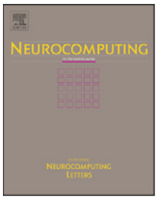
Neurocomputing 102 (2013) 45–51

Contents lists available at SciVerse ScienceDirect

Neurocomputing

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



Regularized extreme learning machine for regression with missing data

Qi Yu [a,*], Yoan Miche [a], Emil Eirola [a], Mark van Heeswijk [a], Eric Séverin [b], Amaury Lendasse [a,c,d]

[a] Department of Information and Computer Science, Aalto University, Espoo, 02150, Finland
[b] LEM, Université Lille 1, 59043 Lille cedex, France
[c] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[d] Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia-San Sebastián, Spain

ARTICLE INFO

Available online 5 June 2012

Keywords:
ELM
Ridge regression
Tikhonov regularization
LARS
Missing data
Pairwise distance estimation

ABSTRACT

This paper proposes a method which is the advanced modification of the original extreme learning machine with a new tool for solving the missing data problem. It uses a cascade of L_1 penalty (LARS) and L_2 penalty (Tikhonov regularization) on ELM (TROP-ELM) to regularize the matrix computations and hence makes the MSE computation more reliable, and on the other hand, it estimates the expected pairwise distances between samples directly on incomplete data so that it offers the ELM a solution to solve the missing data issues. According to the experiments on five data sets, the method shows its significant advantages: fast computational speed, no parameter need to be tuned and it appears more stable and reliable generalization performance by the two penalties. Moreover, it completes ELM with a new tool to solve missing data problem even when half of the training data are missing as the extreme case.

1. Introduction

Missing data problem [1,2] is very common to confront for many different research fields, for example, data from surveys, experiments, observational studies, etc., typically contain missing values. Because most analysis procedures were not designed to handle incomplete data, and researchers often resort to editing procedures (deleting incomplete cases, replacing the missing values[1] with sample means, etc.) to lend an appearance of completeness. A method for inference from incomplete data was only developed in 1976. Immediately afterwards, Dempster et al. invented the expectation maximization (EM) algorithm that resulted in the use of the maximum likelihood (ML) methods for missing data estimation [3]. Barely a decade later, Lit et al. did acknowledge the limitations of case deletion and single imputations and then introduced multiple imputations [4]. Multiple imputations would not have been achievable without parallel progress in computational power because generally they are computationally expensive [5–8].

On the other hand, data sets in many research fields become larger and larger, which are very time consuming when using some classic methods to deal with, like support vector machine, multi-layer neural network, etc. In this sense, extreme learning machine (ELM) is a competitively good solution for such tasks. ELM as presented by Huang et al. in [9,10] is fast enough to accommodate relatively large data sets, where other traditional machine learning techniques have very large computational times. The main idea lying in ELM is the random weights of a single hidden layer feedfoward neural network (SLFN). The essence of ELM is that the hidden layer of SLFNs need not be tuned. Compared with those traditional computational intelligence techniques, ELM provides better generalization performance at a much faster learning speed and with least human intervention [11–14].

Since learning in the presence of missing data is pervasive problems in machine learning and statistical data analysis, we propose to extend ELM, particular TROP-ELM [15] in order to handle missing data. The goal of using TROP-ELM is to take all the advantages of ELM like speed, and at the same time, the method needs to be robust and more reliable. That is why we need the double regularization. Indeed, it is shown in [16] that using TROP-ELM, the generalization performances of the ELM models are improved, the complexity of ELM model is decreased. The double regularization included in the TROP-ELM is suboptimal because they are done sequentially and not simultaneously but it has the advantage to keep the computational time comparable to the computational time of the original ELM or the single-regularized OP-ELM [15]. As to the missing data part, our method only has to calculate the distances between samples, instead of the traditional imputation methods which normally increase a lot the complexity.

[*] Corresponding author. Tel.: +35 8442519088.
E-mail address: qi.yu@aalto.fi (Q. Yu).

[1] Missing data, or missing values, occur when no data value is stored for the variable in the current observation. If input data has N observations (samples) with d dimensions (variables), then, when we refer to a missing data in these data, it implies one missing point among the original ($N \times d$) points.

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.02.040

In a word, this paper proposes a method which uses the advanced modification of the original extreme learning machine with a new tool to solve the missing data problem. In Section 2, the tool used to solve MD problem is introduced as well as some general discussion on missing data. Section 3 shows the details of the double-regularized ELM using LARS and Tikhonov regularization. The entire method is summarized in Section 4 with several major steps, and followed by experiments in Section 5 and a short conclusion in Section 6.

2. Pairwise distance estimation with missing data (MD)

Missing data (MD) are a part of almost all research, and researchers have to decide how to deal with it from time to time. There are a number of alternative ways of dealing with missing data, and in this section, a pairwise distance estimation is highlighted and introduced to solve the MD problem.

