Classification 1: Linear regression of indicators, linear discriminant analysis

Ryan Tibshirani
Data Mining: 36-462/36-662

April 2 2013

Optional reading: ISL 4.1, 4.2, 4.4, ESL 4.1–4.3

Classification

Classification is a predictive task in which the response takes values across discrete categories (i.e., not continuous), and in the most fundamental case, two categories

Examples:

- Predicting whether a patient will develop breast cancer or remain healthy, given genetic information
- Predicting whether or not a user will like a new product, based on user covariates and a history of his/her previous ratings
- Predicting the region of Italy in which a brand of olive oil was made, based on its chemical composition
- Predicting the next elected president, based on various social, political, and historical measurements

Similar to our usual setup, we observe pairs (x_i, y_i), $i = 1, \ldots n$, where y_i gives the class of the ith observation, and $x_i \in \mathbb{R}^p$ are the measurements of p predictor variables

Though the class labels may actually be $y_i \in \{\text{healthy}, \text{sick}\}$ or $y_i \in \{\text{Sardinia}, \text{Sicily}, ...\}$, but we can always encode them as

$$y_i \in \{1, 2, \ldots K\}$$

where K is the total number of classes

Note that there is a big difference between classification and clustering; in the latter, there is not a pre-defined notion of class membership (and sometimes, not even K), and we are not given labeled examples (x_i, y_i), $i = 1, \ldots n$, but only x_i, $i = 1, \ldots n$

Constructed from training data (x_i, y_i), $i = 1, \ldots n$, we denote our classification rule by $\hat{f}(x)$; given any $x \in \mathbb{R}^p$, this returns a class label $\hat{f}(x) \in \{1, \ldots K\}$

As before, we will see that there are two different ways of assessing the quality of $\hat{f}$: its predictive ability and interpretative ability

E.g., train on (y_i, x_i), $i = 1, \ldots n$, the data of elected presidents and related feature measurements $x_i \in \mathbb{R}^p$ for the past n elections, and predict, given the current feature measurements $x_0 \in \mathbb{R}^p$, the winner of the current election

In what situations would we care more about prediction error? And in what situations more about interpretation?

Binary classification and linear regression

Let's start off by supposing that $K = 2$, so that the response is $y_i \in \{1, 2\}$, for $i = 1, \ldots n$

You already know a tool that you could potentially use in this case for classification: linear regression. Simply treat the response as if it were continuous, and find the linear regression coefficients of the response vector $y \in \mathbb{R}^n$ onto the predictors, i.e.,

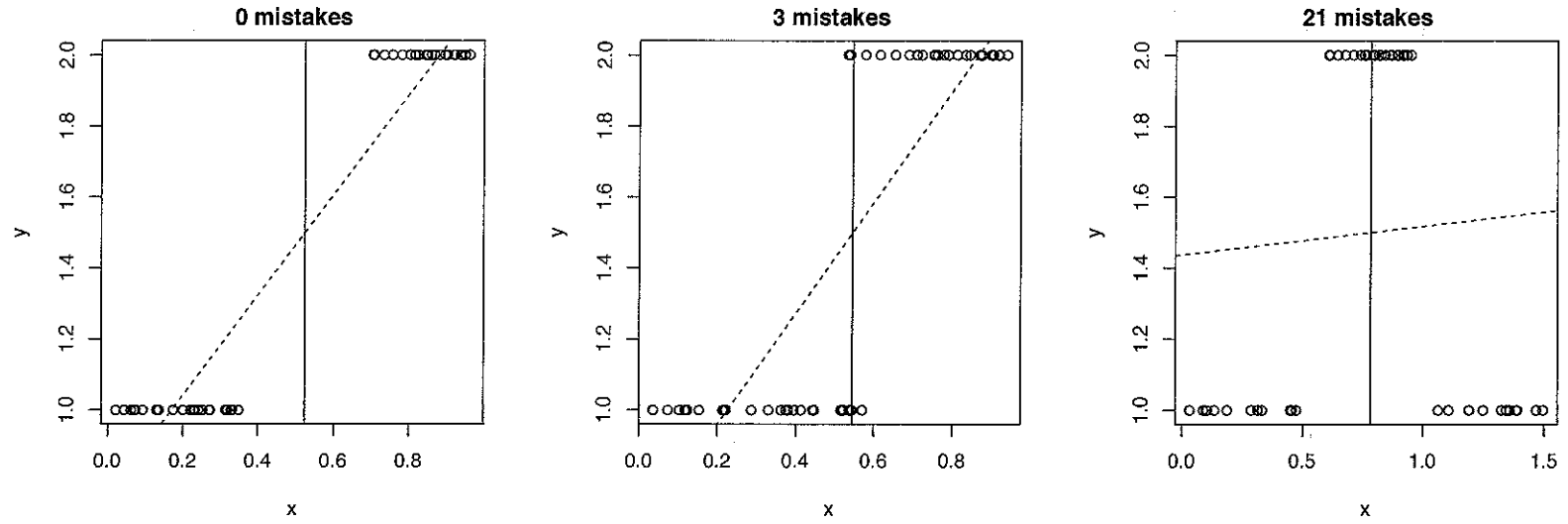
$$\hat{\beta}_0, \hat{\beta} = \underset{\beta_0 \in \mathbb{R},\, \beta \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - x_i^T \beta)^2$$

