Neurocomputing 174 (2016) 179–186

Contents lists available at ScienceDirect

Neurocomputing

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

Regression and classification using extreme learning machine based on L_1-norm and L_2-norm ☆

Xiong Luo *, Xiaohui Chang, Xiaojuan Ban

School of Computer and Communication Engineering, University of Science and Technology Beijing (USTB), Beijing 100083, China

ARTICLE INFO

Article history:
Received 28 September 2014
Received in revised form
27 February 2015
Accepted 1 March 2015
Available online 12 August 2015

Keywords:
Extreme learning machine
Ridge regression
Elastic net
Model selection
Bayesian information criterion (BIC)

ABSTRACT

Extreme learning machine (ELM) is a very simple machine learning algorithm and it can achieve a good generalization performance with extremely fast speed. Therefore it has practical significance for data analysis in real-world applications. However, it is implemented normally 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 handle regression and multiple-class classification problems in a unified framework. The proposed model called L_1–L_2-ELM combines the grouping effect benefits of L_2 penalty and the tendency towards sparse solution of L_1 penalty, thus it can control the complexity of the network and prevent over-fitting. To solve the mixed penalty problem, the separate elastic net algorithm and Bayesian information criterion (BIC) are adopted to find the optimal model for each response variable. We test the L_1–L_2-ELM algorithm on one artificial case and nine benchmark data sets to evaluate its performance. Simulation results have shown that the proposed algorithms outperform the original ELM as well as other advanced ELM algorithms in terms of prediction accuracy, and it is more robust in both regression and classification applications.

1. Introduction

At the information stage, there has been a growing interest in the study of data analysis techniques. Techniques of data analysis can extract previously unknown, hidden, but potentially useful information and knowledge from original data, which is helpful to provide suggestions or decisions for future actions [1]. Single-hidden layer feedforward network (SLFN) is one of the classic methods used in data analysis due to its powerful nonlinear mapping capability [2]. However, it is clear that the learning speed of SLFN is far slower than required because of its slow gradient-based learning algorithm [3,4], which has imposed very challenging obstacles in practical applications.

Recently, extreme learning machine (ELM) for SLFNs was proposed by Huang et al. [5]. It randomly chooses the parameters of hidden nodes and analytically determines the output weights through the use of ordinary least square method. It tends to reach the solution straightforward without facing such trivial issues like local minima, improper learning rate, poor computational scalability, and so on. Compared with other traditional learning techniques, ELM can achieve a better generalization performance for classification and regression with extremely fast speed, which makes it work as an emergent technology for data analysis in many practical applications, such as wireless sensor networks [6]. However, basic ELM solutions may tend to generate an over-fitting model and are less stable in some situations [7]. Moreover, there is another question in the ELM design: how to obtain an appropriate neural network (NN) structure?

To overcome the problems ELM faced, several schemes have been proposed. A regularized ELM based on L_2 penalty was proposed by Deng et al. [8]. It avoids the generation of an over-fitting model and can provide more robust estimate as well as better generalization ability than ELM. But it cannot provide a suitable NN structure, which may lead to some issues while facing irrelevant variables. To find an appropriate NN structure with better robustness and generality, in [9], Rong et al. proposed a fast pruned ELM based on statistical tests for classification problems. Then using a similar statistical method that measures the relativity between the hidden nodes and the output nodes of ELM, Martínez-Martínez et al. proposed a regularized ELM based on L_1 penalty and hybrid penalties for regression problems in [10]. In [9,10], although those algorithms can generate a spare NN

☆This work was jointly supported by the National Natural Science Foundation of China under Grant nos. 61174103, 61272357, and 61004021, the National Key Technologies R&D Program of China under Grant no. 2015BAK38B01, the Aerospace Science Foundation of China under Grant no. 2014ZA74001, and the Fundamental Research Funds for Central Universities under Grant no. FRF-TP-11-002B.

* Corresponding author.
E-mail address: xluo@ustb.edu.cn (X. Luo).

http://dx.doi.org/10.1016/j.neucom.2015.03.112
0925-2312/

structure, they do not provide a unified NN framework for both regression and classification problems. Miche et al. proposed an optimally pruned ELM called OP-ELM for regression and classification in [11]. Actually, it is also a regularized ELM through the use of 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. In fact, it just selects one variable from the group, which may lead to a suboptimal model finally.

In order to develop a more appropriate NN structure to overcome the hurdle that a suboptimal model faced, 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 with the purpose of improving the robustness and generality of the model. Based on the L_1 norm, the L_2 penalty is also introduced to ELM for the purpose to encourage a group effect, then groups of correlated hidden nodes can be selected. Therefore, our proposed L_1–L_2-ELM benefits from ridge regression and the tendency towards sparse solution of the L_1 penalty, and it can generate a more suitable NN structure than OP-ELM. And elastic net algorithm is used to solve these mixed penalties [12]. Here, the multiple-class classification problem is transformed into a multiple-output regression problem. In general, elastic net is applied to deal with the special cases that have the single response variable. Moreover, it is obvious that different hidden nodes have varying degrees of relativity to the output nodes of ELM, some of hidden nodes may be weakly correlated with a certain output node, and some of other hidden nodes may be weakly correlated with another output node. Thus, for different output nodes, we may prune different hidden nodes, which can obtain a more appropriate NN structure. Considering the above two points, the separate elastic net algorithm and the Bayesian information criterion (BIC) [13] are adopted to find the optimal model for each response variable. Thus, the proposed algorithm may encourage a grouping effect and produce a sparse model with a higher predict accuracy, which tends to reduce over-fitting and provide a more robust model.

