GENETIC ALGORITHM FOR FEATURE REDUCTION IN BRAIN COMPUTER INTERFACES

Kouhyar Tavakolian  
Computer Science, UNBC  
Prince George, Canada  
kouhyartavakoulian@yahoo.de

Siamak Rezaei  
Computer Science, UNBC  
Prince George, Canada  
siamak@unbc.ca

Caro Lucas  
Elect and Computer Eng. Dept.  
University of Tehran, Tehran  
Iran

Abstract: In this research the practicality of an integer coded Genetic Algorithm (GA) in the classification of mental tasks is investigated. By applying GA to the search space of 30 different EEG extracted features, the optimal 6 feature combinations in the classification of five mental tasks are determined. This would help us to improve accuracy and reliability of Brain Computer Interfaces by choosing better features

Keywords: EEG extracted features, Genetic Algorithm, Mental task classification, Brain-Computer Interfacing (BCI).

1. Introduction

In many Brain-Computer Interface systems developed so far, classification of two to five different mental tasks from EEG signal has served as a basis [1]. This would enable the paraplegic patient to communicate with the computer by giving him/her a new alphabet of classifiable mental tasks. The main purpose of this research is to investigate the usability of Genetic Algorithm in finding optimal features to be extracted from EEG signal.

In our approach we considered reducing the number of EEG extracted features from 30 features to 6. So our search space consists of 593775 points or all 6 combinations of 30 features.

For this research we made use of the EEG dataset gathered by Z.Keirn et al [2]. We selected this dataset because in our past works we used the same dataset in classification of five mental tasks with hybrid feature vectors. We were also inspired by B.Y.Kim et al [4] who have applied GA in the selection of superior EEG extracted features for scoring of sleep stages.

In our previous research we used genetic algorithm as a criterion for finding superior EEG channels in classification of mental tasks [7].

The performance of our system is acquired from the outputs of a neural network classifier hence GA seemed suitable because of its non gradient method of search. GA is basically discrete in nature and need not gradient information. It is also resistant in becoming trapped in local optima. All of our algorithms were developed in Matlab environment.

We introduce our method in part two. In the first section of this part the characteristics of the dataset are explained .In the second section, the EEG extracted features are analyzed. In the third section, our special integer-coded GA is explained, and in the next part, the implementation of the algorithm in reducing the number of EEG features is explained separately.

2. Method

2-1 EEG signal datasets and Mental Tasks:

Recordings were made with reference to electrically linked mastoids A1 and A2 from c3, c4, p3, p4, o1, o2 channels. Electro-Oculogram (EOG) was also recorded between the forehead above the left brow line and another on the left cheekbone. Samples were taken at 250 Hz for 10 seconds, for 2500 samples. Recording was performed with a bank of Grass 7P511 amplifiers whose bandpass analog filters were set at 0.1 to 100 Hz. Subjects 1 and 2 were employees of a university and were left-handed age 48 and right-handed age 39, respectively. Subjects 3 through 7 were right-handed college students between the age of 20 and 30 years old. All were male subjects with the exception of Subject 5. Subjects performed five trials of each task in one day. They returned to do a second five trials on another day. Subjects 2 and 7 completed only one 5-trial session. Subject 5 completed three sessions. The EEG signals were taken during performance of 5 mental tasks:

1-Base line: subjects were told to be relaxed and think of nothing in particular.

2-Multiplication: subjects were given nontrivial multiplication problems and were asked to solve them without vocalizing or making any other physical movements (e.g., 49×78). The problems were not repeated and the numbers selected in such away that an immediate answer was not attainable. Our goal here was to engage the mind with arithmetic calculation and not to get an
answer. This was done by asking the subjects if they had found the answer in the 10s recording.

3-Geometric Figure Rotation: subjects were asked to visualize a previously shown figure and to rotate it about an axis while their eyes were closed.

4- Letter: subjects were instructed to mentally compose a letter to a friend or a relative without vocalizing it.

5- Visual counting: subjects were asked to imagine a blackboard and to visualize numbers being written on the board sequentially.

