Extreme Learning Machine for Regression and Classification Using L_1-Norm and L_2-Norm

Xiong Luo*, Xiaohui Chang, and Xiaojuan Ban

School of Computer and Communication Engineering,
University of Science and Technology Beijing (USTB), 100083 Beijing, China
xluo@ustb.edu.cn

Abstract. Extreme learning machine (ELM) has been studied extensively in recent years. It is a very simple machine learning algorithm which can achieve a good generalization performance with extremely fast speed. Thus, it has practical significance for Big Data analysis. Normally, it is implemented under the empirical risk minimization scheme and it may tend to generate a large-scale and over-fitting model. In this paper, an ELM model based on L_1-norm and L_2-norm regularizations is proposed to deal with regression and multiple class classification problems in a unified framework, and it can reduce the complexity of the network and prevent over-fitting. We test the proposed algorithm on eight benchmark data sets. Simulation results have shown that the proposed algorithm outperforms the original ELM and other advanced ELM algorithm in terms of prediction accuracy and stability.

Keywords: extreme learning machine, ridge regression, elastic net, model selection.

1 Introduction

More recently, data is being collected at an unprecedented scale. There are increasing demand of effective data analysis for making decisions to fully realize the potential of Big Data. Single-hidden layer feedforward network (SLFN) based on extreme learning machine (ELM) [1] is one of the important methods used in data analysis due to its powerful nonlinear mapping capability and extremely fast learning speed. However, original ELM solution may tend to generate an over-fitting model and are less stable in some situations [2]. Moreover, the structure of the neural network (NN) is still a question in ELM design.

To overcome the problems ELM faced, several schemes have been proposed. In [3], Rong *et al.* proposed a fast pruned ELM for classification problems. Martínez-Martínez *et al.* proposed a regularized ELM for regression problems in [4]. In [3] and [4], although those algorithms can generate a spare NN structure, they do not provide a unified NN framework for both regression and classification problems.

* This work was jointly supported by the National Natural Science Foundation of China under Grants 61174103, 61174069, and 61004021, and the Fundamental Research Funds for Central Universities under Grant FRF-TP-11-002B.

J. Cao et al. (eds.), *Proceedings of ELM-2014 Volume 1*,
Proceedings in Adaptation, Learning and Optimization 3, DOI: 10.1007/978-3-319-14063-6_25

Miche *et al.* proposed an optimally pruned ELM for regression and classification in [5], which was a regularized ELM by using the least angle regression (LARS) algorithm, i.e., a L_1 penalty, but this algorithm has its limitation while facing a group of high correlated variables.

Considering those problems in ELM design analyzed above, we propose a novel ELM algorithm based on L_1 penalty and L_2 penalty to deal with both multiple output regression tasks and multiple class classification tasks in a unified framework. Here, elastic net algorithm is used to solve this mixed penalties [6]. Then separate elastic net algorithm and the Bayesian information criterion (BIC) [7] are adopted to find the optimal model for each response variable. Thus, the proposed algorithm tends to reduce over-fitting and provide a more robust model.

This paper is organized as follows. Section 2 analyses the SLFN based on ELM and classic regularization methods. Section 3 presents the proposed ELM model based on L_1-norm and L_2-norm regularizations. Section 4 provides the simulation results and discussion. Section 5 summarizes the conclusion.

2 Model Description

2.1 SLFN Based on ELM

ELM theories claim that the hidden node learning parameters can be randomly assigned and the output weights can be determined by solving a linear system [8], thus the ELM can be implemented with few steps and low computational cost.

For P arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, \cdots, x_{im}]^T \in \mathbb{R}^m$ and $t_i = [t_{i1}, t_{i2}, \cdots, t_{in}]^T \in \mathbb{R}^n$, a standard SLFN with L hidden nodes can be mathematically modeled as:

$$o_i = \sum_{j=1}^{L} \beta_j G(a_j, b_j, x_i), \quad i = 1, 2, \cdots, P \tag{1}$$

where a_j and b_j are the learning parameters of hidden nodes, β_j is the link connecting the j-th hidden node to the output nodes, $G(a_j, b_j, x_i)$ is the output of the j-th hidden node with respect to the input x_i, and o_i is the actual output.

