Pattern Recognition 69 (2017) 52–60

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Adjusted weight voting algorithm for random forests in handling missing values

Jing Xia[a], Shengyu Zhang[a], Guolong Cai[b], Li Li[b], Qing Pan[c], Jing Yan[b,*], Gangmin Ning[a,*]

[a] Department of Biomedical Engineering, Key Laboratory of Biomedical Engineering of Ministry of Education, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China
[b] Department of ICU, Zhejiang Hospital, 12 Lingyin Road, Hangzhou 310013, China
[c] College of Information Engineering, Zhejiang University of Technology, 288 Liuhe Road, Hangzhou 310023, China

ARTICLE INFO

Article history:
Received 19 July 2016
Revised 11 March 2017
Accepted 4 April 2017
Available online 5 April 2017

Keywords:
Random forests
Missing values
Imputation approaches
Surrogate decisions
Weighted voting

ABSTRACT

Random forests (RF) is known as an efficient algorithm in classification, however it depends on the integrity of datasets. Conventional methods in dealing with missing values usually employ estimation and imputation approaches whose efficiency is tied to the assumptions of data features. Recently, algorithm of surrogate decisions in RF was developed and this paper proposes a random forests algorithm with modified surrogate splits (Adjusted Weight Voting Random Forest, AWVRF) which is able to address the incomplete data without imputation.

Differing from the present surrogate method, in AWVRF algorithm, when the primary splitting attribute and the surrogate attributes of an internal node are all missing, the undergoing instance is allowed to exit at the current node with a vote. Then the weight of the vote is adjusted by the strength of the involved attributes and the final decision is made by weighted voting. AWVRF does not comprise imputation step, thus it is independent of data features.

AWVRF is compared with the methods of mean imputation, LeoFill, knnimpute, BPCAfill and conventional RF with surrogate decisions (surrRF) using 50 times repeated 5-fold cross validation on 10 acknowledged datasets. In a total of 22 experiment settings, the method of AWVRF harvests the highest accuracy in 14 settings and the largest AUC in 7 settings, exhibiting its superiority over other methods. Compared with surrRF, AWVRF is significantly more efficient and remain good discrimination of prediction. Experimental results show that the present AWVRF algorithm can successfully handle the classification task for incomplete data.

1. Introduction

Machine learning is widely used in solving real-world problems, including medical diagnosis, face recognition and document categorization [1–6]. Among various classification algorithms in machine learning, the random forests (RF) algorithm receives more and more attention due to its outstanding performance [7–11]. Random forests is an ensemble of decision tree classifiers [12] with acknowledged advantages, such as superior classification accuracy, ability of handling high dimensional data without attribute selection, and the feasibility to be conducted by parallel computing [7–9,11–13]. Its main idea is to train a series of decision trees (CART) [14] in which each node attribute is selected from a random set of attributes and the trees are built on bootstrap samples of the data. And the total decision trees make up the forest. For the established forest, the prediction result is determined by majority vote of the decision trees in it [9,12,15–19].

The design of random forests algorithm is based on complete data, however, it is common to have incomplete data in classification cases [20–23]. In the UCI repository [24], one of most commonly used dataset collection, only 55% of datasets are complete while the rest 45% have missing values [23]. And it is more serious in clinical information databases [25,26]. In clinic, certain physiological data of patients, such as lab test results, may be missed due to insufficient monitoring, equipment malfunctioning, unaffordable examination fee or other reasons [27,28].Hence, how to deal with missing data is crucial for the random forests in solving clinical classification problems.

To handle missing data, the most popular strategy is data imputation before classification, that is to estimate and fill the missing values according to the information from existing data. There are

* Corresponding author.
E-mail addresses: zjicu@vip.163.com (J. Yan), gmning@zju.edu.cn (G. Ning).

http://dx.doi.org/10.1016/j.patcog.2017.04.005
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

various imputation approaches, such as mean imputation, hot-deck imputation, k-nearest neighbors imputation (knnimpute), regression imputation, Bayesian estimation and Expectation Maximization (EM), etc. [20–22,25,27,29,30]. It is reported in literatures that imputation approaches are restricted to specific assumptions of data distribution or data correlation [27,30,31]. Conventional mean imputation is a kind of widely applied approach, however, it may lead to the underestimation of the data variance after averaging operation [22].Hot-deck imputation and knnimpute methods require sufficiently dense data which is not common [27,31].Bayesian estimation and EM ask for prior knowledge of data distribution, nevertheless such knowledge is often vague or non-existed [27].Incorrect distribution estimation will lead to inferior performance. And linear regression imputation has the disadvantage of difficulty in handling high dimensions and moreover, the presupposition of linear relationship between attributes is not always satisfied in practice [23,27].

To accommodate the imputation associated problems, efforts have been done to develop classification algorithms independent of explicit imputations [23]. Some works combined an ensemble of neural network classifiers to attenuate the effect of missing data on final decision, and each of the classifiers was trained on a single or a small proportion of attributes [27,32]. These approaches are based on neural network and the solution of them is to use reduced number of attributes for base classifiers. But the idea is not suitable for RF as the reduction of attributes for base classifiers would decrease the randomness in attributes selection, weaken the strength of single tree and eventually affect the performance of RF. Some work suggested the usage of surrogate splits on the nodes where the primary split is unable to proceed due to the presence of missing values [13,14,33–37]. Surrogate splits resemble the action of the primary split and yield the similar partitioning of the observed values as the primary split. Once an attribute selected for the next split in a tree is missing, the corresponding case is able to further down the tree by means of surrogate attributes. If all surrogate attributes are missing, the strategy of prominent decision is employed to determine which direction to go. The whole process is known as surrogate decision. The strategy of prominent decision only considers the majority of training cases, however neglects the individual characteristics of the undergoing case, which may lead to incorrect judgments [33,34]. The surrogate decision strictly demands for a full prediction path from the root to the leaf node even it goes in a wrong direction at an internal node where a bad prominent decision is made. When such algorithm applied in random forests, the computing efficiency will also be reduced.

In this paper, we proposed a RF algorithm with modified surrogate splits, namely as Adjusted Weight Voting Random Forest (AWVRF), which is able to accomplish the classification task for incomplete data. AWVRF differs from the imputation methods in the aspects: It does not estimate the missing values or employ data imputation. Instead, AWVRF adjusts the voting weights of each tree by estimating the influence of missing data on the decision of the tree. In simple words, low voting weight is assigned to those trees whose decisions are severely influenced by missing values and on the contrary high voting weight is assigned to the less affected trees. And the final prediction of the AWVRF comes from the ensemble of weighted votes. AWVRF employs the strategy of surrogate splits, but different from the surrogate decision: it allows the prediction process to exit at the internal node whose primary splitting attribute and surrogate attributes are all missing. The present algorithm avoids the pitfalls due to the estimation and imputation based techniques, in particular the restriction of data features, while it is also expected to improve the performance of surrogate decision based RF algorithm.

2. Methods

2.1. AWVRF algorithm

In conventional RF algorithm with surrogate decision, only if an instance runs through a complete path from the root node to the leaf node, the tree casts a vote, whereas in AWVRF algorithm, when missing values are encountered in a tree, the undergoing instance is allowed to exit at the current internal node with a voting. However, the voting is weighted by the strength of the involved attributes. Finally, the overall prediction is made by weighted integration of votes of all trees.

The procedure of AWVRF algorithm is provided in Fig. 1.

