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] and Amaury Lendasse [a]

[a] *Department of Information and Computer Science, Aalto University, Espoo, 02150, Finland*

[b] *IAE, Université Lille 1, 59043, Lille cedex, France*

Abstract

This paper proposes a method which is the advanced modification of the original Extreme Learning Machine with a new tool to solve the missing data problem. It 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 on 9 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.

Key words: ELM, Ridge Regression, Tikhonov Regularization, LARS, Missing data, Pairwise distance estimation

1 Introduction

Missing data is very common to confront for many different research fields, for example, data from surveys, experiments, observational studies and etc. typically contain missing values. Because most analysis procedures were not

* Corresponding author
Email address: qi.yu@aalto.fi (Qi Yu).

designed to handle incomplete data, 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[1]. Barely a decade later, Lit *et al.* did acknowledge the limitations of Case Deletion and Single Imputations and then introduced Multiple Imputations [2]. Multiple Imputations would not have been achievable without parallel progress in computational power because generally they are computationally expensive [3–6].

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 [7] is fast enough to accomodate 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). In addition to its speed, which takes the computational time down by several orders of magnitude, the ELM is usually capable to compare with state of the art machine learning algorithms in terms of performance [8].

However, ELM tends to suffer from the presence of irrelevant variables in the data sets, as is likely to happen when dealing with real-world data. In order to reduce the effect of such variables on the ELM model, Miche *et al.* proposed in [9] a wrapper methodology around the original ELM, which includes a cascade of neuron ranking step (via a L_1 regularization), along with a criterion (L_2 regularization) used to prune out the most irrelevant neurons of the model.

Therefore, 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.

[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 this data, it implies one missing point among the original $(N \times d)$ points.

2 Pairwise Distance Estimation with Missing Data (MD)

Missing data (MD) is 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 first common question you may ask is why. There are several reasons why data may be missing, that is, the nature of Missing Data can be categorized into three main types [10],

- Missing completely at random (MCAR)[11]. 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 [12].
- 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 focus 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 approach for MD problem

By far the most common approach is to simply omit those observations with missing data and to run the analyses 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 [13]. Besides, Mean substitution method uses the idea of substituting a mean for the missing data [14,15], etc.

There are also some advanced methods such as Maximum Likelihood and Multiple Imputation [16–18]. 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 [19]. 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 datasets with missing data [20]. 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 $M_x, M_y \subseteq [d] = 1, ..., d$ 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,

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

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:

$$\sigma_{x,i}^2 = \begin{cases} Var[X_i|x_{obs}] \text{ if } \quad i \in M_x, \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{2}$$

Let $y_i^{'}$ and $\sigma_{y,i}^2$ be defined analogously. Then, the expectation of the squared distance can be expressed as:

$$E[\parallel x - y \parallel^2] = \sum_i ((x_i' - y_i')^2 + \sigma_{x,i}^2 + \sigma_{y,i}^2) \tag{3}$$

or, equivalently,

$$E[\parallel x - y \parallel^2] = \parallel x' - y' \parallel^2 + \sum_{i \in M_x} \sigma_{x,i}^2 + \sum_{i \in M_y} \sigma_{y,i}^2 \tag{4}$$

According to Eirola [20], covariance matrix can be achieved through the ECM (Expectation Conditional Maximization) method provided in the MATLAB Financial Toolbox [21], implementing the method of [19] with some improvements by [22], 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 with k nearest neighbors to solve missing data problem.

3 Double-Regularized ELM: TROP-ELM

Miche *et al.* in [23] 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 [7] as an original way of building a single Hidden Layer Feedforward Neural

Network (SLFN). The essence of ELM is that the hidden layer of it need not to be iteratively tuned [8,7], 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 $(x_i, y_i), 1 \leq N.$ 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:

$$\sum_{i=1}^{m} \beta_i f(\omega_i x_j + b_i), \quad 1 \leq j \leq N \tag{5}$$

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

$$\mathbf{H} = \begin{pmatrix} f(\omega_1 x_1 + b_1) & \dots & f(\omega_m x_1 + b_m) \\ \vdots & \ddots & \vdots \\ f(\omega_1 x_n + b_1) & \dots & f(\omega_m x_n + b_m) \end{pmatrix} \tag{6}$$

$\beta = (\beta_1, ..., \beta_m)^T$ and $\mathbf{y} = (y_1, ..., y_n)^T$. The ELM approach is thus to initialize randomly the ω_i and b_i and compute the output weights $\beta = \mathbf{H}^\dagger \mathbf{y}$ by a Moore-Penrose pseudo-inverse [24].

