2013 10th IEEE International Conference on Control and Automation (ICCA)
Hangzhou, China, June 12-14, 2013

A Systematic Method to Guide The Choice of Ridge Parameter in Ridge Extreme Learning Machine*

Meng Joo Er[1] and Zhifei Shao[2] and Ning Wang[3]

Abstract— **Extreme Learning Machine (ELM) has attracted many researchers as a universal function approximator because of its extremely fast learning speed and good generalization performance. Recently, a new trend in ELM emerges to combine it with ridge regression, which has been shown improved stability and generalization performance. However, this ridge parameter is determined through a trial-and-error manner, an unsatisfactory approach for automatic learning applications. In this paper, the differences between ridge ELM and ordinary Neural Networks are discussed as well as special properties of ridge ELM and various approaches to derive the ridge parameter. Furthermore, a semi-cross-validation ridge parameter selection procedure based on the special properties of ridge ELM is proposed. This approach, termed as Semi-Cross-validation Ridge ELM (SC-R-ELM), is also demonstrated to achieve robust and reliable results in 11 regression data sets.**

I. INTRODUCTION

Extreme Learning Machine (ELM) has attracted a large amount of research attention since last decade because of its extremely fast learning speed and good generalization performance. The unique feature of ELM compared to existing Neural Networks is that the input weights of its hidden neurons can be randomly generated rather than tuned, resulting in a faster learning speed [1].

Recently, a new trend in ELM emerges to combine it with ridge regression [2], and good testing accuracies can be achieved as long as the number of neurons is large enough [3]. The benefit of doing so is to transform the selection of ELM structure into the selection of ridge parameter, which is much easier and more efficient since only a single variable needs to be defined. However, right now this ridge parameter is tuned manually through a trial and error manner, which is impractical for automated learning algorithms and human errors might also be involved in the selection procedure. Traditionally, ridge regression is used to estimate the parameters of a linear regression model. For the cases when high collinearity/ high dimensionality (dimension higher than the number of samples) exist in the variables, it can ensure the $\mathbf{H^T H}$ invertibility by adding a constant $\frac{1}{C}$ to the sum of squared coefficients [1] [4]. Possibly because of the difficulty of handling more than one variable, this method has rarely been applied in Neural Network algorithms. However, since performance of ridge ELM is not sensible to the change of Neural Network structure, it provides the foundation to combine ELM and ridge regression.

The research on finding a suitable ridge estimator has been going on for decades since the seminal publications. Three principle method exists for choosing C:

1) Graphical driven, as in ridge trace [2], a subjective method.
2) Data driven, as in [5] and [6], an objective method.
3) Cross-validation, a computational intensive method [7].

The ridge trace is a plot of regression estimators vs. ridge estimator C. According to the plot, coefficients appear to change rapidly as C varies when data is collinear, and the rate slows down to zero as C decreases. Then the optimal value is picked from the point where the curve starts to stabilize [4]. However, this process is mainly subjective, since different person may have different opinions as where the coefficients stabilize. In fact, the differences could be large and affect the corresponding estimator performance.

Data driven methods use mathematical formula to derive C, and therefore it is an objective method. Various algorithms in this category have been proposed [5], [6], [8], [9]. It has the advantage of fully utilizing the provided data and fast calculation without user intervention. However, no consensus method offers a universally optimum solution [10]. For most cases, the ridge estimator performance has to be verified before usage and therefore lose efficiency.

Apart from trial and error, a more stable approach is to use cross-validation, including leave-one-out and n-fold. n-fold is more popular since it has less computational requirement. Even though, n-fold-cross-validation requires roughly $n \times k$ times more computational power compared to simple ridge ELM (considering k ridge parameter candidates provided).

There are several interesting properties can be observed in the behavior of ridge ELM. One of them is that for a ridge ELM with large enough hidden nodes, the testing errors generally Cmonotonically decreases as C increase from zero before hitting the optimal value. Base on this special property, we proposed an algorithm termed as Semi-Cross-validation ridge ELM (SC-R-ELM), which can produce similar stable results as n-fold-cross-validation while require much less computational power. Furthermore, 11

*This work is supported by the National Natural Science Foundation of China (under Grant 51009017), Applied Basic Research Funds from Ministry of Transport of P. R. China (under Grant 2012-329-225-060), China Postdoctoral Science Foundation (under Grant 2012M520629), and Fundamental Research Funds for the Central Universities of China (under Grant 2009QN025, 2011JC002).

