Image Analysis
Exercise - Advanced segmentation
Fisherman's Linear discriminant analysis for segmentation

Introduction

The exercise is to the extent of the pixel-wise classification problem from being based on the intensity histogram of a single image modality to combining multiple image modalities. Hence, here we wish to segment image features into two classes by training a classifier based on the intensity information from multiple image modalities.
Multiple-image modalities just mean a series of images that contain different but complementary image information of the same object. It is assumed that the image modalities have the same size, so we have a pixel-to-pixel correspondence between the two images. An image feature is an identifiable object in the image e.g., of a dog, moon rocket, or brain tissue types that we wish to segment into individual classes.

Here we aim to segment two types of brain tissues into two feature classes. To improve the segmentation, we wish to combine two MRI image modalities that contain different intensity information of the feature classes of interest: one is a "T1 weighted MRI" and the other is a "T2 weighted MRI". Both are acquired at the same time and are overlapping.

Get the data from **courses.compute.dtu.dk/02502**.

Exercise - You simply go step-by-step and fill in the command lines and answer/discuss the questions (Q1-Q12).

Theory

The Linear Discriminate Classifier

As a classifier, we will use a class of linear discriminate functions that aims to place a hyperplane in the multi-dimensional feature space that acts as a decision boundary to segment two features into classes. Since we only look at image intensities of two image modalities our multi-dimensional feature space is a 2D intensity histogram. The linear discriminant classifier is based on the Bayesian theory where the posterior probability is the probability of voxel $\boldsymbol{x}$ belonging to class $\boldsymbol{C_i}$. The voxel x belongs to the class with the highest posterior probability. A class posterior probability is expressed by Bayes:

$$P(Ci|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i},)\boldsymbol{P(C_i)}}{P(\boldsymbol{x})} \quad \text{[Eq. 1]}$$

Where

- $P(x|\mu_i, \Sigma_i,) = K_i\, exp((x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))$ is the conditional probability. We assume that the distribution of data for a feature can be modeled using a parametric Gaussian model in multiple dimensions i.e.: $\mathcal{N}(\mu_i, \Sigma_i)$
- $P(C_i)$ is the prior probability for each class and $\sum_i P(\boldsymbol{C_i}) = 1$
- $P(x)$ is the marginal probability and is $\sum_i P(\boldsymbol{C_i}|x)$.

In Eq. 1 we need to train the model using training examples representative of the distribution of features to be segmented. The training examples we often draw ourselves or are provided by an expert. Given the training examples, we train the model parameter which are the means, covariances, and prior probabilities for each class.

The linear discriminant function for two classes is based on the log-ratio between the two posterior probabilities:

$$\text{Log}(\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})}) > \log(T) \text{ [Eq. 2]}$$

Note, we take the log() on both sides as a trick to make the expressions easier to realize the model practically but then remember: To calculate the final class posterior probabilities once the classifier model has been trained one needs to account for the log(P(Ci|x)).

We derive the model based on the expression in Eq 2 and its model assumptions that define the model. Firstly, we assume homoscedasticity of variances meaning that the variance of the covariance matrix of each feature is the same for all classes ($\Sigma_1 = \Sigma_2 = \Sigma_0$). Further, we assume to have isotropic covariances i.e., no preferred direction in the variance meaning they have round shape distribution in 2D as illustrated in Figure 1. Hence, covariances in the off-diagonal of the covariance matrix are zero and we have equal variances in the diagonal.

With these assumptions, Eq. 2 is a Linear Discriminant Analysis (LDA) classifier and can be expressed by the means and the covariances of the two gaussian distributions and their prior probabilities:

$$\log\left(\frac{P1}{P2}\right) - \frac{1}{2}(\boldsymbol{\mu_2} + \boldsymbol{\mu_1})^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) + \boldsymbol{x}^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) > \log(T) \text{ [Eq. 3]}$$

The expression can look a bit complicated at first glimpse but if we add colors a general structure of the expression appears. The green part $\Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$ we name a weight vector, $\boldsymbol{w}$ which is normal to the hyperplane, and along $\boldsymbol{w}$ the decision boundary separating the two classes is to be placed. We can say that $\boldsymbol{w}$ determines how the hyperplane is orientated in the coordinate system to best separate the two classes. In other words, $\boldsymbol{w}$ is our model like in Eq. 3. Remember, it is defined by its assumptions and therefore it is not guaranteed that it is the best solution if the data do not agree with the assumptions. The orange part $\log\left(\frac{P1}{P2}\right) - \frac{1}{2}(\boldsymbol{\mu_2} + \boldsymbol{\mu_1})^T$ we call the bias $\boldsymbol{c}$ which describes where the hyperplane hence the decision boundary along the $\boldsymbol{w}$ is placed. Basically, the expression for $\boldsymbol{c}$ says that the decision boundary is placed halfway between the means of the two class distributions weighted with the ratio of the prior probability for each class. So, it is halfway between the class means if the two classes have equal prior probabilities, $P(C_i)$ (i.e., 0.5). In 1D and in this case, it is the same as a parametric classifier where we define a threshold at the point where the two distributions cross having equal probabilities.

