2015 International Symposium on Nonlinear Theory and its Applications
NOLTA2015, Kowloon, Hong Kong, China, December 1-4, 2015

Time-series Prediction and Classification of NIRS Data Using the Extreme Learning Machine

Hiroki Ogihara† and Masaharu Adachi‡

†Graduate School of Engineering, Tokyo Denki University
5 Senju-Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
‡School of Engineering, Tokyo Denki University
5 Senju-Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan
Email: 14kme08@ms.dendai.ac.jp, adachi@eee.dendai.ac.jp

Abstract—In recent years, brain activity measurement by near-infrared spectroscopy (NIRS) has been applied to brain–machine interfaces (BMIs). Classification of brain activity based on measurement data is a fundamental step in the development of BMIs. It has been reported that classification of NIRS data by support vector machines is promising. In this paper, we introduce the extreme learning machine (ELM) for the classification of brain activity measurement data by NIRS. As a result, ELM improves classification accuracy and reduces calculation times in comparison with conventional methods.

1. Introduction

In recent years, brain activity measurement technology has been applied to brain-machine interfaces (BMI), which are interfaces that connect the brain and machinery. BMI using near-infrared spectroscopy (NIRS) in particular has attracted considerable attention. Successful development of BMI should lead to quality-of-life improvements for persons with physical disabilities.

There are two broad types of brain activity measurement methods: those that measure the neural activity of the brain itself and those that measure changes in brain blood flow associated with neural activity. Typical examples of the former include electroencephalography (EEG), and the latter includes NIRS and functional magnetic resonance imaging (fMRI). EEG has the merit of high temporal resolution, but it is limited by low spatial resolution and small signal strength. fMRI has the drawbacks of low temporal resolution and high cost. In contrast, NIRS has a higher spatial resolution than EEG and a higher temporal resolution than fMRI, so in this paper we use NIRS for a BMI. NIRS is a method for sensing brain activity by measuring hemoglobin in brain blood flow by using near-infrared light. Classification of brain activity using NIRS has been performed with various classifiers [1][2], but it has been found that in many cases classification by this method is not satisfactory.

This paper introduces classification abilities of the extreme learning machine (ELM) [3]. We adopt a support vector machine (SVM) [4] as the conventional method for comparison.

2. Experiment

We conducted experiments with three right-handed volunteers. A probe was placed over the primary motor cortex, based on the 10–20 international system (Fig. 1). Each volunteer performed five trials consisting of a before-task rest, a task, and an after-task rest, each lasting 20 s. The task content was left-hand grasping, which participants were instructed to perform as quickly as possible. Learning data was obtained from four trials and test data from one trial. A low-pass filter with a 0.5 Hz cut-off frequency was applied to the NIRS data. Additionally, the NIRS data were normalized to be within a range of 0 to 1.

The study was approved by the ethics committee of Tokyo Denki University and was conducted in accordance with the current version of the Helsinki declaration. All participants gave informed consent after the study was explained to them.

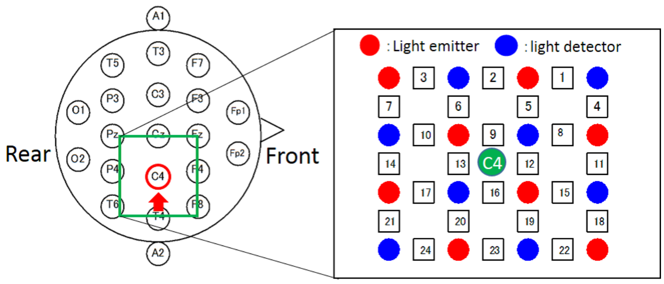

Figure 1: 10–20 international system.

3. Estimation of Response Delay in Cerebral Blood Flow

Previous studies have demonstrated that a time delay of several seconds occurs in the response of brain blood flow effects due to neural activity. It is possible to improve the reliability of analysis results by estimating this delay time because NIRS measures changes in cerebral blood flow due to neuronal activity. Delay-time estimation is performed for 4-channel data (ch. 9, ch. 12, ch. 13, ch. 16) surround-

Table 1: Number of significant windows differences around the start-task time.

