HT2015: SC4
Statistical Data Mining and Machine Learning

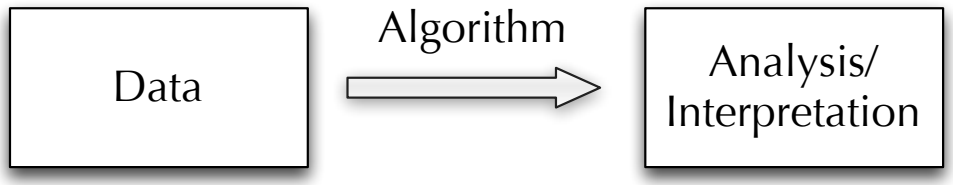
Dino Sejdinovic
Department of Statistics
Oxford

http://www.stats.ox.ac.uk/~sejdinov/sdmml.html

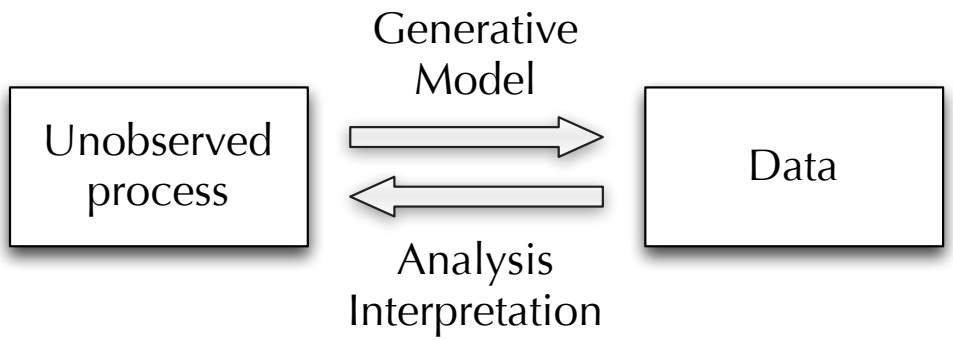
Probabilistic Unsupervised Learning

Probabilistic Methods

- Algorithmic approach:



- Probabilistic modelling approach:



Mixture Models

- Mixture models suppose that our dataset $\mathbf{X}$ was created by sampling iid from K distinct populations (called **mixture components**).
- Typical samples in population k can be modelled using a distribution F_{μ_k} with density $f(x|\mu_k)$. For a concrete example, consider a Gaussian with unknown mean μ_k and known diagonal covariance $\sigma^2 I$,

$$f(x|\mu_k) = |2\pi\sigma^2|^{-\frac{p}{2}} \exp\left(-\frac{1}{2\sigma^2}\|x-\mu_k\|_2^2\right).$$

- Generative model: for $i = 1, 2, \ldots, n$:
  - First determine which population item i came from (independently):

$$Z_i \sim \text{Discrete}(\pi_1, \ldots, \pi_K) \qquad \text{i.e., } \mathbb{P}(Z_i = k) = \pi_k$$

    where **mixing proportions** are $\pi_k \geq 0$ for each k and $\sum_{k=1}^K \pi_k = 1$.
  - If $Z_i = k$, then $X_i = (X_{i1}, \ldots, X_{ip})^\top$ is sampled (independently) from corresponding population distribution:

$$X_i | Z_i = k \sim F_{\mu_k}$$

- We observe that $X_i = x_i$ for each i, and would like to learn about the unknown parameters of the process.

Mixture Models

- Unknowns to learn given data are
  - **Parameters**: $\pi_1, \ldots, \pi_K \in [0,1]$, $\mu_1, \ldots, \mu_K \in \mathbb{R}^p$, as well as
  - **Latent variables**: $z_1, \ldots, z_n$.
- The joint probability over all cluster indicator variables $\{Z_i\}$ are:

$$p_Z((z_i)_{i=1}^n) = \prod_{i=1}^n \pi_{z_i} = \prod_{i=1}^n \prod_{k=1}^K \pi_k^{1(z_i=k)}$$

- The joint density at observations $X_i = x_i$ given $Z_i = z_i$ are:

$$p_X((x_i)_{i=1}^n | (Z_i = z_i)_{i=1}^n) = \prod_{i=1}^n \prod_{k=1}^K f(x_i|\mu_k)^{1(z_i=k)}$$

- So the joint probability/density[1] is:

$$p_{X,Z}((x_i, z_i)_{i=1}^n) = \prod_{i=1}^n \prod_{k=1}^K (\pi_k f(x_i|\mu_k))^{1(z_i=k)}$$

[1] In this course we will treat probabilities and densities equivalently for notational simplicity. In general, the quantity is a density with respect to the product base measure, where the base measure is the counting measure for discrete variables and Lebesgue for continuous variables.

Mixture Models - Posterior Distribution

- Suppose we know the parameters $(\pi_k, \mu_k)_{k=1}^K$.
- Z_i is a random variable and its posterior distribution given data set $\mathbf{X}$ is:

$$Q_{ik} := p(Z_i = k|x_i) = \frac{p(Z_i = k, x_i)}{p(x_i)} = \frac{\pi_k f(x_i|\mu_k)}{\sum_{j=1}^K \pi_j f(x_i|\mu_j)}$$

where the marginal probability of i-th instance is:

$$p(x_i) = \sum_{j=1}^K p(Z_i = j, x_i) = \sum_{j=1}^K \pi_j f(x_i|\mu_j).$$

- The posterior probability Q_{ik} of $Z_i = k$ is called the **responsibility** of mixture component k for data point x_i.
- The posterior distribution **softly partitions** the dataset among the k components.

