Linear Discriminant Analysis (LDA)

LDA is a "classical" method for doing classification. It is a *model-based* method. It may not work well, compared to other methods, if its assumptions are violated, but in some such cases it can perform decently.

LDA is based on the assumption that, for each class, the p-variate vectors of predictors are independent observations from a multivariate normal dist'n. It is further assumed that the normal dist'ns for the various classes all have the same covariance matrix. Only the dist'n means can differ. (So we have identically-shaped dist'ns which can differ in location.)

For example, in a two class setting, we could have mean vectors

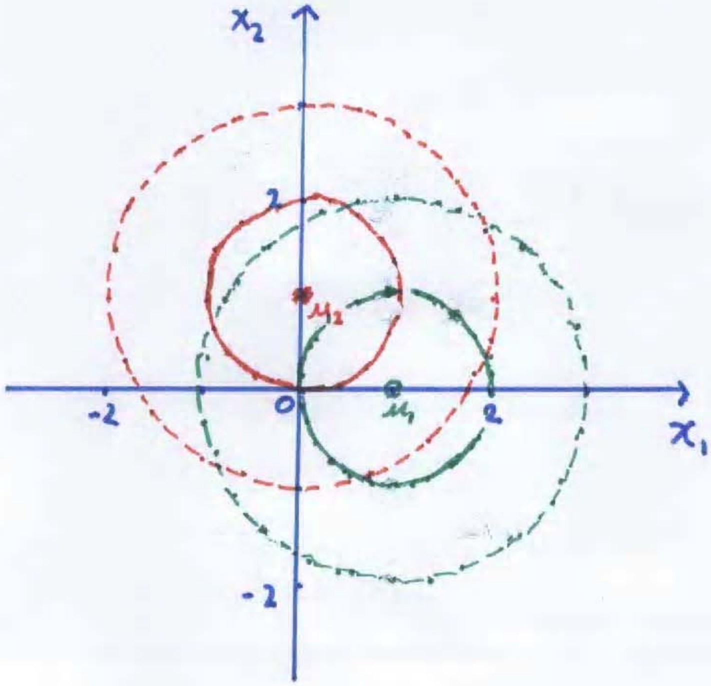
$$\vec{\mu}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \& \quad \vec{\mu}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

and both classes can have

$$\underset{\sim}{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

as their covariance matrix.

(Here we have $p = 2$.)



In general, if there are two classes, and f_1 and f_2 are the joint densities of the predictors for the two classes, and π_1 and π_2 are the "prior" probabilities (for cases to be classified) for the two classes, then the Bayes classifier will classify a case having $\vec{x}$ as the observed predictor values as belonging to class 1 if

$$\pi_1 f_1(\vec{x}) - \pi_2 f_2(\vec{x}) > 0.$$

Usually, f_1, f_2, π_1, and π_2 are unknown. However, if they can be estimated, to approximate the Bayes classifier, we can classify $\vec{x}$ as being of class 1 if

$$\hat{\pi}_1 \hat{f}_1(\vec{x}) - \hat{\pi}_2 \hat{f}_2(\vec{x}) > 0.$$

π_1 and π_2 can possibly be estimated using the training sample, by simple sample proportions, or estimates/guesstimates can be obtained some other way.

To estimate f_1 and f_2, one could use a nonparametric density estimator. (This approach will be covered in a future class.) But obtaining accurate nonparametric density estimates can be difficult if the sample sizes are too small, relative to p. Therefore, an alternative approach is to assume a parametric model, and use the training data to estimate the unknown parameters of the densities. With LDA, the assumed model is multivariate normal dist'ns having a common cov. matrix, and ML estimation can be used.