Assume there are n training instances and the ith instance is represented as $(x_{i,}y_i)$, $i = 1, 2, \ldots, n$. The vector x_i consists of a values (a is the number of attributes) and can be also written as $(x_{i1}, x_{i2}, \ldots, x_{ia})$. And y_i is the target label of the ith instance. In binary classification problems, y_i has only two possible values denoted as C_1 and C_2.

AWVRF algorithm uses complete training dataset to create an ensemble of trees in the first step. Each tree learns from bootstrapping instances of the complete training dataset and the splitting attribute for each node is selected from a randomly chosen subset of attributes [12,19]. Let $\{h_1(x), h_2(x), \ldots, h_T(x)\}$ denote the set of T decision trees. Each node of each tree is labeled with C_1 or C_2. The label of each node is determined by the major label of training instances reaching the node.

Then in the testing phase, AWVRF is able to treat testing instances with missing values. Detailed procedure is described as follows.

2.1.1. Individual tree prediction

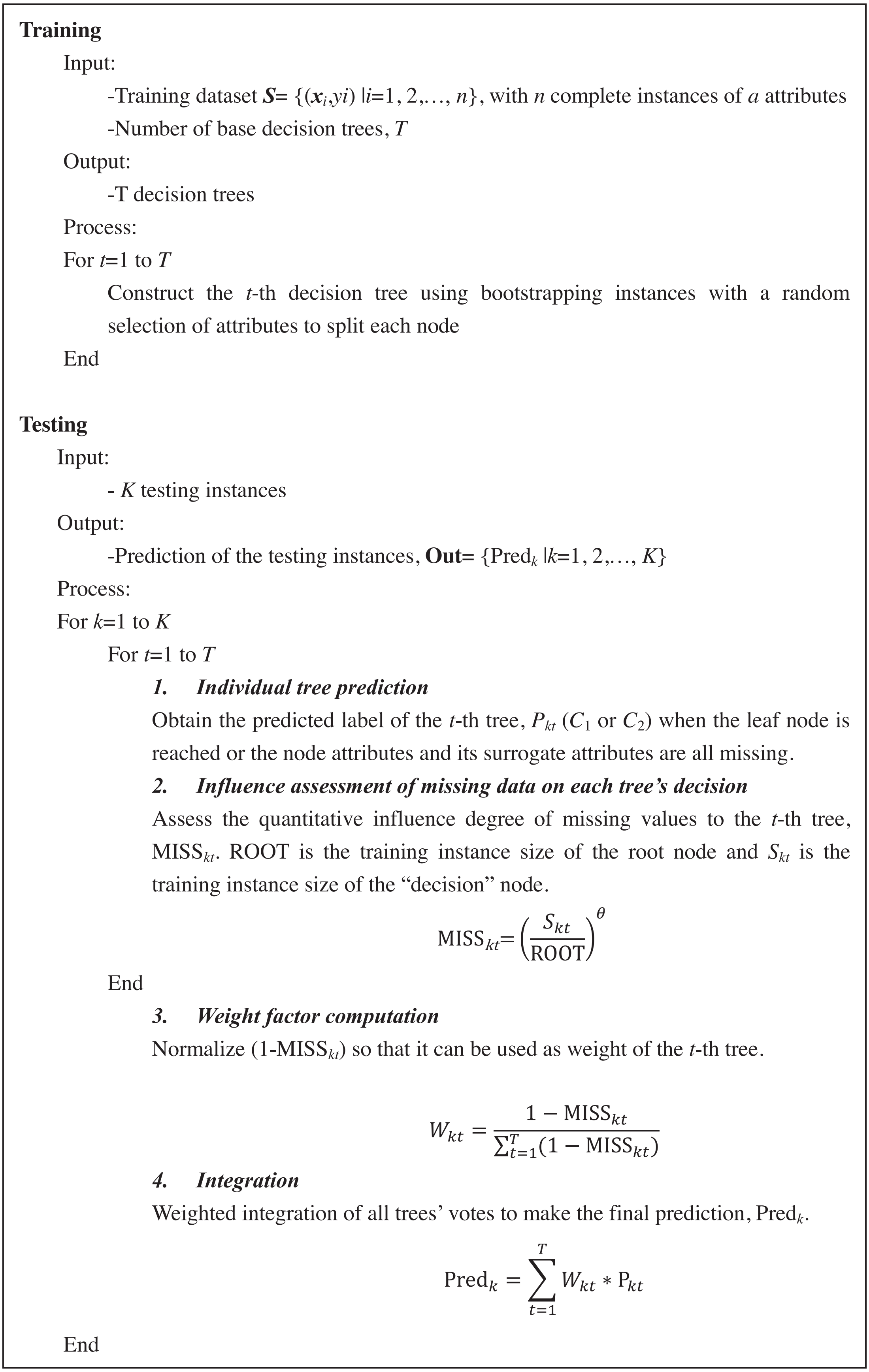
When a complete testing instance goes through a tree, the prediction process obeys the principle of tree with surrogate splits. The instance starts at the root node, stops at a leaf node, and then outputs the class label of the stopping leaf node. When a testing instance with missing values is being classified, it stops at an internal node whose primary splitting attribute and all surrogate attributes are missing. In this case, the output is the corresponding class label of the stopping internal node. In this way, each tree casts a vote, no matter the testing instance is complete or not. The aforementioned stopping leaf node and stopping internal node, is defined as “decision” node of the tree for the undergoing testing instance.

For the kth testing instance, obtain the predicted label of the tth tree, denoted as P_{kt} (C_1 or C_2). And obtain training instance size of the “decision” node, denoted as S_{kt}. Meanwhile, record training instance size of the root node denoted as ROOT for the next step. The parameter S_{kt} is seen to be a measurement of the distance between the “decision” node and the root node. When the “decision” node is near the root node, that is, the route between them is short, then the corresponding S_{kt} value is big. On the contrary, when the “decision” node is far away from the root node, that is, the route is long, then the corresponding S_{kt} value is small.

2.1.2. Influence assessment of missing data on each tree’s decision

For a new instance with missing values, it’s not proper to equally treat each tree as the missing values have different influence on different trees’ votes. To address the problem, AWVRF algorithm adjusts the weight of each tree according to the degree of the tree being influenced by missing values. It is a key issue to evaluate the degree of each tree being influenced by missing values and then assign weights to different trees.

In AWVRF algorithm, MISS_{kt} is computed to quantitatively assess the influence degree of missing values on the tth tree $h_t(x)$, as formula (1) shows. In the case that the “decision” node is near



```
Training
    Input:
        -Training dataset $\boldsymbol{S}$= {($\boldsymbol{x}_i$,$yi$) |$i$=1, 2,…, $n$}, with $n$ complete instances of $a$ attributes
        -Number of base decision trees, $T$
    Output:
        -T decision trees
    Process:
    For $t$=1 to $T$
        Construct the $t$-th decision tree using bootstrapping instances with a random selection of attributes to split each node
    End

Testing
    Input:
        - $K$ testing instances
    Output:
        -Prediction of the testing instances, Out= {Pred$_k$ |$k$=1, 2,…, $K$}
    Process:
    For $k$=1 to $K$
        For $t$=1 to $T$
            1. Individual tree prediction
            Obtain the predicted label of the $t$-th tree, $P_{kt}$ ($C_1$ or $C_2$) when the leaf node is reached or the node attributes and its surrogate attributes are all missing.
            2. Influence assessment of missing data on each tree's decision
            Assess the quantitative influence degree of missing values to the $t$-th tree, MISS$_{kt}$. ROOT is the training instance size of the root node and $S_{kt}$ is the training instance size of the "decision" node.
                $\mathrm{MISS}_{kt} = \left(\frac{S_{kt}}{\mathrm{ROOT}}\right)^{\theta}$
        End
            3. Weight factor computation
            Normalize (1-MISS$_{kt}$) so that it can be used as weight of the $t$-th tree.
                $W_{kt} = \frac{1 - \mathrm{MISS}_{kt}}{\sum_{t=1}^{T}(1 - \mathrm{MISS}_{kt})}$
            4. Integration
            Weighted integration of all trees' votes to make the final prediction, Pred$_k$.
                $\mathrm{Pred}_k = \sum_{t=1}^{T} W_{kt} * \mathrm{P}_{kt}$
    End
```

Fig. 1. Procedure of AWVRF algorithm.