The SLFN with L hidden nodes can approximate these P samples with zero error, which means that the cost function $E = \sum_{i=1}^{P} ||o_i - t_i||_2 = 0$, i.e., there exist (a_j, b_j) and β_j such that:

$$t_i = \sum_{j=1}^{L} \beta_j G(a_j, b_j, x_i), \quad i = 1, 2, \cdots, P \tag{2}$$

where $\|\cdot\|_2$ represents the L_2-norm.

The above P equations can be written compactly as :

$$H\beta = T \tag{3}$$

where

$$H = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_P) & \cdots & G(a_L, b_L, x_P) \end{bmatrix}_{P \times L}, \beta = \begin{bmatrix} {\beta_1}^T \\ \vdots \\ {\beta_L}^T \end{bmatrix}_{L \times n}, T = \begin{bmatrix} {t_1}^T \\ \vdots \\ {t_P}^T \end{bmatrix}_{P \times n}.$$

Here, H is called the hidden layer output matrix of the SLFN. Thus, the system (3) becomes a linear model and the output weights can be analytically determined by finding a least-square solution of this linear system as follows:

$$\beta = H^{\dagger} T \tag{4}$$

where $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H [1].

Although ELM has been developed to work at a much faster learning speed with the higher generalization performance, it also has some drawbacks:

1) ELM is designed with the empirical risk minimization (ERM) principle and may tend to generate an over-fitting model.

2) ELM provides weak control capacity and is less stable because it is implemented by using a classical least-square method.

3) Users have to choose the number of hidden nodes through trial-and-error.

2.2 Regularization Methods

Multiple linear regression is often used to investigate the relationship between the predictor variables and the response variables. Then both the prediction accuracy and the size of the model should be considered.

Considering the general setup for a single-output regression problem:

$$y = H\beta + \varepsilon \tag{5}$$

where H is the inputs data set, and it is a $P \times L$ matrix. Here y is the actual output, β is the regression weights, and ε is the residuals. The traditional approach used to solve the above problem is the ordinary least square (OLS) estimates, which can be formulated as follows:

$$\widehat{\beta} = \arg \min_{\beta} ||y - H\beta||_2^2 \tag{6}$$

where $\widehat{\beta} = [\widehat{\beta}_1, \widehat{\beta}_2, \cdots, \widehat{\beta}_L]^T$ is the estimated regression weights. It is well known that OLS often performs not well in terms of both prediction accuracy and the model size [9]. Regularization techniques have been proposed to improve OLS.

The L_1-norm, which is also called the least absolute shrinkage and selection operator (Lasso) [10], represents the most basic augmentation of the OLS solution. The Lasso estimate $\widehat{\beta}$ is defined by:

$$\widehat{\beta} = \arg \min_{\beta} \left\{ ||y - H\beta||_2^2 + \lambda ||\beta||_1 \right\} \tag{7}$$

where λ is a positive regularization parameter, $\|\cdot\|_1$ represents the L_1-norm. As λ increases, the number of nonzero components of $\widehat{\beta}$ decreases.

Due to its nature of both continuous shrinkage and automatic variable selection simultaneously, the Lasso has shown success in many situations. But it has some limitations as noted by Zou and Hastie in [6].

To overcome the drawbacks of L_1-norm, both the L_1 penalty and the L_2 penalty are used in the same minimization problem. The mathematic model of this mixed penalties can be formulated as follows:

$$\widehat{\beta} = \arg\min_{\beta} \left\{ ||y - H\beta||_2^2 + \lambda||\beta||_1 + \xi||\beta||_2^2 \right\} \tag{8}$$

where both λ and ξ are tuning parameters. In [11], the elastic net was proposed to solve (8).

The elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select group of correlated variables. It has been shown that the elastic net often outperforms the Lasso in terms of prediction accuracy, while enjoying a similar sparsity of representation.

