Research Article

Sample-Based Extreme Learning Machine with Missing Data

Hang Gao, Xin-Wang Liu, Yu-Xing Peng, and Song-Lei Jian

Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Hang Gao; hanggao1821@163.com

Received 21 August 2014; Revised 8 November 2014; Accepted 10 November 2014

Academic Editor: Zhan-li Sun

Copyright © 2015 Hang Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Extreme learning machine (ELM) has been extensively studied in machine learning community during the last few decades due to its high efficiency and the unification of classification, regression, and so forth. Though bearing such merits, existing ELM algorithms cannot efficiently handle the issue of missing data, which is relatively common in practical applications. The problem of missing data is commonly handled by imputation (i.e., replacing missing values with substituted values according to available information). However, imputation methods are not always effective. In this paper, we propose a sample-based learning framework to address this issue. Based on this framework, we develop two sample-based ELM algorithms for classification and regression, respectively. Comprehensive experiments have been conducted in synthetic data sets, UCI benchmark data sets, and a real world fingerprint image data set. As indicated, without introducing extra computational complexity, the proposed algorithms do more accurate and stable learning than other state-of-the-art ones, especially in the case of higher missing ratio.

1. Introduction

Extreme learning machine (ELM) was proposed by Huang et al. in [1]. It works for the generalized single-hidden layer feedforward networks (SLFNs) [2]. Different from traditional tenet in neural network learning, hidden node parameters of ELM are randomly generated and do not need to be adjusted [3]. Therefore, it achieves fast learning speed as well as excellent generalization ability with least human intervention. As an open, scalable, and unified learning framework, ELM can be used in various kinds of learning tasks, for example, classification, regression, representation learning, and so forth. Its learning theory has become an active research topic in machine learning domain in recent years. On one hand, many improved variants have been developed. On another hand, ELM learning mechanism has been integrated into some well-known platforms and systems. The following is a brief review. The main objective of variants is mainly twofold: (1) achieving more accurate learning by introducing different regularization methodologies and (2) reducing computational complexity by simplifying ELM network architecture. Regularized ELM [4] considers heteroscedasticity in real applications and reduces the effect

of outliers in the data set. In [5], ridge regression, elastic net, and lasso methods are used to prune ELM network, which leads to compact architecture. Sparse Bayesian ELM [6] estimates the marginal likelihood of network outputs and automatically prunes most of the redundant hidden neurons. Localized error model based ELM utilizes principal component analysis to reduce the feature dimension and then selects the optimal architecture of the SLFN. OP-ELM uses LASSO regularization technique to rank the neurons of hidden layer and obtains a more parsimonious model. TROP-ELM [7] further improves OP-ELM by using a cascade regularization method based on LASSO and Tikhonov criteria. OP-ELM-ER-NCL [8] combines ensemble of regularization techniques with negative correlation penalty. TS-ELM [9] introduces a systematic two-stage mechanism to determine the network architecture. MK-ELM [10] considers multiple heterogeneous data sources and uses a multikernel strategy to optimize ELM. Moreover, ELM is modified according to different characteristics of data sets and applied in various applications. OS-ELM [11] learns the data one-by-one or chunk-by-chunk with fixed or varying chunk size. EOS-ELM [12] further improves OS-ELM's learning stability by adapting the node location, adjustment, and pruning methods. Weighted ELM

[13] assigns different weights to different samples according to users' needs, realizing cost sensitive learning. T2FELA [14] inherits the merits of ELM and randomly generates the parameters of the antecedents, successfully applying ELM in type 2 fuzzy logic system. In [15], ELM is integrated into MapReduce framework for large scale regression. All in all, those extensions and variants inherit the merits of ELM and are more suitable for specific application scenarios. However, missing data problem which is highly pervasive in real world tasks is rarely considered in ELM learning.

Missing data is a common situation where null value is stored for some samples in the data sets. This problem has complex patterns. It occurs by several practical reasons such as equipment malfunction in data sampling or transmission and noise value being deleted in data preprocessing [16]. Besides, there are some inherently missing caused by non-existing features. For example, an image sample may not contain all the predefined components. Standard ELM learning requires all samples in the data set to be complete and have the same dimensions. Obviously, it cannot directly handle the issue of missing data. Traditional approaches depend on the preprocessing for missing data before learning. Inevitably, those approaches introduce extra preprocessing overhead. Worse still, they may mistakenly omit some useful information and produce a certain amount of error information [17]. Therefore, learning precision can be seriously affected. In this paper, we propose a sample-based learning framework and develop a sample-based ELM classification algorithm and a sample-based ELM ϵ -insensitive regression algorithm. In the proposed learning framework, missing data can be directly learned without any extra preprocessing.

