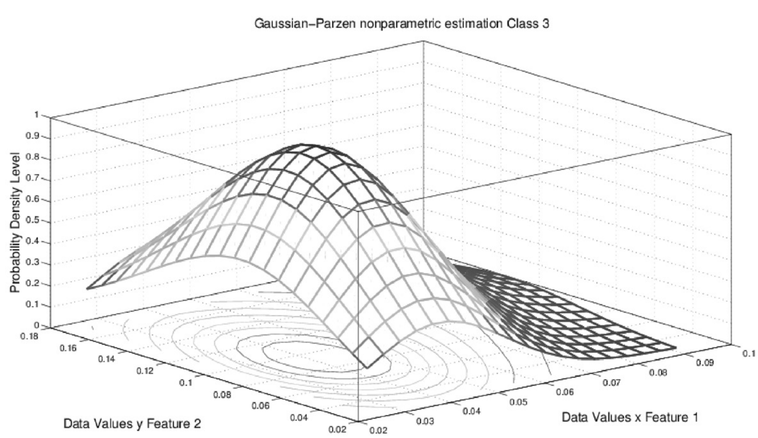
EEG signals classification using linear and non-linear discriminant methods

Juan manuel Mayor torres*



Abstract

This article was developed with the particular interest of characterize and study EEG signals as a pattern which in general has a high dimensionality, and has obviously a particular behavior in frequency and time. Here we have developed a wavelet decomposition to reduce a little bit the dimensionality and PCA (Principal Components Analysis) to accurate the result in a better way (only two features representation). After that the EEG signals, with their respective characteristics and representation has been able to train and test some linear and non-linear classifiers such as (Parzen, k-NN, Radial Basis Neural Network, linear and non-linear perceptron and so on.) This evaluation is an analysis of general EEG's behavior signals with this kind of characterization and classification processes respectively.

Keywords: linear classifiers, non-linear classifiers, EEG signals, feature extraction, biomedical signals.

(*) Pontificia Universidad Javeriana. Cali, Colombia. E-mail: jmmayor@javerianacali.edu.co.

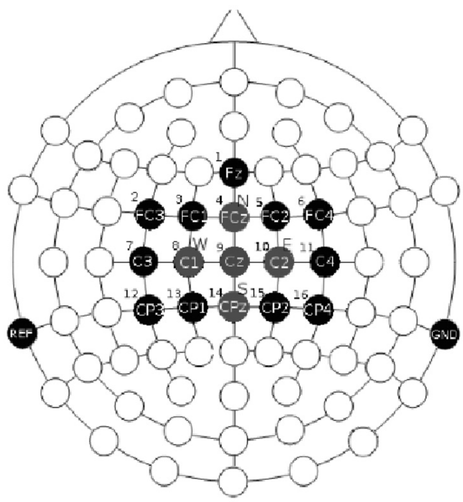
Este documento se ha construido a partir de la ponencia del mismo nombre presentada por los autores en el "I Congreso Internacional de Ingeniería Mecatrónica y Automatización", organizado por el Programa de Ingeniería Mecatrónica de la Facultad de Ingeniería de la Universidad Autónoma de Occidente, con el apoyo del Capítulo ACOFI de los programas de Ingeniería Mecatrónica y de Automatización. El documento es inédito.

Fecha de recepción: 18/04/2013 • Fecha de aceptación: 30/04/2013.

1. Introduction (to EEG signals)

First of all, we have to enunciate the source of the EEG data files that we have used for this study that included on this web page http://www.bbci.de/competition/iv/#dataset3, http://www.bbci.de/competition/iv/dataset3. In this dataset we found the distribution for a fourty series of data. The entire data distribution shows a variation following the 10-20 standard for EEG-MEG signal acquisition (Homan, Herman & Purdy, 1987) that is shown in the Figure 1.

Figure 1. The Typical Disposition of the EEG non-invasive sensors over the head



Source: by the author.

The electrodes are non-invasive, and requires an electrical gel to improve the signal qualities, beside it a rehabilitation expert or medical doctor should be there in case where the helmet was putting in the patient head. This approach has two important limitations.

