Chapter 3

Adaptive Network Fuzzy Inference Systems for Classification in a Brain Computer Interface

Vahid Asadpour, Mohammd Reza Ravanfar and Reza Fazel-Rezai

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/55989

1. Introduction

Fuzzy theory provides the basis for Fuzzy Inference Systems (FIS) which is a useful tool for classifications of data, static and dynamic process modeling and identification, decision making, classification, and control of processes. These characteristics could be used in different kinds of FIS and be applied to Brain Computer Interface (BCI) systems which are discussed in this chapter.

The first kind of FIS is designed based on the ability of fuzzy logic to model human perception. These FIS elaborates fuzzy rules originates from expert knowledge and they are called fuzzy expert system. Expert knowledge was also used prior to FIS to construct expert systems for simulation purposes. These expert systems were based on Boolean algebra and were not well defined to adapt to regressive intrinsic of underlying process phenomena. Despite that, fuzzy logic allows the rules to be gradually introduced into expert simulators due to input in a knowledge based manner. It also depicts the limitations of human knowledge, particularly the ambiguities in formalizing interactions in complex processes. This type of FIS offers high semantic degree and good generalization ability. Unfortunately, the complexity of large systems may lead to high ambiguities and insufficient accuracies which lead to poor performances [1].

Another class of modeling tools is based on adaptive knowledge based learning from data. This category includes supervised learning and outputs of observations when the training data is provided. A numerical performance index can be defined in such simulators which are usually based on the mean square error. Neural networks have become very popular and efficient in this field. Their main characteristic is the numerical accuracy while they also

INTECH
open science | open minds

© 2013 Asadpour et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

provide a qualitative black box behavior. The first self-learning FIS was proposed by Sugeno that provided a way to design the second kind of FIS [2]. In this case, even if the fuzzy rules are expressed in the form of expert rules, a loss of semantic occurs because of direct generation of weights from data. These types of simulators are usually named Adaptive Network Fuzzy Inference System (ANFIS).

There are methods for constructing fuzzy structures by using rule-based inference. These methods extract the rules directly from data and can be considered as rule generation approaches. Generation of the rules includes preliminary rule generation and rule adaptation according to input and output data. Automatic rule generation methods were applied to simple systems with limited number of variables. These simple systems do not need to optimize the rule base. It is different for complex systems. The number of generated rules becomes enormous and the description of rules becomes more complex due to the number of variables. The simulator will be easier to interpret if it is defined by the most influential variables. Also, the system behavior will be more comprehensive when the number of rules becomes smaller. Therefore, variable selection and rule reduction are two important subcategories of the rule generation process which is called structure optimization. A FIS has many more parameters that can also be optimized including membership functions and rule conclusions. A thorough study in these fields and their respective advantages and considerations are provided in the following sections.

In this chapter, several feature extraction and classification methods which could be applied to BCI systems are discussed.

2. Feature extraction

In the following sections, the features which are used to compose feature vectors are discussed. These features provide various views of EEG signal which can be used in ANFIS classification system.

2.1. Energy ratio features

EEG features in a BCI system can be obtained by the frequency analysis of the observed data sequence. For example, in steady state visual evoked potential (SSVEP) BCIs, the frequencies of the light oscillation should be detected. The frequency domain analysis gives a clear picture of changes. Because of frequency changes during BCI, energy ratios between different EEG sub-bands can be computed for each channel during BCI. It is shown that there would be BCI related changes in the EEG according to the brain activities and electrode locations ([3], [4] and [5]). For instance, to discover the brain rhythms during BCI, alpha (8-13 Hz), beta (13-35 Hz), delta (0-4 Hz), and theta (4-8 Hz) band energy ratios of spectrogram SPEC(t, f) at time t and frequency f may be calculated as shown in equations (1) to (4) [6]. They show the total energy of each defined spectral band relative to total signal energy:

http://dx.doi.org/10.5772/55989

$$\alpha = \frac{\int_{8}^{13} \mathrm{SPEC}(t,\ f)\mathrm{df}}{\int_{0}^{35} \mathrm{SPEC}(t,\ f)\mathrm{df}} \tag{1}$$

$$\beta = \frac{\int_{13}^{35} \mathrm{SPEC}(t,\ f)\mathrm{df}}{\int_{0}^{35} \mathrm{SPEC}(t,\ f)\mathrm{df}} \tag{2}$$

$$\delta = \frac{\int_{0}^{4} \mathrm{SPEC}(t,\ f)\mathrm{df}}{\int_{0}^{35} \mathrm{SPEC}(t,\ f)\mathrm{df}} \tag{3}$$

$$\theta = \frac{\int_{4}^{8} \mathrm{SPEC}(t,\ f)\mathrm{df}}{\int_{0}^{35} \mathrm{SPEC}(t,\ f)\mathrm{df}} \tag{4}$$

