Regularized ELM bagging model for Tropical Cyclone Tracks prediction in South China Sea

Maocan Yang a, Jun Zhang a, Hong Lu b, Jian Jin a,⇑

a School of Computer Science and Technology, East China Normal University, Shanghai 200062, PR China
b Guangxi Climate Center, Nanning 530022, PR China

Received 18 October 2018; received in revised form 26 June 2019; accepted 22 September 2020

Available online 1 October 2020

Abstract

This paper aims to improve the prediction accuracy of Tropical Cyclone Tracks (TCTs) over the South China Sea (SCS) with 24 h lead time. The model proposed in this paper is a regularized extreme learning machine (ELM) ensemble using bagging. The method which turns the original problem into quadratic programming (QP) problem is proposed in this paper to solve lasso and elastic net problem in ELM. The forecast error of TCTs data set is the distance between real position and forecast position. Compared with the stepwise regression method widely used in TCTs, 8.26 km accuracy improvement is obtained by our model based on the dataset with 70/1680 testing/training records. By contrast, the improvement using this model is 16.49 km based on a smaller dataset with 30/720 testing/training records. Results show that the regularized ELM bagging has a general better generalization capacity on TCTs data set.

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Keywords: Regularized extreme learning machine; Bagging; Lasso; Quadratic programming; Tropical Cyclone Tracks

1. Introduction

The Tropical Cyclone (TC) ranks first in the top ten natural disasters of the world, which often causes heavy casualties and property losses. So, delivering the accurate Tropical Cyclone Tracks (TCTs) forecast is extremely significant to reduce the loss of disasters.

In the past, The forecasting techniques about TCTs can be divided into the following three major categories: (1) statistical; (2) dynamical and numerical; (3) statistical-dynamical (Roy & Kovordányi, 2012). In recent years, Artificial Neural Networks (ANNs) based techniques have been developed which announcing the better forecasting accuracy in TCTs (Ali, Kishtawal, & Jain, 2007; Chaudhuri, Dutta, Goswami, & Middey, 2015; Wang et al., 2011; Zhu, Jin, Cannon, & Hsieh, 2016). However, the prediction should be done as efficiently as possible for the short term TCTs. The learning speed of ANNs is in general far slower than expected. Extreme Leaning Machine (ELM) is a new learning algorithm with fast learning speed and good generalization ability proposed by Huang et al., which has been successfully applied to a number of real-world applications in recent years (Huang, Zhu, & Siew, 2004, 2006; Huang, Wang, & Lan, 2011b). Further, regularized ELM for regression problems is proposed by Escandell-Montero et al., which can obtain the optimal architecture of the network (Escandell-Montero, Soria-Olivas, & Magdalena-Benedito, 2011). The regularized regression methods include \( l_2 \) penalty

* Corresponding author.

E-mail address: jjin@cs.ecnu.edu.cn (J. Jin).
(ridge), $l_1$ penalty (lasso) and the mixture of the both (elastic net). Huang et al. derived the closed-form expression for the ridge problem in ELM (Huang, Zhou, Ding, & Zhang, 2012). The lasso and elastic net problem in ELM is usually solved with the coordinate descent method due to its high speed (Friedman, Hastie, & Tibshirani, 2010).

Inspired by Elad which explained how to solve the lasso problem of Linear Systems, we propose a new method from a new perspective in this paper, which turns the lasso and elastic net problem in ELM into quadratic programming (QP) problem with expanding the scale of independent variable in $l_1$ norm (Elad, 2010). Quadratic programming problems can be solved with many methods in convex optimization (Boyd, Vandenberghe, & Faybusovich, 2006). Encouraging results have been achieved on TCTs data set with ensemble methods (Goerss, 2000; Huang, Jin, & Shi, 2011a; Wang, 2012; Lee and Wong, 2002). The performance of a single neural network can be expected to improve using an ensemble of neural networks with a plurality consensus scheme (Hansen, 1990). Bagging is one of the popular ensemble machine-learning methods. There are two key advantages on bagging algorithm: (1) reducing the learning time greatly with the network trained in parallel. (2) the out-of-bag examples produced by bootstrap sampling, which can be used to estimate the expert (Breiman, 1996; Zhou, 2012).

