EEG signal processing methods for BCI applications

Ignas Martisius, *Kaunas University of Technology*
(**01.09.2011**, prof. Robertas Damasevicius, *Kaunas University of Technology*)

Abstract

Brain-computer interface (BCI) is a communication system that translates brain activity into commands for a computer or other digital device.

The majority of BCI systems work by reading and interpreting cortically-evoked electro-potentials ("brain waves") via an electroencephalogram (EEG) data. The EEG data is inherently complex. The signals are non-linear, non-stationary and therefore difficult to analyze. After acquisition, pre-processing, feature extraction and dimensionality reduction is performed, after witch machine learning algorithms can be applied to classify the signals into classes, where each class corresponds to a specific intention of the user. BCI systems require correct classification of signals interpreted from the brain for useful operation.

This paper reviews our proposed methods for EEG signal processing and classification, which include Wave Atom transform, use of nonlinear operators, class-adaptive denoising using Shrinkage Functions and real time training of Voted Perceptron artificial neural networks.

1. Introduction

BCI technology is a radically new communication option for those with neuromuscular impairments that prevent them from using conventional communication methods. BCI's provide these users with communication channels that do not depend on peripheral nerves and muscles. Other applications for BCI systems include multimedia communication, augmented reality applications and game development.

BCI systems use EEG data received from electrodes placed onto the head of the subject, which record the electrical activity of neurons in the brain. The frequencies of these brain waves range from 0.5 to 100 Hz, and their characteristics change dynamically depending on the activity of the human brain [1].

2. Experiment Data

For direct result comparison a standard EEG dataset was used in the experiments. Data set Ia (Tübingen, ‹self-regulation of SCPs›, subject 1) [2] from the BBCI competition datasets (http://bbci.de/competition/) was used. Datasets were taken from a healthy subject. The subject was asked to move a cursor up and down on a computer screen, while his cortical potentials were taken. During the recording, the subject received visual feedback of his slow cortical potentials (SCPs). The dataset consists of 135 trials belonging to class 0 and 133 trials belonging to class 1. Each trial consists of 896 samples from each of 6 channels. The sampling rate of 256 Hz and the recording length is 3.5s.

3. Wave Atom Transform

The efficiency (accuracy and speed) of a BCI system depends upon the feature dimensionality of the EEG signal and the number of mental states required for control. Therefore classifying EEG data requires the reduction of its high-dimensional feature space to identify fewer intrinsic feature dimensions relevant to specific mental states of a subject.

Feature reduction can help improve system learning speed and, in some cases, classification accuracy. We consider Wave Atom Transform (WAT) of the EEG data as a feature reduction method [3].

WAT has been recently proposed by Demanet and Ying [4]. This new transform performs a multi-resolutional analysis of a signal, i.e., decomposes a signal into different frequency sub-bands. Wave atoms are a variant of wavelets that have a sharp frequency localization and offer a sparser expansion for oscillatory functions than wavelets. WAT has been previously used mainly in image processing for denoising, watermarking, hashing, fingerprint recognition, signal analysis, as well as dimensionality reduction and numerical analysis.

WAT is a promising approach for EEG processing because of its denoising and feature extraction capabilities, and is particularly useful when the signal has discontinuities and sharp spikes as is in case of EEG.

In this experiment, data classification was performed using artificial neural networks (ANN) due to their ability to generalize and work well with noisy data. Strictly feed-forward ANNs with one input layer, one output layer and one hidden neuron layer were used, initialized with random values. A tangent sigmoid threshold function was used both in hidden and output layers. A 15-fold cross validation was performed for every ANN hidden layer. Raw (unprocessed) EEG data was used for result comparison.

Various hidden layer sizes were chosen and 3 network training functions were tested:

- Levenberg-Marquardt training function (LM),
- Fletcher-Powell Conjugate Gradient Backpropagation training (CGF)
- Bayesian Regularization training function (BR).

Results of the experiment are presented in Tab.1. Best results are shown in bold.

