A NOVEL LIE DETECTION METHOD BASED ON EXTREME LEARNING MACHINE USING P300

Yijun Xiong[1], Yong Yang[2], Junfeng Gao[3*]

[1] College of Mechanical and Electrical Engineering, Wuhan Donghu University, Wuhan, 430212, China
xyjwsry@163.com
[2] School of Information Technology, Jiangxi University of Finance and Economics, Nanchang, China
[3] College of Biomedical Engineering, South-Central University for Nationalities, Wuhan, 430074, China*
junfengmst@163.com

Keywords: Lie detection, P300, extreme learning machine, Probe stimuli

Abstract

Machine learning-based lie detection has drawn much attention recently. In this paper, we used extreme learning machine (ELM), a recently-proposed machine learning method based on a single layer feedforward network (SLFN), to classify P300 potentials from guilty subject and non-P300 potentials from innocent subject. Back-propagation network and support vector machine classifiers were also used to compare with the proposed method. The number of hidden nodes in ELM was tuned using training with the 10-fold cross validation. The experimental results show that the proposed method reaches the highest classification accuracy with extremely less training and testing time, compared with the other classification models.

1 Introduction

Lie detection has drawn much attention in recent years and has found many important applications in legal and clinical fields [1-2]. Currently, many studies using neurophysiological signals such as Magnetic Resonance Imaging [3] and Event Related Potential (ERP) [4] have been conducted on lie detection. P300 (P3), an endogenous ERP component, has been widely used for lie detection [5].

The adoption of pattern recognition (PR) classifiers in lie detection is not yet widely reported. Davatzikos et al. [6] proposed a support vector machine (SVM)-based method to classify the pattern of brain activity obtained during lying and truth-telling by using fMRI data. Abootalebi et al. [7] used linear discrimination analysis (LDA) to identify P3 responses and obtained the highest detection rate of 86%. Gao et al. [8] used SVM for the first time in the lie detection and the results showed that SVM outperformed fisher discrimination analysis (FDA) and back-propagation neural network (BPNN) by obtaining the highest classification accuracy of 91.8%.

When developing a detection system for EEG task, we should take into account the time of training the classification models. Hence, taking this issue into account, SVM and BPNN may be unsuitable due to their high computational cost [9-10]. Extreme learning machine (ELM) was proposed by Huang et al. [9] to overcome some inherent drawbacks of SVM and BPNN. ELM randomly specifies the input weights and biases, and then analytically calculates the output weights with the smallest norm. Hence, ELM tends to provide good generalization power at extremely faster training speed [11]. During the past several years, ELM has drawn considerable attention in many fields related to PR [12-13].

To date, there is no assessment that uses ELM to detect lying and classify guilty and the innocent subjects. In order to obtain the good balance between high accuracy of lie detection and short training time, ELM is introduced to classify the two kinds of subjects (the guilty and the innocent) in this paper.

For ELM, although the number of hidden nodes (*NHN*) for ELM is randomly assigned in ordinary ELM algorithm, the classification boundary may not be the optimal one when this number remains unchanged during training procedure [14]. In addition, too many or few *NHN* might lead to over-fitting or under-fitting problem [15]. This is one of the hottest issues related to the investigation of ELM algorithm itself and hence a few methods were proposed to solve it [14, 16-18]. For one thing, these improved ELM algorithms are relatively complex for an online or real-time classification system. For another, we assume that there should be a close relation between the *NHN* and the dimension of feature space. Hence, in this study, we did not assign the *NHN* for ELM, but tuned it by using grid searching to decide its optimal values for classification. In addition, our expectation was that the training and testing time could be significant decreased compared with SVM and BPNN.

2 Materials and Methods

2.1 Subjects and EEG acquisition

The experiment was performed on 30 healthy participants (15 female, mean age: 21.5). Pz electrode from an International 10-20 system was used. The vertical EOG (VEOG) signal was recorded from the right eye (2.5 cm below and above the pupil), and the horizontal EOG (HEOG) signal was recorded from the outer canthus. EEG and EOG signals were filtered online with a band pass filter of 0.1–30 Hz, and digitized at 500 Hz using Neuroscan Synamps. All electrodes were referenced to the right earlobe. Electrode impedance was less than 2 kΩ.