[1]Meng Joo Er is with Marine Engineering College, Dalian Maritime University, Dalian, China & School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore emjer@ntu.edu.sg

[2] Zhifei Shao is with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore zshao1@e.ntu.edu.sg

[3] Ning Wang is with Marine Engineering College, Dalian Maritime University, Dalian, China n.wang.dmu.cn@gmail.com

[1] H is the hidden layer matrix of ELM and C is the ridge parameter.

regression datasets are selected to compare the performances between SC-R-ELM and other approaches, including manual tuning, cross-validation, graphical and data driven methods.

II. Preliminaries

A. Extreme Learning Machine

ELM can be considered as a generalized single hidden layer feedforward Neural Network. Compared to traditional Neural Networks, the unique feature of ELM is that the input weights of its hidden neurons $(\boldsymbol{\omega}_i, b_i)$ can be randomly generated instead of being tuned [11], and therefore greatly reduces the tuning effort and accelerates the learning speed.

Figure 1 shows a typical ELM structure. The output y with L hidden nodes can be represented by:

$$y = \sum_{i=1}^{L} \beta_i g_i(\mathbf{x}) = \sum_{i=1}^{L} \beta_i G(\boldsymbol{\omega}_i, b_i, \mathbf{x}) = \mathbf{H}\boldsymbol{\beta} \tag{1}$$

where $\mathbf{x}, \boldsymbol{\omega}_i \in \mathbb{R}^d$ and g_i denotes the i^{th} hidden node output function $G(\boldsymbol{\omega}_i, b_i, \mathbf{x})$; $\mathbf{H}$ and $\boldsymbol{\beta}$ are the hidden layer output matrix and output weight matrix respectively. For N distinct samples (x_j, t_j), $j = 1, \ldots N$, Equation 1 can be written as:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2}$$

where

$$\mathbf{H} = \begin{bmatrix} G(\boldsymbol{\omega}_1, b_1, x_1) & \cdots & G(\boldsymbol{\omega}_L, b_L, x_1) \\ \vdots & \cdots & \vdots \\ G(\boldsymbol{\omega}_1, b_1, x_N) & \cdots & G(\boldsymbol{\omega}_L, b_L, x_N) \end{bmatrix}_{N \times L} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{N \times 1} \tag{4}$$

where $\mathbf{T}$ is the target matrix.

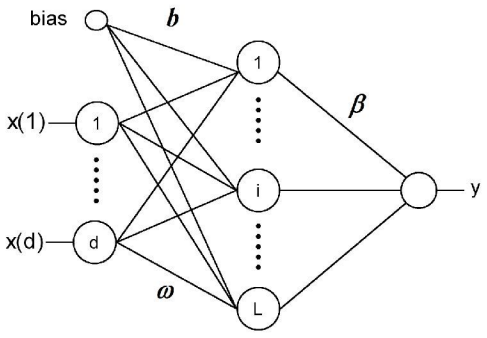

Fig. 1. ELM network structure

Since the input weights of its hidden neurons $(\boldsymbol{\omega}_i, b_i)$ can be randomly generated rather than tuned [3], the only parameters that need to be calculated in ELM is the output weight matrix $\boldsymbol{\beta}$, which can be easily done through Ordinary Least Squares (OLS):

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{5}$$

where $\mathbf{H}^{\dagger}$ is the *Moore-Penrose generalized inverse* of matrix $\mathbf{H}$ [12], which can be calculated through various methods [13]. The common approach is orthogonal projection, where $\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathbf{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathbf{T}}$ [3].

The procedure of ELM goes as follows:

1) Randomly generate hidden neuron parameters $(\boldsymbol{\omega}, \boldsymbol{\beta})$.
2) Calculate hidden layer matrix $\mathbf{H}$ through Equation 3.
3) Calculate the output weight $\boldsymbol{\beta}$ using Equation 5.

