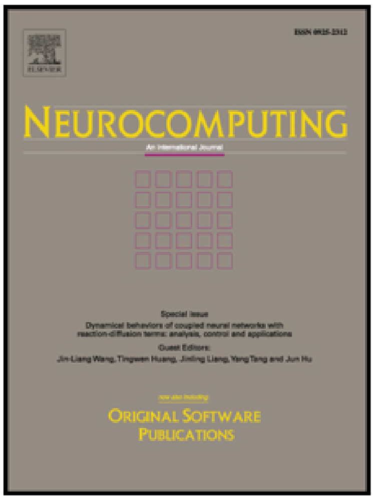
Accepted Manuscript



An Online Semi-Supervised P300 Speller Based on Extreme Learning Machine

Junjie Wang, Zhenghui Gu, Zhuliang Yu, Yuanqing Li

PII: S0925-2312(17)30995-5
DOI: 10.1016/j.neucom.2016.12.098
Reference: NEUCOM 18525

To appear in: *Neurocomputing*

Received date: 23 September 2016
Revised date: 20 December 2016
Accepted date: 23 December 2016

An Online Semi-Supervised P300 Speller Based on Extreme Learning Machine

Junjie Wang[a], Zhenghui Gu[a,*], Zhuliang Yu[a], Yuanqing Li[a]

[a]*School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, 510640*

Abstract

Semi-supervised learning has been applied in brain-computer interfaces (BCIs) to reduce calibration time for user. For example, a sequential updated self-training least squares support vector machine (SUST-LSSVM) was devised for online semi-supervised P300 speller. Despite its good performance, the computational complexity becomes too high after several updates, which hinders its practical online application. In this paper, we present a self-training regularized weighted online sequential extreme learning machine (ST-RWOS-ELM) for P300 speller. It achieves much lower complexity compared to SUST-LSSVM without affecting the spelling accuracy performance. The experimental results validate its effectiveness in the P300 system.

Keywords: brain-computer interface, semi-supervised learning, extreme learning machine

1. Introduction

A brain-computer interface (BCI) allows a person to communicate with or control a computer or other devices without using peripheral nerves and muscles.

*Corresponding author
Email address: zhgu@scut.edu.cn (Zhenghui Gu)
This work was supported by the National Key Basic Research Program of China (973 Program) under Grant 2015CB351703, the National Natural Science Foundation of China under Grants 61573150,61573152,51275094,61633010, and Guangdong Natural Science Foundation under Grants 2014A030312005,2014A030313253.

In the paper, we focus on P300 speller BCI, which is based on oddball paradigm. A rare target stimulus, in particular, a target character flashing, is presented to the user among usual non-target stimuli. The P300 wave can be elicited around 300ms after the target stimulus appears [1]. It is a positive displacement of electroencephalography (EEG) amplitude. Its most famous application in BCI is P300 speller.

Unlike image or text classification task, EEG signals show strong non-stationary characteristic. A pre-trained classifier often has limited generalization performance when applied to a different subject or session. Hence, a long calibration time is needed for each use, which makes it quite inconvenient for BCI users. There are diverse research to solve the problem, e.g., transfer learning finds a way to transfer useful information to target domain. Another way is semi-supervised learning that could make use of unlabeled data to improve the classifier. For example, Li et al. proposed a self-training support vector machine (SVM) algorithm for P300 speller in [2], which used small labeled training data and large unlabeled data to train a strong classifier with spelling accuracy above 95%. Other semi-supervised method for BCI could also be used, e.g., EM with native Bayes classifier, co-training, etc. But these methods assume that all the unlabeled data are available, which makes the algorithms only suitable for offline analysis. Gu et al. proposed a sequential update self-training least squares SVM (SUST-LSSVM) method for online semi-supervised learning in [3]. However, the computational complexity still increases non-linearly with respect to the number of data.

The extreme learning machine (ELM) algorithm is known for its extremely fast learning speed. Hence, we consider the self-training regularized weighted online sequential version of ELM in this paper to reduce the computational complexity. The original ELM algorithm has been proposed by Huang in [4], which was designed to train a single-hidden layer feedforward neural network (SLFN). Unlike the gradient-based algorithm, the input weights and bias of the network are fixed and randomly chosen from a continuous distribution. Then, least squares (LS) method is applied to obtain the output weights. This makes

ELM extremely fast in learning speed, and the empirical studies show that ELM also has a similar generalization performance compared with the classical SVM and LS-SVM [5]. The regularized ELM has been proposed to avoid over-fitting problem in [6] and [7]. In [8], Liang et al. presented an online sequential ELM (OS-ELM). Weighting strategy was also discussed in [9] and [10]. Actually, the work of Zong in [9] contained regularization procedure, which is referred to as regularized weighted ELM (RW-ELM) in this paper. The weighted OS-ELM (WOS-ELM) has been proposed in [11].

