat 30 times: randomly select 50 % features from original features, and then introduce the level of missing values in the chosen features. Hence, from one dataset and one level of missing values, 30 artificial datasets containing missing values were generated. Therefore, from one complete dataset, 180 (=30 × 6) artificial datasets containing missing values were generated and a total of 1260 (=180 × 7) artificial datasets containing missing values were used in the experiments.

None of the datasets in the experiments comes with a specific test set. Moreover, in some datasets, the number of instances is relatively small. Therefore, the ten-fold cross-validation method was used to measure the performance of the learned classifiers. With the first seven incomplete datasets, the ten-fold cross-validation method was performed 30 times. With the last seven complete datasets, with each dataset and each level of missing values, the ten-fold cross-validation method was performed on the 30 incomplete datasets. Consequently, for each incomplete dataset in the first seven datasets and each level of missing values on one dataset in the last seven datasets, 300 pairs of training and testing sets were generated.

3.3 Imputation algorithms

Three imputation methods including mean imputation, KNN-based imputation and EM-based imputation were used in the experiment. Mean imputation and KNN-based imputation were in-house implementations. With KNN-based imputation, for each incomplete dataset, different values for the number of neighbors K (5, 10, 15, 20, 30) were checked to find the optimal values by using the ten-fold cross-validation method. The experiments utilised WEKA [18] for EM-based imputation implementation by setting its parameters as the default values.

3.4 Classification algorithm

C4.5 is a decision tree able to directly classify incomplete datasets. The experiments utilised C4.5 to classify data and evaluate feature subsets in feature selection. The experiments utilised WEKA [18] for C4.5 implementation by setting its parameters as the default values.

Table 2 Size of trees with different thresholds

Dataset	All features	Threshold					
		0.05	0.2	0.4	0.6	0.8	0.95
Breast	22.8	19.3	18.1	16.5	16.1	15.2	12.7
Cleveland	79.4	73.9	70.2	53.8	32.8	17.8	12.6
Hepatitis	17.3	14.5	12.5	11.9	10.2	7.5	5.3
Mamomgraphic	10.3	10.1	10.0	9.9	9.9	9.3	6.1
Marketing	1368.2	1453.6	1398.5	873.9	305.7	285.3	189.4
Ozone	24.6	39.2	37.2	25.4	15.7	9.6	7.3
Wisconsin	23.3	19.7	18.2	16.3	15.9	15.4	12.7

Table 3 Classification accuracy with different thresholds

Dataset	All features	Threshold					
		0.05	0.2	0.4	0.6	0.8	0.95
Breast	94.55 ± 0.46	94.63 ± 0.45	**94.68** ± 0.54	94.51 ± 0.43	94.44 ± 0.44	94.52 ± 0.55	93.74 ± 0.72
Cleveland	54.29 ± 1.92	54.86 ± 1.83	55.74 ± 1.43	55.94 ± 1.98	57.21 ± 1.51	**57.82** ± 1.27	57.43 ± 1.74
Hepatitis	79.13 ± 2.02	79.35 ± 2.02	79.08 ± 1.69	79.22 ± 1.75	79.73 ± 2.22	80.27 ± 1.59	**81.53** ± 1.68
Mammographic	82.21 ± 0.45	**82.86** ± 0.41	82.72 ± 0.41	82.70 ± 0.64	82.68 ± 0.38	82.27 ± 0.53	81.80 ± 0.61
Marketing	31.40 ± 0.74	31.30 ± 0.74	31.33 ± 0.70	31.29 ± 0.85	31.62 ± 0.88	**31.72** ± 0.93	31.03 ± 1.24
Ozone	96.40 ± 0.43	96.26 ± 0.58	96.14 ± 0.47	95.95 ± 0.41	96.52 ± 0.52	96.67 ± 0.59	**96.98** ± 0.26
Wisconsin	94.70 ± 0.41	94.61 ± 0.64	**94.72** ± 0.57	94.44 ± 0.61	94.52 ± 0.64	94.54 ± 0.66	94.10 ± 0.72

Bold values indicate the best result for each dataset

Table 4 The average of accuracy comparison between C4.5FS and the other methods with datasets containing natural missing values

Dataset	C4.5FS	C4.5	T	C4.5MI	T	C4.5KNNI	T	C4.5EMI	T
Breast	94.68 ± 0.54	94.55 ± 0.46	=	94.31 ± 0.49	=	94.70 ± 0.43	=	**94.78** ± 0.51	=
Cleveland	**57.82** ± 1.27	54.29 ± 1.92	+	53.42 ± 2.24	+	53.67 ± 1.98	+	53.58 ± 1.99	+
Hepatitis	**81.53** ± 1.68	79.13 ± 2.02	+	77.70 ± 1.91	+	77.64 ± 1.75	+	78.25 ± 2.39	+
Mammographic	**82.86** ± 0.41	82.21 ± 0.45	+	81.92 ± 0.62	+	82.09 ± 0.64	+	82.35 ± 0.63	+
Marketing	**31.72** ± 0.93	30.80 ± 0.74	+	30.27 ± 0.82	+	30.43 ± 0.85	+	30.32 ± 0.77	+
Ozone	**96.98** ± 0.26	96.40 ± 0.43	+	96.17 ± 0.42	+	96.08 ± 0.41	+	96.03 ± 0.71	+
Wisconsin	94.62 ± 0.66	94.70 ± 0.41	=	94.47 ± 0.43	=	**94.87** ± 0.61	=	94.76 ± 0.61	=