Then, given a new input $x_0 \in \mathbb{R}^p$, we predict the class to be

$$\hat{f}^{\mathrm{LS}}(x_0) = \begin{cases} 1 & \text{if } \hat{\beta}_0 + x_0^T \hat{\beta} \leq 1.5 \\ 2 & \text{if } \hat{\beta}_0 + x_0^T \hat{\beta} > 1.5 \end{cases}$$

(Note: since we included an intercept term in the regression, it doesn't matter whether we code the class labels as $\{1, 2\}$ or $\{0, 1\}$, etc.)

In many instances, this actually works reasonably well. Examples:



Overall, using linear regression in this way for binary classification is not a crazy idea. But how about if there are more than 2 classes?

Linear regression of indicators

This idea extends to the case of more than two classes. Given K classes, define the indicator matrix $Y \in \mathbb{R}^{n \times K}$ to be the matrix whose columns indicate class membership; that is, its jth column satisfies $Y_{ij} = 1$ if $y_i = j$ (observation i is in class j) and $Y_{ij} = 0$ otherwise

E.g., with $n = 6$ observations and $K = 3$ classes, the matrix

$$Y = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{6 \times 3}$$

corresponds to having the first two observations in class 1, the next two in class 2, and the final 2 in class 3

To construct a prediction rule, we regress each column $Y_j \in \mathbb{R}^n$ (indicating the jth class versus all else) onto the predictors:

$$\hat{\beta}_{j,0}, \hat{\beta}_j = \underset{\beta_{j,0} \in \mathbb{R},\, \beta_j \in \mathbb{R}^p}{\operatorname{argmin}} \sum_{i=1}^{n} (Y_{ij} - \beta_{0,j} - \beta_j^T x_i)^2$$