- The non-invasive nature makes that any movement or element between the sensors and the craneal cavity that has a frequency representation too.
- The interference is given electronic devices or synthetic or metalic prosthetics inside the craneal cavity makes the signals that differ a lot.

CNBI-BCI research group from the EPFL Switzerland http://cnbi.epfl.ch/cnbi-pujcCNBI-PUJC works in this approach and use this device to get the EEG signals. These signals (for the lab case were not taken by the same device), are taken using a gTec device called http://www.gtec.at/Products/Hardware-and Accessories/g.USBamp-Specs-FeaturesgUSBAmp, this Austrian company develops this device to amplify and add somo filter capabilities to the result signal, the signals are condensed in 4 channels that represent the sensor position. In Figure 2.

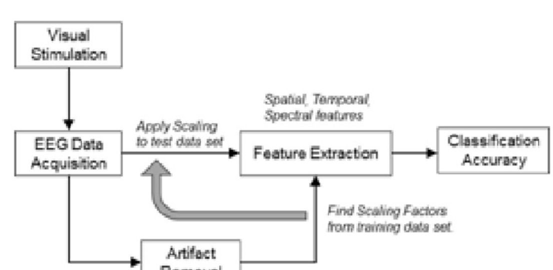
Figure 2. gTec device to make the signal preprocessing

Source: by the author.

EPFL has developed an entire BCI library to process the signal acquired by the device and extract it in terms of some labels for the future classification. The main interface with the user is the http://cnbi.epfl.ch/software/eegview.html EEGVIEW and all linux packages that have been developed for this are from CNBI.

Now, inside these apps appear a similar disposition or a standard of signal treatment that is used commonly in signal preprocessing and a future postprocessing or classification algorithm, as we can see on Figure 3. First of all CNBI-BCI (Millán & Chavarriaga, 2011; Millán, Ferrez, Galán, Lew & Chavarriaga, 2008), description expresses the following.

Figure 3. Basic diagram for EEG signals processing



Source: by the author.

I). For the preprocessing task, CNBI libraries use a similar DFT technique called PSD that is the sequential calculation of the power spectrum following the equation 1, that is the typical calculation of the power spectrum given by the instantaneuos square norm of the signal spectrum.

$$P[\omega]=\lim_{T\to\infty}\varepsilon\left[\left|\int_{-\infty}^{\infty} f(t)\exp(-j\omega t)\partial t\right|^{2}\right] \quad (1)$$

Now this equation does not avoid the prescence of the noise power (equation 2), that can be infered by the integral nature of the numerical expression. However CNBI has developed some laplacian and DC filters to avoid the neighbor sensors interference.

$$P[\omega]=P[f(\omega)]+P[N_a(\omega)] \quad (2)$$

II). Now , in the posprocessing case, CNBI uses GMM (Gaussian Mixture Model), that describe basically a linear and quadratic (equation 4) classifier that employs multiple decision region over the entire data space, the probability densities for all Gaussian models are described on 3

$$a_{i,j}=\frac{1}{2\pi\sqrt{|\Sigma_{i,j}|}}\exp\left(-\frac{(x-\mu_{i,j})^2}{\Sigma_{i,j}}\right) \quad (3)$$

Besides it, the discriminant region varies depending on the covariance matrices of the different propability estimations per each class including in the problem. However, CNBI optimizes the values of covariance matrices minimizing the error over the discriminant region, in terms of the covariance matrices itself. Although, this methology could improve the classifier performance, the lower degrees of the discriminant regions does not allow a better classification.

$$R(x)_{i,j}=(x-\mu_{i,j})\Sigma_{i,j}(x-\mu_{i,j})^T+\alpha_{i,j}(\Sigma_{i,j}) \quad (4)$$

To sum it up, these aspects are the critical variables, which could affect the EEG signals at time, to make an adequate classification for a hypothetically rehabilitation improving process in real neurological scenario with affected patients.

