Artificial Intelligence in Medicine 55 (2012) 117–126

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Channel selection and classification of electroencephalogram signals: An artificial neural network and genetic algorithm-based approach

Jianhua Yang[a,*], Harsimrat Singh[b], Evor L. Hines[c], Friederike Schlaghecken[d], Daciana D. Iliescu[c], Mark S. Leeson[c], Nigel G. Stocks[c]

[a] School of Biosciences, University of Birmingham, Birmingham B15 2TT, UK
[b] Department of Computer Science, University College London, London WC1E 6BT, UK
[c] School of Engineering, University of Warwick, Coventry CV4 7AL, UK
[d] Department of Psychology, University of Warwick, Coventry CV4 7AL, UK

ARTICLE INFO

Article history:
Received 18 November 2010
Received in revised form 27 January 2012
Accepted 21 February 2012

Keywords:
Genetic algorithm
Artificial neural networks
Least square approximation
Brain–computer-interface
EEG channel selection

ABSTRACT

Objective: An electroencephalogram-based (EEG-based) brain–computer-interface (BCI) provides a new communication channel between the human brain and a computer. Amongst the various available techniques, artificial neural networks (ANNs) are well established in BCI research and have numerous successful applications. However, one of the drawbacks of conventional ANNs is the lack of an explicit input optimization mechanism. In addition, results of ANN learning are usually not easily interpretable. In this paper, we have applied an ANN-based method, the genetic neural mathematic method (GNMM), to two EEG channel selection and classification problems, aiming to address the issues above.

Methods and materials: Pre-processing steps include: least-square (LS) approximation to determine the overall signal increase/decrease rate; locally weighted polynomial regression (Loess) and fast Fourier transform (FFT) to smooth the signals to determine the signal strength and variations. The GNMM method consists of three successive steps: (1) a genetic algorithm-based (GA-based) input selection process; (2) multi-layer perceptron-based (MLP-based) modelling; and (3) rule extraction based upon successful training. The fitness function used in the GA is the training error when an MLP is trained for a limited number of epochs. By averaging the appearance of a particular channel in the winning chromosome over several runs, we were able to minimize the error due to randomness and to obtain an energy distribution around the scalp. In the second step, a threshold was used to select a subset of channels to be fed into an MLP, which performed modelling with a large number of iterations, thus fine-tuning the input/output relationship. Upon successful training, neurons in the input layer are divided into four sub-spaces to produce if-then rules (step 3).

Two datasets were used as case studies to perform three classifications. The first data were electrocorticography (ECoG) recordings that have been used in the BCI competition III. The data belonged to two categories, imagined movements of either a finger or the tongue. The data were recorded using an 8×8 ECoG platinum electrode grid at a sampling rate of 1000 Hz for a total of 378 trials. The second dataset consisted of a 32-channel, 256 Hz EEG recording of 960 trials where participants had to execute a left- or right-hand button-press in response to left- or right-pointing arrow stimuli. The data were used to classify correct/incorrect responses and left/right hand movements.

Results: For the first dataset, 100 samples were reserved for testing, and those remaining were for training and validation with a ratio of 90%:10% using *K*-fold cross-validation. Using the top 10 channels selected by GNMM, we achieved a classification accuracy of 0.80 ± 0.04 for the testing dataset, which compares favourably with results reported in the literature. For the second case, we performed multi-time-windows pre-processing over a single trial. By selecting 6 channels out of 32, we were able to achieve a classification accuracy of about 0.86 for the response correctness classification and 0.82 for the actual responding hand classification, respectively. Furthermore, 139 regression rules were identified after training was completed.

Conclusions: We demonstrate that GNMM is able to perform effective channel selections/reductions, which not only reduces the difficulty of data collection, but also greatly improves the generalization of the classifier. An important step that affects the effectiveness of GNMM is the pre-processing method. In this paper, we also highlight the importance of choosing an appropriate time window position.

* Corresponding author. Tel.: +44 0121 414 5471.
E-mail addresses: J-Yang@ieee.org, jianhua.email@gmail.com (J. Yang).

0933-3657
doi:10.1016/j.artmed.2012.02.001

1. Introduction

An electroencephalogram-based (EEG-based) brain–computer-interface (BCI) provides a new communication channel between the human brain and a computer. Patients who suffer from severe motor impairments (e.g., late stage of amyotrophic lateral sclerosis (ALS), severe cerebral palsy, head trauma and spinal injuries) may use such a BCI system as an alternative form of communication based on mental activity [1]. Most BCIs which are designed for use by humans are based on extracranial EEG. Compared with invasive methods such as electrocorticography (ECoG), this presents a great advantage in that it does not expose the patient to the risks of brain surgery. On the other hand, however, invasive EEG signals contain less noise.

Artificial neural networks (ANNs) as a pattern recognition (PR) technique are well established in BCI research and have numerous successful applications [2–5]. In fact, Lotte et al. [6], presenting an exhaustive review of the algorithms which are already being used for EEG-based BCI, conclude that ANNs are the classifiers which are most frequently used in BCI research. For example, Shuter et al. [2] proposed an ANN-based system to process EEG data in order to monitor the depth of awareness under anaesthesia. They analysed the awareness states of patients undergoing clinical anaesthesia based on the variations in their EEG signals using a three-layer back propagation (BP) network. The network accurately mapped the frequency spectrum into the corresponding awareness states for different patients and different amounts of anaesthetics. In a recently published study, Singh et al. [5] investigated EEG data using a combination of common spatial patterns (CSP) and multi-layer perceptrons (MLPs) to achieve feature extraction and classification. Event-related synchronization/desynchronization (ERS/ERD) maps were also used to investigate the spectral properties of the data. As a result, they achieved an accuracy of 97% using the training data and 86% in the case of the test data. Robert et al. [3] have reviewed more than one hundred EEG-based ANN applications, and divided them into 'prediction' and 'classification' applications. A prediction application aims to predict the side of hand movements based on EEG recordings prior to voluntary right or left hand movements. While in some studies correct prediction rates were low to medium (from 51% to 83%), accuracies as high as 85–90% were achieved in some other cases. In the classification category, ANN-based systems were trained to classify movement intention of the left and right index finger or the foot using EEG autoregressive model parameters. A correct recognition rate of 80% was achieved in some applications. Thus overall, ANN-based BCI systems appear to be a very promising approach.

However, depending on the application, one of the drawbacks of conventional ANNs is that there is no explicit input optimization mechanism. Typically, all available signals or features are typically fed into the network to accomplish the PR task(s). The consequences of this are, as discussed in Yang et al. [7]:

1. Irrelevant signals or features may add extra noise, hence reducing the accuracy of the model.
2. Understanding complex models may be more difficult than understanding simple models that give comparable results.
3. As input dimensionality increases, the computational complexity and memory requirements of the model increase.

This input optimization problem becomes particularly relevant when the ANN input consists of multi-channel EEG signals, which can be very noisy and contaminated by various motion artefacts produced at certain electrodes. Both data acquisition and data processing could be made more efficient if only a relevant subset of possible electrode locations could be selected in advance.

The problem of selecting a minimum subset of channels falls into a broader field of feature selection (FS). In general, FS can be classified into two categories: filter methods and wrapper methods. Indeed, researchers have investigated both approaches to optimize EEG channels [8–11]. For example, Tian et al. [8] proposed a filter approach using mutual information (MI) maximization, where EEG channels were ranked according to the MI between the selected channel and a class label. Channel selection results were then evaluated using classifiers such as a kernel density estimator. They found that the selected EEG channels exhibited high consistency with the expected cortical areas for these mental tasks. Lal et al. [9] introduced a support vector FS method based on recursive feature elimination (RFE) for the problem of EEG data classification. They compared Fisher criterion, zero-norm optimization, and recursive feature elimination methods, and concluded that the number of channels used can be reduced significantly without increasing the classification error. A more recent study by Wei et al. [10] used genetic algorithms (GAs) to select a subset of channels. The selection was then analysed using CSP; Fisher discriminant analysis was used as a classifier to evaluate selection accuracy. They confirmed that classification accuracy can be improved using the optimal channel subsets.