Mixture Models - Maximum Likehood

- How can we learn about the parameters $\theta = (\pi_k, \mu_k)_{k=1}^K$ from data?
- Standard statistical methodology asks for the **maximum likelihood** estimator (MLE).
- The goal is to maximize the marginal probability of the data over the parameters

$$\begin{aligned}
\hat{\theta}_{\mathsf{ML}} = \operatorname*{argmax}_{\theta} p(\mathbf{X}|\theta) &= \operatorname*{argmax}_{(\pi_k,\mu_k)_{k=1}^K} \prod_{i=1}^n p(x_i|(\pi_k, \mu_k)_{k=1}^K) \\
&= \operatorname*{argmax}_{(\pi_k,\mu_k)_{k=1}^K} \prod_{i=1}^n \sum_{k=1}^K \pi_k f(x_i|\mu_k) \\
&= \operatorname*{argmax}_{(\pi_k,\mu_k)_{k=1}^K} \underbrace{\sum_{i=1}^n \log \sum_{k=1}^K \pi_k f(x_i|\mu_k)}_{:=\ell((\pi_k,\mu_k)_{k=1}^K)}.
\end{aligned}$$

Mixture Models - Maximum Likehood

- Marginal log-likelihood:

$$\ell((\pi_k, \mu_k)_{k=1}^K) := \log p(\mathbf{X}|(\pi_k, \mu_k)_{k=1}^K) = \sum_{i=1}^n \log \sum_{k=1}^K \pi_k f(x_i|\mu_k)$$

- The gradient w.r.t. μ_k:

$$\begin{aligned}\nabla_{\mu_k} \ell((\pi_k, \mu_k)_{k=1}^K) &= \sum_{i=1}^n \frac{\pi_k f(x_i|\mu_k)}{\sum_{j=1}^K \pi_j f(x_i|\mu_j)} \nabla_{\mu_k} \log f(x_i|\mu_k) \\ &= \sum_{i=1}^n Q_{ik} \nabla_{\mu_k} \log f(x_i|\mu_k).\end{aligned}$$

- Difficult to solve, as Q_{ik} depends implicitly on μ_k.

Mixture Models - Maximum Likehood

$$\sum_{i=1}^{n} Q_{ik} \nabla_{\mu_k} \log f(x_i|\mu_k) = 0$$

- What if we ignore the dependence of Q_{ik} on the parameters?
- Taking the mixture of Gaussian with covariance $\sigma^2 I$ as example,

$$\sum_{i=1}^{n} Q_{ik} \nabla_{\mu_k} \left(-\frac{p}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \|x_i - \mu_k\|_2^2 \right)$$

$$= \frac{1}{\sigma^2} \sum_{i=1}^{n} Q_{ik}(x_i - \mu_k) = \frac{1}{\sigma^2} \left(\sum_{i=1}^{n} Q_{ik} x_i - \mu_k \left(\sum_{i=1}^{n} Q_{ik} \right) \right) = 0$$

$$\boxed{\mu_k^{\text{ML?}} = \frac{\sum_{i=1}^{n} Q_{ik} x_i}{\sum_{i=1}^{n} Q_{ik}}}$$

Mixture Models - Maximum Likehood

- The estimate is a weighted average of data points, where the estimated mean of cluster k uses its responsibilities to data points as weights.

$$\mu_k^{\text{ML?}} = \frac{\sum_{i=1}^n Q_{ik} x_i}{\sum_{i=1}^n Q_{ik}}.$$

- Makes sense: Suppose we knew that data point x_i came from population z_i. Then $Q_{iz_i} = 1$ and $Q_{ik} = 0$ for $k \neq z_i$ and:

$$\mu_k^{\text{ML?}} = \frac{\sum_{i:z_i=k} x_i}{\sum_{i:z_i=k} 1} = \text{avg}\{x_i : z_i = k\}$$

- Our best guess of the originating population is given by Q_{ik}.

Mixture Models - Maximum Likehood

- Gradient w.r.t. mixing proportion π_k (including a Lagrange multiplier $\lambda\left(\sum_k \pi_k - 1\right)$ to enforce constraint $\sum_k \pi_k = 1$).

$$\begin{aligned}
&\nabla_{\pi_k}\left(\ell((\pi_k,\mu_k)_{k=1}^K) - \lambda(\textstyle\sum_{k=1}^K \pi_k - 1)\right) \\
&= \sum_{i=1}^n \frac{f(x_i|\mu_k)}{\sum_{j=1}^K \pi_j f(x_i|\mu_j)} - \lambda \\
&= \sum_{i=1}^n \frac{Q_{ik}}{\pi_k} - \lambda = 0 \quad \Rightarrow \quad \pi_k \propto \sum_{i=1}^n Q_{ik}
\end{aligned}$$

Note: $\displaystyle\sum_{k=1}^K \sum_{i=1}^n Q_{ik} = \sum_{i=1}^n \underbrace{\sum_{k=1}^K Q_{ik}}_{=1}$

$$\boxed{\pi_k^{\text{ML?}} = \frac{\sum_{i=1}^n Q_{ik}}{n}}$$

- Again makes sense: the estimate is simply (our best guess of) the proportion of data points coming from population k.

Mixture Models - The EM Algorithm

- Putting all the derivations together, we get an iterative algorithm for learning about the unknowns in the mixture model.
- Start with some initial parameters $(\pi_k^{(0)}, \mu_k^{(0)})_{k=1}^K$.
- Iterate for $t = 1, 2, \ldots$:
  - **Expectation Step**:

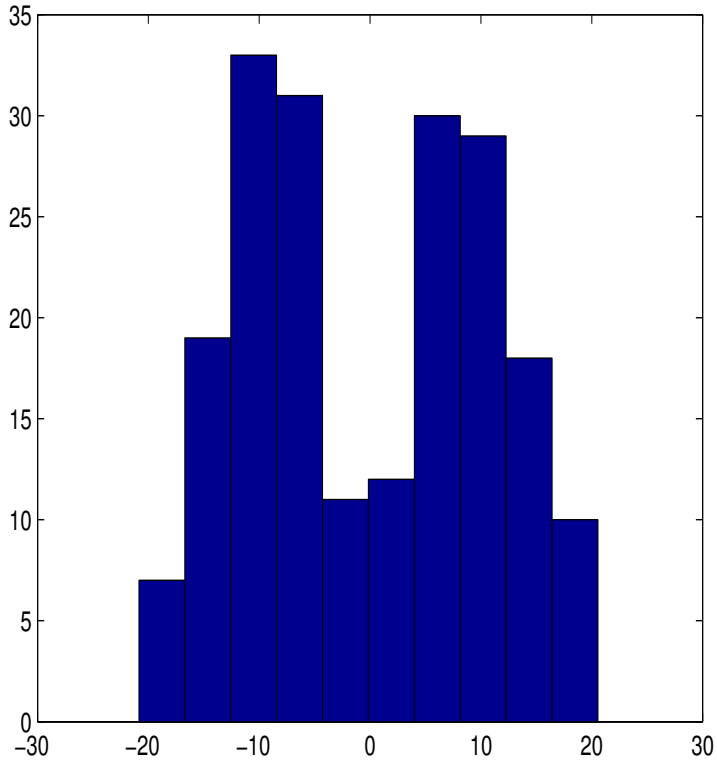
$$Q_{ik}^{(t)} := \frac{\pi_k^{(t-1)} f(x_i|\mu_k^{(t-1)})}{\sum_{j=1}^K \pi_j^{(t-1)} f(x_i|\mu_j^{(t-1)})}$$