2.1. Nature of missing data

When confronting the missing data, the common question you may ask is why and how they are distributed. Well, the nature of missing data can be categorized into three main types [17]:

- *Missing completely at random* (*MCAR*) [18]: When we say that data are missing completely at random, we mean that the probability that an observation (X_i) is missing is unrelated to the value of X_j or to the value of any other variables. Thus, a nice feature of data which are MCAR is the analysis remains unbiased. We may lose power for our design, but the estimated parameters are not biased by the absence of data.
- *Missing at random* (*MAR*): Often data are not missing completely at random, but they may be classifiable as missing at random if the missingness does not depend on the value of X_i after controlling for another variable. The phraseology MAR is a bit awkward because we tend to think of randomness as not producing bias, and thus might well think that missing at random is not a problem. Unfortunately it is a problem, although we have ways of dealing with the issue so as to produce meaningful and relatively unbiased estimates [19].
- *Missing not at random* (*MNAR*): If data are not missing at random or completely at random then they are classed as missing not at random (MNAR). When we have data that are MNAR we have a problem. The only way to obtain an unbiased estimate of parameters is to model missingness. In other words we would need to write a model that accounts for the missing data. Therefore, MNAR is not covered in this paper. This paper focuses on developing the method to solve the MD problem using extreme learning machine, rather than to analyze the data of any specific field or MD for any specific reasons.

2.2. Existing approaches for MD problem

By far the most common approach is to simply omit those observations with missing data and to run the analysis on what remains. This is so-called listwise deletion. Although listwise deletion often results in a substantial decrease in the sample size available for the analysis, it does have important advantages. In particular, under the assumption that data are missing completely at random, it leads to unbiased parameter estimates.

Another branch of approach is imputation, meaning to substitute the missing data point with a estimated value. A once common method of imputation was Hot-deck imputation where a missing value was imputed from a randomly selected similar record [20]. Besides, mean substitution method uses the idea of substituting a mean for the missing data [21], etc.

There are also some advanced methods such as maximum likelihood and multiple imputation [22,23]. There are a number of ways to obtain maximum likelihood estimators, and one of the most common is called the expectation–maximization algorithm (EM). This idea is further extended in expectation conditional maximization (ECM) algorithm [24]. ECM replaces each M-step with a sequence of conditional maximization (CM) steps in which each parameter θ_i is maximized individually, conditionally on the other parameters remaining fixed. In the following paragraph, a distance estimation method is presented based on ECM.

2.3. Pairwise distance estimation

Pairwise distance estimation efficiently estimates the expectation of the squared Euclidean distance between observations in data sets with missing data [25]. Therefore, in general, it can be embedded into any distance-based method, like k nearest neighbors, support vector machine (SVM), multidimensional scaling (MDS), etc., to solve missing data problem.

Given two samples x and y with missing values, in a d-dimensional space. Denote by $M_x,M_y \subseteq [d]=1,\ldots,d$ the indexes of the missing components in the two samples. Here we assume the data are MCAR or MAR, that is, the missing value can be modeled as random variables, $X_i, i \in M_x$ and $Y_i, i \in M_y$. Thus

$$x'_i = \begin{cases} E[X_i|x_{obs}] & \text{if } i \in M_x, \\ x_i & \text{otherwise,} \end{cases} \tag{1}$$

$$y'_i = \begin{cases} E[Y_i|y_{obs}] & \text{if } i \in M_y, \\ y_i & \text{otherwise.} \end{cases} \tag{2}$$

Where x' and y' is the imputed version of x and y which the missing value has been replaced by its conditional mean. The corresponding conditional variance becomes

$$\sigma^2_{x,i} = \begin{cases} Var[X_i|x_{obs}] & \text{if } i \in M_x, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

$$\sigma^2_{y,i} = \begin{cases} Var[Y_i|y_{obs}] & \text{if } i \in M_y, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

Then, the expectation of the squared distance can be expressed as

$$E[\|x-y\|^2] = \sum_i ((x'_i - y'_i)^2 + \sigma^2_{x,i} + \sigma^2_{y,i}), \tag{5}$$

or, equivalently,

$$E[\|x-y\|^2] = \|x'-y'\|^2 + \sum_{i \in M_x} \sigma^2_{x,i} + \sum_{i \in M_y} \sigma^2_{y,i}. \tag{6}$$

According to Eirola [25], covariance matrix can be achieved through the ECM (expectation conditional maximization) method provided in the MATLAB Financial Toolbox [26], implementing the method of [24] with some improvements by [27], which makes the calculation of conditional means and variances of the missing elements possible. Therefore, each pairwise squared distance can be calculated with the missing values replaced by their respective conditional means and by adding the sum of the conditional variances of the missing values, respectively.

Since this algorithm is suitable for methods which rely only on the distance between samples, in this paper, we use this estimation algorithm embedded extreme learning machine to solve missing data problem.

3. Double-regularized ELM: TROP-ELM

Miche et al. in [16] proposed a double regularized ELM algorithm, which uses a cascade of two regularization penalties: first a L_1 penalty to rank the neurons of the hidden layer, followed by a L_2 penalty on the regression weights (regression between hidden layer and output layer). This section introduces this algorithm briefly.

3.1. Extreme learning machine (ELM)

The extreme learning machine algorithm is proposed by Huang et al. in [9] as an original way of building a single hidden layer feedforward neural network (SLFN). The essence of ELM is that its the hidden layer of needs not to be iteratively tuned [14,9], and moreover, the training error $\|\mathbf{H}\beta - \mathbf{y}\|$ and the norm of the weights $\|\beta\|$ are minimized.