Here we applied the regularized ELM bagging to the short term forecasting of TCTs recorded in the 44-year period(1960–2003). The forecasting accuracy of our proposed model will be analysed in comparison with that obtained by stepwise regression.

This work is extended from that using a smaller dataset, and more description and analysis are provided (Zhang & Jin, 2018).

2. Methodology

2.1. Basic extreme learning machine

ELM was proposed by Huang et al. for the single hidden layer feedforward neural networks (SLFNs) (Huang et al., 2004, Huang, Zhu, & Siew, 2006). The calculation flow diagram is shown in Fig. 1. The structure of the ELM network is shown in Fig. 2.

For $N$ arbitrary distinct samples $(x_i, t_i)$, where $x_i \in \mathbb{R}^n$ and $t_i \in \mathbb{R}^m, i = 1, \cdots, N$. The hidden layer output matrix $H$ is given by

$$H = \begin{bmatrix} G(w_1, b_1, x_1) & \cdots & G(w_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(w_1, b_1, x_N) & \cdots & G(w_L, b_L, x_N) \end{bmatrix}_{N \times L}$$

where $G(\bullet)$ is the activation function, $W_{N \times L} = [w_1, \cdots, w_L]$ is the weight matrix connecting input layer to hidden layer, $L$ is the number of hidden nodes and $b_i$ is the bias of the $i$-th hidden node in each hidden layer. $W$ and $b_i$ can be initialized randomly without tuning.

Only the weights $B$ in the output layer are to be adjusted which could be done by the Moore–Penrose Generalized Inverse (Banerjee, 1971)

$$B = H^T T.$$  \hspace{1cm} (2)

where $H^T$ is the Moore–Penrose inverse of matrix $H$ and $T = [t_1, \cdots, t_N]^T$

Thus, the essence of basic ELM can be summarized in Algorithm 1.

![Fig. 1. Calculation flow diagram.](image-url)
Algorithm 1. Basic ELM

Require: Training set \( D = \{(x_i, t_i), x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \ldots, N\} \);
activation function \( G(\bullet) \);
the number of hidden nodes: \( L \);
Ensure: weights of input layer: \( \mathbf{W} \), bias of hidden layer: \( \mathbf{b} \),
weights of output layer: \( \mathbf{B} \)
1: Randomly generate the parameters
\((w_i, b_i), i = 1, \ldots, L;\)
2: Calculate the hidden layer output matrix \( \mathbf{H} \) based on
Eq. 1;
3: Calculate the output weight matrix \( \mathbf{B} \) based on Eq. 2;
4: return \( \mathbf{W}, \mathbf{b}, \mathbf{B} \).

2.2. Regularized ELM

The ELM algorithm is based on empirical risk minimization principle which is known as lack of robustness. In order to overcome the drawback, ridge ELM is proposed by Huang et al. (2012), lasso ELM and elastic net ELM is proposed by Escandell-Montero et al. (2011).

2.2.1. Ridge ELM

Ridge regression adds the \( l_2 \) norm of the output layer’s weights on the loss function, which makes all the weights of output layer tend to small. That coincide with Bartlett’s theory, the smaller the weights in the output layer are, the better generalization performance the feedforward neural networks tend to have (Bartlett, 1998). The mathematical model is

\[
\begin{align*}
\text{Min} & : \frac{1}{2} \sum_{i=1}^{m} \xi_i^T \xi_i + \lambda \frac{1}{2} \| \mathbf{B} \|^2_F \\
\text{s.t.:} & \quad \mathbf{t}_i = \mathbf{H} \beta_i, i = 1, \ldots, m
\end{align*}
\]

where
\[
\mathbf{B}_{L \times m} = [\beta_1, \ldots, \beta_m]
\]
\[
\mathbf{t}_i = \mathbf{H} \beta_i
\]
and
\[
\mathbf{T}_{N \times m} = [\mathbf{t}_1, \ldots, \mathbf{t}_m]
\]
\( \lambda \in (0, \infty) \). Based on the Karush-Kuhu-Tucker theorem, different solutions can be obtained according to the size of training set (Huang et al., 2012).