Now, given a new input $x_0 \in \mathbb{R}^p$, we compute

$$\hat{\beta}_{0,j} + x_0^T \hat{\beta}_j, \quad j = 1, \ldots K$$

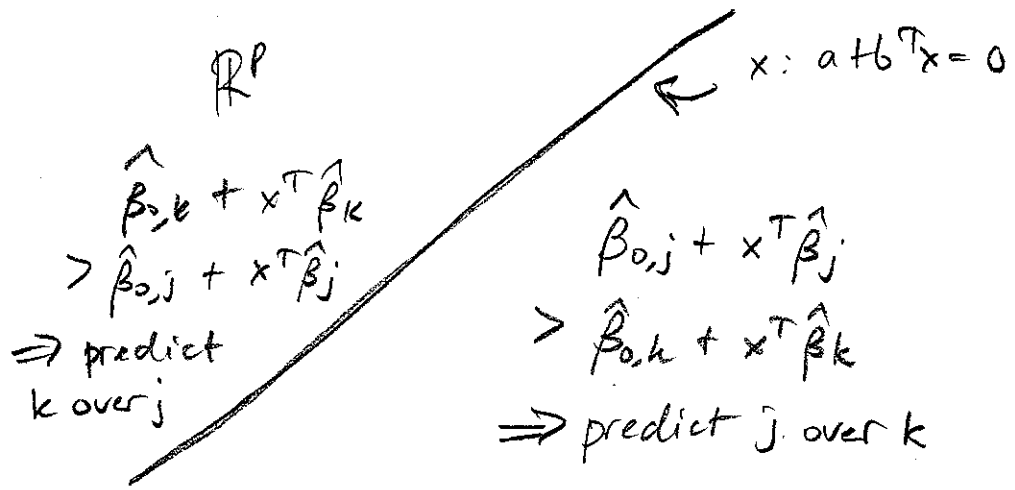
take predict the class j that corresponds to the highest score. I.e., we let each of the K linear models make its own prediction, and then we take the strongest. Formally,

$$\hat{f}^{\mathrm{LS}}(x_0) = \underset{j=1,\ldots K}{\operatorname{argmax}} \; \hat{\beta}_{0,j} + x_0^T \hat{\beta}_j$$

The decision boundary between any two classes j, k are the values of $x \in \mathbb{R}^p$ for which

$$\hat{\beta}_{0,j} + x^T \hat{\beta}_j = \hat{\beta}_{0,k} + x^T \hat{\beta}_k$$

i.e., $\underbrace{\hat{\beta}_{0,j} - \hat{\beta}_{0,k}}_{a} + \underbrace{(\hat{\beta}_j - \hat{\beta}_k)}_{b}{}^T x = 0$ $\quad \{x \in \mathbb{R}^p : a + b^T x = 0\}$

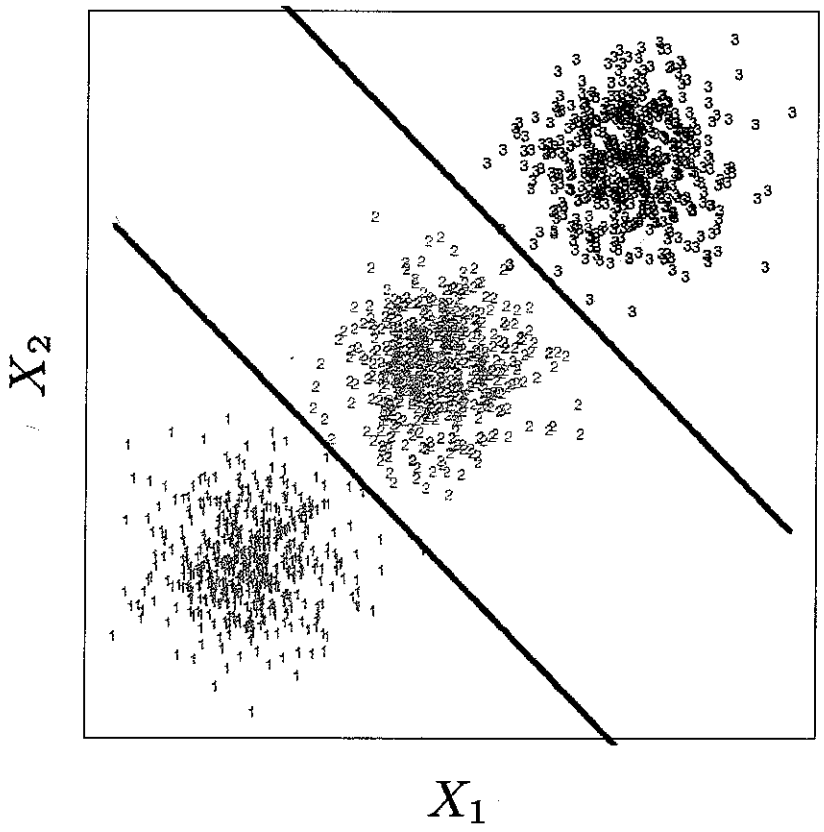


This defines a $(p-1)$-dimensional affine subspace in $\mathbb{R}^p$. To one side, we would always predict class j over k; to the other, we would always predict class k over j

