EEG Volume Control Using Optimized SVM & RBFN

Deep Tayal
MS Electrical Engineering
Rochester Institute of Technology, NY USA
dt2804@rit.edu

Aaron Pulver
BS Computer Engineering
Rochester Institute of Technology, NY USA
atp1317@rit.edu

***Abstract*- Human computer interfacing technology using Biopotentials has paved a new way in providing services to people with special needs. A lot of research has been done to develop applications based on biosignals with machines. This paper proposes a method to adjust the volume of a computer automatically depending upon the comfort level of the brain by using EEG signals. Band powers were used as features from the EEG raw data. Classification was done using support vector machine (SVM), one-versus-rest and a radial basis function network (RBFN). These techniques were optimized to different data sets using particle swarm optimization (PSO). Experimental results displays that both methods provide high classification accuracy.**

Keywords- Biopotentials, EEG signals, Feature Selection, Machine Learning, Particle Swarm Optimization, Support Vector Machine, Radial Basis Function Network

I. INTRODUCTION

Understanding the operation of the brain is one the biggest mysteries in life and among the utmost challenges to science. Physically, the human brain consists of about three pounds of grey matter and contains around 10 billion neurons. This seemingly unimpressive organ is responsible for human awareness, emotions, and sensations. Using only thoughts to control the surrounding environment will provide a paralyzed person with a powerful tool which will not rely on peripheral nerves or on emotions. By cautiously studying brain activation patterns, functional maps of the brain have been developed in which the surface of the cortex has been segregated into a set of functionally distinct regions. These different regions are now been studied to determine the working of the brain [1].

With the improvement in biomedical technology and machine learning, the analysis of human bio-potential signals have become an important area of research. Among the human bio-potentials, Electroencephalogram (EEG) signals have been widely studied because of their medical importance [2] [3]. EEG signals are electrical action potential signals obtained as a result neurons firing in the brain.

A typical EEG signal, measured from the scalp, will have an amplitude of 10 µV to 100 µV and a frequency in the range of 1 Hz to 100 Hz. The approach is based on earlier observations that the EEG spectrum contains some characteristics waveforms that fall primarily within four frequency bands: delta (<4 Hz), theta (4-8 Hz), alpha (8-13 Hz) and beta (13-30 Hz).

Difficult characteristics of EEG signals such as poor signal to noise ratio (SNR) compared to other bio-potentials give rise to the employment of robust algorithms in order to achieve the goal. Hence, employing efficient classification algorithms has been an important goal and highly attractive area in the research community [2] [3] [4].

Analysis of EEG signals is a three stage process. The first stage is preprocessing the EEG data. Preprocessing is the phase of preparing the data in order to remove the undesired components/noise which may be present as outliners in the data set such as eye blink artifacts. It is usually achieved by filtering the data. This stage is followed by feature extraction process. In this, various pattern recognition and signal processing techniques, e.g. PCA, Band Powers or FFT can be employed in order to extract meaningful features that will be used for classification. The third and the final stage, termed as classification, is to employ a machine learning technique in order to make decisions utilizing the features extracted in the second stage. Our study focuses on designing an EEG based automatic volume control with the application of two different classifiers (SVM & RBFN) in accordance with an efficient optimizing technique (PSO).

II. DATA COLLECTION

Data were collected using the BioRadio150 and a standard EEG cap based on the international 10-20 system as shown in "Fig 1." Four electrodes were recorded: T3, C3, C4, and T4. Cz was used as the ground. These electrodes were chosen because they are close to the ear and have been shown to be effective for two-class classification [11]. One subject was used for recording the data. He was instructed to choose a song which was approximately 3.5 minutes long. An iPhone 5s was used as the device controlling the music and volume. The user was instructed to set the volume to 10% and to begin listening to the song. Once the song ended, the recording was stopped. This same process was repeated but the volume was changed to 100%. The data were recorded at a rate of 960 Hz. The first and last five seconds of the song was removed to eliminate crescendos and decrescendos. Approximately 30 minutes of both loud and quiet music was recorded. The data were then broken into one second intervals for training, validation, and testing.