The significant advantages of ELM are its extreme fast learning speed, relative better generalization performance while being a simple method [7]. There has been recent advances based on the ELM algorithm, to improve its robustness (OPELM [9], CS-ELM [25]), or make it a batch algorithm, improving at each iteration (EM-ELM [26], EEM-ELM [27]).

3.2 L_1 penalty: LASSO

An important part in ELM is to minimize the training error $\| \mathbf{H}\beta - \mathbf{y} \|$, which is a ordinary regression problem. One technique to solve this is called Lasso, for 'least absolute shrinkage and selection operator' proposed by Tibshirani [28].

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 worked on is

$$\min_{\lambda,\omega} \left(\sum_{i=1}^{N} (y_i - \mathbf{x_i}\omega)^2 + \lambda \sum_{j=1}^{p} |\omega_j| \right) \tag{7}$$

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 [29], 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 [29].

The disadvantages of the LARS method is 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 article [29]. 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 [30]. In statistics, the method is also known as ridge regression.

The general form of Tikhonov regularization is to minimize:

$$\min_{\lambda,\omega} \left(\sum_{i=1}^{N} (y_i - \mathbf{x_i}\omega)^2 + \lambda \sum_{j=1}^{p} \omega_j^2 \right) \tag{8}$$

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 doesn't have sparseness

in the final solution and hence, it doesn't 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 [9] 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 [31,32].

$$\epsilon^{PRESS} = \frac{y_i - h_i b_i}{1 - h_i P h_i^T} \tag{9}$$

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

$$\epsilon^{PRESS} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - x_i (X^T X)^{-1} x^T y}{1 - x_i (X^T X)^{-1} x_i^T} \right)^2 \tag{10}$$

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:

$$\epsilon^{PRESS}(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_i - x_i (X^T X + \lambda I)^{-1} x^T y}{1 - x_i (X^T X + \lambda I)^{-1} x_i^T} \right)^2 \tag{11}$$

This new modified version uses the Singular Value Decomposition (SVD) approach [33] 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 [34] 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.

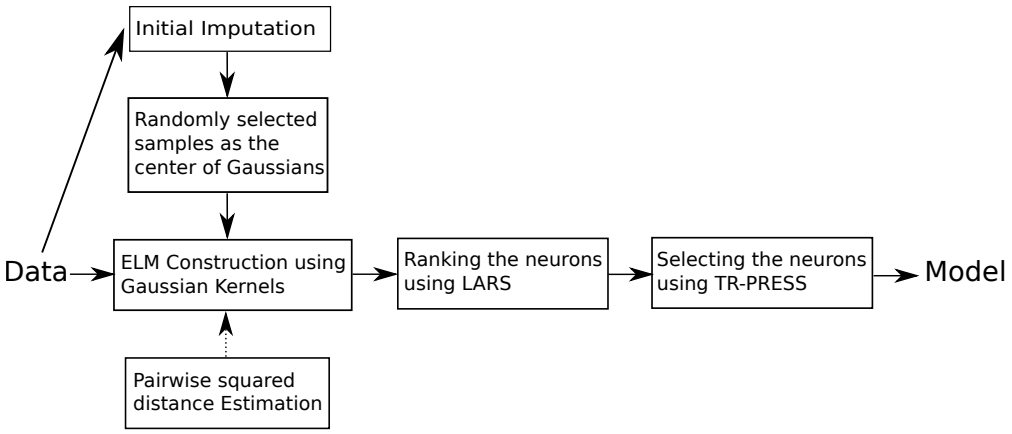

Figure 1. The framework of the proposed method

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 are 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.
- Secondly, 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's 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 are needed, which are exactly the same thing the Pairwise squared distance estimation method achieved. The hidden node parameters (σ^2, μ) are randomly generated, which remains the advantage of ELM that the parameters in hidden layer need not to be tuned. More specifically, parameter σ^2 is choosen from a interval (20% to 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 (β) 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 previous 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 11.

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, 9 datasets are chosen in this paper to evaluate the method. These data sets can be found from UCI Machine Learning Repository for free.

Table 1 shows the information of the 10 selected data sets. Many articles use the same data sets, for example, Miche *et.al* in [9], etc., they could be found to compare the results with ours.

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 dataset 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 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.