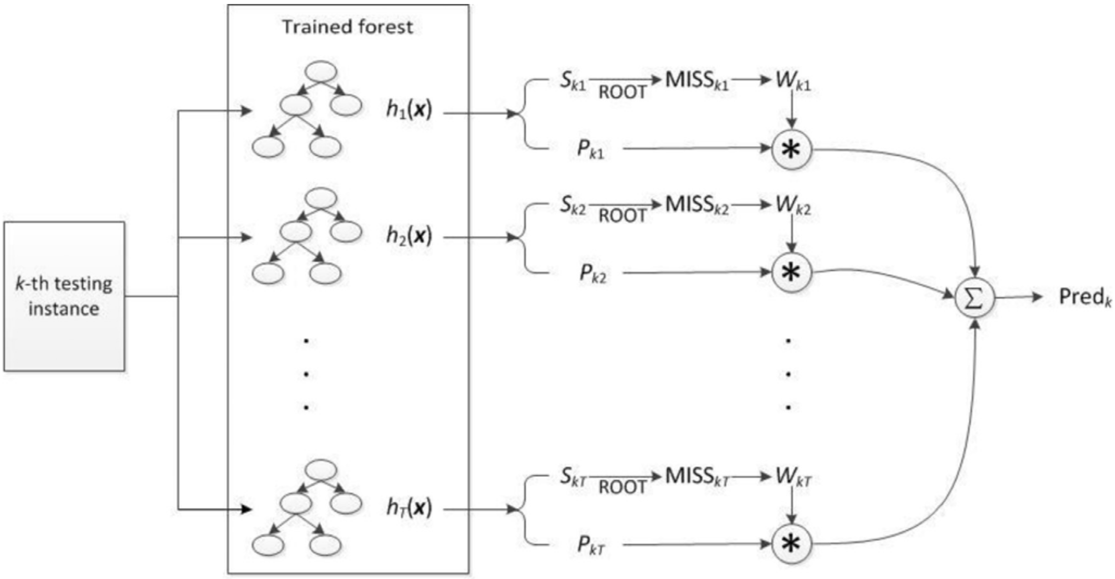

Trained forest
k-th testing instance
$h_1(\mathbf{x})$
$h_2(\mathbf{x})$
$h_T(\mathbf{x})$
S_{k1} ROOT MISS_{k1} W_{k1}
P_{k1}
S_{k2} ROOT MISS_{k2} W_{k2}
P_{k2}
S_{kT} ROOT MISS_{kT} W_{kT}
P_{kT}
∗
Σ
Pred_k

Fig. 2. Flow diagram of AWVRF's testing phase.

the root node and S_{kt} is large, the corresponding MISS_{kt} index is big, representing big influence of missing value on the tth tree. The value of MISS_{kt} is small otherwise. From the above analysis, it can be seen MISS_{kt} is a reasonable index for quantitative evaluation of influence degree of missing data on trees' decisions. The parameter θ is set to be 1 for brevity.

$$\mathrm{MISS}_{kt} = \left(\frac{S_{kt}}{\mathrm{ROOT}}\right)^{\theta} \tag{1}$$

2.1.3. Weight factor computation

Through the formula (2), (1-MISS_{kt}) is normalized resulting in the weight of each tree W_{kt}. If the kth testing instance is complete, all decisions is made by the leaf node of trees and the values of S_{k1},S_{k2},…,S_{kT} is similarly small, resulting in similar weights of trees' decisions. But if some attributes of the kth testing instance is missing, values of S_{kt} differs and eventually the weight vector changes dramatically. More weight is assigned to the trees whose decisions are less influenced by missing attributes and trees that are severely influenced by missing attributes receive low weight.

$$W_{kt} = \frac{1 - \mathrm{MISS}_{kt}}{\sum_{t=1}^{T} (1 - \mathrm{MISS}_{kt})} \tag{2}$$

2.1.4. Integration

The final prediction result is obtained by weighted combination of all trees' votes, as formula (3) illustrates.

$$\mathrm{Pred}_k = \sum_{t=1}^{T} W_{kt} * P_{kt} \tag{3}$$

The testing process of AWVRF is summarized in Fig. 2.

2.2. Experimental design

2.2.1. Datasets

To validate the algorithm, typical benchmark real-world datasets with various dimensions and diverse sample sizes are selected from the UCI data repository, including six complete datasets (liver disorders, diabetes, breast cancer, heart, WDBC and sonar) and four originally incomplete datasets (hepatitis, chronic kidney disease echocardiogram and mammographic masses) [24]. They are frequently cited in literature [12,17–19,29]. Basic characteristics of these datasets are summarized in Table 1.

2.2.2. Generating process of missing values

In order to evaluate the performance on various complete datasets, missing values are artificially generated. As AWVRF is specially designed to handle the prediction process, missing values are only created in testing instances while training instances remain complete. For originally incomplete datasets, training instances with missing values are simply excluded such that RF model is built on complete training instances.

Three missing mechanisms are introduced as follows and the details are described in the reference [37].

(a) Missing completely at random (MCAR): Randomly choose a given number of locations and eliminate the values in these locations. That is, the possibility of missing is not influenced by values of any other attributes or the attribute itself.

(b) Missing at random (MAR): The possibility of missing one attribute is influenced by values of the other attribute. In the study, missing values are only created in one attribute (denoted as x_1) and the missing probability of x_1 in a certain location is computed by dividing the rank of another attribute (denoted as x_2) in the corresponding location by the sum of all ranks of x_2.

(c) Missing not at random (MNAR): It means the probability of missing is related to the value of the attribute itself. In the study, missing values are only created in one attribute and large values of the attribute is eliminated with a certain proportion.

To investigate the effect of the data missing rate on the classification, the values in the involved datasets are randomly removed with a fraction of 10%, 20% and 30%, respectively.

2.2.3. Comparison settings

The proposed AWVRF is compared with mean imputation, LeoFill [38], knnimpute, BPCAfill [39] and the reported RF algorithm with surrogate decision [14,33,35] (named as surrRF) methods using 50 times repeated 5-fold cross validation and the classification results are compared in terms of accuracy and area under the ROC curve (AUC). The mean imputation method means imputing the missing data with the mean value of the missing attribute in the training set. The LeoFill method is an imputation method proposed by Breiman [38] to replicate a testing instance n times (n=number

Table 1
Characteristics of the datasets.