For K classes total, there are $\binom{K}{2} = \frac{K(K-1)}{2}$ decision boundaries

Ideal result

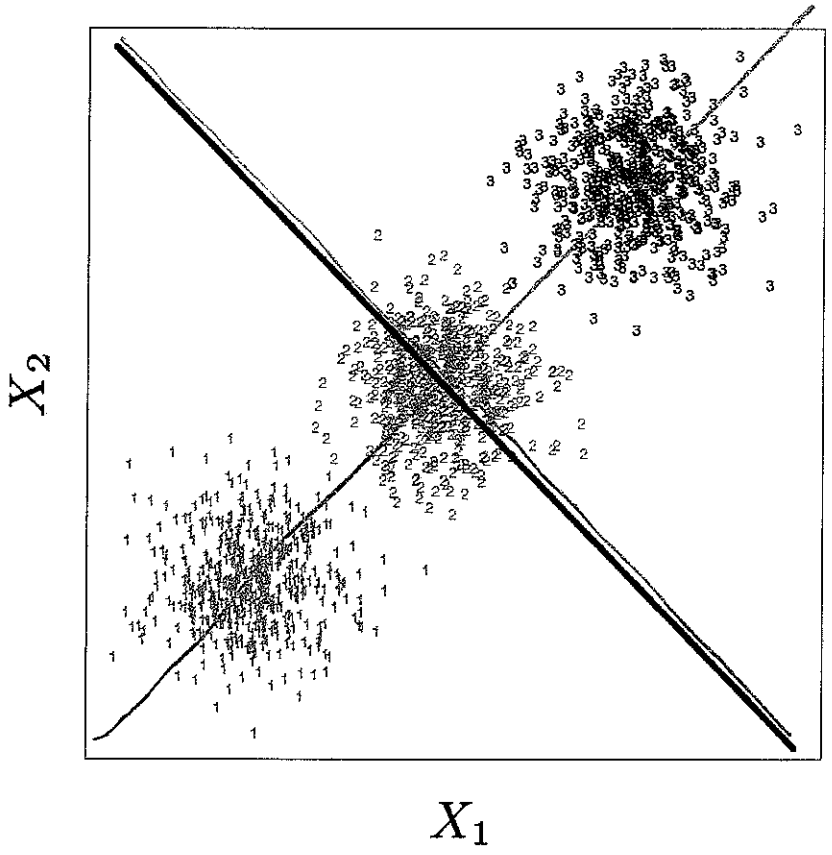
What we'd like to see when we use linear regression for a 3-way classification (from ESL page 105):



The plotted lines are the decision boundaries between classes 1 and 2, and 2 and 3 (the decision boundary between classes 1 and 3 never matters)