The contributions of this paper are as follows. First, we analyze the limitations of state-of-the-art approaches for learning with missing data. Second, we propose a sample-based ELM learning framework for learning missing data. Third, we develop two sample-based ELM algorithms for classification and regression, respectively. Experiment results show that the proposed algorithms achieve more accurate classification and regression compared with traditional approaches, especially in the situation that missing ratio is relatively intensive.

2. ELM Learning

In Section 2.1, we review the basic concepts of ELM. Then, from general ELM optimization formula, we discuss several ELM variants and illustrate their characteristics. In Section 2.2, we introduce ELM ϵ -insensitive regression and explain its advantage over standard ELM regression.

2.1. Standard ELM. From the viewpoint of neural network, ELM can be seen as generalized SLFNs. Figure 1 gives the network architecture. ELM randomly chooses hidden nodes and analytically determines the output weights of SLFNs. In theory, ELM can approximate any target continuous function and classify any disjoint regions [2]. Its interpolation capability and universal approximation capability have been

FIGURE 1: ELM's network architecture.

investigated by Huang et al. in [18, 19]. The corresponding output function is given as

$$f(\mathbf{x}) = \sum_{i=1}^L \beta_i * h(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i) \cdot \boldsymbol{\beta}, \quad (1)$$

where $\boldsymbol{\beta}$ is the output weight vector and $\mathbf{h}(\mathbf{x})$ is the random mapping function. Being superior to traditional learning algorithm, ELM tends to not only minimize the empirical risk but also minimize the structural risk. Its general optimization formula is as

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_p^{\sigma_1} + C \times \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_q^{\sigma_2}, \quad (2)$$

$$\mathbf{H} = \begin{pmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_L(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{pmatrix}. \quad (3)$$

From the viewpoint of learning theory, ELM considers empirical risk as well as structural risk which can be observed in ELM's general optimization formula (2). $\mathbf{H}$ is the random mapping matrix, and $\mathbf{Y}$ is the training data targets. Evolved from (1), with different combinations of parameters and constraints in general formula, ELM derives many variants of different regularization methods. Additionally, kernel methods are used to enhance the learning ability especially for multiple data sources learning [20]. With competitive performance in generalization abilities, they are widely used in classification, regression, representation learning and clustering, and so forth. The following are some representative forms.

Basic ELM. There are two kinds of basic ELM. When $C = +\infty$ (i.e., ELM only concerns empirical risk), the optimization objective equals $\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\|_q^{\sigma_2}$. In another extreme; when $C = 0$ (i.e., ELM only concerns the structural risk), the optimization objective equals $\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_p^{\sigma_1}$. Obviously, basic ELM learning has high learning efficiency. But both of them take only one optimization objective into account, which can cause overfilling or underfitting.

Inequality Optimization Constraints Based ELM. With the parameter setting of $\sigma_1 = 2$, $\sigma_2 = 1$, $p = 2$, $q = 1$, and inequality constraints, the general optimization formula can be written as (4), which is common in binary classification. Since this form is applied widely and has good sparsity, we use it as the base model of our extension for classification:

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i} \quad & \frac{1}{2} \|\beta^{(i)}\|^2 + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & \mathbf{y}_i * \left(h(\mathbf{x}_i) \cdot \beta^{(i)^T} \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \quad (4)$$

Applying KKT conditions, (4) can be transformed into (5); then it can be solved in dual space:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \langle h(\mathbf{x}_i), h(\mathbf{x}_j) \rangle, \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n. \end{aligned} \quad (5)$$

Equality Optimization Constraints Based ELM. With the parameter setting of $\sigma_1 = 2$, $\sigma_2 = 1$, $p = 2$, $q = 1$, and equality constraint, the general ELM optimization formula is equivalent to (6) which can be used in regression and classification:

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i} \quad & \frac{1}{2} \|\beta^{(i)}\|^2 + C \sum_{i=1}^n \xi_i^2, \\ \text{s.t.} \quad & h(\mathbf{x}_i) \cdot \beta^{(i)^T} = y_i - \xi_i, \quad i = 1, 2, \dots, n. \end{aligned} \quad (6)$$

The corresponding KKT optimal conditions are shown in

$$\begin{aligned} \beta &= H\alpha, \\ \alpha_i &= C\xi_i, \quad i = 1, 2, \dots, n, \\ \mathbf{h}(\mathbf{x}_i) \beta - \mathbf{y}_i^T + \xi_i^T &= 0, \quad i = 1, 2, \dots, n. \end{aligned} \quad (7)$$