Time	significant difference
-2s to 0s and 0s to 2s	13
0s to 2s and 2s to 4s	21
2s to 4s and 4s to 6s	18
4s to 6s and 6s to 8s	14
6s to 8s and 8s to 10s	20
8s to 10s and 10s to 12s	26

Table 2: Number of significant windows differences around the end-task time.

Time	significant difference
-2s to 0s and 0s to 2s	13
0s to 2s and 2s to 4s	20
2s to 4s and 4s to 6s	23
4s to 6s and 6s to 8s	19
6s to 8s and 8s to 10s	18
8s to 10s and 10s to 12s	20

ing C4 in the 10–20 system. In the estimation, the task start time is set to 0 s, and the Mann–Whitney U test is performed in every 2-s window by shifting the window from −2 s to 12 s. Next, the after-task rest start time was set to 0 s, and then the U test was performed as above. Tables 1 and 2 show the number of significant differences for task-start and task-end analysis at a significance level of 1%. These tables show an estimated 2-s delay, because there are a number of significant differences between 0 and 2 s and between 2 and 4 s from the previous window.

4. Classification Algorithm

4.1. ELM

ELM is a learning algorithm for single-hidden-layer feedforward neural networks (SLFNs) proposed by Huang et al. [3]. Figure 2 shows the structure of ELM. The algorithm is as follows. Assume a training set, activation function $g(x)$, and hidden node number I. An input weight w_i and bias b_i are randomly generated, and the hidden layer output matrix $\mathbf{H}$ is calculated as

$$\mathbf{H} = \begin{bmatrix} g(w_1 x_1 + b_1) & \cdots & g(w_I x_1 + b_I) \\ \vdots & \ddots & \vdots \\ g(w_1 x_N + b_1) & \cdots & g(w_I x_N + b_I) \end{bmatrix}, \quad (1)$$

where

$$g(a) = \frac{1}{1 + e^{-a}}. \quad (2)$$

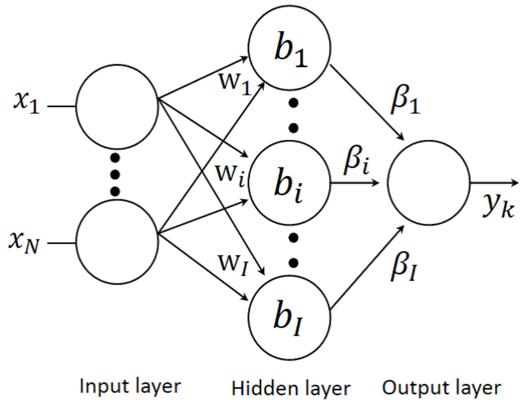

Figure 2: Structure of ELM.

Next, calculate the output weight β as

$$\beta = \mathbf{H}^{+} \times T, \quad (3)$$

where T is the training data set and $\mathbf{H}^{+}$ denotes the pseudo-inverse matrix of $\mathbf{H}$. Finally, we calculate the output value y as

$$y = \mathbf{H}\beta. \quad (4)$$

4.2. SVM

SVM is a classifier proposed by Vapnik et al. [4]. A kernel trick can be easily applied to SVM. The decision function of SVM is

$$D(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b, \quad (5)$$

where

$$b = y_j - \sum_{i \in S} \alpha_i y_i K(x_i, x_j). \quad (6)$$

Here, α, y and $K(x_i, x)$ are a Lagrange multiplier, output training data, and a kernel function, respectively[5]. In this, α is obtained by solving the following optimization problem:

$$\max \quad Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \quad (7)$$

$$\text{s.t} \quad \sum_{i=1}^{M} y_i \alpha_i = 0, \quad (0 < \alpha_i < C) \ . \quad (8)$$