3 L_1-L_2-ELM Model

3.1 Solution of the Elastic Net

The basic idea of solving the elastic net is to reduce the elastic net problem to an equivalent Lasso problem.

For data set (y, H) and (λ, ξ) defined in (8), an artificial data set (y^*, H^*) is generated as follows:

$$\begin{cases} {H^*}_{(P+L)\times L} = \frac{1}{\sqrt{(1+\xi)}} \begin{pmatrix} H \\ \sqrt{\xi} I \end{pmatrix} \\ {y^*}_{(P+L)\times L} = \begin{pmatrix} y \\ 0 \end{pmatrix} \end{cases} \tag{9}$$

Then the naive elastic net criterion can be written as:

$$\widehat{\beta^*} = \arg\min_{\beta^*} \left\{ ||y^* - H^*\beta^*||_2^2 + r||\beta^*||_1 \right\} \tag{10}$$

where $r = \frac{\lambda}{\sqrt{1+\xi}}$ and $\beta^* = \sqrt{1+\xi}\beta$. Thus, $\widehat{\beta}$ can be represented as follows:

$$\widehat{\beta} = \frac{1}{\sqrt{1+\xi}} \widehat{\beta^*} \tag{11}$$

However, the above solution may incur a double shrinkage, which may introduce unnecessary extra bias. Then the elastic net estimates $\widehat{\beta}$ as follows:

$$\widehat{\beta} = \sqrt{1+\xi}\, \widehat{\beta^*} \tag{12}$$

Thus the elastic net avoids the ridge shrinkage effect by introducing a scaling factor $(1 + \xi)$, while still keep the grouping effect feature of ridge regression. Hence, the solution of elastic net has been successfully transformed into the

lasso problem, and an efficient LARS-EN algorithm was implemented to solve the elastic net solution paths for any fixed ξ [6] .

For each fixed ξ, the LARS-EN algorithm will produce a set of candidate models. Then we adopt the BIC to do the model selection to balance the accuracy and the network size. The BIC was defined as follows:

$$\text{BIC}= -2\ln(Q)+M\ln(P) \tag{13}$$

where Q is the value of the likelihood function for the estimated model, M is the number of hidden nodes to be estimated, and P is the number of samples.

3.2 L_1-L_2-ELM Model

Both multiple output regression and the multiclass classification tasks can be implemented using a unified network model in the proposed algorithm. For multiclass classification problem, it can be transformed into a multiple output regression problem. Assume a set of multiclass training samples (x_i, t_i) $(i = 1, 2, \cdots, P)$, $t_i \in \{1, 2, \cdots, n\}$, each class label is expanded into a label vector of length n according to the original ELM algorithm. For example, in a training sample $(x_i,\ t_i)$, if x_i is the third class, the corresponding output label vector is $t_i = [-1, -1, 1, -1, \cdots, -1]$, i.e, the output node with the largest value indicates its class label.

Then, for the regression problem with n output nodes, the proposed algorithm uses n separate elastic nets to generate the entire solution paths for each output node, then adopts the BIC to find the optimal candidate model. Thus, the output weight of ELM consists of all the optimal candidate models for each response variable. Overall, the proposed algorithm, namely, L_1-L_2-ELM, can be summarized as Algorithm 1.

4 Simulation Results and Discussion

4.1 Experimental Setup

To verify the effectiveness of the proposed algorithm L_1-L_2-ELM, eight data sets from the UCI machine learning repository [12] have been used to test this algorithm, and we compare it with the original ELM and the OP-ELM [13]. The number of hidden neurons L=100 is assigned in ELM, while the OP-ELM and the L_1-L_2-ELM use a maximum number of 100 neurons. In L_1-L_2-ELM algorithm, the fixed ξ is assigned the value of 10^{-3}. In the experiments, each data set is normalized to zero mean and unit variance, and 50 trials have been conducted for all the algorithms. Then the best performance and the standard deviations (DEV) are recorded. Nodes required by L_1-L_2-ELM in each data set can be obtained by calculating the average of the numbers of selected neurons for each response node.