Further, the final output is given in

$$\mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \beta = \mathbf{h}(\mathbf{x}) \mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T}. \quad (8)$$

2.2. ELM for ϵ -Insensitive Regression. For regression, ELM provides general model for standard setting. It achieves better predictive accuracy than traditional SLFNs [3]. In addition, many variants and extensions of ELM regression algorithms have been proposed. Inspired by Vapnik's epsilon insensitive loss function, [21] proposed ϵ -insensitive ELM. Its optimization formula is as

$$\min_{\beta} \left(\frac{1}{2} \times \|\beta\|^2 + \frac{C}{2} \times \|h(\mathbf{x}) \beta - \mathbf{y}\|_{\epsilon}^2 \right), \quad (9)$$

where ϵ is insensitive factor and the error loss function is calculated by

$$\|h(\mathbf{x}) \beta - \mathbf{y}\|_{\epsilon}^2 = \sum_{i=1}^n |f(\mathbf{x}_i) - y_i|_{\epsilon}^2, \quad (10)$$

$$\text{with } |f(\mathbf{x}_i) - y_i|_{\epsilon} = \max\{0, |f(\mathbf{x}_i) - y_i| - \epsilon\}.$$

Compared with conventional ELM regression, ELM with ϵ -insensitive loss function uses margin ϵ to measure the empirical risk. It controls the sparsity of the solution [22] and is less sensitive to different levels of noise [21]. In this paper, we extend ELM regression algorithm based on this variant.

3. Missing Data Problem in ELM Learning

3.1. Missing Data Problem. Nowadays, with ever-increasing data velocity and volume, missing data becomes a common phenomenon. Generally, there are two missing patterns, that is, missing feature and missing label. In this paper, we focus on the issue of missing feature.

From the causes of missing data, there are two circumstances. In the first circumstance, the missing features exist but their values are unobserved for the reason that information is lost or some features are too costly to be acquired [23]. Examples of such case can be found in many domains. Sensors in a remote sensor network may be damaged and fail to collect data intermittently. Certain regions of a gene micro array may fail to yield measurements of the underlying gene expressions due to scratches, fingerprints, or dust [16]. Second is inherently missing. In this circumstance, different samples inherently contain different features. For instance, in packed malware identification, instances contain some unreasonable values. In the web-page task, one useful feature of a given page may be the most common topic of other sites that point to it. If this particular page has no such parents, however, the feature is null, and should be considered structurally missing [24]. Obviously, imputation for this circumstance is meaningless.

3.2. Traditional Approaches for Missing Data Learning. Generally, there are three approaches for dealing with missing features in machine learning. The first approach is omitting, which includes sample deletion and feature filtering. Sample deletion simply omits the samples containing missing features and applies standard learning algorithms in the remaining samples. An example is shown in Figure 2; *sample 2* with two missing features is deleted. Feature filtering omits the features that are missing in most samples. Figure 3 interprets this approach. Obviously, the advantage of omitting based approaches is simple and computationally inexpensive. Notably, the key point of omitting is keeping as much as possible useful information while omitting. But it is difficult to do that. Both of them inevitably omit some useful information. When there is massive information retained after being partly omitted, this approach can be a better choice. Otherwise, in the situation of much useful information being omitted while few being retained, this kind of approaches affects learning precision seriously. Second approach is imputation. In data preprocessing phase, missing features are filled with most possible values [25]. Simple imputations fill the missing features with some default value such as zero or average value of other samples. Complex imputations use some probabilistic density function or distribution function to estimate the missing features. The computational complexity of imputation varies with different estimation methods. Imputation makes sense when the features are known to

FIGURE 2: Sample deletion.

FIGURE 3: Feature filtering.

exist and relatively easy to be estimated. But in some cases, the missing values are difficult to impute. Besides, in the situation of inherent missing, imputation is meaningless. Last but promising approach appears in recent years. Researchers manage to extend standard learning algorithms to deal with missing features. Different from the above two, it needs no extra preprocessing for missing data. It keeps the intactness of incomplete samples and extends standard learning algorithms according to their learning theory. Consequently, the extended algorithms learn over complete samples as well as incomplete samples. In this paper, the proposed sample-based learning belongs to this category.