Actual result

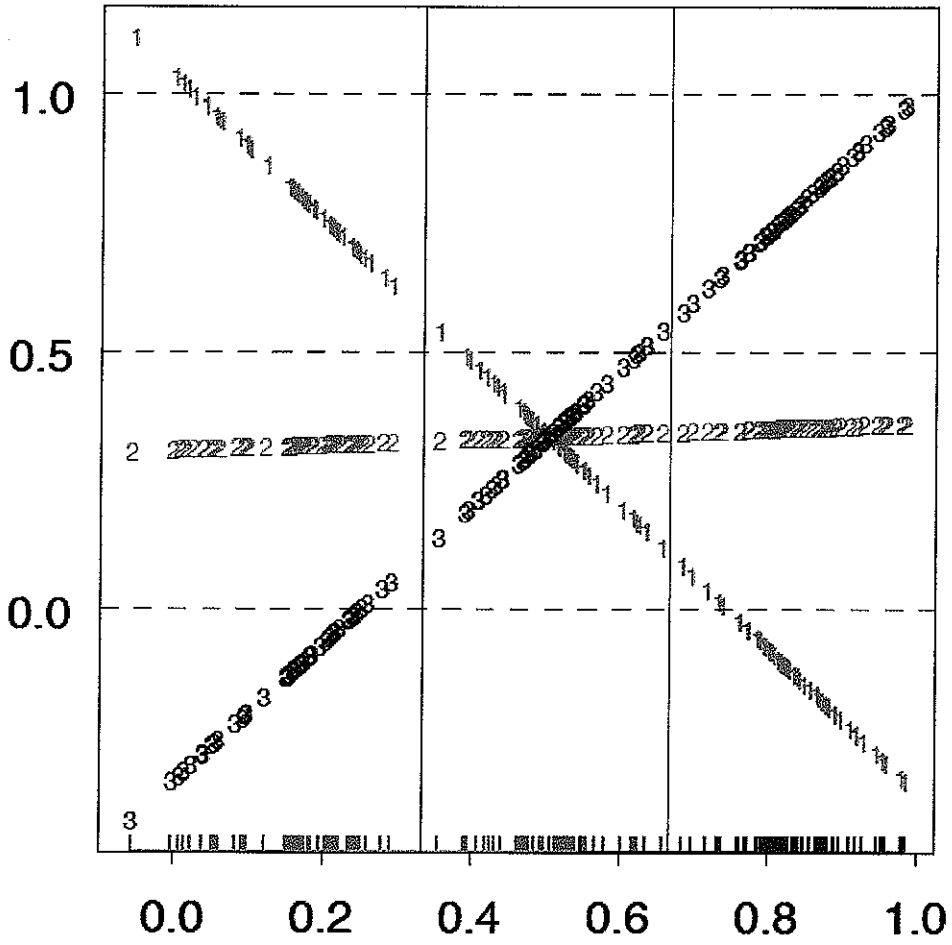
What actually happens when we use linear regression for this 3-way classification (from ESL page 105):



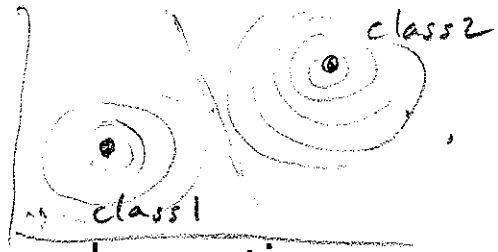
The decision boundaries between 1 and 2 and between 2 and 3 are the same, so we would never predict class 2. This problem is called masking (and it is not uncommon for moderate K and small p)

Why did this happen?

Projecting onto the line joining the three class centroids gives some insight into why this happened (from ESL page 106):



Statistical decision theory



Let C be a random variable giving the class label of an observation in our data set. A natural rule would be to classify according to

$$f(x) = \underset{j=1,\ldots K}{\operatorname{argmax}} \ \mathrm{P}(C = j | X = x)$$

This predicts the most likely class, given the feature measurements $X = x \in \mathbb{R}^p$. This is called the Bayes classifier, and it is the best that we can do (think of overlapping classes)

$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

Note that we can use Bayes' rule to write

$$\mathrm{P}(C = j | X = x) = \frac{\mathrm{P}(X = x | C = j) \cdot \mathrm{P}(C = j)}{\mathrm{P}(X = x)}$$

π_j

doesn't depend on j

Let $\pi_j = \mathrm{P}(C = j)$ be the prior probability of class j. Since the Bayes classifier compares the above quantity across $j = 1, \ldots K$ for $X = x$, the denominator is always the same, hence

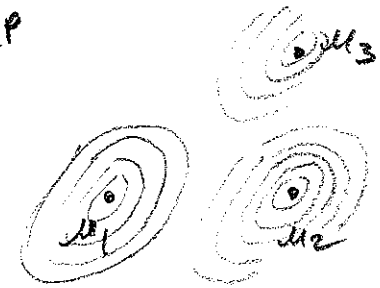
$$f(x) = \underset{j=1,\ldots K}{\operatorname{argmax}} \ \mathrm{P}(X = x | C = j) \cdot \pi_j$$

Linear discriminant analysis

Using the Bayes classifier is not realistic as it requires knowing the class conditional densities $\mathrm{P}(X = x|C = j)$ and prior probabilities π_j. But if estimate these quantities, then we can follow the idea