Bold values indicate the best results for each dataset

3.5 PSO settings

PSO was used as a search technique for searching feature subsets in the feature selection method. The parameters of PSO in the feature selection method were chosen according to common settings proposed by Clerc and Kennedy [8]. The detailed settings are shown as follows: $\omega = 0.729844$, $c_1 = c_2 = 1.49618$, population size was set to 70, and the maximum iteration was set to 100. The fully connected topology is used. For each incomplete dataset, different values for the threshold θ to determine whether or not a feature is selected (0.05, 0.2, 0.4, 0.6, 0.8, 0.95) were checked to find the optimal values by using ten-fold cross-validation. The fitness function to evaluate particles is based on the performance of C4.5. For each dataset in the first seven incomplete datasets and each level of missing values on one dataset in the last seven datasets, 300 pairs of training set and test set were generated, so PSO was repeated 300 times on each dataset.

4 Results and analysis

This section presents the effects of threshold in PSO for feature selection, classification accuracy comparisons, the complexity of the learned models and further analysis.

4.1 Threshold in PSO for feature selection

To use PSO for feature selection, a threshold is required to determine whether a feature will be selected or not. To evaluate how the threshold affects the average number of selected features and the classification accuracy, PSO with different thresholds were used to do feature selection. Tables 2 and 3 show the average size of trees and the

Table 5 The average of accuracy comparison between C4.5FS and the other methods with datasets using several missing rates

Dataset	Missing rate (%)	C4.5FS	C4.5	T	C4.5MI	T	C4.5KNNI	T	C4.5EMI	T
Climate	5	**91.50** ± 0.52	90.19 ± 0.87	+	89.46 ± 0.97	+	89.81 ± 0.92	+	89.76 ± 0.85	+
	10	**91.49** ± 0.43	90.13 ± 1.04	+	89.21 ± 1.10	+	89.40 ± 1.40	+	89.62 ± 1.19	+
	20	**91.58** ± 0.37	90.51 ± 1.21	+	89.42 ± 1.01	+	89.54 ± 0.89	+	89.51 ± 1.04	+
	30	**91.41** ± 0.33	91.17 ± 0.71	+	88.97 ± 0.89	+	88.90 ± 0.98	+	89.20 ± 1.07	+
	40	**91.48** ± 0.27	91.29 ± 0.60	+	88.90 ± 1.23	+	88.98 ± 1.26	+	89.20 ± 0.91	+
	50	**91.41** ± 0.28	**91.29** ± 0.50	=	89.10 ± 1.33	+	89.14 ± 1.32	+	89.12 ± 1.33	+
Ionosphere	5	**90.73** ± 1.11	89.87 ± 1.66	+	89.98 ± 1.42	+	89.49 ± 1.42	+	89.59 ± 1.56	+
	10	**90.67** ± 1.35	89.86 ± 1.32	+	88.92 ± 1.58	+	89.68 ± 1.28	+	89.62 ± 1.38	+
	20	**90.45** ± 1.60	89.53 ± 1.14	+	88.81 ± 1.63	+	87.55 ± 1.81	+	89.07 ± 1.39	+
	30	**89.73** ± 1.64	89.62 ± 1.89	=	88.91 ± 1.92	+	88.45 ± 2.28	+	89.10 ± 1.89	+
	40	**88.96** ± 1.78	88.56 ± 1.46	=	87.48 ± 1.96	+	87.51 ± 1.95	+	88.30 ± 1.95	+
	50	**88.95** ± 2.54	88.17 ± 2.46	+	87.37 ± 2.24	+	87.37 ± 2.24	+	87.64 ± 2.36	+
Liver	5	65.69 ± 1.91	**65.88** ± 2.03	=	64.84 ± 2.30	=	65.24 ± 2.12	=	64.92 ± 2.51	=
	10	**64.38** ± 2.66	63.69 ± 2.50	=	63.68 ± 2.67	=	63.42 ± 2.72	=	63.56 ± 2.31	=
	20	**64.35** ± 2.63	64.06 ± 2.48	=	62.31 ± 2.58	+	63.54 ± 2.76	=	62.80 ± 2.65	+
	30	**62.98** ± 3.65	62.83 ± 3.66	=	62.51 ± 2.97	=	61.78 ± 3.29	+	62.18 ± 3.16	=
	40	**62.05** ± 3.79	62.01 ± 3.52	=	60.88 ± 4.42	=	61.19 ± 3.85	=	60.89 ± 3.27	+
	50	**61.52** ± 3.60	61.29 ± 2.93	=	59.99 ± 3.95	+	61.02 ± 3.11	+	60.41 ± 2.84	=
Parkinsons	5	**86.72** ± 2.28	85.63 ± 1.90	=	85.13 ± 2.34	+	84.83 ± 2.36	+	85.13 ± 1.95	+
	10	**86.09** ± 1.83	85.42 ± 2.16	=	84.07 ± 1.98	+	84.16 ± 2.29	+	84.84 ± 2.36	+
	20	**86.52** ± 2.07	85.40 ± 2.59	+	84.30 ± 2.01	+	84.13 ± 2.37	+	84.53 ± 2.40	+
	30	**86.47** ± 2.51	85.31 ± 2.24	+	83.14 ± 2.61	+	82.65 ± 2.44	+	83.57 ± 2.03	+
	40	**85.78** ± 1.78	84.77 ± 2.29	+	83.69 ± 2.79	+	84.08 ± 2.56	+	83.05 ± 1.90	+
	50	**85.55** ± 2.47	84.29 ± 2.54	+	81.76 ± 2.89	+	81.61 ± 2.79	+	82.07 ± 2.47	+
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	+
Statlog	5	**81.16** ± 2.18	78.92 ± 2.21	+	77.98 ± 2.03	+	78.02 ± 2.07	+	77.96 ± 2.15	+
	10	**80.06** ± 2.05	78.17 ± 2.07	+	77.22 ± 2.26	+	76.86 ± 2.01	+	76.91 ± 2.08	+
	20	**79.75** ± 2.88	77.56 ± 2.54	+	76.97 ± 2.42	+	76.46 ± 2.99	+	76.33 ± 2.37	+
	30	**78.81** ± 3.53	77.26 ± 2.99	+	77.08 ± 2.25	+	76.44 ± 3.07	+	75.58 ± 2.34	+
	40	75.93 ± 3.98	**76.44** ± 3.86	=	73.48 ± 4.54	+	73.83 ± 4.46	+	74.70 ± 2.80	=
	50	75.96 ± 3.90	**76.66** ± 3.81	=	72.75 ± 4.26	+	72.79 ± 4.76	+	72.30 ± 4.10	+