Tab.1.
Classification accuracy and network training time using WAT

Training function	Features	Neurons	F-measure	Accuracy, %	Time, s
LM	RAW	10	0.78	79	234.4
	WAT	2	0.88	87	**0.49**
CGF	RAW	10	0.87	87	0.93
	WAT	1	0.88	88	0.79
BR	RAW	5	0.84	84	11906
	WAT	2	**0.90**	**90**	1.1

As shown in Tab. 1, WAT coefficients extracted from EEG data samples had retained enough information to permit correct classification, while feature reduction dramatically decreased system training and classification time. Classification using WAT transform was more accurate with all training functions. All classification quality results were in line with the best results obtained in the BBCI competition II. Best accuracy of 90% is achieved using Bayesian Regularization training, however it is the slowest. The use of the Levenberg-Marquardt training reduces ANN training time by half, with a negligible accuracy loss.

This speed improvement would mostly benefit real-time BCI applications.

4. Teager-Kaiser Energy Operator

This section describes a nonlinear operator based on the generalization of the Teager-Kaiser Energy Operator, called Homogeneous Multivariate Polynomial Operator (HMPO).

Recently, the non-linear operators such as the Teager-Kaiser energy operator (TKEO) [5] have attracted the attention of researchers in the BCI domain.

The Teager-Kaiser Energy Operator (TKEO) is a special case of nonlinear models. For a continuous real-valued signal $x(t)$, the $\Psi[x(t)]$ is defined as follows:

$$\Psi[x(t)] = \left(\frac{\partial x}{\partial t}\right)^2 - x(t)\cdot\frac{\partial^2 x}{\partial t^2}. \qquad (1)$$

An approximation of the derivatives by one-sample differences provides the definition of the TKEO for the discrete-time signal

$$\Psi[x(n)] = x^2(n) - x(n-1)x(n+1). \qquad (2)$$

Moore et al. [6] proposed a generalization of the Teager operator as 1-D Volterra filter

$$\Psi^m[x(n)] = [x(n)]^{2/m} - [x(n-1)x(n+1)]^{1/m}. \qquad (3)$$

Tomar et al. [7] introduced two generalizations of TKEO. A variable length TKEO (VTEO) is defined as:

$$\Psi_i[x(n)] = x^2(n) - x(n-1)x(n+1). \qquad (4)$$

The Summed-over Variable length Teager Energy Operator (S-VTEO) is defined as

$$\xi_i[x(n)] = \sum_{k=1}^{\omega} \Psi_k[x(n)]. \qquad (5)$$

A generalization of the continuous TKEO as the higher-order energy operator (HOEO) Ψ_k is prosed in [8]

$$\Psi[x(t)] = \frac{\partial x}{\partial t}\frac{\partial^{k-1} x}{\partial t^{k-1}} - x(t)\cdot\frac{\partial^k x}{\partial t^k}. \qquad (6)$$

For discrete-time series, the HOEO can be rewritten as the discrete energy operator (DEO) [8]:

$$\begin{aligned}\Psi_{km}[x(n)] &= x(n)x(n+k) - \\ &- x(n-m)x(n+k+m).\end{aligned} \qquad (7)$$

The advantage of the TKEO family of operators over the traditional DSP analysis methods such as Fourier Transform or wavelet analysis is the ability of the TKEO to discover high-frequency low-amplitude components in analyzed data. The TKEO unlike conventional energy takes into account the frequency component of the signal as well as the signal amplitude.

In a general case, the TKEO operator can be generalized to the Homogeneous Multivariate Polynomial Operator (HMPO) $\Psi_m^2[x(n)]$ where the 2nd order HMPO is defined as:

$$\Phi_m^2[x(n)] = \sum_{i=-z}^{z}\sum_{j=-z}^{z} A_{ij}\, x(n+i)x(n+j). \qquad (8)$$

where $z[m/2]$ and A is the coefficient matrix.

The properties of the $\Phi_m^k[x(n)]$ operator are as follows:

- *Symmetry*. Reversing the signal in time does not change the resulting value;
- *Robustness*. The operator is robust even if the signal passes through zero, i.e. $x(n)=0$ i.e. there is no division operation;
- *Complexity*. Complexity of the operator is $\Theta(m^k)$

A Support Vector Machine (SVM) was used for classification. To evaluate the precision of classification the metrics of Accuracy, F-measure and Area Under Curve (AUC) were used. Experimental results are presented in Tab.2. Best results are shown in bold.

Tab.2.

