1 Pattern Recognition

Pattern recognition is a technique whose goal is classification and the principle by which it is performed is learned independently from the data, i.e., training set. There are two main types of pattern recognition: supervised and unsupervised. Supervised pattern recognition implies that the classes of the training set are known and are used to obtain the model. New inputs are identified as one of the predetermined classes. On the other hand, unsupervised pattern recognition is used when no labels are available and samples are assigned to unknown classes. This technique is more appropriate for the clustering problem because the classes are determined automatically by the system, whereas supervised approach is more appropriate for the classification because classes are defined by the system designer.

In statistical pattern recognition, each sample is composed of m measures that form the pattern, i.e., features $(x_0, x_1, \dots, x_{m-1})$. The goal of the algorithm is to obtain a decision rule, i.e., the decision boundary which separates well samples of different classes. There are many *state-of-the-art* classifiers that use various principles to construct these boundaries. However, many researchers agree that the fidelity of the classification in EMG applications depends mostly on selection of features. In other words, with appropriate selection of features, all classifiers will give similar classification result. Since pattern recognition is not the topic of this thesis, only a short introduction is provided on the two methods used in the thesis: Linear Discriminant Analysis and Support Vector Machine.

1.1 Linear Discriminant Analysis

All models are wrong; some models are useful.
George E. P. Box

Linear Discriminant Analysis is a computationally simple and efficient classifier with linear decision boundary and it is based on the Bayesian equation. In a classical problem with n samples in training set, which consist of m features, the dataset of available samples is a matrix of dimension $[n \times m]$, whereas the vector of labels that describe the belonging of each sample to one of the classes is y, where $y \in (0, 1, 2, ..., K-1)$.

According to Bayesian equation, the probability that a sample $\mathbf{x}_0$ belongs to a class k is equivalent to the:

$$P(y = k \mid \mathbf{x} = \mathbf{x}_0) = \frac{P(\mathbf{x} = \mathbf{x}_0 \mid y = k)\ P(y = k)}{P(\mathbf{x} = \mathbf{x}_0)} \tag{1}$$

,where k represents the class. Term $P(\mathbf{x} = \mathbf{x}_0 \mid y = k)$ is called the *class-conditional* probability and describes the probability that the sample with exact features $\mathbf{x}_0$ is encountered within the group of samples belonging only to the class k. Term $P(y = k)$ is called the *a priori* probability and describes the probability that the sample with class k is found within the group of all samples, regardless of the features. Finally, the term $P(\mathbf{x} = \mathbf{x}_0)$ is called *marginal* probability and describes the probability of finding the exact set of features in the dataset, regardless of the class. Marginal probability can be written as a sum of class-conditional probabilities multiplied by the a priori probabilities for each class:

$$\begin{aligned} P(\mathbf{x} = \mathbf{x}_0) = P(\mathbf{x} = \mathbf{x}_0 \mid y = 1)\ P(y = 1) + \\ P(\mathbf{x} = \mathbf{x}_0 \mid y = 2)\ P(y = 2) + \cdots + \\ P(\mathbf{x} = \mathbf{x}_0 \mid y = K)\ P(y = K) \end{aligned} \tag{2}$$

Following Bayesian theory, the hypothesis, i.e., the predicted class of a sample $\mathbf{x}_0$ is chosen as the class which has the highest probability $P(y = k \mid \mathbf{x} = \mathbf{x}_0)$:

$$h(\mathbf{x}_0) = \underset{k}{\operatorname{argmax}}\, P(y = k \mid \mathbf{x} = \mathbf{x}_0) \tag{3}$$

Statistically speaking, this is the best possible classifier. The problem arises in the implementation. The exact probability density functions are unknown and have to be estimated from the available data, which is the source of error. Estimated version of the stated probabilities will be marked with a different symbols to stress out the fact they are just an estimates:

$$p_k(\mathbf{x}) \coloneqq P(y = k \mid \mathbf{x}) \tag{4}$$

$$g_k(\mathbf{x}) \coloneqq P(\mathbf{x} \mid y = k) \tag{5}$$

$$\pi_k \coloneqq P(y = k) \tag{6}$$

Linear Discriminant Analysis estimates marginal probability term (π_k) as a ratio of number of samples belonging to class k and the total number of samples, whereas the class-conditional probability term in the Bayesian equation is estimated as a Gaussian function:

$$g_k(\mathbf{x}) = \frac{1}{(2\pi)^{m/2}\,|\Sigma_k|^{1/2}}\; e^{-1/2(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k)} \tag{7}$$

