Outline

Administrivia and Introduction
- Course Structure
- Syllabus
- Introduction to Data Mining

Dimensionality Reduction
- Introduction
- Principal Components Analysis
- Singular Value Decomposition
- Multidimensional Scaling
- Isomap

Clustering
- Introduction
- Hierarchical Clustering
- K-means
- Vector Quantisation
- Probabilistic Methods

Probabilistic Methods

- So far, we have found clusters in high-dimensional data by posing sensible partition based problems and hierarchical clustering problems which were tackled with heuristic approaches.
- Probabilistic methods attempt to find clusters in high-dimensional data using a model based approach by fitting mixture models to data.
- Though well founded in probabilistic arguments, such an approach comes at the expense of greater computation.
- Such methods can work well if good models are proposed (or if the distribution of the data is close to the proposed model in a suitable sense).
- We again need to specify/estimate the number of clusters K.

Mixture Models

- Probabilistic methods for clustering work by seeking to model the distribution of points in $\mathbb{R}^p$ using mixture models. In doing so, areas of high density (i.e. clusters) can be accurately described.
- Mixture models have densities of the form

$$f(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k f(\mathbf{x}|\phi_k)$$

  for some densities $f_k(\mathbf{x}|\phi_k)$ and priors over these densities $\pi_1, \ldots, \pi_K$ which satisfy $\pi_k \geq 0 \; \forall k$ and $\sum_{k=1}^{K} \pi_k = 1$.
- We want to estimate the unknown parameters $\theta = \{\pi_k, \phi_k\}_{k=1}^{K}$ given $\mathbf{x}_{1:n}$.

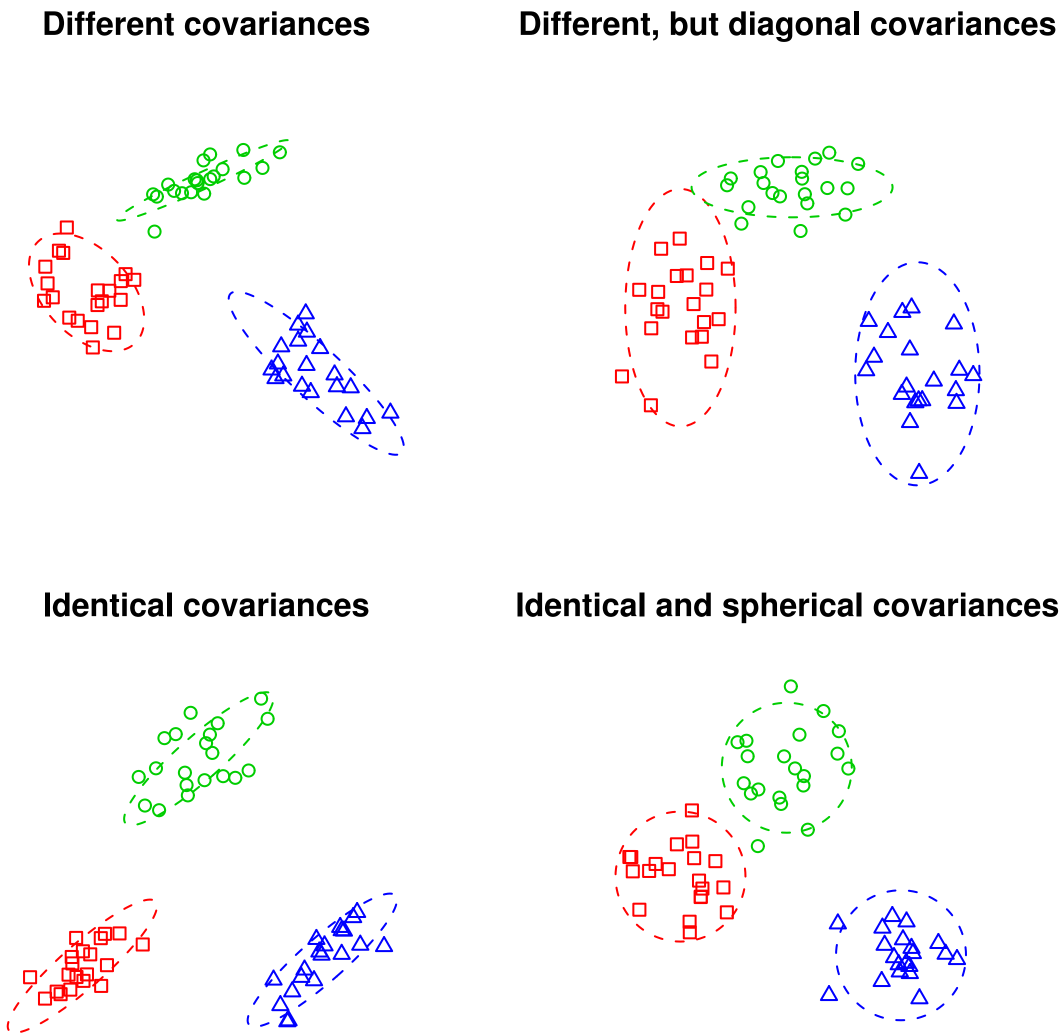
Mixture Models

- To make things easier, let $f(\mathbf{x}|\theta_k) = f(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \sim N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ where

$$f(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^p \cdot |\boldsymbol{\Sigma}_k|}} \exp\left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}.$$

- Posing a Gaussian Mixture Model corresponds to assuming that each of the K clusters that we intend to model...
  - is Gaussian with different means $\boldsymbol{\mu}_k$ and covariance structures $\boldsymbol{\Sigma}_k$.
  - and each observation $\mathbf{x}$ comes from cluster k with probability π_k.
- Allowing each cluster to have its own mean and covariance structure allows greater flexibility in the model.