Classification accuracy using nonlinear operators

Operator applied	Classification metric		
	Acc	F	AUC
None	0.7800	0.7891	0.9018
TKEO	0.4740	0.5849	0.4670
TKEO-Volterra	0.5931	0.6583	0.5428
VTEO	0.5635	0.6466	0.6500
VTEO-Volterra	0.4813	0.4214	0.1381
DEO	0.5262	0.6063	0.5851
HMPO	**0.8283**	**0.8349**	**0.8450**

Experimental results obtained demonstrate an improvement of the classification results. The proposed operator can be used for developing new EEG signal processing algorithms, which can be used in Brain-Computer Interface applications, e.g., for robot control.

5. Class-Adaptive Denoising

In this section a Class-Adaptive denoising method is used for selecting optimal parameter values of a standard shrinkage function by maximizing the class distance between frequency domain components of the positive and negative data classes [9]. The denoised data was classified using SVM and quality of classification was evaluated using standard metrics (F-measure, Area Under Curve, Accuracy).

There are many types of shrinkage function proposed in the signal denoising domain. For our analysis, we classify shrinkage functions depending upon the dimensionality of their parameter space as: single-parameter, two-parameter, three-parameter and multi-parameter shrinkage functions. Dimensionality of the parameter space is important for the selection of an optimization method to find best parameter values. Below we provide a short description and analysis of some of these functions.

5.1 Single-parameter shrinkage functions

Donoho and Johnston [10] propose hard (Eq. 1) and soft (Eq. 2) shrinkage functions:

$$\hat{y}=\begin{cases}0, & |y|\le\lambda\\ y, & |y|>\lambda\end{cases}, \qquad (9)$$

$$\hat{y}=\begin{cases}0, & |y|\le\lambda\\ y-\lambda, & y>\lambda\\ y+\lambda, & y<-\lambda\end{cases} \qquad (10)$$

where y is the noisy value, $\hat{y}$ is the shrunken value, and λ is universal threshold.

Norouzzadeh and Jampour [11] propose the following shrinkage function:

$$\hat{y}=y-\frac{\lambda^2 y}{y^8+\lambda^2} \qquad (11)$$

Poornachandra and Kumaravel [12] propose a hyper trim shrinkage function:

$$\hat{y}=\begin{cases}0, & |y|\le\lambda\\ \operatorname{sgn}(y)\cdot\sqrt{y^2-\lambda^2}, & |y|>\lambda\end{cases}. \qquad (12)$$

5.2 Two-parameter shrinkage functions

Poornachandra and Kumaravel also propose a hyper shrinkage function:

$$\hat{y}=\begin{cases}0, & |y|\le\lambda\\ \tanh(\rho\cdot y), & |y|>\lambda\end{cases}. \qquad (13)$$

Another two-parameter shrinkage function is proposed by Mrazek et al. [13]:

$$\hat{y}=y\cdot\left(1-2\cdot 10^{-\frac{2y^2}{\lambda_1^2}}+10^{-\frac{2y^2}{\lambda_2^2}}\right). \qquad (14)$$

where ρ, λ_1 and λ_2 are the parameters of the functions.

5.3 Three-parameter shrinkage functions

Yang and Wei [14] propose a generalization of soft, firm and Yasser shrinkage function:

$$\hat{y}=\begin{cases}0, & |y|\le\lambda_L\\ \operatorname{sgn}(y)\cdot\left[\dfrac{|y-\lambda_L|^{\lambda}\cdot\lambda_H}{|\lambda_H-\lambda_L|^{\lambda}}\right], & \lambda_L<|y|\le\lambda_H\\ y, & |y|>\lambda_H\end{cases}, \qquad (15)$$

where γ, λ_L and λ_H are the parameters of the functions.

Atto et al. [15] propose the smooth sigmoid based shrinkage function:

$$\hat{y} = \frac{\mathrm{sgn}(y)\cdot(|y|-t)_+}{1+e^{-\tau(|y|-\lambda)}}, \qquad (16)$$

where t,τ,λ are the parameters of the function.

