Linear discriminant analysis

Linear discriminant analysis (LDA) is a common "plug-in" method for classification which operates by estimating $\pi_k f_{X|Y}(\boldsymbol{x}|k)$ for each class $k = 0, \ldots, K-1$ and then simply plugging these into the formula for the Bayes classifier in order to make a decision. In LDA we make the (strong) assumption that class conditional pdfs are given by the multivariate normal distribution, but with differing means. Mathematically, this corresponds to the assumption that

$$f_{X|Y}(\boldsymbol{x}|k) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)}$$

for $k = 0, \ldots, K-1$. Note that under this assumption, each class has a distinct mean $\boldsymbol{\mu}_k$, but all classes share the same covariance matrix $\boldsymbol{\Sigma}$.

In LDA, we assume that $\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_{K-1}$ and $\boldsymbol{\Sigma}$, as well as the prior probabilities $\pi_0, \ldots, \pi_{K-1}$ are all unknown, but can be estimated from the data. In particular, we can use the estimates

$$\begin{aligned}
\widehat{\pi}_k &= \frac{|\{i : y_i = k\}|}{n} \\
\widehat{\boldsymbol{\mu}}_k &= \frac{1}{|\{i : y_i = k\}|} \sum_{i:y_k=k} \boldsymbol{x}_i \\
\widehat{\boldsymbol{\Sigma}} &= \frac{1}{n} \sum_{k=0}^{K-1} \sum_{i:y_i=k} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_k)^T.
\end{aligned}$$

The LDA classifier is then defined by

$$\widehat{h}(\boldsymbol{x}) = \arg\max_k \ \widehat{\pi}_k \cdot \frac{1}{(2\pi)^{d/2}|\widehat{\boldsymbol{\Sigma}}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)^T\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)}.$$

Since the log is a monotonic transformation (meaning that if $x > y$ then $\log(x) > \log(y)$), we can equivalently state the classifier as

$$\begin{aligned}\widehat{h}(\boldsymbol{x}) = &\ \arg\max_k \log\left(\widehat{\pi}_k\right) + \log\left(\frac{1}{(2\pi)^{d/2}|\widehat{\boldsymbol{\Sigma}}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)^T\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)}\right) \\ &= \arg\max_k \ \log\left(\widehat{\pi}_k\right) - \frac{1}{2}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)^T\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k) \\ &= \arg\min_k \ \frac{1}{2}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k)^T\widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_k) - \log\left(\widehat{\pi}_k\right)\end{aligned}$$

where the second equality above follows from the fact that

$$\log\left(\frac{1}{(2\pi)^{d/2}|\widehat{\boldsymbol{\Sigma}}|^{1/2}}\right)$$

is constant across all k and so does not affect which k maximizes the expression.

It is enlightening to consider what happens in the special case of $K = 2$ (i.e., binary classification). In this case, LDA results in a classifier such that $\widehat{h}(\boldsymbol{x}) = 1$ when

$$(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_0) - 2\log\widehat{\pi}_0 \geq (\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_1) - 2\log\widehat{\pi}_1.$$

We can rewrite this as

$$(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_0)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_0) - (\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_1)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\widehat{\boldsymbol{\mu}}_1) + 2\log\frac{\widehat{\pi}_1}{\widehat{\pi}_0} \geq 0.$$

Using the fact that $\boldsymbol{\Sigma}$ is symmetric, which implies that we have

Georgia Tech ECE 6254 Spring 2024; Notes by M. Bloch, M. A. Davenport, and J. Romberg. Last updated 20:12, February 7, 2024

$(\boldsymbol{\Sigma}^{-1})^T = \boldsymbol{\Sigma}^{-1}$, we can simplify this rule to

$$\begin{aligned}
0 &\leq (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_0)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_0) - (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_1)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_1) + 2 \log \frac{\widehat{\pi}_1}{\widehat{\pi}_0} \\
&= \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - 2\widehat{\boldsymbol{\mu}}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} + \widehat{\boldsymbol{\mu}}_0^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_0 \\
&\quad - \left(\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - 2\widehat{\boldsymbol{\mu}}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x} + \widehat{\boldsymbol{\mu}}_1^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_1 \right) + 2 \log \frac{\widehat{\pi}_1}{\widehat{\pi}_0} \\
&= 2(\widehat{\boldsymbol{\mu}}_1^T - \widehat{\boldsymbol{\mu}}_0^T) \boldsymbol{\Sigma}^{-1} \boldsymbol{x} + \widehat{\boldsymbol{\mu}}_0^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_0 - \widehat{\boldsymbol{\mu}}_1^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_1 + 2 \log \frac{\widehat{\pi}_1}{\widehat{\pi}_0} \\
&= (\boldsymbol{\Sigma}^{-1} (\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0))^T \boldsymbol{x} + \frac{1}{2} \widehat{\boldsymbol{\mu}}_0^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_0 - \frac{1}{2} \widehat{\boldsymbol{\mu}}_1^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_1 + \log \frac{\widehat{\pi}_1}{\widehat{\pi}_0}.
\end{aligned}$$

Thus, if

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1} (\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0)$$

and

$$b = \frac{1}{2} \widehat{\boldsymbol{\mu}}_0^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_0 - \frac{1}{2} \widehat{\boldsymbol{\mu}}_1^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_1 + \log \frac{\widehat{\pi}_1}{\widehat{\pi}_0},$$

we can re-write this as

$$\boldsymbol{w}^T \boldsymbol{x} + b \geq 0.$$

This is the expression of a simple linear classifier, and thus LDA will always result in a linear classifier.