B. Ridge ELM

According to ridge regression theory [2], more stable and better generalization performance can be achieved by adding a positive value $\frac{1}{C}$ to the diagonal elements of $\mathbf{H}^{\mathbf{T}}\mathbf{H}$ when calculating the output weight $\boldsymbol{\beta}$ [3], [14]. Therefore, the corresponding ELM with ridge regression becomes:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{T}\mathbf{H} + \frac{\mathbf{I}}{C})^{-1}\mathbf{H}^{T} \tag{6}$$

It has been shown that Equation 6 actually aims at minimizing $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2 + \frac{1}{C}\|\boldsymbol{\beta}\|^2$ [3]. Comparing to OLS, in which the target is to minimize $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2$, an extra penalty term $\frac{1}{C}\|\boldsymbol{\beta}\|^2$ is added to the target of ridge ELM. This is actually consistent to the theory that smaller output weights $\boldsymbol{\beta}$ play an important role for ELM in achieving better generalization ability [15], [16].

The procedure of ridge ELM goes as follows:

1) Randomly generate hidden neuron parameters $(\boldsymbol{\omega}, \boldsymbol{\beta})$.
2) Calculate hidden layer matrix $\mathbf{H}$ through Equation 3.
3) Calculate the output weight $\boldsymbol{\beta}$ using Equation 5 with $\mathbf{H}^{\dagger}$ derived from Equation 6.

To better illustrate the relationship between generalization ability and L and C, Figure 2 shows training vs. testing error across a wide range of C and L using Boston Housing dataset (Table I). Normally the optimal $\frac{1}{C}$ falls within $[0, 1]$ [17], therefore to search more intensively for $C > 1$, the C value is in the form of power of 2, same as in [3].

for

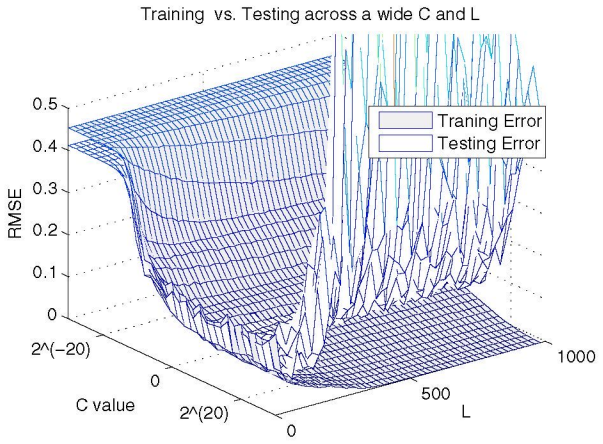

Fig. 2. Training and testing results comparison using housing dataset

It can be clearly observed that the C value has a big influence on the generalization of ridge ELM. As $C \rightarrow \infty$, meaning ridge ELM approaches the ordinary ELM, the overtraining problem becomes increasingly severe. However,

as shown in Figure 3, the testing error with optimal C has little relation with the size of L when $L > 750$.

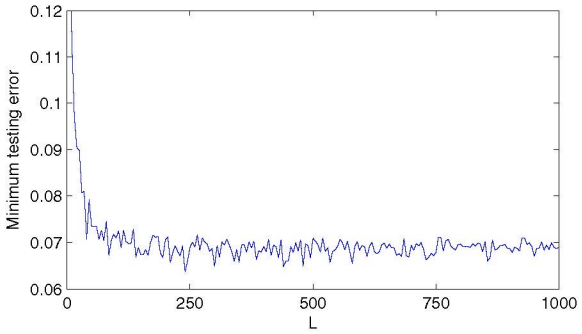

Fig. 3. Minimum testing error with optimal C across a wide range of L

Therefore the generalization ability of ridge ELM is not sensitive to the feature space dimensionality, given an appropriate C value is chosen. This observation is consistent with the statement in [3] that ridge ELM requires less human intervention compared to Support Vector Machine [18] and Neural Networks, since only one parameter C needs to be specified, given L is large enough (e.g., $L \geq 1000$ as in [3]).