Given a set of N observations (x_i, y_i), $i \leq N$. with $x_i \in \mathbf{R}^p$ and $\mathbf{y} \in \mathbf{R}$. A SLFN with m hidden neurons in the middle layer can be expressed by the following sum:

$$\sum_{i=1}^{m} \beta_i f(\omega_i x_j + b_i), \quad 1 \leq j \leq N, \tag{7}$$

where β_i is the output weights, f be an activation function, ω_i is the input weights and b_i is the biases. Suppose the model perfectly describe the data, the relation can be written in matrix form as $\mathbf{H}\boldsymbol{\beta} = \mathbf{y}$, with

$$\mathbf{H} = \begin{pmatrix} f(\omega_1 x_1 + b_1) & \dots & f(\omega_m x_1 + b_m) \\ \vdots & \ddots & \vdots \\ f(\omega_1 x_n + b_1) & \dots & f(\omega_m x_n + b_m) \end{pmatrix}, \tag{8}$$

$\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^T$ and $\mathbf{y} = (y_1, \dots, y_n)^T$. The ELM approach is thus to initialize randomly the ω_i and b_i and compute the output weights $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{y}$ by a Moore–Penrose pseudo-inverse [28].

The significant advantages of ELM are its extreme fast learning speed, relative better generalization performance while being a simple method [9]. There has been recent advances based on the ELM algorithm, to improve its robustness (OPELM [15], CS-ELM [29]), or make it a batch algorithm, improving at each iteration (EM-ELM [30], EEM-ELM [31]).

3.2. L_1 penalty: LASSO

An important part in ELM is to minimize the training error $\|\mathbf{H}\beta - \mathbf{y}\|$, which is an ordinary regression problem. One technique to solve this is called Lasso, for 'least absolute shrinkage and selection operator' proposed by Tibshirani [32].

Lasso solution minimizes the residual sum of squares, subject to the sum of the absolute value of the coefficients being less than a constant, that's why it is also called 'L_1 penalty'. The general form which Lasso works on is

$$\min_{\lambda,\omega} \left(\sum_{i=1}^{N} (y_i - \mathbf{x_i}\boldsymbol{\omega})^2 + \lambda \sum_{j=1}^{p} |\omega_j| \right). \tag{9}$$

Because of the nature of the constant, Lasso tends to produce some coefficients that are exactly 0 and hence give interpretable models. The shrinkage is controlled by parameter λ. The smaller λ is, the more ω_j coefficients are zeros and hence less variables are retained in the final model.

Computation of Lasso solution is a quadratic programming problem, and can be tackled by standard numeral analysis algorithms. However, a more efficient computation approach is developed by Efron et al. in [33], called least angle regression (LARS). LARS is similar to forward stepwise regression, but instead of including variables at each step, the estimated parameters are increased in a direction equiangular to each one's correlations with the residual. Thus, it is computationally just as fast as forward selection. If two variables are almost equally correlated with the response, then their coefficients should increase at approximately the same rate. The algorithm thus behaves as intuition would expect, and also is more stable. Moreover, LARS is easily modified to produce solutions for other estimators, like the Lasso, and it is effective when the number of dimensions is significantly greater than the number of samples [33].

The disadvantages of the LARS method are that it has problem with highly correlated variables, even though this is not unique to LARS. This problem is discussed in detail by Weisberg in the Discussion section of the paper [33]. To overcome this, next paragraph introduces Tikhonov Regularization method.

3.3. L_2 penalty: Tikhonov regularization

Tikhonov regularization, named for Andrey Tychonoff, is the most commonly used method of regularization [34]. In statistics, the method is also known as ridge regression.

The general form of Tikhonov regularization is to minimize

$$\min_{\lambda,\omega} \left(\sum_{i=1}^{N} (y_i - \mathbf{x_i}\boldsymbol{\omega})^2 + \lambda \sum_{j=1}^{p} \omega_j^2 \right). \tag{10}$$

The idea behind of Tikhonov regularization is at the heart of the "bias-variance tradeoff" issue, thanks to it, the Tikhonov regularization achieves better performance than the traditional OLS solution. Moreover, it outperforms the Lasso solution in cases that the variables are correlated. One advantage of the Tikhonov regularization is that it tends to identify/isolate groups of variables, enabling further interpretability.

One big disadvantage of the ridge-regression is that it does not have sparseness in the final solution and hence, it does not give an easily interpretable result. Therefore, a new idea is created to use a cascade of the two regularization penalties, which is introduced in the next paragraph.

3.4. TROP-ELM

Miche et al. in [15] proposed a method OP-ELM, which uses LARS to rank the neurons of the hidden layers in ELM and select the optimal number of neurons by leave-one-out (LOO). One problem with LOO error is that it can be very time consuming, especially when the data has large number of samples. Fortunately, the PREdiction Sum of Squares (PRESS) statistics provide a direct and exact formula for the calculation of the LOO error for linear models [35]

$$\epsilon^{PRESS} = \frac{y_i - h_i b_i}{1 - h_i P h_i^T}, \tag{11}$$

where P is defined as $P = (H^T H)^{-1}$ and H is the hidden layer output matrix. It can be also expressed as

$$\epsilon^{PRESS} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - x_i (X^T X)^{-1} x^T y}{1 - x_i (X^T X)^{-1} x_i^T} \right)^2, \tag{12}$$

which means that each observation is estimated using the other $N-1$ observations and the residuals are finally squared and summed up. The main drawback of this approach lies in the use of a pseudo-inverse in the calculation, which can be lead to numeral instabilities if the data set X is not full rank. This is happen very often in the real world data. Thus, a Tikhonov-regularized version of PRESS is created

$$\epsilon^{PRESS}(\lambda) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - x_i(X^TX+\lambda I)^{-1}x^Ty}{1-x_i(X^TX+\lambda I)^{-1}x_i^T}\right)^2. \tag{13}$$

This new modified version uses the singular value decomposition (SVD) approach [36] of X to avoid computational issues, and introduces the Tikhonov regularization parameter in the calculation of the pseudo-inverse by the SVD. In practice, the optimization of λ in this method is performed by a Nelder–Mead [37] minimization approach, which converges quickly on this problem.