4. Sample-Based ELM Framework and Algorithms

In this section, we start by illustrating ELM learning in a max-margin way as Huang explains and formalizes the learning problem with nonexistent features. Then, we explain why standard max-margin learning must be modified for learning missing data. Based on this, we propose the S-ELM classification algorithm in Section 4.2 and the S-ELM ϵ -insensitive regression algorithm in Section 4.3.

4.1. Sample-Based Learning Framework. In the viewpoint of geometric formulation, maximizing margin is equivalent to maximizing the dot products of sample vectors and output

FIGURE 4: ELM's max margin in feature space.

weight vectors. Meanwhile, the smaller the norms of output weights are, the better generalization the network tends to have. ELM learning is the process of adjusting output weight vectors by the above two objectives. For samples containing missing features, the margins are incorrectly scaled [24]. For example, in Figure 4, *sample i* misses feature 1. The distance between *sample i* and separating plane should be calculated only in existing features (i.e., d_i). However, in standard ELM, distance is calculated in full space. So d_i is incorrectly scaled up, that is, d'_i . Based on this observation, we are inspired to consider each sample in its own relevant subspace. This is just the essence of sample-based learning framework. Samples with missing features should be treated as residing in the relevant subspaces rather than in full space. The optimization formula of S-ELM is naturally derived as follows:

$$\min_{\beta} \|\beta^{(i)}\|_p^{\sigma_1} + C \times \|\mathbf{H}\beta^{(i)} - \mathbf{Y}\|_q^{\sigma_2}. \quad (11)$$

Compared with standard general ELM, the essential transformation is substitution original output weight β with sample-based output weight $\beta^{(i)}$. Specifically speaking, there are two objectives in minimization optimization formula (i.e., empirical risk and structural risk). Missing features affect the calculation of empirical risk (i.e., dot product of output weight and sample vector). Fortunately, considering distances in sample-based subspaces, we keep the two optimization objectives consistent by transforming β into $\beta^{(i)}$. Actually, $\beta^{(i)}$ is the β 's projection in *sample i*'s relevant subspace. For samples without missing feature, sample-based output weights are just the same with standard one.

4.2. S-ELM Classification Algorithm. According to sample-based learning framework, the optimization formula of sample-based ELM(S-ELM) classification is constructed as (12). Here, the extension is based on (4):

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i} \quad & \frac{1}{2} \|\beta^{(i)}\|^2 + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & \mathbf{y}_i * (\beta^{(i)T} \cdot \mathbf{x}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \quad (12)$$

As it can be seen, different missingness among samples may induce different $\beta^{(i)}$ s. In order to guarantee better generalization, the maximum of all $\beta^{(i)}$ s should be minimized. Consequently, (12) can be transformed into

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i} \quad & \left(\max_{1 \leq i \leq n} \frac{1}{2} \|\beta^{(i)}\|^2 \right) + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & \mathbf{y}_i * (\beta^{(i)^T} \cdot \mathbf{x}_i) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \quad (13)$$

By introducing an indication matrix $\mathbf{S}$ (each of its components $s_{i,j} \in \{0, 1\}^{n \times d}$ indicates whether the corresponding feature of sample is missing or not), (13) can be equivalently rewritten as

$$\begin{aligned} \min_{\beta, \xi_i} \quad & \max_{1 \leq i \leq n} \frac{1}{2} \sum_{p=1}^d \mathbf{S}(i, p) \beta_p^2 + C \sum_{i=1}^n \xi_i, \\ \text{s.t.} \quad & \mathbf{y}_i * \left(\sum_{p=1}^d \mathbf{S}(i, p) \beta_p \cdot \mathbf{x}_{i,p} \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \quad (14)$$

Since (14) is not a convex optimization problem, it is more difficult to be solved than standard ELM. Fortunately, we can transform it into a convex one (i.e., (15)) by using an auxiliary variable θ ($\theta = \max_{1 \leq i \leq n} (1/2) \sum_{p=1}^d \mathbf{S}(i, p) \beta_p^2$). Consider the following:

$$\begin{aligned} \min_{\beta, \xi_i, \theta} \quad & \theta + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \mathbf{y}_i * \left(\sum_{p=1}^d \mathbf{S}(i, p) \beta_p \cdot \mathbf{x}_{i,p} \right) \geq 1 - \xi_i, \\ & \theta \geq \frac{1}{2} \sum_{p=1}^d \mathbf{S}(i, p) \beta_p^2, \\ & \xi_i \geq 0. \end{aligned} \quad (15)$$

From (15), it can be found that (1) the objective is a quadratic function, which is convex; (2) the first constraint is linear in variables, which is also convex; and (3) the second constraint is a quadratic cone, which is again convex. Therefore, (15) is a convex optimization problem, which can be efficiently solved by many off-the-shelf convex optimization packages. In this paper, we use monqp package to solve it. Further, (15) is a quadratic constraint quadratic programming (QC-QP) problem; more advanced optimization techniques such as Nesterov's methods [26] can be applied to improve its computation efficiency.