In the case that the size of training set is small relatively, we have
\[
\mathbf{B} = \mathbf{H}^T \left( \frac{\mathbf{I}}{\lambda} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{T}
\]
In the other case that the number of training samples is much larger than the dimension of the feature space, \( N \gg L \), we have
\[
\mathbf{B} = \left( \frac{\mathbf{I}}{\lambda} + \mathbf{H}^T \mathbf{H} \right)^{-1} \mathbf{H}^T \mathbf{T}
\]
In theory, the above solutions can be used in any size of training data sets. The generalization performance of ELM is not sensitive to the dimension of the feature space (L) (Huang et al., 2012). Thus, one may prefer to Eq. 9 in order to reduce computational costs. The detailed algorithm of ridge ELM is shown in Algorithm 2.

Algorithm 2. Ridge ELM

Require: Given training set
\( D = \{(x_i, t_i), x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \ldots, N\} \);
activation function \( G(\bullet) \) in hidden layer;
the number of hidden nodes: \( L \);
hyper-parameters: \( \lambda \).
Ensure weights of input layer: \( \mathbf{W} \), bias of hidden layer: \( \mathbf{b} \),
weights of output layer: \( \mathbf{B} \).
1: Randomly generate the parameters
\((w_i, b_i), i = 1, \ldots, N;\)
2: Calculate the hidden layer output matrix \( \mathbf{H} \);
3: Calculate the output weight matrix \( \mathbf{B} \) based on Eq. 8
or Eq. 9;
4: return \( \mathbf{W}, \mathbf{b}, \mathbf{B} \).

2.2.2. Lasso ELM and ELastic net ELM

The lasso method has a tendency to prefer sparse solutions. We can obtain more zero values about the weights of output layer. The structure of neural network tends to be more compatible with lasso method. The loss function of lasso problem in ELM can be written as

\[
\begin{align*}
\text{Min} & : \frac{1}{2} \sum_{i=1}^{m} \xi_i^T \xi_i + \lambda \frac{1}{2} \| \mathbf{B} \|^2_F
\end{align*}
\]
where \( \lambda \in (0, \infty) \). Because
\[
||B||_1^2 = ||\beta_1||_1^2 + \cdots + ||\beta_m||_1^2
\]
(11)

Bring Eq. 11 into Eq. 10, the lasso function can be rewritten as
\[
\text{Min} : \frac{1}{2} \sum_{i=1}^{m} \xi_i^2 + \lambda \frac{1}{2} ||\beta_m||_1^2
\]
(12)

The Eq. 12 can be divided into m equations to solve
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} \sum_{i=1}^{m} \xi_i^2 + \lambda \frac{1}{2} ||\beta_1||_1^2 \\
\text{s.t.} & : o_i = t_i - \xi_i \\
\end{align*}
\right.
\]
(13)

We can obtain \( B \) by solving the above \( m \) expressions separately. Each of the \( m \) equations can be solved in the same way because they have the same form.

By taking one of the \( m \) equations as example, we illustrate how to solve the lasso problem in ELM with a new trick.
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} \xi_1^2 + \lambda \frac{1}{2} ||\beta_1||_1^2 \\
\text{s.t.} & : o_1 = t_1 - \xi_1 \\
\end{align*}
\right.
\]
(14)

We make \( \beta_i = u - v \) in Eq. 14, \( u, v \in \mathbb{R}^{n} \), if we add non-negative constraints on \( u, v \), if we assume the \( k \)-th entry in both \( u \) and \( v \) is non-zero and positive, then by replacing these two entries by \( u'_k = u_k - v_k \), \( v'_k = 0 \), we will eventually choose \( u'_k, v'_k \) as the final solution in order to minimize the loss function. Thus
\[
\text{Min} : ||\beta_1||_1^2 \text{ is equivalent to Min} : 1^T u + 1^T v
\]
(15)

