Multidim Syst Sign Process (2017) 28:1309–1324
DOI 10.1007/s11045-016-0393-4

A subspace ensemble framework for classification with high dimensional missing data

Hang Gao[1] · **Songlei Jian**[1] · **Yuxing Peng**[1] · **Xinwang Liu**[1]

Received: 13 October 2015 / Revised: 27 February 2016 / Accepted: 3 March 2016 /
Published online: 31 March 2016

Abstract Real world classification tasks may involve high dimensional missing data. The traditional approach to handling the missing data is to impute the data first, and then apply the traditional classification algorithms on the imputed data. This method first assumes that there exist a distribution or feature relations among the data, and then estimates missing items with existing observed values. A reasonable assumption is a necessary guarantee for accurate imputation. The distribution or feature relations of data, however, is often complex or even impossible to be captured in high dimensional data sets, leading to inaccurate imputation. In this paper, we propose a complete-case projection subspace ensemble framework, where two alternative partition strategies, namely bootstrap subspace partition and missing pattern-sensitive subspace partition, are developed for incomplete datasets with even missing patterns and uneven missing patterns, respectively. Multiple component classifiers are then separately trained in these subspaces. After that, a final ensemble classifier is constructed by a weighted majority vote of component classifiers. In the experiments, we demonstrate the effectiveness of the proposed framework over eight high dimensional UCI datasets. Meanwhile, we apply the two proposed partition strategies over data sets with different missing patterns. As indicated, the proposed algorithm significantly outperforms existing imputation methods in most cases.

Keywords High dimensional data · Missing data · Subspace ensemble · Extreme learning machine

✉ Hang Gao
hanggao1821@163.com; hanggao72@gmail.com

Songlei Jian
jiansonglei@nudt.edu.cn

Yuxing Peng
pengyuxing@aliyun.com

Xinwang Liu
xinwangliu@nudt.edu.cn

[1] Science and Technology on Parallel and Distributed Processing Laboratory College of Computer, National University of Defense Technology, Changsha 410073, People's Republic of China

1 Introduction

Missing data are an extremely common phenomenon in machine learning (Benjamin 2008). If any feature value of any sample is unobserved, we call it incomplete sample (or sample with missing value). It may happen in data collecting, data transmission or data integration. Traditional treatments for missing data are developed in statistical analysis (Little and Rubin 2014). Imputation is the most widely used method. It speculates missing value based on observed value (or existing value). There are several kinds of imputations. Simple imputation such as mean or mode imputation (MI) preserves mean value of variables. But it underestimates variance (Donders et al. 2006). A more precise method is k nearest neighbor imputation (KNNI). KNNI substitutes missing value with mean or mode of K nearest neighbors (Batista and Monard 2002). It believes that nearest samples share similar value in all features (dimensions). However, when partial distances (i.e. distances measured in feature subspaces) is not proportional to distances in whole feature space, KNNI is inaccurate. More advanced imputation methods depend on model assumption. For example, maximum likelihood imputation assumes a data distribution and estimates missing value by most probable value (Enders 2001). Regression imputation assumes a linear or non linear model between features (i.e. relation of feature), then it calculates model parameters by learning methods (Donders et al. 2006). They work well when model assumptions are reasonable and accurate. However, in many cases, data distributions and relations of features are always complex. It is impossible to assume an accurate feature relation or data distribution (Graham et al. 2007).

Generally, standard classification algorithms are applied on complete data sets. For classification with missing data, researchers often share experience in statistical analysis. However, it should be noted that the purpose of imputation is to keep accuracy of statistical indicators with missing data (Scheffer 2002). While the purpose of classification is to get an accurate classification model and prediction. Moreover, an inaccurate imputation introduces much biased value. Consequently, it has side effect on classification. Therefore, when an accurate imputation is not guaranteed, or observed values (already-known information) is enough for model training, we should focus on how to training with observed value rather than estimating missing value.

Complete-case learning (CCL), which is the simplest and most efficient missing data treatment, is practical and effective in many real cases (Batista and Monard 2003). CCL marginalizes (omits or deletes) incomplete samples during model learning. Different from imputation, CCL does not estimate missing value and no bias or error is introduced. However, there are two drawbacks in CCL. First, some observed values of incomplete samples are marginalized as well as missing value. Possibly they are useful information for classification. It is a good choice when missing values are not important and marginalized information is relatively small. For high dimensional data, the amount of marginalized values is much larger and this kind of lost may be even more serious and unacceptable. Second, CCL does not work for the situation that missing value exists in testing samples. In other words, a classification model trained in full feature space can not classify incomplete samples.

In this work, we propose an extreme learning machine (ELM) (Huang 2015) based complete-case projection subspace (CPS) ensemble framework for classification with high dimensional missing data. Meanwhile, two subspace partition strategies are developed. Inspired by bagging, we invent an overlapped subspace partition for incomplete datasets with uniform missing patterns. Considering that dissimilar missing pattern of different features, we invent a missing pattern-sensitive partition strategy for incomplete datasets with non-uniform missing patterns.

CPS learns from feature values of incomplete samples as well as complete samples. Additionally, it manages to classify incomplete samples in subspaces. It constructs several diverse models by training data in different feature subspaces and then integrates them by weighted majority vote. For ensemble learning, diversity is a good guarantee for learning accuracy. CPS achieves its diversity by different feature subspaces and complete-case projection. In this work, we choose ELM as the component classifier (actually, most existing classification algorithms can be used by the same way). Two partition strategies are verified in uniform missing patterns and non-uniform missing patterns respectively.