Bold values indicate the best results for each dataset

classification accuracy with different thresholds in PSO for the first seven datasets containing natural missing values.

It is clear from Table 2 that higher thresholds result in smaller trees than lower thresholds, and in all the datasets, high thresholds result in smaller trees than using all features. However, feature selection with low thresholds does not always result in smaller trees than using all features. For example, threshold needs be at least 0.4 for Marketing dataset and 0.6 for Ozone dataset to achieve smaller trees.

Table 3 shows that classification accuracy not only depends on thresholds, but also depends on datasets. With difficult datasets which have many classes such as Cleveland and Marketing or many features such as Ozone, to achieve classification improvement, threshold needs be high enough. In contrast, with datasets having a small number of features such as Breast, Mammographic and Wisconsin, low thresholds help to achieve classification improvement.

The purpose of feature selection is to reduce the number of original features, and retain or improve classification accuracy compared with using all features. Therefore, chosen thresholds have to not only help to reduce redundant features, but also retain or improve classification. In the experiments, for each dataset, different values of thresholds (0.05, 0.2, 0.4, 0.6, 0.8, 0.95) were checked to find the optimal values by using ten-fold cross-validation. The optimal threshold was chosen such that maximises classification accuracy and achieves smaller trees than using all of features. By using ten-fold cross-validation to choose thresholds, thresholds were chosen 0.05 for Mammographic, 0.2 for Breast and Wisconsin, 0.95 for Hepatitis and Ozone, and 0.8 for the other datasets.

4.2 Classification performance

Table 4 shows the average of classification accuracy and standard deviation of the first seven datasets. In the table, and in the following ones, C4.5FS column presents results from the first setup shown in Fig. 1, C4.5 column presents results from the second setup shown in Fig. 2; C4.5MI, C4.5KNNI and C4.5EMI columns present results from the third experimental setup shown in Fig. 3 by using mean imputation, KNN-based imputation and EM-based imputation, respectively.

With each dataset in the first seven datasets, the classification accuracy is the average of accuracies of the 30 times performing ten-fold cross-validation. Table 5 shows the average of classification accuracy and standard deviation of the last seven datasets with six levels of missing values. With each dataset and each missing level in the last seven datasets, the classification accuracy is the average of accuracies of the 30 generated incomplete datasets with the corresponding missing level.

To compare the performance of C4.5FS with the other methods, the Wilcoxon signed-ranks tests at 95 % confidence interval is used to compare the classification accuracy achieved by C4.5FS with the other methods. “T” columns in Tables 4 and 5 show significant test of the columns before them against C4.5FS, where “+”, “=” and “−” mean C4.5FS is significantly more accurate, not significantly different, and significantly less accurate, respectively.

Table 4 shows that C4.5FS can achieve significantly better classification accuracy or at least similar classification accuracy to the other methods with the datasets containing natural missing values. C4.5FS achieves similar classification accuracy to other methods on Breast and Wisconsin, significantly better classification accuracy than other methods on the other five datasets, and never significantly worse classification accuracy than the other methods.

The results from Table 5 are summarised in Fig. 4. Figure 4 shows that C4.5FS can obtain significantly better or at least similar classification accuracy compared to the other methods with artificial incomplete datasets. It is also clear from Fig. 4 that C4.5FS can achieve more times significantly better than C4.5MI and followed by C4.5KNNI, C4.5EMI and C4.5.