2.2 Experimental protocol

We used three stimuli protocol [19] here. Six different jewels and their pictures were served as stimuli during detection. A safe containing one (for the innocent) or two (for the guilty) jewels was given to each participant. The guilty subjects were instructed to steal only one object from the safe, which was served as P stimulus. The other one in the safe was T stimulus; the remaining four pictures were I stimuli. The innocent only memorized the object in the safe that was T stimulus. Then, P stimulus was selected randomly from the remaining five pictures. After that, the remaining four images were I stimuli.
Following above preparation, each stimuli picture was randomly displayed at a screen, which remained 1.1 s with 30 iterations for one session. Each session was about 5 minutes with 2 minutes' resting time. The inter-stimulus interval was 1.6 s. Each subject performed 5 sessions. One push button was given to each subject who pressed "Yes" and "No" button facing with familiar and unknown items, respectively. The guilty group pressed "Yes" and "No" button when facing with T and I stimuli, respectively. With a P stimulus, they were asked to press the "No" button, trying to hide the stolen act. The innocent made honest responses to all the stimuli.

2.3 Feature extraction

Two groups of features based on time-domain, frequency-domain were extracted from each P response (response wave from P stimuli). They are listed as follows: maximum amplitude, latency, latency/amplitude ratio, Minimum amplitude, peak-to-peak, positive area, maximum frequency, mean frequency and the power of frequency band involving the P3. By the feature extraction, two feature sample sets, were obtained with the class label being 1 and -1 from guilty and innocent, respectively. Each feature sample consisted of 9 feature values. Each feature was normalized to [-1, 1] before classification.

2.4 Extreme learning machine

Given N different training samples $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^{\mathrm{T}} \in \mathrm{R}^n$ and $\boldsymbol{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^{\mathrm{T}} \in \mathrm{R}^m$, we train a SLFN with K hidden nodes and activation function $g(x)$. This network can be mathematically modelled as

$$\sum_{i=1}^{K} \beta_i g_i(x_j) = \sum_{i=1}^{K} \beta_i g(a_i \times x_j + b_i) = t_j, \quad j = 1, \ldots, N \tag{1}$$

where $\boldsymbol{a}_i = [a_{i1}, a_{i2}, \ldots, a_{in}]^T$ denotes the weight vector connecting ith hidden node and n input nodes; b_i is bias of ith hidden node; $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ denotes the weigh vector connecting ith hidden node and the m output nodes.
Above N equations can be rewritten in a matrix as

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

where

$$\mathbf{H} = \begin{bmatrix} g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(\boldsymbol{a}_K \cdot \boldsymbol{x}_1 + b_K) \\ \vdots & \cdots & \vdots \\ g(\boldsymbol{a}_1 \cdot \boldsymbol{x}_N + b_1) & \cdots & g(\boldsymbol{a}_K \cdot \boldsymbol{x}_N + b_K) \end{bmatrix}_{N \times K},$$

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_K^T \end{bmatrix}_{K \times m} \text{ and } \mathbf{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix}_{N \times m}. \tag{3}$$

$\mathbf{H}$ is called the hidden layer output matrix with the ith column being the output of the ith hidden node (Huang et al., 2000). To learn N instances for a SLFN, one traditionally find the solution set $\mathbf{W}$, including $\boldsymbol{a}_i$, $\boldsymbol{\beta}_i$ and b_i, to minimize following cost function

$$E(\mathbf{W}) = \sum_{j=1}^{N} \left(\sum_{i=1}^{K} \boldsymbol{\beta}_i g(\boldsymbol{a}_i \cdot \boldsymbol{x}_j + b_i) - \boldsymbol{t}_j \right)^2 \tag{4}$$

Given an arbitrarily small value $\varepsilon > 0$, Huang et al. [9] proved that if the input weights and biases of hidden nodes are assigned randomly and the activation function in the SLFN is infinitely differentiable, the SLFN can approximate the N training data with ε error, i.e., $\|\mathbf{H}\beta - \mathbf{T}\| \le \varepsilon$. In this case, matrix $\mathbf{H}$ has been randomly fixed. Hence, the training procedure of SLFN is equal to seeking a least-squares (LS) solution of the linear system.
ELM has not yet used in the tasks of lying identification and classification. In this study, we focused on the application of ELM on the P300-based lie detection and evaluate its performances by comparing with those of BPNN and SVM classifiers.