Dataset name	Number of attributes	Number of instances	Number of classes	Mean correlation between attributes	Percentage of missing values
Liver disorders	6	345	2	0.2646	0
Diabetes	8	768	2	0.1717	0
Breast cancer	9	683	2	0.6019	0
Heart	13	297	2	0.1573	0
WDBC	30	569	2	0.3949	0
Sonar	60	208	2	0.2294	0
Hepatitis	19	155	2	/	5.67%
Chronic kidney disease	24	400	2	/	10.98%
Echocardiogram	9	74	2	/	2.83%
Mammographic masses	4	961	2	/	4.16%

/: means the value that cannot be calculated.

of classes). For each replicate, the instance is assumed to be a certain class and the median value of the missing attribute in the training set with the same class is used to replace missing values. Finally, the one receiving the most votes determines the class of the original instance. The knnimpute method replaces the missing values with a weighted mean of the missing attribute of the k nearest neighbor instances. And $k = 10$ is chosen in the research. BPCAfill (Bayesian principle component analysis) is a missing value estimating approach within the framework of Bayes inference [39]. The surrRF algorithm handles missing data with the usage of surrogate decisions and makes the final prediction by majority voting. For imputation based methods (mean imputation, LeoFill, knnimpute and BPCAfill), the RF algorithm is used for classification after imputation step. In the forest algorithms, 100 trees are constructed and the number of randomly selected attributes serving as candidates for splits is equal to the square root of a (i.e. the number of predictor attributes). The maximum number of surrogate splits to retain at each node is set to 10.

In addition, AWVRF algorithm is also compared with surrRF in terms of consuming time and number of splitting.

All experiments are implemented with the software of MATLAB 2015b.

3. Results

3.1. Comparison of prediction precision

3.1.1. Performance on the complete datasets with artificial missing values

All methods are tested on multiple UCI datasets with different proportions and different mechanisms of artificial missing values.

For datasets with values MCAR, results in terms of accuracy using six different methods are summarized in Table 2. The AUC values of all methods are close for the integrated dataset and decline as the missing rate increases. For instance, in the liver disorders dataset, the methods of mean imputation, LeoFill, knnimpute, BPCAfill, surrRF and AWVRF have similar accuracy of 0.7195, 0.7195, 0.7195, 0.7195, 0.7188 and 0.7190, respectively for the integrated dataset. As the missing rate reaches 30%, the values of accuracy drop to 0.6415, 0.6394, 0.6443, 0.6363, 0.6545 and 0.6525, respectively.

Overviewing the results of all methods, AWVRF has the best accuracy compared to other methods on diabetes, breast cancer, heart, WDBC and sonar datasets with 30% values MCAR achieving the largest value of 0.7325 ± 0.0106, 0.9668 ± 0.0039, 0.7963 ± 0.0152, 0.9535 ± 0.0051 and 0.7984 ± 0.0194, respectively. And surrRF gets the largest accuracy on liver disorders dataset.

On the other hand, results in terms of AUC for all methods on six UCI datasets with values MCAR are given in Table 3. The AWVRF algorithm has the largest AUC on heart and WDBC dataset (heart with 30% missing values: 0.8776 ± 0.0123; WDBC with 30% missing data: 0.9886 ± 0.0019), while the surrRF algorithm has the largest value of AUC on liver disorders and diabetes dataset (liver disorders with 30% missing values: 0.6896 ± 0.0215; diabetes with 30% missing data: 0.7900 ± 0.0105). On breast cancer dataset, BPCAfill performs best with the largest AUC of 0.9918 ± 0.0017 and knnimpute on sonar dataset with the largest AUC of 0.9059 ± 0.0155.

In addition, the performance of AWVRF under the missing mechanism of MAR and MNAR is also investigated.

Results of accuracy on datasets with 30% values MAR and MNAR are displayed in Fig. 3. AWVRF has the largest accuracy on liver disorders, WDBC and sonar datasets with 30% values MAR while achieves the largest accuracy on liver disorders, breast cancer, WDBC and sonar datasets with 30% MNAR. As other methods, mean imputation gets the largest accuracy on heart dataset with missing values under MAR and MNAR mechanisms, BPCAfill performs best on diabetes with values MNAR and surrRF is the best on diabetes and breast cancer datasets with values MAR.

From the results of AUC on multiple datasets with 30% missing values under MAR and MNAR mechanisms presented in Fig. 4, the performance of AWVRF is superior to other methods on liver disorders (MAR) dataset, WDBC (MAR and MNAR) datasets and sonar (MAR and MNAR) datasets; surrRF is the best on liver disorders (MNAR) dataset and diabetes (MAR and MNAR) datasets; mean imputation perform best on heart (MAR) dataset, breast cancer (MNAR) dataset and heart (MNAR) dataset; The largest value of AUC on breast cancer (MAR) dataset is achieved by BPCAfill method.

3.1.2. Performance on the originally incomplete datasets

The performances of the AWVRF and other methods on various originally incomplete datasets are compared. In terms of accuracy shown in Table 4, the AWVRF has the largest accuracy on two datasets (hepatitis: 0.8391 ± 0.0148; chronic kidney disease: 0.9829 ± 0.0048), while LeoFill gets the largest accuracy on mammographic masses dataset (0.7918 ± 0.0062) and knnimpute on echocardiogram dataset (0.7231 ± 0.0355).

As given in Table 5, the algorithms of BPCAfill exhibits the superior performance on chronic kidney and mammographic masses with the largest AUC value of 1.0000 ± 0.0001 and 0.8411 ± 0.0044, while knnimpute has the largest AUC of 0.8799 ± 0.0198 on hepatitis dataset and mean imputation achieves the largest AUC of 0.7754 ± 0.0363 on echocardiogram dataset.

3.2. Comparison of computation efficiency with surrRF

The computing efficiencies of the surrRF and AWVRF are compared regarding the consuming time and number of splitting for each decision process (running on the personal computer with

Table 2
Results of accuracy on different datasets with various missing rate (MCAR mechanism).