In order to confirm if C4.5FS is really significantly better than the others, we perform Friedman’s test on the accuracies of all the algorithms in the 49 datasets. Table 6 shows the ranking of the algorithms using Friedman’s test. It is clear from Table 6 that C4.5FS is the best algorithm, followed by C4.5, C4.5EMI, C4.5KNNI and C4.5MI. Furthermore, we perform post hoc tests to carry out pairwise comparisons. Table 7 shows pairwise comparisons using Holm and Shaffer as a post-hoc procedure [14]. It’s clear form Table 7 that C4.5FS is significantly better than each of the other algorithms. Moreover, the three imputation methods cannot help to improve C4.5, and this observation is similar to the observation in [12], where C4.5 was proved to be better than the combination of C4.5

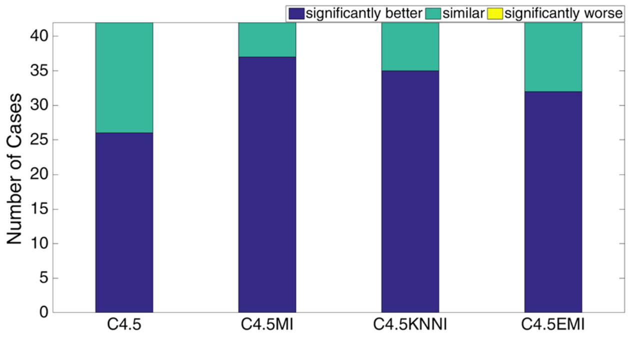

Fig. 4 Accuracy comparison of C4.5FS with the other methods

and imputation methods. Therefore, the new approach to using feature selection to improve C4.5 is essential.

In summary, the feature selection method for incomplete datasets is able to help enhance classification accuracy of C4.5 not only with natural incomplete datasets, but also with artificial incomplete datasets.

4.3 Complexity of the learned models

Tables 8 and 9 show the average number of selected features, the average time of feature selection and the average size of decision trees (the number of nodes in the trees) by utilising C4.5 in different ways in the first seven datasets and the last seven datasets with six levels of missing values, respectively.

Table 6 The ranking of the algorithms using Friedman's test

Algorithm	Ranking
C4.5FS	1.1939
C4.5	2.1939
C4.5EMI	3.6327
C4.5KNNI	3.8163
C4.5MI	4.1633

Table 7 Pairwise comparisons using a post-hoc procedure

Algorithms	Holm	Shaffer
C4.5FS versus C4.5	**0.0125**	**0.0125**
C4.5FS versus C4.5MI	**0.0050**	**0.0050**
C4.5FS versus C4.5KNNI	**0.0056**	**0.0083**
C4.5FS versus C4.5EMI	**0.0063**	**0.0083**
C4.5 versus C4.5MI	**0.0071**	**0.0083**
C4.5 versus C4.5KNNI	**0.0083**	**0.0083**
C4.5 versus C4.5EMI	**0.0100**	**0.0125**
C4.5MI versus C4.5KNNI	0.0250	0.0250
C4.5MI versus C4.5EMI	0.0167	0.0167
C4.5KNNI versus C4.5EMI	0.0500	0.0500

Bold values indicate the significant difference

According to Table 8, with the first seven datasets containing natural missing values, the feature selection helps to reduce at least 30 % the number of original features. Moreover, in some datasets such as Hepatitis and Ozone, the feature selection helps to reduce more than 90 % the number of original features.

According to Table 9, with artificial incomplete datasets, the feature selection helps to reduce at least 50 % the number of original features. Moreover, in some datasets such as Robot and Sonar, the feature selection helps to reduce around 90 % the number of original features.

Table 8 shows that C4.5FS is able to generate significantly smaller decision trees than the other methods in all cases. For example, with Marketing and Ozone datasets, the average of tree sizes generated by C4.5FS is around one fifth of the average of tree sizes generated by C4.5 and more than one fifth of the average of tree sizes generated by utilising imputation methods before using C4.5.

Figure 5 presents the minimum, average and maximum ratios between the average of tree sizes generated by C4.5, C4.5MI, C4.5KNNI and C4.5EMI with C4.5FS from Table 9. The minimum ratio between the average of tree sizes generated by the other methods with C4.5FS depicts that C4.5FS can generate smaller trees than the other methods. On average, the average of tree sizes generated by C4.5 is about 50 % bigger than those generated by C4.5FS, and the average of tree sizes generated by the other methods is over nearly three times bigger than those of C4.5FS. Moreover, the maximum of ratio between the average of tree sizes generated by the other methods with C4.5FS depicts that the average of tree sizes generated by using imputation methods in some cases are dramatically bigger than C4.5FS. The major reason is possibly that imputation methods often generate further values for missing features; hence, if the incomplete features are

Table 8 The average of tree sizes generated by C4.5FS and the other methods with datasets containing natural missing values

Dataset	#Features		Tree size					Feature selection time (ms)
	All	Selected features	C4.5FS	C4.5	C4.5MI	C4.5KNNI	C4.5EMI	
Breast	9	6.1	**16.2**	23.3	23.6	22.6	22.9	2.7×10^4
Cleveland	13	2.9	**17.8**	79.4	82.0	82.5	82.1	5.0×10^3
Hepatitis	19	1.6	**5.3**	17.3	19.6	21.0	18.9	2.3×10^3
Mammographic	5	3.7	**10.1**	10.3	20.5	11.5	13.6	4.3×10^4
Marketing	13	7.8	**285.3**	1368.1	1721.3	1676.3	1718.4	9.8×10^5
Ozone	73	4.1	**7.3**	24.6	29.5	30.9	30.2	4.9×10^4
Wisconsin	9	5.0	**18.2**	23.3	24.1	22.5	22.4	2.8×10^4

