# Classification

14/05 – 15/05

Last week we talked about the difference between **supervised** and **unsupervised** techniques.

Next 4 lectures: Classification (supervised) (to May 21)

Next 2 lectures: Clustering (unsupervised) (May 27-28)

- Review Lecture
- Student presentation

# Two kinds of supervised problem:

```
model y = x1 x2 x3 … xp;
```

If y continuous, we have a **regression** problem

If y is discrete, we have a **classification** problem

Classification problems are very important nowadays, even though they weren’t so popular in 20th century statistics. Examples:

- (E. Anderson ‘35; Fisher ‘36) Classify iris into one of three species (Versicolor, Virginica, Setosa) based on length & width of sepals and petals.

- Classify patient into one of several diseases based on symptoms (may be discrete or continuous)

- Classify email as spam/not spam based on the words in the message (features are discrete here)

- Classify a speech as one of several topics based on words in the speech

- Classify handwritten digits as 0-9 based on pixel greyscale values

- *(see ESL Introduction for many more)*

There may be several classes (eg. apple/orange/pear) or two classes (diabetes/no diabetes; spam/not spam)

Usually focus on the two-class problem. Can code y as 0 (for one class) and 1 (for the other class) and then use linear regression to predict `y` using `x1`, … `xp`.

In econometrics, this is called the *Harvard Model* (not terrible, but never the best you can do.)

**Classifier:**

A rule which assigns a class `y` to each possible `(x1, x2, … xp)`

*How can we measure the performance of a classifier?*

We can count how many incorrect identifications it makes:

$$Error = \sum I(y_i \neq f(\boldsymbol{x}_i))$$

(Here, our classifier is called $f$)

Of course, we want to minimise the error on unseen test data, not on the training data (equivalently: generalise from a sample to a population.)

Can use a validation set or cross-validation in exactly the same way as in regression problems.

Note:

In real life, often there is a different cost associated with different kinds of misclassification, e.g.

- Let a guilty person go free or punish an innocent person?
- Remove tonsils and adenoids from healthy child?

You might have a **cost matrix** describing the costs of different kinds of misclassification, and then you want to minimise the **cost**

$$\sum C_{ij} I(y_j = f(\boldsymbol{x}_i))$$

We will ignore this issue, but it is very important in practice when selecting the “best” from a number of possible models.

Ignoring this issue, there is a unique solution to the problem of minimising $\sum I(y_i \neq f(\boldsymbol{x}_i))$, called the **Bayes optimal classifier**