In BCI systems that comprise both channel selectors and classifiers, wrapper-type FS techniques present advantages in that they optimize the channel subsets to be used by the final classifier. From this point of view, optimization techniques such as GAs have great potential in BCI research. Indeed, a GA as a stochastic method outperforms many deterministic optimization techniques in high-dimensional space, especially when the underlying physical relationships are not fully understood [12]. However, although being the most widely used classifier and having many desirable characteristics such as adaptivity and noise-tolerance, to the best of our knowledge, little research has been undertaken to combine ANNs with a wrapper method such as the GA to perform channel selection. This is probably due to the fact that EEG signals are usually sampled at a high frequency, and thus training ANNs with such large numbers of inputs is not feasible; this problem will be addressed in the current study.

In this paper, we present an MLP-based channel selection method for EEG signal classification. MLPs are used both as the final classifier and the fitness function for GAs to select the optimal channel subset. Instead of using full-length or partially filtered signals as inputs to the BCI system, we applied a preprocessing method that only extracts a small number of parameters from each channel. This is to ensure fast off-line data analysis, and to simplify on-line data acquisition. We demonstrate the effectiveness of channel reduction by investigating if-then regression rules extracted from successfully trained MLPs. Furthermore, we applied our methods to two case study datasets.

2. Methods

2.1. Preprocessing

Since a major difficulty in the processing of EEG data comes from the usually very large size of the dataset due to the high sampling frequency, preprocessing becomes important. In the present study, we focus on the preprocessing on a single-channel basis and do not consider methods that work on multiple channels such as common spatial patterns (CSP). Another popular preprocessing method is frequency filtering, where raw EEG signals are filtered using a desired frequency band. However, the problem with this is that the resulting signals may still be too large to be used to train ANNs in a repetitive manner. To significantly reduce the signal size, we consider both the time and the frequency domains.

In the time domain, it is anticipated that under external visual stimulus/experimental tasks, different brain areas will respond differently. Accordingly, signals from different EEG channels will have different overall trends (i.e., generally increasing vs generally decreasing), which will enable us to apply a least square (LS) approximation on a single trial basis. In fact, partial least square (PLS) has been used as a regression method to extract spatiotemporal patterns from EEG signals [13,14]. The LS technique used in the current work is the linear LS approximation of the EEG signal over a specific time period. We let $x_{(t,b)}$ be the EEG signal measurements on channel b at time t. A linear LS approximation for EEG signals on this particular channel for a single trial could then be formed thus:

$$x = mt + n \tag{1}$$

Also, the derivative of Eq. (1) gave:

$$\frac{dx}{dt} = m \tag{2}$$

which was the slope of the linear LS approximation. This value m was indicative of the changes in the signal for each channel during a specific time slot.

To verify that different channels do indeed have different overall trends during a specific time window, we averaged the slope value for different channels of case 1 data (64 channels in total, see Section 3 regarding the experimental datasets). The histogram of these 64 mean values is shown in Fig. 1(a), along with a Gaussian distribution of zero mean and sample variance calculated from the data. The figure clearly shows that the overall trend of the signal slope for different channels is not random. The Wilcoxon signed rank test [15] was applied to verify the statistical significance of the observation, which resulted in a p-value $< 1.33 \times 10^{-5}$.

An example of LS preprocessing is shown in Fig. 1(b), where we have an EEG signal recorded on channel Cz for the first epoch of data case 2 event No. 1 and its LS approximations during the whole trial period (OVR) and different time slots (INT1 to INT7). By performing LS, the data size was greatly reduced while significant information (i.e., signal changing rate and direction over a specific time window) still remained.

In the frequency domain, a commonly used approach, and the one adopted here, is to apply bandpass-filters to obtain the desired frequency. Before the signals are transferred to the frequency domain, we first applied locally weighted polynomial regression (Loess) [16,17]. Loess is effectively a low-pass filter that passes low frequency components and reduces the amplitudes of high frequency ones. This was achieved by fitting a low-degree polynomial to a fixed-width subset of data using weighted least squares. The procedure was repeated for every single data point to produce nonparametric estimates presenting a smoothing effect. In general, smoothers such as Loess have the advantage that filtered data points can be computed rapidly in comparison to fast Fourier transform-based (FFT-based) filters [18]. However, in the present study, Loess smoothed signals were subsequently transferred to the frequency domain using FFT to achieve further data size reduction.

An example of the smoothing effect and FFT is shown in Fig. 1(c) and (d), where Loess was implemented in Matlab using a 2nd degree polynomial and a span of 0.1 (i.e., 10% of all data points). To achieve further data reduction, we extracted the mean and standard deviation (STD) of signals in the frequency domain for frequencies between 0.1 and 80 Hz, which covers the mu (8–13 Hz), beta (14–30 Hz), and 80 Hz rhythms [10,19,20]. These two statistical parameters were indicative of the strength and variations of low frequency signals for each channel. From Fig. 1(d) it is evident that signal amplitudes were greatly reduced above ~20 Hz. However, because of their psychological importance, signals in this range were included in the scope of the analysis.

2.2. The genetic neural mathematic method

Our genetic neural mathematic method (GNMM) has been previously presented [7,21] in the context of optimizing input variables for ANNs and the extraction of regression rules upon successful training. In terms of EEG channel selection and classification, our approach here consisted of three steps (see Fig. 2(a)): (1) GA-based EEG channel selection, (2) MLP-based pattern classification and finally (3) mathematical programming-based rule extraction. Let us now consider each in turn.

2.2.1. Channel selection

We assumed that there were two datasets $\boldsymbol{X} = \{x_{(1,1)}, \ldots, x_{(a,b)}\}$ and $\boldsymbol{Y} = \{y_1, \ldots, y_a\}$, where $\boldsymbol{X}$ contained the EEG measurements, $\boldsymbol{Y}$ was the corresponding classification target, a was the number of trials recorded in the experiments and b denoted EEG channels. The channel selection process can be summarized as follows:

1. An initial population of chromosomes of size N_p was randomly generated. A chromosome consisted of b genes, each representing an input channel. The encoding of a gene was binary, meaning that a particular variable was considered as an input variable (represented by '1') or not (represented by '0'). The assessment of the fitness of a chromosome was the mean squared error (MSE) when a three-layer MLP was being trained with the input variable subset $\boldsymbol{X}_i$ and output target $\boldsymbol{Y}$ for a certain number of epochs N_e using the Levenberg–Marquardt (LM) algorithm.
2. The GA input determination process was then realized by altering N_p, N_e, generation size N_g, crossover probability p_c and mutation probability p_m. As a result, the input channels which occurred most frequently throughout all the populations could therefore be identified. The final subset formed by these channels $\boldsymbol{X}_f$ was the subset that produces the minimal classification error within a given number of epochs.

The interactions between the GA and the MLPs are illustrated in Fig. 2(b) and (c). It should be noted that, in order to minimize any randomness associated with the MLPs and to accelerate training, we employed a weight initialization method. It is common practice to initialize MLP weights and thresholds with small random values. However, this method was ineffective here because of the lack of prior information about the mapping function between the input and the output data samples [22]. There are several approaches [23–25] to estimate optimal values for the initial weights so that the number of training iterations is reduced. GNMM utilized the independent component analysis-based (ICA-based) weight initialization algorithm proposed by Yam et al. [24]. The algorithm was able to initialize the hidden layer weights that extract the salient feature components from the input data. The initial output layer weights were evaluated in such a way that the output neurons were kept inside the active region. Furthermore, when they were being used to evaluate a chromosome, the MLPs were designed to train several times and return the mean results.