In general, TROP-ELM is an improvement of original ELM. It first constructs a SLFN like ELM, then ranks the best neurons by LARS (L_1 regularization), finally selects the optimal number of neurons by TR-PRESS (L_2 regularization).

4. The entire methodology

In this section, the general methodology is presented as well as the details of the implementation steps.

Fig. 1 illustrates the main components of the whole algorithm, and how they connected. Therefore, when confronting a regression problem with incomplete data, there are several steps to follow in order to implement this method:

- First of all, it is necessary to replace the missing values with their respective conditional means mentioned in Section 2.3. This is a so-called ‘imputation’ step. The reason of this move is because we want to make the whole method more robust.
  Thus, the accuracy of the distances calculated afterwards is not really based on these imputed values. The main purpose here is to make it possible to use Gaussians as the active function in ELM. Next step explains more about why the imputation is done at the beginning.
- Second, we decide to use Gaussian as the active function of the hidden node to build the single layer feedforward network. Then, m samples are randomly selected from original N samples ($m \leq N$) as the center of Gaussians, that is why the imputation is done in the first step. Choosing the randomly selected samples as the center could anyway guarantee the neural network built here adjoin the data. Therefore, when calculating the output of each neuron, the squared distance between each sample and the selected ones is needed, which is exactly the same thing the pairwise squared distance estimation method achieved. The hidden node parameters (σ^2,μ) are randomly generated, which remain the advantage of ELM that the parameters in hidden layer need not to be tuned. More specifically, parameter σ^2 is chosen from a interval (20–80%) of the original random generations, to further make sure that the model surrounds the data.
- When the distance matrix is ready (by pairwise distance estimation), with the random generated parameter (σ^2,μ), it is easy to compute the outputs of all the neurons in the hidden layer. The next step would be to figure out the weights ($\boldsymbol{\beta}$) between hidden layer and the output of the data (Y).
- The assumption to use LARS is that the problem to be solved should be linear. In fact, this is exactly the case when the neural network built in previous step, the relationship between the hidden layer and the output in ELM is linear. Therefore, LARS is used to rank neurons according to the output.
- Finally, as mentioned in Section 3.4, TR-PRESS is used to select the optimal number of neurons, mean square error is minimized though the optimization of parameter λ in Eq. (13).

The entire algorithm inherits most of the advantage of original ELM, fast computational speed, no parameter need to be tuned, comparatively high generalization performance, etc. Moreover, it perfects ELM with a new tool to solve missing data problem and offers more stable and accurate results with double regularization method.

5. Experiments

In order to test the proposed method for regression problem, five data sets are chosen in this paper to evaluate the method. These data sets can be found from UCI machine learning repository for free [38].

Table 1 shows the specification of the five selected data sets.

On the other hand, how to get a more general performance of the model remains to be a problematic issue. A common solution is to split the whole data set into training, validating and testing sets, which is a good practice. In this paper, we only need to separate training and testing set because leave-one-out validation is used with the training set, i.e. the error we get from the training

Table 1
Specification of the five tested regression data sets.

Data sets	# Attributes	# Training data	# Testing data
Ailerons	5	4752	2377
Elevators	6	6344	3173
Bank	8	2999	1500
Stocks	9	633	317
Boston Housing	13	337	169

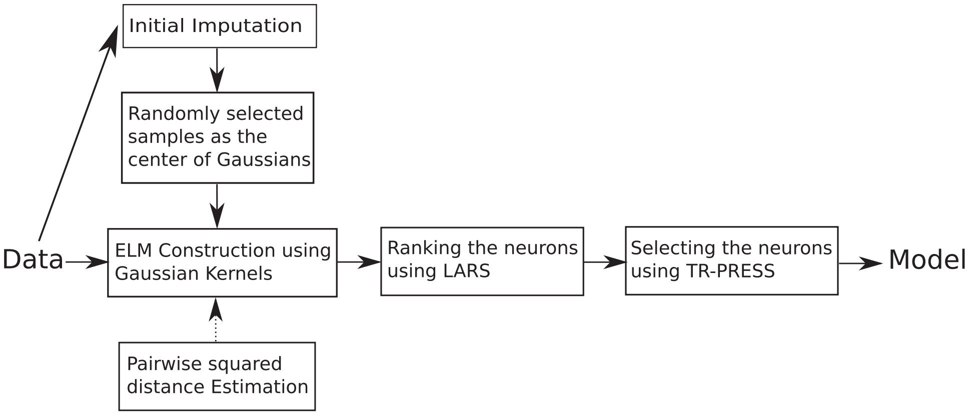

Initial Imputation
Randomly selected samples as the center of Gaussians
Data
ELM Construction using Gaussian Kernels
Ranking the neurons using LARS
Selecting the neurons using TR-PRESS
Model
Pairwise squared distance Estimation

Fig. 1. The framework of the proposed method.

set is actually the validation error. Furthermore, Monte-Carlo method is performed to split the data in order to reduce the effect of limited data size.

