Neurocomputing 148 (2015) 544–550

Contents lists available at ScienceDirect

Neurocomputing

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

Confidence-weighted extreme learning machine for regression problems

Zhigen Shang *, Jianqiang He

Department of Automation, Yancheng Institute of Technology, Yancheng 224003, Jiangsu, China

ARTICLE INFO

Article history:
Received 15 February 2014
Received in revised form
9 May 2014
Accepted 2 July 2014
Communicated by G.-B. Huang
Available online 14 July 2014

Keywords:
Extreme learning machine
Relative entropy
Gaussian margin machine
Regression

ABSTRACT

Based on Gaussian margin machine (GMM) and extreme learning machine (ELM), confidence-weighted ELM (CW-ELM) is proposed to provide point forecasts and confidence intervals. CW-ELM maintains a multivariate normal distribution over the output weight vector. It is applied to seek the least informative distribution from those that keep the targets within the forecast confidence intervals. For simplicity, the covariance matrix is assumed to be diagonal. The simplified problem of CW-ELM is approximately solved by using Leave-One-Out-Incremental ELM (LOO-IELM) and the interior point method. Our experimental results on both synthetic and real-world regression datasets demonstrate that CW-ELM has better performance than Bayesian ELM and Gaussian process regression.

1. Introduction

Extreme learning machine (ELM), which is an efficient learning algorithm for single-hidden layer feedforward neural networks (SLFNs), has been recently proposed by Huang et al. [1]. ELM randomly initializes the parameters of the hidden layer and the weights of the output layer are analytically computed by using Moore–Penrose generalized inverse. Thus, ELM obtains an extremely low computational time. However, the generalization ability of ELM is influenced by changing the number of hidden neurons. Like other similar models based on feedforward neural networks, ELM also needs to address the problem of determining the optimal number of hidden neurons.

Many approaches have been proposed to determine the most suitable structure of ELM. Deconstructive methods (or pruning methods) are one type of algorithms to solve this problem. Rong et al. proposed a pruned ELM (P-ELM) for classification problems [2]. Miche et al. developed a method named optimally pruned ELM (OP-ELM) in [3] and its improvement in [4]. Deconstructive methods, however, in general are inefficient since the most of time they are dealing with a network that is larger than necessary.

There are also some researchers managing to solve the problem based on constructive methods (or growing methods). Huang and Chen presented the incremental ELM (I-ELM) [5] and its modifications [6–7], which are examples of constructive methods. Wang et al. proposed an algorithm for architecture selection of ELM based on localized generalization error model [8]. But in these methods, the expected training accuracy or the maximum number of hidden neurons needs to be set in advance. Yu et al. proposed a method called Leave-One-Out-Incremental ELM (LOO-IELM) in which the LOO error is directly calculated by the PRESS statistics [9]. LOO-IELM adds hidden nodes one-by-one and stops automatically based on the stop criteria. However, the PRESS has the problem of numerical instabilities because of the use of a pseudo-inverse in the calculation. Fortunately, the Tikhonov-regularized PRESS can eliminate this problem [4].

In many regression problems, it is advantageous to have both point forecasts and confidence intervals (CIs). To derive CIs, neural networks are usually combined with other methods, such as bootstrap methods [10] and Bayesian methods [11]. Bootstrap methods are nonparametric approaches of statistical inference based on re-sampling. Their high computational cost makes them less attractive. Bayesian methods for neural networks have been researched intensively in recent years due to their efficiency and effectiveness [12]. For example, Bayesian neural network was employed for rainfall–runoff modeling in [13] and for short time load forecasting in [14]. It is worth noting that Emilio et al. presented a Bayesian approach to extreme learning machine and proposed Bayesian ELM (BELM) in which a normal distribution was introduced on the output weight vector [15]. Compared with ELM, BELM has the advantages of allowing regularization automatically and producing point forecasts and CIs simultaneously. However, BELM lacks proper adaptability to complex noise (e.g., heteroscedastic noise) since an isotropic normal distribution is

* Corresponding author.
E-mail address: zgshang@ycit.edu.cn (Z. Shang).