In common with many other optimization techniques, GA stopping criteria include convergence and a set of pre-defined parameters such as N_p, N_e. However, as the GA relies on random number generators for creating the population, for selection, for crossover, and for mutation, a different random number seed will usually produce different results. Therefore, we ran the GAs several times until a reasonable solution was found. The following parameters were used to differentiate one GA run from another: N_p, N_e, N_g, p_c, and p_m. Within a single GA run, we also monitored implementation time and any improvement of fitness values over successive generations to determine whether or not to terminate the GA before it reached the final generation. Here, a reasonable

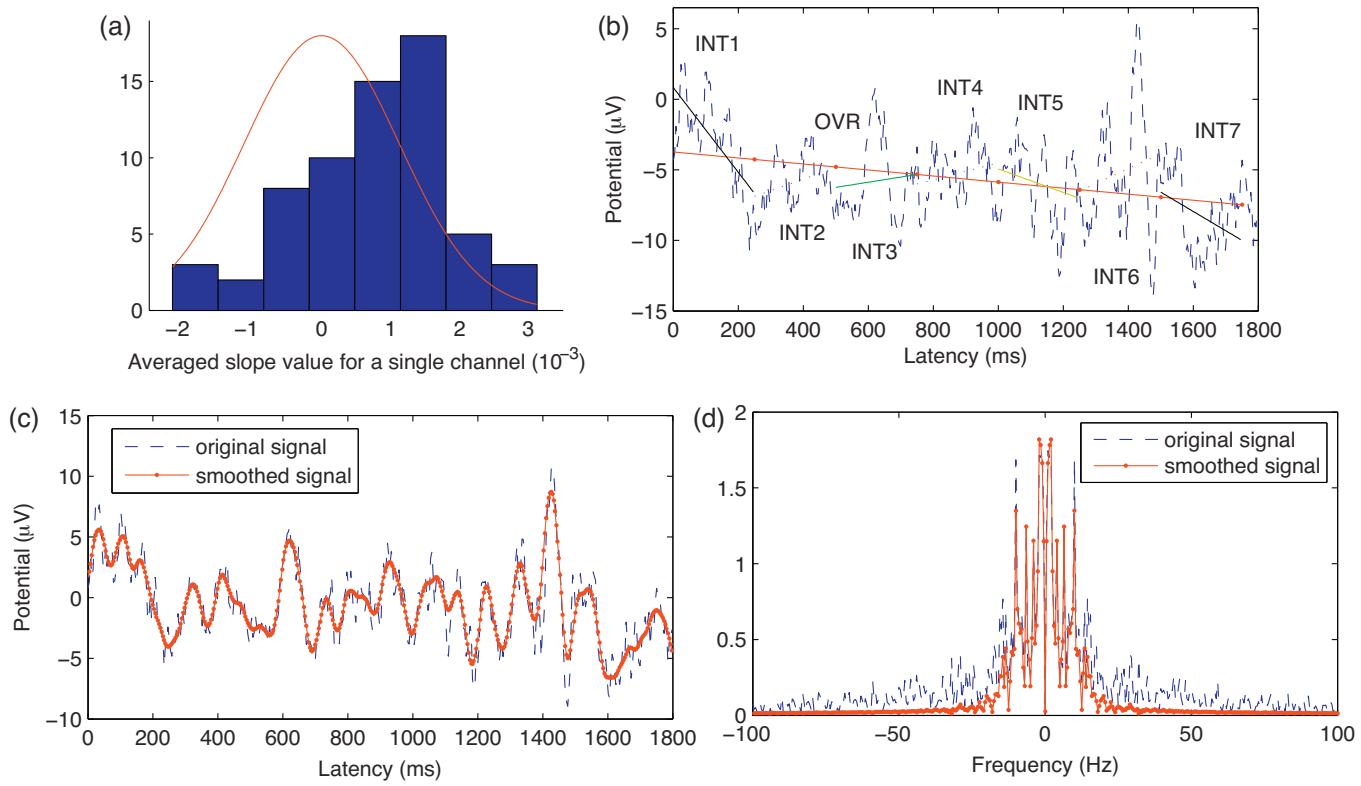

Fig. 1. Preprocessing methods: (a) distribution of channel mean slope for case 1 data (bar chart) and the Gaussian distribution (curved line) with zero mean and sample variance. It is evident that the mean slope is not normally distributed; (b) EEG signal of channel Cz for the first epoch of case 2 event No. 1 and its LS approximations across different time windows; the same signal used in (b) is now presented in both the time (c) and frequency (d) domain. Original signals are in red, whereas the locally weighted polynomial regression (Loess) smoothed signals are in blue. It can be seen that high frequency noise is filtered. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

solution was one in which the difference between channels (group of channels) occurring in the winning chromosome was significant.

GNMM's GA process followed what is known as simple GA (SGA) [26,27], apart from two modifications. One was that GNMM also incorporates an adaptive mutation rate. The algorithm for updating the mutation rate is depicted in Fig. 3. In summary, when the population had higher fitness (i.e., lower MSE), the mutation rate was reduced to encourage exploitation of what has been found. Conversely, when a lower fitness value prevailed, we increased the mutation rate to try to force further exploration. The other modification was the introduction of an elite group into GAs [28]. The elite group was a collection of chromosomes that performed best and were made exempt from crossover and mutation and automatically retained in the next generation. Introducing this elite group into GAs strengthened the ability to search, which could be described as exploitative with respect to high yielding regions and explorative with respect to other regions.

2.2.2. Pattern classification and rule extraction

Taking $\boldsymbol{X}_f$ and $\boldsymbol{Y}$ as inputs and targets respectively, an MLP was used to perform the pattern classification task. As in the previous step, training was performed using a LM algorithm. However, the aim of using an MLP in the current step was to minimize the classification error and thus the number of epochs (iterations) was large, whereas in the previous step MLPs were used as the fitness function needing a relatively small number of epochs.

In this step, GNMM also utilized a K-fold cross-validation technique to define the training and validation data. Each time, a small randomly selected portion of $\boldsymbol{X}_f$ and $\boldsymbol{Y}$ (e.g., $10\% \times a$) was set aside for validation before any training in order to avoid over-fitting [29], and the rest were used for the training. As a consequence of cross-validation, the MLP did not necessarily reach its final epoch N_e.

Apart from channel selection and pattern classification, GNMM also consisted of a rule extraction process. The activation function for all hidden layers was the hyperbolic tangent function, $\tanh(x)$:

$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{2}{1 + e^{-2x}} - 1 \tag{3}$$

and a linear function was used in the output layer. The following equation was used to approximate $\tanh(x)$:

$$g(x) = \begin{cases} 1 & x \geq \kappa \\ \beta_1 x + \beta_2 x^2 & 0 \leq x \leq \kappa \\ \beta_1 x - \beta_2 x^2 & -\kappa \leq x \leq 0 \\ -1 & x \leq -\kappa \end{cases} \tag{4}$$

in which $\beta_1 = 1.002$, $\beta_2 = -0.2510$, $\kappa = 1.9961$. Eq. (4) divided the input domain into four sub-domains. Therefore, once the training was complete, rules associated with the trained MLP could be derived.

For further details of GNMM in addition to the general description above the reader is referred to Yang et al. [7,21].

3. Experimental datasets

For evaluation we applied the GNMM method to two sets of experimental data. This section briefly outlines the experimental setup and preprocessing results.