, where m is the dimensionality of the feature space, i.e., number of features representing each sample. Function g_k is estimated class-conditional probability of class k, and μ_k and Σ_k are the mean and co-variance matrix for class k, respectively, and they are estimated from the available data as:

$$\mu_k = \frac{1}{n_k}\sum_i \mathbf{x}_i \Bigg|_{\forall \mathbf{x} \in k} \tag{8}$$

$$\Sigma_k = \frac{1}{n_k - K}\sum_i \Big(\mathbf{x}_i - \mu_k\Big)\Big(\mathbf{x}_i - \mu_k\Big)^T \Bigg|_{\forall \mathbf{x} \in k} \tag{9}$$

, where n_k represents the number of samples belonging to a class k. To simplify the model, LDA assumes that the co-variance matrices Σ_k are the same for all classes:

$$\Sigma_0 = \Sigma_1 = \cdots = \Sigma_{K-1} = \Sigma \tag{10}$$

and they are usually calculated using the weighted average:

$$\Sigma = \frac{\sum_{k=1}^{K} n_k \Sigma_k}{\sum_{k=1}^{K} n_k} \tag{11}$$

The consequence of this assumption is the linearity of the decision boundary. Without this assumption the same calculus would lead to quadratic discriminant analysis, which has non-linear boundary.

In a two class example ($y \in \{0, 1\}$), all samples on the decision boundary will have the same probability of belonging to class 0 or 1:

$$D.B. = \Big\{\mathbf{x} \;\Big|\; P\big(y = 0 \,\big|\, \mathbf{x} = \mathbf{x}_0\big) = P\big(y = 1 \,\big|\, \mathbf{x} = \mathbf{x}_0\big)\Big\} \tag{12}$$

Following this idea, the decision boundary can be estimated by solving the equation:

$$\frac{g_0(\mathbf{x})\,\pi_0}{\cancel{\sum_{k=1}^{K} g_k\,\pi_k}} = \frac{g_1(\mathbf{x})\,\pi_1}{\cancel{\sum_{k=1}^{K} g_k\,\pi_k}} \tag{13}$$

$$\frac{1}{(2\pi)^{d/2}\,|\Sigma_0|^{1/2}}\; e^{-1/2(\mathbf{x}-\mu_0)^T\Sigma_0^{-1}(\mathbf{x}-\mu_0)}\;\;\pi_0 = \frac{1}{(2\pi)^{d/2}\,|\Sigma_1|^{1/2}}\; e^{-1/2(\mathbf{x}-\mu_1)^T\Sigma_1^{-1}(\mathbf{x}-\mu_1)}\;\;\pi_1 \tag{14}$$

If making the assumption on the equal co-variance matrices for both classes:

$$\Sigma_0 = \Sigma_1 = \Sigma \tag{15}$$

and taking the logarithm, the equation takes the form:

$$-\frac{1}{2}\Big(\mathbf{x} - \mu_0\Big)^T \Sigma^{-1}\Big(\mathbf{x} - \mu_0\Big) + \log\Big(\pi_0\Big) = -\frac{1}{2}\Big(\mathbf{x} - \mu_1\Big)^T \Sigma^{-1}\Big(\mathbf{x} - \mu_1\Big) + \log\Big(\pi_1\Big) \tag{16}$$

This equation can be written in the form of the linear function $x^T\beta + \alpha = 0$ as:

$$\mathbf{x}^T\Big(\Sigma^{-1}\mu_0 - \Sigma^{-1}\mu_1\Big) + \frac{1}{2}\Big(\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0\Big) + \log\Big(\frac{\pi_0}{\pi_1}\Big) = 0 \tag{17}$$

The equation represents the decision boundary between two classes, i.e., all samples lying on this line will have equal probability of belonging to class 0 and class 1. It should be noted that the slope of the line depends only on the class means and co-variance matrix, whereas a priori probabilities (which are the result of number of samples belonging to class 0 or 1) have effect only on the y-intercept term, i.e., the offset of the function. This is an interesting point that demands caution. If groups are unbalanced, that is, number of samples of one group is higher than in the other group, y-intercept of the decision boundary will be affected and the classifier will be biased by this disproportion. If groups are unbalanced because of the incomplete or missing data, whereas in reality they are balanced, this can have a negative effect.