Algorithm 1. L_1-L_2-ELM

Input: a training set: $\{(x_i, t_i) \mid x_i \in \mathbb{R}^m, t_i \in \mathbb{R}^n, i = 1, \cdots, P\}$;
hidden node activation function: $g(x)$;
the max hidden node number: L;
fixed L_2 penalty term: ξ .

Output: the output weight: β.

Assign arbitrary learning parameters of hidden nodes a_j and b_j, $1 \leqslant j \leqslant L$;
Calculate the hidden layer output matrix H based on (3);
for $1 \leqslant i \leqslant n$ **do**
  β'= LARS-EN($H, y(i), \xi$), where $y(i) = [t_{1i}, \cdots, t_{Pi}]^T$;
  $\beta' = (1 + \xi)\beta'$;
  for $1 \leqslant k \leqslant \text{size}(\beta')$ **do**
    Calculate the BIC(k) for every candidate model based on (13) and $\beta'(k)$.
  end
  $k^* = \arg\min_k \{\text{BIC}(k)\}_{k=1}^{\text{size}(\beta')}$, where k^* is the index of the minimum value in vector BIC;
  $\beta'_{\text{optimal}} = \beta'(k^*)$;
  $\beta = [\beta \quad \beta'_{\text{optimal}}]$.
end

Table 1. Information of the regression data sets

Data sets	Attributes	Samples	
		Training	Testing
Abalone	8	2000	2177
Delta_elevators	6	6300	3217
Machine_CPU	6	139	70
Servo	4	110	57

4.2 Real-World Regression Problems

The specifications of the 4 real-world benchmark data sets [12] are listed in Table 1 while the comparison results of algorithms are provided in Table 2. As we can see from Table 2, the proposed algorithm is better than the OP-ELM and the original ELM in terms of the testing average root mean square error (RMSE) and the DEV, which means that the proposed algorithm has a better predicting accuracy and is more robust than the other two algorithms. And the L_1-L_2-ELM algorithm has a better variable selection procedure than the OP-ELM in most cases.

4.3 Real-World Classification Problems

The specifications of the 4 real-world classification data sets [12] are listed in Table 3. The comparison results are shown in Table 4. In Table 4, the L_1-L_2-

Table 2. RMSE and DEV in ELM, OP-ELM, and L_1-L_2-ELM on regression data sets

Methods	Datasets	RMSE		DEV		Nodes
		Training	Testing	Training	Testing	
ELM	Abalone	1.9807	2.1804	0.0071	0.0321	100
OP-ELM		2.0539	2.2042	0.0224	0.0287	55
L_1-L_2-ELM		2.0744	2.1087	0.0137	0.0123	37
ELM	Delta_elevators	0.0014	0.0015	2.5996e-05	3.0653e-05	100
OP-ELM		0.0014	0.0014	8.6406e-06	8.7159e-06	65
L_1-L_2-ELM		0.0014	0.0014	2.9027e-06	1.7586e-06	30
ELM	Machine_CPU	137.7941	235.7793	7.1300	4.1434e+11	100
OP-ELM		19.0124	75.8023	14.5489	27.8916	65
L_1-L_2-ELM		23.2940	45.2975	1.2069	3.5647	48
ELM	Servo	0.0424	3.5394	0.0155	1.4206	100
OP-ELM		0.2556	0.8962	0.1022	0.1219	60
L_1-L_2-ELM		0.2631	0.6979	0.0196	0.0393	63

Table 3. Information of the classification data sets

Data sets	Attributes/Classes	Samples	
		Training	Testing
Iris	4/3	100	50
Wine	13/3	120	58
Glass Identification	9/6	170	44
Landsat Satellite	36/6	4435	2000

Table 4. Success rate and DEV in ELM, OP-ELM, and L_1-L_2-ELM on classification data sets