2.2. Approximate entropy

Approximate entropy is a recently formulated family of parameters and statistics quantifying regularity (orderliness) in serial data [4]. It has been used mainly in the analysis of heart rate variability [7, 8], endocrine hormone release pulsatility [9], estimating regularity in epileptic seizure time series data [10] and estimating the depth of anesthesia [2]. Approximate entropy assigns a non-negative number to a time series, with larger values corresponding to more complexity or irregularity in the data. EEG signal represents regular and uniform pattern during synchronized cooperative function of cortical cells. This pattern results to low entropy values. In contrast, concentric functions and higher levels of brain activity lead to high values of entropy. Shannon entropy H is defined as:

$$H = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{5}$$

in which P_i is the average probability that amplitude of i th frequency band of brain rhythm be greater than r times standard deviation and N is the total number of frequency bands. H is 0 for a single frequency and 1 for uniform frequency distribution over total spectrum. Because of the non-linear characteristics of EEG signals, approximate entropy can be used as a powerful tool in the study of the EEG activity. In principle, the accuracy and confidence of the entropy estimate improve as the number of matches of length r and $m+1$ increases. Although m and r are critical in determining the outcome of approximate entropy, no guidelines exist for optimizing their values. $m=3$ and $r=0.25$ could be selected based on an investigation on original data sequence. Therefore, one dimensions of feature vector could be provided.

2.3. Fractal dimension

Fractal dimension emphasizes the geometric property of basin of attraction. These dimension show geometrical property of attractors and is also computed very fast [15]. Our goal was to associate each 5-second segment data as a trial to its corresponding class. To do this, features were extracted from each 1 second segment with 50% overlap, and sequence of 9 extracted

features were considered as the feature vector of a 5-second segment, which was to be modeled and classified. In Higuchi's algorithm, k new time series are constructed from the signal $x(1),\ x(2),\ x(3),\ \ldots,\ x(N)$ under study is [3]:

$$x_m^k = \left\{ x(m),\ x(m+k),\ x(m+2k),\ \ldots,\ x\left(m + \left[\frac{N-m}{k}\right]k\right) \right\} \tag{6}$$

in which $m = 1,\ 2,\ \ldots, k$ and k indicate the initial time value, and the discrete time interval between points, respectively. For each of the k time series x_m^k the length $L_m(k)$ is computed by:

$$L_m(k) = \frac{\sum_i |x(m+ik) - x(m+(i-1)k)|(N-1)}{\left[\frac{N-m}{k}\right]k} \tag{7}$$

in which N is the total length of the signal x. An average length is computed as the mean of the k lengths $L_m(k)$ (for $m = 1,\ 2,\ \ldots, k$). This procedure is repeated for each k ranging from 1 to k_{max}, obtaining an average length for each k. In the curve of $ln(L(k))$ versus $ln(1/k)$, the slope of the best fitted line to this curve is the estimate of the fractal dimension.

2.4. Lyapunov exponent

Lyapunov exponents are a quantitative measure for distinguishing among the various types of orbits based upon their sensitive dependence on the initial conditions, and are used to determine the stability of any steady-state behavior, including chaotic solutions [4]. The reason why chaotic systems show aperiodic dynamics is that phase space trajectories that have nearly identical initial states will separate from each other at an exponentially increasing rate captured by the so called Lyapunov exponent. The Lyapunov exponents can be estimated from the observed time series [5]. This approach is described as follows: consider two (usually the nearest) neighboring points in phase space at time 0 and at time t, distances of the points in the i*th* direction being $\|\delta X_i(0)\|$ and $\|\delta X_i(t)\|$, respectively. The Lyapunov exponent is then defined by the average growth rate λ i of the initial distance [6]

$$\frac{\|\delta X_i(t)\|}{\|\delta X_i(0)\|} = 2^{\lambda_i}(t \to \infty) \tag{8}$$

or

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \log_2 \frac{\|\delta X_i(t)\|}{\|\delta X_i(0)\|}. \tag{9}$$

Generally, Lyapunov exponents can be extracted from observed signals in two different ways [7]. The first method is based on the idea of following the time evolution of nearby points in the state space [17]. This method provides an estimation of the largest Lyapunov exponent

http://dx.doi.org/10.5772/55989

only. The second method is based on the estimation of local Jacobi matrices and is capable of estimating all the Lyapunov exponents [18]. Vectors of all the Lyapunov exponents for particular systems are often called their Lyapunov spectra. This method was used for Lyapunov vector extraction in this section. An optimized size of 7 is considered for the vector which led to the best mean classification rate on the support vector machine (SVM) classifier.