Linear discriminant analysis (LDA) does this by assuming that the data within each class are normally distributed:

$$h_j(x) = \mathrm{P}(X = x|C = j) = N(\mu_j, \Sigma) \text{ density}$$



We allow each class to have its own mean $\mu_j \in \mathbb{R}^p$, but we assume a common covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$. Hence

$$h_j(x) = \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}} \exp\left\{ -\frac{1}{2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)\right\}$$

multivariate normal density

So we want to find j so that $\mathrm{P}(C = j|X = x) \cdot \pi_j = h_j(x) \cdot \pi_j$ is the largest

Since $\log(\cdot)$ is a monotone function, we can consider maximizing $\log(h_j(x)\pi_j)$ over $j = 1, \ldots K$. We can define the rule:

$$
\begin{aligned}
f^{\mathrm{LDA}}(x) &= \underset{j=1,\ldots K}{\operatorname{argmax}} \log\left[\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)}{2}\right) \cdot \pi_j\right] \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \left(\log(2\pi)^{p/2}|\Sigma|^{1/2}\right) - \frac{(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)}{2} + \log(\pi_j) \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \; x^T\Sigma^{-1}\mu_j - \frac{x^T\Sigma^{-1}x}{2} - \frac{\mu_j^T\Sigma^{-1}\mu_j}{2} + \log(\pi_j) \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \; \delta_j(x)
\end{aligned}
$$

We call $\delta_j(x)$, $j = 1, \ldots K$ the discriminant functions. Note

$$
\delta_j(x) = x^T\underbrace{\Sigma^{-1}\mu_j}_{b_j} - \underbrace{\frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \log\pi_j}_{a_j} = a_j + b_j^T x
$$

is just an affine function of x

In practice, given an input $x \in \mathbb{R}^p$, can we just use the rule f^{LDA} on the previous slide? Not quite! What's missing: we don't know π_j, μ_j, and Σ. Therefore we estimate them based on the training data $x_i \in \mathbb{R}^p$ and $y_i \in \{1, \ldots K\}$, $i = 1, \ldots n$, by:

- $\hat{\pi}_j = n_j/n$, the proportion of observations in class j
- $\hat{\mu}_j = \frac{1}{n_j} \sum_{y_i = j} x_i$, the centroid of class j
- $\hat{\Sigma} = \frac{1}{n-K} \sum_{j=1}^{K} \sum_{y_i = j} (x_i - \hat{\mu}_j)(x_i - \hat{\mu}_j)^T$, the pooled sample covariance matrix

(Here n_j is the number of points in class j)

This gives the estimated discriminant functions:

$$\hat{\delta}_j(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_j - \frac{1}{2} \hat{\mu}_j^T \hat{\Sigma}^{-1} \hat{\mu}_j + \log \hat{\pi}_j \quad = \quad a_j + b_j^T x$$

and finally the linear discriminant analysis rule,

$$\hat{f}^{\mathrm{LDA}}(x) = \underset{j=1,\ldots K}{\operatorname{argmax}} \ \hat{\delta}_j(x)$$

LDA decision boundaries

The estimated discriminant functions

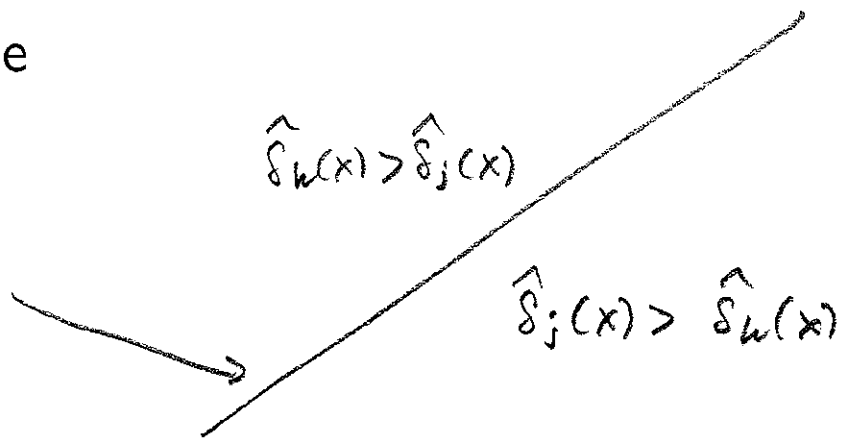
$$\begin{aligned}\hat{\delta}_j(x) &= x^T\hat{\Sigma}^{-1}\hat{\mu}_j - \frac{1}{2}\hat{\mu}_j^T\hat{\Sigma}^{-1}\hat{\mu}_j + \log\hat{\pi}_j \\ &= a_j + b_j^T x\end{aligned}$$