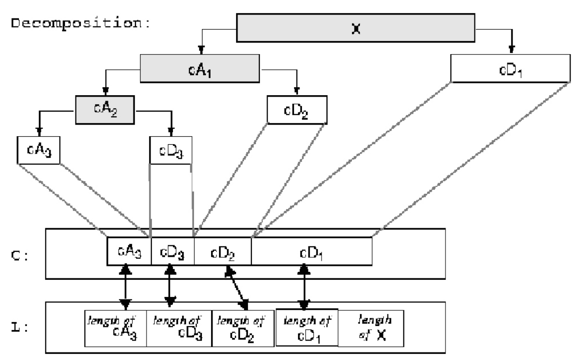
- Method evaluation

First of all, we have to say that downloading the datafiles we found two data sequence of 4 classes labeled, in wrist movement (left, right, away and towards subject body movements) respectively. The datasets contain 40 samples package with 10 sensor samples and each of them with 400 samples per each class, the sampling frequency in this case was 400 *Hz* signal; the frequency nature of these signals stay around (5-30 *Hz*), depending on the nature of the movement.

2. Linear discriminants

First we have to characterize the signals that have an entire dimensionality *D*=10x4x40x400=320000 for a total amount of characterisitcs that is a huge problem for any classifier type. For this case we employ, wavelet decomposition to obtain the first charateristics minimization, the dynamic of wavelet applycation is shown in Figure 4

Figure 4. Wavelet decompositon basic diagram



Source: by the author.

- Wavelet decomposition

Wavelet transform (Mallat, 1999), uses an specific equation that transforms time domain to frequency domain and backwards, as we show in the equation 5, but this expression has the characteristic, that its spectrum represents a QMF filter (Quadrature Mirror Filter)

$$C(a,b,\Psi(t),f(t))=\int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{(a)}}\psi\left(\frac{t-a}{b}\right)\partial t \quad (5)$$

Ψ is the mother source wavelet, that could be a lot of mathematical waves constructions such as: Daebuchies, Morlet, Haar, Biorthogonal, Symlet, Coiflet and so on, a lot of cases presented there. The values *a, b* are the scaling values that allow the enlargement or make short the mother wave. This could be a better behavior to capture the wave component that the problem really needs.

Now, with the entire wavelet decomposition and its coefficient downsampling the entire signal 2 per 2 per each level in the wavelet decomposition. Then, we can reconstruct the signal using that filters or simply to use these coefficient to represent the signal. Finally the representation would be the approximation coeffiencients and all the detail coefficient minimized per each signal level, those could be choosing between them and with this, we can minimize the entire data description at least to a half of the input size.

This process has developed for every channel of 400 samples and we use it for all the study. Besides it, we probe the Daebuchies signal type (Hinterberger, Kübler, Kaiser, Neumann & Birbaumer, 2003), and the size diminishes at least to half with a great description per each level of wavelet decomposition.

- PCA

The characteristics extraction techniques was PCA or (Principal Components Analysis) (Subasi & Gursoy, 2010), this is a technique that diminishes the entire signal or data dimensionality, using the most variability criterion inside the entire data class. This could be accomplished evaluating the covariance matrix of the entire data aggregate, if we evaluate the covariance matrix we need their mathematical representative data for all features spatial system and it could be represented with the eigenvalues and the eigenvector of that matrix.

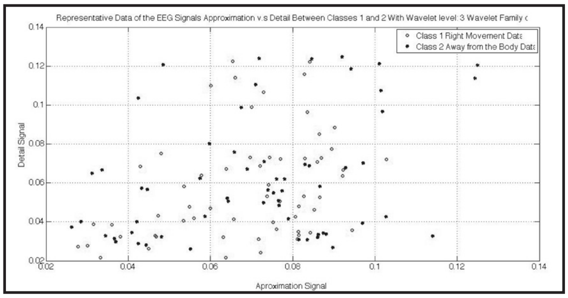
We use this criterion for all classes to evaluate PCA. First we normalize the data substracting using mean value as a operator parameter per each class and after that we put the eigenvalues of the covariance matrix choosing the eigenvector that correspond to maximun variance eigenvalue of the entire matrix. These data would represent the most important variation side of the data aggregate, equation six shows the linear system relationship to find the eigvalues using matrix decomposition.

$$\Sigma x = \Sigma_{eig}\Sigma_{val} \qquad (6)$$

We employed that analysis per each channel and their respective wavelet coefficientes [approximation (low frequency)] and [detail (high frequency)] finally we obtain only one channel of wavelet cofficient per feature (approximation and detail).