2.5. Kalman feature extractor

The algorithm to be discussed in this section is based on the Kalman estimation, which is well known in statistical estimation and control theory ([8], [9], and [10]) but perhaps not so in parameter estimation. Therefore, the next paragraphs explain its function in the special context. Kalman filter is essentially a set of mathematical expressions that provides a predictor-modifier estimator. This estimator minimizes the error covariance and therefore is an optimum estimator if appropriate initial condition is selected. The condition for optimum estimation is rarely satisfied however the estimator performs well in sub-optimum situations. Kalman estimator is used for adaptive estimation of dynamic parameters of EEG. The estimator reduces the error variance adaptively and after a period of time a unique estimation is achieved [11].

A Kalman filter computes the response $x \in \mathsf{R}^n$ for the system which is defined by linear differential equation:

$$x_k = A x_{k-1} + B u_k + w_{k-1} \tag{10}$$

in which x is the system state, A and B are the state and input matrices, u is input and w is the process error. $z \in \mathsf{R}^m$ is the measured value and is defined as:

$$z_k = H x_k + v_k \tag{11}$$

in which H is the output matrix and v is the measurement error.

The estimate and process errors are considered independent additive white Gaussian noises. In practice these are time varying processes which are considered stationary for simplicity.

If there is not input or process noise the matrix $A_{n\times n}$ relates the state of the system at last stage to current stage. A is practically a time varying matrix but it is considered constant in computations. $B_{n\times 1}$ relates the control input u to the state x. $H_{m\times n}$ relates the state x to the measurements z_k. H is also time varying but is considered constant in computations.

Consider the system

$$x_{k+1} = (A + \Delta A_k) x_k + B w_k \tag{12}$$

$$y_k = (C + \Delta C_k) x_k + v_k \tag{13}$$

in which $x_k \in \mathsf{R}^n$ is state space, $w_k \in \mathsf{R}^q$ is process noise, $y_k \in \mathsf{R}^m$ is measurements and $v_k \in \mathsf{R}^m$ is measurement noise. Furthermore ΔA_k and ΔC_k represent the variations of parameters. They could be considered as

$$\begin{bmatrix} \Delta A_k \\ \Delta C_k \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} F_k E \tag{14}$$

in which $F_k \in \mathsf{R}^{i \times j}$ is a real time invariant matrix that satisfies the condition

$$F_k^T F_k \leq I, \quad k \geq 0 \,] \tag{15}$$

and H_1, H_2 and E are real matrices that define how A and C elements are affected due to F_k variations. These matrices are estimated using a separate recursive least square estimation [12].

The state-space representation of a linear system is much more flexible and useful than the transfer function form because it includes both time-dependent and time-independent systems and also encompasses stochastic and deterministic systems. Furthermore, it is possible to evaluate precisely concepts of observability and controllability, which are useful in determining whether the desired unknown parameters of a system can be estimated from the given observations for instance.

A modification is performed on state estimation algorithm in this section to overcame the lake of deterministic and stationary input w_k. The algorithm is based on observations rather than inputs. The algorithm estimates the state vector given observations up to sample k. The algorithm will fail if the desired unknown states cannot be found from the observations gathered, therefore the observability of the system states must first be verified. This is performed by determining the rank of the observability matrix in the observation interval $n_1 \leq n \leq n_2$, defined as [13]

$$O = \sum_{i=n_1}^{n_2} \left[\left| \prod_{k=0}^{i-1} A_k^T \middle| H_i^T H_i \middle| \prod_{k=0}^{i-1} A_k^T \right|^T \right] \tag{16}$$

If O is rank-deficient, then it is not possible to obtain unique estimates of $\{x_{n_1}, \ \ldots, \ x_{n_2}\}$. A time-invariant system, in which H and A are not time-varying, O can be on a much simpler form, but in the nonlinear parameter estimator described here, this simplification is not available. Assuming the system model observable, the state-space parameter estimation problem may be stated as follows. Given observations $\{y_0, \ \ldots, \ y_n\}$ and the state-space model (2) and (3), find the optimal estimate of x_n, denoted $\hat{x}_{n|n}$.

If the noise vectors w_k and v_k assumed to be independent individually and mutually and uncorrelated with correlation matrices

http://dx.doi.org/10.5772/55989

$$E\left[w_i w_j^T\right]=Q_i\delta_{ij} \tag{17}$$

$$E\left[v_i v_j^T\right]=R_i\delta_{ij}\Big] \tag{18}$$

$$E\left[w_i v_j^T\right]=0 \tag{19}$$

Where δ_{ij} is the two dimensional delta function, then the Kalman filter provides the MMSE estimate of x_k as

$$\hat{x}_{n|n}=arg\ \min_{\hat{x}_n} E\left\{\|x_n - \hat{x}_n\|^2|n\right\} \tag{20}$$