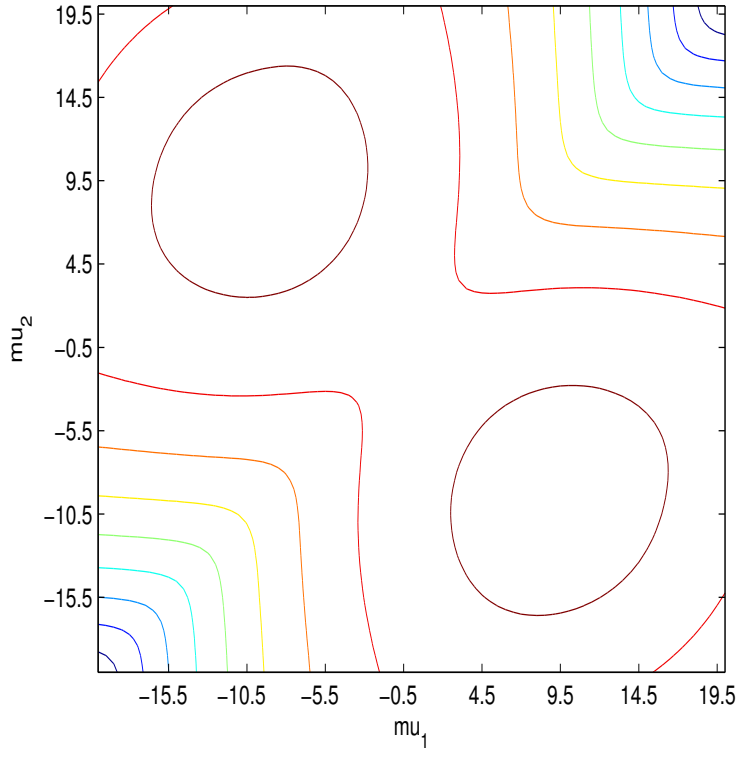
  - **Maximization Step**:

$$\pi_k^{(t)} = \frac{\sum_{i=1}^n Q_{ik}^{(t)}}{n} \qquad \mu_k^{(t)} = \frac{\sum_{i=1}^n Q_{ik}^{(t)} x_i}{\sum_{i=1}^n Q_{ik}^{(t)}}$$

- Will the algorithm converge?
- What does it converge to?

Likelihood Surface for a Simple Example

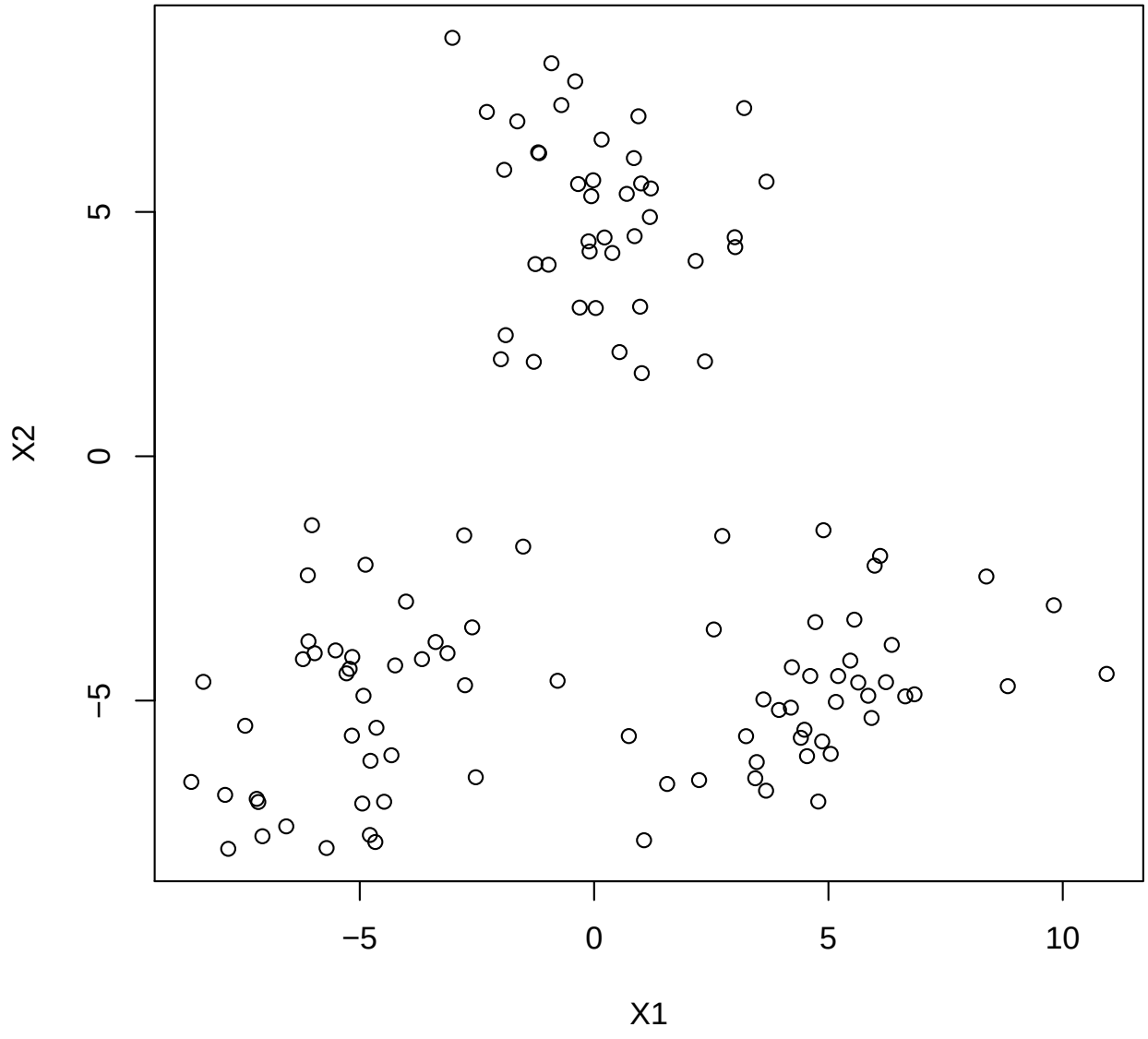




(left) $n = 200$ data points from a mixture of two 1D Gaussians with $\pi_1 = \pi_2 = 0.5$, $\sigma = 5$ and $\mu_1 = 10, \mu_2 = -10$.
(right) Log likelihood surface $\ell(\mu_1, \mu_2)$, all the other parameters being assumed known.