Logistic regression

The key idea behind (binary) logistic regression is to *assume* that $\eta_1(\boldsymbol{x})$ is of the form

$$\frac{1}{1+\exp(-(\boldsymbol{w}^\intercal \boldsymbol{x} + b))} = 1 - \eta_0(\boldsymbol{x}),$$

and to directly estimate $\boldsymbol{w}$ and b from the data. Since the function $f(x) = \frac{1}{1+e^{-x}}$ is called the *logistic function*, the corresponding classifier inherited the name and is defined as

$$\begin{aligned} \widehat{h}(\boldsymbol{x}) &= \begin{cases} 1 & \text{if } \eta_1(\boldsymbol{x}) \geq \frac{1}{2}, \\ 0 & \text{if } \eta_1(\boldsymbol{x}) < \frac{1}{2}, \end{cases} \\ &= \begin{cases} 1 & \text{if } \widehat{\boldsymbol{w}}^\intercal \boldsymbol{x} + \widehat{b} \geq 0 \\ 0 & \text{if } \widehat{\boldsymbol{w}}^\intercal \boldsymbol{x} + \widehat{b} < 0. \end{cases} \end{aligned}$$

This is again a linear classifier. Note that LDA led to a similar classifier with the specific choice of parameters

$$\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_0) \quad b = \frac{1}{2}\widehat{\boldsymbol{\mu}}_0^\intercal \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_0 - \frac{1}{2}\widehat{\boldsymbol{\mu}}_1^\intercal \widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_1 + \log\frac{\widehat{\pi}_1}{\widehat{\pi}_0}$$

This is *not* what is done in logistic regression. Instead, in logistic regression we directly compute the maximum likelihood estimates of the parameters $\boldsymbol{w}$ and b.

Specifically, to analyze the MLE, we start with a standard trick to simplify notation, which consists in defining $\widetilde{\boldsymbol{x}} = [1, \boldsymbol{x}^\intercal]^\intercal$ and $\boldsymbol{\theta} = [b\, \boldsymbol{w}^\intercal]^\intercal$. This allows us to write the logistic model as

$$\eta(\boldsymbol{x}) = \eta_1(\boldsymbol{x}) = \frac{1}{1+\exp(-\boldsymbol{\theta}^\intercal \widetilde{\boldsymbol{x}})}.$$

To avoid carrying a tilde repeatedly in our notation, we will now simply write $\boldsymbol{x}$ in place of $\widetilde{\boldsymbol{x}}$, but keep in mind that we operate under the assumption that the first component of $\boldsymbol{x}$ is set to one.

Given our dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ the likelihood is $\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^n \mathbb{P}[y_i|\boldsymbol{x}_i; \boldsymbol{\theta}]$, where we do not try to model the distribution of $\boldsymbol{x}_i$. For $K = 2$ and $\mathcal{Y} = \{0, 1\}$, we obtain

$$\mathcal{L}(\boldsymbol{\theta}) = \prod_{i=1}^n \eta(\boldsymbol{x}_i)^{y_i}(1 - \eta(\boldsymbol{x}_i))^{1-y_i}$$

In case you are not familiar with this way of writing the likelihood, note that

$$\eta(\boldsymbol{x}_i)^{y_i}(1 - \eta(\boldsymbol{x}_i))^{1-y_i} = \begin{cases} \eta(\boldsymbol{x}_i) = \eta_1(\boldsymbol{x}_i) & \text{if } y_i = 1 \\ (1 - \eta(\boldsymbol{x}_i)) = \eta_0(\boldsymbol{x}_i) & \text{if } y_i = 0. \end{cases}$$

The log likelihood can therefore be written as

$$\begin{aligned} \ell(\boldsymbol{\theta}) = \log \mathcal{L}(\boldsymbol{\theta}) &= \sum_{i=1}^n (y_i \log \eta(\boldsymbol{x}_i) + (1 - y_i) \log(1 - \eta(\boldsymbol{x}_i))) \\ &= \sum_{i=1}^n \left(y_i \log \frac{1}{1 + e^{-\boldsymbol{\theta}^\intercal \boldsymbol{x}}} + (1 - y_i) \log \frac{e^{-\boldsymbol{\theta}^\intercal \boldsymbol{x}}}{1 + e^{-\boldsymbol{\theta}^\intercal \boldsymbol{x}}} \right) \\ &= \sum_{i=1}^n \left(y_i \boldsymbol{\theta}^\intercal \boldsymbol{x}_i - \log(1 + e^{\boldsymbol{\theta}^\intercal \boldsymbol{x}_i}) \right). \end{aligned}$$

To find the minimum with respect to $\boldsymbol{\theta}$, a necessary condition for

Georgia Tech ECE 6254 Spring 2024; Notes by M. Bloch, M. A. Davenport, and J. Romberg. Last updated 20:12, February 7, 2024

optimality is $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) = \mathbf{0}$. Here, this means that

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \left(y_i \boldsymbol{x}_i - \frac{e^{\boldsymbol{\theta}^\intercal \boldsymbol{x}_i}}{1 + e^{\boldsymbol{\theta}^\intercal \boldsymbol{x}_i}} \boldsymbol{x}_i \right) \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i \left(y_i - \frac{1}{1 + e^{-\boldsymbol{\theta}^\intercal \boldsymbol{x}_i}} \right) \\
&= 0.
\end{aligned}
$$

Solving this equation means solving a nonlinear system of $d+1$ equations, for which there exists no clear methodology. Hence, we must resort to a numerical algorithm to find the solution of $\arg\min_\theta -\ell(\theta)$.

You should check for yourself $-\ell(\boldsymbol{\theta})$ is *convex* in $\boldsymbol{\theta}$, and there exists algorithms with *provable* convergence guarantees. We will mention a few specific techniques, such as gradient descent, Newton's method, but there are many more that especially useful in high dimension.