In this manner, Eq. 14 can be represented by
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} \xi_1^2 + \lambda \frac{1}{2} (1^T u + 1^T v) \\
\text{s.t.} & : o_1 = t_1 - \xi_1 \\
u, v & \geq 0
\end{align*}
\right.
\]
(16)

2.2.3. Elastic net ELM

Elastic net is a mixture of ridge and lasso. The loss function can be described as
\[
\text{Min} : \frac{1}{2} \sum_{i=1}^{N} \xi_i^2 + \lambda \frac{1}{2} ||B||_1^2 + \eta \frac{1}{2} ||B||_2^2
\]
(17)

Similar to the lasso problem solving, we can also decompose this equation into the same \( m \) sub-equations. one of the \( m \) sub-equations can be written as
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} \xi_1^2 + \lambda \frac{1}{2} ||\beta_1||_1^2 + \eta \frac{1}{2} ||\beta_1||_2^2 \\
\text{s.t.} & : o_1 = t_1 - \xi_1 \\
\end{align*}
\right.
\]
(18)

\( \lambda \in (0, \infty), \eta \in (0, \infty) \), Using the same trick as Eq. 16, Eq. 18 can be expressed as
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} \xi_1^2 + \lambda \frac{1}{2} (1^T u + 1^T v) + \eta \frac{1}{2} (u - v)^T (u - v) \\
s.t. & : o_1 = t_1 - \xi_1 \\
u, v & \geq 0
\end{align*}
\right.
\]
(19)

The loss function of Eq. 16 and Eq. 19 are convex function. And obviously, both Eq. 16 and Eq. 19 are QP problems (Boyd et al., 2006). The QP problem can be described as
\[
\left\{
\begin{align*}
\text{Min} & : \frac{1}{2} x^T P x + q^T x + r \\
s.t. & : G x \leq h \\
A x & = b
\end{align*}
\right.
\]
(20)

and we can use quadprod function in MATLAB’s Optimization Toolbox to solve Eq. 20.

The detailed algorithm of lasso and elastic net ELM is shown in Algorithm 3.

**Algorithm 3. Lasso and Elastic net ELM**

**Require:** Given training set
\[D = \{(x_i, t_i), x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \cdots, N\}\];
activation function \( G(\bullet) \) in hidden layer;
the number of hidden nodes: \( L \);
hyper-parameters \( \lambda \);
hyper-parameters \( \eta \) is only for elastic net ELM.

**Ensure:** weights of input layer: \( W \), bias of hidden layer: \( b \),
weights of output layer: \( B \).

1: Randomly generate the parameters
\((w_i, b_i), i = 1, \cdots, L\).
2: Calculate the hidden layer output matrix \( H \):
3: Calculate the output weight matrix \( B \) by solving m sub-problems:
for \( i = 1:m \) do
(i) \( \beta_i \) replaced by \( u - v, u, v \geq 0 \). The initial ith sub-
problem become a QP problem.
(ii) obtain \( \beta_i \) by solving QP problem.
endfor
4: return \( W, b, B \).

2.3. Bagging

The two key components of bagging are bootstrap and aggregation. Bagging adopts bootstrap sampling to generate different experts, that is, the training set is further divided into a training set and a validation set which is different for each expert by putting back the samples, and then the model for each expert can be trained with base learning algorithm. Bagging adopts the most popular strategies for aggregating the model’s outputs of each expert, in other words, voting for classification and averaging for regression (Breiman, 1996; Zhou, 2012). The bagging algorithm is summarized in Algorithm 4.
Algorithm 4. Bagging

Require: Given training set
\[ D = \{(x_i, t_i), x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \cdots, N\}; \]
Base Learning Algorithm \( \Phi \);
The number of experts: \( E \).
ensure: Aggregation of \( E \) experts.
Process:
1: for \( i = 1:E \) do
   (i) obtain training set \( DATA\_TRA \) and validation set \( DATA\_VAL \) by bootstrap sample in \( D \);
   (ii) train expert with \( \Phi \) in \( DATA\_TRA \) and \( DATA\_VAL \).
endfor
2: return the model of \( E \) experts.