Datasets	# Attributes	# Training data	# Testing data
Ailerons	5	4752	2377
Elevators	6	6344	3173
Computer	12	5461	2731
Auto Price	15	106	53
Machine CPU	6	139	70
Breast Cancer	32	129	65
Bank	8	2999	1500
Stocks	9	633	317
Boston Housing	13	337	169

Table 1
Specification of the tested regression data sets

5.1 Monte-Carlo split for preprocessing

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

5.2 Generating the missing data

There is no missing value originally in these 9 datasets. Therefore, missing data is artificially added in each datasets, in order to test the performance on incomplete data of the method. More precisely, the missing data is added 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 is added $(N \times d)/200$ 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 for each dataset.

5.3 Experiments results

For each dataset, the same experiment procedure is done to evaluate the method. Firstly, Monte-Carlo split is performed for 100 times (Data Machine CPU, for 1000 times as an exception), 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 error and test error are recorded for 100 times from those Monte-Carlo splits. All the results shown here are the normalized results.

Take the Boston housing data for instance. There are 506 samples and 13 variables originally in this data, and one output. For each Monte-Carlo split, 337 samples are randomly selected for training, and the rest for testing. As to the training set, $(337 \times 13)/200 \approx 22$ 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 100 Monte-Carlo split. More specifically, the dash line refers to the mean LOO error, and solid line is the mean test error.

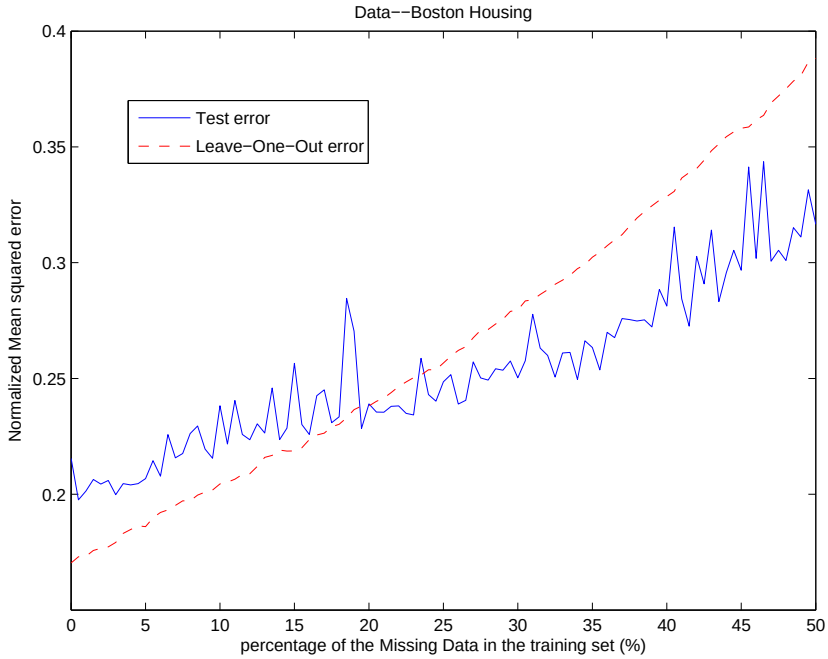

Figure 2. Boston Housing data results

From Fig 2, we can see the LOO error starts from a very low value 0.09, then ascends 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 0.38 which is still acceptable. As to the test error, it starts from value 0.22, which is very common case that test error performs worse than validation error. However, after adding around 20% MD, test errors appears lower than LOO error which is a significant result we are looking forward to. Even though this is not always the case, it implies the model built not at all 'overfits' the data, and moreover, shows the efficiency and stability of the

model. On the other hand, test error line (solid one) vibrates a lot due to the randomness of MD emergences. Nevertheless, the tendency of both LOO and test error keep the same, and more smooothness can be expected from more rounds of Monte-Carlo test.

Fig 3 shows the results for another two data sets. The results are quite similar with the Boston Data. From both of these two Data results, test error are less than LOO error from the beginning, and much less vibration. These proves that models are more stable and reliable.