4.3. SVR

SVR is a modified SVM algorithm adapted for application to regression problems [5][6][7]. The decision function of SVR is

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^{*}) K(x_i, x) + b, \quad (9)$$

where

$$b = y_i - w^\top \phi(x_i) - \varepsilon \quad (0 < \alpha_i < C), \tag{10}$$

$$b = y_i - w^\top \phi(x_i) + \varepsilon \quad (0 < \alpha_i^* < C). \tag{11}$$

Here, α_i^*, w, ϕ, and ε are a Lagrange multiplier, a coefficient vector, a mapping function, and the width of a ε tube, respectively. The ε tube is used to reduce error close to the regression curve, with the error defined as

$$E(r) = \begin{cases} 0 & (|r| - \epsilon \le 0), \\ |r| - \epsilon & \text{otherwise}, \end{cases} \tag{12}$$

where r is a residual and α and α_i^* are obtained by solving the following optimization problem:

$$\begin{aligned} min \quad Q(\alpha, \alpha^*) \quad = \quad & \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(x_i, x_j) \\ & + \varepsilon \sum_{i=1}^{m} (\alpha_i + \alpha_i^*) - \sum_{i=1}^{m} y_i (\alpha_i - \alpha_i^*) \end{aligned} \tag{13}$$

$$s.t \quad \begin{cases} \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0, \\ 0 \le \alpha_i \le C, 0 \le \alpha_i^* \end{cases} . \tag{14}$$

5. Results

5.1. Time-series Prediction

The conventional evaluation technique for parameter setting of classification algorithms is based on only the success rate of the classification. In this paper, we use a time-series prediction for evaluation of parameter settings. We performed time-series prediction using three algorithms: ELM, SVR, and back propagation (BP) [8]. Table 3 shows the parameter values used for each algorithm. The parameter values in Table 3 were obtained by trials and errors for each models.Table 4 shows the root-mean-squared error (RMSE) and computation time for the training. Figures 3–5 show the prediction results. ELM has the best performance among the three algorithms for NIRS data prediction, and the best number of hidden-layer neurons for ELM is 27.

5.2. Classification

We used ELM and SVM for classification algorithms. Evaluation of classification ability was performed using 5-fold cross-validation. Table 5 shows parameter values used in each algorithm. Table 6 shows the classification results and computation time for the training.

In the results, the classification rate for ELM was 85.88% and the computation time was 0.023 s. Using SVM, the classification rate was 85.48% and the computation time was 1.487 s. Table 4 thus indicates that ELM performs better than SVM.

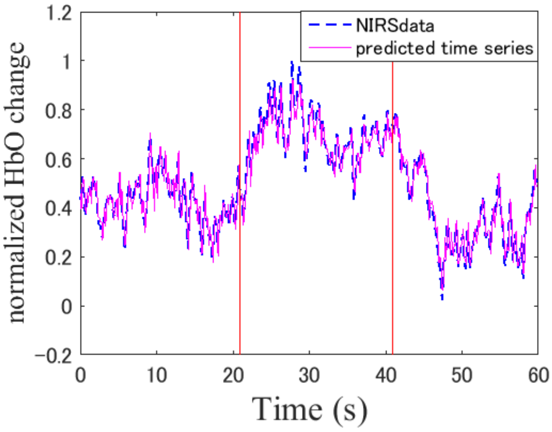

Figure 3: NIRS data (solid line) and predicted time series (dashed line) by ELM.

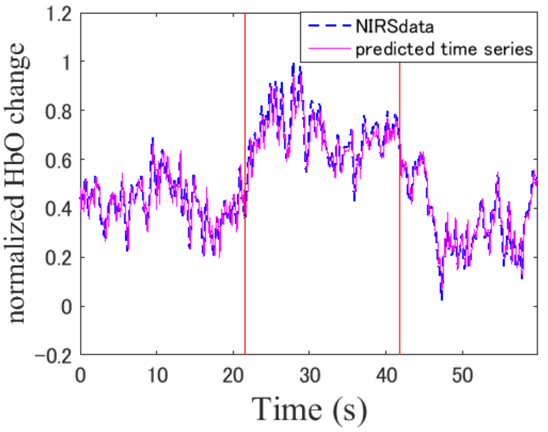

Figure 4: NIRS data (solid line) and predicted time series (dashed line) by BP.

