

# A new linear classification method for an EEG-based Brain-Computer Interface

Dr. Alois Schlögl  
 <a.schloegl@ieee.org>  
 June 2001, Graz, Austria

## **Abstract:**

For an EEG-based Brain-Computer-Interface (BCI) is a new linear classification system proposed. The proposed method considers different variance-covariances for each class and, therefore, it has an advantage against Linear Discriminant Analysis.

## **Introduction**

Linear classification methods, like Linear Discriminant Analysis (LDA), need less examples in order to obtain a reliable classifier (Pfurtscheller et al. 2000). In the past, LDA was used, because LDA is able to produce an output which is continuous in time as well as in amplitude. LDA was also successfully applied to many different EEG parameters, like bandpower values (ERD), as well as common spatial patterns (CSP) (Pfurtscheller et al. 2000) and Adaptive autoregressive (AAR) parameters (Schlögl, 2000).

However, sometimes the LDA output seemed to be biased towards one class. This is surprising because LDA provides the weights for the “best” linear separation of the data. After more detailed analysis, some results (unpublished) have suggested, that the different variability for each class causes this bias in the LDA output. This report describes in detail an alternative linear classification method without a bias due to different variability.

## **Linear discriminant analysis (LDA)**

At the beginning lets remind the analysis with linear discriminants. Lets assume each data element  $s_i$  has  $m$  features. Then, an element  $s_i$  is one point in an  $m$ -dimensional feature space. The number of examples is  $n$ , each example is assigned to one out of two classes  $C=\{0,1\}$ ; Then,  $S$  is a matrix of size  $n \times m$ , and  $C$  is a vector of size  $n$ .  $N_0$  and  $N_1$  are the number of elements for class 0 and 1, respectively.

The mean  $\mu_c$  of each class  $c$  is the mean over all  $s_i$  with  $i$  being all elements with in class  $c$ . The total mean  $\mu$  of the data is

$$\mu = (N_0 * \mu_0 + N_1 * \mu_1) / (N_0 + N_1) \quad (1.1)$$

The covariance matrix  $C$  of the data is the expectation value for

$$C = E \langle (s - \mu)(s - \mu)^T \rangle \quad (1.2)$$

Then, the weight vector  $w$  and the offset  $w_0$  are

$$w = C^{-1} \cdot (\mu_1 - \mu_0)^T \quad (2.1)$$

$$w_0 = -\mu \cdot w \quad (2.2)$$

The weight vector  $w$  determines a separating hyperplane in the  $m$ -dimensional feature space. The normal distance  $D(x)$  of any element  $x$  is

$$D(x) = x \cdot w + w_0 = \quad (3.1)$$

$$= (x - \mu) \cdot w = \quad (3.2)$$

$$= (x - \mu) \cdot C^{-1} \cdot (\mu_1 - \mu_0)^T \quad (3.3)$$

If  $D(x)$  is larger than 0,  $x$  is assigned to class 1, if  $D(x)$  is smaller than 0  $x$  is assigned to class 0.  $D(x)=0$  defines all elements  $x$  that are part of the separating hyperplane.

### ***Mahalanobis distance based classifier (MDBC)***

Lets assume the data elements  $s$  have dimension  $m$ . We can say, an element  $s_i$  is one point in an dimensional feature space. The number of examples is  $n$ , each example is assigned to one of two classes  $C=\{0,1\}$ ; Then,  $S$  is a matrix of size  $n \times m$ , and  $C$  is a vector of size  $n$ .  $N_0$  and  $N_1$  are the number of elements for class 0 and 1, respectively.

The mean  $\mu_c$  of each class  $c$  is the mean over all  $s_i$  with  $i$  being all elements with in class  $c$ . The covariance matrix  $C_c$  for each class  $c$  is the expectation value for

$$C_c = E\langle (s-\mu_c)^T \cdot (s-\mu_c) \rangle \quad (4.1)$$

It can be calculated as the mean over all elements  $s_i$  within the class  $c$ .

Now, the mean  $\mu_c$  and the covariance  $C_c$  determine the multivariate normal probability density function (pdf) that corresponds to class  $c$ . Now, any point in the  $n$ -dimensional features space can be associated with a certain distance to each class  $c$ . Because, we have assume a multivariate normal distribution  $N(\mu_c, C_c)$ , the Mahalanobis distance (Mahalanobis, 1936) is an appropriate distance function.

The mahalanobis distance  $d_c$  of some point  $x$  in the feature space to the multivariate normal distribution  $N(\mu_c, C_c)$  is defined by the following equation

$$d_c^2(x) = (x-\mu_c) \cdot C_c^{-1} \cdot (x-\mu_c)^T \quad (5.1)$$

Furthermore, the differences of the distances  $D(x) = d_1(x) - d_0(x)$  can be calculated;

$$D(x) = d_1(x) - d_0(x) = \quad (6.1)$$

$$= ((x-\mu_1) \cdot C_1^{-1} \cdot (x-\mu_1)^T)^{1/2} - ((x-\mu_0) \cdot C_0^{-1} \cdot (x-\mu_0)^T)^{1/2} \quad (6.2)$$

If  $D(x)$  is larger than 0,  $x$  is closer to the pdf of class 0, if  $D(x)$  is smaller than zero,  $x$  is closer to class 1. Lets say  $D(x)$  is a discrimination function based on the Mahalanobis distance, or MD-based discriminant. A MD-based classification (MDBC) is obtained if  $D(x)$  is applied to the threshold 0.

## **Discussion**

Both, LDA and MDBC, are based on the mean and the covariance of the data only. In other words, only second order statistics is used, both are linear methods, higher order statistics and non-linearity are not considered.

The main difference between LDA and MDBC is, that in case of LDA the Variance-covariance of the whole data is used, whereby in case of MDBC the different covariance-matrices are considered for each class. Therefore, MDBC takes into account the different variance-covariances. The separating hyperplane can become a hypercurve, with  $D(x)=0$ , in the  $n$ -dimensional feature space.

In both methods, LDA and MDBC, the largest computational effort is due to the matrix inversions. The computational effort is of order  $O(m^3)$ . However, the matrix inversion can be performed once during the offline analysis. Then, the inverted matrix can be used for the online classification. The order of the computational effort for the online classifier increases from a vector multiplication  $O(m)$  to a matrix multiplication  $O(m^2)$  for LDA to MDBC, respectively. As long  $m$  is limited to some hundreds of features, the computational effort is hardly an argument against MDBC.

## **Conclusion**

If the covariances are similar for both classes, the LDA and the MDBC will yield the same results. If the covariance matrices are different in each classes, MDBC takes this into account. Hence, the bias due to different variances in the data is removed with MDBC. Therefore, the MDBC classifier is preferable to LDA.

## **References**

P. C. Mahalanobis, *Proc. Natl. Institute of Science of India*, 2, 49, (1936)

Pfurtscheller G, Neuper C, Guger C, Harkam W, Ramoser H, Schlögl A, Obermaier B, Pregenzer M. Current trends in Graz Brain-Computer Interface (BCI) research. *IEEE Trans Rehabil Eng.* 2000 Jun;8(2):216-9.

A. Schlögl (2000), *The electroencephalogram and the adaptive autoregressive model: theory and applications*. Shaker Verlag, Aachen, Germany.