Bold values indicate the best results for each dataset

Table 9 The average of tree sizes generated by C4.5FS and the other methods with datasets using several missing rates

Dataset	Missing rate (%)	#Features		Tree size					Feature selection time (ms)
		All	Selected features	C4.5FS	C4.5	C4.5MI	C4.5KNNI	C4.5EMI	
Climate	5	20	4.5	**5.4**	23.8	25.6	25.4	23.9	7.9×10^3
	10		3.7	**4.6**	20.4	24.7	24.1	23.5	7.7×10^3
	20		4.5	**4.3**	17.3	27.9	26.1	25.1	6.3×10^3
	30		4.6	**2.7**	7.8	26.5	25.6	23.5	7.5×10^3
	40		3.9	**2.3**	5.4	26.3	26.7	20.9	6.6×10^3
	50		4.0	**2.8**	8.5	27.3	27.7	22.1	8.7×10^3
Ionosphere	5	34	4.8	**15.0**	25.8	26.3	26.2	26.0	1.9×10^4
	10		5.3	**14.8**	23.7	25.5	25.6	25.7	2.4×10^4
	20		5.8	**15.9**	23.5	25.8	25.1	25.4	1.9×10^4
	30		5.6	**15.5**	22.2	26.5	26.2	25.2	1.1×10^4
	40		6.1	**16.9**	21.4	28.1	28.0	25.9	1.0×10^4
	50		5.1	**17.4**	20.7	27.9	27.9	25.4	1.1×10^4
Liver	5	7	2.7	**13.2**	40.9	46.3	47.3	46.6	4.8×10^4
	10		2.0	**11.8**	33.9	42.2	41.2	40.1	3.5×10^3
	20		3.0	**10.4**	26.2	40.7	36.0	34.8	4.4×10^3
	30		2.7	**15.0**	17.0	19.6	19.5	18.8	3.4×10^3
	40		2.6	**9.0**	20.4	35.2	29.4	30.5	4.1×10^3
	50		1.9	**7.6**	14.2	29.1	24.7	22.7	2.5×10^3
Parkinsons	5	23	4.1	**15.0**	17.8	19.0	18.7	18.8	7.4×10^3
	10		4.6	**15.5**	17.9	18.7	19.2	18.5	7.6×10^3
	20		3.9	**15.4**	17.8	19.9	19.5	18.7	6.1×10^3
	30		4.2	**15.0**	17.0	19.6	19.5	18.8	6.6×10^3
	40		4.4	**13.9**	15.8	19.7	19.6	18.3	1.2×10^4
	50		4.0	**13.3**	14.6	19.3	19.2	18.7	1.1×10^4
Robot	5	90	6.9	**63.6**	71.3	118.4	106.6	100.9	1.9×10^5
	10		8.7	**69.4**	76.0	133.9	133.9	120.7	3.2×10^5
	20		7.5	**73.1**	86.4	131.7	131.7	129.2	1.4×10^6
	30		10.1	**74.6**	85.4	129.2	129.2	126.8	1.3×10^6
	40		6.3	**70.8**	79.4	128.4	128.4	125.0	1.0×10^6
	50		10.5	**63.7**	73.1	129.2	129.2	121.3	1.1×10^6
Sonar	5	60	9.0	**25.1**	27.7	28.0	27.5	27.9	2.7×10^3
	10		7.9	**25.4**	28.1	28.7	28.3	27.9	2.8×10^3
	20		9.7	**24.8**	28.4	29.5	29.5	27.6	6.9×10^3
	30		8.2	**23.2**	27.5	30.3	30.3	28.2	6.1×10^3
	40		7.4	**22.4**	26.7	30.6	30.7	28.7	6.9×10^3
	50		7.6	**22.7**	26.3	31.8	32.0	29.2	6.3×10^3
Statlog	5	13	3.7	**12.8**	30.4	34.6	34.3	33.9	5.9×10^3
	10		3.5	**12.9**	29.4	36.8	35.6	35.3	5.3×10^3
	20		3.7	**12.6**	26.2	36.0	36.1	35.0	5.9×10^3
	30		4.0	**12.0**	23.1	36.4	36.3	35.5	7.2×10^3
	40		4.0	**11.5**	21.8	38.5	38.1	36.9	5.7×10^3
	50		4.4	**10.9**	19.0	38.2	37.9	36.2	7.1×10^3

Bold values indicate the best results for each dataset

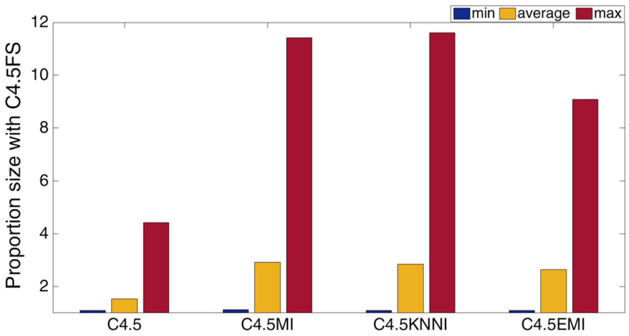

Fig. 5 Tree size ratio between the other methods and C4.5FS

selected to make decision trees, the further values can make decision trees bigger.