3. Experiments & results

3.1. Data set

The TCTs data set used in the experiments is published by China Meteorological Administration which formed in or moved into the SCS in July, August and September during 1960–2003 and last for at least 48 h. The TCTs data is sampled every 12 h since the TC generated in the SCS. A TCT and its changes are associated with its intensity, accumulation and replenish of energy, and various nonlinear changes in its environment flow field, which are referred to as variable in this paper. The variables can be divided into two categories: (1) the climatology and persistence factors representing changes of TC itself, such as changes in the latitude, longitude, and intensity of a TC at 12 and 24 h before prediction time, and (2) the physical variables calculated from the NCEP/NCAR global reanalysis data, representing the ambient flow field of the TC center (Jin, Huang, & Shi, 2010). The observations of TCTs is a position on the surface of our planet, which are latitude and longitude of the earth. The predictors (v1 to v16) and observations (Lat.t and Lon.t) of TCTs data set are listed in Table 1. Our object is to predict the latitude and longitude based on the predictors in the next 24 h.

The total number of TCTs dataset is 1750. The first 1680 samples are used as the training set and the last 70 samples as the testing set.

3.2. Experiment setup

From the above description of the data set, obviously the problem in this paper is a regression problem. Thus, the output of our model is the average of all the experts’ output. In order to obtain better generalization capability, all experts should be able to make a difference while producing good forecasting accuracy. There are two differences between all experts: (i) the training set \( DATA\_TRA \) and the validation set \( DATA\_VAL \), and (ii) hyper-parameters in the base learning algorithm.

The hyper-parameters which have great influence on the performance of the model need to be set before model training. Table 2 shows the hyper-parameters that three regularized ELM have. Table 3 shows the hyper-parameters that unique to each of the three regularized ELM. In the paper, the hyper-parameters of the model are determined by grid search combined with sobol series (Joe & Kuo, 2003; Sobol, 1967), owing to the points sampled by sobol series are better uniform in multidimensional space. In order to choose a reasonable value about the hyper-parameters in the model, we set the initial ranges of each hyper-parameter firstly, and then, \( C \) candidate pairs of points generated by sobol series in the initial hyper-parameters space, which are used as hyper-parameters inputs to the model; next, the first \( k \)

| Table 1 Predictions of the TCTs data set. |
|------------------------------------------|
| Variable | Min. | Max. | Mean | Std. |
|-----------|------|------|------|------|
| v1        | 102.1| 123.0| 114.6| 4.2  |
| v2        | 50.3 | 36.4 | 9.9  | 10.6 |
| v3        | 46.9 | 24.7 | 10.0 | 10.9 |
| v4        | 105.1| 130  | 116.9| 4.9  |
| v5        | 50.3 | 36.4 | 10.2 | 10.8 |
| v6        | 5.8  | 4.2  | 1.1  | 1.2  |
| v7        | 102.8| 124.7| 115.2| 4.3  |
| v8        | 104.2| 128.3| 116.3| 4.6  |
| v9        | 10.8 | 23.5 | 18.5 | 2.5  |
| v10       | 17.4 | 30.3 | 4.0  | 6.5  |
| v11       | 0.0  | 525.2| 50.0 | 62.1 |
| v12       | 10.0 | 25.0 | 18.0 | 2.5  |
| v13       | 10.5 | 5.7  | 2.3  | 2.3  |
| v14       | 10.3 | 23.8 | 18.3 | 2.5  |
| v15       | 9.6  | 25.3 | 17.9 | 2.6  |
| v16       | 0    | 65   | 23.5 | 10.9 |
| Lat.t     | 95.0 | 128.3| 112.4| 4.6  |
| Lon.t     | 11.3 | 30.0 | 19.5 | 2.7  |
The hyper-parameters selection is summarized as Algorithm 5.