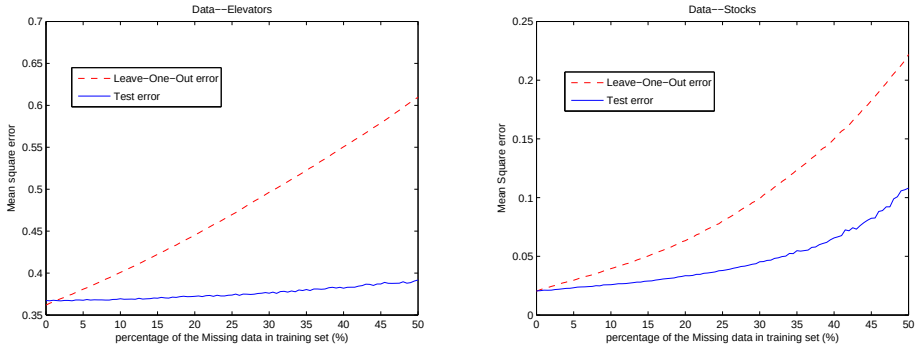

Figure 3. Normalized MSE for the datasets: Leave-One-Out results (dashed line) and test results (solid line)

For the rest of the Data sets, the test error and LOO error are not easy to present in the same figure, so that separate figures are used for them. Fig 4 and Fig 5 includes the rest 7 Data results. As to some data sets, like Breast Cancer, Elevators, Auto Price, they are born to be very difficult to regress. Thus, the test errors contain more vibrations and the performs less stable. The relative analysis and results about these data sets can be easily searched from other research articles as references.

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 9 data sets with 100 times Monte-Carlo tests,