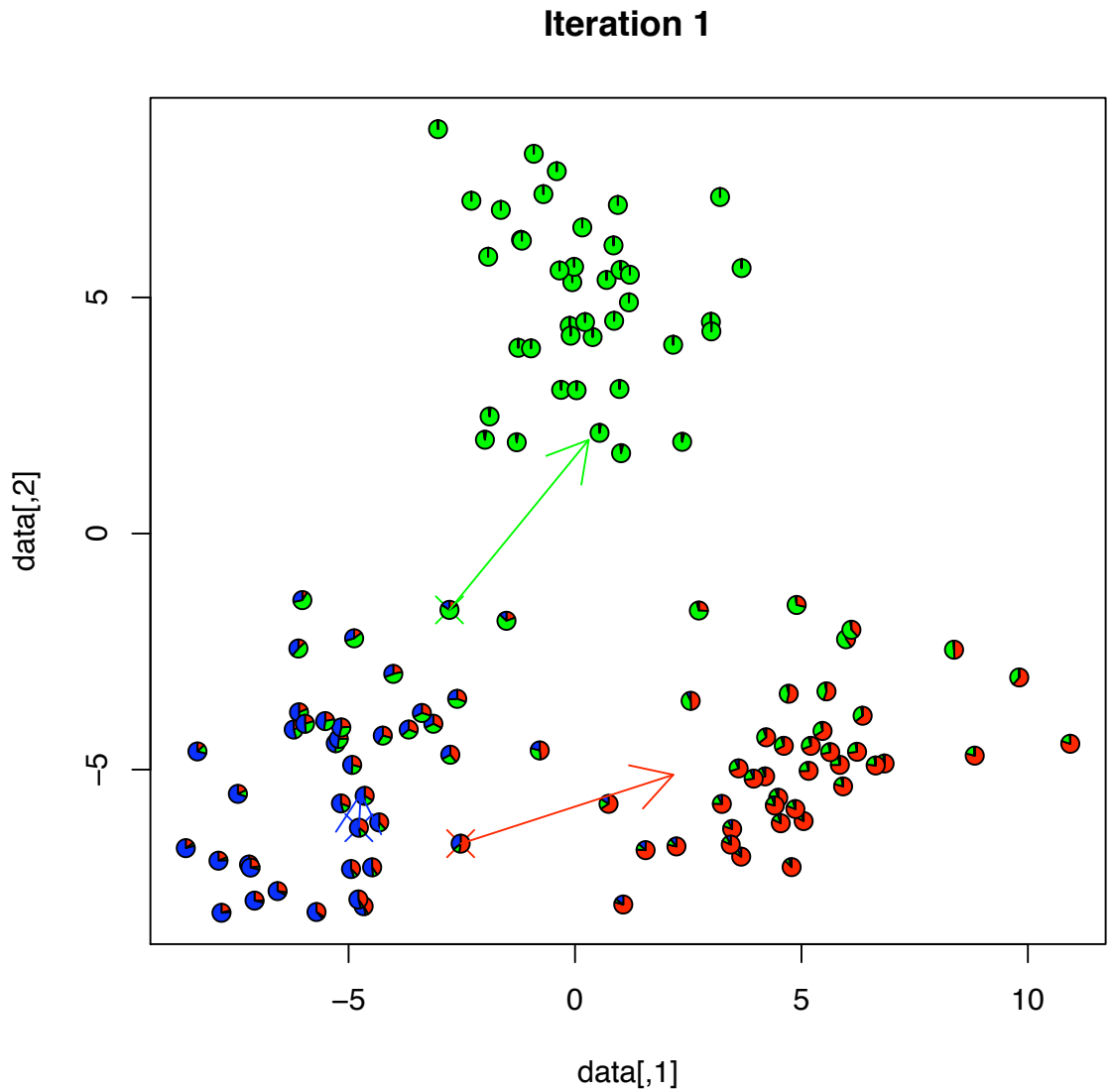
Example: Mixture of 3 Gaussians

An example with 3 clusters.