Dataset and missing rate	Accuracy (mean ± standard deviation from 50 experiments)					
	Mean imputation	LeoFill	knnimpute	BPCAfill	surrRF	AWVRF
Dataset 1: Liver disorders						
0	0.7195 ± 0.0192	0.7195 ± 0.0192	0.7195 ± 0.0192	0.7195 ± 0.0192	0.7188 ± 0.0192	0.7190 ± 0.0189
10%	0.6908 ± 0.0195	0.6943 ± 0.0238	0.6927 ± 0.0189	0.6931 ± 0.0193	**0.6962 ± 0.0185**	0.6951 ± 0.0184
20%	0.6677 ± 0.0229	0.6678 ± 0.0204	0.6722 ± 0.0200	0.6688 ± 0.0205	**0.6827 ± 0.0202**	0.6794 ± 0.0187
30%	0.6415 ± 0.0203	0.6394 ± 0.0199	0.6443 ± 0.0219	0.6363 ± 0.0216	**0.6545 ± 0.0190**	0.6525 ± 0.0213
Dataset 2: Diabetes						
0	0.7605 ± 0.0077	0.7605 ± 0.0077	0.7605 ± 0.0077	0.7605 ± 0.0077	0.7603 ± 0.0077	0.7601 ± 0.0073
10%	0.7488 ± 0.0108	0.7506 ± 0.0098	0.7510 ± 0.0085	0.7493 ± 0.0105	0.7515 ± 0.0103	**0.7516 ± 0.0091**
20%	0.7342 ± 0.0097	0.7394 ± 0.0100	**0.7421 ± 0.0099**	0.7354 ± 0.0102	0.7416 ± 0.0109	0.7407 ± 0.0100
30%	0.7186 ± 0.0130	0.7285 ± 0.0128	0.7283 ± 0.0118	0.7211 ± 0.0121	0.7312 ± 0.0115	**0.7325 ± 0.0106**
Dataset 3: Breast cancer						
0	0.9720 ± 0.0027	0.9720 ± 0.0027	0.9720 ± 0.0027	0.9720 ± 0.0027	0.9720 ± 0.0025	0.9709 ± 0.0029
10%	0.9681 ± 0.0041	**0.9707 ± 0.0039**	0.9703 ± 0.0032	0.9698 ± 0.0034	0.9706 ± 0.0034	0.9704 ± 0.0037
20%	0.9629 ± 0.0046	0.9660 ± 0.0037	0.9671 ± 0.0037	0.9664 ± 0.0041	0.9670 ± 0.0041	**0.9676 ± 0.0036**
30%	0.9562 ± 0.0061	0.9645 ± 0.0041	0.9661 ± 0.0047	0.9641 ± 0.0045	0.9655 ± 0.0044	**0.9668 ± 0.0039**
Dataset 4: Heart						
0	0.8160 ± 0.0121	0.8160 ± 0.0121	0.8160 ± 0.0121	0.8160 ± 0.0121	0.8168 ± 0.0116	0.8143 ± 0.0112
10%	0.8110 ± 0.0150	0.8126 ± 0.0148	0.8090 ± 0.0139	0.8091 ± 0.0150	0.8146 ± 0.0133	**0.8149 ± 0.0123**
20%	0.7879 ± 0.0179	0.7984 ± 0.0177	0.7888 ± 0.0183	0.7895 ± 0.0157	**0.8060 ± 0.0145**	0.8049 ± 0.0158
30%	0.7694 ± 0.0205	0.7877 ± 0.0142	0.7693 ± 0.0162	0.7683 ± 0.0200	0.7960 ± 0.0150	**0.7963 ± 0.0152**
Dataset 5: WDBC						
0	0.9600 ± 0.0040	0.9600 ± 0.0040	0.9600 ± 0.0040	0.9600 ± 0.0040	0.9599 ± 0.0037	0.9624 ± 0.0042
10%	0.9568 ± 0.0046	0.9589 ± 0.0045	0.9583 ± 0.0036	0.9525 ± 0.0047	0.9576 ± 0.0041	**0.9602 ± 0.0045**
20%	0.9530 ± 0.0050	0.9571 ± 0.0045	0.9536 ± 0.0048	0.9453 ± 0.0057	0.9547 ± 0.0050	**0.9568 ± 0.0048**
30%	0.9462 ± 0.0067	0.9527 ± 0.0048	0.9469 ± 0.0049	0.9370 ± 0.0054	0.9520 ± 0.0042	**0.9535 ± 0.0051**
Dataset 6: Sonar						
0	0.8244 ± 0.0181	0.8244 ± 0.0181	0.8244 ± 0.0181	0.8244 ± 0.0181	0.8252 ± 0.0184	0.8315 ± 0.0182
10%	0.8043 ± 0.0195	0.8114 ± 0.0199	0.8127 ± 0.0187	0.8053 ± 0.0211	0.8120 ± 0.0186	**0.8214 ± 0.0165**
20%	0.7840 ± 0.0212	0.7997 ± 0.0204	0.8054 ± 0.0212	0.7873 ± 0.0176	0.8038 ± 0.0207	**0.8119 ± 0.0217**
30%	0.7606 ± 0.0210	0.7919 ± 0.0222	0.7940 ± 0.0217	0.7673 ± 0.0201	0.7933 ± 0.0176	**0.7984 ± 0.0194**

The best values are highlighted in bold type in each dataset with 10%, 20% or 30% of missing values.

Table 3
Results of AUC on different datasets with acritical missing values (MCAR mechanism).

Dataset and missing rate	AUC (mean ± standard deviation from 50 experiments)					
	Mean imputation	LeoFill	knnimpute	BPCAfill	surrRF	AWVRF
Dataset 1: Liver disorders						
0	0.7609 ± 0.0139	0.7609 ± 0.0139	0.7609 ± 0.0139	0.7609 ± 0.0139	0.7609 ± 0.0137	0.7602 ± 0.0139
10%	0.7206 ± 0.0198	0.7244 ± 0.0191	0.7265 ± 0.0193	0.7257 ± 0.0184	**0.7359 ± 0.0185**	0.7344 ± 0.0179
20%	0.6901 ± 0.0265	0.6954 ± 0.0209	0.7037 ± 0.0212	0.6987 ± 0.0226	**0.7206 ± 0.0193**	0.7175 ± 0.0189
30%	0.6536 ± 0.0241	0.6591 ± 0.0255	0.6675 ± 0.0252	0.6600 ± 0.0242	**0.6896 ± 0.0215**	0.6839 ± 0.0221
Dataset 2: Diabetes						
0	0.8226 ± 0.0057	0.8226 ± 0.0057	0.8226 ± 0.0057	0.8226 ± 0.0057	0.8225 ± 0.0057	0.8216 ± 0.0058
10%	0.8102 ± 0.0087	0.8101 ± 0.0083	0.8130 ± 0.0072	0.8101 ± 0.0084	0.8149 ± 0.0076	**0.8152 ± 0.0077**
20%	0.7921 ± 0.0075	0.7931 ± 0.0091	0.8008 ± 0.0078	0.7935 ± 0.0079	**0.8030 ± 0.0082**	0.8021 ± 0.0095
30%	0.7739 ± 0.0131	0.7757 ± 0.0123	0.7849 ± 0.0120	0.7773 ± 0.0126	**0.7900 ± 0.0105**	0.7898 ± 0.0108
Dataset 3: Breast cancer						
0	0.9928 ± 0.0010	0.9928 ± 0.0010	0.9928 ± 0.0010	0.9928 ± 0.0010	0.9928 ± 0.0010	0.9930 ± 0.0008
10%	0.9918 ± 0.0013	0.9918 ± 0.0015	0.9923 ± 0.0014	**0.9926 ± 0.0014**	0.9925 ± 0.0014	0.9923 ± 0.0015
20%	0.9905 ± 0.0013	0.9899 ± 0.0022	0.9914 ± 0.0019	**0.9919 ± 0.0014**	0.9918 ± 0.0014	0.9915 ± 0.0017
30%	0.9887 ± 0.0019	0.9879 ± 0.0028	0.9905 ± 0.0019	**0.9918 ± 0.0017**	0.9913 ± 0.0016	0.9912 ± 0.0017
Dataset 4: Heart						
0	0.8966 ± 0.0092	0.8966 ± 0.0092	0.8966 ± 0.0092	0.8966 ± 0.0092	0.8967 ± 0.0092	0.8954 ± 0.0096
10%	0.8872 ± 0.0104	0.8870 ± 0.0112	0.8863 ± 0.0110	0.8860 ± 0.0106	0.8929 ± 0.0095	**0.8930 ± 0.0100**
20%	0.8673 ± 0.0120	0.8715 ± 0.0125	0.8692 ± 0.0101	0.8701 ± 0.0103	0.8853 ± 0.0101	**0.8856 ± 0.0101**
30%	0.8538 ± 0.0144	0.8567 ± 0.0134	0.8513 ± 0.0134	0.8529 ± 0.0165	0.8762 ± 0.0124	**0.8776 ± 0.0123**
Dataset 5: WDBC						
0	0.9900 ± 0.0013	0.9900 ± 0.0013	0.9900 ± 0.0013	0.9900 ± 0.0013	0.9900 ± 0.0013	0.9902 ± 0.0015
10%	0.9875 ± 0.0018	0.9885 ± 0.0018	0.9886 ± 0.0018	0.9863 ± 0.0019	0.9890 ± 0.0014	**0.9892 ± 0.0016**
20%	0.9862 ± 0.0025	0.9882 ± 0.0017	0.9884 ± 0.0015	0.9846 ± 0.0016	0.9889 ± 0.0013	**0.9893 ± 0.0016**
30%	0.9847 ± 0.0032	0.9863 ± 0.0022	0.9869 ± 0.0019	0.9818 ± 0.0026	0.9882 ± 0.0018	**0.9886 ± 0.0019**
Dataset 6: Sonar						
0	0.9238 ± 0.0111	0.9238 ± 0.0111	0.9238 ± 0.0111	0.9238 ± 0.0111	0.9237 ± 0.0112	0.9277 ± 0.0111
10%	0.9139 ± 0.0129	0.9161 ± 0.0122	0.9207 ± 0.0106	0.9150 ± 0.0129	0.9181 ± 0.0116	**0.9215 ± 0.0117**
20%	0.9023 ± 0.0116	0.9035 ± 0.0104	**0.9137 ± 0.0121**	0.9004 ± 0.0120	0.9078 ± 0.0119	0.9117 ± 0.0125
30%	0.8862 ± 0.0142	0.8905 ± 0.0161	**0.9059 ± 0.0155**	0.8831 ± 0.0165	0.9000 ± 0.0135	0.9020 ± 0.0141