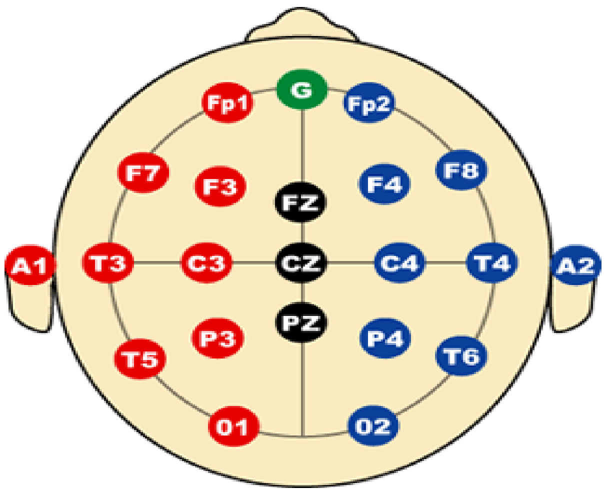


Figure 1. 10-20 EEG System

II. PREPROCEESING

The mean alpha and beta band powers were chosen as the features which were extracted from each electrode. These are two of the most active frequency bands for brain biosignals [4]. Band powers have been proven to be an efficient method for extracting features for a two-class problem [11]. Taking the band powers requires creating a band-pass filter for the desired range of frequencies, squaring the signal, and then taking the mean. This technique provided a vector of eight features [$T3_\alpha$ $T3_\beta$ $C3_\alpha$ $C3_\beta$ $C4_\alpha$ $C4_\beta$ $T4_\alpha$ $T4_\beta$].

III. CLASSIFICATION

Two classification techniques were analysed: Particle Swarm Optimization with Support Vector Machine (PSO-SVM) and Particle Swarm Optimization with a Radial Basis Function Network (PSO-RBFN).

A. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic approach for solving continuous and discrete optimization problems. It belongs to the class of swarm intelligence techniques that are used to solve optimization problems.

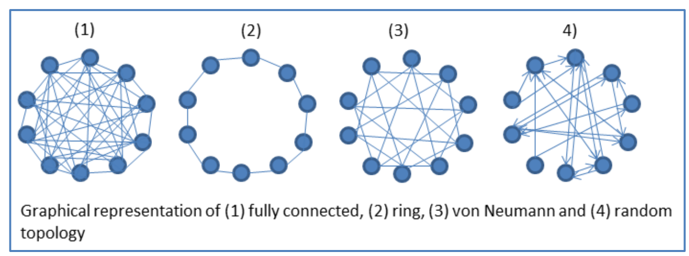
In particle swarm optimization, simple software agents, called particles, move in the search space of an optimization problem. The position of a particle represents a candidate solution to the optimization problem at hand. Each particle searches for better positions in the search space by changing its velocity according to rules originally inspired by behavioral models of bird flocking [5].

The PSO algorithm is population-based: a set of potential solutions develops to approach an appropriate solution (or set of solutions) for a problem. Being an optimization method, the goal is finding the global optimum of a real-valued function (fitness function) defined in a given space (search space).

The social representation that led to this algorithm can be summarized as follows: the individuals that are part of a society hold a view that is part of a "belief space" (the search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors:

- The knowledge about the environment i.e. its fitness value
- The individual's previous knowledge of states
- The previous knowledge of states of the individual's neighborhood/surrounding.

An individual's neighborhood may be defined in several ways, constructing somehow the "social network" of the individual. Some neighborhood topologies exist (full, ring, star, etc.) subject on whether an individual interacts with all, some, or only one of the rest of the population (figure 2).



Graphical representation of (1) fully connected, (2) ring, (3) von Neumann and (4) random topology

Figure 2. Particle Swarm Optimization Topology

In PSO algorithm each individual is termed as a ‘particle’, and is subjected to a random movement in the multidimensional space. Particles have a memory which helps that retain their previous state. Each particle’s movement is the combination of its own (initial) velocity and the two randomly weighted influences, individualism, the tendency to return to its best previous position, and globally, the tendency to move towards the neighborhood’s best previous position [6] [7] [8].

$$v_{id}^{t+1} = w^* \, v_{id}^{t} + c_1{*}\psi_1{*}(\, p_{id}^{t} - x_{id}^{t}\,) + c_2{*}\psi_2{*}(\, p_{gd}^{t} - v_{gd}^{t}\,)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

v_{id}^{t}: Component in d dimension of the i^{th} particle velocity

x_{id}^{t} : Component in d dimension of the i^{th} particle position

c_1, c_2: Individual & Social constant weighted factors

p_i: Best position achieved so long by the particle

p_g: Best position found by the neighbors of the particle

ψ_1, ψ_2: Random factors [0 1]

w: Inertia weight

The PSO used in this paper was tested using a basic two dimensional equation (3). After several iterations, the particles easily converged to X=15 and Y=20. To further test the validity of our PSO, the Rosenbrock equation (4) was used to test for convergence to a global minimum at X=1 and Y=1.

$$f(x, y) = (X - 15)^2 + (Y - 20)^2 \qquad (3)$$

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \qquad (4)$$

B. Support Vector Machine

Support Vector Machine algorithm was originally invented by Vladimir N. Vapnik and the current standard incarnation (soft margin) was proposed by Vapnik and Corinna Cortes in 1995 [9].