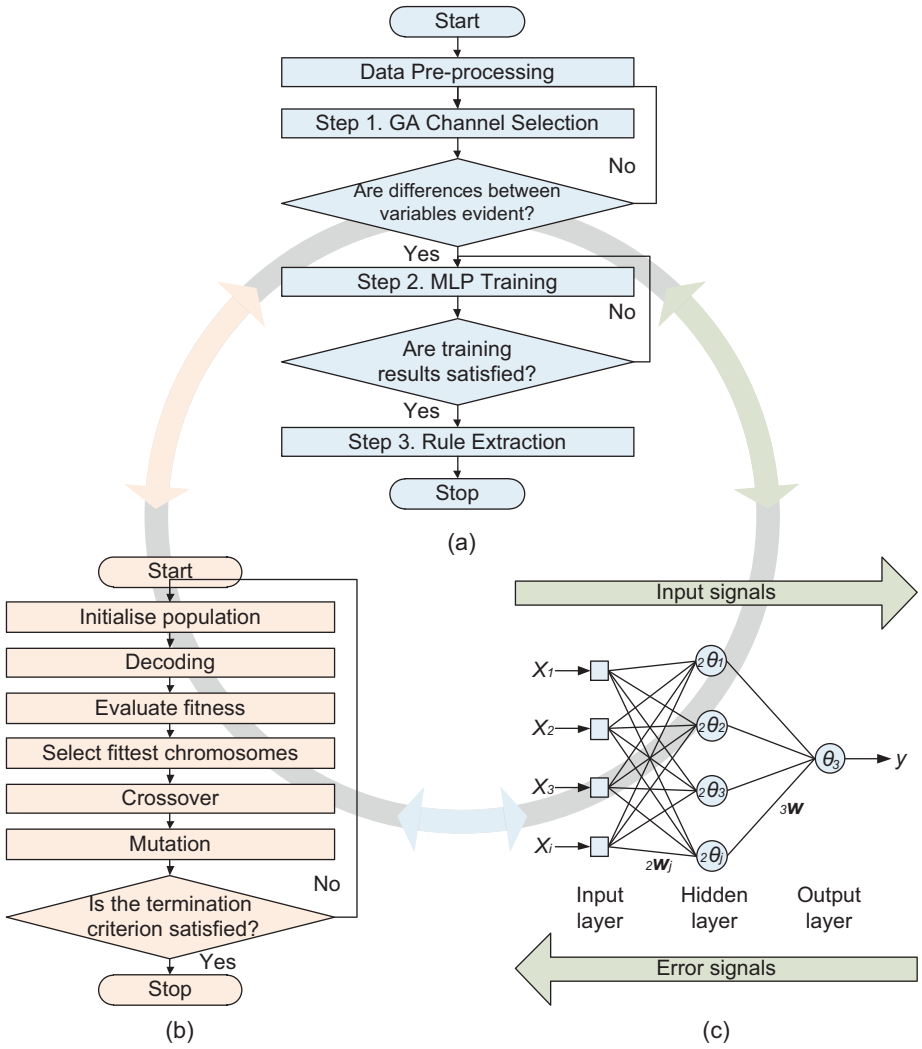

Fig. 2. The GNMM method and interaction between GA and MLP. GNMM consists of three steps (a): GA channel selection (b), MLP training (c), and rule extraction. MLPs are used both in the channel selection and final classification stages.

1 //Initial mutation rate
$p_m = 0.005$
2 //Compute average fitness of first generation
$\hat{f}(1)$
3 //Iterate through the rest of generations
FOR $t = 2{:}N_p$
// Compute average fitness of the tth generation
$\hat{f}(t)$
// Switch depending on whether average fitness increases
IF $\dfrac{\hat{f}(t)}{\hat{f}(t-1)} \leq 0.1$
$p_m = p_m \times 0.1$
ELSE
$p_m = p_m \times \log_{10}\left(\dfrac{\hat{f}(t)}{\hat{f}(t-1)}\right) + 1$
END IF
END FOR

Fig. 3. Adaptive mutation rate. Mutation rate will increase/decrease if the current generation's mean fitness is lower/higher than that of the previous generation.

3.1. Case 1 – two-class motor imagery

The intracranial ECoG recording was explicitly selected to validate the technique developed as it was expected to contain higher quality brain signals with low values of impedances. The dataset[1] (denoted case 1), which was used in the BCI competition III [9,30], comprised a large number of labelled trials which made it advantageous for the evaluation of the performance measures for the technique.

During the experiment, a subject had to perform imagined movements of either the little finger or the tongue (Fig. 4(a)). The ECoG signal was picked up during these trials using an 8×8 ECoG platinum electrode grid which was placed on the contralateral (right) motor cortex (Fig. 4(b)). The grid was assumed to cover the right motor cortex completely, but due to its size (approx. 8 cm × 8 cm) it also partly covered surrounding cortical areas. All recordings were performed with a sampling rate of 1000 Hz. Each trial consisted of either an imagined tongue or an imagined finger movement and was recorded for 3 s duration. To avoid visually

[1] BCI Competition III, dataset I, http://www.bbci.de/competition/iii/desc_I.html, accessed 9th August 2011.