The main contributions of this work are: (1) We develop a CPS ensemble classification framework for high dimensional missing data. (2) We design an bootstrap subspace partition strategy for incomplete datasets with uniform missing patterns. (3) We invent a missing pattern-sensitive subspace partition strategy for incomplete datasets with non-uniform missing patterns. The remainder of this paper is organized as follows. Section 2 presents a brief review of two classical ensemble methods. Section 3 contains detailed description of CSP ensemble framework and two subspace partition strategies. Section 4 discusses experimental results. Section 5 gives conclusion.

2 Preliminary

2.1 Extreme learning machine

Extreme learning machine (ELM) is an unifying learning algorithm which can be used for several learning tasks including classification. It was originally developed for the single-hidden-layer feed forward neural networks (SLFNs), and then extended to the generalized SLFNs (Huang et al. 2012). ELM has already been used in many real tasks such as biomedical engineering (Huang et al. 2015), image 3D shape segmentation (Xie et al. 2014), and been extended to many kind of learning tasks (Li and Mao 2016). Particularly, ELM is well suitable for some high dimensional tasks (Cao and Lin 2015) and been integrated in some ensemble methods (Cao et al. 2012).

Given training set consisting of N training samples : $\{(\boldsymbol{x_j}, \boldsymbol{t_j})|\boldsymbol{x_j} \in R^n, \boldsymbol{t_j} \in R^m, j = 1, 2, \ldots, N\}$, where $\boldsymbol{x_j}$ is a n-dimensional feature vector and $\boldsymbol{t_j}$ is a m-dimensional label vector. The output of ELM is formulated as Eq. (1)

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{\beta} = [\beta_1,\ldots, \beta_L]$ is the vector of the weights between the hidden layer of L nodes to the output nodes, and $\mathbf{h}(.)$ represents the vector of the activation function. $h_i(\boldsymbol{x}) = \boldsymbol{\omega_i}.\mathbf{x} + b_i$, $\boldsymbol{\omega_i}$ and b_i are random parameters. In training phase, ELM approximates these N samples with zero error means that there exist $\boldsymbol{\beta}$ such that $\boldsymbol{t_j} = f_L(\boldsymbol{x_j}) = \sum_{i=1}^{L} \beta_i h(\boldsymbol{\omega_i}, b_i, \boldsymbol{x_j})$, which can be equally formulated as Eq. (2),

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} h(\omega_1, b_1, x_1) & \cdots & h(\omega_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ h(\omega_1, b_1, x_N) & \cdots & h(\omega_L, b_L, x_N) \end{bmatrix}_{N*L}$$

ELM aims to minimize the training error $||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||$ and the norm of weights $||\boldsymbol{\beta}||$, the smallest norm least-squares solution of the above linear system is

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{3}$$

Compared with other learning algorithms, the most significant advantage of ELM is efficiency. ELM randomly generates parameters $\boldsymbol{\omega}$ and $\boldsymbol{b}$ for hidden nodes rather than iteratively adjusting network parameters. Meanwhile, with the objective of reaching the smallest training error and the smallest norm of output weights, ELM achieves accurate learning as well as good generalization. All these characteristics make ELM a good candidate for component classifier in our subspace ensemble framework.

2.2 Ensemble learning

In this section, we give a brief introduction of two classical ensemble methods, i.e. random subspace ensemble and bagging ensemble, which are just the inspirations of our work.

Subspace ensemble (SE) consists of several classifiers each operating in a subspace of the original feature space. Random subspace ensemble (RSE) is the simplest and classical RS. It partitions feature space randomly and trains component classifier on each feature subspace. Component classifiers trained in different subspaces leads to diversity and final prediction is based on the outputs of all component classifiers (Bryll et al. 2003). Random subspace method has already successfully been used for classical learning algorithms, e.g. extreme learning machine (Huang et al. 2014), support vector machines (Bertoni et al. 2005), nearest neighbors (Ho 1998), tree-based algorithms (Banfield et al. 2007) and etc. The framework is an attractive choice for problems where the number of features is large, such as fMRI data (Kuncheva et al. 2010) or gene expression data (Bertoni et al. 2005).

Bagging ensemble is a classical ensemble method. It generates diversified component classifiers by different training sets (Skurichina and Duin 2001). Bagging samples training set repeatedly with replacement, which produces different versions of training sets. Different components are trained by different versions of training set. Final classification model is built by votes of components. Bagging has been successfully used in many real classification tasks. Compared with single classifier, it improves the stability and accuracy. It is worth mentioning that some samples may occur in many training sets while some samples may not be included in any training set. This tells us that ensemble can compensate for some loss of samples, which gives us inspiration for missing data treatment. In addition, different versions of training set are independent. Therefore, they can be parallel generated and corresponding components ca be trained simultaneously.

3 Proposed methodology

In this section, we propose complete-case projection subspace ensemble framework. In this framework, ELM is selected as a component classifier. Two subspace partition strategies are developed for uniform missing patterns and non-uniform missing patterns respectively.