The recursive Riccati equations are used to estimate the parameters [14]. The algorithm for kalman feature extraction is as follows [15]:

1. Initial estimates for $\hat{x}_k^+$ and $P_{k\text{-}1}^+$

2. Project the state ahead:

$\hat{x}_k^- = A_k \hat{x}_{k\text{-}1}^+$

3. Project the error covariance ahead:

$P_k^- = A_k P_{k\text{-}1}^+ A_k^T + w_k$

4. Compute the Kalman gain:

$K_k = P_k^- A_k^T \left(A_k P_k^- A_k^T + R_k\right)^{-1}$

5. Update feature vector estimate $\hat{x}_k^+$ with measurement y_k:

$\hat{x}_k^+ = \hat{x}_k^- + K_k\left(y_k - A_k \hat{x}_k^-\right)$

6. Update the error covariance:

$P_k^+ = \left(I - K_k A_k\right)P_k^-$

Kalman feature extractor uses the estimates of dynamics system state equation, with the new information contained in y_k fed back into the system through the Kalman gain. The block diagram description of the Kalman filter is given in Figure 1. A very important feature of the Kalman filter is that the error covariance does not depend on the observations. Hence $P_{k|k\text{-}1}$ can be pre-computed and the accuracy of the filter assessed before the observations are made. In particular, we may investigate the asymptotic behaviour of the filter by analyzing the discrete time Riccati equation.

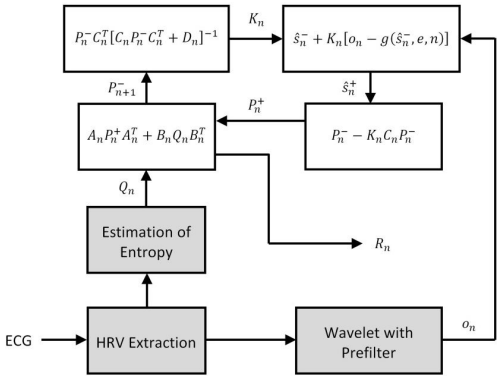

Figure 1. Block diagram of Kalman feature extractor [15].

3. Classification

The support vector machine (SVM) method has been used extensively for classification of EEG signals [16]. It is shown that EEG signal has separable intrinsic vectors which could be used in SVM classifier. SVM classifiers use discriminant hyper-planes for classification. The selected hyper-planes are those that maximize the margin of classification edges. The distance from the nearest training points usually measured based on a non-linear kernel to map the problem to a linear solvation space [17]. A Radial Basis Function (RBF) kernel based SVM is proposed here that Lagrangian optimization is performed using an adjustable ANFIS algorithm. It will be shown that due to conceptual nature of BCI for patients this proposed method leads to adjustable soft decision classification.

3.1. Classification using nonlinear SVM with RBF kernel

Training the SVM is a quadratic optimization problem in which the hyperplane is defined as [18]:

$$y_i\left(w\Phi\left(x_i,\ y_j\right)+b\right)\geq 1-\xi_i,\ \ \xi_i\geq 0,\ \ i=1,\ \ldots,\ l,\ \ j=1,\ \ldots,\ m \tag{21}$$

in which x_i is input vector, b is bias, w is adapted weights, ξ_i is class separation, $\Phi(x_i, y_j)$ is the mapping kernel, l is the number of training vectors, j is the number of output vectors, and y_i is desired output vector. The weight parameters should be achieved so that the margin between the hyperplane and the nearest point to be maximized. The only free parameter, C, in SVMs controlled the trade-off between the maximization of margin and the amount of misclassifications. Optimization of equation (9) yields to optimum w which is the answer of problem. It could be done using Lagrange multipliers defined as [11]:

$$L\ (w,\ b,\ \alpha,\ \mu)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{l}\xi_i-\sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i\alpha_j\left(y_j\left(w\Phi\left(x_i,\ y_j\right)+b-1+\xi_i\right)\right)-\sum_{i=1}^{l}\mu_i\xi_i \tag{22}$$

http://dx.doi.org/10.5772/55989

in which $C>\alpha_i$, $\alpha_j \geq 0$ *and* $\mu_i \geq 0$ *for* $i=1, \ldots, l$ *and* $j=1, \ldots, m$ and C is the upper bond for Lagrange coefficient. The coefficient C is a representation of error penalty so that higher values yield to bigger penalties. Karush-Kuhn-Tucher conditions lead to optimization of Lagrange multipliers [19]:

$$\frac{\partial L}{\partial W_v} = W_v - \sum_i \alpha_i y_i x_{iv} = 0 \tag{23}$$