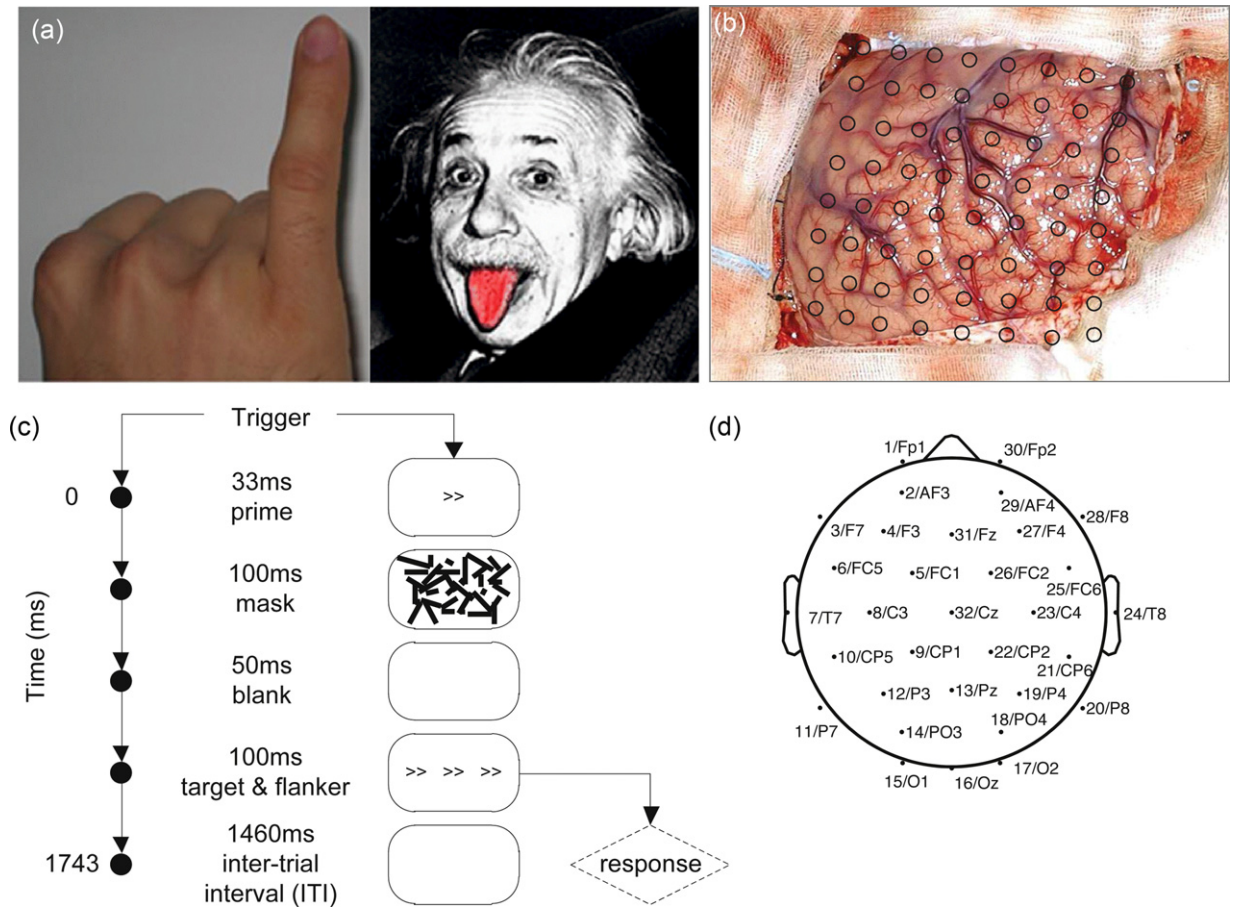

Fig. 4. Experimental setup. Case 1 is a two-class motor imagery experiment, where the subject had to imagine finger or tongue movement according to visual cues (a). An 8×8 ECoG electrode grid was placed on the contralateral (right) motor cortex (b); case 2 is a 2-alternative speeded choice reaction time (RT) task with each trial lasting for 1743 ms (c), where EEG signals were collected from 32 channels (d).

evoked potentials being reflected by the data, the recording intervals started 0.5 s after the visual cue had ended.

The whole data-set consisted of 278 trials for training and 100 trials for testing respectively. Within each trial, there were 3000 data points per channel (i.e., electrode) and a total of 64 channels available. A linear LS approximation was performed on these data on a single trial basis, as well as Loess smoothing and FFT transformation. As a result, the dimension was reduced to 278×192 and 100×192 for the training and testing sets respectively. Target values of 1 and -1 were used for imaginary finger and tongue movement (each class constituting 50% of the whole set).

3.2. Case 2 – response priming paradigm

In a 2-alternative speeded choice reaction time (RT) task, participants had to execute a left-hand or right-hand button-press in response to briefly presented arrow stimuli pointing to the left or right. Each arrow target was preceded by an arrow prime, which could point either in the same or in the opposite direction as the target. These primes were visually 'masked' and therefore easy to ignore (see e.g., Schlaghecken and Eimer [31], for a detailed description of the masked prime procedure). Furthermore, target arrows were flanked by response irrelevant (to-be-ignored) distractor stimuli associated with either the same response as the target or the opposite response, which added a certain level of difficulty to response selection and execution (Eriksen flanker task, e.g., Eriksen and Eriksen [32]). However, for the purpose of the present study, prime- and flanker-related categories were ignored, and the categories to-be-identified were (a) left vs right hand response, and (b) correct vs incorrect response. The experimental structure is shown in Fig. 4(c). The interval from one prime onset to the next was fixed at 1743 ms and the whole experiment consisted of 96 randomized trials per block and 10 blocks per participant. EEG signals were measured using the BioSemi[2] ActiveTwo 32-channel EEG system. The electrode arrangement is shown in Fig. 4(d). The EEG was sampled at a frequency of 256 Hz.

In order to trace the development of response-related EEG signals over time, the trial period was divided into 7 intervals spanning 250 ms each (INT1-INT7, e.g., in Fig. 1(b)). Additionally, analysis was conducted on one overarching time window spanning the whole length of a trial (OVR). Consequently, 8 sets of features are extracted from each EEG channel for each trial. As with the case 1 data, three features are extracted within a single time window: slope for LS approximation, and mean and STD of Loess smoothed signal in the frequency domain. A particular challenge with case 2 is that the number of incorrect responses accounted for only a small fraction of the whole dataset (127/960). This problem, often called biased/unbalanced class distribution, is not unusual in the field of PR [33,34]. To address this issue, a small amount of random noise (<5% of mean value) was added to three duplicates of incorrect response samples, which increased the percentage of incorrect samples to ~40% and the total samples to 1341. (The effect of adding random noise will be discussed in Section 5.) Two training sets were

[2] http://www.biosemi.com/products.htm, accessed 9th August 2011.

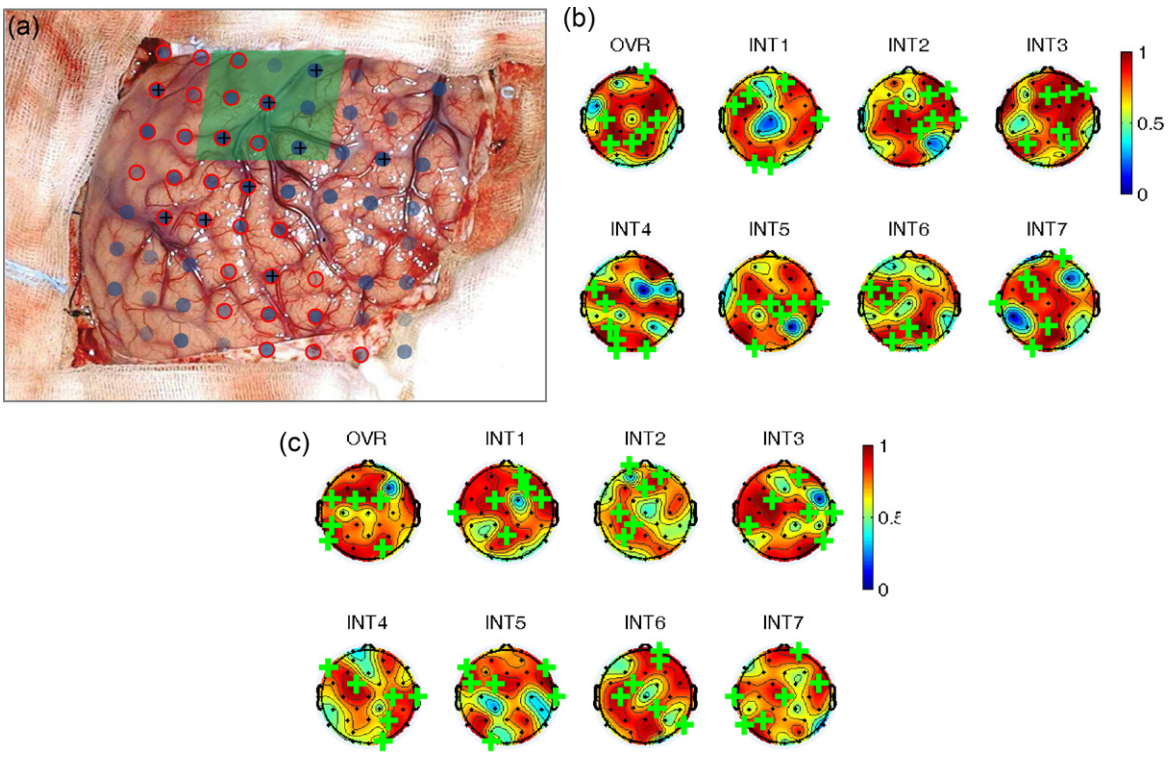

Fig. 5. Appearance percentage (AP) for case 1 (a), case 2A (b), and case 2B (c). In (a), dots are electrodes and the face colour transparency indicates AP – more solid colour means higher value, top 10 ranked channels are marked with '+'; red circles mark the motor cortex as identified by the electric stimulation method. The green parallelogram corresponds to the epileptic focus. (b) and (c) show AP distribution around the scalp for case 2 feature subset OVR, INT1-INT7, where in (b) the training targets are the actual responding hand and in (c) the targets are response correctness. The colour bar indicates chances of a particular channel being selected by the GAs for final classification. '+' marks 6 top-ranking channels used for final classification. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

then formed using different training targets: case 2A contained all 1341 samples with training targets being the actual responding hand (left, 733 samples vs right, 608 samples); case 2B had the same number of samples but training targets were the correctness of response (correct, 833 samples vs incorrect, 508 samples).