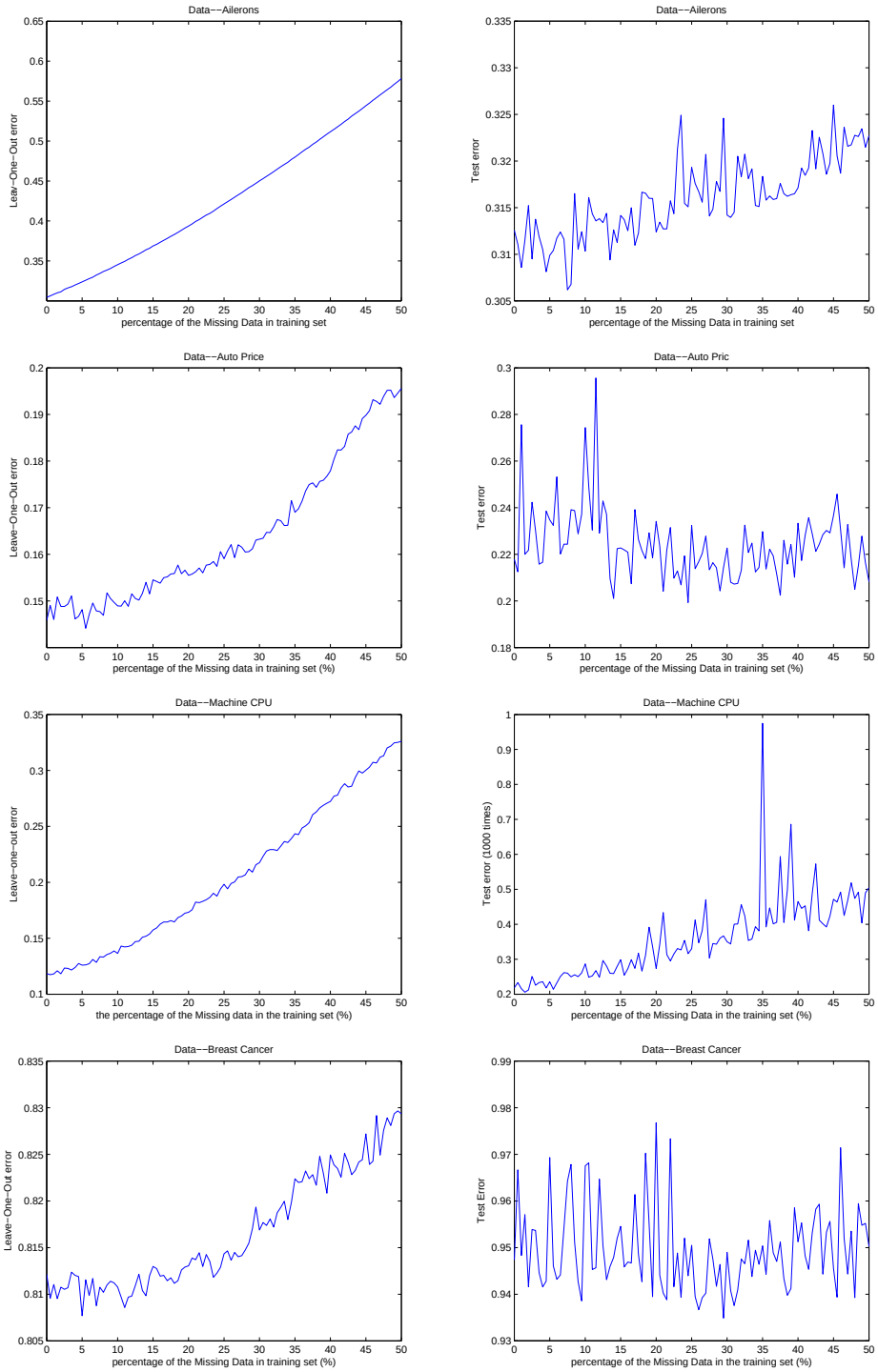

Figure 4. Normalized MSE for the datasets: Leave-One-Out results (left column) and test results (right column)

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, it completes ELM with a new tool to solve missing data

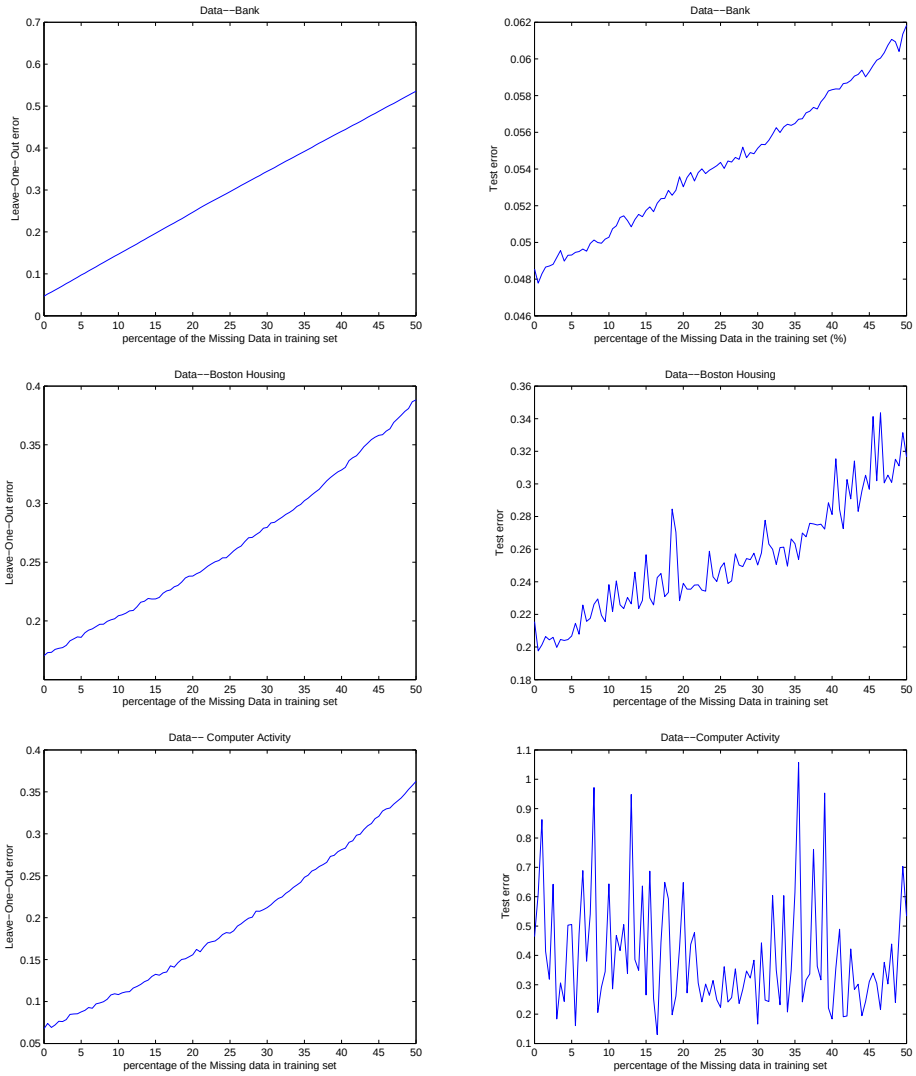

Figure 5. Normalized MSE for the datasets: Leave-One-Out results (left column) and test results (right column)

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. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[2] R. J. A. Little and D. B. Rubin, "Statistical analysis with missing data," *Journal of the Royal Statistical Society.*

[3] P. Ho, M. C. M. Silva, and T. A. Hogg, "Changes in colour and phenolic composition during the early stages of maturation of port in wood, stainless steel and glass," *Journal of the Science of Food and Agriculture*, vol. 81, no. 13, pp. 1269–1280, 2001.