Gaussian Mixture Models: Examples



Fitting Gaussian Mixture Models

- To fit such a model, we need to estimate the parameters

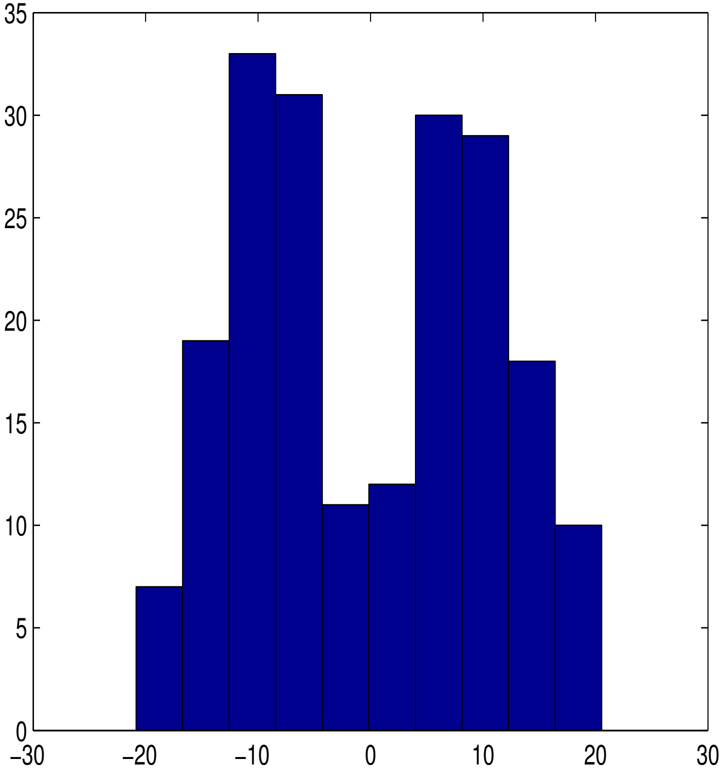
$$\theta = \{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

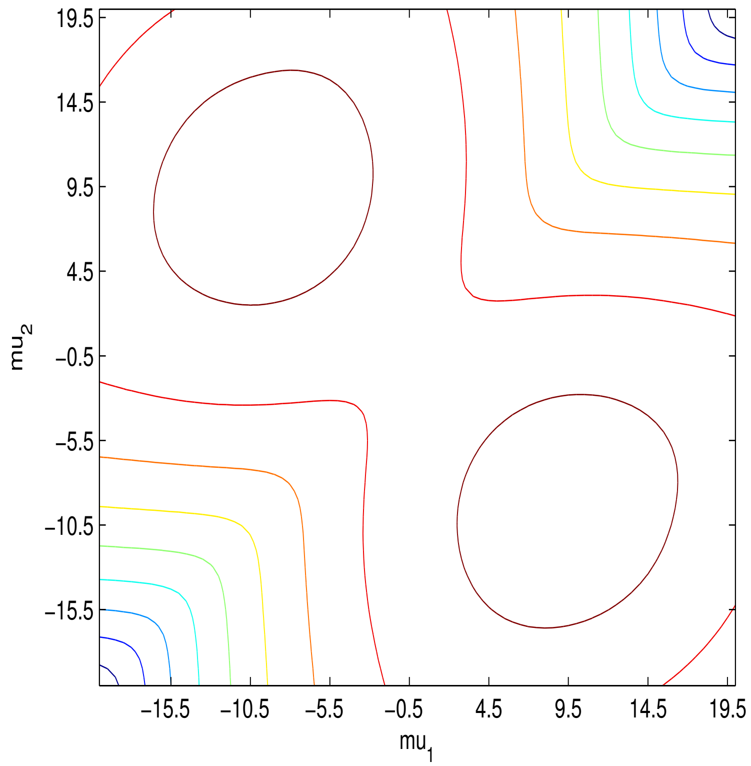
  from the data.
- We can do this by maximum likelihood choosing θ to maximise $L(\theta) = \prod_{i=1}^{n} f(\mathbf{x}_i|\theta)$ or equivalently $\ell(\theta) = \sum_{i=1}^{n} \log f(\mathbf{x}_i|\theta)$ where

$$\ell(\theta) = \sum_{i=1}^{n} \log \left(\pi_1 f_{\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1}(\mathbf{x}_i) + \ldots + \pi_K f_{\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K}(\mathbf{x}_i) \right).$$

- Differentiating to maximise such a log-likelihood analytically or even numerically is difficult as there are too many unknowns to handle simultaneously.
- The Expectation-Maximisation (EM) Algorithm is a very popular method to help find maximum likelihood estimates in the presence of unobserved variables.

Likelihood Surface for a Simple Example





(left) $n = 200$ data points from a mixture of two 1D Gaussians with $\pi_1 = \pi_2 = 0.5$, $\sigma_1 = \sigma_2 = 5$ and $\mu_1 = -\mu_2 = 10.$ (right) Log-Likelihood surface $\ell\left(\mu_1, \mu_2\right)$, all the other parameters being assumed known.

