Linear Discriminant Analysis - cs534

Given training set $\{(\mathbf{x}_i, y_i) : i = 1, \cdots, N\}$, $y_i \in \{0, 1\}$, and $\mathbf{x}_i \in R^d$. We aim to learn $p(\mathbf{x}, y)$. Specifically, we factorize $p(\mathbf{x}, y) = p(\mathbf{x}|y)p(y)$, where $p(y)$ is the prior distribution of y. We will denote $p(y = 1) = \pi$. We further make the simplifying assumption that $p(\mathbf{x}|y) = N(\mathbf{x}|\mu_y, \Sigma)$. That is, we assume that the data from class 0 and class 1 come from two different Gaussians, each with distinct mean but shared covariance. This is the basic assumption behind Linear discriminant analysis.

1 Parameter Estimation

There are two problems we need to solve. First, the learning problem — given the training set, we need to learn the parameters to fully specify the joint distribution $p(\mathbf{x}, y)$, which includes π, μ_0, μ_1 and Σ. We will apply maximum likelihood estimation for this. The likelihood function is as follows.

$$\begin{aligned}
\log P(D|M) &= \sum_{i=1}^{N} \log p(\mathbf{x_i}, y_i) \\
&= \sum_{i=1}^{N} \log\{[\pi \cdot N(\mathbf{x}_i|\mu_1, \Sigma)]^{y_i} [(1-\pi) \cdot N(\mathbf{x}_i|\mu_0, \Sigma)]^{1-y_i}\} \\
&= \sum_{i=1}^{N} \{y_i \log \pi + y_i \log N(\mathbf{x}_i|\mu_1, \Sigma) + (1 - y_i) \log(1 - \pi) + (1 - y_i) \log N(\mathbf{x}_i|\mu_0, \Sigma)\}
\end{aligned}$$

Let's first estimate π. Consider the parts that contain π, we have:

$$\sum_{i=1}^{N} \{y_i \log \pi + (1 - y_i) \log(1 - \pi)\}$$

Take the derivative over π:

$$\frac{1}{\pi} \sum_{i=1}^{N} y_i - \frac{1}{1-\pi} \sum_{i=1}^{N} (1 - y_i)$$

Setting it to zero and let $N_1 = \sum_{i=1}^{N} y_i$ and $N_1 = \sum_{i=1}^{N} (1 - y_i)$, we have:

$$\begin{aligned}
\frac{N_1}{\pi} &= \frac{N_2}{1-\pi} \\
\pi &= \frac{N_1}{N_1 + N_2}
\end{aligned}$$

We now move onto estimating μ_1 (μ_0 is exactly the same). Consider the parts that contain μ_1, we have:

$$\begin{aligned}
\sum_{i=1}^{N} y_i \log N(\mathbf{x}_i|\mu_1, \Sigma) &= \sum_{i=1}^{N} y_i \frac{-(\mathbf{x}_i - \mu_1)^T \Sigma^{-1} (\mathbf{x}_i - \mu_1)}{2} + const \\
&= \sum_{y_i=1} \frac{-(\mathbf{x}_i - \mu_1)^T \Sigma^{-1} (\mathbf{x}_i - \mu_1)}{2} + const
\end{aligned}$$

Take the derivative over μ_1 and set it to zero, we have:

$$\sum_{y_i=1} \Sigma^{-1}(\mathbf{x}_i - \mu_1) = 0$$

$$\mu_1 = \frac{1}{N_1} \sum_{y_i=1} \mathbf{x}_i$$

Similarly we have:

$$\mu_1 = \frac{1}{N_2} \sum_{y_i=0} \mathbf{x}_i$$

Finally, we will estimate Σ. Taking the part that contains Σ, we have:

$$-\frac{N_1}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{N} y_i(\mathbf{x}_i-\mu_1)^T\Sigma^{-1}(\mathbf{x}_i-\mu_1) - \frac{N_2}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(1-y_i)(\mathbf{x}_i-\mu_0)^T\Sigma^{-1}(\mathbf{x}_i-\mu_0)$$

$$\begin{aligned}
&= -\frac{N}{2}\ln|\Sigma| - \frac{N_1}{2}Tr(\Sigma^{-1}S_1) - \frac{N_2}{2}Tr(\Sigma^{-1}S_2) \\
&= -\frac{N}{2}\ln|\Sigma| - \frac{N}{2}Tr(\Sigma^{-1}(\frac{N_1}{N}S_1 + \frac{N_2}{N}S_2)) \\
&= -\frac{N}{2}\ln|\Sigma| - \frac{N}{2}Tr(\Sigma^{-1}S)
\end{aligned}$$

Taking derivative over Σ and set it to zero, we have:

$$\Sigma = S = \frac{N_1}{N}S_1 + \frac{N_2}{N}S_2$$

2 Decision Boundary

The second problem that we need to solve is the inference problem — given $\mathbf{x}$, we need to infer its y value, aka, the prediction problem. Recall from our previous lectures that to minimize the probability of misclassifying a given example $\mathbf{x}$, we predict y to be the value that maximizes $p(\mathbf{x}, y)$. Thus, we can consider the ratio:

$$\frac{p(\mathbf{x}, y=1)}{p(\mathbf{x}, y=0)}$$

and predict 1 if this ratio is greater than 1. This is equivalent to predicting $y = 1$ if $\log\frac{p(\mathbf{x},y=1)}{p(\mathbf{x},y=0)} > 0$. Note that

$$\begin{aligned}
\log\frac{p(\mathbf{x}, y=1)}{p(\mathbf{x}, y=0)} &= \log\frac{\pi N(\mathbf{x}|\mu_1,\Sigma)}{(1-\pi)N(\mathbf{x}|\mu_0,\Sigma)} \\
&= \log\frac{\pi}{1-\pi} + \log\frac{N(\mathbf{x}|\mu_1,\Sigma)}{N(\mathbf{x}|\mu_0,\Sigma)} \\
&= \log\frac{\pi}{1-\pi} - \frac{1}{2}(\mathbf{x}-\mu_1)^T\Sigma^{-1}(\mathbf{x}-\mu_1) + \frac{1}{2}(\mathbf{x}-\mu_0)^T\Sigma^{-1}(\mathbf{x}-\mu_0) \\
&= \log\frac{\pi}{1-\pi} - \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} + \mu_1^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x} - \mu_0^T\Sigma^{-1}\mathbf{x} + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 \\
&= (\mu_1-\mu_0)^T\Sigma^{-1}\mathbf{x} - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \log\frac{\pi}{1-\pi} \\
&= \mathbf{w}^T\mathbf{x} + w_0
\end{aligned}$$

where $\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_0)$ and $w_0 = -\frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \log\frac{\pi}{1-\pi}$. This indicates that LDA learns a linear decision boundary.

Note that if we relax our modeling assumption such that the different classes have different covariance matrix, the above derivation will result in a quadratic decision boundary.

3 Dimension reduction view of LDA

As shown in the slides posted on class website, we can also arrive at a similar solution by seeking a projection vector $\mathbf{w}$ that maximizes the separation between the two classes. It turns out that $\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_0)$ is the optimal projection vector in this sense. We will skip this part at this point of the class and revisit when we discuss dimension reduction techniques.