http://dx.doi.org/10.1016/j.neucom.2014.07.009
0925-2312/

used. Moreover, Gaussian process regression (GPR) can provide both point forecasts and CIs simultaneously [16]. The hyper-parameters of GPR are estimated by maximizing the likelihood of the samples. Like BELM, GPR also lacks the adaptability to complex noise. Gaussian margin machine (GMM) [17] offers another method for obtaining CIs. GMM, originally proposed for linear classification problems, assumes that the weight vector follows a multivariate normal distribution, and aims to seek the least informative distribution that classifies each training sample with a high probability. The probability that a sample belongs to a certain class is automatically provided by GMM.

The present study proposes confidence-weighted extreme learning machine (CW-ELM) for regression problems by combining GMM and ELM. The output weight vector of CW-ELM follows a multivariate normal distribution. The method aims to seek the least informative distribution from those that keep the targets within the forecast CIs. The covariance matrix of the normal distribution is taken to be diagonal for simplicity, and the simplified problem is approximately solved by two steps. The first step is to implement LOO-IELM and to substitute the results into the simplified problem. That is, the hidden layer parameters of LOO-IELM are used as those of CW-ELM, and its corresponding output weight vector is set to be the mean vector of the normal distribution. The second step is to solve the final problem by using the interior point method [18]. It should be noted that, in LOO-IELM in this study, the Tikhonov-regularized PRESS is applied instead of the PRESS. Like BELM and GPR, CW-ELM offers the CIs automatically. The diagonal covariance matrix used in CW-ELM is more complex than the isotropic one adopted in BELM. Thus, CW-ELM may have proper adaptability to complex noise.

The rest of this paper is organized as follows: Gaussian margin machine is introduced in Section 2, which is followed by Section 3 describing some preliminaries of LOO-ELM and BELM. In Section 4, CW-ELM is proposed. Our CW-ELM is evaluated using synthetic and real-world regression datasets, and CW-ELM is compared with BELM and GPR in Section 5. Section 6 draws the final conclusions.

2. Gaussian margin machine

Suppose the samples $\{(\boldsymbol{x}_i, o_i)\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbf{R}^m$ is a column vector and $o_i \in \{-1, 1\}$ is a scalar output. The weight vector $\boldsymbol{w}_1$ of the linear classifier is assumed to follow a normal (Gaussian) distribution $N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\boldsymbol{\mu}_1 \in \mathbf{R}^m$ is a column vector and $\boldsymbol{\Sigma}_1 \in \mathbf{R}^{m\times m}$ is a definite matrix. For the sample $\boldsymbol{x}_i$, we get

$$\boldsymbol{x}_i^T \boldsymbol{w}_1 \sim N(\boldsymbol{x}_i^T \boldsymbol{\mu}_1, \boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i), \tag{1}$$

where T means matrix transposition.

The linear classifier is required to correctly classify the sample $\boldsymbol{x}_i$ with a high probability, that is

$$\Pr(o_i \boldsymbol{x}_i^T \boldsymbol{w}_1 \geq 0) \geq \delta, \tag{2}$$

where $\delta \in (0.5, \quad 1]$ is a confidence parameter.

Combining Eqs. (1) and (2) yields

$$\Pr\left(\frac{o_i \boldsymbol{x}_i^T \boldsymbol{w}_1 - o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \leq \frac{-o_i \boldsymbol{x}_i{}^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \right) \leq 1-\delta. \tag{3}$$