We can reformulate Eq. 3 in a general way:

$$\boldsymbol{y_{C\in 2}(x)} = \boldsymbol{x^T w} + \boldsymbol{w_o} \text{ ; where } \boldsymbol{w_o} = \boldsymbol{c\,w} \text{ [Eq. 4]}$$

The weight vector **w** is multiplied both with the vector of a data point $\boldsymbol{x}$ and with the vector of the bias $\boldsymbol{c}$. Multiplying two vectors is the same as the dot-product between two vectors i.e., $\boldsymbol{ab} = \|\boldsymbol{a}\|\|\boldsymbol{b}\| \cos\theta$. Hence, the function of the weight vector is to rotate the hyperplane in feature space with an angle that best separates the two classes. Therefore, when $\boldsymbol{w}$ is multiplied with the bias $\boldsymbol{c}$ it is projected along $\boldsymbol{w}$ and defines the decision boundary at $\boldsymbol{w_o}$. When multiplied with a data point vector $\boldsymbol{x}$ it projects the data point along $\boldsymbol{w}$ and whether the projected data point belongs to class 2 depends on its projected position in relation to the decision boundary. Figure 1 illustrates the LDA model and two gaussian classes with equal isotropic variances - as the model assumption subscribes - the principle of how the weight vector projects the hyperplane and decision boundary independent of the orientation between the two class means and the coordinate system Figure 1 (A vs B).

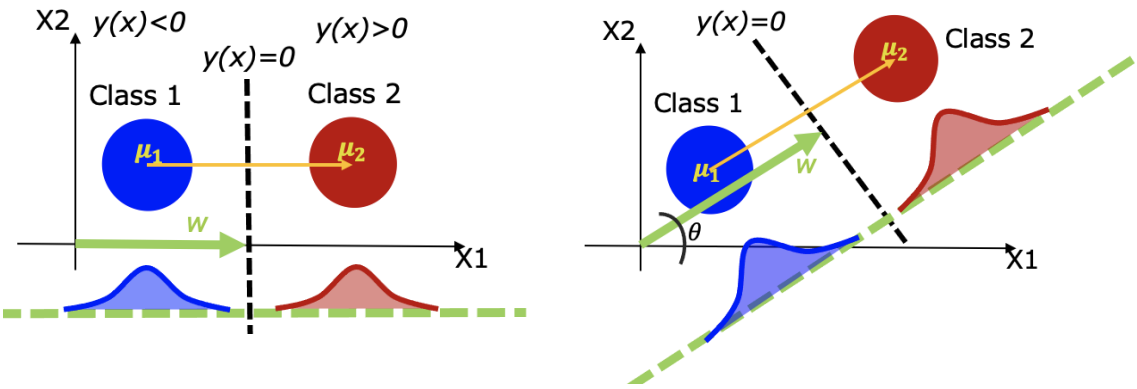

Figure 1 Principle how the LDA model can be explained by using a weight vector ***w*** that is normal to the hyperplane. The weight vector projects the hyperplane to define the decision boundary (***c***'***w***) (striped line) that separates the two classes. Data points are likewise projected along ***w***. Left: the hyperplane follows the coordinate system. Right: The hyperplane is not aligned with the coordinate system and has a rotation angle.