With the mult. norm. model assumed for LDA, for which the pdfs are of the form

$$f_k(\vec{x}) = \frac{\exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu}_k)^T \underset{\sim}{\Sigma}^{-1}(\vec{x}-\vec{\mu}_k)\right)}{(2\pi)^{p/2}\,|\underset{\sim}{\Sigma}|^{1/2}},$$

instead of using

$$\pi_1 f_1(\vec{x}) - \pi_2 f_2(\vec{x}) > 0$$

to express the Bayes rule (in the two class setting), it's more convenient to express the Bayes rule as

$$\frac{\pi_1 f_1(\vec{x})}{\pi_2 f_2(\vec{x})} > 1,$$

or equivalently as

$$\log(\pi_1/\pi_2) + \log f_1(\vec{x}) - \log f_2(\vec{x}) > 0.$$

Since the contributions from the denominators of the pdfs cancel, for the Bayes rule we have

$$\log(\pi_1/\pi_2) - \tfrac{1}{2}\left[(\vec{x}-\vec{\mu}_1)^T \underset{\sim}{\Sigma}^{-1}(\vec{x}-\vec{\mu}_1) - (\vec{x}-\vec{\mu}_2)^T \underset{\sim}{\Sigma}^{-1}(\vec{x}-\vec{\mu}_2)\right] > 0,$$

or equivalently,

$$\left(\log \pi_1 + \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_1 - \tfrac{1}{2} \vec{\mu}_1^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_1\right)$$
$$- \left(\log \pi_2\right) + \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_2 - \tfrac{1}{2} \vec{\mu}_2^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_2)$$
$$> 0.$$

Letting

$$\delta_k(\vec{x}) = \log \pi_k + \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_k - \tfrac{1}{2} \vec{\mu}_k^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_k,$$

we should classify $\vec{x}$ as being of class 1 if $\delta_1(\vec{x}) > \delta_2(\vec{x})$, and we should classify it as being of class 2 if $\delta_2(\vec{x}) > \delta_1(\vec{x})$. (There is no preferred class if $\delta_1(\vec{x}) = \delta_2(\vec{x})$.)

If there are more than two classes, we could, if we know the parameter values, obtain the value of $\delta_k(\vec{x})$ for each class, and predict the class which yields the largest value. The $\delta_k(\vec{x})$ are called the *linear discriminant functions*, and in practice we have to use estimated parameters with them to do classification (unless we know $\underset{\sim}{\Sigma}$ and the $\vec{\mu}_k$). Classification done in this way is typically referred to as *linear discriminant analysis* (although I think that some believe that this classification procedure should not be called discrim. analysis).

(To better understand the previous simplification, note that

$$(\vec{x} - \vec{\mu}_k)^T \underset{\sim}{\Sigma}^{-1} (\vec{x} - \vec{\mu}_k)$$

$$= \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{x} - \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_k - \vec{\mu}_k^T \underset{\sim}{\Sigma}^{-1} \vec{x} + \vec{\mu}_k^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_k .$$

Due to the subtraction of the $k = 2$ term from the $k = 1$ term, the $\vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{x}$ terms cancel out. Further simplification is achieved by noting that

$$\vec{\mu}_k^T \underset{\sim}{\Sigma}^{-1} \vec{x} = \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_k ,$$

which follows from the fact that $\vec{\mu}_k^T \underset{\sim}{\Sigma}^{-1} \vec{x}$ is 1 by 1, and is therefore equal to its transpose, and by noting that $\underset{\sim}{\Sigma}^{-1}$ is symmetric (since $\underset{\sim}{\Sigma}$ is), and is therefore equal to its transpose.)

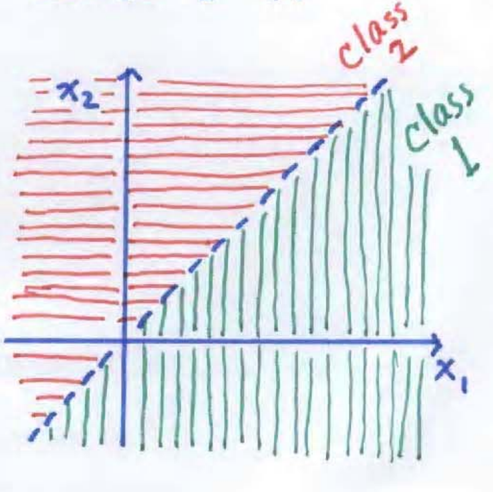
Using $\vec{\mu}_1$, $\vec{\mu}_2$, and $\underset{\sim}{\Sigma}$ from the prev. example, and letting $\pi_1 = \pi_2 = \frac{1}{2}$, the condition (for the Bayes rule)