Another important property observed in the ridge ELM behavior is that its generalization ability monotonically improving as C increases from zero before hitting the optimal value, when L is large enough. Figure 4 shows the generalization performances of ridge ELM with C goes from 2^{-30} to 2^{30}, with L=1000 (large enough) and L=10 (small).

From Figure 4(a), it can be observed that the testing error monotonically decreases as C increases from 2^{-30} before hitting the optimal solution (around $C = 2^{10}$). On the other hand, from Figure 4(b) with $L = 10$, the generalization performance fluctuates before hitting the optimal solution (worse than the one derived with $L = 1000$). The reason could be that 10 hidden neurons are not enough to explain the feature-target mapping and adding penalty to the output weight is not appropriate.

III. Selection of the Tuning Parameter C

A. Graphical Driven Methods

As the name implies, the graphical driven methods in the selection of tuning parameter C is based on the plot of some properties of the ridge estimator, and a number of approaches have been introduced. Arguably the most famous and popular one is the ridge trace plot [2], which is based on the test of significance of regression coefficients. Others includes p-value trace [19] and Variance Ination Factor (VIF) plot [20].

Apart from the fact that these methods all require users to derive C by observing the plot, and therefore subjective, the plots tend to become messy when the number of predictors is large. Figure 5 shows the ridge trace plot using the same housing data.

Since the number of hidden nodes is quite large ($L = 1000$), it is hard to tell at which C value the plot starts to stabilize, a choice from $C = 2^{10}$ to $C = 2^{0}$ all seem possible.

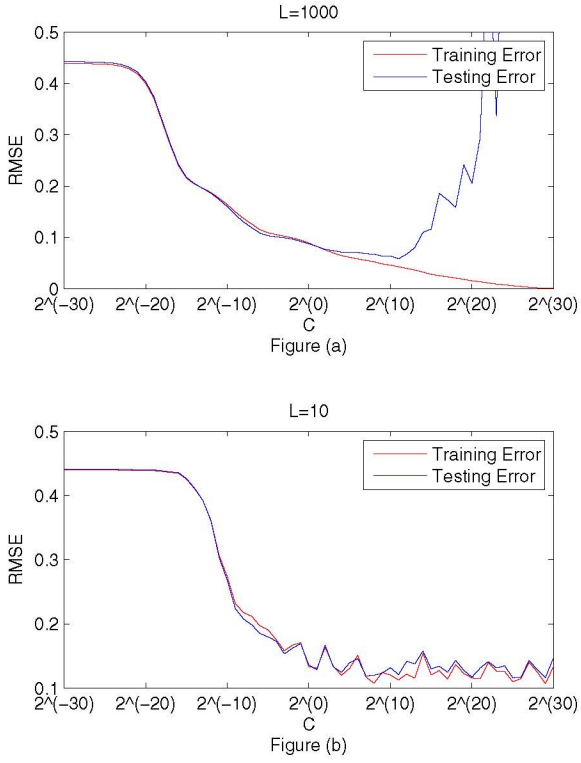

Fig. 4. Generalization performance of ridge ELM with C across a wide range. (a) $L = 1000$ (b) $L = 10$

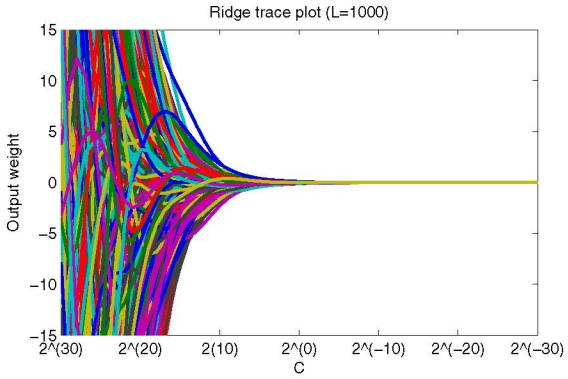

Fig. 5. Ridge trace plot for housing data with $L = 1000$

For traditional ridge regression applications, the graphical driven methods could be applicable since the number of predictors is small. However, ridge ELM requires L to be a sufficiently large number to work properly, therefore these methods are not suitable for our purpose.