4.3. S-ELM ϵ -Insensitive Regression Algorithm. In this section, we extend the ϵ -insensitive ELM [21] based on the sample-based learning framework. Based on (9), we substitute

standard output weight with the sample-based one. The resultant optimization formula is

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i, \xi_i^*} \quad & \frac{1}{2} \|\beta^{(i)}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*), \\ \text{s.t.} \quad & \mathbf{y}_i - \beta^{(i)^T} \mathbf{x}_i \leq \epsilon + \xi_i, \\ & \beta^{(i)^T} \mathbf{x}_i - \mathbf{y}_i \leq \epsilon + \xi_i^*. \end{aligned} \quad (16)$$

In (16), $\beta^{(i)}$ corresponding to different samples may have different components. So we minimize their maximum value. We change its optimization objective as (17). Then, we utilize the same transformation technique in Section 4.2 and derive the final convex optimization. In specific, we list the transformation process as follows:

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i, \xi_i^*} \quad & \left(\max_i \frac{1}{2} \|\beta^{(i)}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \right), \\ \text{s.t.} \quad & \mathbf{y}_i - \beta^{(i)^T} \mathbf{x}_i \leq \epsilon + \xi_i, \\ & \beta^{(i)^T} \mathbf{x}_i - \mathbf{y}_i \leq \epsilon + \xi_i^*, \end{aligned} \quad (17)$$

$$\begin{aligned} \min_{\beta^{(i)}, \xi_i, \xi_i^*} \quad & \theta + C \sum_{i=1}^n (\xi_i + \xi_i^*), \\ \text{s.t.} \quad & \mathbf{y}_i - \beta^{(i)^T} \mathbf{x}_i \leq \epsilon + \xi_i, \\ & \beta^{(i)^T} \mathbf{x}_i - \mathbf{y}_i \leq \epsilon + \xi_i^*, \\ & \theta \geq \frac{1}{2} \sum_{p=1}^d \mathbf{S}(i, p) \beta_p^2. \end{aligned} \quad (18)$$

From (18), we can see that (1) the objective is convex; (2) the first and third constraint is linear inequality which is convex; (3) the second constraint is a quadratic inequality which is again convex. Consequently, (18) is a convex optimization problem. It can be efficiently solved by many existing methods.

5. Experiments

In this section, We demonstrate the proposed algorithms over synthetic data sets, UCI benchmark data sets, and a real world fingerprint image data set to validate the effectiveness of S-ELM learning framework. Missing data used in experiments is produced artificially. In detail, we generate indication matrix (which is depicted in Section 4.2) for each raw data set by the same numbers of row and column. First, all the components of indication matrix are initialized with one. Then, according to specific missing ratio, some randomly chosen components are set to zero. Note that we make sure there is no single row or column to be set all to zeros. For the same data set, indication matrixes are produced for training and testing data, respectively.

To validate the effectiveness of the proposed algorithms, we compare the performance of proposed algorithms with

TABLE 1: The UCI data sets used in classification.

Dataset	Number of samples	Number of features
Breast cancer	286	9
Diabetes	768	20
Flare solar	1389	10
German	1000	20
Splice	3190	61
Waveform	5000	21

two imputation approaches (i.e., zero-filling and mean-filling). In specific, S-ELM learning algorithms run on the data set with missing data while standard ELM algorithms run in the imputed data set. For abbreviation, ZF-ELM represents standard ELM algorithm running in zero-filling data set, and MF-ELM represents standard ELM algorithm running in mean-filling data set.

5.1. Experiment Settings. All our experiments are implemented on MATLAB 2013b environment running in Core 3.0 GHz CPU and 8 GB RAM. Prior to each algorithm execution, we do some data preprocessing. First, we normalize data sets to $[0, 1]$. Then, we permute data sets randomly and divide them into training data and testing data (#train/#test equals to 3/2). In order to eliminate randomness, final results are the average of fifty times repetitions. We also calculate the standard deviation to show the stability of algorithms.

5.2. Results of S-ELM Classification. In this section, we use aggregated classification accuracy as the standard to measure the learning precision of different algorithms. Missing ratios are between 0.1 and 0.6 with 0.1 interval. For each algorithm and each data set, the regularization factor C is chosen by fivefold cross validations. The UCI benchmark data sets are varied in the number of features and samples; detailed information is shown in Table 1.