are just affine functions of x. The decision boundary between classes j, k is the set of all $x \in \mathbb{R}^p$ such that $\hat{\delta}_j(x) = \hat{\delta}_k(x)$, i.e.,

$$a_j + b_j^T x = a_k + b_k^T x$$

This defines an affine subspace in x:

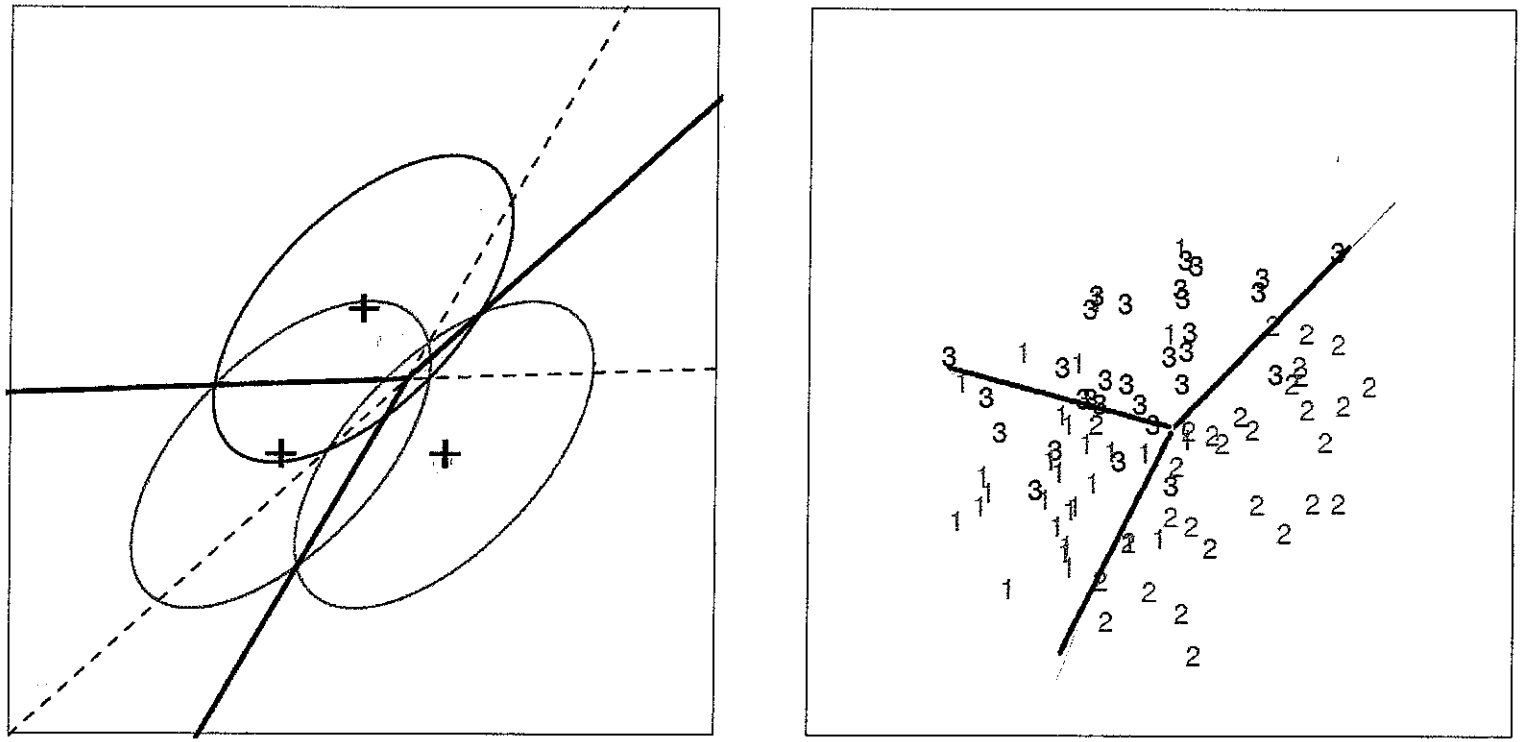
$$a_j - a_k + (b_j - b_k)^T x = 0$$



Example: LDA decision boundaries

Example of decision boundaries from LDA (from ESL page 109):

$f^{\mathrm{LDA}}(x)$ $\hat{f}^{\mathrm{LDA}}(x)$



Are the decision boundaries the same as the perpendicular bisectors (Voronoi boundaries) between the class centroids? (Why not?)

LDA computations, usages, extensions

The decision boundaries for LDA are useful for graphical purposes, but to classify a new point $x_0 \in \mathbb{R}^p$ we don't use them—we simply compute $\hat{\delta}_j(x_0)$ for each $j = 1, \ldots K$

LDA performs quite well on a wide variety of data sets, even when pitted against fancy alternative classification schemes. Though it assumes normality, its simplicity often works in its favor. (Why? Think of the bias-variance tradeoff)

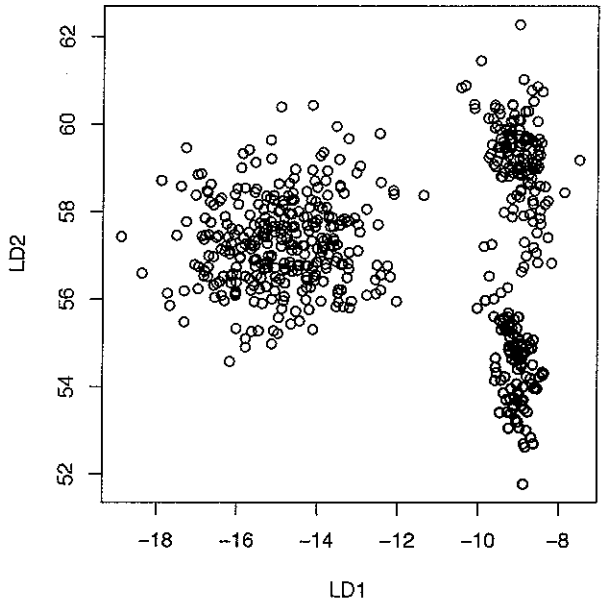
Still, there are some useful extensions of LDA. E.g.,

- Quadratic discriminant analysis: using the same normal model, we now allow each class j to have its own covariance matrix Σ_j. This leads to quadratic decision boundaries
- Reduced-rank linear discriminant analysis: we essentially project the data to a lower dimensional subspace before performing LDA. We will study this next time

Example: olive oil data

Example: $n = 572$ olive oils, each made in one of three regions of Italy. On each observation we have $p = 8$ features measuring the percentage composition of 8 different fatty acids. (Data from the `olives` data set from the R package `classifly`)

From the `lda` function in the `MASS` package:



This looks nice (seems that the observations are separated into classes), but what exactly is being shown? More next time...

Recap: linear regression of indicators, linear discriminant analysis

In this lecture, we introduced the task of classification, a prediction problem in which the outcome is categorical

We can perform classification for any total number of classes K by simply performing K separate linear regressions on the appropriate indicator vectors of class membership. However, there can be problems with this—when $K > 2$, a common problem is masking, in which one class is never predicted at all

Linear discriminant analysis also draws linear decision boundaries but in a smarter way. Statistical decision theory tells us that we really only need to know the class conditional densities and the prior class probabilities in order to perform classification. Linear discriminant analysis assumes normality of the data within each class, and assumes a common covariance matrix; it then replaces all unknown quantities by their sample estimates

Next time: more linear discriminant analysis; logistic regression

Logistic regression is a natural extension of the ideas behind linear regression and linear discriminant analysis.