Lecture 16: Unsupervised Learning: Mixture Models

And a return to dimensionality reduction. . .

Iain Styles

29 November 2019

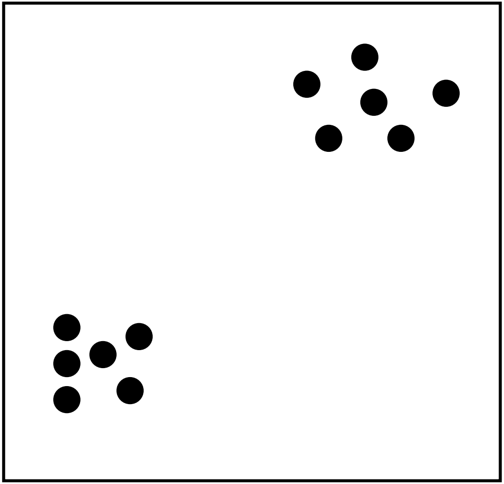
Learning Outcomes

By the end of this lecture you should

- Understand the principles of Gaussian mixture modelling
- Understand how dimensionality reduction can improve classifier performance.

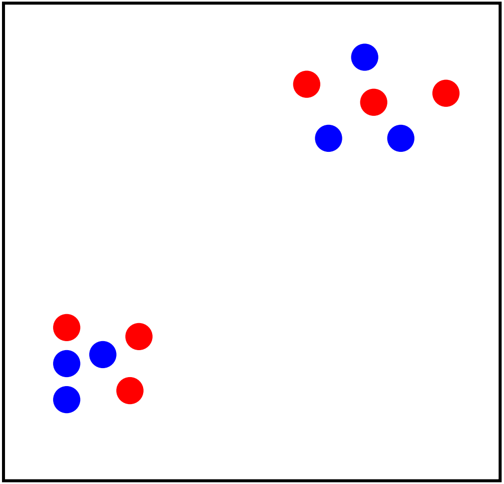
Recap: The k-means Algorithm

Initial data points



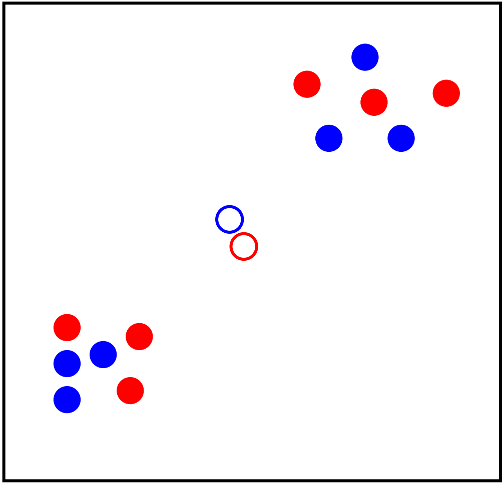
Recap: The k-means Algorithm

Randomly assign points to groups

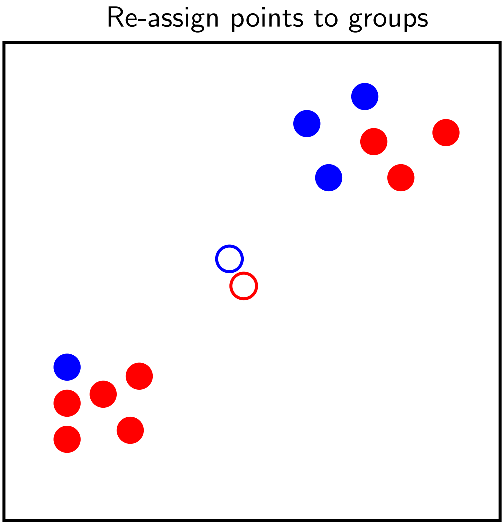


Recap: The k-means Algorithm

Compute group average

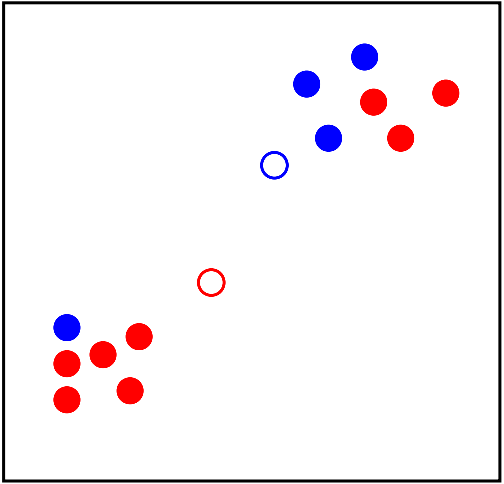


Recap: The k-means Algorithm



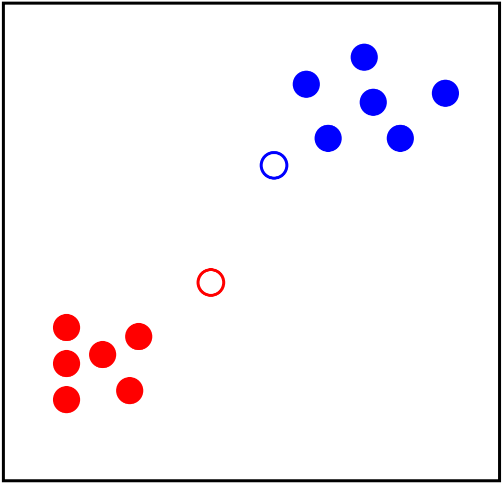
Recap: The k-means Algorithm

Re-compute group averages



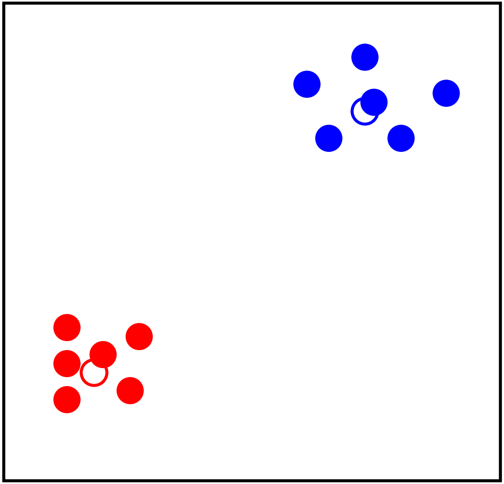
Recap: The k-means Algorithm

Re-assign points to groups

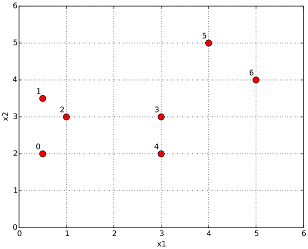


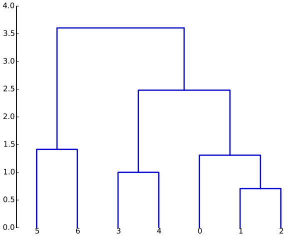
Recap: The k-means Algorithm

Re-compute group averages



Recap: Agglomerative Clustering





Mixture Models

- k-means and hierarchical clustering are heuristic
- "Hard" clustering assignments
- Weak statistical assumptions
- No data model makes theory difficult

Mixture Models

- k-means and hierarchical clustering are heuristic
- "Hard" clustering assignments
- Weak statistical assumptions
- No data model makes theory difficult
- A statistically principled method would be good:
- Mixture Models

Mixture Models

- k-means and hierarchical clustering are heuristic
- "Hard" clustering assignments
- Weak statistical assumptions
- No data model makes theory difficult
- A statistically principled method would be good:
- Mixture Models
- Assume data is generated by a statistical process that is a mixture of components

Mixture Models

- k-means and hierarchical clustering are heuristic
- "Hard" clustering assignments
- Weak statistical assumptions
- No data model makes theory difficult
- A statistically principled method would be good:
- Mixture Models
- Assume data is generated by a statistical process that is a mixture of components
- Clustering: which component generated the data point

Gaussian Mixture Models

- We will consider Gaussian Mixture Models