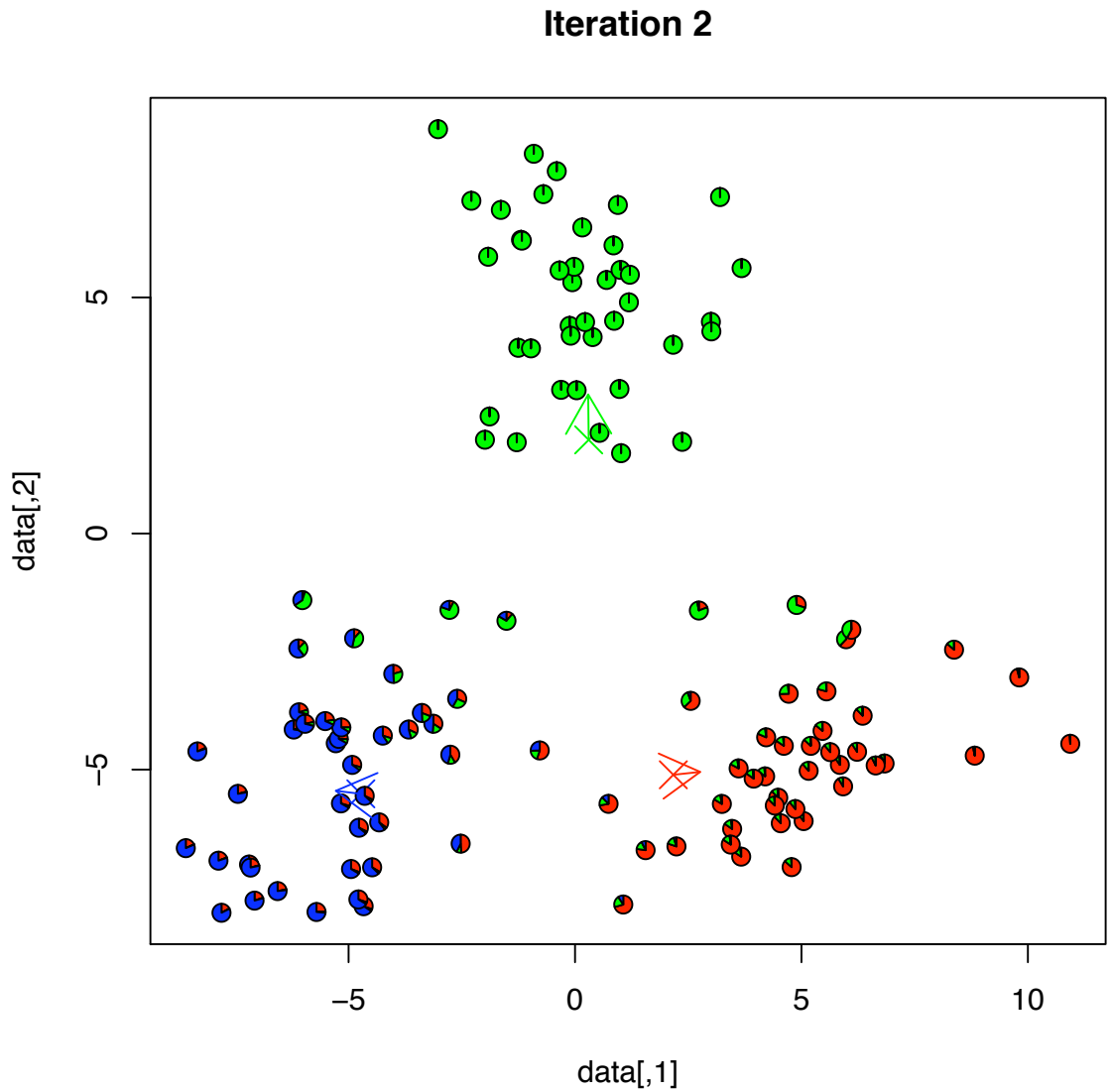
Example: Mixture of 3 Gaussians

After 1st E and M step.



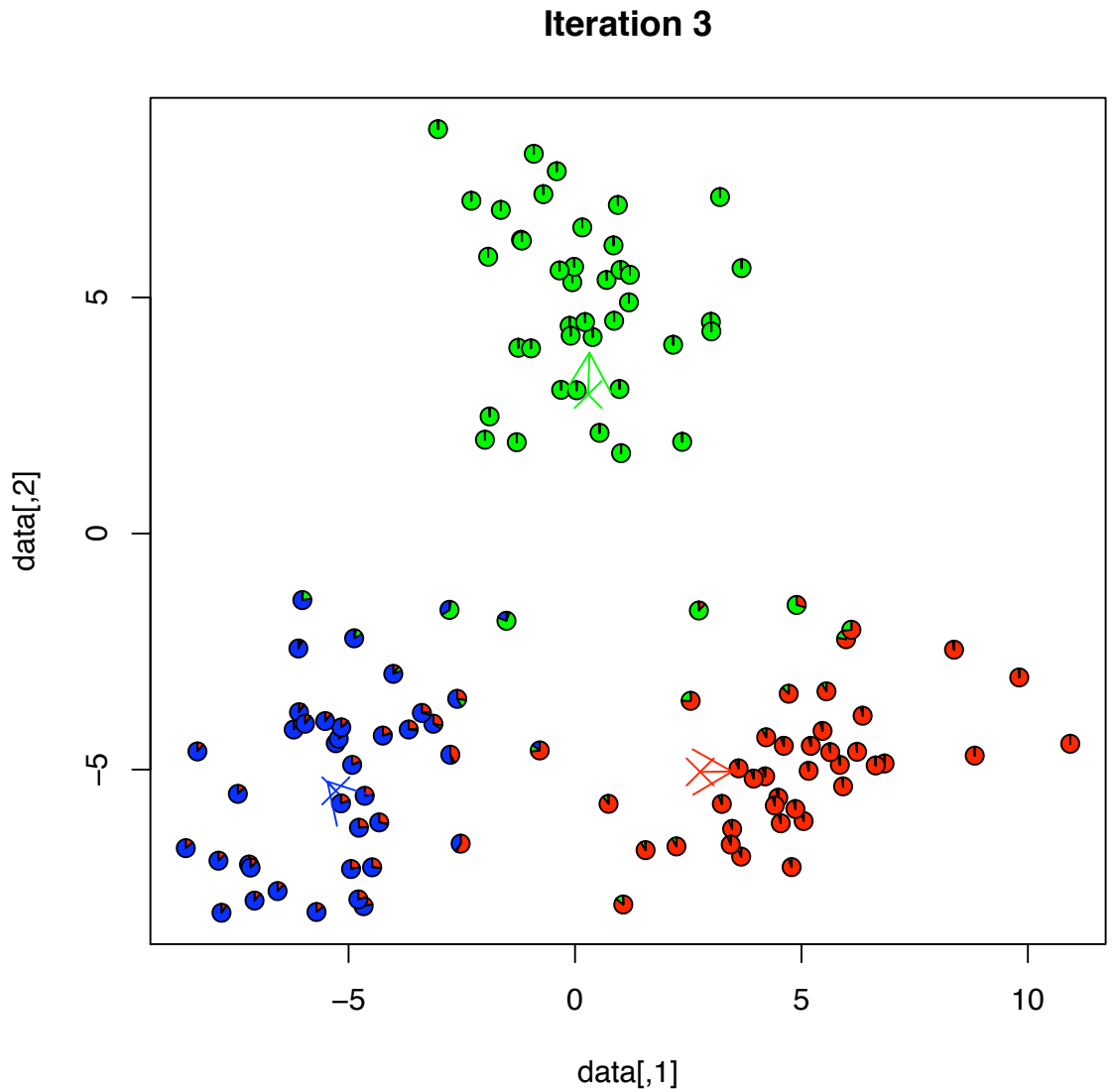
Example: Mixture of 3 Gaussians

After 2nd E and M step.