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i = 0 \tag{24}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha - \mu_i \tag{25}$$

subject to

$$\mu_i\left(w\Phi\left(x_i, y_j\right) + b - 1 + \xi_i\right) \geq 0, \quad \xi_i \geq 0, \quad \alpha_i \geq 0 \tag{26}$$

and,

$$\alpha_i\left(y_i\left(w\Phi\left(x_i, y_j\right) + b - 1 + \xi_i\right)\right) = 0. \tag{27}$$

Usually nonlinear kernels provide classification hyper-planes that cannot be achieved by linear weighting [20]. Using appropriate kernels, SVM offered an efficient tool for flexible classification with a highly nonlinear decision boundary. RBF kernel was used in this section which is defined as:

$$\Phi(x, y) = exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \tag{28}$$

in which σ is the standard deviation. The proposed feature extraction method is depicted in Figure 2. The outputs are fed to the adjustable ANFIS described in next section. Two parameters had to be selected beforehand: the trade-off parameter C and the kernel standard deviation σ. They could be optimized for an optimal generalization performance in the traditional way, by using an independent test set or n-fold cross-validation. It has been suggested that the parameters could be chosen by optimizing the upper bound of the generalization error solely based on training data [26]. The fraction of support vectors, i.e., the quotient between the number of support vectors and all training samples, gave an upper bound on the leave-one-out error estimate because the resulting decision function changed only when support vectors were omitted. Therefore, a low fraction of support vectors could be used as a criterion for the parameter selection.

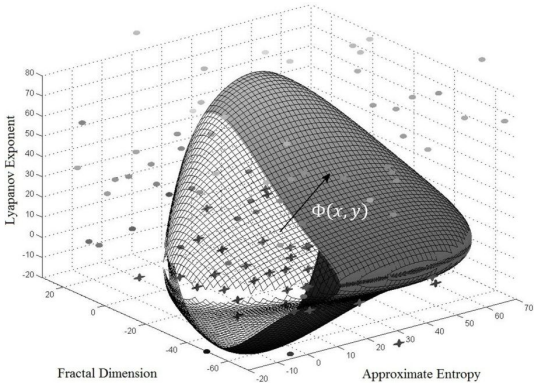

Figure 2. An example of feature vector for radial basis function kernel [15].

3.2. Adjustable ANFIS optimization

An ANFIS system could be used with Sugeno fuzzy model for fine adjustment of SVM classification kernels. Such framework makes the ANFIS modeling more systematic and less reliant on expert knowledge and therefore facilitates learning and adjustment. The ANFIS structure is shown in Figure 3. In the first layer, all the nodes are adaptive nodes. The outputs of layer 1 are the fuzzy membership grade of the inputs, which are given by [21]:

$$O_i^1=\mu_{A_i}(x),\ \ i=1,\ 2 \text{ and } O_i^1=\mu_{B_{i-2}}(x),\ \ i=3,\ 4 \tag{29}$$

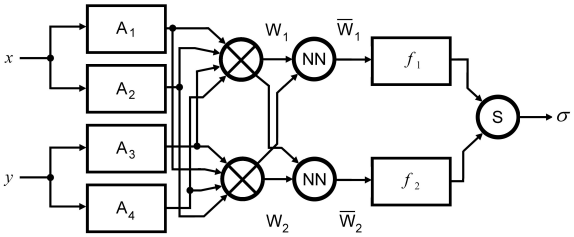

Figure 3. An example of feature vector for radial basis function kernel.

O_i^1 is the i th output of layer 1, $\mu_{A_i}(x)$ and $\mu_{B_{i-2}}(x)$ are type A and type B arbitrary fuzzy membership functions of nodes i and i - 2, respectively. In the second and third layer, the nodes are fixed nodes. They are labeled M and N respectively, indicating they perform as a simple multiplier. The outputs of these layers can be represented as:

$$O_i^2=w_i=\mu_{A_i}(x)\mu_{B_i}(x),\ \ i=1,\ \ \ldots,\ 4 \tag{30}$$

http://dx.doi.org/10.5772/55989

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_i + w_{i+1}} i = 1, \quad \ldots, 4 \tag{31}$$

which are the so-called normalized firing strengths. In the fourth layer, the nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first order polynomial for the first order Sugeno model. The outputs of this layer are given by:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) i = 1, \quad \ldots, 4 \tag{32}$$

in which f_i is the firing rate, p_i is the x scale, q_i is the y scale, and r_i is the bias for i th node. In the fifth layer, there is only one single fixed node that performs the summation of all incoming signals:

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \sum_{i=1}^{2} \frac{w_i f_i}{w_i + w_{i+1}} \tag{33}$$

It can be observed that there are two adaptive layers in this ANFIS architecture, namely the first layer and the fourth layer. In the first layer, there are three modifiable parameters $\{a_i, \ b_i, \ c_i\}$, which are related to the input membership functions. These parameters are the so-called premise parameters. In the fourth layer, there are also three modifiable parameters $\{p_i, \ q_i, \ r_i\}$, pertaining to the first order polynomial. These parameters are so-called consequent parameters.