The EM Algorithm

- EM is a very popular approach to maximize $\ell(\theta)$ in this missing data context.
- The key idea is to introduce explicitly the unobserved cluster labels z_i which indicate from which cluster data $\mathbf{x}_i$ is coming from.
- If the cluster labels where known then we would estimate θ by maximizing the so-called complete likelihood

$$\begin{aligned}\ell_c(\theta) &= \sum_{i=1}^{n} \log p(\mathbf{x}_i, z_i | \theta) \\ &= \sum_{i=1}^{n} \log \pi_{z_i} f(\mathbf{x}_i | \phi_{z_i})\end{aligned}$$

Maximization of Complete Likelihood

- We have

$$\begin{aligned}\ell_c(\theta) &= \sum_{k=1}^{K}\left(\sum_{i:z_i=k}\log\ \pi_{z_i} f(\mathbf{x}_i|\phi_{z_i})\right)\\ &= \sum_{k=1}^{K} n_k \log(\pi_k) + \sum_{i:z_i=k}\log\ f(\mathbf{x}_i|\phi_k)\end{aligned}$$

  where $n_k = \sum_{i:z_i=k} 1$ is the number of observations assigned to cluster k.
- We would obtain the MLE for the complete likelihood

$$\begin{aligned}\widehat{\pi}_k =& \frac{n_k}{n},\\ \widehat{\phi}_k =& \arg\max_{\phi_k} \sum_{i=1:z_i=k}^{n} \log\ f(\mathbf{x}_i|\phi_k)\end{aligned}$$

Finite Mixture of Scalar Gaussians

- In this case, $\phi = \left(\mu, \sigma^2\right)$

$$f\left(\mathbf{x}|\phi\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(\mathbf{x}-\mu\right)^2}{2\sigma^2}\right)$$

  and $\theta = \left\{\pi_k, \mu_k, \sigma^2\right\}_{k=1}^{K}$.
- The resulting MLE estimate of the complete likelihood is

$$\begin{aligned}
\widehat{\pi}_k &= \frac{n_k}{n}, \\
\widehat{\mu}_k &= \frac{1}{n_k} \sum_{i=1:z_i=k}^{n} \mathbf{x}_i, \\
\widehat{\sigma}_k^2 &= \frac{1}{n_k} \sum_{i=1:z_i=k}^{n} \left(\mathbf{x}_i - \widehat{\mu}_k\right)^2
\end{aligned}$$

- Problem: We don't have access to the cluster labels!

Expectation-Maximization

EM is an iterative algorithm which generates a sequence of estimates $\left\{\theta^{(t)}\right\}$ such that

$$\ell\left(\theta^{(t)}\right) \geq \ell\left(\theta^{(t-1)}\right).$$

At iteration t, we compute

$$\begin{aligned}
&\mathcal{F}\left(\theta, \theta^{(t-1)}\right) \\
=&\mathbb{E}\left[\ell_c(\theta)\middle|\, \mathbf{x}_{1:n}, \theta^{(t-1)}\right] \\
=&\sum_{z_{1:n}\in\{1,2,\ldots,K\}^n} p\left(z_{1:n}\middle|\, \mathbf{x}_{1:n}, \theta^{(t-1)}\right)\left(\sum_{i=1}^{n}\log\, p\left(\mathbf{x}_i, z_i\middle|\, \theta\right)\right) \\
=&\sum_{i=1}^{n}\sum_{k=1}^{K} p\left(z_i = k\middle|\, \mathbf{x}_i, \theta^{(t-1)}\right)\log\, p\left(\mathbf{x}_i, z_i = k\middle|\, \theta\right)
\end{aligned}$$

and set

$$\theta^{(t)} = \arg\max_{\theta} \mathcal{F}\left(\theta, \theta^{(t-1)}\right)$$

Expectation-Maximization

We have

$$\begin{aligned}
\mathcal{F}\left(\theta, \theta^{(t-1)}\right) &= \sum_{i=1}^{n} \sum_{k=1}^{K} p\left(z_i = k \middle| \mathbf{x}_i | \theta^{(t-1)}\right) \log\ p\left(\mathbf{x}_i, z_i = k \middle| \theta\right) \\
&= \sum_{i=1}^{n} \sum_{k=1}^{K} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right) \left\{\log\ \pi_k + \ \log f\left(\mathbf{x}_i | \phi_k\right)\right\} \\
&= \sum_{k=1}^{K} \left(\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)\right) \left\{\log \pi_k + \log f\left(\mathbf{x}_i | \phi_k\right)\right\}
\end{aligned}$$

We obtain

$$\begin{aligned}
\widehat{\pi}_k^{(t)} &= \frac{\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)}{n}, \\
\phi_k^{(t)} &= \arg\max_{\phi_k} \sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)\ \log f\left(\mathbf{x}_i | \phi_k\right)
\end{aligned}$$

Finite mixture of scalar Gaussians

In this case, the EM algorithm iterates

$$\widehat{\pi}_k^{(t)} = \frac{\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)}{n}$$

$$\widehat{\mu}_k^{(t)} = \frac{\sum_{i=1}^{n} \mathbf{x}_i p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)}{\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)},$$

$$\widehat{\sigma}_k^{2\ (t)} = \frac{\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right) \left(\mathbf{x}_i - \widehat{\mu}_k^{(t)}\right)^2}{\sum_{i=1}^{n} p\left(z_i = k \middle| \mathbf{x}_i, \theta^{(t-1)}\right)}.$$

with

$$p\left(z_i = k | \mathbf{x}_i, \theta\right) = \frac{\pi_k f\left(\mathbf{x}_i | \phi_k\right)}{\sum_{\ell} \pi_\ell f\left(\mathbf{x}_i | \phi_\ell\right)}$$

Proof of Convergence for EM Algorithm

Proposition: $\ell\left(\theta^{(t+1)}\right) \geq \ell\left(\theta^{(t)}\right)$ for $\theta^{(t+1)} = \arg\max_{\theta} \mathcal{F}\left(\theta, \theta^{(t)}\right)$.