**Algorithm 5. Hyper-parameters selection**

**Require:** Given original training set D;
- Base Learning Algorithm Φ;
- The initial ranges of hyper-parameters RH;
- The number of model’s candidates in hyper-parameters selection: C;
- The number of best-performing candidates: k.

**Ensure:** The final ranges of hyper-parameters, DATA_TRA, DATA_VAL.
1: obtain DATA_TRA and DATA_VAL by bootstrap sampling in D.
2: generate C points in RH by sampling with sobol.
3: obtain C models by using each point as the hyper-parameters in Φ.
4: train C models in DATA_TRA.
5: choose the best performing k models in DATA_VAL.
6: the minimum and maximum value are obtained for each hyper-parameter in k models and used as the final ranges of hyper-parameters.
7: return the final ranges of hyper-parameters, DATA_TRA, DATA_VAL.

Finally, taking expert of ridge ELM as an example, we explain the choice about the specific ranges of each hyper-parameter and parameters in the experiment. The hyper-parameter in ridge ELM which is the trade-off between \( \frac{1}{2} \sum_{i=1}^{m} e_i^2 \) and \( \|B\|_2^2 \) and L is the number of hidden nodes. The initial ranges of \( \lambda \) and L are \((0,1000)\) and \((1,1000)\). Obviously, the effects of \( \lambda \in (0,1) \) and \( \lambda \in (1,1000) \) on the experiment results are two different situations. It is unreasonable and a waste of computing resources if we perform hyper-parameters selection on \( \lambda \in (0,1000) \). That’s because, the length of \( \lambda \in (0,1) \) and \( \lambda \in (1,1000) \) varies greatly. 200 models are trained separately in both cases in order to determine whether \( \lambda \in (0,1) \) or \( \lambda \in (1,1000) \). The initial ranges of \( \lambda \) and L to \((0,1)\) and \((1,1000)\) are set respectively based on the experiment results. C = 200, k = 5 are set in the Algorithm 5 empirically in order to determine the final hyper-parameters ranges. The final hyper-parameters and parameters is determined by the best performing candidate among 40 candidates which hyper-parameters generated in the final hyper-parameters ranges. Taking expert of elastic ELM as an example, L’s ranges of elastic ELM are shown in Fig. 3. A final expert can be obtained based on the above mentioned idea.

3.3. Result and analysis

The number of testing set DATA_TES which consist of four TCSs is 70. 70 prediction points of latitude and longitude are obtained after making prediction on DATA_TES with final ensemble model. The mean distance errors (MDE) \( \Delta d \) evaluates the model more intuitively in general, which is calculated by

\[
\Delta d = \sqrt{\Delta x^2 + \Delta y^2} \times 110(\text{km})
\]

where \( E \) is the number of experts, \( \Delta x \) and \( \Delta y \) are mean absolute errors (MAE) between the predictions \( t = [t_1, t_2] \) and observations \( t = [t_1, t_2] \) of longitude and latitude on the DATA_TES. Stepwise regression which has good performance in meteorological data is a kind of multivariable regression, specifically focusing on selecting variables. The results of stepwise regression are commonly used as benchmark in meteorological area. Table 4 shows the MAE of longitude and latitude and the MDE in the ensemble model and the expert in ensemble model, which indicates that ensemble model improve model’s performance significantly compared with a single expert. The MDE of Lasso ELM bagging and stepwise regression are shown in Fig. 4. We can see that the most MDE of the lasso ELM bagging is lower than the MDE of the stepwise regression from Fig. 4.
MDE between lasso ELM ensemble and stepwise have no much difference, but the gap between the structure of the network is obvious. Table 5 shows the average of number of hidden nodes in regularized ELM ensemble models. The results shows that the elastic ELM ensemble is able to find more efficient networks (similar accuracy and more compact structure). The MDE of the first DATA_TST is nearly 400 km in all predication models, which is an unacceptable forecast. we think the first DATA_TST is a singular TC which is difficult to make accurate predictions. Regularized ELM ensemble models have more consistent performance compared to its base learner, the comparison of single model and integrated model are shown in Fig. 5–7. Fig. 8 shows comparison of ridge ELM’s training error and testing error. Table 6 shows that experimental results of regularized ELM ensemble, stepwise regression.