Gaussian Mixture Models

- We will consider Gaussian Mixture Models
- Assume that the data is drawn from a probability density function of the form

$$p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

Gaussian Mixture Models

- We will consider Gaussian Mixture Models
- Assume that the data is drawn from a probability density function of the form

$$p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

- The sum of K discrete Gaussian clusters
- Clustering is then formulated as
  1. Learn the parameters of the GMM which best describe the data.
  2. Determine from which component a data point is most likely to have been generated.

Learning Gaussian Mixture Models

- Some observations about $p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Learning Gaussian Mixture Models

- Some observations about $p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- Since $\mathcal{N}(\cdot)$ and $p(\mathbf{x})$ are both normalised and strictly positive it can easily be shown that the *mixing coefficients* satisfy

$$\sum_{k=1}^{K} A_k = 1 \quad \text{and} \quad 0 \leq A_k \leq 1 \tag{2}$$

Learning Gaussian Mixture Models

- Some observations about $p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- Since $\mathcal{N}(\cdot)$ and $p(\mathbf{x})$ are both normalised and strictly positive it can easily be shown that the *mixing coefficients* satisfy

$$\sum_{k=1}^{K} A_k = 1 \quad \text{and} \quad 0 \leq A_k \leq 1 \tag{2}$$

Learning Gaussian Mixture Models

- We also note that by sum and product rules

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k) \tag{3}$$

Learning Gaussian Mixture Models

- We also note that by sum and product rules

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k) \tag{3}$$

and by comparison with

$$p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

Learning Gaussian Mixture Models

- We also note that by sum and product rules

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(k)p(\mathbf{x}|k) \tag{3}$$

and by comparison with

$$p(\mathbf{x}) = \sum_{k=1}^{K} A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{4}$$

we interpret

- $A_k = p(k)$ as the prior probability of choosing a point from component k
- $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\mathbf{x}|k)$ as the class-conditional likelihoods.