2.5 The proposed method

A grid search technique [20] was used in the proposed method to optimize the number of hidden nodes (*NHN*) in ELM. Let K denote the *NHN*, which was initialized 1 in this paper. The proposed method is divided into following steps:

1) Denote the *NHN* of ELM by K and initialize K=1.
2) The feature set is fed into an ELM classifier with K hidden nodes. The training procedure is carried out with the specific parameter K. A grid search of the parameter combination K is performed on the training data. A 10-fold CV is performed, resulting in 10 pairs of training sets and testing sets. Hence, an averaged balanced accuracy, namely *BA_train*, across the 10 pairs of training sets can be obtained by averaging sensitivity accuracy TR_{sen} and specificity accuracy TR_{spe}.
3) Update K by K+1. Repeat *step 2-3* until K equals *30*.
4) Comparing all the *BA_train* obtained in step 2, the optimal parameter combination K is finally obtained when the *BA_train* reached highest.

To evaluate the performance of the proposed method objectively, BPNN and SVM, were also performed. Like the above steps, each model was trained to obtain the optimal classifier parameters when the training accuracy *BA_train* reached highest.
For the BPNN, learning rate η and control precision ε were set to be 0.002 and 0.001, respectively; The Levenberg-

Marquardt algorithm was used; For SVM, the penalty parameter C and radial width σ for radial basis function (RBF) (kernel function $K(x,y) = e^{-1/2*(\|x-y\|/\sigma)^2}$, Burges, 1998) were tuned with the following grid: $C = [2^{-2}, \ldots, 2^{10}]$, $\sigma = [2^{-3}, \ldots, 2^{10}]$ (step size = 2^1). Hence for SVM, it is a three-dimensional grid-searching procedure and more complex than ELM and BPNN.

By the optimization procedure above, the following measures were used to evaluate and compare the performance of 3 classification models:

1) The training accuracy. It consists of sensitivity, specificity and their respective standard deviation (SD). They refer to the results of $TR_{sen} \pm SD$ and $TR_{spe} \pm SD$ when the corresponding *BA_train* reached the highest.
2) The training time of classification models.
3) The optimal classifier parameters. For the BPNN and ELM, it referred to the optimal *NHN*; for SVM, it refers to the number of the support vectors (*NSV*).

3 Results

The experimental result is shown in Table 1. First, it can be seen from the table that the training accuracy of ELM is significantly higher than BPNN and SVM. The TR_{sen} and TR_{spe} of ELM reach 97.83% and 97.75%, respectively. More importantly, ELM spend extremely less time of 151 s than BPNN and SVM, which is nearly 20 times and 500 times faster than BPNN and SVM. Finally, we can see than the optimal *NHN* of ELM is tuned and the result is 38, which is less than 42 for BPNN and less than the *NSV*=52 for SVM.

By testing on the testing sets, sensitivity and specificity of 96.86% and 96.30% are finally obtained, respectively.

Models	Training Accuracy (%)		Training time (s)	*NHN*/ *NSV*
	$TR_{sen} \pm SD$	$TR_{spe} \pm SD$		
BPNN	93.37±3.23	90.42±4.40	2880	42
SVM	95.25±0.35	94.07±0.51	86400	52
ELM	97.83±0.30	97.75±0.48	151	38

Table 1. Experimental result (classification accuracy of the three models with the optimal *NHN*/*NSV* and training time)

4 Discussions

ELM has been used in many benchmark classification or regression problems. However, little work has been conducted to classifying brain state. In this study, a mock crime scenario with three stimuli protocol was performed on 30 guilty and innocent subjects. Then two groups of features were extracted from P responses. Finally, ELM was used to classify guilty subjects.

ELM has some good properties compared with other machine learning algorithms. First, there is almost no classifier parameter needed to be tuned in its learning procedure compared with BPNN and SVM, which significantly lessens the computational cost and training complexity. Second, the learning speed is also very slow when a large number of training data need to learn and hence too slow to meet the requirements of real-time applications. Third, artificial neural networks such as BPNN are prone to fall into a local minimum. In contrast, ELM has a surprising learning speed because the output weights are determined analytically instead of by an iterative strategy (Huang et al., 2004, 2006a). In spite of its advantages over most classifiers, the parameter of *NHN* should be decided before training. In this study, we used a grid searching procedure to optimize this parameter. Experimental results show that ELM outperforms BPNN and SVM not only with the highest training accuracy but the less training time.

Although the lie detection method is proposed in this study, the method reported here should not been limited to the lie detection. The proposed method can be efficiently used in a variety of classifications tasks of brain states such as Brain machine interfaces.