Finally the result that I obtained to evaluate the class 1 (left movement) and class 3 (away of the body movement), could be seen in Figure 5, the data are similar in all cases and we have to found the better evaluation of Daebuchies levels to make a discrimination region easier.

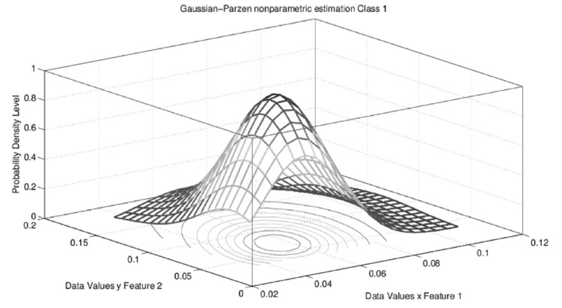
Figure 5. Data Distribution for class 1 and class 3



Source: by the author.

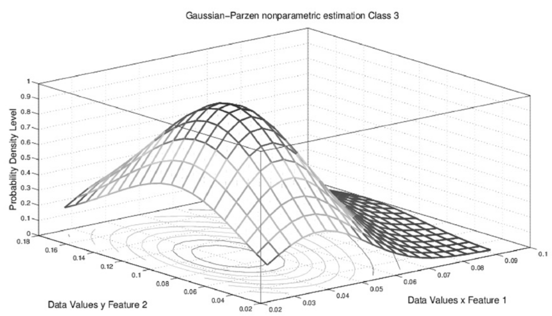
For this case (class 1 and class 3 experiment) we obtain specifically the two following graphs in Figure 6 and Figure 7.

Figure 6. Probability distribution using a Parzen radio σ_h=0.1 for class 1



Source: by the author.

Figure 7. Probability distribution using a Parzen radio σ_h=0.1 for class 3



Source: by the author.

Table 1. Linear Discriminants Evaluation

	LDC	NMC	Fisher
Training error	0.425	0.475	0.425
Testing error	0.55	0.48	0.51

Source: by the author.

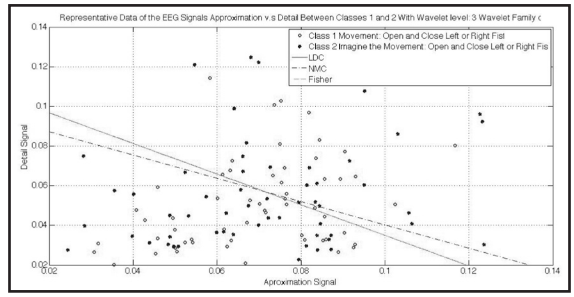
The values in the axes x and y are two representative features of each class, if anybody could put upon the probability density in all cases over the points over xy axis, the common tendency could be evaluated as the accumulation of data in space features.

In this case we solved the classification phase, using Prtools 4.2.1 for october 2011 over Matlab 2010a, the use of Prtools is quite simple and we could have the characterization values of data to classify the extracted datasets.

Now with dataset objects we can apply a classifier only multiplying the dataset with the classifier object empty, or we can use the classifier function generation to train the classify given the dataset as a function input parameter.

We plotted the following classifiers on Figure 8 just upside the data distribution and the results were enough for the signals characteristics and our still incomplete analysis of the preprocessing task.

Figure 8. Linear classifiers region for the classes 1 and 3 and Daebuchies level 3

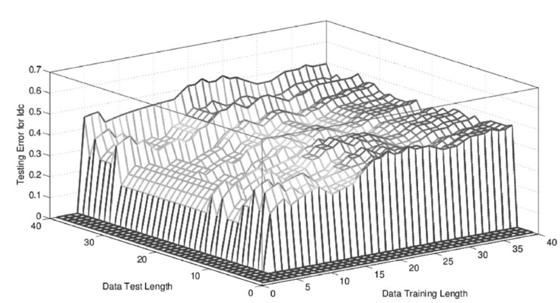


Source: by the author.