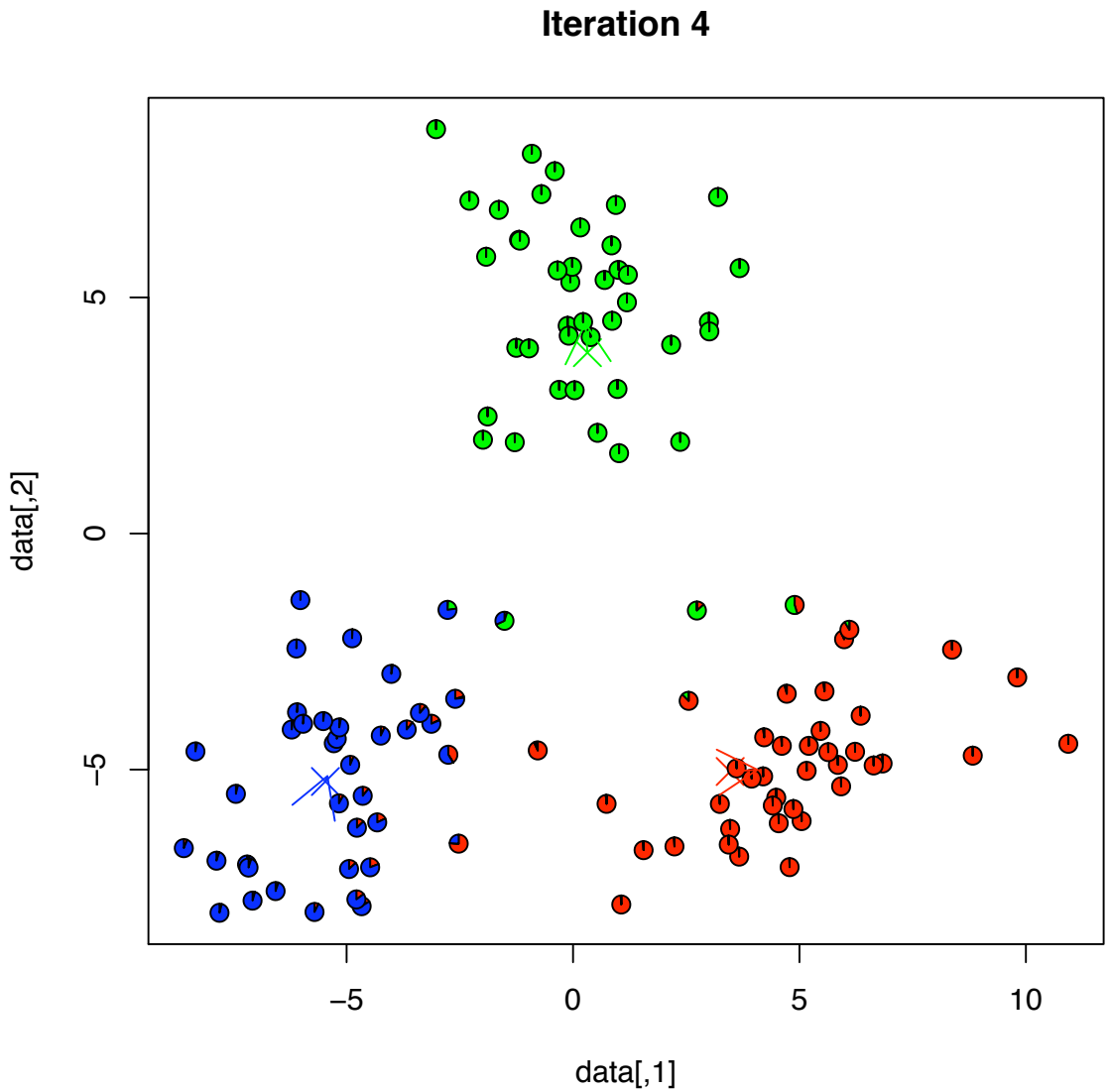
Example: Mixture of 3 Gaussians

After 3rd E and M step.