5.4 Multi-parameter shrinkage functions

Poornachandra and Kumaravel [16] propose a sub-band dependent adaptive shrinkage function that generalizes the hard and soft shrinkage functions:

$$\hat{y} = \begin{cases} \rho\left[\dfrac{1-\lambda_j^{-2\lambda_j y}}{1+\lambda_j^{-2\lambda_j y}}\right], & |y| \geq \lambda_j \\ 0, & |y| < \lambda_j \end{cases}, \qquad (17)$$

where λ_j are parameters for each sub-band j.

Signal denoising by thresholding is based on the observation that a limited numbers of the DSP transform coefficients in the lower bands are sufficient to reconstruct the original signal. The key steps of signal denoising using DSP transforms are the selection of shrinkage function and its parameter(s). The goal of the shrinkage function is to remove noise so that separability of positive class and negative class in a binary classification problem is increased.

Assume that the observed data $X(t) = S(t) + N(t)$ contains the true signal $S(t)$ corrupted with additive noise $N(t)$ in time t. Let $T(\cdot)$ and $T^{-1}(\cdot)$ be the forward and inverse transform operators. Let $H(Y,\Lambda)$ be the denoising operator with a set of parameters $\Lambda = (\lambda_1, \lambda_2 ..., \lambda_k)$. Then the denoising algorithm is defined as follows:

Compute the DSP transform for a noisy signal $X(t)$: $Y = T(X)$;

Perform frequency shrinkage in the frequency domain: $\hat{Y} = H(Y,\Lambda)$;

Compute the inverse DSP transform to obtain a denoised signal $\hat{S}(t)$ as an estimate of $S(t)$: $\hat{S} = T^{-1}(\hat{Y})$.

This can be generalized into a single equation as follows:

$$\hat{S} = T^{-1}(H(T(X),\Lambda)) . \qquad (18)$$

5.5 Proposed class-adaptive shrinkage method

The scheme described in subsection 5.4 might not work well in case where signal $S(t)$ and noise $N(t)$ have many different components as is the case with the EEG data. Also the selection of the shrinkage function and its parameters is problematic due to a large number of shrinkage functions proposed in the literature and large variability in signal data. Therefore, some adaptivity must be introduced when selecting shrinkage function and its parameters. Below, we provide a description of the proposed Class-Adaptive (CA) shrinkage method.

Let P and Q be the positive and negative classes of data. Let $D(X_P, X_Q)$ be a distance function between datasets X_P and X_Q belonging to P and Q, respectively. We improve the denoising algorithm by optimizing shrinkage function parameters for each frequency component f of X_P and X_Q. Fisher distance was used to calculate a distance between data classes,.

The proposed CA shrinkage algorithm is as follows:

Convert the time domain signals to frequency domain signals using a standard DSP transform.

For each frequency f :

1. Maximize distance between frequency components of positive class and negative class with respect to a set of shrinkage function parameters Λ;
2. Retain Λ for maximal distance as $\Lambda_{\max}$.
3. Perform shrinkage of the DSP transform coefficients using $\Lambda_{\max}$.
4. Convert the shrunken frequency domain signal to the time domain using an inverse DSP transform.

The dataset was randomly partitioned into 5 parts, and 5-fold cross-validation was used to evaluate the classification results.

On each channel data, a FFT was applied and the shrinkage function parameters were optimized to obtain maximal distance between positive and negative classes. To achieve maximal separation between positive and negative classes, distance metric values are maximized using Nelder-Mead (downhill simplex) optimization method. The optimized shrinkage function is then used for EEG data denoising.

Tab.3.
Classification accuracy using Shrinkage functions operators

CA shrinkage function	Parameter No.	F-measure	AUC	Avg. Prec.

Not applied	-	80.36	90.06	91.45
Hard	1	83.84	91.53	92.60
Soft	1	78.18	88.60	89.97
Norouzzadeh	1	79.16	87.76	90.07
Hyperbolic	1	81.42	88.60	91.26
Hyper	2	85.37	90.87	91.44
Mrazek	2	74.65	87.23	89.71
Yang	**3**	**88.79**	**94.45**	**94.64**
Atto	**3**	**88.16**	**94.38**	**94.67**

The experimental results show that CA denoising can improve the classification results as compared with the case were no signal denoising is used. Best denoising is achieved using three-parameter shrinkage functions with their parameter values optimized for each frequency component of the frequency domain representation of the EEG signal, while soft denoising has failed due to large bias of the denoised signal.