5.1. *Generating the missing data*

There is no missing value originally in these five data sets. Therefore, missing data are artificially created in each data set, in order to test the performance on incomplete data with the method. More precisely, the missing data are created (same as deleting the existing data) at randomly position once 1/200 of the total points till only half data points left. For example, if we have training set with N observations and d features ($N \times d$ data point totally), missing data are created $(N \times d)/200$ at a time, and continue 100 times till there is only half data points left $((N \times d)*100/200)$. Thus, the model is trained and tested 100 times which is so-called one round of the experiments.

5.2. *Monte-Carlo split for preprocessing*

Monte-Carlo methods [39] refer to various techniques. In this paper, Monte-Carlo methods are used to preprocess the data, aiming to two tasks. First, training sets are drawn randomly about two-thirds of the whole data sets, the rest one-third leaves for test set. Second, this Monte-Carlo preprocessing is repeated many times for each data set independently. Therefore, after these rounds of training and testing, an average test error is computed to represent the more general performance of the method.

5.3. *Other methods used in this paper*

For comparison, mean imputation and 1-nearest neighbor (1-NN) imputation [40,41] combined with TROP-ELM are tested in this paper. Specifically, in the mean imputation method, the mean of corresponding variable is calculated based on the existed samples to replace the missing data; in the 1-NN imputation method, the missing data are replaced by the corresponding variable of its first nearest neighbor whose value is the not missing. Therefore, pairwise distance estimation (PDE), mean imputation (mean) and 1-nearest neighbor imputation (1-NN) are used as three different tools here for TROP-ELM to solve the MD problem.

Moreover, this paper also tests all the incomplete data sets using TROP-ELM without any MD tools, that means, those samples which contain missing variables are removed (deleted) in order to perform normal TROP-ELM. The main drawback of this method is the huge loss of the training samples. Since the data are missing at random, so when the number of missing points is larger than the sample size, the worst case may happen that no samples left for training. Especially when the percentage of the missing data in the training sets continues to increase, this may happen more and more often. This kind of phenomenon can be seen in the following experiments results.

5.4. *Experiments results*

For each data set, the same experiment procedure is done to evaluate the method. First, Monte-Carlo split is performed for hundreds of rounds, then for each Monte-Carlo split, missing values are added to training part set by set for 100 times till half of the training values are missing. Once the new missing values are added, the model is trained and tested respectively. Thus, LOO and test results are calculated 100 times with different amount of missing value. In other words, for each different amount of missing value, the mean LOO errors and test errors are recorded for hundreds of rounds from those Monte-Carlo splits. All the results shown here are the normalized results.

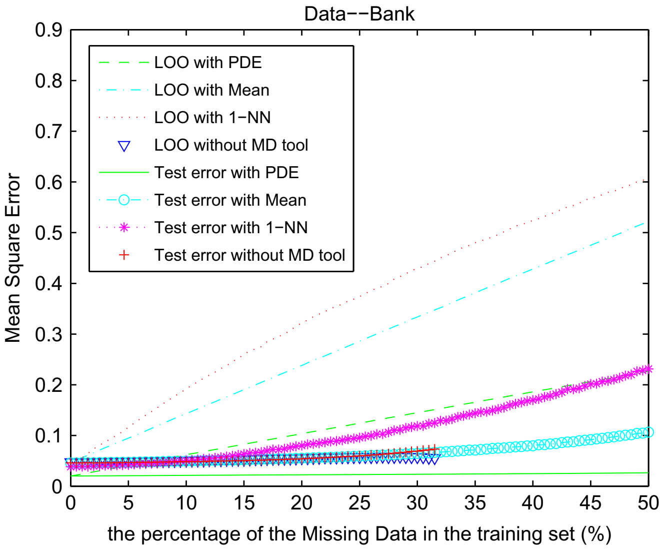


Fig. 2. Normalized MSE for the data set.

Take the bank data for instance. There are 4499 samples and eight variables originally in these data, and one output. For each Monte-Carlo split, 2999 samples are randomly selected for training, and the rest for testing. As to the training set, $(2999 \times 8)/200 \approx 120$ data points are added continuously for 100 times, meaning models are trained and tested for 100 times. Fig. 2 illustrates the Boston Housing data results. x-axis represents the percentage of the missing data from 0% to 50%, while the y-axis represents the mean error of the 500 rounds of Monte-Carlo split. More specifically, the results are compared with mean imputation, 1-NN imputation and without any MD tool which are shown in the same figure.

