Lecture 12: A probabilistic approach to classification

Iain Styles

8 November 2018

Learning Outcomes

By then end of this lecture you should be able to:

- Understand the solutions to the second assignment and how you could improve
- Understand how **I made a mistake** and how I am trying to correct it
- Explain multiclass classification with LDA and understand how it forms its decision boundaryies

Assignment 2

- Very well done by most people
- Here's how I approached this: https://drive.google.com/open?id=1fQ-8SUYqD8ASeEjABuEXWRi23OFiGIzu

Confession

- The theory of k-nearest neighbours and random projection is all correct...

Confession

- The theory of k-nearest neighbours and random projection is all correct...
- But the examples are all wrong

Confession

- The theory of k-nearest neighbours and random projection is all correct...
- But the examples are all wrong
- Why? Lets' look at the code:

```
def knn(test_set, train_set, train_labels, k):
  ### Returns the most common label in the
      training set of the k-nn for each element in
       the test set.
  predictions = []
  for i in test_set:
    distances = [np.linalg.norm(i-j) for j in
        train_set]
    indices = np.argsort(distances)[0:k]
    predictions.append(mode(train_labels[indices])
        [0][0])
  return predictions
```

What's wrong?

- Possibly the simplest piece of the code:

What's wrong?

- Possibly the simplest piece of the code: `i-j`

What's wrong?

- Possibly the simplest piece of the code: `i-j`
- Why?

What's wrong?

- Possibly the simplest piece of the code: `i-j`
- Why?
- Datatypes. . .

What's wrong?

- Possibly the simplest piece of the code: `i-j`
- Why?
- Datatypes. . .
- MNIST data is `uint8` – 8-bit unsigned integers, range [0,255]

What's wrong?

- Possibly the simplest piece of the code: `i-j`
- Why?
- Datatypes...
- MNIST data is `uint8` – 8-bit unsigned integers, range [0,255]
- Result of arithmetic is `uint8`

What's wrong?

- Possibly the simplest piece of the code: `i-j`
- Why?
- Datatypes...
- MNIST data is `uint8` – 8-bit unsigned integers, range [0,255]
- Result of arithmetic is `uint8`
- Truncated at 0 below and 255 above
- So distance calculations are wrong...
- The perils of dynamically typed languages...

What's wrong?

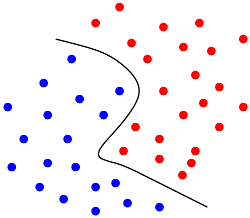
- Possibly the simplest piece of the code: `i-j`
- Why?
- Datatypes...
- MNIST data is `uint8` – 8-bit unsigned integers, range [0,255]
- Result of arithmetic is `uint8`
- Truncated at 0 below and 255 above
- So distance calculations are wrong...
- The perils of dynamically typed languages...
- Demonstrations of dimensionality reduction are not quite as convincing as I had hoped...
- I am working on a revised examples – hopefully next Tuesday.

Recap: Linear Discriminant Analysis

- Build statistical models of classes

Recap: Linear Discriminant Analysis

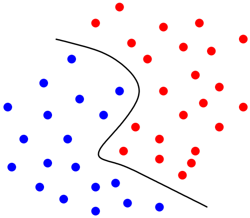
- Build statistical models of classes
- Construct explicit decision boundary



- Uses the probability distributions of the classes to find the line of equal probability dividing two classes

Recap: Linear Discriminant Analysis

- Build statistical models of classes
- Construct explicit decision boundary



- Uses the probability distributions of the classes to find the line of equal probability dividing two classes

Formulating LDA

- Bayes rule:

$$P(\Pi_i|\mathbf{x}) = \frac{P(\mathbf{x}|\Pi_i)P(\Pi_i)}{P(\mathbf{x})} \tag{1}$$

$$= \frac{f_i(\mathbf{x})\pi_i}{f_1(\mathbf{x})\pi_1 + f_2(\mathbf{x})\pi_2} \tag{2}$$

Formulating LDA

- Bayes rule:

$$P(\Pi_i|\mathbf{x}) = \frac{P(\mathbf{x}|\Pi_i)P(\Pi_i)}{P(\mathbf{x})} \tag{1}$$

$$= \frac{f_i(\mathbf{x})\pi_i}{f_1(\mathbf{x})\pi_1 + f_2(\mathbf{x})\pi_2} \tag{2}$$

- New points are assigned to the group with the highest probability:

$$\frac{P(\Pi_1|\mathbf{x})}{P(\Pi_2|\mathbf{x})} > 1 \mapsto \mathbf{x} \in \Pi_1 \quad \text{else}\, \mathbf{x} \in \Pi_2$$

Formulating LDA

- Bayes rule:

$$P(\Pi_i|\mathbf{x}) = \frac{P(\mathbf{x}|\Pi_i)P(\Pi_i)}{P(\mathbf{x})} \tag{1}$$

$$= \frac{f_i(\mathbf{x})\pi_i}{f_1(\mathbf{x})\pi_1 + f_2(\mathbf{x})\pi_2} \tag{2}$$

- New points are assigned to the group with the highest probability:

$$\frac{P(\Pi_1|\mathbf{x})}{P(\Pi_2|\mathbf{x})} > 1 \mapsto \mathbf{x} \in \Pi_1 \quad \text{else}\, \mathbf{x} \in \Pi_2$$

- In terms of f and π, $\mathbf{x}$ belongs to Π_1 if

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{\pi_2}{\pi_1}$$

  [Note error in last lecture's slide, notes are correct]
- otherwise to Π_2.