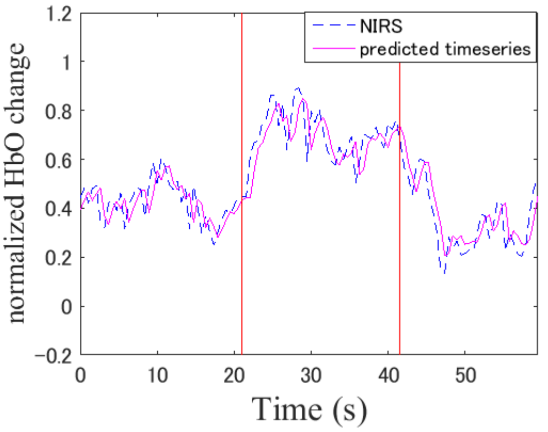

Figure 5: NIRS data (solid line) and predicted time series (dashed line) by SVR.

Table 3: Parameter values used in the three prediction techniques.

	ELM	BP	SVR
Input dimension	9	6	9
Lag	1	1	1
# of training data	1846	1846	461
# of Hidden layer neuron	27	10	(N.A.)
Margin parameter: C	(N.A.)	(N.A.)	26000
width of ε tube: ε	(N.A.)	(N.A.)	9×10^{-7}

Table 4: RMSE and computation time for predictions.

Algorithm	RMSE	Computation time (s)
ELM	0.0425	0.35
BP	0.0435	232.71
SVR	0.0833	11651.22

6. Conclusion

We introduced ELM for time-series prediction and classification of NIRS data. In the results for time-series prediction, we found that ELM is highly suitable for prediction. The results for classification too indicated that ELM has better performance than conventional SVM. We have thus shown that the ELM is an efficient classifier for NIRS data.

In future research we will verify the classification rate with more participants to see whether the proposed classifier is efficient with NIRS signals for other tasks.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number 25420385.

Table 5: Parameter values used in the two classifiers.

	ELM	SVM
Input dimension	9	9
Lag	1	1
# of training data	1846	1846
# of Hidden layer neuron	27	(N.A.)
Margin parameter: C	(N.A.)	300
kernel function	(N.A.)	RBF

Table 6: Classification rates and computation times.

Algorithm	Classification rate	Computation time (s)
ELM	85.88	0.02328
SVM	85.48	1.487

References

[1] B. Xu, Y. Fu, L. Miao, Z.Wang, H. Li, "Classification of fNIRS Data Using Wavelets and Support Vector Machine during Speed and Force Imagination, " Proceedings of the IEEE International Conference on Robotics and Biomimetics, 1224-1229, 2011.

[2] K. Yanagisawa, H. Tsunashima, "Development of Portable Brain-Computer Interface Using NIRS, " UKACC International Conference on Control, 702-707, 2014.

[3] G.B. Huang, Q.Y. Zhu and C.K. Siew, "Extreme Learning Machine: Theory and Applications, " *Neurocomputing*, 70, 489-501, 2006.

[4] V.N. Vapnik, "Statistical Learning Theory, " John Wiley & Sons, 1998.

[5] S. Abe, "Support Vector Machines for Pattern Classification, " Springer-Verlag London, 2010.

[6] H. Ogihara, M. Adachi, "Time Series Prediction of BOLD sigunal Using SVR, "Tech. Report of IEICE, 114(250), 151-155, 2014. (in Japanese)

[7] V.N. Vapnik, S.E. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," in Advances in Neural Information Processing Systems, 9, 281-287, 1997.

[8] D. E. Rumelhart, G.E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors", Nature, 323, 533-536, 1986.