The analysis delivers to us some statements with the estimation error for these classifiers and this data aggregate here is the Table 1 with the respective output statements.

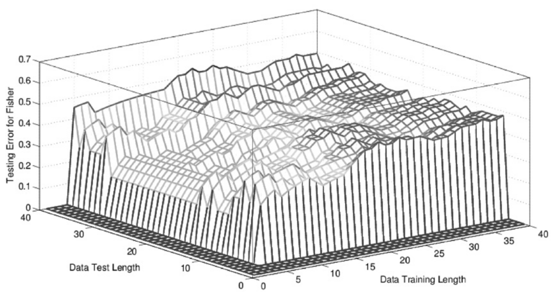
Now, after that we evaluated the classifiers performance, but it could be a difficult task in terms of analitical approach, then we have to analize the linear classifiers performance using empirically the data which have been clasified in a correct way, using a confusion matrix. For error evaluation we obtain the following behaviors in Figure 9 and 10 for LDC and Fisher classifiers respectively.

Figure 9. The error of LDC classifier in terms of the training and testing dataset lengths



Source: by the author.

Figure 10. The error of LDC classifier in terms of the training and testing dataset lengths

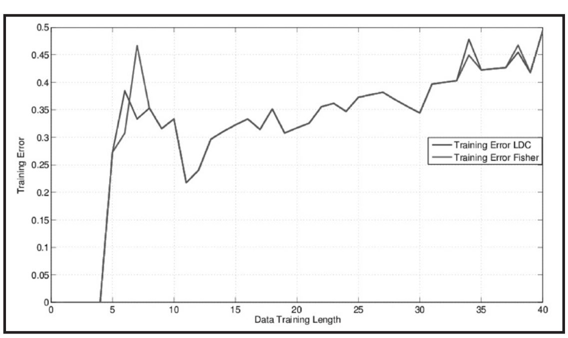


Source: by the author.

As we can see on the figures above, the testing error, oscilates and has some regions where has local minimums and maximums. It is similar between both linear classifiers, the classifieres performance in terms of error is fixed in terms of the evaluation, that has developed for the first element of the entire dataset.

In case of training error, we have only the training dataset length as an independant variable and we can see a similar behavior (Figure 11), where LDC and Fisher classifiers have similar behaviors and varies almost in the same way, Fisher is a little bit more variable than LDC.

Figure 11. The error LDC and Fisher classifier's error in terms of the Training dataset length



Source: by the author.

The next task that we developed is the analysis of Parzen and k-NN classifiers, both are non-linear but depends a lot on the spatial data distribution and the probabilistic estimation over the data. Now we are going to evaluate two cases of different sizes of training dataset, we want to vary the two parameter of both classifiers, in case of Parzen, the parzen ratio or σ_h and in case of k-NN the number of neighbors that involves the decision area per any sample k_n.

To sum it up, Parzen classifier works with a Mahalanobis distance (equation 4) to make an estimation of the probability density per each sample inside data system, later with it we can estimate the distance and substitute the belogness probablity as a Gaussian dynamic and with the parzen ratio factor we can infer that.

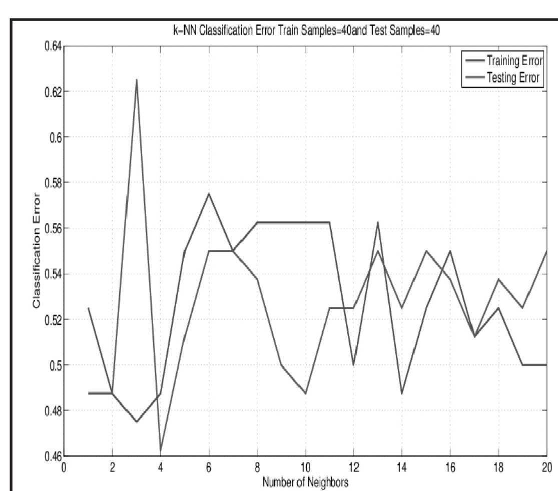
The decision region classifies if the likehood per class is greater than the class j, then the data label could change or remains the same.