Learning Gaussian Mixture Models

- Finally, using Bayes' theorem we compute the posterior *responsibilities*

$$
\begin{aligned}
r_k(\mathbf{x}) &= p(k|\mathbf{x}) && (5) \\
&= \frac{p(k)p(\mathbf{x}|k)}{\sum_{k'=1}^{K} p(k')p(\mathbf{x}|k')} && (6) \\
&= \frac{A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} A_{k'} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} && (7)
\end{aligned}
$$

Learning Gaussian Mixture Models

- Finally, using Bayes' theorem we compute the posterior *responsibilities*

$$
\begin{aligned}
r_k(\mathbf{x}) &= p(k|\mathbf{x}) && (5)\\
&= \frac{p(k)p(\mathbf{x}|k)}{\sum_{k'=1}^{K} p(k')p(\mathbf{x}|k')} && (6)\\
&= \frac{A_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} A_{k'} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} && (7)
\end{aligned}
$$

- Probability that component k explains $\mathbf{x}$
- GMM gives *soft* cluster assignments

`https://colab.research.google.com/drive/1Gta0oTu_86UFkAMHqrhpQ7EdKrgmaPXW`

Recap: Dimensionality Reduction

- Core idea: high dimensional data problematic, but...

Recap: Dimensionality Reduction

- Core idea: high dimensional data problematic, but. . .
- **Earlier discussions about this were incorrect due to a bug. . .**
  - Volume of space is near the "corners"
  - Distance functions converge
  - Computations are expensive

Recap: Dimensionality Reduction

- Core idea: high dimensional data problematic, but…
- **Earlier discussions about this were incorrect due to a bug…**
  - Volume of space is near the "corners"
  - Distance functions converge
  - Computations are expensive
- Is it problematic in practice?
  - Convergence of distances is, but only in data that lacks structure
  - Data is not usually uniformly distributed
  - Computational expense is always a problem – comparing two 10,000-dimensional vectors is expensive

Recap: Johnson-Lindenstrass Lemma

- Given a set X of N data points in $\mathbb{R}^M$...

Recap: Johnson-Lindenstrass Lemma

- Given a set X of N data points in $\mathbb{R}^M$...
- There is a linear map $f : \mathbb{R}^M \mapsto \mathbb{R}^K$ where $K < M$

Recap: Johnson-Lindenstrass Lemma

- Given a set X of N data points in $\mathbb{R}^M$...
- There is a linear map $f : \mathbb{R}^M \mapsto \mathbb{R}^K$ where $K < M$
- The map obeys

$$1 - \varepsilon \leq \frac{\|f(x_1) - f(x_2)\|^2}{\|x_1 - x_2\|^2} \leq 1 + \varepsilon \tag{8}$$

for all $x_1, x_2 \in X$ and for $0 < \varepsilon < 1$ and $K > 8 \ln(N)/\varepsilon^2$.

Recap: Johnson-Lindenstrass Lemma

- Given a set X of N data points in $\mathbb{R}^M$...
- There is a linear map $f : \mathbb{R}^M \mapsto \mathbb{R}^K$ where $K < M$
- The map obeys

$$1 - \varepsilon \leq \frac{\|f(x_1) - f(x_2)\|^2}{\|x_1 - x_2\|^2} \leq 1 + \varepsilon \tag{8}$$

  for all $x_1, x_2 \in X$ and for $0 < \varepsilon < 1$ and $K > 8 \ln(N)/\varepsilon^2$.
- f preserves relative distances
- Holds when "f is a random, orthonormal linear transformation"

Why does this matter?

- Most data has structure so convergence of distances is not a practical obstacle to learning.
- But computational expensive is.

Why does this matter?

- Most data has structure so convergence of distances is not a practical obstacle to learning.
- But computational expensive is.
- Johnson-Lindenstrass provides a cheap, data-independent way to reduce the dimensionality.
- Distance calculations can be sped up dramatically.

https://colab.research.google.com/drive/1HOoR2sHwKefc5a-AIePoX3u7C9H-GsJL

Summary

- Dimensionality does not necessarily improve classifier accuracy
- But a low-cost classifier such as random projection can dramatically reduce computational cost without affecting accuracy
- Important in practical applications of classifiers
- Next (last) lecture: a few words on fairness and bias in machine learning