B. Data Driven Methods

Data driven methods derive the tuning parameter C through a mathematical formula, which analyze the statistical properties of the data. Their benefits are quite evident. First of all, it is an objective approach without user intervention, which limits the potential of subjective errors and saves human effort. Secondly, the training data can be thoroughly

utilized in a single run, while cross-validation has to further divide the training data. Finally, they often can be executed very fast since the formulas are relatively simple and not requiring cross-validation.

Quite a number of algorithms in this category has been proposed for applications under various conditions. Three notable choices to estimate C are as follows:

- Hoerl and Kennard [2]

$$\frac{1}{C} = \frac{L\hat{\sigma}^2}{\hat{\beta}_{OLS}^T \hat{\beta}_{OLS}}, \tag{7}$$

- Lawless and Wang [6]

$$\frac{1}{C} = \frac{L\hat{\sigma}^2}{\sum_{i=1}^{P} \lambda_i \hat{\beta}_{OLS,i}^2}, \tag{8}$$

- Khalaf and Shukur [8]

$$\frac{1}{C} = \frac{\lambda_{\max}\hat{\sigma}^2}{(N-L-1)\hat{\sigma}^2 + \lambda_{\max}\hat{\beta}_{\max}^2} \tag{9}$$

where $\hat{\beta}_{OLS}$ is the OLS solution of the hidden neuron output weights, λ_i is the i^{th} eigenvalue, $\lambda_{\max}$ is the highest eigenvalue, and $\hat{\sigma}^2$ is the estimated variance of regression error, given by:

$$\hat{\sigma}^2 = \frac{(Y - H\hat{\beta})^T (Y - H\hat{\beta})}{\nu} \tag{10}$$

where $\nu = N - L$, the residual effective degrees of freedom [21]. Using the same Boston Housing dataset as before, the ridge parameter C selected using Equation 7-9, together with the optimal choice is shown in Figure 6.

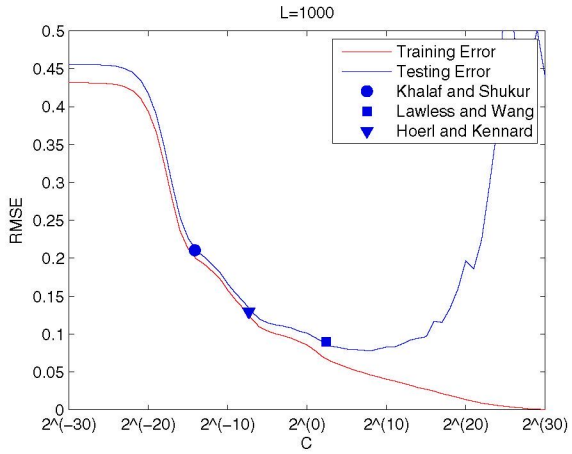

Fig. 6. Data driven algorithms performance comparison for housing data with $L = 1000$

As shown in Figure 6, the Lawless and Wang [6] gives relatively good result, although still has some distance from the optimal value (around $C = 2^8$), and the other two are far from desirable. Since limited information can be extracted from the statistics, data driven methods may still need some form of cross-validation methods to give reliable models, which can greatly slow down the algorithms and lose their original advantages.

C. Cross-validation Method

Cross-validation is a relatively accurate way to estimate the performance of a predictive model. First the data is partitioned into subsets, then the predictive model is derived from some subsets and the performance is tested on the rest. To reduce variability, multiple rounds are performed until all subsets are used for training and testing. The size of the partition is defined by the user, including leave-one-out and leave-k-out. The specific steps using cross-validation to choose ridge parameter C goes as follows [4]:

- Dropping 1 or k samples at each rotation, deriving the model with rest samples and estimating the performance of each choice of C using the dropped samples.
- For each choice of C, the estimated performance is the average RMSE results from all rotations.
- Choosing the C value that gives lowest RMSE.