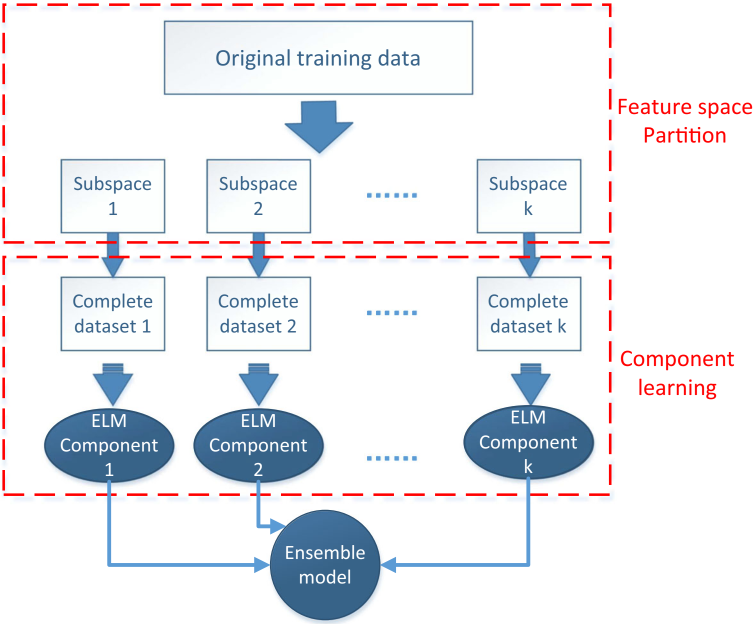

Fig. 1 Complete-case projection subspace ensemble framework

3.1 Complete-case projection ensemble framework

Inspired by classical ensemble methods (introduced in Sect. 2.2) and CCL, we propose a complete-case projection subspace (CPS) ensemble framework for classification with high dimensional missing data (depicted in Fig. 1).

CPS projects incomplete data into different feature subspaces. Specifically, for each subspace projection, once a sample is fully observed in a subspace (rather than in whole feature space), it will be kept in the projection. Otherwise, it will be omitted in this subspace.

CPS is defined as follows: Let $D_{M*N} = \{x_j | j = 1, 2 \ldots, M\}$ be a dataset with M samples and N features. Sample $\boldsymbol{x_j} = (\boldsymbol{x_j}^{(1)}, \boldsymbol{x_j}^{(2)}, \ldots, \boldsymbol{x_j}^{(N)})$. D_{M*N} is projected into subspace S ($|S| = n < N$) as Eq. (4).

$$\mathrm{CPS}_S(D) = \left\{ \left(\boldsymbol{x_j}^{(s_1)}, \boldsymbol{x_j}^{(s_2)} \ldots \boldsymbol{x_j}^{(s_n)} \ldots \right) | \forall k \forall j, \quad \boldsymbol{x_j}^{(s_k)} \neq NaN \right\}, \tag{4}$$

where NaN denotes missing value. For an incomplete dataset, the less features in its subspaces, the more samples are preserved after complete-sensitive subspace projection.

Both features and training samples are projected into different subspaces, which naturally results in diverse component classifiers. Through projection, all training subsets are complete. Therefore, many incomplete samples in whole space turns into complete in subspaces and their observed value are used in training. In addition, for most incomplete testing samples, it can be predicted by some component classifiers at least.

3.2 Bootstrapping subspace partition

For datasets with not too many features, if we divide feature space in a mutually-exclusive way (such as random subspace ensemble), the number of the feature in subspace is too

Fig. 2 Missing indication matrix

$$\begin{array}{c|cccccc} & f1 & f2 & f3 & f4 & f5 & f6 \\ \hline s1 & 1 & 0 & 1 & 0 & 1 & 1 \\ s2 & 1 & 1 & 1 & 0 & 1 & 1 \\ s3 & 0 & 1 & 1 & 1 & 0 & 1 \\ s4 & 1 & 1 & 0 & 1 & 1 & 0 \\ s5 & 1 & 1 & 0 & 1 & 1 & 1 \\ s6 & 0 & 1 & 1 & 1 & 0 & 1 \end{array}$$

small. It makes learning in subspaces meaningless. Inspired by bagging methods, we propose bootstrapping subspace ensemble method (BS). As a kind of overlapped partition method, BS generates subspaces by bootstrapping features rather than samples. In order to make equal chance for all possible feature combinations in subspaces, BS maintains a dynamic distribution for feature selection. During feature selection for subspace s, BS draws a random bootstrap sample of nof features according to then current feature distribution D_s (nof is a preset value). Initially, all features are of equal opportunity to be selected (i.e. $D_1(1) = D_1(2) = \cdots = D_1(N)$). Once feature f is picked for a subspace, its opportunity for next selection decreases as $D_{t+1}(f) = \frac{D_t(f)}{\gamma}$ ($\gamma \geq 1$).

3.3 Missing pattern-sensitive subspace partition

For incomplete datasets with non-uniform missing patterns, random subspace ensemble is a favorable choice. Traditional subspace ensemble method divides a feature space randomly. In recent years, some feature selection methods are used for feature space partition. The main focus of those methods is relevance and correlation of features.

Unlike existing methods, we aim to get as much observed values (from incomplete samples) as possible in complete-case projection. We propose a missing pattern-sensitive subspace (MPS) partition strategy. For missing data, traditional subspace methods may lead to serious loss of observed value in each subspace under complete-case projection. While MPS partition keeps as many observed values as possible by the way of grouping features by missing pattern. Here, we illustrate it by a toy example in Fig. 2. Indication matrix for a dataset consists of 0 (which denotes missing value) and 1 (which denotes observed value). Each column f_i indicates a missing pattern of a feature (i.e. mp-col) while each row s_i indicates a missing pattern of a sample (i.e. mp-row). It can be observed that f_1 and f_5 have similar missing pattern, while f_1 and f_4 are quite different.

We quantify similarity of missing pattern by Eq. (5).

$$Sim\_MP(i, j) = \frac{\left[f_i \; and \; f_j\right]_1}{\left[f_i \; or \; f_j\right]_1} \tag{5}$$

where operator $[.]_1$ counts the number of value 1 in a vector. *and* and *or* are vector logical operation. For example, $Sim\_MP(1, 4)$ equals to 0.33 and $Sim\_MP(1, 5)$ equals to 1.

The detail of MPS partition is given in algorithm 1. First, it initializes each subspace with one feature. These initial features are of relatively low Sim_MP value and are determined by a heuristic way. Then, remaining features are added to its corresponding subspaces (i.e. the subspace which shares most similar missing pattern with the feature). In step 7, function $SUBS\_MATCH$ returns the subspace which has the most similar missing pattern with feature f. Missing pattern of a subspace is logical *and* of all feature pattern vectors in the subspace. The Sim_MP value of a feature and a subspace is calculated by the same way as Eq. (5).