4. Results

GNMM was implemented in Matlab R2011a, using the Global Optimization Toolbox and Neural Network Toolbox, and the the FastICA toolbox[3] [35].

4.1. Channel selection

GA researchers often report statistics, such as the best fitness found in a run and the generation at which the individual with that best fitness was discovered, averaged over many different runs of the GA on the same problem [36]. In GNMM, the averaging was extended to not only calculate different runs, but also different generations within the same run. A channel's *appearance percentage* (AP) was defined as the mean appearance of a specific channel in the winning chromosome (minimizing the RMSE) of each population over all GA runs. Thus, this percentage depicted the chance for a channel to be selected by the GAs in the solution that produced the most successful classification results. Investigating AP distribution yielded not only the importance of individual channel in the final pattern classification, but also the energy distribution across the scalp.

Six iterations of the GA were performed for case 1. The resulting AP is shown in Fig. 5(a) with the highest being channel number 22 (95%) and the lowest channel number 13 (14%). It can be seen that, instead of obtaining a single best fit solution as in conventional GAs, AP allowed us to rank available channels according to the probability that they appear in the winning chromosome. Also shown in Fig. 5(a) is the motor cortex of the patient as identified by the electric stimulation method and the epileptic focus. It can be seen that the top 10 channels, which appeared in more than 80% of all the generations, correspond well with the results from the electrostimulation diagnosis. Hence these were specifically selected as the input data for the final classification. The other 54 channels were removed from further analysis.

The GAs were configured to run four times to explore different combinations of input channels for each of those 8 feature sets of case 2. The AP of each channel for different feature subsets of case 2A and 2B are illustrated in Fig. 5(b) and (c) (figure generated using the EEGLAB [37]). Overall, the AP distributions are quite different for different time windows. Furthermore, by examining the top 6 highest ranked channels (marked with '+' in Fig. 5(b) and (c)), we can see that in most cases these channels form two clusters that are not close to each other. This is most likely a consequence of the complexity of the way EEG signals are generated. In addition, in agreement with the phenomenon that is to be classified (manual motor response), the channels located near the hand-area of the left and right motor cortices (Cz, C3 and C4) were the most likely to be selected in case 2A, where the actual responding hand was the classification target; whereas occipital (Oz, O1/O2) and frontopolar (Fp1/2) channels are the most highly ranked in case 2B, where response correctness was the classification target. In order to make comparisons, the first 6 top-ranking channels (~20%) from each feature subsets are selected for the final classification tasks.

[3] Laboratory of Computer and Information Science, the Helsinki University of Technology, http://www.cis.hut.fi/projects/ica/fastica/, accessed 9th August 2011.

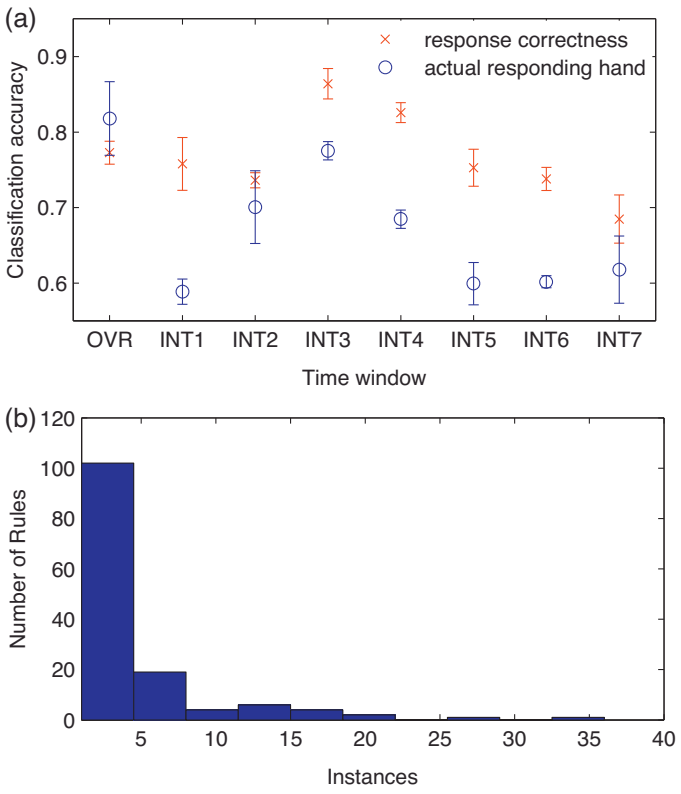

Fig. 6. Classification and rule extraction results for case 2. (a) MLPs were run 5 times for each time window of case 2A and 2B to obtain mean classification accuracy and STD; (b) histogram of number of regression rules extracted from case 2A OVR. There exist two rules fired for more than 60 data samples.

4.2. Classification and rule extraction

For case 1 data, the subset of only 10 channels was fed into a three-layer MLP and trained using the LM algorithm to perform the final classification. The number of neurons in the hidden layer was increased to 10 to maximize the classification rate. As mentioned previously, *K*-fold cross validation was introduced to improve the generalization. The training was performed 10 times, and the mean classification results are shown in Table 1. In comparison, Lal et al. [9] have performed analysis on the same data, using RFE for channel selection and support vector machines (SVMs) for pattern classification. They achieved a classification rate of 0.732 ± 0.080 using 10 best channels and 50 repetitions. The channels selected using their method were different from the ones selected here. However, it can be seen that our results (for the testing subset, i.e., 0.80 ± 0.04) compare favourably with those obtained using RFE and SVM (*t*-test *p*-value = 0.0115).

Also shown in Table 1 are training results using all available 64 channels using the same number of hidden neurons and configurations (training algorithm, cross validation and so on). It can be seen that although similar results for the training subset were obtained, using the 10 best channels we achieved a significant increase for the validation and test subsets, and hence the overall classification accuracy. This implies that the model using fewer channels has a better generalization, due to the fact that noisy and irrelevant channels were removed from the model.

By feeding the channels selected into MLPs and training with the LM algorithm, we were able to compare the classification accuracy between different time windows and training targets for case 2. Fig. 6(a) shows the mean and STD of classification accuracy achieved by running the classifier 5 times. It can be seen that the highest accuracy for the actual responding hand classification (case 2A) was achieved by time windows OVR (~82%); while for correctness classification (case 2B) it is INT3 with a slightly higher rate of ~86%. In general, correctness classifications were easier to achieve than hand classifications: All time windows except INT7 achieved a mean rate of >70%, whereas for the hand classification only OVR and INT3 achieved accuracy of the same level. It should be noted that RT (time from trial onset to the depression of a response button beyond a certain threshold) in this task was approximately 500–550 ms. Therefore, the high classification accuracy in INT3 in both cases reflects the fact that the most distinguishable EEG signals were collected directly after response execution. In addition, better response correctness classification was achieved in the 500–1000 ms time-windows (INT3 and INT4), that is, after an incorrect response had been executed. In line with recent neurophysiological studies [38], this indicates that the most distinguishing feature of response errors lay in the cognitive post-error processes, not in preceding 'erroneous' cognitive processes. Furthermore, classification accuracy gradually decreases as time elapses after response execution, as distinguishable patterns decrease over time.

Rule extraction was not discussed for case 1, as in that case the data were obtained from a single subject with specific channel locations; while in case 2 channel locations have been widely studied and rules can be tested and extended to a wider range of participants. Taking the MLP trained using case 2A OVR (i.e., hand classification, overall time window) for instance, a total of 139 regression rules were extracted from dataset. The histogram of rules extracted from OVR can be seen in Fig. 6(b). Considering that there are 6 channels and 8 hidden neurons, which in theory produces 65,536 (4^8) possible rules, the actual rules implemented are only a small proportion of this number. From this point of view, the data have been narrowed down to the important rules rather than being spread over the rule space.