The task of the learning algorithm for this architecture is to tune all the above mentioned modifiable parameters to make the ANFIS output match the training data. When the premise parameters of the membership function are fixed, the output of the ANFIS model can be written as:

$$\sigma = (\overline{w}_1 x) p_1 + (\overline{w}_1 x) q_1 + (\overline{w}_1 x) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2. \tag{34}$$

This is a linear combination of the modifiable consequent parameters $p_1, \ q_1, \ r_1, \ p_2, \ q_2$ and r_2. When the premise parameters are fixed the least squares method is used easily to identify the optimal values of these parameters after adjustment of ANFIS weights using SVM. When the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower.

A hybrid algorithm combining the least squares method and the gradient descent method was adopted to identify the optimal values of these parameters [22]. The hybrid algorithm is composed of a forward pass and a backward pass. The least squares method (forward pass) was used to optimize the consequent parameters with the premise parameters fixed. Once the optimal consequent parameters are found, the backward pass starts immediately. The gradient descent method (backward pass) was used to adjust optimally the premise

parameters corresponding to the fuzzy sets in the input domain. The output of the ANFIS was calculated by employing the consequent parameters found in the forward pass. The output error was used to adapt the premise parameters by means of a standard back propagation algorithm. It has been shown that this hybrid algorithm is highly efficient in training the ANFIS [21].

A classification method base on ANFIS adapted SVM proposed in this section. It showed that non-linear features improve classification rate as an effective component. Through the feature space constructed using approximate entropy and fractal dimension in addition to conventional spectral features, different stages of EEG signals can be recognized from each other expressly. The successful implementations of ANFIS-SVM for EEG signal classification was reported in this context. The results confirmed that this method provides better performance in datasets with lower dimension. This performance achieved for RBF kernel of SVM modified by ANFIS. This section can be a strong base for improved methods in the field of BCI cognition and analysis for therapeutic applications.

3.3. Recursive least square

In this method a third order AR model is used that directly estimates EEG parameters without using any intrinsic model. Such linear methods could be used to estimate nonlinear systems in piece wise linear manner. $\varphi^T(t-1) = -[X(t-1) \quad X(t-2) \quad X(t-3)]^T$ is input matrix in which $X(t)$ is input at time t and $\hat{\theta}(t) = -[b_1 \quad b_2 \quad b_3]^T$ is the model parameters. Iterative least square parameter modification is computed by following equations [23].

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \left(\varphi^T(t)\varphi(t)\right)^{-1}\varphi(t-1)\xi(t) \tag{35}$$

$$\xi(t) = X(t) - \varphi^T(t-1)\hat{\theta}(t-1) \tag{36}$$

Use of parametric model provides a time variant estimate of state equations of system at the working point. It is in accordance to the highly variable nature of EEG signals. In fact the modeling of nonlinear and time variant signal using a feature vector with limited dimensions provides a filtering on data. The abstract characteristics of signal would be extracted that leads to lower variance compared to the frequency feature extraction methods. Dimension of feature vector is a critical point to avoid over learning for high values of features or lake of convergence due to small size of feature vector.

3.4. Coupled hidden Markov models

Use of multimodal approaches provides improvement in robustness of classification under disturbances. This could extend the margin of security for BCI systems. Integration of two sets of features namely set *a* and set *b* could be done by various methods. Several integration models have been proposed in the literature that can be divided into early integration (EI) and late integration (LI). In the EI model, information is integrated in the feature space to form a feature vector combined of both features. Classification is based on this composite feature vector. The

http://dx.doi.org/10.5772/55989

model is based on assumption of conditional dependence between different modes and therefore is more general than the LI model. In the LI model, the modules are pre-classified independently of each other. The final classification is based on the fusion of both modules by evaluating their joint occurrence. This method is based on assumption of conditional independency of both data streams. It is generally accepted that the auditory system performs partial identification in independent channels, whereas, BCI classification seems to be based on early integration which assumes conditional dependence between both modules [24]. This theory is based on LI method and models pseudo-synchronization between modules to account some temporal dependency between them. It is a compromise of EI and LI schemes. The bimodal signal is considered as an observation vector consists of two sets of features. Optimum classifier which is based on Bayesian decision theory could be obtained using the maximum *a posteriori* probability function [25]:

$$\lambda_0 = \max_{\lambda} P(\lambda \mid (O^a, O^b)) = P((O^a, O^b) \mid \lambda) P(\lambda) / P(O^a, O^b) \tag{37}$$

$$\lambda = (A, B, \pi) \tag{38}$$

Where A is state transition matrix, B is observation probability matrix, π is initial condition probability matrix, O represents the sequence of feature vectors. In this context the superscript a, denotes the first stream parameters and superscript b denotes the second stream parameters.