Algorithm 1 Missing Pattern-sensitive Subspace Partition

```
procedure MPS_PARTITION($mp$, $k$)                                ▷ $I$: miss indication matrix

                                                                  ▷ $k$: number of subspaces
    $inifs \leftarrow GET\_INITIAL\_FEATURES(mp, k)$
    $initial\ subs_1, subs_2 \ldots, subs_k\ with\ inifs$         ▷ $subs$: feature subspace
    $subspaces \leftarrow \{subs_1, subs_2 \ldots, subs_k\}$
    for $f \in FS$ do                     ▷ $FS$: set of features that are not added into any $subs$
        $s \leftarrow SUBS\_MATCH(f, subspaces)$                  ▷ find the most similar subspace for $f$
        $insert\ f\ into\ s$
    end for
    return $subspaces$
end procedure

procedure GET_INITIAL_FEATURES($mp$, $k$)
    $inifs \leftarrow \{f_a, f_b\}$              ▷ $f_a$, $f_b$ are features with minimum $Sim\_MP$ value
    $count \leftarrow 2$
    while $count < k$ do
        $f \leftarrow feature\ which\ has\ minimum\ mps\ with\ inifs$
        $insert\ f\ into\ inifs$
        $count++$
    end while
    return $inifs$                 ▷ $inifs$ consists of features with relatively low $Sim\_MP$ value
end procedure
```

3.4 Components combination

CPS combines component classifiers with weighted majority vote. There are two considerations in our weighting strategy. On the one hand, it is possible that a component is more accurate when it is acquired by learning more complete samples. On another hand, a model with higher training accuracy is more reliable. Therefore, component weights are calculated as Eq. (6) and then normalized as Eq. (7). The In addition, more complete datasets leads to more accurate learning.

$$\omega_i = completeness\_ratio * \log_2\left(\frac{acc_i}{1 - acc_i}\right), \quad i = 1, 2, \ldots, N \tag{6}$$

where $completeness\_ratio = \frac{\#complete\ samples\ in\ the\ subspace}{\#total\ samples}$ ($completeness\_ratio$ reflects the number of complete samples in each subspace).

$$\omega_i = \frac{\omega_i}{\sqrt{\omega_1^2 + \omega_2^2 + \cdots + \omega_N^2}}, i = 1, 2, \ldots, K, \quad K \leq N \tag{7}$$

In testing phase, for complete samples, the final outputs of ensemble are given by Eq. (8) ($K = N$). In the case of prediction for an incomplete sample, only the component classifiers whose input space do not include the missing features of the incomplete sample are weighted and combined ($K \leq N$).

$$pred(x) = \arg\max_{c} \sum_{i=1}^{K} \omega_i * \frac{p_i^{(c)}(x)}{\sum_{c=1}^{C} p_i^{(c)}(x)}, \tag{8}$$

where C is class label set, and $p_i^{(c)}(x)$ denotes i component classifier predicts sample x belongs to class c.

3.5 Discussion

Classification with missing data by ensemble method is not new. As the most representative work, (Peter and Solly 1995) uses multiple neural network for classify thyroid disease data (with missing values). In order to get deeper understanding of the proposed methods of our work, we illustrate the novelty of the proposed method by discussing the difference between this work from Peter and Solly (1995). The differences are mainly in three aspects: (1) The proposed framework is specially designed for high dimensional data. In detail, it uses two partition methods, .i.e. bootstrap partition and missing pattern (from the perspective of features) based partition. While the reduced network is not well fit for high dimensional data. If the number of features are high (e.g. there are N features), there may be 2^N missing patterns (from the perspective of samples), leading to 2^N component learners, which can be expensive and infeasible for high dimensional data. (2) The proposed framework chooses extreme learning machine learning algorithm as the base learner. Due to its learning efficiency and diversity (randomly generated input weights and biases). Compared with other neural network, Extreme learning machine is more suitable to ensemble method. (3) The way of combination are different. The outputs of base classifier in reduced neural network take a range of values between 0.0 and 1.0 and the final output decided by winner-takes-all strategy. In our proposed method, outputs of base classifiers are either 0 or 1. Final output is ensembled by voting strategy.

4 Experiments

This section demonstrates the effectiveness of CPS framework and two proposed partition methods. All simulations are implemented on MATLAB 2015a environment running in Core(TM) 3.0 GHz CPU and 16 GB RAM. The data sets and their characteristics are presented in the Table 1. They are from the UCI Machine Learning Repository (Lichman

Table 1 Datasets used in the experiments

Dataset	Training samples	Testing samples	Feature dimension	Class number
Ionosphere	234	117	34	2
Segmentation	1540	770	19	7
WBCD	466	233	30	2
Vehicle	564	282	18	4
Madelon	2933	1467	500	2
Isolet	6238	1559	618	26
Multiple features	1333	667	649	10
Internet ads.	2186	1093	1558	2