2-2 EEG extracted features and classifier

In this section a brief explanation of the extracted features from the EEG signals will be presented. In GA for reducing the number of the features all the following 30 features have been included by the same order that follows next.

Frequency features are important because of their vast application and ease of their interpretation. The power spectrum was calculated by an Autoregressive model (6th order) and with the Burg method. Power spectral density in delta (0-4 Hz), theta (4-8 Hz), alpha (8-13 Hz), beta1 (14-22 Hz), beta2 (22-34 Hz) and gamma (36-44 Hz) bands were obtained in a window of length one second (250 points). The average and the maximum of power spectrum for each of the above bands were considered as features. The average of power spectrum in the whole range of (0-44) also the frequency at the maximum power of this range was considered as other features.

The calculated AR coefficients themselves were considered as 6 next features. The order 6 was selected to comply with Anderson and Keirn in their researches [2],[6]. Eight statistical time domain features were also extracted [3]. They included average, maximum and standard deviation of amplitude and also of the absolute amplitude, minimum amplitude, difference between maximum and minimum and the number of zero crossing in each window.

The fractal [3] dimension is a powerful method to study the nonlinear dynamics. These dimensions show geometrical property of the attractors and can be computed very fast. In Higuchi’s algorithm for computing FD, the k new time series are constructed from the time sequence \( x(1), x(2), \ldots, x(N) \) under study:

\[
x_k^m = \left[ x(m), x(m+k), x(m+2k), \ldots, x(m + \left\lfloor \frac{N-m}{k} \right\rfloor k) \right], m = 1, \ldots, k
\]

(2-1)

Where \( m \) indicates the initial time value, \( k \) is the discrete time interval between the points respectively, and \( \lfloor \cdot \rfloor \) defines the integer part of \( a \). For each of the \( k \) time series or curves \( x_k^m \), the length \( L_m(k) \) is computed by:

\[
L_m(k) = \frac{\sum_{i=1}^{N-m/k} |x(m+ik) - x(m+(i-1)k)|}{(N-1)k/N_m}
\]

(2-2)

Where \( N \) the numerical length of the data or the sequence \( x \) and \( (N-1)/N_m \) is the normalization factor. An average length is computed as the mean of the \( k \) lengths \( L_m(k) \) (for \( m = 1, \ldots, k \)). This procedure is repeated for each \( k \) ranging from 1 to \( k_{max} \), obtaining the average length for each \( k \). In the curve of \( \log(L(k)) \) versus \( \log(1/k) \), the slope of the least-squares linear best fit is the estimate of the fractal dimension.

Petrosian’s algorithm uses a quick estimate of the FD. Since waveforms are analog signals, a digital signal is derived by subtracting consecutive samples from the waveform record. From this sequence of subtractions, a binary sequence is created assigning +1 or –1 depending on the result of the subtraction being positive or negative. The FD is then computed as:

\[
D = \frac{\log_{10} n}{\log_{10} (n + \frac{n}{N_\Delta})}
\]

(2-3)

Where \( n \) is the length of the sequence (number of points), and \( N_\Delta \) is the number of sign changes (number of dissimilar pairs) in the generated binary sequence. The above features extraction methods were easily implemented using Matlab. In our algorithm for reducing the number of features a number between one and thirty was assigned to each of the above features.

We used a supervised classifier to classify mental tasks. A feedforward neural network that was trained with the error back propagation algorithm. The momentum and learning rate were updated by an adaptive method. Details of this classifier for each phase of the research will be explained.

2-3 Implementation of Genetic Algorithm

In order to implement our integer coded Genetic Algorithm we considered 6-Gene Chromosomes The value of each Gene corresponds to an EEG feature number between 1 and 30. The following is a sample Chromosome:

\[
2-9-14-16-17-29
\]

For our algorithm for reducing number of features the above chromosome has the meaning of: (2) average of power spectral density in theta range, (9) maximum of power spectral density in alpha range, (14,16 and 17) 2nd, 3rd, and 4th features extracted [3]. They included average, maximum and minimum of power spectral density in the whole range of (0-44) also the frequency at the maximum power spectral density of this range was considered as other features.
4th and 5th AR coefficients and (29) fractal dimension calculated by Petrosian method as explained in the previous section. The algorithm works in 6 steps:

a) **Initial Population:** An Initial population of limited number of chromosomes (N) is selected randomly from the search space.