6. Real-time training of Voted Perceptron

BCI applications have strict time constraints for signal processing. Therefore, BCI systems and parts thereof must be considered as real-time systems subject to the requirements for correctness and guaranteed response. The classifier is the most computationally expensive part of these systems. Most of the modern classification methods produce good results on the benchmark EEG datasets; however their training and classification times are unacceptable for real-time applications.

To make the BCI response fast, the classifier should output control signals to external electronic devices at least every 0.5 s, though up to 10 s of data may be used for classifier training. A Voted Perceptron is an algorithm for linear classification, which combines the Rosenblatt's perceptron algorithm with Helmbold and Warmuth's leave-one-out method [17]. All weight vectors encountered during the learning process vote on a prediction. A measure of correctness of a weight vector, based upon the number of successive trials in which it correctly classified instances, can be used as the number of votes given to the weight vector. The output of the Voted Perceptron is calculated as follows:

$$y_i = \operatorname{sgn}\left\{\sum_{n=0}^{N} c_n \operatorname{sgn}\left(w_n x_{i,n}\right)\right\} \qquad (19)$$

where $x_{i,n}$ are inputs, w_n are weights, y_i is the predicted class label.

The result of training is a collection of linear separators $w_1, w_2 \dots, w_n$ along with the w_n survival time c_n , which is a measure of the reliability of w_n .

The shortcoming of the Voted Perceptron is that its training time is usually unbounded and depends on the size and complexity of training data. If the data is linearly separable, the number of iterations is finite. Otherwise, the algorithm will loop infinitely; therefore, a maximum number of iterations must be specified. To make Voted Perceptron suitable for real-time BCI applications, we propose the following modification of the training algorithm [18]: the algorithm measures the time elapsed from the beginning of the training and cuts the training procedure as soon as the time bound is reached. Fig. 1 shows the quality of classification evaluated using the Kappa statistic (inter-annotator agreement)[19] Higher values are better.

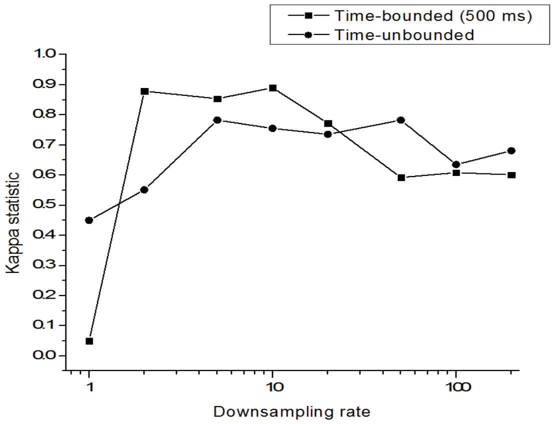

Fig.1. Classification quality (Kappa statistic) after down-sampling

The experimental results show that the size of data can be reduced significantly using signal down-sampling without significant loss of classification quality (measured using Kappa statistic) while satisfying the real-time constraints for NN training.

7. Conclusions and future work

In this paper, we have described several EEG signal processing methods for use in the BCI domain.

The use of Wave Atom transform allows for data size reduction without the loss of important signal information. Data classification was performed using artificial neural networks with various hidden layer sizes and training functions. Results show improved training speed and classification quality.

The use of higher order non-linear operators, such as the proposed Homogeneous Multivariate Polynomial Operator (HMPO) allows for improved classification result with an SVM. This operator can be used to develop new EEG signal processing algorithms.

Class-adaptive denoising uses Fisher distance metric to evaluate distances between frequency components belonging to positive and negative dataset classes. To achieve maximal separation between positive and negative classes, distance metric values are maximized using Nelder-Mead (downhill simplex) optimization method. The optimized shrinkage function is used for EEG data denoising. The denoised data is classified using an SVM with linear kernel. Experimental results show that CA denoising can improve classification results.

Real-Time Voted Perceptron is a time-aware training algorithm for the Voted Perceptron. To cope with large amounts of the EEG data BCI applications must handle to make real-time training feasible. Experimental results show no significant loss of classification quality with reduction in data size with down-sampling, while satisfying the real-time constraints for NN training.

Future work will focus on the use of these algorithms on self-recorded data for use in a personal portable BCI system rather than publicly available datasets.