When considering multiclass classification problem, probability of a sample belonging to each class is firstly estimated by the equation:

$$p_k = -\frac{1}{2}\log\big|\Sigma\big| - \frac{1}{2}\Big(\mathbf{x} - \mu_k\Big)^T\Sigma^{-1}\Big(\mathbf{x} - \mu_k\Big) + \log\Big(\pi_k\Big) \tag{18}$$

and then the class is estimated as the one with the highest probability as:

$$h(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\, p_k(\mathbf{x}) \tag{19}$$

1.2 Support Vector Machine

> *Try to solve the problem directly and never solve a more general problem as an intermediate step.*
>
> Vladimir Vapnik

Support vector machine is nowadays known as a very powerful classifier with a lot of different applications. The big advantage over LDA is the fact that it is a *non-parametric* classifier. The model is not obtained using assumptions of the form of the class density function and estimation of it's parameters, which is inevitably erroneous. Instead, SVM forms the decision boundary using the samples (not their density estimates) by maximizing the distance between samples and the boundary. This was the idea Vladimir Vapnik, the inventor of this method stood for. It is better to try to solve the problem directly and simply, without many intermediate steps that can be complicated and inaccurate.

In pattern recognition, the decision rule (h) is usually obtained by multiplying the sample ($\mathbf{x}$) by predefined weights (Θ):

$$\Theta^T\mathbf{x} + \Theta_0 \tag{20}$$

, where Θ_0 is a constant. If samples $\mathbf{x}_0$ and $\mathbf{x}_1$ lay on the decision boundary, following statements are true:

$$\Theta^T\mathbf{x}_0 + \Theta_0 = \Theta^T\mathbf{x}_1 + \Theta_0 \tag{21}$$

$$\Theta^T(\mathbf{x}_0 - \mathbf{x}_1) = 0 \tag{22}$$

This result implies that Θ is perpendicular to the boundary:

$$\Theta \perp (\mathbf{x}_0 - \mathbf{x}_1) \tag{23}$$

The goal of the SVM is to find the decision boundary so that the distance between the samples and the decision boundary, i.e., the margin is maximized. The distance (d) from a sample to the decision boundary can be defined as the distance between the sample $\mathbf{x}$ and any point lying on the boundary, $\mathbf{x}_0$, projected onto the vector Θ.

$$d = \frac{\Theta^T \left(\mathbf{x} - \mathbf{x}_0\right)}{|\Theta|} \tag{24}$$

Term $|\Theta|$ is introduced to normalize the vector Θ. Without this normalization the distance would depend on the norm of Θ.

Since $\mathbf{x}_0$ is on the decision boundary, the expression $\Theta^T \mathbf{x}_0 + \Theta_0 = 0$ is valid, and, therefore, the expression for the distance can be written as:

$$d = \frac{\Theta^T \mathbf{x} + \Theta_0}{|\Theta|} \tag{25}$$

Margin (M) can be defined as the distance from the boundary to the closest sample:

$$M = \min_i d_i \tag{26}$$

Depending on which side of the boundary the sample is located, the distance can be positive or negative. In order to keep it strictly positive, term y is introduced, where $y \in \{-1, 1\}$:

$$M = \min_i \left\{ y_i d_i \right\} \tag{27}$$

$$M = \min_i \left\{ \frac{y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right)}{|\Theta|} \right\} \tag{28}$$

The objective is to maximize the margin M. Since Θ can be rescaled, a certain Θ exists so that $y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) = 1$, which implies

$$\exists \Theta,\ y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) = 1 \quad \Rightarrow \quad M = \min_i \left\{ \frac{1}{|\Theta|} \right\} \tag{29}$$

Therefore, to maximize the margin, a hyperplane, that is, a decision boundary should be found such that a norm of vector orthogonal to the hyperplane (Θ) is minimal.