The best values are highlighted in bold type in each dataset with 10%,20% or 30% of missing values.

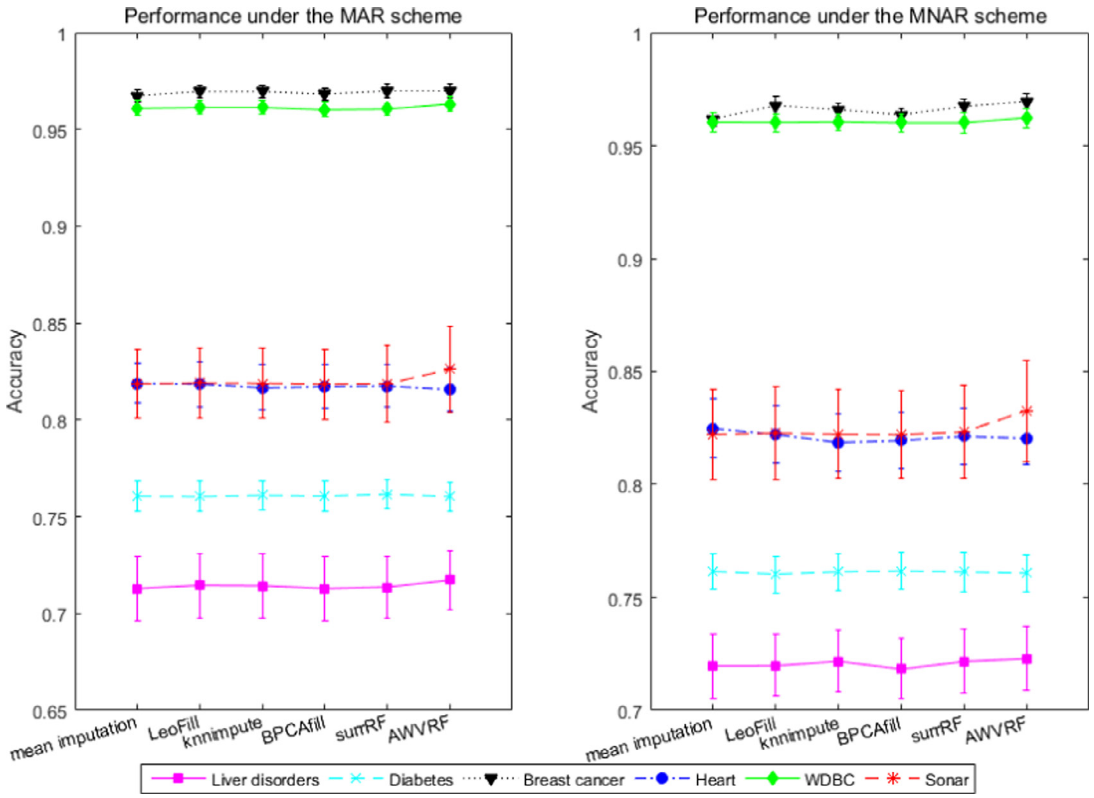


Fig. 3. Accuracy on six datasets with 30% missing values under MAR and MNAR mechanisms.

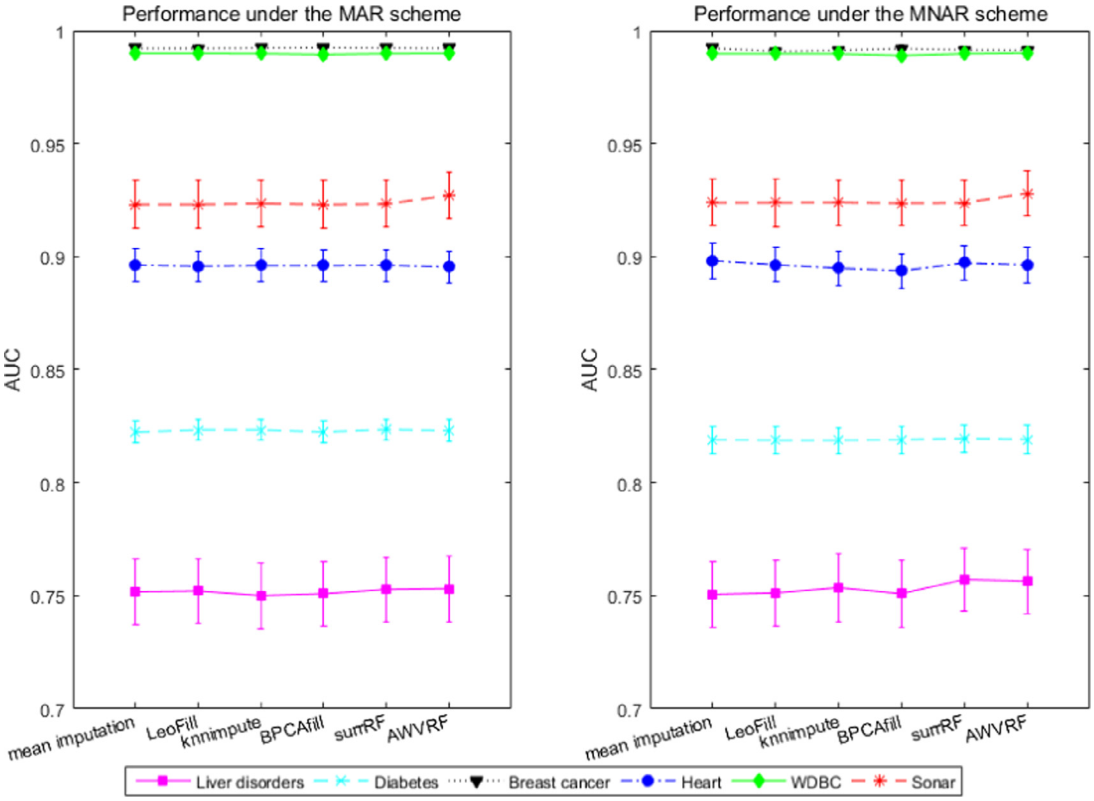


Fig. 4. AUC on six datasets with 30% missing values under MAR and MNAR mechanisms.

Table 4
Results of accuracy on four originally incomplete datasets.