[4] P. D. Faris, W. A. Ghali, R. Brant, C. M. Norris, P. D. Galbraith, and M. L. Knudtson, "Multiple imputation versus data enhancement for dealing with missing data in observational health care outcome analyses," *Journal of Clinical Epidemiology*, vol. 55, no. 2, pp. 184–191, 2002.

[5] D. Hui, S. Wan, G. K. B. Su, and Y. L. R. Monson, "Gap-filling missing data in eddy covariance measurements using multiple imputation (mi) for annual estimations," *Agricultural and Forest Meteorology*, vol. 121, no. 2, pp. 93–111, 2004.

[6] N. Sartori, A. Salvan, and K. Thomaseth, "Multiple imputation of missing values in a cancer mortality analysis with estimated exposure dose," *Computational Statistics & Data Analysis*, vol. 49, no. 3, pp. 937–953, 2005.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, 2006.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," in *Proceedings of international joint conference on neural networks*, vol. 2, (Budapest, Hungary), pp. 985–990, 2004.

[9] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "Op-elm: Optimally-pruned extreme learning machine," *In IEEE Transactions on Neural Networks*, vol. 21, pp. 158–162, 2010.

[10] R. J. A. Little and D. B. Rubin, "Statistical analysis with missing data (second ed.)," (Wiley, NJ, USA), 2002.

[11] D. F. Heitjan and S. Basu, "Distinguishing "missing at random" and "missing completely at random"," *The American Statistician*, vol. 50, no. 3, pp. 207–213, 1996.

[12] G. Lu and J. Copas, "Missing at random, likelihood ignorability and model completeness," *Annals of Statistics*, vol. 32, no. 2, pp. 754–765, 2004.

[13] B. L. Ford, "An overview of hot-deck procedures, in: Incomplete data in sample surveys," in *Academic Press*, (New York, USA), pp. 185–207, 1983.

[14] J. Cohen and P. Cohen, "Applied multiple regression/correlation analysis for the behavioral sciences (2nd ed.)," 2003.

[15] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, "Applied multiple regression/correlation analysis for the behavioral sciences (rdnd ed.)," 2003.

[16] J. L. Schafer, "Analysis of incomplete multivariate data," 1997.

[17] J. L. Schafer, "Multiple imputation: a primer," *Statistical Methods in Medical Research*.

[18] F. Scheuren, "Multiple imputation: How it began and continues," *The American Statistician*, vol. 59, pp. 315–319, 2005.

[19] X. Meng and D. B. Rubin, "Maximum likelihood estimation via the ecm algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[20] E. Eirola, Y. Miche, and A. Lendasse, "Estimating expected pairwise distances in a data set with missing values," *Tecnical report*, 2011.

[21] MathWorks, "Matlab financial toolbox: ecmnmle," *URL http://www.mathworks.com/help/toolbox/finance/ecmnmle.html*, 2010.

[22] J. Sexton and A. R. Swensen, "Ecm algorithms that converge at the rate of em," *Biometrika*, vol. 87, no. 3, pp. 651–662, 2011.

[23] Y. Miche, M. van Heeswijk, P. Bas, O. Simula, and A. Lendasse, "Trop-elm: a double-regularized elm using lars and tikhonov regularization," *Neurocomputing*, 2010.

[24] C. R. Rao and S. K. Mitra, "Generalized inverse of matrices and its applications," *New York: John Wiley & Sons*, p. 240, 1971.

[25] Y. Lan, Y. C. Soh, and G.-B. Huang, "Constructive hidden nodes selection of extreme learning machine for regression," *Neurocomputing*, 2010.

[26] G. Feng, G.-B. Huang, Q. Lin, and R. Gay, "Error minimized extreme learning machine with growth of hidden nodes and incremental learning," *IEEE Transactions on Neural Networks,*, vol. 20, no. 8, pp. 1352–1357, 2009.

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

[28] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[29] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[30] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, 1970.

[31] R. Myers *Classical and Modern Regression With Applications*, 1990.

[32] G. Bontempi, M. Birattari, and H. Bersini, "Recursive lazy learning for modeling and control," in *Proc. Eur. Conf. Machine Learn*, pp. 292–303, 1998.

[33] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[34] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.