The classification accuracy of three algorithms is plotted in Figure 5. As it can be observed, the curves corresponding to S-ELM are above other two in most cases, indicating its superior performance. Furthermore, Table 2 lists the average accuracy and standard deviation. In general, for various UCI benchmark data sets, the S-ELM classification algorithm achieves better accuracy and stability over different missing ratios. The results are consistent with our analysis in Section 4. The two imputation methods classify all samples in full feature space, while S-ELM learning assumes samples with the different vector components lying in their own relevant feature spaces. Particularly, compared with original ELM, S-ELM classification algorithm does not bring extra computational cost.

5.3. Fingerprint Classification. As an application, we evaluate our methods in a real world problem, that is, automatic fingerprint classification, which is an effective index scheme for reducing matching candidate in fingerprint identification. Through gross classification, a query finger is only compared with samples of the same class rather than all samples in the

TABLE 2: Classification performance comparisons over different missing ratios on the UCI benchmark data sets. Performance includes mean aggregate accuracy $\pm$ standard deviation and training time (in seconds).

	S-ELM	ZF-ELM	MF-ELM
Breast cancer	0.7185 ± 0.0151 0.3473	0.6570 ± 0.0334 0.2776	0.6901 ± 0.0183 0.2947
Diabetes	0.7375 ± 0.0263 0.9315	0.6852 ± 0.0467 0.7257	0.6957 ± 0.0448 0.7112
Flare solar	0.5635 ± 0.0217 2.938	0.5369 ± 0.0348 2.108	0.5454 ± 0.0441 2.272
German	0.7624 ± 0.0250 3.748	0.6984 ± 0.0301 2.327	0.7150 ± 0.0257 2.374
Splice	0.6704 ± 0.0602 10.83	0.5734 ± 0.0275 11.26	0.5916 ± 0.0273 9.273
Waveform	0.8287 ± 0.0236 120.5	0.7206 ± 0.0271 99.26	0.7456 ± 0.0314 108.5

whole database. It reduce computational complexity greatly. In practice, there are some meaningless features in fingerprint data set due to distortion, wet and dry impression, and so forth. The meaningless features can be seen as a kind of missing data. Therefore, we apply the S-ELM classification algorithm to this application.

The fingerprint image data set used in our experiment contains 2000 samples. First, we categorize all those fingerprints into three basic patterns manually: loop, whorl, and arch. All fingerprint images are 8-bit gray-level bmp files and the image resolution is $328 * 356$. Then, we use the first 1000 pairs of fingerprints for training and the second 1000 pairs of fingerprints for testing. Through fingerprint image preprocessing (e.g., edge detection, segmentation, feature, thinning, and extraction), each fingerprint image is expressed as a feature vector of 400 dimensions.

Too low accuracy does not make sense in the application of fingerprint classification. We set missing ratios in a relatively low range, that is, 0.1, 0.2, and 0.3. Besides, zero filling is meaningless. We only take MF-ELM as comparison. The fingerprint data set contains three classes of samples. We use one-versus-rest method to classify them, respectively; that is, we take a target from three classes each time. All the classification accuracy list in Table 3 is the average values of ten times execution results. It can be observed that (1) the classification accuracy of fingerprint classification seriously affected by missing data; (2) the computational cost of different missing ratios are basically the same; (3) in almost all conditions, S-ELM classification algorithm performs more accurately and stably than MF-ELM with almost the same computational cost.

5.4. Results of S-ELM ϵ -Insensitive Regression. In this section, we evaluate the S-ELM ϵ -insensitive regression algorithm in synthetic data sets and UCI benchmark data sets. Root mean

FIGURE 5: Classification accuracy comparison in the UCI data sets.

TABLE 3: Performance comparison of S-ELM and MF-ELM over different missing ratios on the fingerprint data set.