Proof: We have

$$p\left(z_{1:n} \mid \theta, \mathbf{x}_{1:n}\right) = \frac{p\left(\mathbf{x}_{1:n}, z_{1:n} \mid \theta\right)}{p\left(\mathbf{x}_{1:n} \mid \theta\right)} \Leftrightarrow p\left(\mathbf{x}_{1:n} \mid \theta\right) = \frac{p\left(\mathbf{x}_{1:n}, z_{1:n} \mid \theta\right)}{p\left(z_{1:n} \mid \theta, \mathbf{x}_{1:n}\right)}$$

thus

$$\ell(\theta) = \log p\left(\mathbf{x}_{1:n} \mid \theta\right) = \log p\left(\mathbf{x}_{1:n}, z_{1:n} \mid \theta\right) - \log p\left(z_{1:n} \mid \theta, \mathbf{x}_{1:n}\right)$$

and for any value $\theta^{(t)}$

$$\begin{aligned} \ell(\theta) \quad = \quad & \underbrace{\sum_{z_{1:n}} p\left(z_{1:n} \mid \theta^{(t)}, \mathbf{x}_{1:n}\right) \log p\left(\mathbf{x}_{1:n}, z_{1:n} \mid \theta\right)}_{=\mathcal{F}\left(\theta, \theta^{(t)}\right)} \\ & - \sum_{z_{1:n}} p\left(z_{1:n} \mid \theta^{(t)}, \mathbf{x}_{1:n}\right) \log p\left(z_{1:n} \mid \theta, \mathbf{x}_{1:n}\right). \end{aligned}$$

Proof of Convergence for EM Algorithm

We want to show that $\ell\left(\theta^{(t+1)}\right) \geq \ell\left(\theta^{(t)}\right)$ for the EM, so if we prove that

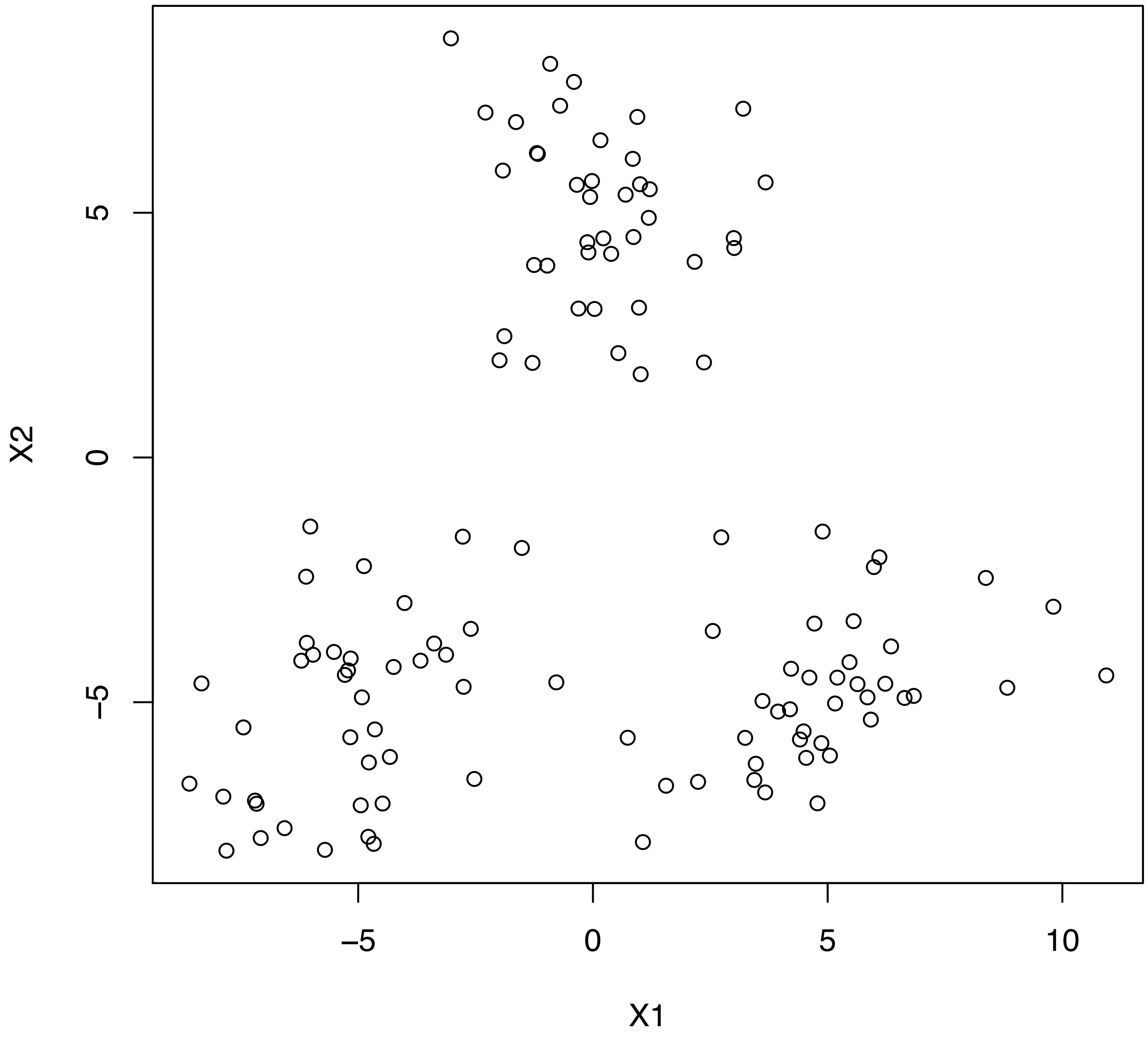
$$\begin{aligned} & \sum_{z_{1:n}} p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right) \log p\left(z_{1:n} \middle| \theta^{(t+1)}, \mathbf{x}_{1:n}\right) \\ \leq & \sum_{z_{1:n}} p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right) \log p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right) \end{aligned}$$