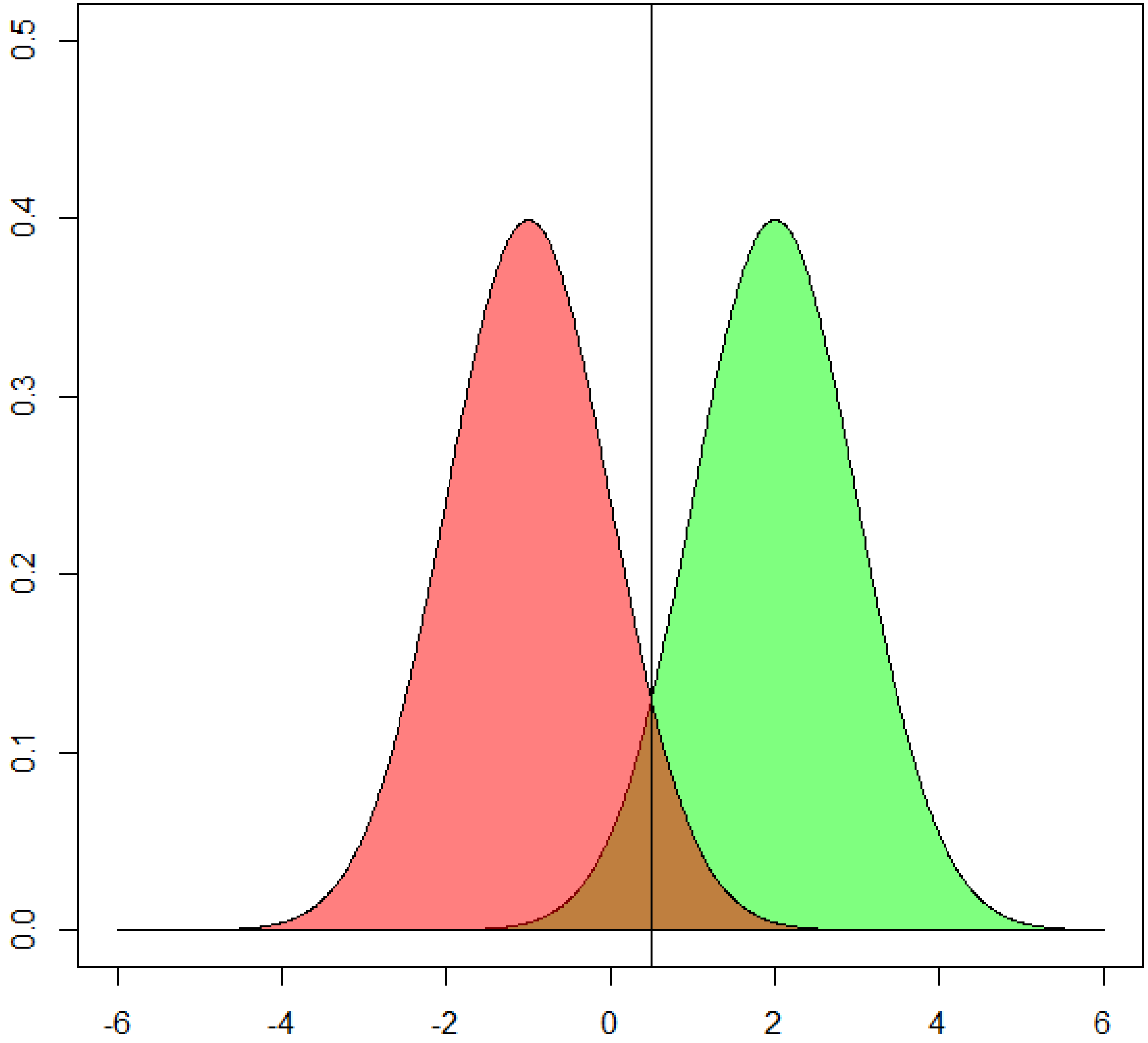
# Bayes Classifier

Classify an observation $\boldsymbol{x}$ to class $j$ if

$$\pi_j f_j(\boldsymbol{x}) > \pi_i f_i(\boldsymbol{x})$$

for all $i \neq j$, where $f_j(\boldsymbol{x})$ is the density of the observations in class $j$ and $\pi_j$ is the (prior) probability of an observation being in class $j$.

# Bayes Classifier Example



Prior probability of being in each class = 0.5

Class 1 density ~ normal(-1, 1)

Class 2 density ~ normal(2, 1)

Optimal classifier: classify as class 1 if
x < 0.5, else class 2.

Bayes rate:

0.5(0.067) + 0.5(0.067) = 0.067

[You can’t get better than 93% correct in this problem!]

So… Problem solved!

Why can’t we just use the Bayes classifier?

*We don’t know the* $f_j(\boldsymbol{x})$.

Two approaches:

- Estimate the $f_j$ from the data in some way (LDA; naïve Bayes; trees; random forests; neural networks)

- Find a boundary between the classes without caring about the $f_j$ (logistic regression; support vector machines)

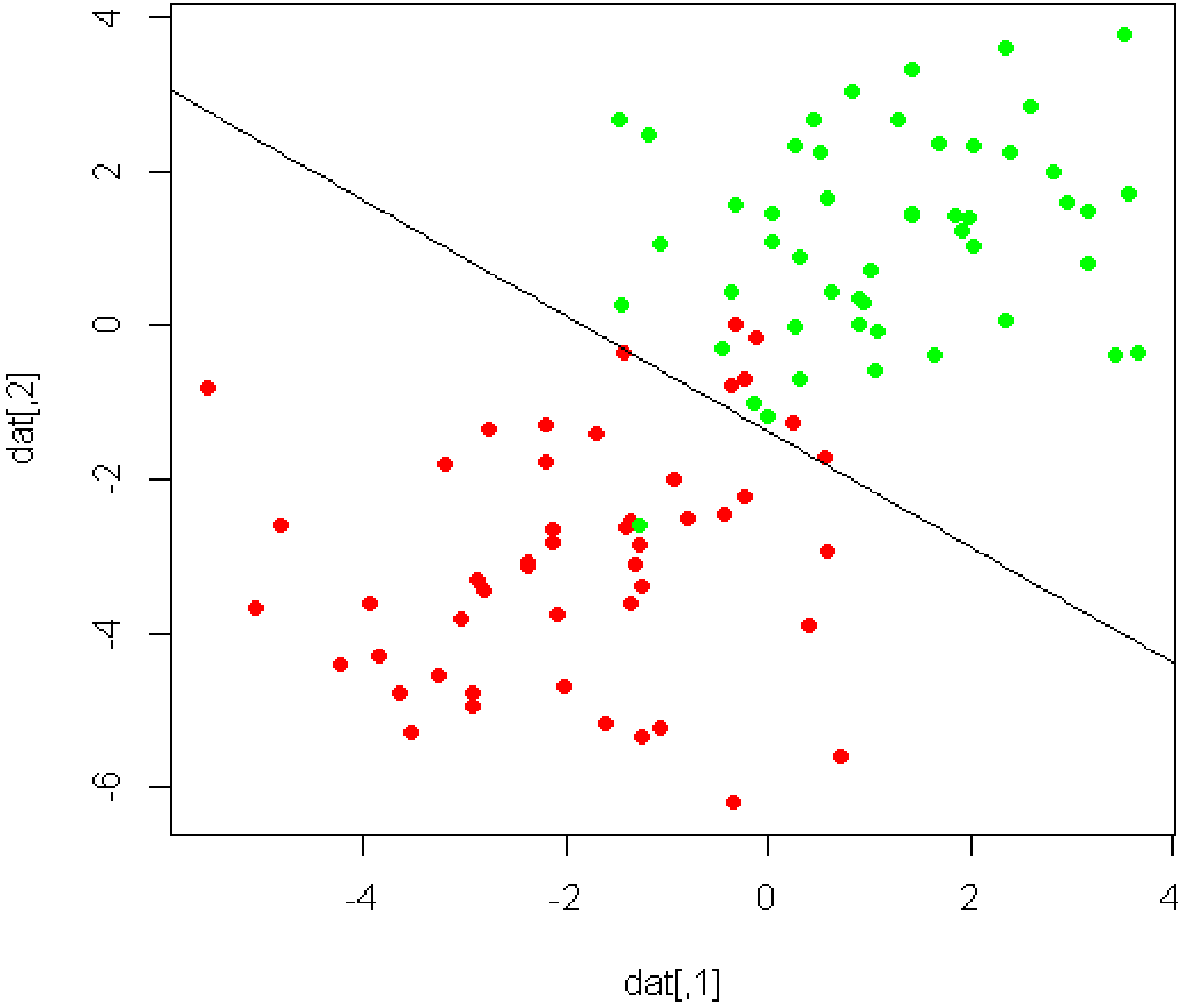
# Linear Discriminant Analysis (LDA)