In summary, to classify if a point ***x*** belongs to class 2 using Eq. 4 we multiply it with ***w*** which projects the point into the same orientation as the hyperplane for maximal class separation and then we subtract the bias. If $\boldsymbol{y_{C\in 2}(x)} > \log(T)$ we have a score defining that the projected data point is closer to class 2 than class 1 and is classified as class 2 (Note, log(*T*) = 0 in Eq 3). Actually, Eq. 4 only states a score if a data point is belonging to class C2 i.e. $\boldsymbol{y_{C\in 2}(x)}$, but we can inverse Eq. 2 to define a model for belonging to C1 i.e., $y_{C\in 1}(x)$ by defining a weight vector and bias for class 1.
When knowing the weight vectors and biases for each class we can from the scoring in Eq. 4 calculate the class probability of a point belonging to a class. Since y(x) are log-values we take the exp(y(x)) for a given class and divide it by the marginal probability P(x) i.e., the denominator in Eq. 1. The P(x) is calculated for each data point as the sum of class probabilities. Example, for the first data point $\boldsymbol{x_1}$: $P(\boldsymbol{x_1}) = \sum_{i=1}^{c} P(\boldsymbol{x_1}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_0},)\boldsymbol{P(C_i)}$.

The Fisher's linear discriminant classifier
The LDA classifier model assumptions are not correct if the variances of the two gaussian distributions are not equal or isotropic. In this case, the hyperplane of the LDA is not guaranteed to optimally separate the two classes as illustrated in Figure 2A. However, if changing the model assumptions to account for non-equal and non-isotropic class variances, the projections of the hyperplane, and decision boundary will ensure a better class separation as shown in Figure 2B. This will result in more precise classifications as shown in Figure 2(Right vs Left).

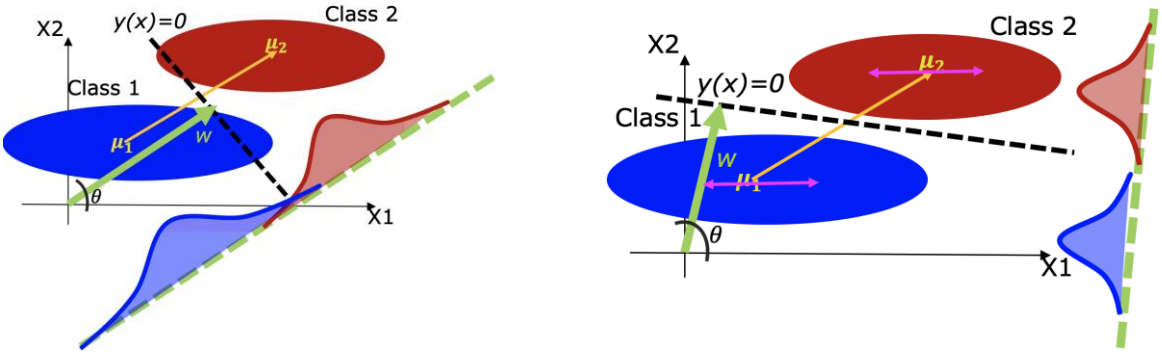

Figure 2 Illustration that the two class distributions have different covariances and are non-isotropic not assumed by the LDA model that risk fails to place the hyperplane for optimal class separation (Left). Right: If we make a new model i.e., Fisher's LDA assuming that covariances can be different and non-isotropic the hyperplane will be placed for optimal class separation.

Fisher's discriminate classifier assumptions account for non-equal and non-isotropic class variances by taking the ratio of *between-class covariance* and the *total within covariance*. The weight vector is to project the hyperplane, so it minimizes the ratio between the two. For this, we can express a cost function to find $\boldsymbol{w}$. The between-class covariance is $S_B = (\boldsymbol{\mu_2} - \boldsymbol{\mu_1})^T(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$ and the total within-covariance is $S_w = \boldsymbol{\Sigma_1} + \boldsymbol{\Sigma_2}$**.** Now we search for a solution for a weight vector and bias that minimizes the ratio between the two covariances.

$$\boldsymbol{J(w)} = \frac{\boldsymbol{w^T S_B w}}{\boldsymbol{w^T S_w w}}$$

This defines the cost function that we differentiate and set to zero to find an optimal solution for $\boldsymbol{w}$:

$$\frac{\boldsymbol{\partial j(w)}}{\boldsymbol{w}} = \boldsymbol{0}$$

The weight vector, $\boldsymbol{w}$**,** now based on Fisher's discriminate classifier model expresses $w = S_w^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$ and is used in Eq 4. The bias, $\boldsymbol{c}$, is the same as for the LDA as we still wish the decision boundary to be placed halfway between the two class means. Note, the Fisher's model assumptions allow both equal and unequal class covariances, as well as isotropic and non-isotropic covariances, hence extending the model capabilities compared with the LDA.
When knowing the expression of $\boldsymbol{w}$ and $\boldsymbol{wo}$ we can use Fisher's linear discriminant classifier to classifier if a sample belongs to a class or not as well as to estimate the class probability in the same way as described for the LDA.