- Then Bayes' theorem states that

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- In general, estimating π_k is easy if we have a random sample from the population: we simply compute the fraction of the training observations that belong to the kth class.

- However, estimating the density function $f_k(x)$ is much more challenging. As we will see, to estimate $f_k(x)$, we will typically have to make some simplifying assumptions.

Linear Discriminant Analysis for $p = 1$

- We assume that $f_k(x)$ is normal or Gaussian. In the one-dimensional setting, the normal density takes the form

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x - \mu_k)^2\right)$$

  where μ_k and σ_k^2 are the mean and variance parameters for the kth class. We will also assume that all the $\sigma_k = \sigma$ are the same.

- Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = \Pr(Y = k|X = x)$:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x - \mu_l)^2\right)}$$

- To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest.

- Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest discriminant score:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

a linear function of x

For instance, if $K = 2$ and $\pi_1 = \pi_2 = 1/2$, then the classifier assigns an observation to class 1 if $\delta_1(x) > \delta_2(x)$ or $2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$, and to class 2 otherwise. (*)

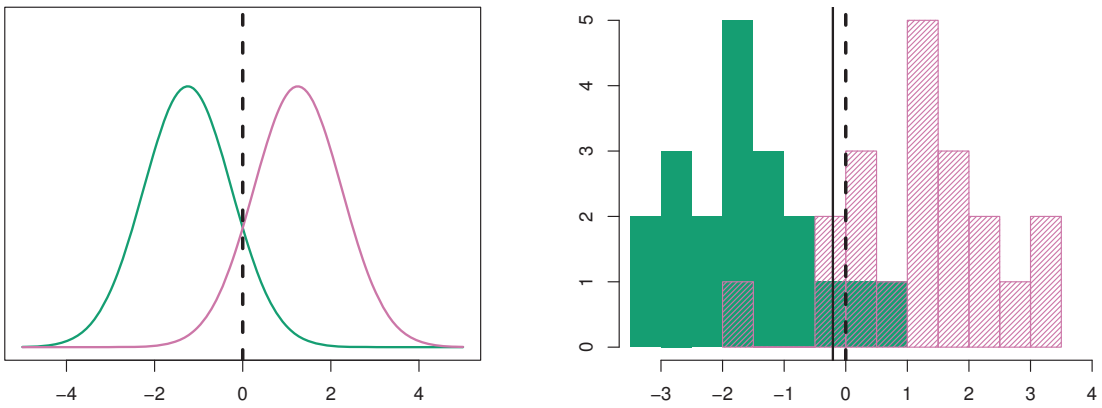
The decision boundary is the point for which $\delta_1(x) = \delta_2(x)$. We can show that this amounts to

$$x = \frac{(\mu_1^2 - \mu_2^2)}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

Consider an example with $\mu_1 = -1.5$, $\mu_2 = 1.5$, $\pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$.

Using (*), we see that the Bayes classifier assigns the observation to class 1 if $x < 0$ and class 2 otherwise.

The decission boundary is $x = 0$.



Left: Two one-dimensional normal density functions are shown. The dashed vertical line represents the Bayes decision boundary. Were it known, it would yield the fewest misclassification errors, among all possible classifiers. Right: 20 observations were drawn from each of the two classes, and are shown as histograms. The Bayes decision boundary is again shown as a dashed vertical line. The solid vertical line represents the LDA decision boundary estimated from the training data.

- Typically we don't know the parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule.

$$\hat{\pi}_k = \frac{n_k}{n}$$

← # of training observations in the kth class

← total # of training observations

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$= \sum_{k=1}^{K} \frac{n_k - 1}{n-K} \hat{\sigma}_k^2$$

where $\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$ is the estimated variance in the kth class.