$$\left(\log \pi_1 + \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_1 - \tfrac{1}{2} \vec{\mu}_1^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_1\right)$$
$$- \left(\log \pi_2 + \vec{x}^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_2 - \tfrac{1}{2} \vec{\mu}_2^T \underset{\sim}{\Sigma}^{-1} \vec{\mu}_2\right)$$
$$> 0$$

becomes (noting that $\underset{\sim}{\Sigma}^{-1} = \underset{\sim}{I}^{-1} = \underset{\sim}{I}$)

$$\left(\log \tfrac{1}{2} + (x_1, x_2)\begin{pmatrix}1\\0\end{pmatrix} - \tfrac{1}{2}(1,0)\begin{pmatrix}1\\0\end{pmatrix}\right)$$
$$- \left(\log \tfrac{1}{2} + (x_1, x_2)\begin{pmatrix}0\\1\end{pmatrix} - \tfrac{1}{2}(0,1)\begin{pmatrix}0\\1\end{pmatrix}\right)$$
$$> 0.$$

In the end we can get that $\vec{x}$ should be classified as being of class 1 if $x_1 > x_2$. Thus we have a linear decision boundary.



The Bayes error rate can be shown to be $\Phi(-\frac{1}{\sqrt{2}}) \doteq 0.240$.