Acknowledgements

The work is supported by National Nature Science Foundation of China (No. 81271659 and No. 61262034), by the Natural Science Foundation of Jiangxi Province (No. 20132BAB201025) and by the Science and Technology Research Project of the Education Department of Jiangxi Province (No. GJJ13302).

References

[1] Gamer, M., and Berti, S., "Task relevance and recognition of concealed information have different influences on electrodermal activity and event-related brain potentials", Psychophysiology, 2010, 47, pp. 355-364.

[2] Ito, A., Abe, N., Fujii, T., et al. "The role of the dorsolateral prefrontal cortex in deception when remembering neutral and emotional events", Neuroscience Research, 2011, 69, pp. 121-128.

[3] Langleben, D.D., Loughead, J.W., Bilker, W.B., et al. "Telling truth from lie in individual subjects with fast event-related fMRI", Human Brain Mapping, 2005, 26, pp. 262-272.

[4] Rosenfeld, J.P., "Event-related potentials in the detection of deception", Handbook of Polygraph Testing, Academic Press, New York, 2002, pp. 265-286.

[5] Meijer, E.H., Smulders, F.T.Y., Merckelbach, H.L.G. Wolf., J.A.G., "The P300 is sensitive to concealed face recognition", International Journal of Psychophysiology, 2007, 66, pp. 231-237.

[6] Davatzikos, C., Ruparel, K., Fan, Y., et al. "Classifying spatial patterns of brain activity with machine learning methods: Application to lie detection", NeuroImage, 2005, 28, pp. 663-668.

[7] Abootalebi, V., Moradi, M.H., Khalilzadeh, M.A., "A new approach for EEG feature extraction in P300-based lie detection", Computer Methods and Programs in Biomedicine, 2009, 94, pp. 48-57.

[8] Gao, J.F., Yan, X.G., Sun, J.C., Zheng, C.X., "Denoised P300 and machine learning-based concealed information test method", Computer Methods and Programs in Biomedicine, 2011, 104, pp. 410-417.

[9] Huang, G.B., Zhu, Q.Y., Siew, C.K., "Extreme learningmachine: theory and applications", Neurocomputing, 2006, 70, pp. 489-501.

[10] Yuan, Q., Zhou, W. D., Li, S. F., Cai. D. M., "Epileptic EEG classification based on extreme learning machine and nonlinear features", Epilepsy Research, 2011, 96, pp. 29-38.

[11] Huang, G.B., Wang, D.H., Lan, Y., "Extreme learning machines: a survey", International Journal of Machine Learning and Cyber, 2011, 2, pp. 107-122.

[12] Deng,W., Chen, L., "Color image watermarking using regularized extreme learning machine", Neural Network World, 2010, 20, pp. 317-330.

[13] Chen, F.L., Ou, T.Y, "Sales forecasting system based on gray extreme learning machine with Taguchi method in retail industry", Expert System Application, 2011, 38, pp. 1336-1345.

[14] Cao, J.W., Lin, Z.P., Huang, G.B., " Composite Function Wavelet Neural Networks with Differential Evolution and Extreme Learning Machine ", Neural Processing Letters, 2011, 33, pp. 251-265.

[15] Shi, L.C., Lu, B.L, "EEG-based vigilance estimation using extreme learning machines", Neurocomputing, http://dx.doi.org/10.1016/j.neucom, 2012.

[16] Huang, G.B., Chen, L. "Enhanced random search based incremental extreme learning machine", Neurocomputing, 2008, 71, pp. 3460-3468.

[17] Miche, Y., Bas, P., Jutten, C., et al., "A methodology for building regression models using extreme learning machine: Op-elm", Proceedings of the 16th European Symposium on Artificial Neural Networks, Bruges, Belgium, 2008, pp. 1-6.

[18] Yuan, L., Yeng, C.S., Huang, G.B., "A constructive enhancement for Online Sequential Extreme Learning Machine", Neural Networks, International Joint Conference, 2009, 1708-1713.

[19] Rosenfeld, J.P., Soskins, M., Bosh, G., Ryan, A., "Simple, effective countermeasures to P300-based tests of detection of concealed information", Psychophysiology, 2004, 41, pp. 205-219.

[20] Hsu, C.W., Lin, C.J., "A comparison of methods for multiclass support vector machines", IEEE Transactions on Neural Networks, 2002, 13, pp. 415-425.