Formulating LDA

- Bayes rule:

$$P(\Pi_i|\mathbf{x}) = \frac{P(\mathbf{x}|\Pi_i)P(\Pi_i)}{P(\mathbf{x})} \tag{1}$$

$$= \frac{f_i(\mathbf{x})\pi_i}{f_1(\mathbf{x})\pi_1 + f_2(\mathbf{x})\pi_2} \tag{2}$$

- New points are assigned to the group with the highest probability:

$$\frac{P(\Pi_1|\mathbf{x})}{P(\Pi_2|\mathbf{x})} > 1 \mapsto \mathbf{x} \in \Pi_1 \quad \text{else}\, \mathbf{x} \in \Pi_2$$

- In terms of f and π, $\mathbf{x}$ belongs to Π_1 if

$$\frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} > \frac{\pi_2}{\pi_1}$$

  [Note error in last lecture's slide, notes are correct]
- otherwise to Π_2.
- Equal ratios, randomly assign to either class.

Multiclass LDA

- Generalisation to multiclass data require significantly more algebra. . . .

Multiclass LDA

- Generalisation to multiclass data require significantly more algebra. . . .
- Compute the pairwise relative probabilities as before and form the discriminant

$$L_{ij}(\mathbf{x}) = \log_e \left(\frac{P(\Pi_i|\mathbf{x})}{P(\Pi_j|\mathbf{x}} \right) = \log_e \left(\frac{f_i(\mathbf{x})\pi_i}{f_j(\mathbf{x})\pi_j} \right)$$

Multiclass LDA

- Generalisation to multiclass data require significantly more algebra. . . .
- Compute the pairwise relative probabilities as before and form the discriminant

$$L_{ij}(\mathbf{x}) = \log_e\left(\frac{P(\Pi_i|\mathbf{x})}{P(\Pi_j|\mathbf{x}}\right) = \log_e\left(\frac{f_i(\mathbf{x})\pi_i}{f_j(\mathbf{x})\pi_j}\right)$$

- $\mathbf{x}$ is assigned to Π_i if $L_{IJ} > 0$ for all $j \neq i$.
- The discriminant function between classes i and j is then

$$L_{ij}(\mathbf{x}) = \mathbf{m}_{ij}^{\mathrm{T}}\mathbf{x} + c_{ij}$$

with

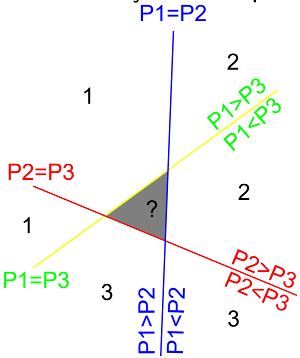
$$\begin{aligned} \mathbf{m}_{ij} &= \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^{\mathrm{T}} \mathbf{\Sigma}^{-1} \text{ and} \\ c_{ij} &= -\frac{1}{2}\left(\bar{\mathbf{x}}_i^{\mathrm{T}}\mathbf{\Sigma}^{-1}\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j^{\mathrm{T}}\mathbf{\Sigma}^{-1}\bar{\mathbf{x}}_j\right) + \log_e\frac{\pi_i}{\pi_j}. \end{aligned}$$

Multiclass LDA: Practicalities

- Consistency of multiple boundaries?

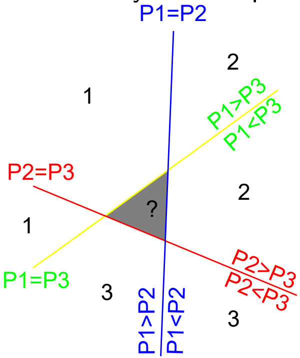
Multiclass LDA: Practicalities

- Consistency of multiple boundaries?



Multiclass LDA: Practicalities

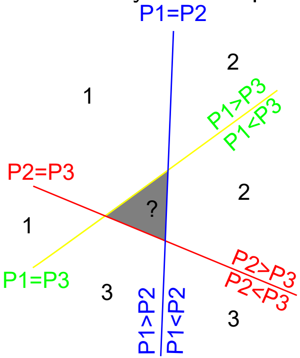
- Consistency of multiple boundaries?



- Pairwise comparisons across all boundaries is the same as taking the class with the max posterior probability

Multiclass LDA: Practicalities

- Consistency of multiple boundaries?



- Pairwise comparisons across all boundaries is the same as taking the class with the max posterior probability
- Let's see how it works. . . `https://colab.research.google.com/drive/1zXjLRI2qhvKoeG_ZAA5xPdWFDc8IENwq`

Summary

- LDA is a statistical way of classifying data

Summary

- LDA is a statistical way of classifying data
- Explicitly construct class-conditional PDFs

Summary

- LDA is a statistical way of classifying data
- Explicitly construct class-conditional PDFs
- Resample from PDFs for generative modelling

Summary

- LDA is a statistical way of classifying data
- Explicitly construct class-conditional PDFs
- Resample from PDFs for generative modelling
- Strong assumptions – do they always matter?

Summary

- LDA is a statistical way of classifying data
- Explicitly construct class-conditional PDFs
- Resample from PDFs for generative modelling
- Strong assumptions – do they always matter?
- Next time: a purely discriminative probabilistic method: logistic regression