Cross-validation methods is a computational intensive method. For a dataset partitioned into n subsets with k choices of C values, it requires roughly $n \times k$ times more computational power then the original ridge ELM. However, this method is much more robust than the graphical and data driven methods. Another limitation is that the final result is limited to the choices of parameter candidates, and a compromise between computational resources and accuracy has to be made.

IV. Semi-Cross-validation Ridge ELM

As mentioned in Section II, a beneficial property of ridge ELM is that its testing accuracy generally monotonically improves as C increases from zero before hitting the optimal value, given L is large enough. Using this property, a novel algorithm termed as SC-R-ELM is proposed.

The general idea of SC-R-ELM is shown in Figure 7. Since the testing error monotonically decreases before hitting the optimal solution, the search starts with sparse points from left to right, until RMSE start to increase rather than decrease. Then more intensive search is carried out within the last three sparse points, and the accuracy can be controlled by the searching intensity.

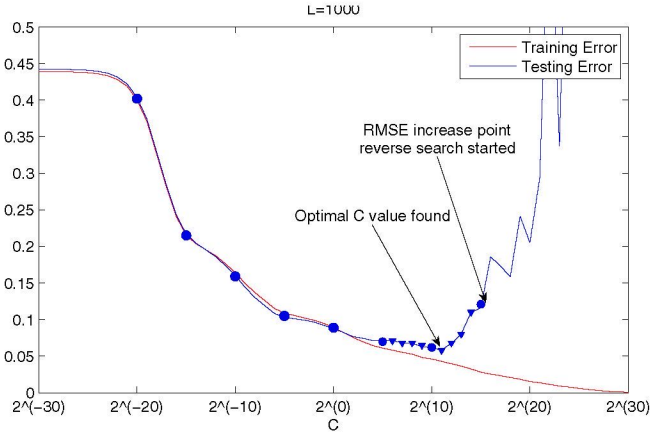

Fig. 7. SC-R-ELM illustration

Similar to cross-validation, the training data is first divided into n subsets. At each rotation, $(n-1)$ sets are used for training and the one dropped is used for testing. The searching process rotated until all data has been used for testing, i.e., n rotations. However, the final solution cannot be obtained the same way as in cross-validation in Section II-C, since many search points are omitted. Therefore the final choice is the average of all optimal value C_k from n rotations. This is possible because there is only one parameter related to each model, which cannot be done with traditional models since they have many parameters. Furthermore, flexible solutions can be created since the final result is no longer confined by user choices. One approach is to average the real values of C_k, termed as SC-R-ELM-Real:

$$C_{final} = \frac{1}{n}\sum_{k=1}^{n} C_k \tag{11}$$

where $C_k = 2^{p_k}$. However, because of the nonlinear distribution of optimal ridge parameters, the average done over the powers of 2 might be a better choice, i.e.:

$$C_{final} = 2^{\frac{1}{n}\sum_{k=1}^{n} p_k} \tag{12}$$

The process of SC-R-ELM is shown in Algorithm 1.

Since the final result is an average rather than a specific predefined choice, a small number can be selected for n (3 for example). On the contrary, normal cross-validation method sets $n = 10$. Comparing to the methods described before, it offers the following advantages:

- No user intervention;
- More robust compared to graphical and data driven methods;
- Reduced computational requirement compared to cross-validation methods;
- The final solution is not limited to user choices;
- Suitable to be implemented into an automated process.

V. Performance Verification

To thoroughly test the performance of SC-R-ELM, 11 datasets for regression, taken from UCI Machine Learning Repository [22] and Statlib [23], are selected (see Table I). Other methods including data driven approach (only Lawless and Wang [6] is shown since others produce very unreliable results), cross-validation, and manually selected ridge ELM are used for performance comparison.

50 trials for each dataset with random permutation are carried out. According to [3], the performance of ridge ELM with sigmoid nodes is not sensitive to the choice of L, as long as it is large enough, therefore L is set uniformly as 1000 for all datasets. To be consistent with the experiments carried out in [3], same ridge parameters for manual approach are used, which may or may not be the same as the C values chosen by other methods. For SC-R-ELM, SC-R-ELM-Real and cross-validation methods, the training data sets is divided into 3 partitions. The testing error results are shown in Table II.