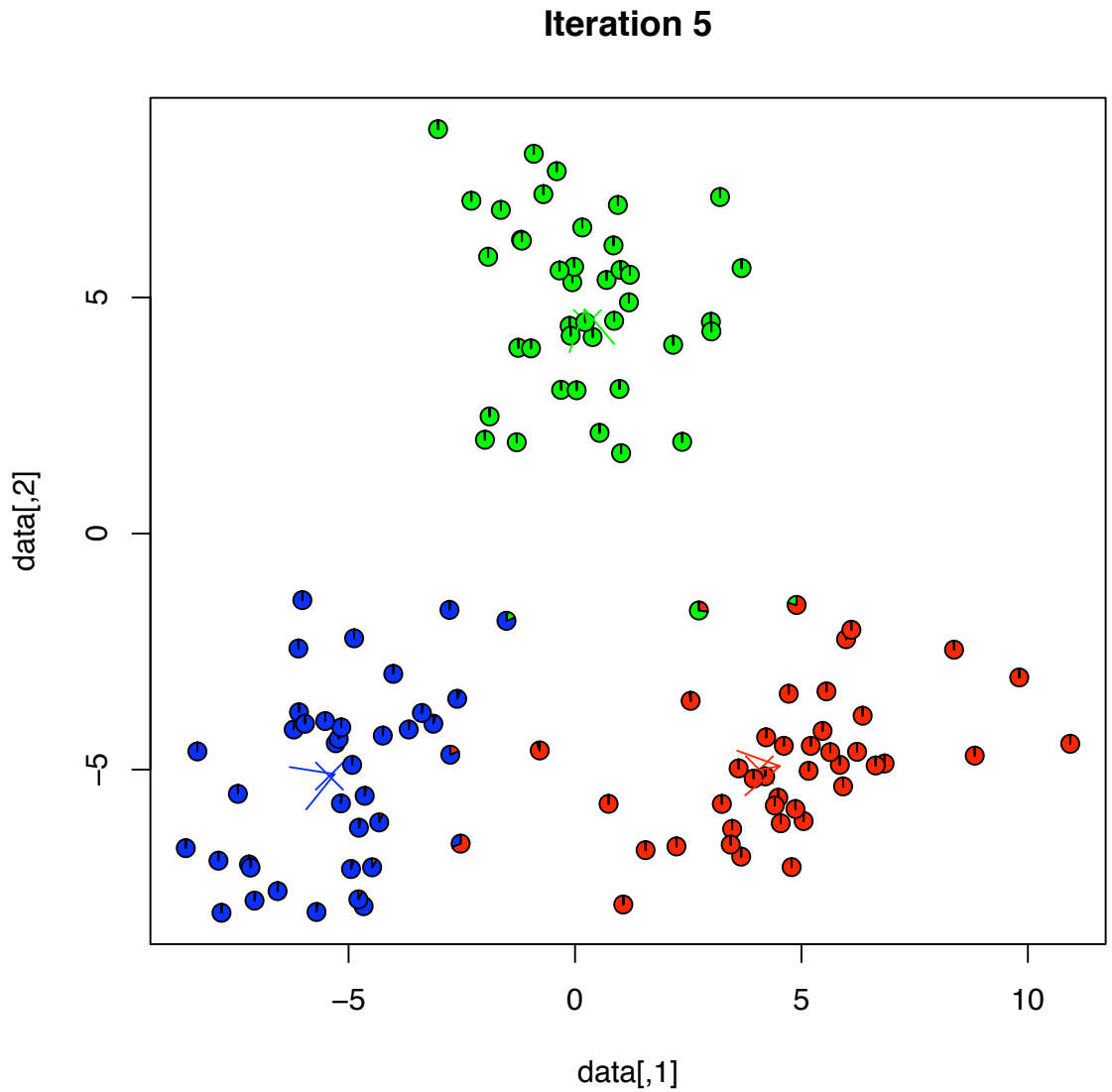
Example: Mixture of 3 Gaussians

After 4th E and M step.



Example: Mixture of 3 Gaussians

After 5th E and M step.



EM Algorithm

- In a maximum likelihood framework, the objective function is the log likelihood,

$$\ell(\theta) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f(x_i|\mu_k)$$

  Direct maximization is not feasible.
- Consider another objective function $\mathcal{F}(\theta, q)$ such that:

$$\mathcal{F}(\theta, q) \leq \ell(\theta) \text{ for all } \theta, q,$$
$$\max_q \mathcal{F}(\theta, q) = \ell(\theta)$$

  $\mathcal{F}(\theta, q)$ is a lower bound on the log likelihood.
- We can construct an alternating maximization algorithm as follows: For $t = 1, 2 \ldots$ until convergence:

$$q^{(t)} := \operatorname*{argmax}_q \mathcal{F}(\theta^{(t-1)}, q)$$
$$\theta^{(t)} := \operatorname*{argmax}_\theta \mathcal{F}(\theta, q^{(t)})$$

EM Algorithm

- The lower bound we use is called the **variational free energy**.
- q is a probability mass function for a distribution over $\mathbf{z} := (z_i)_{i=1}^n$.

$$
\begin{aligned}
\mathcal{F}(\theta, q) =& \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{z}|\theta) - \log q(\mathbf{z})] \\
=& \mathbb{E}_q \left[\left(\sum_{i=1}^{n} \sum_{k=1}^{K} 1(z_i = k) \left(\log \pi_k + \log f(x_i|\mu_k) \right) \right) - \log q(\mathbf{z}) \right] \\
=& \sum_{\mathbf{z}} q(\mathbf{z}) \left[\left(\sum_{i=1}^{n} \sum_{k=1}^{K} 1(z_i = k) \left(\log \pi_k + \log f(x_i|\mu_k) \right) \right) - \log q(\mathbf{z}) \right]
\end{aligned}
$$

EM Algorithm - Solving for q

- Gradient of $\mathcal{F}$ w.r.t q (with Lagrange multiplier for $\sum_{\mathbf{z}} q(\mathbf{z}) = 1$):

$$\nabla_{q(\mathbf{z})} \mathcal{F}(\theta, q) = \sum_{i=1}^{n} \sum_{k=1}^{K} 1(z_i = k) \left(\log \pi_k + \log f(x_i|\mu_k)\right) - \log q(\mathbf{z}) - 1 - \lambda$$

$$= \sum_{i=1}^{n} \left(\log \pi_{z_i} + \log f(x_i|\mu_{z_i})\right) - \log q(\mathbf{z}) - 1 - \lambda = 0$$

$$\Rightarrow q^*(\mathbf{z}) \propto \prod_{i=1}^{n} \pi_{z_i} f(x_i|\mu_{z_i}).$$

$$q^*(\mathbf{z}) = \frac{\prod_{i=1}^{n} \pi_{z_i} f(x_i|\mu_{z_i})}{\sum_{\mathbf{z}'} \prod_{i=1}^{n} \pi_{z_i'} f(x_i|\mu_{z_i'})} = \prod_{i=1}^{n} \frac{\pi_{z_i} f(x_i|\mu_{z_i})}{\sum_k \pi_k f(x_i|\mu_k)} = \prod_{i=1}^{n} p(z_i|x_i, \theta).$$

- Optimal q^* is simply the posterior distribution for fixed θ.
- Plugging in the optimal q^* into the variational free energy,

$$\mathcal{F}(\theta, q^*) = \sum_{i=1}^{n} \log \sum_{k=1}^{K} \pi_k f(x_i|\mu_k) = \ell(\theta)$$

EM Algorithm - Solving for θ

- Setting derivative with respect to μ_k to 0,

$$\nabla_{\mu_k}\mathcal{F}(\theta,q) = \sum_{\mathbf{z}} q(\mathbf{z}) \sum_{i=1}^{n} 1(z_i = k)\nabla_{\mu_k} \log f(x_i|\mu_k)$$
$$= \sum_{i=1}^{n} q(z_i = k)\nabla_{\mu_k} \log f(x_i|\mu_k) = 0$$

- This equation can be solved quite easily. E.g., for mixture of Gaussians,

$$\mu_k^* = \frac{\sum_{i=1}^{n} q(z_i = k)x_i}{\sum_{i=1}^{n} q(z_i = k)}$$

- If it cannot be solved exactly, we can use **gradient ascent** algorithm:

$$\mu_k^* = \mu_k + \alpha \sum_{i=1}^{n} q(z_i = k)\nabla_{\mu_k} \log f(x_i|\mu_k).$$

- Similar derivation for optimal π_k as before.

EM Algorithm

- Start with some initial parameters $(\pi_k^{(0)}, \mu_k^{(0)})_{k=1}^K$.
- Iterate for $t = 1, 2, \ldots$:
  - **Expectation Step**:

$$q^{(t)}(z_i = k) := \frac{\pi_k^{(t-1)} f(x_i|\mu_k^{(t-1)})}{\sum_{j=1}^K \pi_j^{(t-1)} f(x_i|\mu_j^{(t-1)})} = \mathbb{E}_{p(z_i|x_i,\theta^{(t-1)})}[1(z_i = k)]$$

  - **Maximization Step**:

$$\pi_k^{(t)} = \frac{\sum_{i=1}^n q^{(t)}(z_i = k)}{n} \qquad \mu_k^{(t)} = \frac{\sum_{i=1}^n q^{(t)}(z_i = k) x_i}{\sum_{i=1}^n q^{(t)}(z_i = k)}$$

- Each step increases the log likelihood:

$$\ell(\theta^{(t-1)}) = \mathcal{F}(\theta^{(t-1)}, q^{(t)}) \leq \mathcal{F}(\theta^{(t)}, q^{(t)}) \leq \mathcal{F}(\theta^{(t)}, q^{(t+1)}) = \ell(\theta^{(t)}).$$

- Additional assumption, that $\nabla_\theta^2 \mathcal{F}(\theta^{(t)}, q^{(t)})$ are negative definite with eigenvalues $< -\epsilon < 0$, implies that $\theta^{(t)} \to \theta^*$ where θ^* is a local MLE.