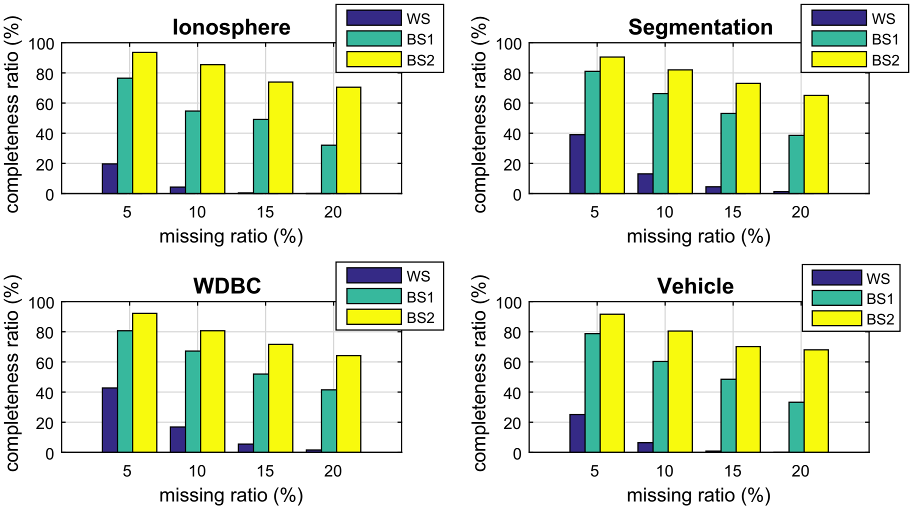

Fig. 3 Completeness ratios in whole feature space (WS), BS1, and BS2

2013) with various dimensions ranging from 19 to 1588 and various numbers of samples from 234 to 6238. The missing value are produced artificially for controlling missing ratio. We simulate missing value with uniform missing patterns on first four datasets (used in Sect. 4.1) and non-uniform missing patterns on the other four datasets (used in Sect. 4.2).

For all simulations, the best parameters (activation function and number of hidden units) of ELM is selected by cross-validation. In addition, we use two popular imputation methods as contrasts. Abbreviations are as follows. MI indicates single classifier running on data with mean imputation and KNNI indicates single classifier running on data with k nearest neighbors imputation. CPS-BS denotes CPS framework with overlapped partition. CPS-MPS means CPS framework with missing pattern-sensitive partition. CPS-RS represents CPS framework with random subspace partition.

4.1 Classification accuracy of incomplete datasets with uniform missing pattern

In this subsection, we aim to demonstrate the advantage of BS partition method for datasets with uniform missing pattern. For controlling missing ratio, we produce missing data artificially and missing data distribute uniformly and randomly. We record the completeness ratios of training sets in whole feature space and BS. The completeness ratios of BS are average values of all subspaces. The number of features in BS is selected empirically. Additionally, BS with different feature numbers in subspace are compared. BS1 denotes overlapped subspace ensemble with $\sqrt{f}$ features and BS2 means overlapped subspace ensemble with $\frac{\sqrt{f}}{2}$ features. (f is the number of features in whole feature space).

Figure 3 shows the completeness ratios of training sets. The completeness ratio in whole feature space drops sharply with missing ratio increasing. While in subspaces, completeness ratio decreases mildly. It can be concluded that more complete samples are derived through subspace projection. Further, the result indicates that more observed values are learned by component classifiers in subspaces than in whole feature space. Meanwhile, it shows that less number of features in subspaces leads to higher completeness ratio.

The corresponding classification accuracy is recorded in Table 2. Figure 4 depicts the trend of classification accuracy change under different missing ratios. The number of classifiers

Table 2 Classification accuracy of incomplete datasets with uniform missing patterns (optimal value of each row is shown in bold)

Dataset	MR (%)	CPS-BS1 (%)	CPS-BS2 (%)	MI (%)	KNNI (%)
Ionosphere	5	90.71	88.52	89.63	**90.82**
	10	**89.06**	86.93	86.24	88.37
	15	**86.92**	83.76	82.76	83.92
	20	**83.07**	81.36	80.19	82.54
Segmentation	5	77.03	**80.69**	78.23	81.07
	10	76.63	**78.23**	76.37	78.38
	15	73.58	**75.39**	71.64	73.63
	20	72.83	**74.62**	70.56	72.9
Wdbc	5	93.85	93.07	93.41	**94.06**
	10	92.64	**92.93**	91.67	92.71
	15	90.53	**91.83**	89.35	91.67
	20	88.39	**90.67**	85.08	87.72
Vehicle	5	74.23	**76.36**	72.63	75.37
	10	71.06	**74.83**	71.38	73.94
	15	69.73	**74.69**	68.06	70.62
	20	64.37	**73.22**	62.05	66.47

(i.e. the number of subspaces) is set to be 50 empirically. In practical, optimal value is task dependent and can be derived by cross-validation. Following are analysis based on observations.

(1) For the Ionosphere dataset, when missing ratio is less than 5 %, all algorithms achieve relatively accurate learning. When missing ratio exceeds 10 %, CPS-BS1 performs a slightly better than KNNI. As missing ratio goes on, CPS-BS1 shows more and more obvious advantage than others. (2) For the Segmentation dataset, when missing ratio is below 10 %, KNNI is the most accurate. When missing ratio is higher than 10 %, CPS-BS1 performs better than others and MI drops sharply. The accuracies of CPS-BS2 and KNNI are close. (3) For the WDBC dataset, the performance of all methods shows similar classification ability. MI is the worst method when missing ratio exceed 10 %. (4) For the Vehicle dataset, CPS-BS achieves the most accurate learning all the time. while the accuracies of other three decrease greatly as missing ratio goes up. KNNI is the second best method and MI is the most inaccurate.

Above all, completeness ratio and classification accuracy are positively correlative. As the missing ratio increases, both the classification accuracy and the completeness ratio drop. CPS-BS1 shows obvious advantage over other methods, especially when missing ratio is high. CPS-BS2 performs worse than CPS-BS1 and KNNI in most cases. MI declines most seriously as missing value becomes more. Compared with imputation methods, CPS drops more mildly as missing ratio goes up.