In this paper, a self-training regularized WOS-ELM (ST-RWOS-ELM) is employed to train a robust P300 classifier online with low computational complexity. Specifically, after a short supervised training, the speller system is switched into an input mode where unlabelled data are collected. At the same time, an SLFN is updated using the ST-RWOS-ELM algorithm with these unlabelled data after every character input. In our experiment, the spelling accuracy could be quickly improved to around 90% after a few character input. In Section 2, we will give a brief description of the RWOS-ELM algorithm. In Section 3, we introduce the ST-RWOS-ELM and the details of how we apply it in our P300 speller. The experimental results are shown in Section 4. Besides, we make a comprehensive comparison between ST-RWOS-ELM and SUST-LSSVM. Section 5 gives a conclusion of this paper.

2. ELM and its sequential version

For N distinct samples $(\mathbf{x}_i, t_i), i = 1, \ldots, \widetilde{N}$, the hidden layer output of an SLFN with $\widetilde{N}$ hidden nodes can be expressed by an $N \times \widetilde{N}$ matrix

$$\mathbf{H} = \begin{pmatrix} G(\mathbf{a}_1, \mathbf{x}_1, b_1) & \cdots & G(\mathbf{a}_{\widetilde{N}}, \mathbf{x}_1, b_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, \mathbf{x}_N, b_1) & \cdots & G(\mathbf{a}_{\widetilde{N}}, \mathbf{x}_N, b_{\widetilde{N}}) \end{pmatrix} \tag{1}$$

where $\mathbf{a}_i,\ i = 1, \ldots, \widetilde{N}$ is the input weights vector connecting the ith hidden node to the input nodes, $b_i,\ i = 1, \ldots, \widetilde{N}$ is the bias term of ith the hidden node. In ELM, the input weight $\mathbf{a}_i$ and b_i are randomly assigned and fixed [4]. $G(\cdot)$ can be any infinitely differentiable activation function such as sigmoidal function. Each row $\mathbf{h}(\mathbf{x}_i),\ i = 1, \cdots, N$ of $\mathbf{H}$ corresponds to the response of $\widetilde{N}$ hidden nodes to one sample. The mathematical model of a SLFN can be expressed by

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

where $\boldsymbol{\beta}$ is the output weight and $\mathbf{T}$ is the target vector.

For RW-ELM as binary classifier, the output weights $\boldsymbol{\beta}$ are obtained by solving the optimization problem [9]:

$$\begin{aligned} &\min_{\boldsymbol{\beta}}\ \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N} w_i\|\xi_i\|^2 \\ &\text{s.t. } \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = t_i - \xi_i,\ i = 1, \cdots, N \end{aligned} \tag{3}$$

when $N > \widetilde{N}$,

$$\boldsymbol{\beta} = \left(\mathbf{H}^T\mathbf{W}\mathbf{H} + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{H}^T\mathbf{W}\mathbf{T} \tag{4}$$

when $N < \widetilde{N}$,

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\mathbf{H}\mathbf{W}\mathbf{H}^T + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{W}\mathbf{T} \tag{5}$$

where $\mathbf{W} = diag\{w_i\},\ i = 1, \ldots, N$ is the weight matrix. The method of determining the weight matrix for our P300 system is given in Section 3.

WOS-ELM can learn imbalanced data one-by-one or chunk-by-chunk with fixed or varying size. The derivation of WOS-ELM can be found in [11]. Based on this work, we propose RWOS-ELM by introducing a regularization term C to WOS-ELM. And RWOS-ELM has the same optimization problem as RW-ELM. The sequential updating form with the regularization factor C is given as follows:

When a new chunk of data containing M samples comes, the hidden layer output matrix becomes

$$\mathbf{H}_{(n+1)} = \begin{bmatrix} \mathbf{H}_{(n)} \\ \mathbf{H}_M \end{bmatrix} \tag{6}$$