The LDA classifier plugs these estimates into $\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$, assigns an observation $X = x$ to the class for which

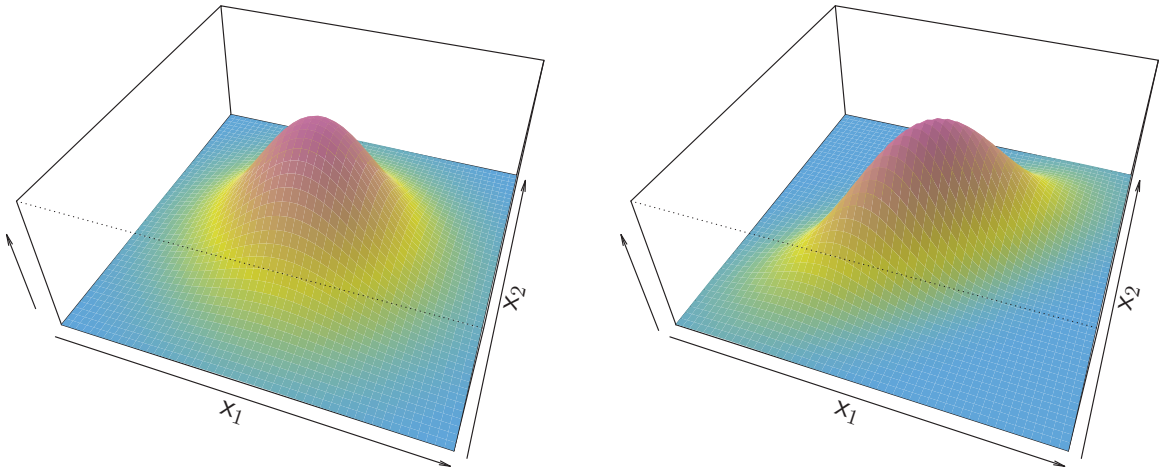
$$\hat{\delta}_k(x) = x\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

is largest.

Linear Discriminant Analysis for $p > 1$

We now extend the LDA classifier to the case of multiple predictors, $X = (X_1, \ldots, X_p)$.

- In the case of $p > 1$ predictors, the LDA classifier assumes that the observations in the kth class are drawn from a multivariate Gaussian distribution $N(\mu_k, \mathbf{\Sigma})$, where μ_k is a class-specific mean vector and $\mathbf{\Sigma}$ is a $p \times p$ covariance matrix that is common to all K classes.



Two multivariate Gaussian density functions are shown, with $p = 2$. Left: The two predictors are uncorrelated. Right: The two variables have a correlation of 0.7.

- The density function for the kth class takes the form

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}^{-1}(x-\mu_k)\right)$$

- Plugging this into Bayes formula and performing a little bit of algebra, we get the discriminant function:

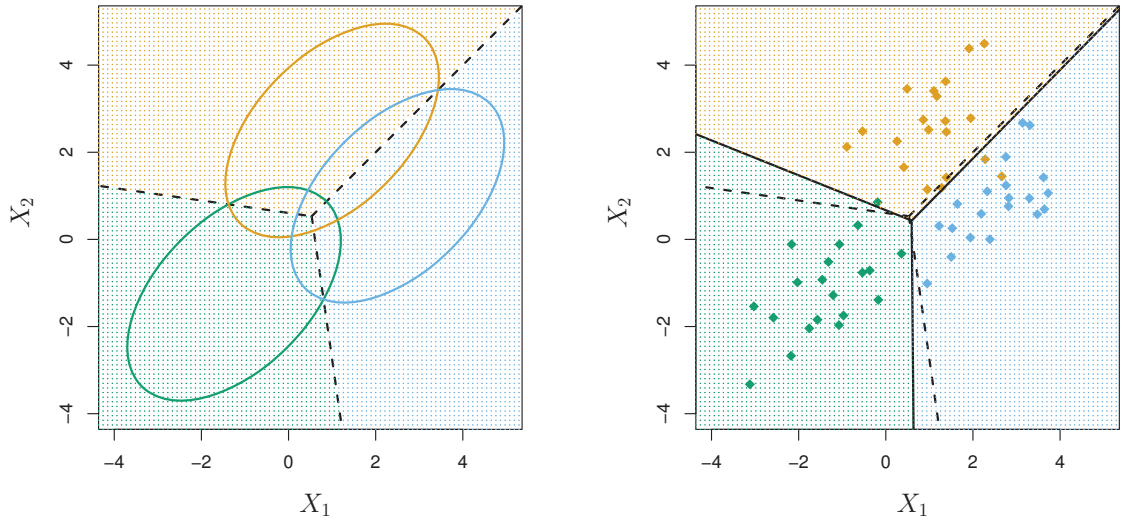
$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k \qquad (1)$$

and then x is assigned to the class with the largest discriminant score.

we can rewrite (1) as

$$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kp}x_p$$

This is a linear function.