For every point not on the boundary the following term is valid:

$$y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) > 0 \tag{30}$$

Value C can be selected such that:

$$y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) > C \tag{31}$$

$$y_i \left(\frac{\Theta^T \mathbf{x}_i}{C} + \frac{\Theta_0}{C} \right) > 1 \tag{32}$$

Since Θ and Θ_0 can be rescaled, it can be written:

$$\Theta := \frac{\Theta}{C}, \quad \Theta_0 := \frac{\Theta_0}{C} \tag{33}$$

$$y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) > 1 \tag{34}$$

Finally the optimization problem states:

$$\min \frac{1}{2} |\Theta|^2, \quad s.t.\ y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) > 1. \tag{35}$$

L_2 norm is preferred because it has continuous derivative, whereas constant $1/2$ is introduced for the mathematical convenience. The optimization problem is solved using Lagrangian method as:

$$L(\Theta, \Theta_0, \alpha_i) = \frac{1}{2} |\Theta|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i \left(\Theta^T \mathbf{x}_i + \Theta_0\right) - 1 \right] \tag{36}$$

$$\frac{\partial L}{\partial \Theta} = \Theta - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \quad \Rightarrow \quad \Theta = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{37}$$

$$\frac{\partial L}{\partial \Theta} = \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{38}$$

By rewriting the problem in 36 in terms of dual variable α, the following expression can be obtained:

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_j \sum_i \alpha_j \alpha_i y_j y_i \mathbf{x}_i^T \mathbf{x}_j \tag{39}$$

Since this is the function with single variable α, the solution can be obtained by maximizing it:

$$\max L(\alpha) \quad s.t. \quad \begin{cases} \alpha_i \geq 0 \\ \sum_i \alpha_i y_i = 0 \end{cases} \tag{40}$$

In this optimization problem, the objective has the form of quadratic function, whereas constraints are linear. This problem is typically solved using quadratic programming. Since it is a convex problem, the solution has global maximum and the algorithm can not stuck in the local solution. By solving it, solution for dual variable α can be found, and then primal variable Θ can be found using equation 37.

In the optimization, Karush-Kuhn-Tucker conditions need to be satisfied citepBoyd2004. One of this condition is *complementary slackness*, stating that in the optimal point product of dual variable and the constraint must be zero:

$$\alpha_i \left[y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) - 1 \right] = 0 \tag{41}$$

This condition is interesting in order to better understand the principle of SVM. Since the dual variable must greater or equal to zero ($\alpha \geq 0$), there are two possibilities:

1. If α is greater than zero, $\left[y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) - 1 \right]$ must equal one:

$$\alpha_i > 0 \quad \Rightarrow \quad y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) = 1 \tag{42}$$

2. If $\left[y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) - 1 \right]$ is greater than zero, α must be zero:

$$y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) > 1 \quad \Rightarrow \quad \alpha = 0 \tag{43}$$

Since for all samples lying on the margin, the statement

$$y_i \left(\Theta^T \mathbf{x}_i + \Theta_0 \right) = 1 \tag{44}$$

holds, α will be different from zero only for the samples lying on the decision hyperplane, whereas for the samples further away from the hyperplane, α will be zero and this points will have no effect on the model. This is especially useful for the outlier samples.

It is important to note that weights Θ depend on the linear combination of the samples. The optimization problem does not depend on $\mathbf{x}$, but on $\mathbf{x}^T \mathbf{x}$. This enables the use of *kernel trick* and allows nonlinear transform of the feature space at no additional cost. Non-linear decision boundary can be achieved by nonlinear transform of features:

$$\mathbf{x} \rightarrow \Phi(\mathbf{x}) \tag{45}$$

However, this operation is computationally expensive. The solution can be achieved using kernel. Kernel is a function $K(x, y)$ for which:

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y}) \tag{46}$$

Since in the equation 38 $\mathbf{x}$ does not appear by itself, but in a form of dot product $\mathbf{x}^T \mathbf{x}$, non-linear transform can be used in a form of kernel trick:

$$L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_j \sum_i \alpha_j \alpha_i y_j y_i K(\mathbf{x}_i, \mathbf{x}_j) \tag{47}$$

Most often used kernel is radial basis kernel ($K_{RBF}(\mathbf{x}_i, \mathbf{x}_j)$):

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{48}$$