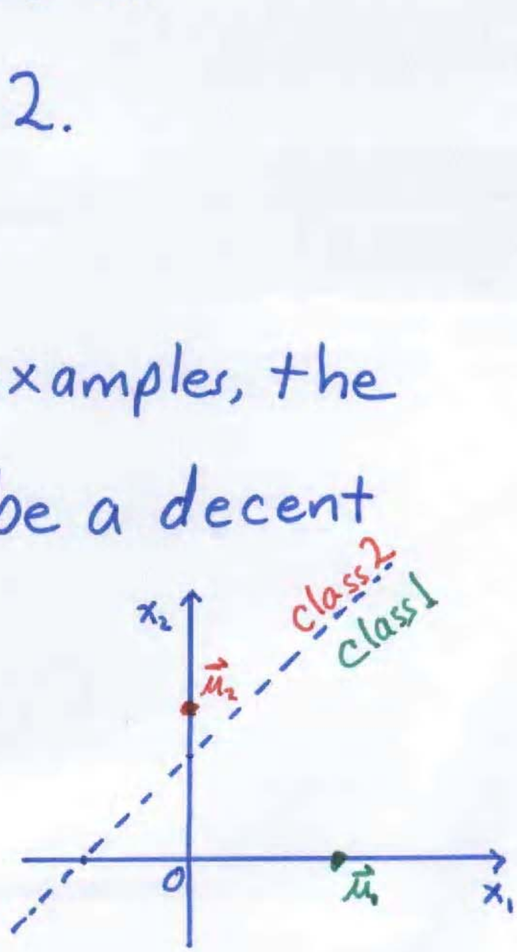
If we have that

$$\pi_1 = 2/3 \quad \& \quad \pi_2 = 1/3,$$

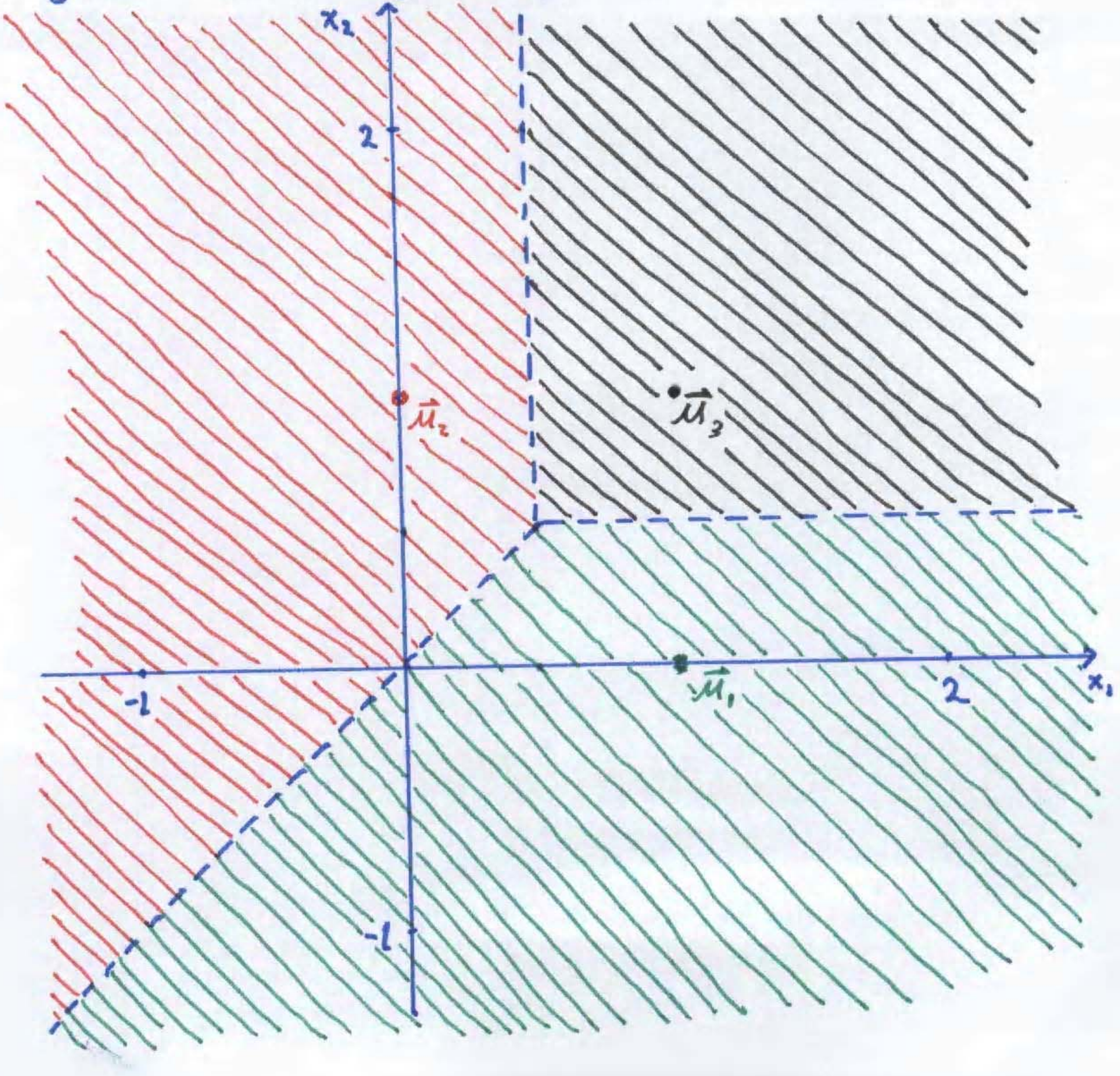
then the Bayes classifier classifies $\vec{x}$ as belonging to class 1 if

$$x_2 < x_1 + \log 2.$$

In each of these two examples, the LDA classifier should be a decent approximation of the Bayes classifier (unless there isn't adequate data).



To consider a three class problem, let's add a third class for which the mean is $(1,1)^T$. Then the Bayes decision boundaries are as shown below, for $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$. Again, LDA might approximate these well.



In the two class setting, if the optimal decision boundary is linear, LDA might still do well even if the assumptions of normality and a common covariance matrix don't hold. (Note that a binary predictor does not have a normal dist'n.) In some cases, transforming predictors improves performance. But if the optimal (Bayes) decision boundary is not close to linear, then LDA can perform poorly. One possible remedy is to expand the set of predictors (e.g., using x_1^2, x_2^2, and $x_1 x_2$ in addition to x_1 and x_2).

Before closing, I'll point out that if $p = 3$, the optimal boundary between two classes is a plane. when the assumptions underlying LDA are met. For $p > 3$, optimal boundaries are hyperplanes. For example, in a two class setting, the set of $\vec{x}$ values for which class 1 is preferred to class 2 will satisfy a rule of the form

$$c_0 + c_1 x_1 + c_2 x_2 + \cdots + c_p x_p > 0.$$