Table 4
Comparison of the MAE in the ridge, lasso and elastic net ELM ensemble and its expert.

| Method                  | MAE(Lon.) | MAE(Lat.) | MDE     |
|-------------------------|-----------|-----------|---------|
| Ridge ELM ensemble      | 1.0602    | 0.7059    | 140.11  |
| Expert in ridge ELM ensemble | 1.1223    | 0.7516    | 148.50  |
| Lasso ELM ensemble      | 1.0635    | 0.7081    | 140.54  |
| Expert in lasso ELM ensemble | 1.1119    | 0.7263    | 146.08  |
| Elastic net ELM ensemble| 1.0504    | 0.7020    | 138.97  |
| Expert in elastic net ELM ensemble | 1.1053    | 0.7097    | 144.48  |

Table 5
Comparison of the average number of hidden nodes in regularized ELM ensemble which include ridge, lasso and elastic net.

| Method     | Average Number of Hidden Nodes |
|------------|--------------------------------|
| Ridge      | 456.1                          |
| Lasso      | 347.5                          |
| Elastic net| 144.8                          |

Fig. 3. $L$’s range of elastic ELM.

Fig. 4. The MDE of regularized Lasso ELM ensemble and stepwise regression on DATA_TST.
indicators. Regularized ELM ensemble is more accurate than the stepwise regression, the MDE dropped 8.26 km.

4. Conclusion

In this study, a novel algorithm for solving lasso ELM and elastic net ELM is presented. If we hope the network is compact, elastic ELM is a good choice. Compared to lasso ELM and elastic net ELM, ridge ELM only takes a little training time because it does not need to solve the QP problem. However, ridge’s forecast accuracy is almost equivalent to stepwise. The regularized ELM bagging has been applied to the forecasting of TCTs over the SCS. Stepwise regression analysis is then used for comparison.
The comparison results show that most models in this paper are superior to the stepwise regression model in forecast accuracy. The data in TCTs data set is trained together in the model, because we cannot recognize whether it is a non-singular TC. In the further work, we will try to normalize and cluster the DATA first effectively to avoid the influence between singular TC and non-singular TC (Chen, Wang, Zhang, Yang, & Wang, 2019; Jin, Li, & Jin, 2015).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is funded by the Science and Technology Commission of Shanghai Municipality (No. 18511106202), the National Natural Science Foundation of China (Grant No. 41565005, 61907015) and the Natural Science Foundation of Guangxi Province(Grant No. 2018GXNSFAA281229).

References

Ali, M. M., Kishawal, C. M., & Jain, S. (2007). Predicting cyclone tracks in the north Indian Ocean: An artificial neural network approach[J]. Geophysical Research Letters, 34(4), 545–559.

Banerjee, K. S. (1971). Generalized inverse of matrices and its applications. Technometrics, 15(1), 197.

Bartlett, P. L. (1998). The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. IEEE Transactions on Information Theory, 44(2), 525–536.

Boyd Vandenbergh & Faybusovich (2006). Convex optimization. IEEE Transactions on Automatic Control, 51(11), 1859.

Breiman, L. (1996). Bagging predictors. Kluwer Academic Publishers.

Chaudhuri, S., Dutta, D., Goswami, S., & Middey, A. (2015). Track and intensity forecast of tropical cyclones over the North Indian Ocean with multilayer feed forward neural nets. Meteorological Applications, 22(3), 563–575.

Chen, Siguang, Wang, Zhihao, Zhang, Haijun, Yang, Geng, & Wang, Kun (2019). Fog-based optimized kronecker-supported compression design for industrial IoT. IEEE Transactions on Sustainable Computing. https://doi.org/10.1109/TSUSC.2019.2906729.