Dataset	Accuracy (mean ± standard deviation from 50 experiments)					
	Mean imputation	LeoFill	knnimpute	BPCAfill	surrRF	AWVRF
Hepatitis	0.8239 ± 0.0143	0.8357 ± 0.0152	0.8236 ± 0.0133	0.8213 ± 0.0153	0.8329 ± 0.0168	**0.8391 ± 0.0148**
Chronic kidney disease	0.8885 ± 0.0106	0.9347 ± 0.0071	0.8948 ± 0.0084	0.9191 ± 0.0073	0.9172 ± 0.0073	**0.9829 ± 0.0048**
Echocardiogram	0.7196 ± 0.0348	0.7136 ± 0.0334	**0.7231 ± 0.0355**	0.7201 ± 0.0319	0.7147 ± 0.0333	0.7193 ± 0.0415
Mammographic masses	0.7880 ± 0.0067	**0.7918 ± 0.0062**	0.7888 ± 0.0068	0.7891 ± 0.0064	0.7869 ± 0.0060	0.7857 ± 0.0056

The best values in each dataset are highlighted in bold type.

Table 5
Results of AUC on four originally incomplete datasets.

Dataset	AUC (mean ± standard deviation from 50 experiments)					
	Mean imputation	LeoFill	knnimpute	BPCAfill	surrRF	AWVRF
Hepatitis	0.8650 ± 0.0219	0.8566 ± 0.0231	**0.8799 ± 0.0198**	0.8673 ± 0.0220	0.8741 ± 0.0219	0.8746 ± 0.0230
Chronic kidney disease	0.9979 ± 0.0009	0.9998 ± 0.0005	0.9988 ± 0.0007	**1.0000 ± 0.0001**	0.9999 ± 0.0004	0.9999 ± 0.0005
Echocardiogram	**0.7754 ± 0.0363**	0.7519 ± 0.0404	0.7592 ± 0.0399	0.7703 ± 0.0374	0.7656 ± 0.0371	0.7635 ± 0.0395
Mammographic masses	0.84106 ± 0.0043	0.8380 ± 0.0045	0.8394 ± 0.0046	**0.84114 ± 0.0044**	0.8386 ± 0.0042	0.8385 ± 0.0044

The best values in each dataset are highlighted in bold type.

Table 6
Time of every prediction process.

Dataset	Time/s (mean ± standard deviation)	
	surrRF	AWVRF
*Liver disorders	1.046 ± 0.151	0.294 ± 0.050
*Diabetes	1.415 ± 0.202	0.318 ± 0.051
*Breast cancer	1.087 ± 0.751	0.190 ± 0.017
*Heart	0.650 ± 0.495	0.189 ± 0.011
*WDBC	0.662 ± 0.160	0.222 ± 0.062
*Sonar	1.103 ± 0.270	0.360 ± 0.110

*mark the difference is significant (all *p*-values for the paired *t*-test are less than 0.001)

Table 7
Number of splits in every prediction process.

Dataset	Number of splits (mean ± standard deviation)	
	surrRF	AWVRF
*Liver disorders	46,246.4 ± 1444.1	37,716.2 ± 5671.1
*Diabetes	116,534.0 ± 3199.9	91,017.6 ± 16,704.2
*Breast cancer	57,706.8 ± 1793.6	50,347.0 ± 5961.5
*Heart	31,546.0 ± 745.4	26,609.5 ± 3589.3
*WDBC	49,964.2 ± 2291.9	45,006.5 ± 4162.2
*Sonar	19,846.8 ± 447.0	18,464.5 ± 984.9

*mark the difference is significant (all *p*-values for the paired *t*-test are less than 0.001)

3.0 GHz CPU and 8GB RAM). As summarized in the Table 6 and Table 7, significant differences are shown between the methods ($p < 0.001$). The consuming time for each decision process in the AWVRF varies between 0.189–0.360 s, while in the surrRF it ranges 0.65–1.415 s, about 2.0–4.7 folds higher than the former. In AWVRF, average number of splits for each decision process is 18,464.5–91,017.6, while in the surrRF it ranges 19,846.8–116,534.0, about 1.1–1.7 times as the former.

4. Discussion

4.1. Validation of the proposed AWVRF algorithm

In a total of 22 settings on 10 datasets (6 complete dataset with 3 missing mechanisms and 4 originally incomplete datasets), the methods of AWVRF and surrRF harvest the highest accuracy in 14 and 3 settings, respectively while the other 4 imputation methods achieve the highest in the rest 5 settings. Similarly, AWVRF and surrRF also lead to the largest AUC in most cases (AWVRF: 7 settings; surrRF: 5 settings). Surrogate based methods (surrRF and AWVRF) exhibit the best performance in most settings, indicating their superiority over imputation methods. Moreover, compared with surrRF, AWVRF has comparable values of AUC but performs much better in terms of accuracy.

Surrogate based methods avoid the process of imputation and work in any data situation with no restriction for data distribution. These methods make use of all the available data to handle each case individually, providing a more accurate analysis in many cases [35,36].

The characteristics of dataset have impacts on the performance of the algorithms. Among them the correlation among the predictor attributes is an important one. The imputation methods estimate the missing values by the existing data in the dataset. When the attributes in the dataset are highly correlated, the estimation of missing values is accurate and consequently a classification is achieved with high precision. Thus, it is not surprised that the imputation methods performed well in high correlated datasets, such as breast cancer dataset (mean correlation: 0.6019). However, when the correlation is low, such as diabetes dataset (mean correlation: 0.1717) and heart dataset (mean correlation: 0.1573), the imputation methods are prone to produce biased estimates resulting in poor classification. In contrast, AWVRF algorithm does not utilize the imputation strategy and depends less on the attributes' correlation. Generally speaking, compared with the imputation based methods, AWVRF is robust with less dependence on the correlation among the predictor attributions.

4.2. Computing efficiency of AWVRF

AWVRF and surrRF have the similarity that they do not estimate the missing values from the existing data but go through the relevant nodes by taking the surrogate attributes into consideration, however they behave differently in quitting the vote and weighting the output of the trees. The surrRF algorithm outputs the class of the leaf node and uses equal weight for each trees vote, while AWVRF allows to exit at internal node and uses adjusted different weight for different trees votes. In surrRF, the prediction process conveys a complete path with large number of splitting while in AWVRF, a testing instance undergoes a shorter path with smaller number of splitting. Regarding the similarity and difference in AWVRF and surrRF, it is easy to understand that the both algorithms achieved comparable performances in all test datasets, but the computing efficiency of AWVRF is remarkably improved by shrinking the steps of splitting and such reduction does not weaken the accuracy of the prediction.

4.3. Parameter sensitivity analysis

In the designing of AWVRF algorithm,θin formula (1) is a free parameter. Sensitivity analysis of the parameter$\theta$is conducted in order to evaluate the effect of parameter value change on AWVRF algorithm. The value of θ is set to be 0.5,1 and 2 for comparison. Experiments results show the performances of the three settings are close, so the simplest way is employed, that isletθ be 1.

In the formula (1), only the training instance size of the "decision" node, namely S_{kt}, is taken into consideration. Nevertheless, not only the training instance size but also depth of the decision node, or others, is likely to have an impact on the distance between the "decision" node and the root node. Future work needs to explore more effective and elaborated settings of MISS$_{kt}$ in order to improve the overall performance.

4.4. Limitation of AWVRF

The present AWVRF is only designed for prediction process to handle missing values in the testing data and requires the training data to be complete, which limits the application of AWVRF. Missing values in the training data is not considered in the current investigation and needs to be dealt with in the future work. In addition, AWVRF is only able to solve binary classification problems presently and should extend to multi-class, even regression cases.