From bank figure, we can see that it is risky not to use any MD tool. If the amount of missing data is very small, removing samples may work in some case even it scorifies many information. But when the amount of missing data increases, there is no reason to take this risk. Like the bank data, there are not enough samples left to run TROP-ELM when the percentage of MD reaches around 32%. In Fig. 2, it also illustrates that PDE tool generally performs better than both mean imputation and 1-NN imputation. Moreover, we can see the LOO error and test error (with PDE) start from a very low value 0.03, then arise smoothly with the increasing number of missing data. When the amount of missing data reaches as high as half of the whole training set, LOO error is just 0.19 which is still acceptable. As to the test error (with PDE), it performs smaller than LOO error since the beginning. After adding 50% of the missing data, test error remains on a stable level, around 0.03, which is a significant result we are looking forward to. The results demonstrate the efficiency and stability of the model. On the other hand, test error line vibrates a lot due to the randomness of MD emergences. Nevertheless, the tendency of both LOO and test error keeps the same, and more smoothness can be expected from more rounds of Monte-Carlo test.

Figs. 3–6 show the results for the other four data sets. The results are quite similar with the bank data. From both of these four data results, PDE tools performs better than mean imputation and 1-NN imputation, test errors are less than LOO error from the beginning, and much less vibration. These proves that models are more stable and reliable.

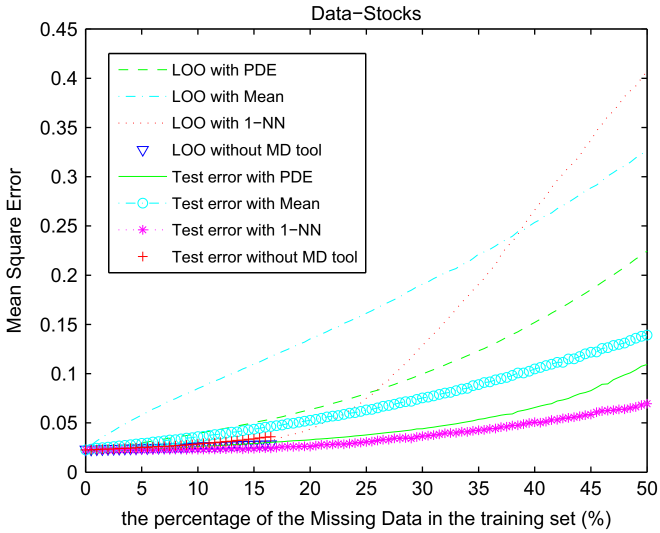


Fig. 3. Normalized MSE for the data set.

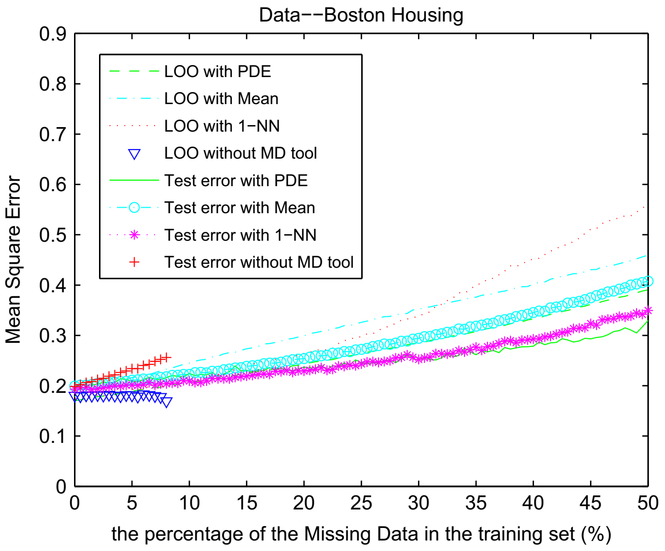


Fig. 4. Normalized MSE for the data set.

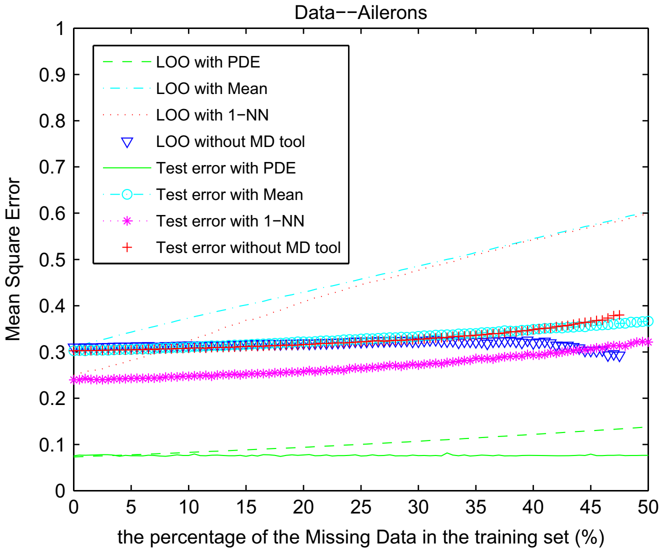


Fig. 5. Normalized MSE for the data set.