Tables 8 and 9 show that the feature selection time strongly depends on the datasets. With datasets containing many instances such as Marketing and datasets containing many classes such as Robot, the feature selection procedure often requires longer time than datasets containing small instances and classes. The main reason is that datasets containing many instances and classes often require longer time for the classifier evaluating feature subsets.

In summary, in all cases, the feature selection method for incomplete datasets is able to help reduce complexity of the learned trees generated by C4.5, especially compared to using imputation methods before using C4.5.

4.4 Further analysis

In order to know how C4.5FS has the ability of achieving better classification accuracy and smaller trees than the other methods, we analysed carefully the trees generated by using C4.5 and C4.5FS on Climate dataset. Climate dataset which has 20 features $\{f_1,..,f_{20}\}$ was chosen because the trees generated by using C4.5 and C4.5FS on the Climate are not so big to analyse. Figures 6 and 7 present two typical trees we observed.

Figure 6 presents two trees generated by using C4.5 and using C4.5FS on Climate with 20 % missing values in nine features $\{f_2,f_3,f_4,f_{12},f_{13},f_{14},f_{15},f_{17},f_{19}\}$. After using the feature selection method on the dataset, seven features $\{f_1,f_4,f_{11},f_{15},f_{16},f_{17},f_{20}\}$ were chosen. The tree generated by C4.5FS achieved slightly higher classification accuracy compared to the tree generated by C4.5 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 tree generated by C4.5 tree included more features which were not included in the tree generated by C4.5FS because these features already had been removed by the feature selection procedure. Consequently, C4.5FS can achieve not only better classification accuracy but also smaller trees than the C4.5.

Figure 7 presents two trees generated using C4.5 and C4.5FS on Climate with 20 % missing values introduced in 10 features $\{f_1,f_4,f_5,f_7,f_9,f_{10},f_{13},f_{14},f_{16},f_{19}\}$. After using the feature selection method, seven features $\{f_7,f_8,f_9,f_{13},f_{15},f_{16},f_{18}\}$ were selected. In C4.5, when computing the information gain of a feature containing missing values, it firstly calculates the gain based on the complete values and discounts the gain by the proportion of complete instances to all instances [33]. It means that missing values reduce the information gain of incomplete features. Hence, C4.5 tends to select complete features to make decision trees; however the bias of selecting complete features to build decision trees is not always good. For instance, on Fig. 7, while the first node of the tree generated by C4.5 is a complete feature f_3, the first node of the tree generated by C4.5FS is an incomplete feature f_{16}. Nevertheless, the tree generated by C4.5FS obtained both better accuracy (91.3 vs 90.1 %) and smaller size than the tree generated by C4.5. A possible reason could be that by removing less suitable features such as f_3, the feature selection is able to reduce the C4.5's bias towards selecting complete features to build decision trees.

In summary, the feature selection method can choose relevant features and remove irrelevant features. Thus, the feature selection is able to make better classifier.

5 Conclusions and future work

This paper has attempted to find the impact of a wrapper feature-based feature selection method for incomplete datasets. To achieve this goal, a wrapper-based feature selection method for incomplete datasets is proposed and compared with the two other common methods coping with incomplete datasets: one using a classifier able to directly classify incomplete datasets and the other using an imputation method to transfer incomplete datasets to complete datasets. The three setups were compared on 14 datasets where seven datasets contain natural missing values and the other seven datasets contain six levels of artificial missing values. The experiments used C4.5 as an evaluation and PSO as a search method for the feature selection approach. The experimental results showed that the proposed wrapper-based feature selection method for incomplete datasets is able to help to enhance the classification accuracy of C4.5, significantly reduce the number of original features and significantly reduce the complexity of the learned classifier.

The experiments in this paper used C4.5 as a classifier since C4.5 can cope with incomplete datasets. There are some other classifiers that are able to classify incomplete datasets such as CART [10] and CN2 [7]. Future work could perform this investigation with CART and CN2.