4.2 Classification accuracy of incomplete datasets with non-uniform missing pattern

For incomplete datasets with non-uniform missing patterns, missing values concentrate in a few features. Instead of producing missing value completely randomly, we simulate this

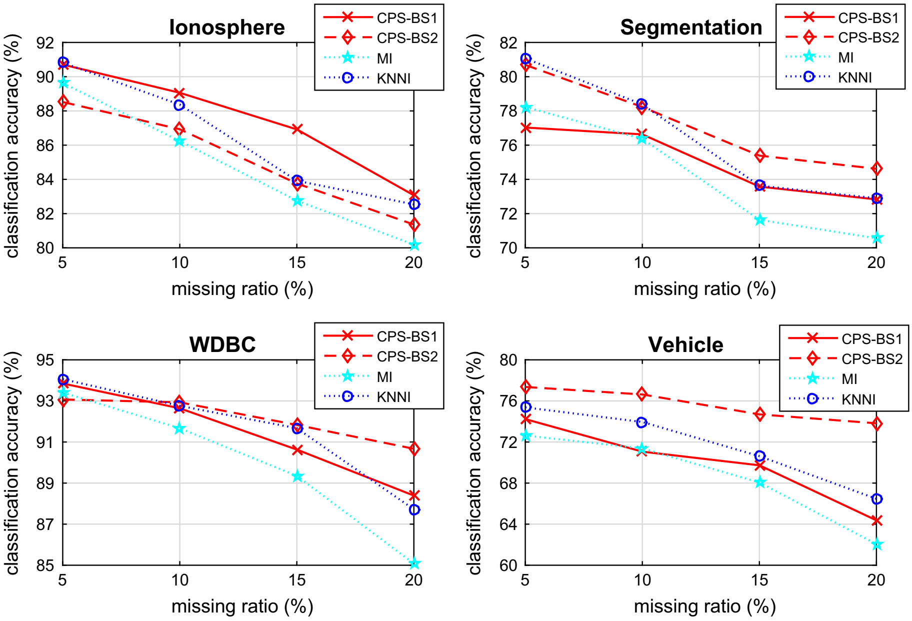

Fig. 4 Classification accuracy of incomplete datasets with uniform missing pattern

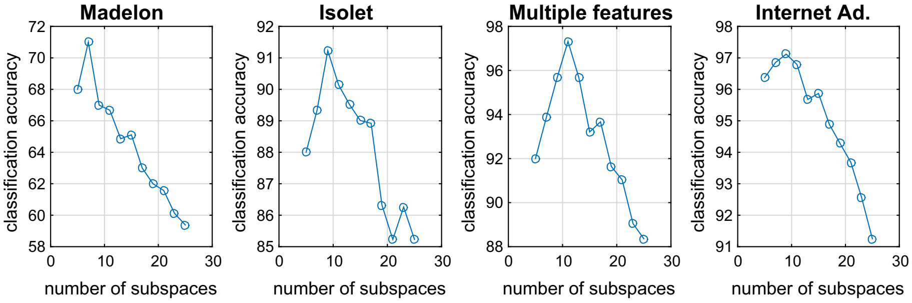

Fig. 5 Classification accuracy over different numbers of subspaces

situation by selecting some easy-missing features in which missing value resides. Specifically, we randomly choose a subset (of 10 %) of features as easy-missing features and set missing ratio to be (2, 4, 6, 8 %) for them.

The number of subspaces is critical for subspace ensemble classification. In our experiments, the value selected for the number of subspaces (i.e. k) are 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25. Accordingly, the subsampling number of features in subspaces are 20, 14,… and 4 %. Figure 5 presents the classification accuracy of complete datasets with different numbers of subspaces. It can be seen that different k lead to different classification accuracies. Appropriate choice of k value is task-dependent. Note that for incomplete data sets, k value affects average completeness (in subspaces), which further affects classification accuracy. Therefore, we decide k for incomplete datasets by cross-validation.

Here, we first compare the completeness of different partition methods (i.e. RS and MPS) and then observe their classification accuracies. Figure 6 depicts the average completeness