The subscription $(\cdot)_{(n)}$ denotes the nth updating, and the subscription $(\cdot)_M$ denotes the corresponding matrix constructed by the newly coming M samples. When $N+M$ is greater than $\widetilde{N}$, the updated output weight can be expressed as

$$\boldsymbol{\beta}_{(n+1)} = \left(\mathbf{H}_{(n+1)}^T \mathbf{W}_{(n+1)} \mathbf{H}_{(n+1)} + \frac{\mathbf{I}}{C}\right)^{-1} \mathbf{H}_{(n+1)}^T \mathbf{W}_{(n+1)} \begin{bmatrix} \mathbf{T}_{(n)} \\ \mathbf{T}_M \end{bmatrix} \tag{7}$$

where $\mathbf{W}_{(n+1)} = \begin{bmatrix} \mathbf{W}_{(n)} & \\ & \mathbf{W}_M \end{bmatrix}$. Let

$$\mathbf{K}_{(n)} = \mathbf{H}_{(n)}^T \mathbf{W}_{(n)} \mathbf{H}_{(n)} \tag{8}$$

then

$$\begin{aligned} \mathbf{K}_{(n+1)} &= \mathbf{H}_{(n+1)}^T \mathbf{W}_{(n+1)} \mathbf{H}_{(n+1)} \\ &= \mathbf{K}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{H}_M \end{aligned}, \tag{9}$$

and

$$\begin{aligned} &\mathbf{H}_{(n+1)}^T \mathbf{W}_{(n+1)} \begin{bmatrix} \mathbf{T}_{(n)} \\ \mathbf{T}_M \end{bmatrix} \\ &= \mathbf{H}_{(n)}^T \mathbf{W}_{(n)} \mathbf{T}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{T}_M \\ &= \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n)}\right) \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n)}\right)^{-1} \mathbf{H}_{(n)}^T \mathbf{W}_{(n)} \mathbf{T}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{T}_M \\ &= \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n)}\right) \boldsymbol{\beta}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{T}_M \\ &= \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n+1)} - \mathbf{H}_M^T \mathbf{W}_M \mathbf{H}_M\right) \boldsymbol{\beta}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{T}_M \end{aligned} \tag{10}$$

Then the output weight is expressed by

$$\begin{aligned} \boldsymbol{\beta}_{(n+1)} &= \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n+1)}\right)^{-1} \mathbf{H}_{(n+1)}^T \mathbf{W}_{n+1} \begin{bmatrix} \mathbf{T}_{(n)} \\ \mathbf{T}_M \end{bmatrix} \\ &= \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n+1)}\right)^{-1} \left[\left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n+1)} - \mathbf{H}_M^T \mathbf{W}_M \mathbf{H}_M\right) \boldsymbol{\beta}_{(n)} + \mathbf{H}_M^T \mathbf{W}_M \mathbf{T}_M\right] \\ &= \boldsymbol{\beta}_{(n)} + \left(\frac{\mathbf{I}}{C} + \mathbf{K}_{(n+1)}\right)^{-1} \mathbf{H}_M^T \mathbf{W}_M \left[\mathbf{T}_M - \mathbf{H}_M \boldsymbol{\beta}_{(n)}\right] \end{aligned} \tag{11}$$

(9) and (11) present the updating process of RWOS-ELM. The only variable to be stored is the matrix $\mathbf{K} \in R^{\widetilde{N} \times \widetilde{N}}$, whose size is fixed.

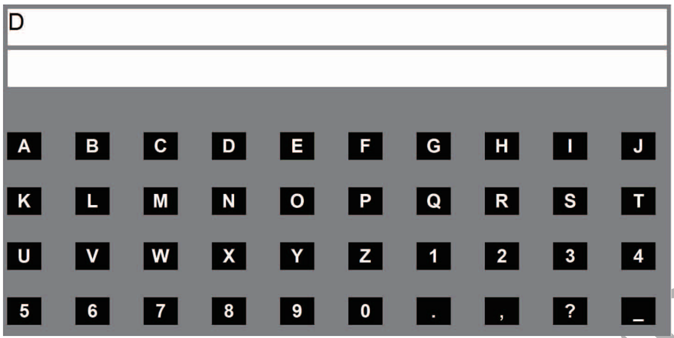


Figure 1: Graphical user interface of the in-house P300 speller.