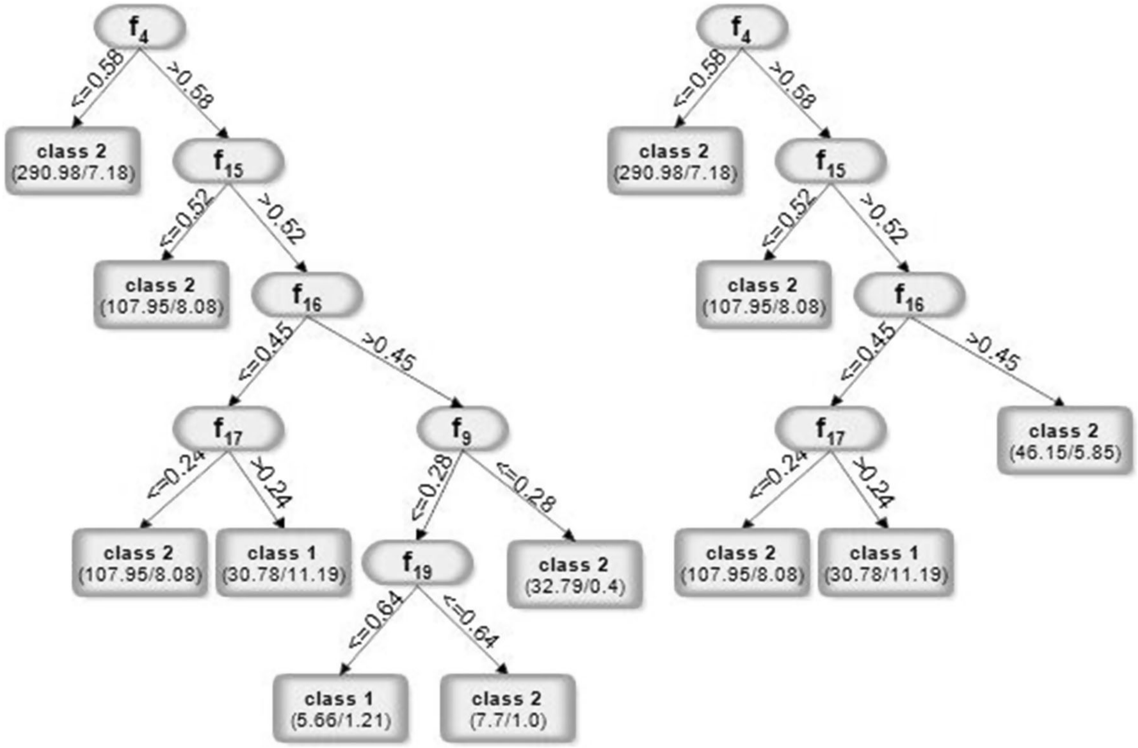

Fig. 6 The left decision tree generated by using C4.5 and the right decision tree generated by using C4.5FS on Climate with 20 % missing fields in features $\{f_2,f_3,f_4,f_{12},f_{13},f_{14},f_{15},f_{17},f_{19}\}$

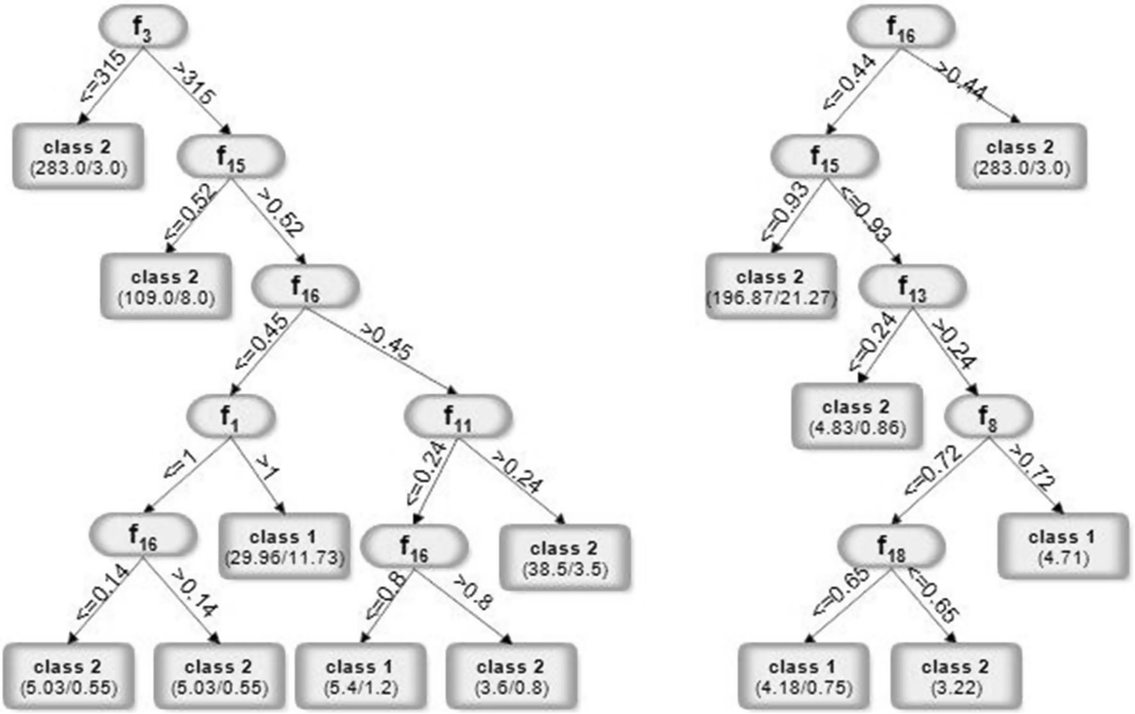

Fig. 7 The left decision tree generated by using C4.5 and the right decision tree generated by using C4.5FS on Climate with 20 % missing fields in features $\{f_1,f_4,f_5,f_7,f_9,f_{10},f_{13},f_{14},f_{16},f_{19}\}$

Furthermore, the experiments tested the proposed method with datasets involving not too many features. Therefore, another future work could consider datasets involving more features to test the scalability of the proposed method.