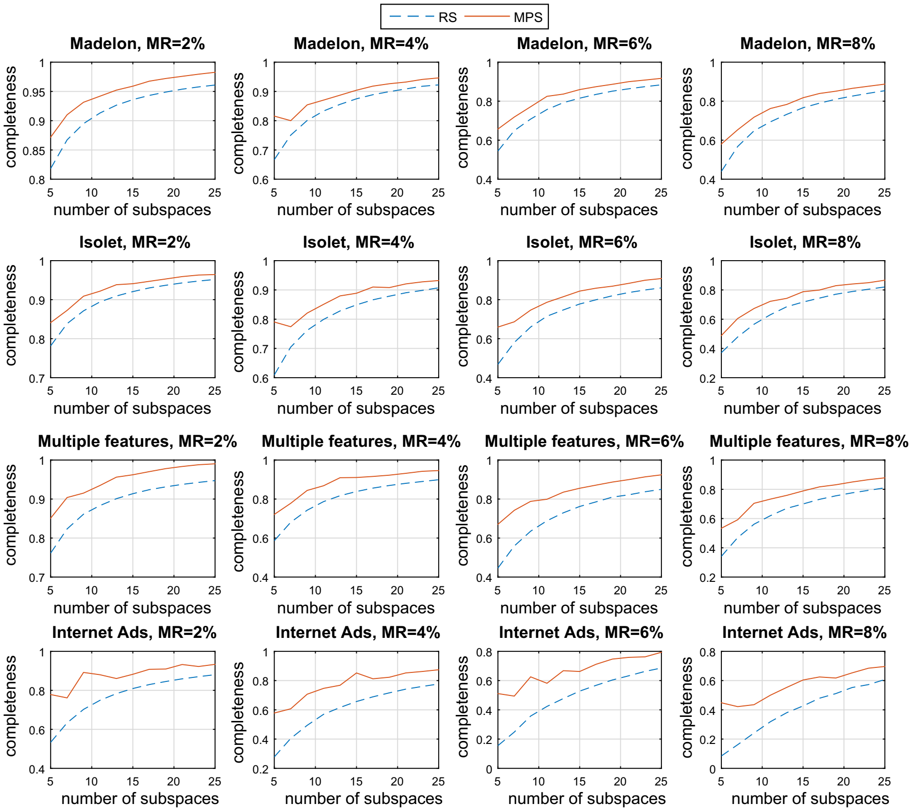

Fig. 6 Average completeness of feature subspaces. MR denotes missing ratio

of RS and MPS with different k value. Table 3 shows the classification accuracy of the four datasets with various missing ratios. It can be observed that (1) For the Madelon dataset, when missing ratio equals to 2 %, CPS-RS is the most accurate method. For the rest of conditions, CPS-MPS performs most accurate. (2) For the Isolet dataset, when missing ratio is less than 4 %, KNNI and CPS-RS achieve comparably accurate classification. When missing ratio exceeds to 6 %, CPS-MPS is the most accurate method and the accuracy of CPS-RS drops sharply. (3) For the Multiple feature dataset, CPS-MPS is the most accurate all the time. CPS-RS performs well when missing ratio is less than 4 %. KNNI preforms stably in various missing ratios. (4) For the Internet Ads. dataset, KNNI shows obvious advantage over others. CPS-MPS is the second accurate method.

Figure 7 demonstrates the change of classification accuracy with increasing missing ratio more clearly. It can be observed that (1) MI curve is at bottom in most cases. (2) Although CPS-RS preforms comparably accurate, it drops fast, especially in Multiple feature and Internet Ads. (3) CPS-MPS achieve accurate and stable classification over different missing ratios. (4) Sometimes KNNI performs most accurate (e.g. Internet Ads.).

To get a deeper understanding of the proposed methods, further explanations are given as following. Two bases of this work are: (1) With the assumption that observed values (including the observed value of incomplete samples) are useful for learning, the more information being utilized, the more accurate a classification model is. (2) The combination of multiple

Table 3 Classification accuracy of datasets with non-uniform missing patterns

Dataset	MR (%)	MI (%)	KNNI (%)	CPS-RS (%)	CPS-MPS (%)
Madelon	2	78.12	79.83	**80.75**	80.52
	4	78.49	79.35	78.36	**80.63**
	6	71.53	77.62	74.29	**79.85**
	8	68.07	76.59	71.79	**75.6**
Isolet	2	89.83	91.26	**91.68**	90.69
	4	85.67	**90.35**	83.67	90.26
	6	82.73	88.24	83.23	**88.37**
	8	80.54	86.33	78.67	**87.68**
Multiple features	2	87.75	88.95	90.92	**91.16**
	4	85.29	87.64	90.38	**90.81**
	6	84.37	87.73	83.37	**89.35**
	8	83.38	86.64	75.79	**88.67**
Internet ads.	2	77.67	**80.59**	75.39	78.46
	4	73.38	**79.21**	72.67	76.52
	6	71.26	**77.69**	69.25	73.14
	8	63.37	**76.34**	65.19	71.26

Optimal value of each row is in bold

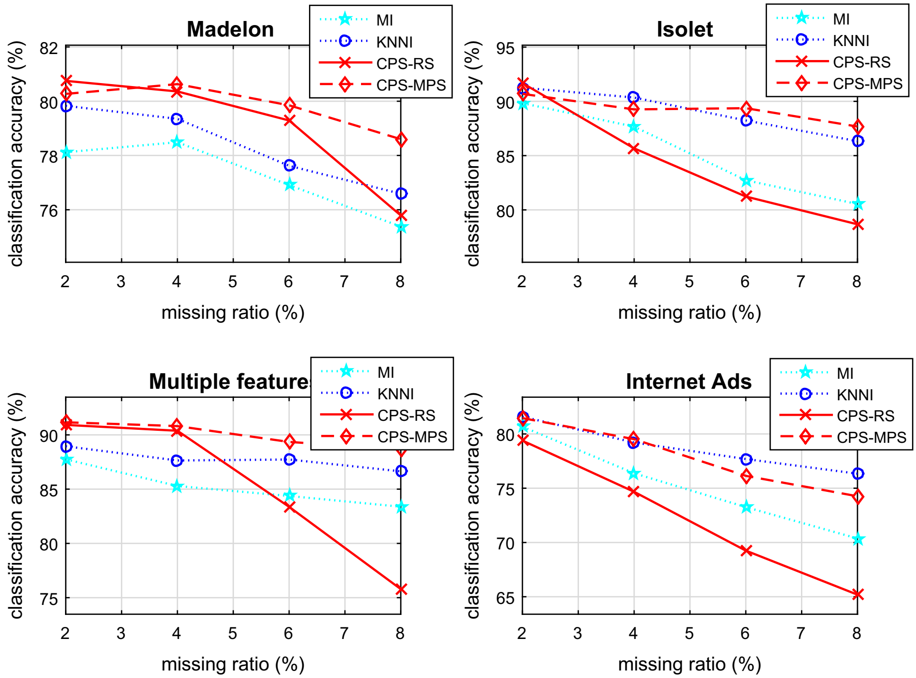

Fig. 7 Classification accuracy of incomplete datasets with non-uniform missing patterns

classifiers is effective for high dimensional data. Since the completeness ratio increases significantly in subspaces, CPS learns more values from incomplete samples than CCL in whole feature space. Compared with imputation methods, CPS does not introduce any estimated value which may lead to additional error. Meanwhile, we use different subspace partition methods for uniform and non-uniform missing patterns respectively. For datasets with the uniform missing pattern, BS bootstraps features from whole feature space. For datasets with the non-uniform missing pattern, MPS partition further improves the completeness ratio of subspaces.