3. ST-RWOS-ELM applied to P300 speller

The graphical user interface (GUI) of our P300 speller is shown in Fig. 1. A 4×10 matrix involving 40 different characters was presented to the subject on the computer monitor. Each character was intensified successively in a random order. For one character input, 10 rounds of 40 flashes were carried out. The inter-stimulus interval (ISI) was 30ms, and the inter-character gap (ICG) between two character input was 3s. We used the data from 12 channels, including Fz, FCz, Cz, CPz, P7, P3, Pz, P4, P8, O1, Oz, O2, and the channel selection was based on previous experience.

The preprocessing procedures included 0.1 to 20HZ band-pass filtering, down-sampling, and normalizing each feature vector to unit length. At last, these steps yielded the feature vector of size $17*12 = 204$ for each intensification. The feature vector of P300 target was labeled as '+1', and non-P300 target as '-1'.

The data acquired in the online P300 speller have their own characteristics. Firstly, since there is only one target among the 40 characters in the P300 GUI, the ratio of positive and negative samples, i.e., P300 and non-P300 samples, is 1:39 and fixed. Secondly, the size of each data chunk (the samples generated by one character input) is 400 and fixed. Hence, considering the W-ELM, the elements of the weight matrix can be easily chosen as follows

$$w_i = \begin{cases} \frac{1}{\#(positive)}, & i\text{th sample} \in \text{positive class} \\ \frac{1}{\#(negtive)}, & i\text{th sample} \in \text{negtive class} \end{cases} \tag{12}$$

where $\#(positive)$ is the size of positive class, $\#(negtive)$ is the size of negative class.

After applying the weighting procedure, the error caused by positive samples and negative samples tends to be rebalanced, and we can roughly think that the sizes of both classes are equal.

There are two parameters to be determined, i.e., the number of hidden nodes $\widetilde{N}$ and the regularization parameter C. Cross-validation on the offline data was used to find the optimal parameters. The experimental results show that RWOS-ELM algorithm is not sensitive to these parameters. Finally, we choose the number of hidden nodes $\widetilde{N} = 1500$, and the regularization parameter $C = 35000$.

Since the sequentially arriving data samples do not contain label information in the online speller system, self-training mechanism is added into RWOS-ELM, which becomes ST-RWOS-ELM. The online P300 speller paradigm is similar to that in [3], except that we use ST-RWOS-ELM for the sequential update of the classifier instead of SUST-LSSVM. The ST-RWOS-ELM is described in detail as follows:

Step 1: Obtain the hidden layer output matrix $\mathbf{H}_{(0)}$ according to (1), and the output weights $\boldsymbol{\beta}_{(0)}$ according to (5).

Step 2: Upon the arrival of an evaluation data set $\overline{D} = (\mathbf{x}_i, t_i)_{i=1}^{M}$, the trained SLFN is used to obtain the labels t_i, $i = 1, \cdots, M$. Construct the new hidden layer output matrix $\mathbf{H}_{(n+1)}$ according to (6), and the label vector as $\mathbf{T}_{(n+1)} = \begin{bmatrix} \mathbf{T}_{(n)} \\ \mathbf{T}_M \end{bmatrix}$. Use (5) to obtain the updated output weights $\boldsymbol{\beta}_{(n+1)}$.

Step 3: Repeat *Step 2* until $N > \widetilde{N}$.

Step 4: Calculate the matrix $\mathbf{K}_0$ according to (8), and use (4) to obtain the updated output weights $\boldsymbol{\beta}_{(n)}$.

Step 5: Upon the arrival of an evaluation data set $\overline{D} = (\mathbf{x}_i, t_i)_{i=1}^{M}$, the trained SLFN is used to obtain the labels $t_i,\ i = 1, \cdots, M$. Use (11) to update the output weights $\boldsymbol{\beta}_{(n+1)}$, and (9) to update $\mathbf{K}_{(n+1)}$.

Step 6: Repeat *Step 5* until the user stops input.

In our experiment, the initial training set D contained 800 samples from 2 input characters, and the evaluation data set $\overline{D}$ arriving at each update contained 400 samples from 1 input characters.

4. Experimental results

In this section, we study the performance of the proposed online P300 BCI speller based on the ST-RWOS-ELM, and a detailed comparison is given between ST-RWOS-ELM and SUST-LSSVM. The experiment involved 8 subjects including 6 males and 2 females. Each subject was asked to input 2 given characters as initial supervised data, then to input 38 hinted characters. All the experiments were conducted on a computer equipped with Intel i7 processor, 8 GB RAM, and running Matlab 2014a.