An example with three classes. The observations from each class are drawn from a multivariate Gaussian distribution with $p = 2$, with a class-specific mean vector and a common covariance matrix. Left: Ellipses that contain 95 % of the probability for each of the three classes are shown. The dashed lines are the Bayes decision boundaries. Right: 20 observations were generated from each class, and the corresponding LDA decision boundaries are indicated using solid black lines. The Bayes decision boundaries are once again shown as dashed lines.

The dashed lines represent the set of values x for which $\delta_k(x) = \delta_l(x)$ for $k \neq l$.

- Once again, we need to estimate the unknown parameters $\mu_1, \ldots, \mu_K$, $\pi_1, \ldots, \pi_K$, and $\mathbf{\Sigma}$.
- To assign a new observation $X = x$, LDA plugs the estimates into $\delta_k(x)$ to obtain the quantity $\hat{\delta}_k(x)$, and classifies to the class for which $\hat{\delta}_k(x)$ is largest.

The solid lines represent the set of values x for which $\hat{\delta}_k(x) = \hat{\delta}_l(x)$ for $k \neq l$.

Now we will perform LDA on the Default data in order to predict whether or not an individual will default on the basis of credit card balance and student status.

In R, we fit an LDA model using the lda() function, which is part of the MASS library.

```
> library(ISLR)
> names(Default)
[1] "default" "student" "balance" "income"
> library(MASS)
> lda_fit_def_1 <- lda(default ~ balance + student, data = Default)
> lda_fit_def_1
Call:
lda(default ~ balance + student, data = Default)

Prior probabilities of groups:
    No    Yes
0.9667 0.0333

Group means:
      balance studentYes
No   803.9438  0.2914037
Yes 1747.8217  0.3813814

Coefficients of linear discriminants:
                    LD1
balance     0.002244397
studentYes -0.249059498
```

Once we have estimates $\hat{\delta}_k(x)$, we can turn these into estimates for class probabilities:

$$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^{K} e^{\hat{\delta}_l(x)}}$$

So classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\hat{\Pr}(Y=k|X=x)$ is largest. When $K=2$, we classify to class 2 if $\hat{\Pr}(Y=2|X=x) > 0.5$, otherwise to class 1.

The predict() function returns a list with three elements. The first element, class, contains LDA's predictions about the default. The second element, posterior, is a matrix whose kth column contains the posterior probability that the corresponding observation belongs to the kth class, computed from $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$ or the formula above. Finally, x contains the linear discriminants.

```
> lda_pred_def_1 <- predict(lda_fit_def_1)
> names(lda_pred_def_1)
[1] "class"     "posterior" "x"
> table(lda_pred_def_1$class, Default$default)

       No  Yes
  No  9644  252
  Yes   23   81
```

```
> mean(lda_pred_def_1$class == Default$default)
[1] 0.9725
> mean(lda_pred_def_1$class != Default$default)
[1] 0.0275
```

		True default status		
		No	Yes	Total
Predicted default status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

A confusion matrix compares the LDA predictions to the true default statuses for the 10,000 training observations in the Default data set.

The LDA model fit to the 10,000 training samples results in a training error rate of 2.75%. This sounds like a low error rate, but some caveats must be noted:

- This is training error, and we may be overfitting.

But this is not a big concern here since $n = 10000$ and $p = 2$.

- If we classified all individuals to class No, we would make 333/10000 errors, or only 3.33%.

- Of the true No's, we make $23/9667 = 0.2\%$ errors; of the true Yes's, we make $252/333 = 75.7\%$ errors!