SVMs are supervised learning models with associated learning algorithms that analyse data and recognize patterns, used for the purpose of classification and regression analysis. The basic SVM, also termed as non-probabilistic binary linear classifier, takes a set of data and predicts, for each given input, which of the two possible classes forms the output.

The SVM training, taking into consideration the training data marked to a specific class, builds a model that assign new data into one category or other. The SVM model is a representation of the data as points marked in space, mapped so that the data of the separate categories are divided clearly by a wide gap. In other words, it constructs a hyperplane, or a set of hyperplanes which can be used for the purpose of classification or regression.

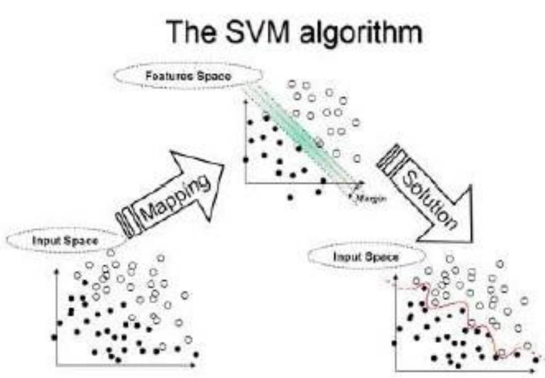


Figure 3: Support Vector Machine Overview

The original optimal hyperplane algorithm proposed by Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik suggested a way to create nonlinear classifiers by applying the kernel trick (originally proposed by Aizerman et al.[9]) to maximum-margin hyperplanes [10]. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be nonlinear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be nonlinear in the original input space.

Support vector machines use a nonlinear kernel function to transform the input features into a higher dimension where the data are more separable [12]. The SVM will create optimal hyper-planes with the largest margin between each class. LIBSVM, a software package was used to create the classifier. Different kernel types were tested: linear, polynomial, radial basis function (RBF), and sigmoid. Support vector machines are a popular newer method of classification and been proven to be very efficient both as a standalone classifier and when combined with other techniques such as fuzzy logic [10] [11]. By changing the cost and gamma of the SVM, accuracy can be fine-tuned. This was implemented by using PSO to evaluate the accuracy with various combinations until the particles converged to a local maximum which produced the best results.

C. Radial Basis Function Network

RBFN is a type of feed-forward Neural Network which consists of three layers: input layer, hidden layer, and output layer as shown in the schematic in Figure 4. The input layer contains n dimensional feature vectors entering the network. The hidden layer is composed of radially symmetric Gaussian kernel functions as shown in “Fig 4.”

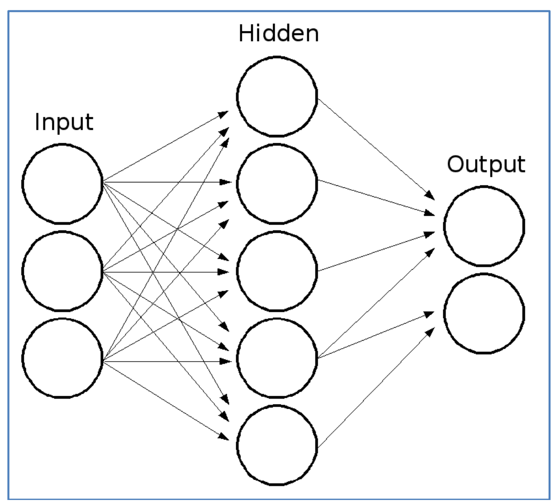


Figure 4. Radial Basis Function Network Overview

In employing RBFN for classification, finding the appropriate centers for kernel functions has critical importance on the generalization capability of the classifier [11].

Therefore, several clustering algorithms are widely used to supervise the cluster centers. In this study, k-means clustering was used to determine the centroids for the hidden layers. The RBFN was modified to have activation functions as shown in (4). Where β is defined in (5) where b is a parameter passed into the RBFN and σ is defined in (6) as the average distance from all the points in a cluster to the center of that cluster.

$$\varphi(x) = e^{-\beta\|x-\mu\|^2} \tag{4}$$

$$\beta = \frac{1}{2\sigma^b} \tag{5}$$

$$\sigma = \frac{1}{n}\sum_{k=1}^{m}\|x_k - \mu\| \tag{6}$$

The outputs of the hidden layer are connected to the output layer by weighted links.These weights were determined by evaluating the activation values for the RBF neurons for each training sample. These values were then used as inputs to gradient descent to train the weight.