GMM is designed to seek the least informative distribution that will classify the training samples with high probability, which is implemented by seeking a distribution with minimum relative entropy with respect to an isotropic distribution $N_m(\boldsymbol{0}, a\boldsymbol{I}_m)$, where a is a prior parameter. The optimization problem of GMM can be expressed as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1} \; D_{KL}(N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \| N_m(\boldsymbol{0}, a\boldsymbol{I}_m)) \\ &s.t. \quad \Pr\left(\frac{o_i \boldsymbol{x}_i{}^T \boldsymbol{w}_1 - o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \leq \frac{-o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1}{\sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i}} \right) \leq 1-\delta, \\ &\boldsymbol{\Sigma}_1 > 0, \quad i = 1, \cdots, l, \end{aligned} \tag{4}$$

where D_{KL} stands for the relative entropy of $N_m(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $N_m(\boldsymbol{0}, a\boldsymbol{I}_m)$, which can be calculated by

$$\frac{1}{2} \ln \det(a\boldsymbol{I}_m \boldsymbol{\Sigma}_1^{-1}) + \frac{1}{2} tr\left((a\boldsymbol{I}_m)^{-1} (\boldsymbol{\mu}_1 \boldsymbol{\mu}_1^T + \boldsymbol{\Sigma}_1 - a\boldsymbol{I}_m) \right). \tag{5}$$

By disregarding the constant terms of objective function and transforming the constraints of Eq. (4), the problem can be reformulated as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1} \tfrac{1}{2}(-\ln \det \boldsymbol{\Sigma}_1 + \tfrac{1}{a} tr(\boldsymbol{\Sigma}_1) + \tfrac{1}{a} \|\boldsymbol{\mu}_1\|^2) \\ &s.t. \quad o_i \boldsymbol{x}_i^T \boldsymbol{\mu}_1 \geq \Phi^{-1}(\delta) \sqrt{\boldsymbol{x}_i^T \boldsymbol{\Sigma}_1 \boldsymbol{x}_i} \\ &\boldsymbol{\Sigma}_1 > 0, \quad i = 1, \cdots, l, \end{aligned} \tag{6}$$

where Φ^{-1} denotes the inverse cumulative distribution function of a standard normal distribution.

The two-sided PAC-Bayesian theorem ensures that GMM is of desirable generalization performance, and the proof process is presented by [17].

3. Preliminaries

ELM can be considered as universal approximation, and has been applied in many fields. This section will briefly describe ELM, LOO-IELM and BELM.

3.1. Extreme learning machine

Let $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{l}$ be a sample set where $\boldsymbol{x}_i \in \mathbf{R}^m$ is the ith input vector and $t_i \in \mathbf{R}$ is its corresponding target. In ELM, the hidden layer parameters are randomly initialized. ELM is mathematically modeled by

$$y_i = \boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\mu}_2, \tag{7}$$

where $\boldsymbol{g}(\boldsymbol{x}_i) \in \mathbf{R}^p$ is the output vector of the hidden layer, $\boldsymbol{\mu}_2 \in \mathbf{R}^p$ is the output weight vector, and p is the number of the hidden neurons. ELM computes the output weight vector $\boldsymbol{\mu}_2 = \boldsymbol{H}^{\dagger} \boldsymbol{t}$ by Moore–Penrose generalized inverse, where $\boldsymbol{H} = [g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_l)]^T$, and $\boldsymbol{t} = [t_1, \ldots, t_l]^T$. Thus, ELM has an extremely low computational time. However, Moore–Penrose generalized inverse leads ELM to suffer from the overfitting problem. Regularization is one of methods to improve the generalization performance of ELM. Regularized ELM aims to minimize not only the training error but also the norm of output weights. The regularized methods include Lasso, Tikhonov, and elastic net [9]. In this study, Tikhonov regularization is used, and is described as

$$\begin{aligned} &\min_{\boldsymbol{\mu}_2, \boldsymbol{\xi}} \tfrac{1}{2} \left(C \sum_{i=1}^{l} \xi_i^2 + \|\boldsymbol{\mu}_2\|^2 \right) \\ &s.t. \quad \boldsymbol{g}(\boldsymbol{x}_i)^T \boldsymbol{\mu}_2 - t_i = \xi_i, \quad i = 1, \ldots, l, \end{aligned} \tag{8}$$

where C is the regularization parameter.

Using the KKT conditions, the output weight vector $\boldsymbol{\mu}_2$ can be analytically computed as

$$\boldsymbol{\mu}_2 = \left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}^T \boldsymbol{H} \right)^{-1} \boldsymbol{H}^T \boldsymbol{t}. \tag{9}$$