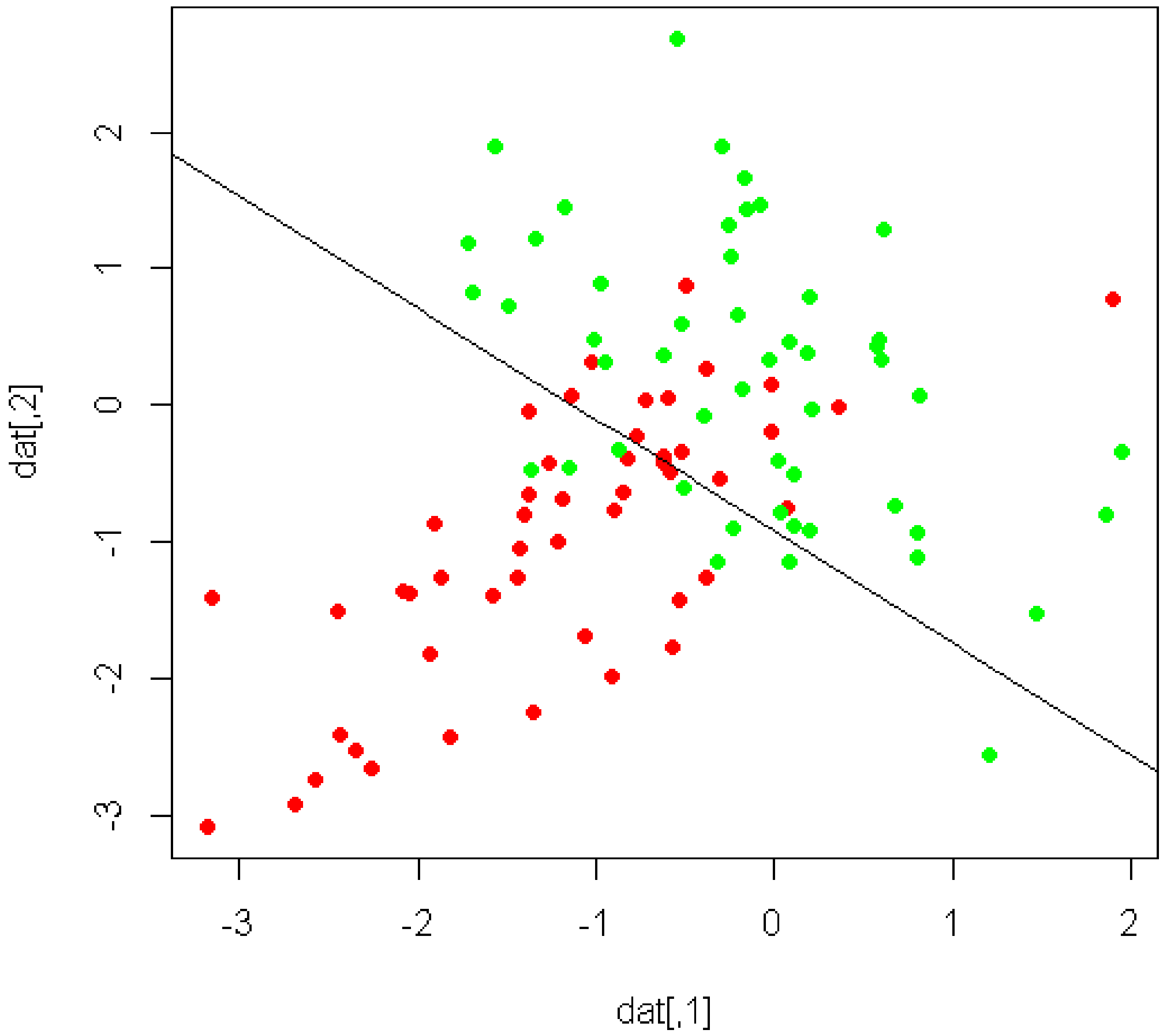
Fisher (1936): Assume the class densities $f_j(\boldsymbol{x})$ are multivariate normal and they all have the same variance-covariance matrix $\Sigma$.