Bibliography

[1] Zhang, J., Wang, Y., Wang, R. :*Application of KIII Model to EEG Classification Based on Nonlinear Dynamic Methods*, International Journal of Artificial Intelligence, Vol. 7, No. A11. 2011

[2] Birbaumer, N.,Ghanayim, N., Hinterberger, T., Iversen, I., Kotchoubey, B., Kübler, A., Perelmouter, J., Taub, E., Flor, H.: *A spelling device for the paralysed.* Nature, 398:297–298. 1999

[3] Ignas Martisius, Darius Birvinskas, Robertas Damasevicius, Vacius Jusas: *EEG Dataset Reduction and Classification Using Wave Atom Transform,* 23rd International Conference on Artificial Neural Networks ICANN2013, Sofia, Bulgaria. 2013

[4] L. Demanet, L. Ying: *Wave atoms and sparsity of oscillatory patterns*, Appl. Comput. Harmon. Anal., 23(3), 368387, 2007.

[5] Kaiser J. F: *On a simple algorithm to calculate the 'energy' of a signal*, Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'90), 1990. – Vol. 1. – P. 381–384.

[6] Moore M., Mitra S., Bernstein R.A *Generalization of the Teager Algorithm*, IEEE Workshop on Nonlinear Signal Porcessing. – Ann Arbor, Michigan, 1997.

[7] Tomar V., Patil H. A.: *On the development of variable length Teager energy operator (VTEO)*, 9th Annual Conf. of the Int. Speech Communication Association (ISCA'08), 2008. – P. 1056–1059.

[8] Maragos Potamianos: *A. Higher order differential energy operators*, IEEE Signal Processing Lett., 1995. Vol. 2. – No. 8. – P. 152–154.

[9] I. Martišius, R. Damaševičius, *Class-Adaptive Denoising for EEG Data Classification*, Artificial Intelligence and Soft Computing (ICAISC). Lecture Notes in Computer Science (LNCS), 2012

[10] Donoho, D.L.: Johnston, I.M.: *Ideal spatial adaptive via wavelet shrinkage.* Biometrika, (81), 425–455. (1994)

[11] Norouzzadeh, Y., Jampour, M.: *A novel curvelet thresholding function for additive gaussian noise removal.* Int. Journal of Computer Theory and Engineering, (3-4). (2011)

[12] Poornachandra, S., Kumaravel, N.: *Hyper-trim shrinkage for denoising of ECG signal.* Digital Signal Processing (15), 317–327. (2005)

[13] Mrazek, P., Weickert, J., Steidl, G.: *Diffusion-inspired shrinkage functions and stability results for wavelet denoising.* Int. J. Comput. Vision 64, (2-3), 171-186. (2005)

[14] Yang, Y., Wei, Y.: *New Threshold and Shrinkage Function for ECG Signal Denoising Based on Wavelet Transform.* Proc. of 3rd Int. Conf. on Bioinformatics and Biomedical Engineering, ICBBE 2009, 1-4. (2009)

[15] Atto, A.M., Pastor, D., Mercier, G.: *Smooth Sigmoid Wavelet Shrinkage For Non-Parametric Estimation.* IEEE Int. Conf. on Acoustics, Speech, and Signal Processing ICASSP'08, Las Vegas, Nevada, USA, 3265-3268. (2008)

[16] Poornachandra, S., Kumaravel N.: *Subband-adaptive shrinkage for denoising of ECG signals,* EURASIP J. Appl. Signal Process., 42-42 (2006)

[17] Freund, Y., Schapire, R.E., *Large Margin Classification Using the Perceptron Algorithm.* Machine Learning, 1999, 37(3), 277-296.

[18] I. Martišius, K. Šidlauskas, R. Damaševičius. *Real-Time Training of Voted Perceptron for Classification of EEG Data,* International Journal of Artificial Intelligence (IJAI), Vol. 10, Nr. S13, 2013.

[19] Carletta, J: *Assessing agreement on classification tasks: The kappa statistic*, Computational Linguistics, 22(2), 249–254. 1996

Author:

Ignas Martisius
Kaunas University of
Technology
Studentu str. 50
Kaunas, Lithuania
email:ignas.martisius@ktu.lt