Algorithm 1: Process of Semi-Cross-validation Ridge ELM

Divide training data into n subsets
Define a as the sparse search margin (e.g., 5)
Define b as the intensive search margin (e.g, 1)
for $k = 1$ to n **do**
  Use subset k as testing and the rest for training
  Start sparse searching process
  $C_m = 2^{p_m}, m = 1$
  Ridge ELM to find testing error $RMSE(C_1)$
  while $RMSE(C_m) < RMSE(C_{m-1})$ **do**
    $m = m + 1$
    $p_m = p_m + a$
    $C_m = 2^{p_m}$
    Find $RMSE(C_m)$
  end while
  $m_{\min} = m$
  $p_m - 2a = p_{low}$
  Start intensive searching process
  while $p_m - b > p_{low}$ **do**
    $m = m + 1$
    $p_m = p_m - b$
    $C_m = 2^{p_m}$
    Find $RMSE(C_m)$
  end while
  $m_{\max} = m$
  $C_k = 2^{p_k} = \underset{C_m \in \{m_{\min} \ldots m_{\max}\}}{\arg\min} RMSE(C_m)$
end for
$C_{final} = 2^{\frac{1}{n}\sum_{k=1}^{n} p_k}$ (SC-R-ELM) or $\frac{1}{n}\sum_{k=1}^{n} C_k$ (SC-R-ELM-Real)

From the evaluation results, it can be seen that SC-R-ELM performs better than cross-validation method in 8 out of 11 tests, but with much less computational requirement, and the final result is not confined with user choices, which means a more sparse search points can be used and further reduce the computational cost. Better results could be acquired by cross-validation method using more intensive search points and more validation rotations, e.g., 10-fold-cross-validation, while it is not ideal because of much slower calculation speed. Manually selected parameters can sometimes yield better results, as shown in Table II, but it is impractical since it involves subjective judgement with human error and costs on human resources. The data driven methods offer unreliable results which make them hard to be implemented in real applications.

VI. Conclusion and Future Work

In this paper, the SC-R-ELM algorithm is proposed to offer a systematic method to guide the choice of ridge parameter in ridge ELM. It can substitute the computational intensive cross-validation approach to derive C with reliable solutions. An interesting property of ridge ELM is discussed, that its generalization ability monotonically improves as C

TABLE I

DATASETS FOR SC-R-ELM PERFORMANCE EVALUATION

Dataset	#Attributes	#Training Data	#Testing Data
Basketball	4	64	32
Cloud	9	72	36
Autoprice	9	106	53
Strike	6	416	209
Bodyfat	14	168	84
Cleveland	13	202	101
Housing	13	337	169
Balloon	2	1334	667
Quake	3	1452	726
Space-ga	6	2071	1036
Abalone	8	2784	1393

TABLE II

SC-R-ELM PERFORMANCE EVALUATION AND COMPARISON

Dataset	C	1	2	3	4	5
Basketball	2^0	0.1611	0.1643	0.1617	**0.1605**	0.1606
Cloud	2^{-5}	**0.3021**	0.3045	0.3027	0.3098	1.4420
Autoprice	2^{-1}	0.1737	0.1745	0.1759	**0.1715**	0.5113
Strike	2^{-5}	0.2916	0.2873	**0.2868**	0.3009	2.6931
bodyfat	2^0	**0.0294**	0.0300	0.0298	0.0296	0.0995
Cleveland	2^{-3}	0.1623	0.1621	0.1632	**0.1618**	1.4186
Housing	2^5	0.0754	**0.0749**	0.0755	0.0750	1.8717
Balloon	2^{20}	0.0546	**0.0543**	0.0544	0.0579	0.1700
Quake	2^0	0.1711	0.1704	**0.1703**	0.1713	0.1703
Space-ga	2^4	**0.0338**	0.0340	0.0339	0.0360	0.0354
Abalone	2^0	**0.0760**	0.0760	0.0761	0.0772	0.0765