During testing with RBFN, outputs are found by using the weights obtained in the training phase and activation values calculated using these test data. These outputs are compared to a threshold value or to each other in the case of two classes at the end in order to generate binary class label outputs. The RBFN classifier accepted a number of kernel functions and the b power as parameters. This allowed PSO to optimize the number of hidden layers as well as the power used to calculate the β coefficients.

IV. RESULTS

After extracting features and using PSO to optimize the parameters to the respective classification methods the best results were recorded in table 1. Standard data sets were used to verify the classification techniques. Then standard BCI data were used to verify preprocessing techniques. Finally, the collected data were preprocessed, optimized, and classified. Cross validation was used for all of these data. This consisted of breaking these data into three sections: learning, validation, and testing. The PSO fitness function tested used the accuracy of both the learning and validation sets to determine the parameters which would give the highest results for these remaining test data. One hundred iterations of PSO were used when optimizing the parameters. The results are comparable to previous work done using the same standard data sets with similar and unique classifiers [4].

TABLE I.
BEST CLASSIFICATION RESULTS.

Best Classification Results		
Data Set	*RBFN Testing %*	*SVM Testing %*
Breast Cancer	99.42 C=3 b=0.4836	98.83 C=9.7038 γ=2.33E-05
Liver Disorder	73.26 C=53 b=4.4452	74.418 C=73.446 γ=1.21E-05
Diabetes Disorder	79.17 C=40 b=2.8317	79.17 C=2.677 γ=5.3E-06
BCI IV 2b	69 C=6 b=0.34641	62 C=64.472 γ=0.7262
Collected Data	91.88 C=41 b=0	92.62 C=13.8162 γ=0.2855

V. CONCLUSIONS AND FUTURE WORK

Raw EEG data pertaining to loud and quiet music were successfully collected, processed, and classified with both PSOSVM and PSORBFN. The results show that an SVM with an RBF kernel is comparable to a RBFN itself. Future work includes recording data from more individuals to better train the system, optimizing the MATLAB code to be more time efficient and robust, and to investigate real-time volume control. Currently, these data were recorded to obtain two classes. In the future, this could be expanded to more classes which would correspond to different desired percentages of volume as well as a content state where the user is satisfied with the current volume.

REFERENCES.

[1] Society for Neuroscience Brain facts: a primer on the brain and nervous system. Society for Neuroscience, 2008.

[2] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche and B. Arnaldi , " A review of classification algorithms for eeg-based brain-computer interfaces," Journal of Neural Engineering, vol. 4, pp. RI--RI3, 2007.

[3] S. Sanei and J. Chambers, EEG signal processing, NJ, USA: Wiley Inter-science, 2007.

[4] Cinar, E.; Sahin, F., "A study of recent classification algorithms and a novel approach for EEG data classification," *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on* , vol., no., pp.3366,3372, 10-13 Oct. 2010

[5] M. Clerc. *Particle Swarm Optimization*. ISTE, London, UK, 2006.

[6] Y. Shi and R.C. Eberhart, “Parameter selection in particle swarm optimization”, In *Proceedings of the Seventh Annual*

Conference on Evolutionary Programming, pages 591-600, 1998.

[7] Y. Shi and R.C. Eberhart, “Empirical study of particle swarm optimization”, In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1945-1950, 1999.

[8] J. Kennedy and R.C. Eberhart, “A discrete binary version of the particle swarm algorithm”, In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, pages 4104-4109, 1997.

[9] Aizerman, Mark A.; Braverman, Emmanuel M.; and Rozonoer, Lev I. (1964). "Theoretical foundations of the potential function method in pattern recognition learning". *Automation and Remote Control* **25**: 821–837.

[10] Boser, Bernhard E.; Guyon, Isabelle M.; and Vapnik, Vladimir N.; *A training algorithm for optimal margin classifiers*. In Haussler, David (editor); 5th Annual ACM Workshop on COLT, pages 144–152, Pittsburgh, PA, 1992. ACM Press

[11] Schwenker, Friedhelm; Kestler, Hans A; Palm, Gunther. “Three learning pahses for radial-basis function networks,” *Neural Networks*, Volume 14, Issues 4-5, May 2001, Pages 439-458.