then we are done. We have

$$\begin{aligned} & \sum_{z_{1:n}} p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right) \log \frac{p\left(z_{1:n} \middle| \theta^{(t+1)}, \mathbf{x}_{1:n}\right)}{p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right)} \\ \leq & \log \sum_{z_{1:n}} p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right) \frac{p\left(z_{1:n} \middle| \theta^{(t+1)}, \mathbf{x}_{1:n}\right)}{p\left(z_{1:n} \middle| \theta^{(t)}, \mathbf{x}_{1:n}\right)} \quad \text{(Jensen)} \\ = & \log 1 = 0. \end{aligned}$$

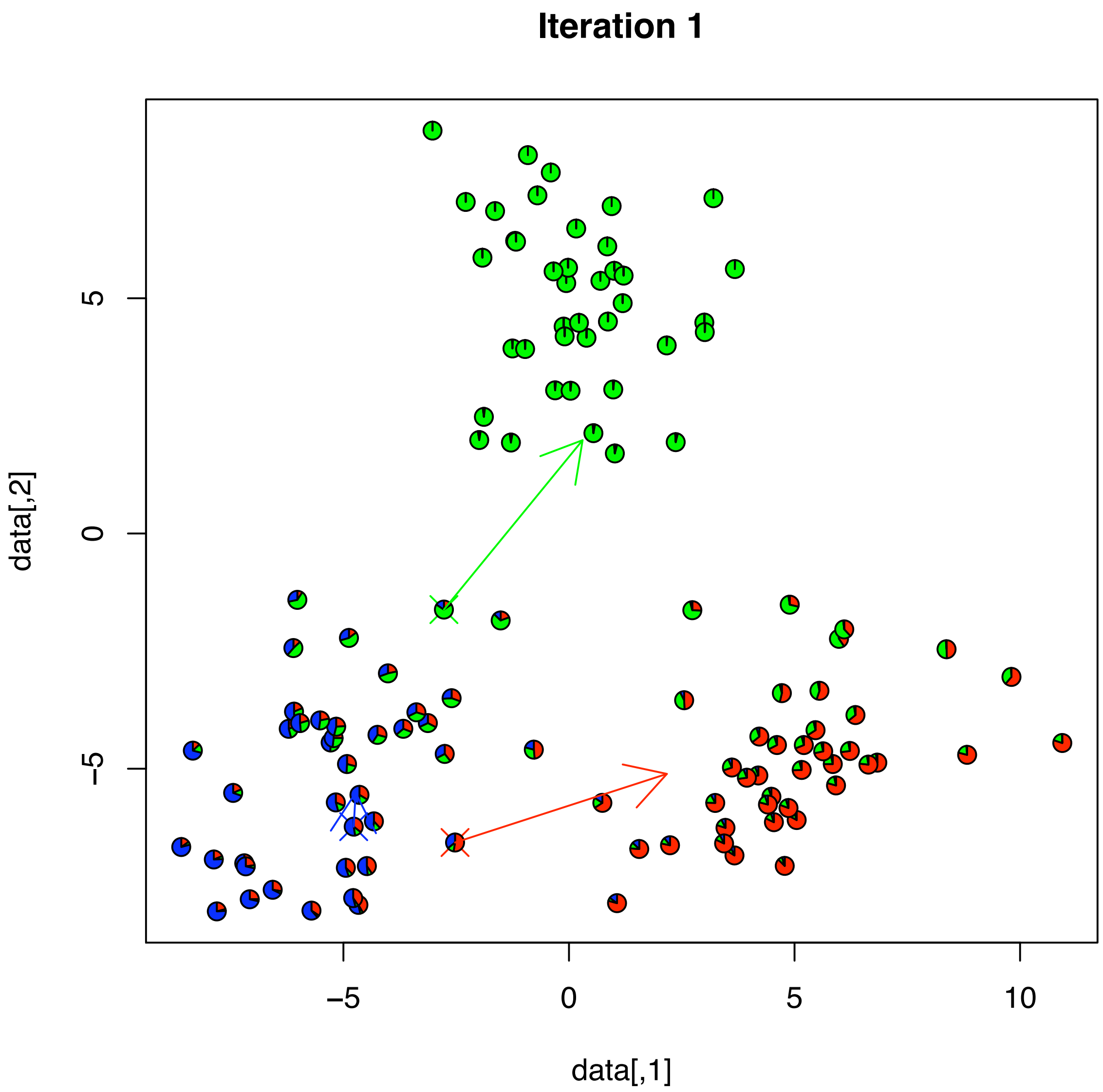
Example: Mixture of 3 Gaussians

An example with 3 clusters.