5. Discussion

5.1. GA parameters

Over the years researchers have been trying to understand the mechanics of GA parameter interactions by using various techniques [39]. However, it still remains an open question as to whether there exists an optimal set of parameters for GAs in general [40]. The interactions among GA parameters do nevertheless follow the generic rules [39,41,42]:

1. GAs with both crossover and mutation operators perform better than GAs based only on crossover or mutation for simple problems.
2. Large mutation steps can be good in the early generations, helping the exploration of the search space, and small mutation steps might be needed in the later generations to help fine-tune the suboptimal chromosomes.

GNMM incorporates these techniques in its structure, such as the adaptive mutation rate as detailed in Fig. 3 and including both selection and mutation operators as in Fig. 2(b).

For parameter values, generally speaking, large population sizes are used to allow thorough exploration of complicated fitness surfaces. Crossover is then the operator of choice to exploit promising regions of fitness space by combining information from promising solutions. Mutation in the less critical genes may result in further exploitation of the current region. Schaffer, Caruana et al. [43] have reported results on optimum parameter settings for SGA. Their approach used the five cost functions in the De Jong's test function suite [28,44]. They found that the best performance resulted

Table 1
Classification results for case 1 data. The model using only 10 channels outperforms the same model using all 64 channels.

	Training	Validation	Testing	Overall
10 best-ranking channels	0.88 ± 0.05	0.80 ± 0.07	0.80 ± 0.04	0.86 ± 0.04
All 64 channels	0.88 ± 0.07	0.72 ± 0.03	0.67 ± 0.05	0.83 ± 0.06

for the following parameter settings: N_p = 20–30, p_c = 0.75–0.95, p_m = 0.005–0.01. Parameter settings for GNMM followed this range except that we increased N_p to 64 for case 1 and 32 for case 2A and 2B respectively. In addition, N_g was set to 100 and 50 for case 1 and case 2 respectively.

It should be noted, however, that selecting the optimal GA parameters is very difficult due to the many possible combinations in the algorithm. In addition, a GA relies on random number generators to create the selection of the population, crossover and mutation. A different random number seed produces different results. This is also the reason why AP is introduced to perform an 'averaging' effect.

5.2. MLP generalization

Generalization refers to the ability of a model to categorize correctly new examples that differ from those used for training [45]. In terms of GNMM, however, because of the randomness associated with MLP/ICA and the fact that training/validation samples may not be representative of the whole data, it is unavoidable that different MLP training sessions produce different results.

To achieve better generalization, we split the data from both case studies into three subsets: the first is the testing subset which is used to evaluate classifier performance, the remainder is then further split into training and validation subsets using *K*-fold cross-validation as described in Section 2.2. Pattern classification and rule extraction. In addition, during MLP training the validation performance was used as the stopping criteria. We have already seen that for case 1 a classification accuracy of 0.80 ± 0.04 was achieved (Table 1). In the case of the 10 MLP runs that produced this result, a mean receiver operating characteristic (ROC) curve with box plot (figure generated using the ROCR [46]) can be seen in Fig. 7. It is evident that although *K*-fold cross-validation was being used, the ROC curves varied for different runs. This was especially true when the true/false positive ratio is high. In Fig. 7 it can be seen that even with successful training results, generalization ability still needed a lot of attention when the classifier is being designed.

The problem of biased/unbalanced class distribution was encountered in case 2. It is already known that in ANN training, if some classes have much fewer samples compared with the other classes, the ANN may respond wrongly for the minority classes because the overwhelming samples in the majority classes dominate the adjustment procedure in training. Various techniques exist for handling this problem [33,34,47]. The approach used in the current study was simply duplicating the under-represented class and adding random noise. By doing this, we increased the proportion of the minority class (i.e., incorrect response), enhanced the MLP tolerance for handling incorrect response data. This consequently improved GNMM's generalization.

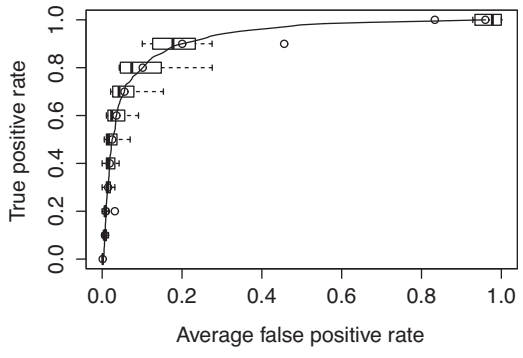

Fig. 7. Mean ROC curve for 10 MLP runs of case 1 data, with box plot indicating variation.

6. Conclusions

In the current paper, we applied the GNMM method to the EEG channel selection and classification problem. Pre-processing steps include: LS approximation to determine the overall signal increase/decrease rate; Loess and FFT to smooth the signals to determine the signal strength and variations. The GNMM method consists of three steps: The first step is to use GAs to optimize input channels so that such channel combinations produce a minimum error, with the fitness function being an MLP for a certain number of epochs. In the second step, EEG channels previously identified are fed into an MLP in order to realize the final pattern classification. During the last stage, regression rules are extracted from trained MLPs so that training results can be easily understood and implemented in other applications, e.g., mobile devices.

As a result, we have presented two case studies and three sets of training data/targets using our data driven technique. The key conclusions that can be drawn are as follows:

1. By applying a GA to optimize channel combinations, the relevance of each channel for a specific task can be evaluated. This is particularly significant in the face of inter-individual differences in functional brain anatomy, which pose a challenge for any EEG-based BCI application, but are particularly relevant in the case of neurological patients suffering from cerebral dysfunctions.
2. Generally, using selected channel subset(s) resulted in a higher classification rate compared to using all the available channels. This is probably because the channels containing irrelevant/noisy data have been removed. More importantly, using a selected subset improves the generalization ability of the model (see also Lal et al. [9]).
3. Using a channel selection technique makes the classifier is easy to understand. In particular, GNMM reduces the number of possible regression rules exponentially if the number of input neurons is reduced.
4. The use of preprocessing has greatly reduced the size of the dataset and improved the effectiveness of GNMM. In the context of the present case studies, it seems that it is appropriate to use a combination of different time windows to achieve a high classification rate for correct and incorrect actual movement. However, establishing the precise number and temporal extent of these time windows for optimal results requires further investigation.
5. In terms of both the topography of the selected channels and the time-course of classification accuracy, the results correspond to the neurophysiology of the processes under investigation, indicating that the present method might be usefully applied not only as a BCI tool, but could also be beneficially applied to basic neuroscientific research as well.

The selection of appropriate channels for EEG pattern classification has been one of the biggest problems for this kind of large datasets. By applying GNMM in two case studies, it is evident that

GA based channel selection provides a potential solution to this problem. However, the computational demands of the GA are very high, currently confining it to offline analysis only. Future research will focus on ways in which improvements can be made to the techniques so that it will be able to quickly and accurately perform channel selection.

Acknowledgements

This work was supported by the Economic and Social Research Council (ESRC RES-000-22-1841), Warwick Postgraduate Research Fellowship (WPRF), UK Overseas Research Students Awards Scheme (ORSAS) and Warwick Institute of Advanced Study (IAS).