5. Conclusion

This paper proposed a novel kind of algorithm in random forests to address the incomplete data in classification. It avoids the process of imputation but exits at the internal node in case missing values are encountered and produces the weighted voting. The experimental results on various datasets show that the algorithm is a prominent method to solve the classification problem on incomplete data.

Conflict of interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant 81271662), the Ministry of the Science and Technology of China (Grant 2014DFT3050), the Natural Science Foundation of Zhejiang Province (Grant LQ14H180001) and the Department of Science and Technology of Zhejiang Province (Grant 2011R50018).

References

[1] P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J.A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, Machine learning in bioinformatics, Briefings Bioinf. 7 (2006) 86–112.
[2] P.R. Harper, A review and comparison of classification algorithms for medical decision making, Health Policy 71 (2005) 315–331.
[3] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (2002) 1–47.
[4] S.G. Kong, J. Heo, B.R. Abidi, J. Paik, M.A. Abidi, Recent advances in visual and infrared face recognition—a review, Comput. Vision Image Understanding 97 (2005) 103–135.
[5] X. Fu, Y. Ren, G. Yang, Q. Pan, S. Gong, L. Li, J. Yan, G. Ning, A computational model for heart failure stratification, in: Computing in Cardiology, 2011, IEEE, 2011, pp. 385–388.
[6] A.S. Fialho, U. Kaymak, R.J. Almeida, F. Cismondi, S.M. Vieira, S.R. Reti, J.M. Sousa, S.N. Finkelstein, Probabilistic fuzzy prediction of mortality in intensive care units, in: Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on, IEEE, 2012, pp. 1–8.
[7] D.R. Cutler, T.C. Edwards Jr, K.H. Beard, A. Cutler, K.T. Hess, J. Gibson, J.J. Lawler, Random forests for classification in ecology, Ecology 88 (2007) 2783–2792.
[8] R. Díaz-Uriarte, S.A. De Andres, Gene selection and classification of microarray data using random forest, BMC Bioinf. 7 (2006) 1.
[9] A. Verikas, A. Gelzinis, M. Bacauskiene, Mining data with random forests: A survey and results of new tests, Pattern Recognit. 44 (2011) 330–349.
[10] X. Chen, H. Ishwaran, Random forests for genomic data analysis, Genomics 99 (2012) 323–329.
[11] L. Rokach, Decision forest: twenty years of research, Inf. Fusion 27 (2016) 111–125.
[12] L. Breiman, Random forests, in: Machine Learning, Kluwer Academic Publishers, 2001, pp. 5–32.
[13] A. Hapfelmeier, T. Hothorn, K. Ulm, C. Strobl, A new variable importance measure for random forests with missing data, Stat. Comput. 24 (2014) 21–34.
[14] L. Breiman, J.H. Friedman, R. Olshen, C.J. Stone, in: Classification and Regression Trees, vol. 81, Wadsworth, Systat Statistics ® Spss Inc. United States of America, 1984, pp. 17–23.
[15] G. Biau, Analysis of a random forests model, J. Mach. Learn. Res. 13 (2012) 1063–1095.
[16] S. Bernard, L. Heutte, S. Adam, Influence of hyperparameters on random forest accuracy, in: Multiple Classifier Systems, Springer, 2009, pp. 171–180.
[17] M. Robnik-Šikonja, Improving random forests, in: Machine Learning: ECML 2004, Springer, 2004, pp. 359–370.
[18] H. Kim, H. Kim, H. Moon, H. Ahn, A weight-adjusted voting algorithm for ensembles of classifiers, J. Korean Stat. Soc. 40 (2011) 437–449.
[19] H.B. Li, W. Wang, H.W. Ding, J. Dong, Trees weighting random forest method for classifying high-dimensional noisy data, e-Business Engineering (ICEBE), in: 2010 IEEE 7th International Conference on, IEEE, 2010, pp. 160–163.
[20] D.J. Stekhoven, P. Bühlmann, MissForest—non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (2012) 112–118.
[21] A. Karahalios, L. Baglietto, J.B. Carlin, D.R. English, J.A. Simpson, A review of the reporting and handling of missing data in cohort studies with repeated assessment of exposure measures, BMC Med. Res. Methodol. 12 (2012) 96.
[22] A.N. Baraldi, C.K. Enders, An introduction to modern missing data analyses, J. School Psychol. 48 (2010) 5–37.
[23] P.J. García-Laencina, J.-L. Sancho-Gómez, A.R. Figueiras-Vidal, Pattern classification with missing data: a review, Neural Comput. Appl. 19 (2010) 263–282.
[24] M. Lichman, UCI Machine Learning Repository, 2013. available online http://archive.ics.uci.edu/ml.
[25] E.M. Hernández-Pereira, D. Álvarez-Estévez, V. Moret-Bonillo, Automatic classification of respiratory patterns involving missing data imputation techniques, Biosyst. Eng. 138 (2015) 65–76.
[26] P.H. Abreu, H. Amaro, D.C. Silva, P. Machado, M.H. Abreu, N. Afonso, A. Dourado, Overall survival prediction for women breast cancer using ensemble methods and incomplete clinical data, in: XIII Mediterranean Conference on Medical and Biological Engineering and Computing 2013, Springer, 2014, pp. 1366–1369.
[27] R. Polikar, J. DePasquale, H.S. Mohammed, G. Brown, L.I. Kuncheva, Learn++. MF: a random subspace approach for the missing feature problem, Pattern Recognit. 43 (2010) 3817–3832.
[28] B. Conroy, L. Eshelman, C. Potes, M. Xu-Wilson, A dynamic ensemble approach to robust classification in the presence of missing data, Mach. Learn. (2015) 1–21.
[29] L. Nanni, A. Lumini, S. Brahnam, A classifier ensemble approach for the missing feature problem, Artif. Intell. Med. 55 (2012) 37–50.
[30] A.D. Shah, J.W. Bartlett, J. Carpenter, O. Nicholas, H. Hemingway, Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study, Am. J. Epidemiol. (2014), doi:10.1093/aje/kwt312.
[31] M. Ghannad-Rezaie, H. Soltanian-Zadeh, H. Ying, M. Dong, Selection–fusion approach for classification of datasets with missing values, Pattern Recognit. 43 (2010) 2340–2350.
[32] P. Juszczak, R.P. Duin, Combining one-class classifiers to classify missing data, in: Multiple Classifier Systems, Springer, 2004, pp. 92–101.
[33] H.C. Valdiviezo, S. Van Aelst, Tree-based prediction on incomplete data using imputation or surrogate decisions, Inf. Sci. 311 (2015) 163–181.
[34] A. Feelders, Handling missing data in trees: surrogate splits or statistical imputation, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 1999, pp. 329–334.
[35] A. Hapfelmeier, T. Hothorn, K. Ulm, Recursive partitioning on incomplete data using surrogate decisions and multiple imputation, Comput. Stat. Data Anal. 56 (2012) 1552–1565.
[36] B. Twala, An empirical comparison of techniques for handling incomplete data using decision trees, Appl. Artif. Intell. 23 (2009) 373–405.
[37] A. Rieger, T. Hothorn, C. Strobl, Random forests with missing values in the covariates, 2010. http://epub.ub.uni-muenchen.de/11481.
[38] L. Breiman, A. Cutler, Random forests, 2008. available online http://www.stat.berkeley.edu/~breiman/RandomForests/.
[39] S. Oba, M.A. Sato, I. Takemasa, M. Monden, K. Matsubara, S. Ishii, A Bayesian missing value estimation method for gene expression profile data, Bioinformatics 19 (2003) 2088–2096.