The parameters are nonlinear which could be seen in equations 3 and 4. The system identification is based on linear ARMA model that means the parameters are computed well only around the operating point of the system. The operating point of system is time-varying and therefore the parameters vary with time. Therefore, the feature vector parameters are computed at each frame during train and test of HMM.

Training of the above mentioned multistream model could be done by synchronous and asynchronous methods. In this section the multistream approach is based on pseudo-synchronous coupled hidden Markov model. As it will be shown, it is an interesting option for multimodal continuous speech identification because of, 1) synchronous multimodal continuous speech identification, 2) consideration of asynchrony between streams. Some resynchronization points are defined at the beginning and end of BCI segments including phonemes or words.

Some resynchronization points are defined at the beginning and end of BCI segments including phonemes or words. Combination of the independent likelihoods is done by multiplying the segment likelihoods from the two streams, thus assuming conditional independence of the streams according to:

$$P((O^a, O^b) \mid \lambda) = P(O^a \mid \lambda^a)^w P(O^b \mid \lambda^b)^{(1-w)} \tag{39}$$

The weighting factor $w0 \leq w \leq 1$ represents the reliability of the two modalities. It generally depends on the performance obtained by each modality and on the presence of noise and

disturbance. Here, we estimate the optimal weighting factor on the development set which is subject to the same noise as the test set. The method used for final experiments however was to automatically estimate the SNR from the test data and to adjust the weighting factor accordingly. It can be observed empirically that the optimal weight is related almost linearly to the SNR ratio.

BCI with dynamic parameters was studied in this section. The algorithm adopted is based on pseudo-synchronous hidden Markov chains to model the asynchrony between events. The proposed combination of ARMA model and Kalman filtering for feature extraction resulted to the best identification rates compared to usual methods. Complimentary effect of video information and dynamic parameters on BCI was studied. Effectiveness of the proposed identification system was more beneficial in low signal to noise ratios which reveals the robustness of the algorithm adopted in this study. Owing to high rate of information, the voice of speakers has significant effect on the identification rate; however, it reduces rapidly due to environmental noise. It was also shown that a specific combination weight of voice and video information provides optimum identification rate which is dependent on the signal to noise ratio and provides low dependency on the environmental noise. The phonetic content of spoken phrases was evaluated and the phonemes were sorted based on their influence on BCI rate. Identification rate of proposed model based system was compared to other parameter extraction methods including Kalman filtering, neural network, ANFIS system and auto regressive moving average. The combination of proposed model with Kalman filter led to the best identification performance. Combination of feature vectors content has a great roll on identification rate, Therefore, more efficient methods rather than pseudo-synchronized hidden Markov chain could be used for better achievements. Application of feature extraction methods like sample entropy, fractal dimension and nonlinear model based approaches have shown appropriate performance in BCI processing and could result to better identification rates in this area. Lip shape extraction is a critical point in this identification method and more robust algorithms provide accurate and precise results.

4. Conclusions

Some promising methods for feature extraction and classification of EEG signal are described in this chapter. The aim of these methods is to overcome the ambiguities encountering BCI applications.

A feature extraction algorithm based on the Kalman estimation discussed. This estimator minimizes the error covariance and therefore is an optimum estimator if appropriate initial condition is selected. Kalman estimator is used for adaptive estimation of dynamic parameters of EEG. The estimator reduces the error variance adaptively and after a period of time a unique estimation is achieved.

The SVM has been used extensively for classification of EEG signals. It is shown that EEG signal has separable intrinsic vectors which could be used in SVM classifier. SVM classifiers use discriminant hyper-planes for classification. The selected hyper-planes are those that

http://dx.doi.org/10.5772/55989

maximize the margin of classification edges. The distance from the nearest training points usually measured based on a non-linear kernel to map the problem to a linear solvation space. A RBF kernel based SVM is proposed here that Lagrangian optimization is performed using an adjustable ANFIS algorithm. It will be shown that this method leads to adjustable soft decision classification.

The combination of proposed model with Kalman filter can lead to the best identification performance. Combination of different feature sets has a great roll on classification rate, Therefore, more efficient methods rather than pseudo-synchronized hidden Markov chain could be used for better achievements. Application of feature extraction methods like sample entropy, fractal dimension and nonlinear model based approaches have shown appropriate performance and could result to better identification rates in this area.

Through the feature space constructed using approximate entropy and fractal dimension in addition to conventional spectral features, different stages of EEG signals can be recognized from each other expressly. These methods provide better performance in datasets with lower dimension. The RBF kernel of SVM modified by ANFIS can be a strong base for improved methods in the field of BCI cognition and analysis for therapeutic applications.