In the online experiment, the online spelling accuracy was studied. The prediction accuracy was evaluated by the 38 character input task. The first and second column of Table 1 give the online accuracies of the 8 subjects and the average accuracies for ST-RWOS-ELM and SUST-LSSVM respectively. For most subjects, both methods could achieve high accuracy. Meanwhile, supervised LS-SVM and supervised RW-ELM were also applied to make a comparison, which was only calibrated by the initial 2 character training data. Compared with the semi-supervised methods, the average accuracies of supervised methods lagged behind more than 10%. And the accuracies of LS-SVM and RW-ELM do not differ significantly.

Table 1: Spelling accuracy of the online P300 speller.

	ST-RWOS-ELM	SUST-LSSVM	RW-ELM	LS-SVM
subject1	97.37%	97.37%	92.11%	86.84%
subject2	100%	100%	61.90%	76.19%
subject3	92.11%	94.74%	86.84%	92.10%
subject4	91.89%	89.47%	86.48%	81.08%
subject5	100%	100%	97.37%	100%
subject6	100%	100%	81.58%	92.10%
subject7	100%	100%	100%	97.44%
subject8	97.44%	100%	66.67%	69.23%
average	97.35%	97.73%	84.12%	86.87%

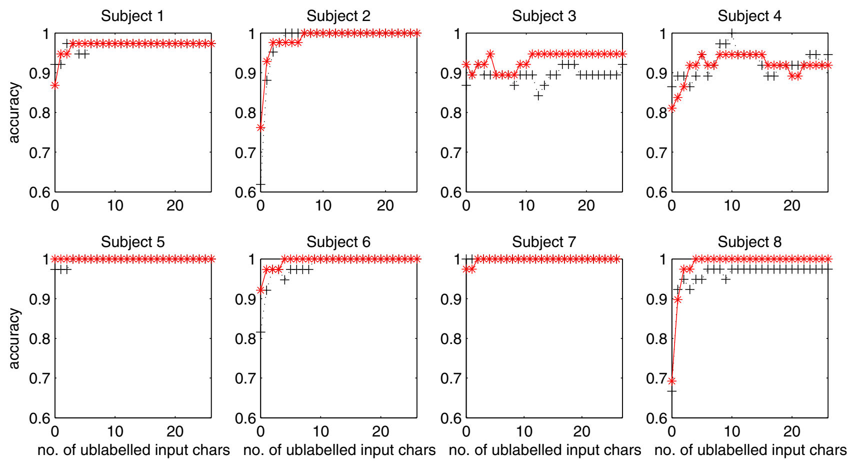


Figure 2: Spelling accuracy of the 8 subjects in offline analysis. The initial supervised training set contains the input data of 2 character input. ($-*-$:SUST-LSSVM, $-+-$:ST-RWOS-ELM)

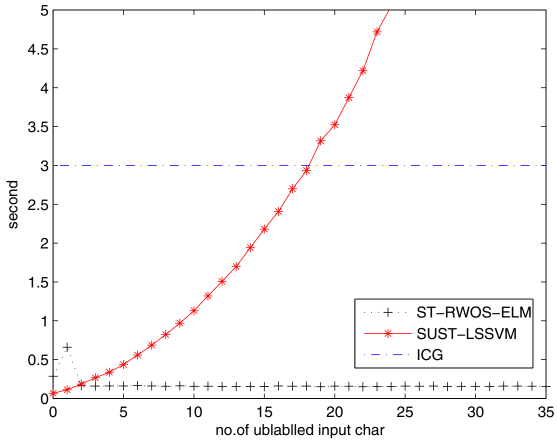


Figure 3: Computation time of each update

In the offline analysis, we studied how the classifier was gradually enhanced by adding unlabeled data, and the results are given in Fig. 2. The test set involved all the 38 input characters. When a new character data came, the classifier was updated and evaluated on the test set. We can find out that with SUST-LSSVM or ST-RWOS-ELM, the accuracy of the system has been improved dramatically with the adding of unlabeled data.

We also made a comparison of the average computation time of the two methods in Fig. 3. It is obvious that the computation time of each update increased non-linearly for SUST-LSSVM. When the number of unlabeled data reaches 20, the computation time will be longer than 3s for SUST-LSSVM. Since the ICG in our system is 3s, which is an upper limit for each update, SUST-LSSVM is not applicable after 20 character updates. While for ST-RWOS-ELM, the computation time is far below the ICG. When the number of training samples N is smaller than the number of hidden nodes $\widetilde{N}$, all the data were retrieved to update the classifier, which caused a small peak in the computation time curve. For $N > \widetilde{N}$, the computation time of ST-RWOS-ELM keeps constant at the value of 0.165 seconds. Hence, ST-RWOS-ELM is more suitable for online semi-supervised P300 speller than SUST-LSSVM in terms of computational complexity.