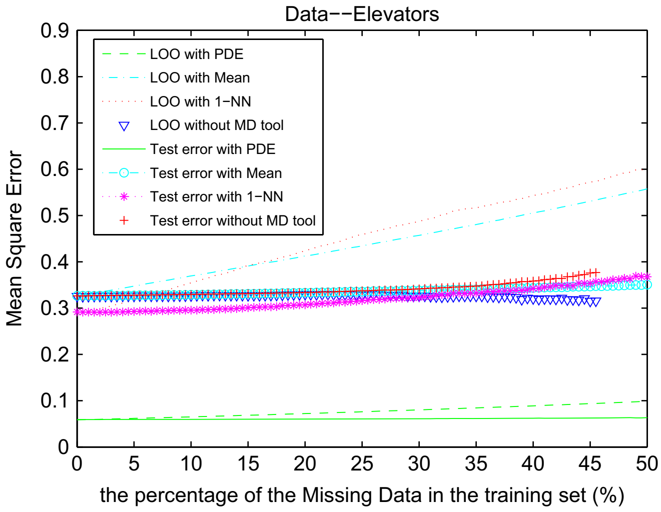


Fig. 6. Normalized MSE for the data set.

6. Conclusions

This paper proposed a method which is the advanced modification of the original extreme learning machine with a new tool to solve the missing data problem.

Briefly speaking, this method uses a cascade of L_1 penalty (LARS) and L_2 penalty (Tikhonov regularization) on ELM to regularize the matrix computations and hence make the MSE computation more reliable, and on the other hand, it estimates the expected pairwise distances directly on incomplete data so that it offers the ELM a solution to solve the missing data issues.

According to the experiments of the five data sets with hundreds of times Monte-Carlo tests, the method shows its significant advantages: it inherits most of the features of original ELM, fast computational speed, no parameter need to be tuned, etc., and it appears more stable and reliable generalization performance by the two penalties. Moreover, according to the results from our proposed methods which perform much better than TROP-ELM without any missing tool, our method completes ELM with a new tool to solve missing data problem even though the half of the training data are missing as the extreme case.

Future work on this method will enrich it to classification tasks, and further improve its performance.

References

[1] A.R.T. Donders, G. van der Heijden, T. Stijnen, K.G. Moons, Review: a gentle introduction to imputation of missing values, J. Clin. Epidemiol. 59 (10) (2006) 1087–1091.

[2] A.N. Baraldi, C.K. Enders, An introduction to modern missing data analyses, J. School Psychol. 48 (1) (2010) 5–37.

[3] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Statist. Soc. 39 (1977) 1–38.

[4] R.J.A. Little, D.B. Rubin, Statistical analysis with missing data, J. R. Statist. Soc. (1987) 292–309.

[5] P. Ho, M.C.M. Silva, T.A. Hogg, Changes in colour and phenolic composition during the early stages of maturation of port in wood stainless steel and glass, J. Sci. Food Agricult. 81 (13) (2001) 1269–1280.

[6] P.D. Faris, W.A. Ghali, R. Brant, C.M. Norris, P.D. Galbraith, M.L. Knudtson, Multiple imputation versus data enhancement for dealing with missing data in observational health care outcome analyses, J. Clin. Epidemiol. 55 (2) (2002) 184–191.

[7] D. Hui, S. Wan, B. Su, G. Katul, Y.L.R. Monson, Gap-filling missing data in eddy covariance measurements using multiple imputation (mi) for annual estimations, Agricult. Forest Meteorol. 121 (2) (2004) 93–111.

[8] N. Sartori, A. Salvan, K. Thomaseth, Multiple imputation of missing values in a cancer mortality analysis with estimated exposure dose, Computat. Statist. Data Anal. 49 (3) (2005) 937–953.

[9] G.B. Huang, Q. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006).

[10] G.B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multi-class classification, IEEE Transactions on Systems, Man, and Cybernetics: Part B: Cybernetics 42 (2) (2012) 513–529.

[11] G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybernet. 2 (2) (2011) 2107–2122.

[12] G.B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, Neurocomputing 71 (2008) 3460–3468.

[13] G.B. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Networks 17 (4) (2006) 879–892.

[14] G. B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, in: 2004 International Joint conference on Neural Networks (2004).

[15] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, Op-elm: optimally-pruned extreme learning machine, IEEE Trans. Neural Networks 21 (2010) 158–162.

[16] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, A. Lendasse, Trop-elm: a double-regularized elm using Lars and Tikhonov regularization, Neurocomputing 74 (16) (2011) 2413–2421.

[17] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, second ed., Wiley, NJ, USA, 2002, pp. 138–149.

[18] D.F. Heitjan, S. Basu, Distinguishing missing at random and missing completely at random, Am. Statist. 50 (3) (1996) 207–213.

[19] G. Lu, J. Copas, Missing at random likelihood ignorability and model completeness, Ann. Statist. 32 (2) (2004) 754–765.

[20] B.L. Ford, An overview of hot-deck procedures, in: Incomplete Data in Sample Surveys, Academic Press, New York, USA, 1983, pp. 185–207.

[21] J. Cohen, P. Cohen, Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences, second ed., 2003, pp. 185–207.

[22] J.L. Schafer, Multiple imputation: a primer, Statistical Methods in Medical Research 8 (1) (1999) 3–15.

[23] F. Scheuren, Multiple imputation: how it began and continues, Am. Statist. 59 (2005) 315–319.

[24] X. Meng, D.B. Rubin, Maximum likelihood estimation via the ecm algorithm: a general framework, Biometrika 80 (2) (1993) 267–278.

[25] E. Eirola, Y. Miche, A. Lendasse, Estimating Expected Pairwise Distances in a Data Set with Missing Values, Technical Report, 2011.

[26] MathWorks, Matlab Financial Toolbox: ecmnmle, URL 〈http://www.mathworks.com/help/toolbox/finance/ecmnmle.html〉, 2010.