b) **Fitness Assignment:** A fitness value is computed and assigned to each Chromosome of the population. For this, the EEG extracted features corresponding to Genes of the Chromosome are applied to the inputs of a feed forward Neural Network. The neural network three times performs the classification of mental tasks. The average and variance of results of this three-step classification is considered as the fitness value. So the Chromosome with the higher average and lower variance is considered as the more fitted chromosome. As it is a common characteristic of genetic algorithms this phase is the most time consuming part of the algorithm. In Figure 2-1 the block diagram of the above procedure is shown.

c) **Natural Selection Step:** In this phase of the algorithm two Chromosomes are drawn at random from the population. The chromosome with higher fitness is placed in the mating subset. Then both chromosomes are returned to the population and the above tournament repeats until the filling of mating subset.

d) **Recombination:** Two chromosomes (parents) from mating subset are selected to be mated. Normally the probability that these two Chromosome are mated (Pma) is set to 0.9 or more. If parents are allowed to mate the uniform crossover method is used to produce children. In this method recombination is applied to the individual Genes in chromosomes. The probability of crossover of two Genes (Pc) is set to 0.1 and lower. If crossover is performed the genes between the parents are swapped.

e) **Mutation:** Mutation simply changes the value for a particular gene. The probability (Pmu) of this to happen is usually set to a value about 0.01 or lower.

f) **New Generation:** Now the population is full of newly created Chromosomes and steps b-e will be repeated for a constant number of generations (Ng).

3. Results

In our approach we considered the problem of reducing number of EEG features. Because of the more amounts of data and number of mental tasks, for this phase we used Z. Keirn EEG dataset.

We assigned a number to each of the features and initialized a population of 16 chromosomes of 6 feature combinations of the above features. For subject five and first session of data acquisition we extracted the above features. For this we considered 1 second windows with overlapping of 0.9 s leading to 91 feature vectors for each mental task. Our classifier had just one hidden layer of 20 neuron. This structure was chosen according to our past trial and error which has been explained in [5]. The last layer of the network had five neurons corresponding to five mental tasks. The training was done for 100 epochs and it was repeated three times. The average and standard deviation of these repetitions were considered as fitness value of each chromosome (combination of features).

The Algorithm went on for 13 generations and this took us 3 days and half on a 1.7 GigaHz processor. In the 7th generation there was a sudden decrease of fitness as can be seen in the next figure and this was because of the sudden undesirable mutation. To avoid this from 8th generation we reduced mutation probability from 0.01 to 0.002.

Since in the 13th generation, most of the genes were repeated in almost all chromosomes, the algorithm was interrupted manually. The final results were consistent with our observations and it was inferred that these features were more powerful in this subject from the point of classification:

1) Feature number 2: Average power spectrum in theta band
2) Feature number 9: Maximum power spectrum in alpha band
3) Feature number 13: 1st AR coefficient
4) Feature number 17: 5th AR coefficient
5) Feature number 18: 6th AR coefficient
6) Feature number 30: fractal dimension calculated by Higuchi method

In figure 3-1 the sketch of fitness versus generations can be seen.
As can be seen during generations the averaged fitness has totally increased. The GA parameters: \( Ng=13 \), \( P_{ma}=0.9 \), \( P_c=0.1 \), \( P_{mu}=0.01 \) before 8th generation and \( P_{mu}=0.002 \) after 8th generation onward.

**4. Conclusion**

As we showed in the previous section the fitness which corresponds to classification accuracy has increased during generations. As a result the final population consists mostly of those combinations of EEG features which can better discriminate mental tasks.

In our next development of the algorithm we might let the chromosomes to have variable lengths or to have repeated genes in a same chromosome, which according to our observation in some cases may result in similar but less time consuming classification.

In our future work we may add some more features to the feature pool to compare them with other features. And in general we may think of this method as an intelligent one for choosing EEG features and to increase the total reliability of brain-computer interface systems.

**References**