References

[1] Guger C, Schlogl A, Walterspacher D, Pfurtscheller G. Design of an EEG-based brain–computer interface (BCI) from standard components running in real-time under Windows. Biomedizinische Technik 1999;44:12–6.
[2] Shuter ML, Hines EL, Williams H, Preece A. Monitoring patient awareness states via neural network interpretation of EEG signals during anaesthesia trials. In: Proceedings of the international conference on neural networks and expert systems in medicine and healthcare. 1994. p. 197–203.
[3] Robert C, Gaudy J-F, Limoge A. Electroencephalogram processing using neural networks. Clinical Neurophysiology 2002;113:694–701.
[4] Robert C, Karasinski P, Arreto CD, Gaudy JF. Monitoring anesthesia using neural networks: a survey. Journal of Clinical Monitoring and Computing 2002;17:259–67.
[5] Singh H, Li XQ, Hines E, Stocks N. Classification and feature extraction strategies for multi channel multi trial BCI data. International Journal of Bioelectromagnetism 2007;9:233–6.
[6] Lotte F, Congedo M, Lecuyer A, Lamarche F, Arnaldi B. A review of classification algorithms for EEG-based brain–computer interfaces. Journal of Neural Engineering 2007;4:1–13.
[7] Yang J, Hines EL, Iliescu DD, Leeson MS. Multi-input optimisation of river flow parameters and rule extraction using genetic-neural technique. In: Hines EL, Leeson MS, Martínez-Ramón M, Pardo M, Llobet E, Iliescu DD, et al., editors. Intelligent systems: techniques and applications. Aachen, Germany: Shaker Publishing; 2008. p. 173–98.
[8] Tian L, Erdogmus D, Adami A, Pavel M, Mathan S, Salient EEG. Channel selection in brain computer interfaces by mutual information maximization. In: Engineering in medicine and biology society, 2005 IEEE-EMBS 27th annual international conference of the Shanghai. 2006. p. 7064–7.
[9] Lal TN, Hinterberger T, Widman G, Schroder M, Hill NJ, Rosenstiel W, et al. Methods towards invasive human brain computer interfaces. Advances in Neural Information Processing Systems 2005:737–44.
[10] Wei Q, Lu Z, Chen K, Ma Y. Channel selection for optimizing feature extraction in an electrocorticogram-based brain–computer interface. Journal of Clinical Neurophysiology 2010;27:321–7.
[11] Schroder M, Lal TN, Hinterberger T, Bogdan M, Hill NJ, Birbaumer N, et al. Channel selection across subjects for brain–computer interfaces. EURASIP Journal on Applied Signal Processing 2005;2005:3103–12.
[12] De Jong K. Genetic algorithms: a 30 year perspective. Perspectives on Adaptation in Natural and Artificial Systems 2005:11.
[13] Kovacevic N, McIntosh AR. Groupwise independent component decomposition of EEG data and partial least square analysis. Neuroimage 2007;35:1103–12.
[14] Martínez-Montes E, Valdés-Sosa PA, Miwakeichi F, Goldman RI, Cohen MS. Concurrent EEG/fMRI analysis by multiway Partial Least Squares. Neuroimage 2004;22:1023–34.
[15] Ott L, Longnecker M. An introduction to statistical methods and data analysis. 6th ed Australia: Brooks/Cole Cengage Learning; 2010.
[16] Cleveland WS. Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association 1979;74:829–36.
[17] Cleveland WS, Devlin SJ. Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association 1988:83.
[18] Edgar JC, Stewart JL, Miller GA. Digital filters in ERP research. In: Handy TC, editor. Event-related potentials: a methods handbook. Cambridge, MA: MIT Press; 2005.
[19] Suk H-I, Lee S-W. Subject and class specific frequency bands selection for multiclass motor imagery classification. International Journal of Imaging Systems and Technology 2011;21:123–30.
[20] Graimann B, Huggins JE, Levine SP, Pfurtscheller G. Visualization of significant ERD/ERS patterns in multichannel EEG and ECoG data. Clinical Neurophysiology 2002;113:43–7.
[21] Yang J, Hines EL, Guymer I, Iliescu DD, Leeson MS, King GP, et al. A genetic algorithm-artificial neural network method for the prediction of longitudinal dispersion coefficient in rivers. In: Porto A, Pazos A, Buño W, editors. Advancing artificial intelligence through biological process applications. Hershey, USA: Idea Group Inc.; 2008. p. 358–74.
[22] Du KL, Swamy MNS. Neural networks in a softcomputing framework. London: Springer; 2006.
[23] Yam JYF, Chow TWS. A weight initialization method for improving training speed in feedforward neural network. Neurocomputing 2000;30:219–32.
[24] Yam Y-F, Leung C-T, Tam PKS, Siu W-C. An independent component analysis based weight initialization method for multilayer perceptrons. Neurocomputing 2002;48:807–18.
[25] Chow TWS, Cho S-Y. Neural networks and computing: learning algorithms and applications. London; Hackensack, NJ: Imperial College Press; Distributed by World Scientific; 2007.
[26] Reeves CR, Rowe JE. Genetic algorithms: principles and perspectives: a guide to GA theory. Boston: Kluwer Academic Publishers; 2003.
[27] Yuen SY, Chow CK. A genetic algorithm that adaptively mutates and never revisits. IEEE Transactions on Evolutionary Computation 2009;13:454–72.
[28] Haupt RL, Haupt SE. Practical genetic algorithms. 2nd ed Hoboken, NJ: John Wiley; 2004.
[29] Lin CT, Lee CSG. Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems. Upper Saddle River, NJ, USA: Prentice-Hall Inc.; 1996.
[30] Blankertz B, Muller K-R, Krusienski DJ, Schalk G, Wolpaw JR, Schlogl A, et al. The BCI competition III: validating alternative approaches to actual BCI problems. IEEE Transactions on Neural Systems and Rehabilitation Engineering 2006;14:153–9.
[31] Schlaghecken F, Eimer M. Active masks and active inhibition: a comment on Lleras and Enns (2004) and on Verleger, Jaskowski, Aydemir, van der Lubbe, and Groen (2004). Journal of Experimental Psychology: General 2006;135:484–94.
[32] Eriksen BA, Eriksen CW. Effects of noise letters upon identification of a target letter in a nonsearch task. Perception and Psychophysics 1974;16:143–9.
[33] Siermala M, Juhola M. Techniques for biased data distributions and variable classification with neural networks applied to otoneurological data. Computer Methods and Programs in Biomedicine 2006;81:128–36.
[34] Mac Namee B, Cunningham P, Byrne S, Corrigan OI. The problem of bias in training data in regression problems in medical decision support. Artificial Intelligence in Medicine 2002;24:51–70.
[35] Hyvarinen A. Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks 1999;10:626–34.
[36] Mitchell M. An introduction to genetic algorithms. Cambridge, MA: MIT Press; 1996.
[37] Delorme A, Makeig S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. Journal of Neuroscience Methods 2004;134:9–21.
[38] Vocat R, Pourtois G, Vuilleumier P. Unavoidable errors: a spatio-temporal analysis of time-course and neural sources of evoked potentials associated with error processing in a speeded task. Neuropsychologia 2008;46:2545–55.
[39] Deb K, Agrawal S. Understanding interactions among genetic algorithm parameters. Foundations of Genetic Algorithms 1999;5:265–86.
[40] De Jong K. Parameter setting in EAs: a 30 year perspective. Parameter Setting in Evolutionary Algorithms 2007:1–18.
[41] Lobo FG, Lima CF, Michalewicz Z. Parameter setting in evolutionary algorithms. Berlin; New York: Springer; 2007.
[42] De Jong KA. Evolutionary computation: a unified approach. Cambridge, MA: MIT Press; 2006.
[43] Schaffer JD, Caruana RA, Eshelman LJ, Das R. A study of control parameters affecting online performance of genetic algorithms for function optimization. In: Proceedings of the third international conference on genetic algorithms. 1989. p. 51–60.
[44] De Jong KA. Analysis of the behavior of a class of genetic adaptive systems. Ph.D. Dissertation. University of Michigan, Ann Arbor. 1975.
[45] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.
[46] Sing T, Sander O, Beerenwinkel N, Lengauer T. ROCR: visualizing classifier performance in R. Bioinformatics 2005;21(20):3940–1.
[47] Fu X, Wang L. Data mining with computational intelligence. New York: Springer; 2005.