References

1. Lichman M (2013) UCI Machine Learning Repository. School of Information and Computer Science, University of California, Irvine, CA. http://archive.ics.uci.edu/ml
2. Barnard J, Meng X-L (1999) Applications of multiple imputation in medical studies: from aids to nhanes. Stat Methods Med Res 8:17–36
3. Batista GE, Monard MC (2002) A study of K-nearest neighbour as an imputation method. HIS 87:251–260
4. Berger JO (2013) Statistical decision theory and Bayesian analysis. Springer, New York
5. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
6. Chuang L-Y, Chang H-W, Tu C-J, Yang C-H (2008) Improved binary pso for feature selection using gene expression data. Comput Biol Chem 32:29–38
7. Clark P, Niblett T (1989) The CN2 induction algorithm. Mach Learn 3:261–283
8. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evol Comput 6:58–73
9. Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1:131–156
10. De'ath G, Fabricius KE (2000) Classification and regression trees: a powerful yet simple technique for ecological data analysis. Ecology 81:3178–3192
11. Doquire G, Verleysen M (2012) Feature selection with missing data using mutual information estimators. Neurocomputing 90:3–11
12. Farhangfar A, Kurgan L, Dy J (2008) Impact of imputation of missing values on classification error for discrete data. Pattern Recogn 41:3692–3705
13. Farhangfar A, Kurgan LA, Pedrycz W (2007) A novel framework for imputation of missing values in databases. IEEE Trans Syst Man Cybern A Syst Hum 37:692–709
14. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the cec2005 special session on real parameter optimization. J Heuristics 15:617–644
15. García-Laencina PJ, Sancho-Gómez J-L, Figueiras-Vidal AR (2010) Pattern classification with missing data: a review. Neural Comput Appl 19:263–282
16. Graham JW (2009) Missing data analysis: making it work in the real world. Annu Rev Psychol 60:549–576
17. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
18. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11:10–18
19. Hall MA (1999) Correlation-based feature selection for machine learning. PhD thesis, The University of Waikato
20. Han J, Kamber M, Pei J (2006) Data mining, southeast asia edition: concepts and techniques. Morgan kaufmann, San Francisco
21. Huang C-L, Dun J-F (2008) A distributed pso-svm hybrid system with feature selection and parameter optimization. Appl Soft Comput 8:1381–1391
22. Jain A, Zongker D (1997) Feature selection: evaluation, application, and small sample performance. IEEE Trans Pattern Anal Mach Intell 19:153–158
23. Kennedy J (2010) Particle swarm optimization. In: Encyclopedia of machine learning, pp 760–766
24. Kennedy J, Kennedy JF, Eberhart RC (2001) Swarm intelligence. Morgan Kaufmann, San Francisco
25. Kohavi R, John GH (1997) Wrappers for feature subset selection. Artif Intell 97:273–324
26. Koller D, Sahami M (1995) Toward optimal feature selection. In: 13th international conference on machine learning, pp 284–292
27. Lane MC, Xue B, Liu I, Zhang M (2014) Gaussian based particle swarm optimisation and statistical clustering for feature selection. In: European conference on evolutionary computation in combinatorial optimization, pp 133–144
28. Lin S-W, Ying K-C, Chen S-C, Lee Z-J (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Syst Appl 35:1817–1824
29. Little RJ, Rubin DB (2014) Statistical analysis with missing data. Wiley, Hoboken
30. MacKay DJ (2003) Information theory, inference, and learning algorithms, vol 7. Citeseer
31. Oh I-S, Lee J-S, Moon B-R (2004) Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell 26:1424–1437
32. Qian W, Shu W (2015) Mutual information criterion for feature selection from incomplete data. Neurocomputing 168:210–220
33. Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier, Amsterdam
34. Schafer JL (1997) Analysis of incomplete multivariate data. CRC Press, Boca Raton
35. Schafer JL, Graham JW (2002) Missing data: our view of the state of the art. Psychol Methods 7:147
36. Tran CT, Andreae P, Zhang M (2015) Impact of imputation of missing values on genetic programming based multiple feature construction for classification. In: 2015 IEEE congress on evolutionary computation (CEC), pp 2398–2405
37. Tran CT, Zhang M, Andreae P (2015) Multiple imputation for missing data using genetic programming. In: Proceedings of the 2015 annual conference on genetic and evolutionary computation, pp 583–590
38. Tran CT, Zhang M, Andreae P (2016) A genetic programming-based imputation method for classification with missing data. In: European conference on genetic programming. Springer, pp 149–163
39. Tran CT, Zhang M, Andreae P, Xue B (2016) A wrapper feature selection approach to classification with missing data. In: Applications of evolutionary computation, pp 685–700
40. Xue B, Zhang M, Browne W, Yao X (2015) A survey on evolutionary computation approaches to feature selection. IEEE Trans Evol Comput 99:1
41. Xue B, Zhang M, Browne WN (2012) Single feature ranking and binary particle swarm optimisation based feature subset ranking for feature selection. In: Proceedings of the thirty-fifth Australasian computer science conference, vol 122, pp 27–36
42. Xue B, Zhang M, Browne WN (2013) Particle swarm optimization for feature selection in classification: a multi-objective approach. IEEE Trans Cybern 43:1656–1671
43. Xue B, Zhang M, Browne WN (2015) A comprehensive comparison on evolutionary feature selection approaches to classification. Int J Comput Intell Appl 14:1550008
44. Yang J, Honavar V (1998) Feature subset selection using a genetic algorithm. In: Feature extraction, construction and selection, pp 117–136