Target class	Missing ratio	S-ELM		MF-ELM	
		Accuracy $\pm$ std.	Time (Sec.)	Accuracy $\pm$ std.	Time (Sec.)
Loop	10%	0.8256 $\pm$ 0.0736	89.37	0.7967 $\pm$ 0.0985	85.72
	20%	0.8029 $\pm$ 0.1253	85.91	0.7265 $\pm$ 0.1371	86.28
	30%	0.7694 $\pm$ 0.1191	90.43	0.6303 $\pm$ 0.1039	87.93
Whorl	10%	0.8137 $\pm$ 0.0627	67.28	0.8129 $\pm$ 0.0922	60.72
	20%	0.7436 $\pm$ 0.0963	71.32	0.7269 $\pm$ 0.1167	67.49
	30%	0.7102 $\pm$ 0.0849	69.09	0.6359 $\pm$ 0.1076	68.25
Arch	10%	0.8361 $\pm$ 0.0852	102.7	0.8057 $\pm$ 0.0937	94.73
	20%	0.7659 $\pm$ 0.0974	98.31	0.7561 $\pm$ 0.1170	96.83
	30%	0.7192 $\pm$ 0.0827	100.3	0.6574 $\pm$ 0.1207	96.67

TABLE 4: Synthetic data sets and corresponding functions.

Data set	Function
Synthetic data set 1	$f_1(x) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$
Synthetic data set 2	$f_2(x) = x_1 \times x_2 \times x_3 + x_2 \times x_3 \times x_4$
	$+ x_3 \times x_4 \times x_5 + x_4 \times x_5 \times x_6$
Synthetic data set 3	$f_3(x) = 2^{x_1} + 2^{x_2} + 2^{x_3} + 2^{x_4} + 2^{x_5} + 2^{x_6}$
Synthetic data set 4	$f_4(x) = \frac{x_1 + x_2 + x_3}{x_4 + x_5 + x_6}$

square error (RMSE) is used as the measurement to evaluate the prediction accuracy of algorithms. It is calculated as

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}}. \quad (19)$$

We generate synthetic data sets by combinations of basic algebraic operation functions. For each function, all independent variables' domains are in $[0, 1]$, and their intervals are randomly different. The detail is given in Table 4. As for S-ELM ϵ -insensitive regression, ϵ is set to be 0.01 empirically.

From Figure 6, it can be observed that the S-ELM ϵ -insensitive regression algorithm achieves better prediction accuracy in different missing ratios. Specifically, S-ELM performs better than ZF-ELM, which proves that S-ELM reduces generalization error by using sample-based output weight vector norm. S-ELM beats MF-ELM with comparatively less obvious advantage. It shows that mean value is closer to missing value than zero.

To further illustrate the advantage of the S-ELM ϵ -insensitive regression algorithm, we use UCI benchmark data sets for comparison. Table 5 specifies the UCI benchmark data sets used in our experiments. Those four data sets are varied in scales.

Mean RMSE and standard deviation are reported in Table 6. S-ELM regression algorithm performs better than ZF-ELM and MF-ELM over different missing ratios. As Figure 7 shows, with the missing ratio increasing, both ZF-ELM and MF-ELM's prediction accuracies decrease rapidly,

TABLE 5: The UCI benchmark data sets for regression.

Dataset	# train	# test	# features
Body fat	152	100	14
Housing	304	202	13
Pyrim	45	29	27
Abalone	2507	1670	8
Bike sharing	439	293	6
Airfoil self-noise	901	602	5

TABLE 6: Performance comparisons with UCI benchmark data sets. The two rows of each cell represent mean RMSE $\pm$ standard deviation over different missing ratios and training time (in seconds).

	S-ELM	ZF-ELM	MF-ELM
Body fat	0.1299 $\pm$ 0.0905	0.1781 $\pm$ 0.1007	0.1457 $\pm$ 0.1088
	0.1560	0.1758	0.1872
Housing	0.2229 $\pm$ 0.1019	0.3234 $\pm$ 0.1697	0.2511 $\pm$ 0.1261
	0.5628	0.3432	0.3869
Pyrim	0.3816 $\pm$ 0.0842	0.5603 $\pm$ 0.2253	0.4537 $\pm$ 0.1396
	0.0624	0.1008	0.0926
Abalone	0.2336 $\pm$ 0.1083	0.3004 $\pm$ 0.1537	0.2643 $\pm$ 0.1193
	213.5	177.6	182.4
Bike sharing	0.3886 $\pm$ 0.1020	0.4662 $\pm$ 0.1596	0.3977 $\pm$ 0.1218
	7.263	7.109	6.993
Airfoil self-noise	0.3765 $\pm$ 0.1591	0.4563 $\pm$ 0.2470	0.3810 $\pm$ 0.1928
	52.38	50.74	51.61

while S-ELM decreases mildly. The most obvious advantage of S-ELM appears in Pyrim data set, especially when missing ratio is high. It proves that it is more difficult for imputation approaches to estimate the true value of missing features in small scale data set.

FIGURE 6: RMSE comparisons in the synthetic data sets.