5 Conclusion and future work

In this paper, we propose an ELM-based complete-case projection subspace ensemble framework for classification with high dimension missing data. Additionally, BS and MPS partition strategies are designed for datasets with different missing patterns. CPS learns from incomplete samples as well as complete ones. Additionally, incomplete samples can be predicted by component classifiers in their corresponding subspaces. The experimental results demonstrate that CPS outperforms imputation methods in classification accuracy in most cases. In future, we are going to parallelize CPS in a cluster environment. Next, we continue to explore feature space partition algorithm considering feature relation as well as the number of complete samples.

Acknowledgements This work is supported by the Major State Basic Research Development Program of China (973 Program) under the Grant No. 2014CB340303, and the Natural Science Foundation under Grant No. 61402490.

References

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2007). A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 173–180.

Batista, G. E. A. P. A., & Monard, M. C. (2002). A study of k-nearest neighbour as an imputation method. *HIS*, *87*(251–260), 48.

Batista, G. E. A. P. A., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5–6), 519–533.

Bertoni, A., Folgieri, R., & Valentini, G. (2005). Bio-molecular cancer prediction with random subspace ensembles of support vector machines. *Neurocomputing*, *63*, 535–539.

Bryll, R., Gutierrez-Osuna, R., & Quek, F. (2003). Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, *36*(6), 1291–1302.

Cao, J., & Lin, Z. (2015). Extreme learning machines on high dimensional and large data applications: A survey. *Mathematical Problems in Engineering*, *2015*, 1–12.

Cao, J., Lin, Z., Huang, G.-B., & Liu, N. (2012). Voting based extreme learning machine. *Information Sciences*, *185*(1), 66–77.

Donders, A. R. T., van der Heijden, G. J. M. G., Stijnen, T., & Moons, K. G. M. (2006). Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, *59*(10), 1087–1091.

Enders, C. K. (2001). A primer on maximum likelihood algorithms available for use with missing data. *Structural Equation Modeling*, *8*(1), 128–141.

Graham, J. W., Olchowski, A. E., & Gilreath, T. D. (2007). How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science*, *8*(3), 206–213.

Ho, T. K. (1998). Nearest neighbors in random subspaces. In *Advances in pattern recognition* (pp. 640–648). Springer.

Huang, W., Yang, Y., Lin, Z., Huang, G.-B., Zhou, J., Duan, Y., Xiong, W. (2014). Random feature subspace ensemble based extreme learning machine for liver tumor detection and segmentation. In *Engineering* *medicine and biology society (EMBC), 2014 36th annual international conference of the IEEE* (pp. 4675–4678). IEEE.

Huang, G.-B. (2015). What are extreme learning machines? Filling the gap between Frank Rosenblatts dream and John von Neumanns puzzle. *Cognitive Computation*, *7*(3), 263–278.

Huang, G., Huang, G.-B., Song, S., & You, K. (2015). Trends in extreme learning machines: A review. *Neural Networks*, *61*, 32–48.

Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man and Cybernetics Society*, *42*(2), 513–529.

Kuncheva, L., Rodríguez, J. J., Plumpton, C. O., Linden, D. E. J., Johnston, S. J., et al. (2010). Random subspace ensembles for fMRI classification. *IEEE Transactions on Medical Imaging*, *29*(2), 531–542.

Lichman, M. (2013). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml.

Li, X., & Mao, W. (2016). Extreme learning machine based transfer learning for data classification. *Neurocomputing*, *174*, 203–210.

Little, R. J. A., & Rubin, D. B. (2014). *Statistical analysis with missing data*. New York: Wiley.

Marlin, B. M. (2008). Missing data problems in machine learning. Doctoral.

Scheffer, J. (2002). Dealing with missing data. *Research Letters in the Information and Mathematical Sciences*, *53*(1), 153–160.

Sharpe, P. K., & Solly, R. J. (1995). Dealing with missing values in neural network-based diagnostic systems. *Neural Computing and Applications*, *3*(2), 73–77.

Skurichina, M., & Duin, R. P. W. (2001). Bagging and the random subspace method for redundant feature spaces. In *Multiple classifier systems* (pp. 1–10). Springer.

Xie, Z., Xu, K., Liu, L., & Xiong, Y. (2014). 3d shape segmentation and labeling via extreme learning machine. In *Computer graphics forum* (Vol. 33. No.5, pp. 85–95). Wiley Online Library.

Hang Gao received the B.Sc. degree in school of computer science and technology from Shandong University of Architecture, Jinan, China, in 2006, and the M.Sc. degree from College of Computer from National University of Defense Technology, ChangSha, China in 2009. His research interests include Extreme learning machine, neural networks, and fuzzy system.

Songlei Jian received the B.Sc. degree in College of Computer from National University of Defense Technology, Changsha, China in 2013, where she is currently working toward the Ph.D. degree. Her research interests include data science and big data analytics, complex network.

Yuxing Peng received the B.Sc. degree in School of Computer Science and Engineering from BeiHang University, Beijing, China, in 1983, and the Ph.D. degree from National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China in 1996, where he is currently a Professor. He has led/implemented several National key research projects (e.g. National High-tech R&D Program of China, and National Program on Key Basic Research Project and etc.). His currently research interests are in area of machine learning, parallel and distributed systems and high performance computing.

Xinwang Liu received the M. Eng. and the Ph.D. degree from National University of Defense Technology, China in 2008 and 2013, respectively. From Jan. 2014, He works as a research assistant at National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China. His research interests focus on designing algorithms on kernel learning, feature selection and multiview clustering.