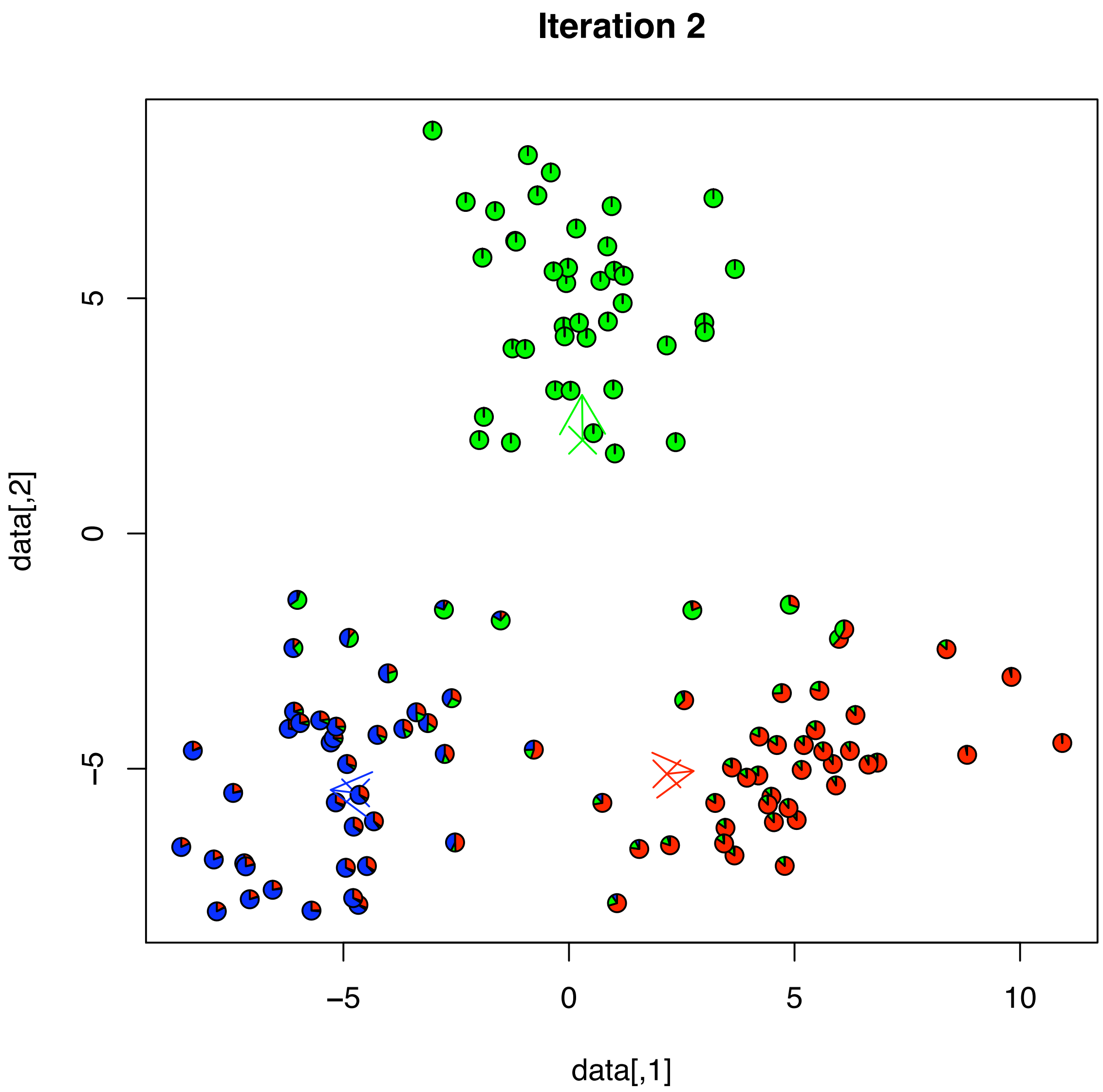
Example: Mixture of 3 Gaussians

After 1st E and M step.