In case of k-NN the procedure has a similarity with the previous one excepts for samples of the data system those classifier take k_n and choose the furthest distance between around the k_n nearest neighbor in a circle or in N-dimension (N characteristics) hypersphere. If the volume of this hypersphere estimates in a proper way the probability to belong class i with these samples is lower than a thershold, the data would not be inside this N-dimensional space, the opposite if this probability is higher. The decision rule is the following (equation 7)

$$\frac{3k_i}{4\pi(|x_n - k_{ni}|)^3 N_i} \geq \frac{3k_j}{4\pi(|x_n - k_{nj}|)^3 N_j} \tag{7}$$

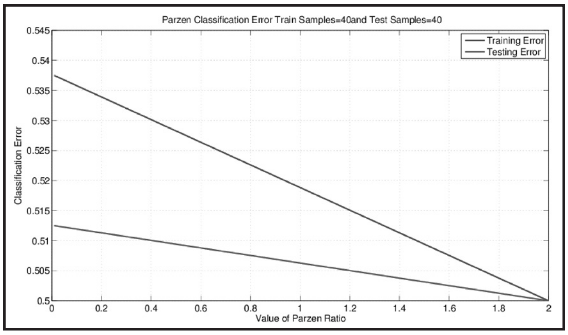
Now for that we have the following results (Figure 12 and 13).

Figure 12. k-NN classifier error using the same data size to train and test



Source: by the author.

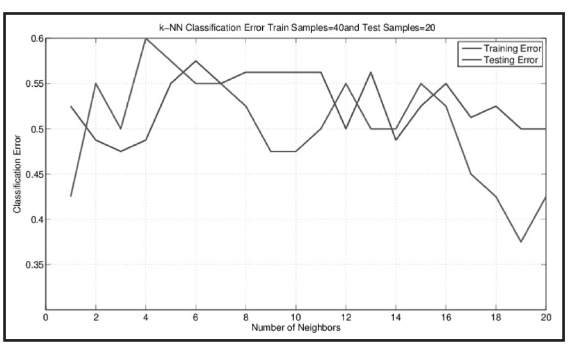
Figure 13. The parzen classifier error using the same size to Train and Test



Source: by the author.

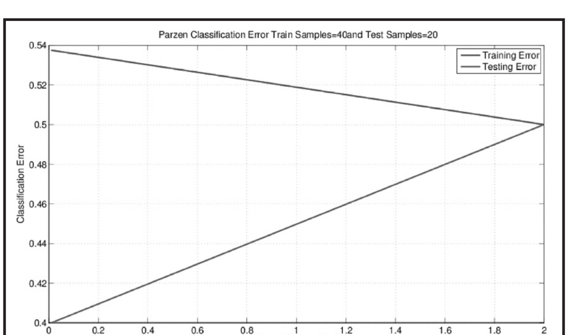
As we can see, the variability of k-NN obeys to the spatiality dependency and the simple use of the decision rule that is so proper to high error with the most part of *k* neighbors options. On the other hand, parzen has a soft decay rate and it is similar for both error but, now in case of a testing set variation the results will be different, such as Figure 14 and 15.

Figure 14. The parzen classifier error using the same size to Train and Test



Source: by the author.

Figure 15. The parzen classifier error using train data set for all samples and test only a half



Source: by the author.

Now we can see a difference, because in k-NN case the train and test error do not obey between themselves a particular behavior in terms of the number of *k* neighbors. On the other hand, in case of Parzen the test error is not decaying but start so low in comparison with the train error, it would be more predictable.

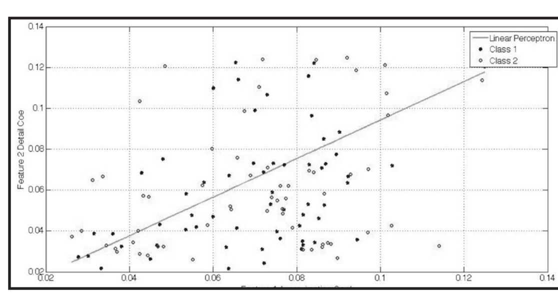
For this case of the last linear classifier we developed a linear perceptron for the data system of the wavelet coefficients. The linear perceptron has a activation function that is equal to a heavyside, step or a lineal output, and for this we did not use a backpropagation training this time. We only use a linear evolution of the weights inside every neurone.