Author details

Vahid Asadpour, Mohammd Reza Ravanfar and Reza Fazel-Rezai

University of North Dakota, USA

References

[1] Guillaume, S. Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. IEEE Transactions on Fuzzy Systems (2001). , 9(3), 426-443.

[2] Guler, I D. Adaptive neuro-fuzzy inference system for classification of EEG signals using wavelet coefficients. Neuroscience Methods (2005). , 148-113.

[3] Anier, A, Lipping, T, Mel, S, & Hovilehto, S. Higuchi fractal dimension and spectral entropy as measures of depth of sedation in intensive care unit. Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2004). , 526-529.

[4] Sandeep, P N, Shiau, D, Principe, J C, Iasemidis, L D, Pardalos, P M, Norman, W M, Carney, P R, Kelly, K M, & Sackellares, J C. An investigation of EEG dynamics in an animal model of temporal lobe epilepsy using the maximum Lyapunov exponent. Experimental Neurology (2010). , 216(1), 115-121.

[5] Derya, E. Recurrent neural networks employing Lyapunov exponents for analysis of ECG signals. Expert Systems with Applications (2010). , 37(2), 1192-1199.

[6] Derya, E. Lyapunov exponents/probabilistic neural networks for analysis of EEG signals 2010. Expert Systems with Applications (2010). , 37(2), 985-992.

[7] Lia, J, Chena, Y, Zhangb, W, & Tiana, Y. Computation of Lyapunov values for two planar polynomial differential systems. Applied Mathematics and Computation (2008). , 204(1), 240-248.

[8] Mendel, J. M. Lessons in Estimation Theory for Signal Processing, Communications, and Control. Englewood Cliffs; (1995).

[9] Franklin, G. F, Powell, J. D, & Chapterman, M. L. Digital Control of Dynamic Systems. Addison-Wesely; (1990).

[10] Choi, J, Lima, A. C, & Haykin, S. Kalman Filter-Trained Recurrent Neural Equalizers for Time-Varying Channels. IEEE Transaction on Communications (2005). , 53(3), 472-480.

[11] Brown, R. G. Hwang PYC. Introduction to Random Signals and Applied Kalman Filtering. Wiley; (1992).

[12] Bishop, G, & Welch, G. An Introduction to the Kalman Filter. University of North Carolina at Chapel Hill, Lesson Course; (2001).

[13] Verdu, S. Minimum probability for error for synchronous Gaussian multiple-access channels. IEEE Transactions on Information Theory (1986). , 32-85.

[14] Grewal, M. S, & Andrews, A. P. Kalman Filtering: Theory and Practice. Prentice Hall; (1993).

[15] Vatankhah, M, & Asadpour, V. Reza Fazel-Rezai, "Perceptual Pain Classification using ANFIS adapted RBF Kernel Support Vector Machine for Therapeutic Usage", Applied Soft Computing, (2013).

[16] Derya, E. Least squares support vector machine employing model-based methods coefficients for analysis of EEG signals. Expert Systems with Applications; (2009).

[17] Cheng, C, Tutwiler, R. L, & Slobounov, S. Automatic Classification of Athletes With Residual Functional Deficits Following Concussion by Means of EEG Signal Using Support Vector Machine. IEEE Transactions on Neural Systems and Rehabilitation Engineering (2008). , 16(4), 327-335.

[18] Taylor, J. S, & Cristianini, N. Support Vector Machines and other kernel-based learning methods. Cambridge University Press; (2000).

[19] Vahdani, B, Iranmanesh, S. H, Mousavi, S. M, & Abdollahzade, M. A locally linear neuro-fuzzy model for supplier selection in cosmetics industry. Applied Mathematical Modeling (2012). , 36(10), 4714-4727.

[20] Cristianini, N, & Shawe-taylor, J. Support Vector and Kernel Machines. Cambridge University Press; (2001).

[21] Jang, J. S. ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on System Man Cybernetic (1993).

[22] Zhan-li, S, & Kin-fan, A. Tsan-Ming Choi. Neuro-Fuzzy Inference System Through Integration of Fuzzy Logic and Extreme Learning Machines. IEEE Transactions on Systems, Man, and Cybernetics (2007). , 37(5), 1321-1331.

[23] Ljung, L, & Soderstorm, T. Theory and Practice of Recursive Identification. MIT Press; (1983).

[24] Fletcher, H. Speech and Hearing in Communication. Krieger; (1953).

[25] Doud, H. Y, & Gururajan, A. Bin He A. Cortical Imaging of Event-Related (de)Synchronization During Online Control of Brain-Computer Interface Using Minimum-Norm Estimates in Frequency Domain. IEEE Transactions on Neural Systems and Rehabilitation Engineering (2008). , 16(5), 425-43.