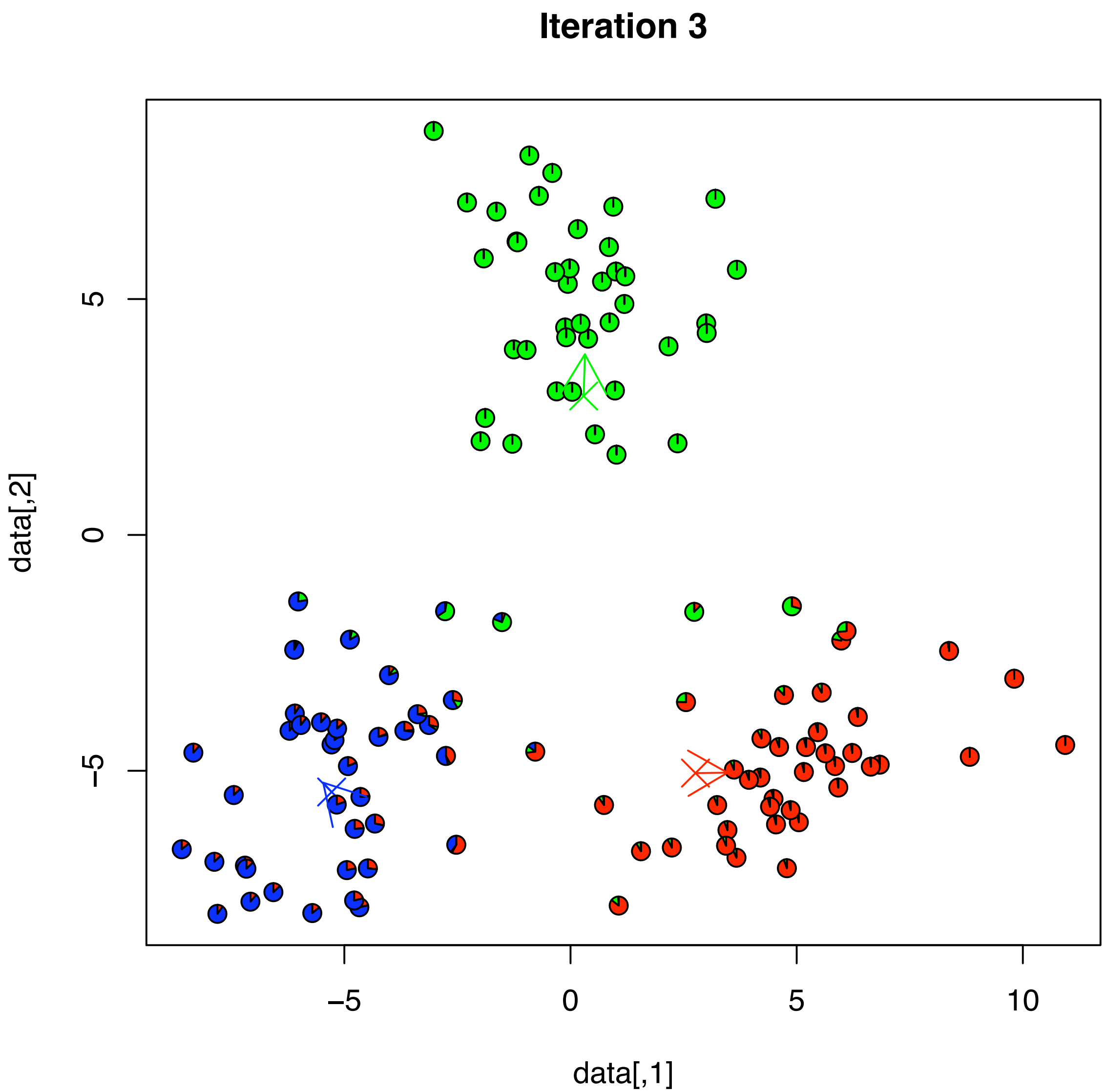
Example: Mixture of 3 Gaussians

After 2nd E and M step.