The weights will evolve just like equation 8, and depends on a growth rate *n* that for our case in this experiment *n* is 0.05 because the numerical dimension of all data is around this and we do not have that error could go down fast. However, this analysis has a problem, the linearity of the activation functions and the fix nature of the outputs (class number), is not great to accurate the linear model if the backpropagation algorithm is not developed.

$$w_k(n+1) = w_k(n) + \eta E_{k,n} x_k \quad (8)$$

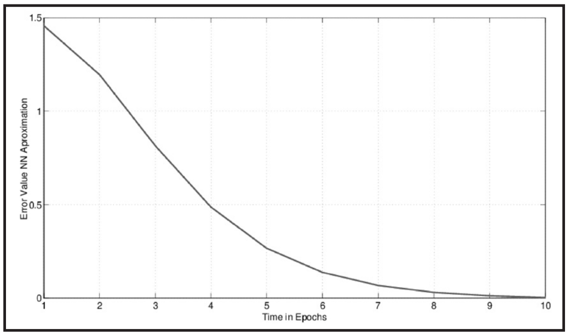
Now the results for this development could be seen in the Figure 16 and 17, where we can see the linear relationship between the features, weights and the error evolution for the output layer.

Figure 16. The linear perceptron discriminant region



Source: by the author.

Figure 17. The linear perceptron mean error for the outputn never arrives totally to zero



Source: by the author.

3. Non-linear discriminants

For this approach we have a better classifieres with a irregular regions and a great variability in the decision approaches, and around it exist three important techniques that we going to evaluate Multilayer Non Linear Perceptron, Radial Basis Networks and Support Vector Machine, each of them have its own limitation and paramaters to be tunned and defined.

- **Multilayer Perceptron.** The MLP (Lotte, Congedo, Lécuyer, Lamarche & Arnaldi, 2007) has a great popularity in the Pattern Recognition bakground. Its implementation queue and work is basically using the backpropagation algorithm, in this case the error will propagate to last neuron to the begin of the network array changing the weights values in non-linear way.

$$w_k(n+1) = w_k(n) - \eta \frac{\partial J}{\partial W} \tag{9}$$

The equation 9 reveals the dynamic of backpropagation algorithm (negative gradient) and within a series of activation function which are non-linear the domain changes for every neuron would be non-linear at the same time.

The J value (equation 10) is the complete addition of all neuron error values, and to make an optimization of this value we have to find the more changeable direction of the errors per any neuron.

$$J = \frac{1}{2}\sum_P \sum_M (d_{m,p} - y_{m,p})^2 \tag{10}$$

- **Radial Basis Network.** The Radial Basis Network (Qiu, Fung, Chan, Lam, Poon & Hamernik, 2002) uses the same scheme of the MLP, but the activation function does not allow a convergent value for infity time, then the activation function is unitary but similar to a unitary window, that is cut for infinity and minus infinity time. The quality for this analysis is that the Radial Basis Network requires a centroid value for any neuron in the neuron array, these arrays could represent a correspond centroid for any class in the data space in analog way.

Another advantage of this method is the gaussian mapping for the weights calculation and the back propagation algorithm, the main equation for this dynamic is shown by the equation 11.

$$g(x) = w_0 + \sum_{p=1}^{P} w_p \exp\left(-\frac{(x-c_p)(x-c_p)^T}{2\sigma_p^2}\right) \tag{11}$$

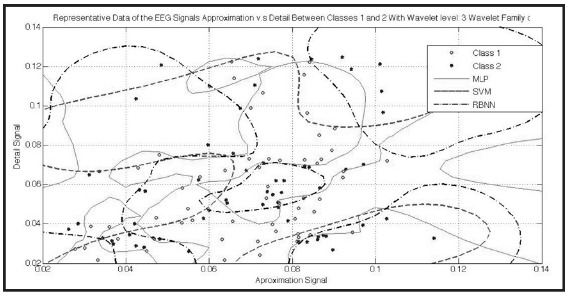
The equation shows the discriminant region for RBN, and c_p are the values of the centroid per any class.