Methods	Datasets	Success Rate		DEV		Nodes
		Training	Testing	Training	Testing	
ELM	Iris	1.0000	0.7800	0.0000	0.1017	100
OP-ELM		0.9800	0.9600	0.0167	0.0341	30
L_1-L_2-ELM		0.9900	0.9600	0.0045	0.0110	14.667
ELM	Wine	0.8333	0.8276	0.0686	0.0932	100
OP-ELM		0.9917	0.9483	0.0121	0.0294	55
L_1-L_2-ELM		0.9917	0.9828	0.0035	0.0151	34
ELM	Glass Identification	0.8412	0.6591	0.0214	0.0521	100
OP-ELM		0.8824	0.6818	0.0413	0.0503	10
L_1-L_2-ELM		0.7353	0.7045	0.0403	0.0412	6.333
ELM	Landsat Satellite	0.7445	0.7320	0.0322	0.0336	100
OP-ELM		0.8397	0.8140	0.0053	0.0066	80
L_1-L_2-ELM		0.8616	0.8420	0.0039	0.0057	29.166

ELM can achieve a higher success rate for testing samples. And the DEV is much lower than the other algorithms, which means that the proposed algorithm has more accurate and stable classification performance.

5 Conclusion

Data analysis plays a guidance role for making future plans in this Big Data era. In this paper, a novel algorithm called L_1-L_2-ELM was proposed as an effective technology in data analysis. It can deal with multiple output regression and multiple class classification problems in a unified framework. In the proposed algorithm, for W multiple output applications, W separate elastic nets need to be used to find the optimal candidate model. Simulation results have shown that the proposed algorithm has a better generalization performance and variable selection ability than the ELM and OP-ELM especially in multiple class applications. Meanwhile the L_1-L_2-ELM is more robust than the other two algorithms.

References

1. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks. In: IEEE International Joint Conference on Neural Networks, vol. 2, pp. 985–990. IEEE Press, New York (2004)
2. Horata, P., Chiewchanwattana, S., Sunat, K.: Robust Extreme Learning Machine. Neurocomputing 102, 31–44 (2013)
3. Rong, H.J., Ong, Y.S., Tan, A.H., Zhu, Z.: A Fast Pruned-Extreme Learning Machine for Classification Problem. Neurocomputing 2, 359–366 (2008)
4. Martínez-Martínez, J.M., Escandell-Montero, P., Soria-Olivas, E., Martn-Guerrero, J.D., Magdalena-Benedito, R., Gómez-Sanchis, J.: Regularized Extreme Learning Machine for Regression Problems. Neurocomputing 74, 3716–3721 (2011)
5. Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.: OP-ELM: Optimally Pruned Extreme Learning Machine. IEEE Trans. Neural Networks 21, 158–162 (2010)
6. Zou, H., Hastie, T.: Regularization and Variable Selection via the Elastic Net. J. R. Statist. Soc. B 67, 301–320 (2005)
7. Burnham, K.P., Anderson, D.R.: Multimodel Inference Understanding AIC and BIC in Model Selection. Sociological Methods & Res 33, 261–304 (2004)
8. Cao, J., Lin, Z., Huang, G.B., Liu, N.: Voting Based Extreme Learning Machine. Inf. Sci. 1, 66–77 (2012)
9. Tibshirani, R.: Regression Shrinkage and Selection via the Lasso. J. R. Statist. Soc. B 58, 267–288 (1996)
10. Jacob, L., Obozinski, G., Vert, J.P.: Group Lasso with Overlap and Graph Lasso. In: 26th Annual International Conference on Machine Learning, pp. 433–440. ACM Press, New York (2009)
11. De Mol, C., De Vito, E., Rosasco, L.: Elastic-Net Regularization in Learning Theory. J. Complexity 25, 201–230 (2009)
12. Bache, K., Lichman, M.: UCI Machine Learning Repository (2013), `http://archive.ics.uci.edu/ml/`
13. Grigorievskiy, A., Miche, Y., Ventelä, A.M., Séverin, E., Lendasse, A.: Long-Term Time Series Prediction Using OP-ELM. Neural Netw. 51, 50–56 (2014)