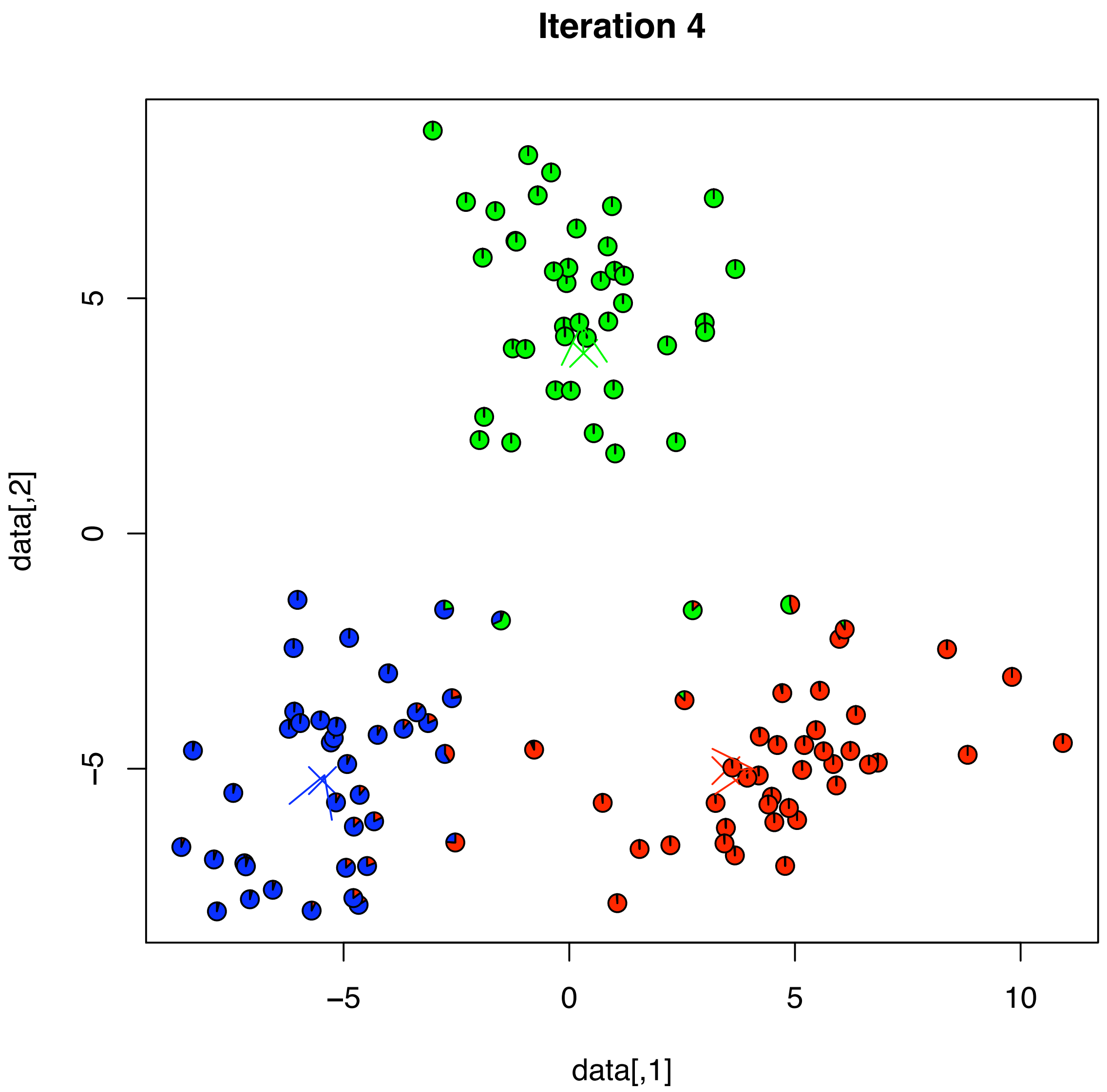
Example: Mixture of 3 Gaussians

After 3rd E and M step.



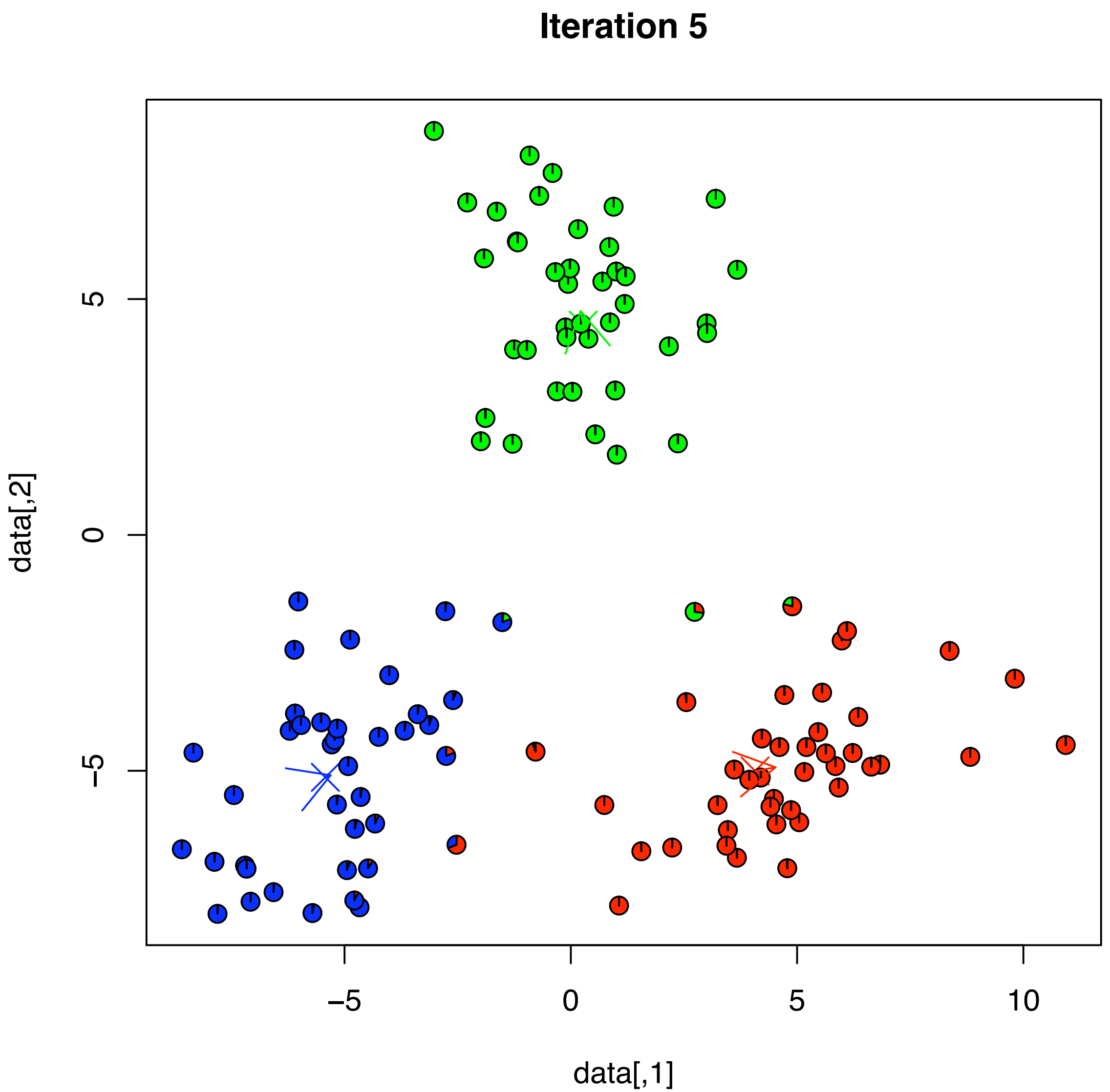
Example: Mixture of 3 Gaussians

After 4th E and M step.



Example: Mixture of 3 Gaussians

After 5th E and M step.



Pros and Cons of the EM Algorithm

Some good things about EM

- no learning rate (step-size) parameter
- automatically enforces parameter constraints
- very fast for low dimensions
- each iteration guaranteed to improve likelihood

Some bad things about EM

- can get stuck in local minima so multiple starts are recommended
- can be slower than conjugate gradient (especially near convergence)
- requires expensive inference step