5. Conclusion

ELM is known for its extremely fast learning speed and good generalization performance, which makes it quite suitable for online learning. Herein, we designed a semi-supervised P300 speller system using ST-RWOS-ELM, which has the advantages of short calibration time and low computational complexity. From the experimental results, the system can provide comparable accuracy to SUST-LSSVM, but with much lower computation complexity. By using this system, the accuracy can be quickly enhanced to around 90%. Such low training effort makes the system more suitable for practical use.

References

References

[1] T. W. Picton, The p300 wave of the human event-related potential, Journal of Clinical Neurophysiology 9 (4) (1992) 456–479.

[2] Y. Li, C. Guan, H. Li, Z. Chin, A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system, Pattern Recognition Letters 29 (9) (2008) 1285–1294.

[3] Z. Gu, Z. Yu, Z. Shen, Y. Li, An online semi-supervised brain–computer interface, IEEE Transactions on Biomedical Engineering 60 (9) (2013) 2614–2623.

[4] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: IEEE International Joint Conference on Neural Networks, 2004. Proceedings. 2004, Vol. 2, IEEE, 2004, pp. 985–990.

[5] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: A review, Neural Networks 61 (2015) 32–48.

[6] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: IEEE Symposium on Computational Intelligence and Data Mining, 2009. CIDM'09, IEEE, 2009, pp. 389–395.

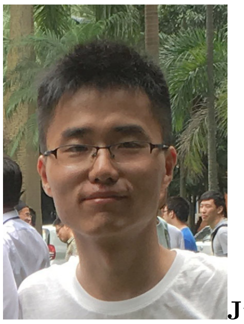
[7] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2012) 513–529.

[8] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423.

[9] W. Zong, G.-B. Huang, Y. Chen, Weighted extreme learning machine for imbalance learning, Neurocomputing 101 (2013) 229–242.

[10] K.-A. Toh, Deterministic neural classification, Neural computation 20 (6) (2008) 1565–1595.

[11] B. Mirza, Z. Lin, K.-A. Toh, Weighted online sequential extreme learning machine for class imbalance learning, Neural processing letters 38 (3) (2013) 465–486.



Junjie Wang received the B.E. degree in automation from the Wuhan University of Science and Technology, Wuhan, China, in 2014. He is currently a postgraduate in the College of Automation Science and Enginerring, South China University of Technology, Guangzhou, China. His research interests include machine learning and its application in brain-computer interfaces.

Zhenghui Gu received the Ph.D. degree from Nanyang Technological University, Singapore, Singapore, in 2003. From 2002 to 2008, she was with the Institute for Infocomm Research, Singapore. In 2008, she joined the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, where she became a professor in 2014. Her current research interests include the fields of signal processing, machine learning and their applications in brain-computer interfaces.

**Zhu Liang YU** received his BSEE in 1995 and MSEE in 1998, both in electronic engineering from the Nanjing University of Aeronautics and Astronautics, China. He received his Ph.D. in 2006 from Nanyang Technological University, Singapore. He joined Center for Signal Processing, Nanyang Technological University from 2000 as a research engineer, then as a Group Leader from 2001. In 2008, he joined the College of Automation Science and Engineering, South China University of Technology and was promoted to be a full professor in 2011. His research interests include signal processing, pattern recognition, machine learning and their applications in communications, biomedical engineering, etc.

**Yuanqing Li** was born in Hunan Province, China, in 1966. He received the B.S. degree in applied mathematics from Wuhan University, Wuhan, China, in 1988, the M.S. degree in applied mathematics from South China Normal University, Guangzhou, China, in 1994, and the Ph.D. degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997. Since 1997, he has been with South China University of Technology, where he became a full professor in 2004. In 2002-04, he worked at the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama, Japan, as a researcher. In 2004-08, he worked at the Laboratory for Neural Signal Processing, Institute for Infocomm Research,

Singapore, as a research scientist. His research interests include, blind signal processing, sparse representation, machine learning, braincomputer interface, EEG and fMRI data analysis. He is the author or coauthor of more than 60 scientific papers in journals and conference proceedings.