This leads to a *linear* boundary between any two classes [calculation at board]

*DISCRIMINANT*

$$a_1 x_1 + a_2 x_2 + \ \ldots + a_n x_n + b = 0$$


4
2
0
-2
-4
-6
-4
-2
0
2
4
dat[,2]
dat[,1]


dat[,2]
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
dat[,1]

In the second picture, the linear boundary doesn’t look so accurate. If we do not assume that the variance-covariance matrices of the classes are the same, we get **Quadratic Discriminant Analysis** or **QDA**.

In QDA the boundary between classes can be quadratic: [demonstrate]

$$b_{11}{x_1}^2 + b_{12}x_1x_2 + \cdots + \ a_1x_1 + a_2x_2 + \ \ldots \\ + a_nx_n + b = 0$$

# Question:

Which method (LDA or QDA) has the higher:

… **bias**<br>
… **variance** ?

| birthplace = 0, DF = 49 | | |
|---|---|---|
| Variable | fresh | marine |
| fresh | 260.607755 | -188.092653 |
| marine | -188.092653 | 1399.086122 |

| birthplace = 1, DF = 49 | | |
|---|---|---|
| Variable | fresh | marine |
| fresh | 326.0902041 | 133.5048980 |
| marine | 133.5048980 | 893.2608163 |

**4a.** Briefly explain the assumptions made for the LDA, QDA and logistic regression. Taking into consideration the variance-covariance matrices for Alaskan and Canadian salmon above and the type of independent variables, which method(s) would you consider to be most suitable here and why? (3 marks)

# Naïve Bayes

The naïve Bayes classifier is another attempt to approximate the Bayes classifier. Instead of assuming that the class densities are multivariate normal, it assumes that the variables are independent.

$$f_j(\boldsymbol{x})= f_{j1}(\boldsymbol{x_1})\ f_{j2}(\boldsymbol{x_2}) \dots f_{j2}\left(\boldsymbol{x_p}\right)$$

The $f_{jr}(\boldsymbol{x}_r)$ have to be approximated somehow. For discrete data, this is just a question of counting.

Example:

More than 200 killed in Turkish mine disaster
Troops accused of Iraq war crimes
Ukrainian soldiers killed in ambush
GSK bribery case targets Briton

Iraq torture probe will cost us 31m
At least 157 miners killed and many more trapped underground
Russia kills off International Space Station over Ukraine sanctions – as six soldiers are killed by seperatist militants
Pope: I'd baptise Martians if they arrive and asked for it.

Test case: Death toll soars to 150 after Turkish mine explosion – and could rise further

|  | G | DM | prob.G | prob.DM |
|---|---|---|---|---|
| Death |  |  |  |  |
| Toll |  |  |  |  |
| Rises |  |  |  |  |
| To |  |  |  |  |
| 150 |  |  |  |  |
| After |  |  |  |  |
| Turkish | 1 | 0 | 2/3 | 1/3 |
| Mine | 1 | 0 | 2/3 | 1/3 |
| Explosion |  |  |  |  |
| And | 0 | 1 | 1/3 | 2/3 |
| Could |  |  |  |  |
| Rise |  |  |  |  |
| Further |  |  |  |  |
| Product: |  |  | 4/27 | 2/27 |

Classify as G.

The naïve Bayes classifier is a stupid idea that can work very well in real life when p is large. Example: CoIL challenge 2000 (n=5822, p=85) caravan insurance; 1st and 2nd prizes were naïve Bayes classifiers.

Important point: outside the classroom, the “assumptions” don’t have to be satisfied for the method to work. In practice, all that matters is whether it works or not.