6. Conclusions and Future Work

In this paper, we propose a sample-based learning framework for ELM in order to cope with the problem of missing data. Further, we extend ELM classification and ϵ -insensitive ELM regression in this framework. For the proposed algorithms, we transform their formulations into convex optimizations. We compare proposed algorithms with two widely used imputation methods in synthetic, UCI benchmark data sets and a real world fingerprint image data set. Result shows

that S-ELM achieves better classification and regression accuracy and stability without introducing extra computational complexity. Particularly, S-ELM shows remarkable advantage when the missing ratio is relatively high. Many following works are worth exploring. In the next, we are going to extend more ELM algorithms based on sample-based learning framework. Meanwhile, we will handle missing data in feature selection to improve learning accuracy [27]. Further, we will consider missing data in multiple data sources learning.

FIGURE 7: RMSE comparisons in the UCI benchmark data sets.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by the Major State Basic Research Development Program of China (973 Program) under the Grant no. 2014CB340303, the National High Technology Research and Development Program of China (863 Program) under Grant no. 2013AA01A213, and the National Natural Science Foundation of China under Grant no. 61402490.

References

- [1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 985–990, Budapest, Hungary, July 2004.
- [2] G.-B. Huang, D. H. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107–122, 2011.
- [3] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [4] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pp. 389–395, Nashville, Tenn, USA, April 2009.
- [5] J. M. Martínez-Martínez, P. Escandell-Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp. 3716–3721, 2011.
- [6] J. Luo, C.-M. Vong, and P.-K. Wong, "Sparse bayesian extreme learning machine for multi-classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 836–843, 2014.
- [7] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization," *Neurocomputing*, vol. 74, no. 16, pp. 2413–2421, 2011.
- [8] A. Mozaffari and N. L. Azad, "Optimally pruned extreme learning machine with ensemble of regularization techniques and negative correlation penalty applied to automotive engine coldstart hydrocarbon emission identification," *Neurocomputing*, vol. 131, pp. 143–156, 2014.
- [9] Y. Lan, Y. C. Soh, and G.-B. Huang, "Two-stage extreme learning machine for regression," *Neurocomputing*, vol. 73, no. 16–18, pp. 3028–3038, 2010.
- [10] X. Liu, L. Wang, G.-B. Huang, J. Zhang, and J. Yin, "Multiple kernel extreme learning machine," *Neurocomputing*, vol. 149, part A, pp. 253–264, 2015.
- [11] N.-Y. Liang, G.-B. Huang, H.-J. Rong, P. Saratchandran, and N. Sundararajan, "A fast and accurate on-line sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [12] Y. Lan, Y. C. Soh, and G.-B. Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009.
- [13] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.
- [14] Z. Deng, K.-S. Choi, L. Cao, and S. Wang, "T2fela: type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 4, pp. 664–676, 2014.
- [15] Q. He, T. Shang, F. Zhuang, and Z. Shi, "Parallel extreme learning machine for regression based on mapreduce," *Neurocomputing*, vol. 102, pp. 52–58, 2013.
- [16] B. M. Marlin, *Missing data problems in machine learning [Ph.D. thesis]*, University of Toronto, 2008.
- [17] P. Royston, "Multiple imputation of missing values," *Stata Journal*, vol. 4, pp. 227–241, 2004.
- [18] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, no. 16–18, pp. 3460–3468, 2008.
- [19] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [20] X. Liu, L. Wang, J. Yin, and L. Liu, "Incorporation of radius-info can be simple with SimpleMKL," *Neurocomputing*, vol. 89, pp. 30–38, 2012.
- [21] S. Balasundaram, "On extreme learning machine for ϵ -insensitive regression in the primal by Newton method," *Neural Computing and Applications*, vol. 22, no. 3–4, pp. 559–567, 2013.
- [22] J. Goeman, R. Meijer, and N. Chaturvedi, "L1 and L2 penalized regression models," cran.r-project.org, 2012.
- [23] J. A. R. Little, "Regression with missing x's: a review," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 1227–1237, 1992.
- [24] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, and D. Koller, "Max-margin classification of data with absent features," *The Journal of Machine Learning Research*, vol. 9, pp. 1–21, 2008.
- [25] N. J. Horton and S. R. Lipsitz, "Multiple imputation in practice: comparison of software packages for regression models with missing variables," *The American Statistician*, vol. 55, no. 3, pp. 244–254, 2001.
- [26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87 of *Applied Optimization*, Springer, Amsterdam, The Netherlands, 2004.
- [27] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1083–1095, 2014.