Elad, M. (2010). Sparse and redundant representations. New York: Springer (pp. 3–14).

Escandell-Montero, P., Soria-Olivas, E., & Magdalena-Benedito, R. (2011). Letters: Regularized extreme learning machine for regression problems. Neurocomputing, 74(17), 3716–3721.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1.

Goerss, J. S. (2000). Tropical cyclone track forecasts using an ensemble of dynamical models. Monthly Weather Review, 128(4), 1187.

Hansen, L. K. (1990). Neural network ensemble. IEEE Trans Pattern Analysis & Machine Intelligence, 12.

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In Proceedings of the international joint conference on neural networks (Vol. 2, pp. 985–990).

Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1–3), 489–501.

Huang, X. Y., Jin, L., Shi, X. M., et al. (2011a). A nonlinear artificial intelligence ensemble prediction model based on EOF for Typhoon track International Joint Conference on Computational Sciences & Optimization. IEEE, 1329–1333.

Huang, G. B., Wang, D. H., & Lan, Y. (2011b). Extreme learning machines: A survey. International Journal of Machine Learning & Cybernetics, 2(2), 107–122.

Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems Man & Cybernetics Part B, 42(2), 513.

Jin, L., Huang, X., & Shi, X. (2010). A study on influence of predictor multicollinearity on performance of the stepwise regression prediction equation. Journal of Meteorological Research, 24(5), 593–601.

Table 6

Experimental results of the regularized ELM ensemble, stepwise regression.

| Method             | MAE(Lon.) | MAE(Lat.) | MDE   |
|--------------------|-----------|-----------|-------|
| Stepwise Regression| 1.1172    | 0.7371    | 147.23|
| Ridge ELM ensemble | 1.0602    | 0.7059    | 140.11|
| Lasso ELM ensemble | 1.0635    | 0.7081    | 140.54|
| Elastic net ELM ensemble | 1.0504    | 0.7020    | 138.97|

Fig. 8. Comparison of ridge ELM’s training error and testing error.
Jin, Jian, Li, Ming, & Jin, Long (2015). Data normalization to accelerate training for linear neural net to predict tropical cyclone tracks. *Mathematical Problems in Engineering, 2015*(5), 1–8.

Joe, Stephen, & Kuo, Frances Y. (2003). Remark on algorithm 659: Implementing Sobol’s quasirandom sequence generator. *ACM Transactions on Mathematical Software, 29*(1), 49–57.

Lee, T. C. & Wong, M. S. (2002). The use of multiple-model ensemble techniques for tropical cyclone track forecast at the Hong Kong Observatory. In *WMO commission for basic systems technical conference on data processing and forecasting systems* (pp. 554–565(12)).

Roy, C. & Kovordányi, R. (2012). Tropical cyclone track forecasting techniques? A review. *Atmospheric Research, 104–105*(1), 40–69.

Sobol, I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals? *USSR Computational Mathematics & Mathematical Physics, 7*(4), 86–112.

Wang, Q. (2012). The study on ensemble prediction of typhoon track. *Journal of the Meteorological Sciences*.

Wang, Y., Zhang, W., & Fu, W. (2011). Back Propogation(BP)-neural network for tropical cyclone track forecast. 1–4.

Zhang, J. & Jin, J. (2018). Regularized extreme learning machine ensemble using bagging for tropical cyclone tracks prediction. In Y. Peng, K. Yu, J. Lu, X. Jiang (eds.), *Intelligence science and big data engineering*. *IScIDE 2018. Lecture Notes in Computer Science* (Vol. 11266). Cham: Springer.

Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. Taylor & Francis.

Zhu, L., Jin, J., Cannon, A. J., & Hsieh, W. W. (2016). Bayesian neural networks based bootstrap aggregating for tropical cyclone tracks prediction in south China sea. In *International Conference on Neural Information Processing* (pp. 475–482). Cham: Springer.