This paper is organized as follows. The SLFN based on ELM and classic regularization methods is analyzed in Section 2. The proposed ELM model based on L_1-norm and L_2-norm regularizations is presented in Section 3. The simulation results and discussion are provided in Section 4. The conclusion is summarized in Section 5.

2. Model description

2.1. ELM-based SLFN

Different from traditional theories that all the parameters of the feedforward NN need to be tuned to minimize the cost function, ELM theories claim that the hidden node learning parameters can be randomly assigned independently and the network output weights can be analytically determined by solving a linear system through the use of the least square method [14]. The training phase can be efficiently completed without time-consuming learning iterations and ELM can achieve a good generalization performance.

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

$$\mathbf{o}_i = \sum_{j=1}^{L} \beta_j G(\mathbf{a}_j, b_j, \mathbf{x}_i), \quad i = 1, 2, \ldots, P \tag{1}$$

where $\mathbf{a}_j$ and b_j are the learning parameters of hidden nodes, and $\beta_j = [\beta_{j1}, \beta_{j2}, \cdots, \beta_{jn}]^{\mathrm{T}}$ is the link connecting the jth hidden node to the output nodes, $G(\mathbf{a}_j, b_j, \mathbf{x}_i)$ is the output of the jth hidden node with respect to the input $\mathbf{x}_i$, $\mathbf{o}_i$ is the actual output of the neural network with respect to the input $\mathbf{x}_i$.

For additive hidden nodes with the sigmoid or threshold activation function $g(x)$: $\mathbb{R} \mapsto \mathbb{R}$, $G(\mathbf{a}_j, b_j, \mathbf{x}_i)$ is given by

$$G(\mathbf{a}_j, b_j, \mathbf{x}_i) = g(\mathbf{a}_j^{\mathrm{T}} \cdot \mathbf{x}_i + b_j), \quad \mathbf{a}_j \in \mathbb{R}^m, b_j \in \mathbb{R} \tag{2}$$

where $\mathbf{a}_j = [a_{j1}, a_{j2}, \ldots, a_{jm}]^{\mathrm{T}}$ is the weight vector connecting the jth hidden node and the input nodes, b_j is the threshold of the jth hidden node.

For radial basis function (RBF) hidden nodes with the gaussian or triangular activation function $g(x)$: $\mathbb{R} \mapsto \mathbb{R}$, $G(\mathbf{a}_j, b_j, \mathbf{x}_i)$ is given by

$$G(\mathbf{a}_j, b_j, \mathbf{x}_i) = g\left(b_j \| \mathbf{x}_i - \mathbf{a}_j \|_2\right), \quad \mathbf{a}_j \in \mathbb{R}^m, b_j \in \mathbb{R}^+ \tag{3}$$

where $\mathbf{a}_j$ and b_j are the center and impact factor of the jth RBF hidden node, $\mathbb{R}^+$ indicates the set of all positive real values. In addition, $\| \cdot \|_2$ denotes the L_2-norm. The SLFN with L hidden nodes can approximate these P samples with zero error, which means that the cost function $E = \sum_{i=1}^{P} \| \mathbf{o}_i - \mathbf{t}_i \|_2 = 0$, i.e., there exist $(\mathbf{a}_j, b_j)$ and β_j such that

$$\mathbf{t}_i = \sum_{j=1}^{L} \beta_j G(\mathbf{a}_j, b_j, \mathbf{x}_i), \quad i = 1, 2, \ldots, P \tag{4}$$

The above P equations can be rewritten compactly as

$$\mathbf{H}\beta = \mathbf{T} \tag{5}$$

where

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_P) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_P) \end{bmatrix}_{P \times L},$$

$$\beta = \begin{bmatrix} \beta_1^{\mathrm{T}} \\ \vdots \\ \beta_L^{\mathrm{T}} \end{bmatrix}_{L \times n} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{t}_P^{\mathrm{T}} \end{bmatrix}_{P \times n}.$$

Here, $\mathbf{H}$ is called the hidden layer output matrix of the SLFN. Thus, the system (5) becomes a linear model and the output weights can be analytically determined by finding a least-square solution of this linear system as follows:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T} \tag{6}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ [15]. Thus, ELM can be summarized as Algorithm 1.

Algorithm 1. ELM.

Input: a training set: $\{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbb{R}^m, \mathbf{t}_i \in \mathbb{R}^n, i = 1, \ldots, P\}$;
hidden node activation function: $g(x)$;
hidden node number: L.

Output β.

1 Assign arbitrary learning parameters of hidden nodes $\mathbf{a}_j$ and b_j, $1 \le j \le L$

2 Calculate the hidden layer output matrix $\mathbf{H}$ based on (5)

3 Calculate the output weights $\beta = \mathbf{H}^{\dagger}\mathbf{T}$.

Although ELM learning 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 tends to generate an over-fitting model.
2. ELM provides weak control capacity and is less stable since it directly calculates the minimum norm least-square solutions.