[27] J. Sexton, A.R. Swensen, Ecm algorithms that converge at the rate of em, Biometrika 87 (3) (2011) 651–662.

[28] C.R. Rao, S.K. Mitra, Generalized Inverse of Matrices and its Applications, John Wiley & Sons, New York, 1971, p. 240.

[29] Y. Lan, Y. Soh, G.B. Huang, Constructive hidden nodes selection of extreme learning machine for regression, Neurocomputing 73 (16) (2010).

[30] G. Feng, G.B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, IEEE Trans. Neural Networks 20 (8) (2009) 1352–1357.

[31] L. Yuan, S.Y.Chai, G. B. Huang, Random search enhancement of error minimized extreme learning machine, in: European Symposium on Artificial Neural Networks (ESANN) 2010, Bruges, Belgium, 2010, pp. 327–332.

[32] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Statist. Soc. 58 (1) (1996) 267–288.

[33] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2) (2004) 407–499.

[34] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.

[35] G. Bontempi, M. Birattari, H. Bersini, Recursive lazy learning for modeling and control, in: Proceedings of the European Conference on Machine Learning, 1998, pp. 292–303.

[36] G.H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, Technometrics 21 (2) (1979) 215–223.

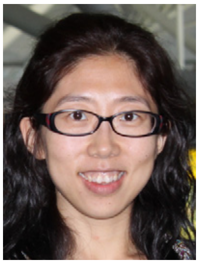
[37] J.A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308–313.

[38] URL 〈http://archive.ics.uci.edu/ml/datasets.html〉.

[39] T.E. Nichols, A.P. Holmes, Nonparametric permutation tests for functional neuroimaging: a primer with examples, Human Brain Map. 15 (1) (2001) 1–25.

[40] J. Chen, J. Shao, Nearest neighbor imputation for survey data, J. Official Statist. 16 (2) (2000) 113–131.

[41] J.V. Hulse, T.M. Khoshgoftaar, Incomplete-case nearest neighbor imputation in software measurement data, Inf. Sci., in press.



Qi Yu was born in China and received her first master degree from Harbin Institute of Technology (HIT) in 2005, China, majoring Telecommunication. After that she came to Finland and got her second master degree about approximation problems in Finance in Aalto University School of Science and Technology (previously Helsinki University of Technology) in Finland, 2007. Now, she is doing her doctoral studies in Environmental and Industrial Machine Learning Group in Aalto University, her research interests include machine learning methods, missing data problem, feature selection, ensemble modeling, neural network and most particularly, for bankruptcy prediction.

Yoan Miche was born in 1983 in France. He received an Engineer's Degree from Institut National Polytechnique de Grenoble (INPG, France), and more specifically from TELECOM, INPG, on September 2006. He also graduated with a master's degree in Signal, Image and Telecom from ENSERG, INPG, at the same time. He recently received his PhD degree in Computer Science and Signal and Image Processing from both the Aalto University School of Science and Technology (Finland) and the INPG (France). His main research interests are steganography/steganalysis and machine learning for classification/regression.

Emil Eirola was born in Helsinki, Finland, in 1984. He received the MSc degree (Mathematics and Information Science) in 2009 from the Helsinki University of Technology. He is currently a doctoral student and research scientist at the Aalto University School of Science Department of Information and Computer Science. His research interests include machine learning with missing data, function approximation, feature selection, high-dimensional data, and ensemble models, particularly in the context of environmental applications.

Mark van Heeswijk has been working as an exchange student in both the EIML (Environmental and Industrial Machine Learning, previously TSPCi) Group and Computational Cognitive Systems Group on his master's thesis on "Adaptive Ensemble Models of Extreme Learning Machines for Time Series Prediction", which he completed in August 2009. Since September 2009, he started as a PhD student in the EIML Group, ICS Department, Aalto University School of Science and Technology. His main research interest is in the field of high-performance computing and machine learning. In particular, how techniques and hardware from high-performance computing can be applied to meet the challenges one has to deal with in machine learning. He is also interested in biologically-inspired computing, i.e. what can be learned from biology for use in machine learning algorithms and in turn what can be learned from simulations about biology. Some of his other related interests include: self-organization, complexity, emergence, evolution, bioinformatic processes, and multi-agent systems.

Eric Séverin is a professor of Finance at USTL (University of Lille) and he is a specialist in corporate finance. His research interests are the following: bankruptcy and financial structure, relationships between economics and finance and financial applications of machine learning in the field of bankruptcy predictions.

Amaury Lendasse was born in 1972 in Belgium. He received the MS degree in Mechanical Engineering from the Universite Catholique de Louvain (Belgium) in 1996, MS in control in 1997 and PhD in 2003 from the same university. In 2003, he has been a postdoctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. Since 2004, he is a senior researcher and a docent in the Adaptive Informatics Research Centre in the Aalto University School of Science and Technology (previously Helsinki University of Technology) in Finland. He has created and is leading the Environmental and Industrial Machine Learning (previously time series prediction and chemoinformatics) Group. He is chairman of the annual ESTSP conference (European Symposium on Time Series Prediction) and member of the editorial board and program committee of several journals and conferences on machine learning. He is the author or the coauthor of around 100 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, non-linear approximation in financial problems, functional neural networks and classification.