- **Support Vector Machine.** This technique is so versatile (Li, Guan, Li & Chin, 2008), and is capable to tansform a linear space such as the perceptron dynamic equation in a non linear data fields, where the dynamic function to make the decision would be a non-linear kernel, the main objective for this analysis is expressed in the equation, that is only a basic approach for the non-linear transformation (trasformation equation 12).

$$x(x_1, x_2) \rightarrow \Theta\left(x_1^2, x_2^2, \exp(|x_1 - x_2|)\right) \tag{12}$$

Now for the evaluation of these classifiers, we are going to use the same data file and for that case the results are much better than the linear classifiers, the Figure 18 shows the discriminant regions.

Figure 18. Non linear classifier evaluation for class 1 and class 3



Source: by the author.

For the entire system, we have the following errors and the confusion tables with the output clasifier values. And then for the testing error case the statements are presenting on Table 2.

Table 2. Non linear discriminants evaluation

	MLP	SVM	RBNN
Training error	0.2625	0.3125	0.325
Testing error	0.45	0.45	0.425

Source: by the author.

Making an adequate comparison with some other works related to this we have that the error is not given by the hipothetycall characterization preprocessing task inefficiency (Galán et. al, 2008), but it is indeed given by the simmilarity between the EEG channels and the noisy nature of EEG non-invasive signals.

4. Conclusions

For the conclusions we have the following hypothesis and some analysis statements to take into account for future works and for possible implementations.

- The nature of the EEG signals, does not provide an easy approach to make a possible characterization or preprocessing task for any kind of classifier. However if we can use any methodology that makes upon the signals drawbacks and the limitation of the probe scenarios, such the external interference and the diagnosis of some patients in a real probe and rehab process.
- Definetely the non-linear classifiers are better than the linear ones, because the decision region form and the robustness of the decision rule for the last ones. The training and the testing errors are critical for this case because the preprocessin should put on continous evaluation, to find the better wavelet level and decomposition mother wavelet level to make a better characterization.
- We have to take into account the memory requirements and the computational task, for the preprocessing and postprocessing to improve the classification, with less data, the best classification performance and portability, taking into account the current algorithms that have been used for BCI interfaces.

References

Galán, F., Nuttin, M., Lew, E., Ferrez, P. W., Vanacker, G., Philips, J. & Millán, J. R. (2008). A brain-actuated wheelchair: asynchronous and non-invasive brain–computer interfaces for continuous control of robots. *Clinical Neurophysiology*, 119, (9), 2159–2169.

Hinterberger, T., Kübler, A., Kaiser, J., Neumann, N. & Birbaumer, N. (2003). A brain–computer interface (bci) for the locked-in: comparison of different eeg classifications for the thought translation device. *Clinical Neurophysiology*, 114, (3), 416–425.

Homan, R. W., Herman, J. & Purdy, P. (1987). Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology*, 66, (4), 376–382.

Li, Y., Guan, C., Li, H. & Chin, Z. (2008). A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system. *Pattern Recognition Letters*, 29, (9), 1285–1294.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F. & Arnaldi, B. (2007). A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4.

Mallat, S. (1999). *A wavelet tour of signal processing*. Academic press.

Millan, J. & Chavarriaga, R. (2011). Tutorial: brain mediated human-robot interaction. *Proceedings of the 6th international conference on Human-robot interaction*, 1–2. ACM.

Millan, J., Pierre, W. F., Galán, F., Lew, E. & Chavarriaga, R. (2008). Non-invasive brain-machine interaction. I*nternational Journal of Pattern Recognition and Artificial Intelligence*, 22, (05), 959–972.

Qiu, W., Fung, K. S. A., Chan, F. H. Y., Lam, F. K., Poon, P. W. F. & Hamernik, R. P. (2002). Adaptive filtering of evoked potentials with radial-basis-function neural network prefilter. *Biomedical Engineering, IEEE Transactions on*, 49, (3), 225–232.

Subasi, A. & Gursoy, M. I. (2010). Eeg signal classification using pca, ica, lda and support vector machines. *Expert Systems with Applications*, 37, (12), 8659–8666.