C: Manual C value; 1: SC-R-ELM; 2:SC-R-ELM-REAL; 3:Cross-validation; 4: Manual selection; 5: Lawless and Wang.

increases from 0 before hitting the optimal value, given L is large enough. SC-R-ELM utilizes this property to reduce the searching effort of cross-validation and therefore reduces the computational requirement. Although the training data in SC-R-ELM is divided into partitions in a similar manner as k-fold-cross-validation, the final solution is obtained as the average (by the power or real value) of all best solutions of each rotation, instead of the best average performance of predefined choices. This is possible because there is only one parameter associated with each model instead of complex structure in traditional Neural Networks. The experiment results on 11 regression datasets shows SC-R-ELM provides robust solutions with performances comparable to cross-validation approach with much less computational cost, and the derived model is not confined by user defined choices.

In future works, we will further investigate the special properties of ridge ELM, and hopefully provide theoretical reasoning. The choice $L = 1000$ appears rather ungrounded, and a clearer guide of choosing L will be studied. Although no consensus optimal solution of choosing ridge parameter has been developed [10], a robust analytical approach is always ideal since it can provide much faster calculation speed. Furthermore, the performance of SC-R-ELM on classification datasets and real applications will be investigated.

REFERENCES

[1] G.-B. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, pp. 1–16, 2011.

[2] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, pp. 55–67, 1970.

[3] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, no. 99, pp. 1–17, 2010.

[4] M. Gruber, *Improving Efficiency by Shrinkage: The JamesStein and Ridge Regression Estimators*. CRC, 1998, vol. 156.

[5] A. Hoerl, R. Kannard, and K. Baldwin, "Ridge regression: some simulations," *Communications in Statistics-Theory and Methods*, vol. 4, no. 2, pp. 105–123, 1975.

[6] J. Lawless and P. Wang, "A simulation study of ridge and other regression estimators," *Communications in Statistics-Theory and Methods*, vol. 5, no. 4, pp. 307–323, 1976.

[7] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *International joint Conference on artificial intelligence*, vol. 14, pp. 1137–1145, 1995.

[8] G. Khalaf and G. Shukur, "Choosing ridge parameter for regression problems," 2005.

[9] M. El-Salam, "An efficient estimation procedure for determining ridge regression parameter," *Asian Journal of Mathematics and Statistics*, vol. 4, no. 2, pp. 90–97, 2011.

[10] E. Cule and M. D. Iorio, "A semi-automatic method to guide the choice of ridge parameter in ridge regression," *Annals of Applied Statistics*, 2012.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[12] D. Serre, *Matrices: Theory and applications*. Springer Verlag, 2010, vol. 216.

[13] C. Rao and S. Mitra, "Generalized inverse of a matrix and its applications," *J. Wiley, New York*, 1971.

[14] K. Toh, "Deterministic neural classification," *Neural computation*, vol. 20, no. 6, pp. 1565–1595, 2008.

[15] P. Bartlett, "For valid generalization, the size of the weights is more important than the size of the network," *Advances in neural information processing systems*, pp. 134–140, 1997.

[16] ——, "The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network," *Information Theory, IEEE Transactions on*, vol. 44, no. 2, pp. 525–536, 1998.

[17] S. Mardikyan and E. Cetin, "Efficient choice of biasing constant for ridge regression," *Int. J. Contemp. Math. Sciences*, vol. 3, no. 11, pp. 527–536, 2008.

[18] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[19] C. Erika and V. Paolo, "Significance testing in ridge regression for genetic data," 2011.

[20] N. Draper, H. Smith, and E. Pownell, *Applied regression analysis*. Wiley New York, 1998, vol. 3.

[21] A. Halawa and M. El Bassiouni, "Tests of regression coefficients under ridge regression models," *Journal of Statistical Computation and Simulation*, vol. 65, no. 1-4, pp. 341–356, 2000.

[22] D. N. A. Asuncion, "UCI machine learning repository," 2007.

[23] P. Vlachos, "Statlib project repository," *Carnegie Mellon University*, 2000.