Notes on Probabilistic Approach and EM Algorithm

Some good things:

- Guaranteed convergence to locally optimal parameters.
- Formal reasoning of uncertainties, using both Bayes Theorem and maximum likelihood theory.
- Rich language of probability theory to express a wide range of generative models, and straightforward derivation of algorithms for ML estimation.

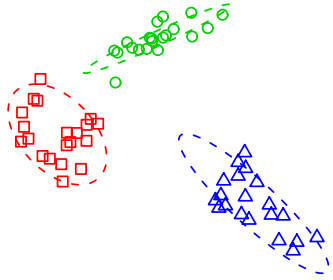
Some bad things:

- Can get stuck in local minima so multiple starts are recommended.
- Slower and more expensive than K-means.
- Choice of K still problematic, but rich array of methods for model selection comes to rescue.

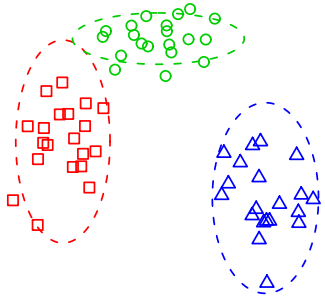
Flexible Gaussian Mixture Models

- We can allow each cluster to have its own mean and covariance structure allows greater flexibility in the model.

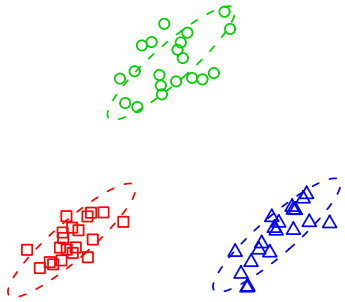
Different covariances



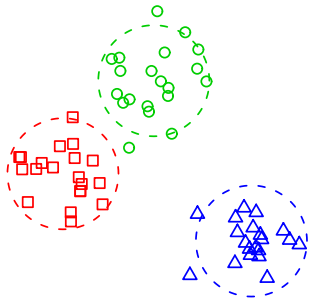
Different, but diagonal covariances



Identical covariances



Identical and spherical covariances



Probabilistic PCA

- A probabilistic model related to PCA has the following generative model: for $i = 1, 2, \ldots, n$:
  - Let $k < n, p$ be given.
  - Let Y_i be a (latent) k-dimensional normally distributed random variable with 0 mean and identity covariance:

$$Y_i \sim \mathcal{N}(0, I_k)$$

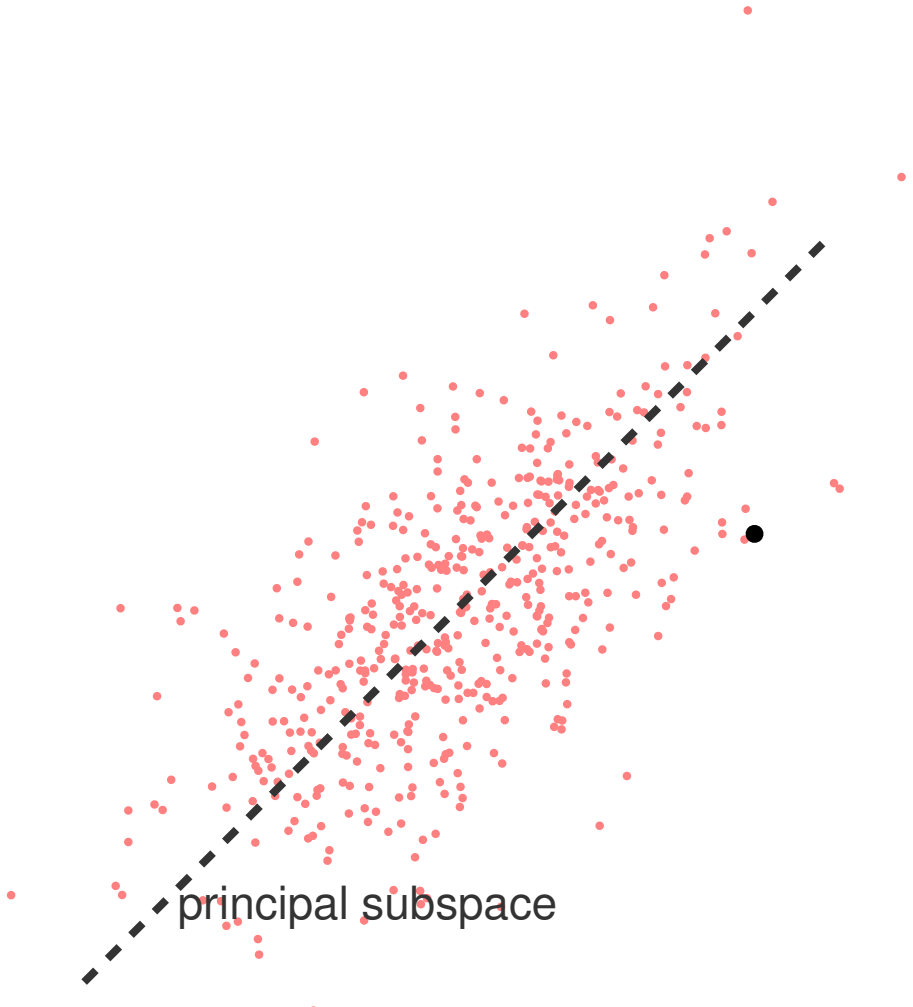
  - We model the distribution of the ith data point given Y_i as a p-dimensional normal:

$$X_i \sim \mathcal{N}(\mu + LY_i, \sigma^2 I)$$

    where the parameters are a vector $\mu \in \mathbb{R}^p$, a matrix $L \in \mathbb{R}^{p \times k}$ and $\sigma^2 > 0$.

Probabilistic PCA

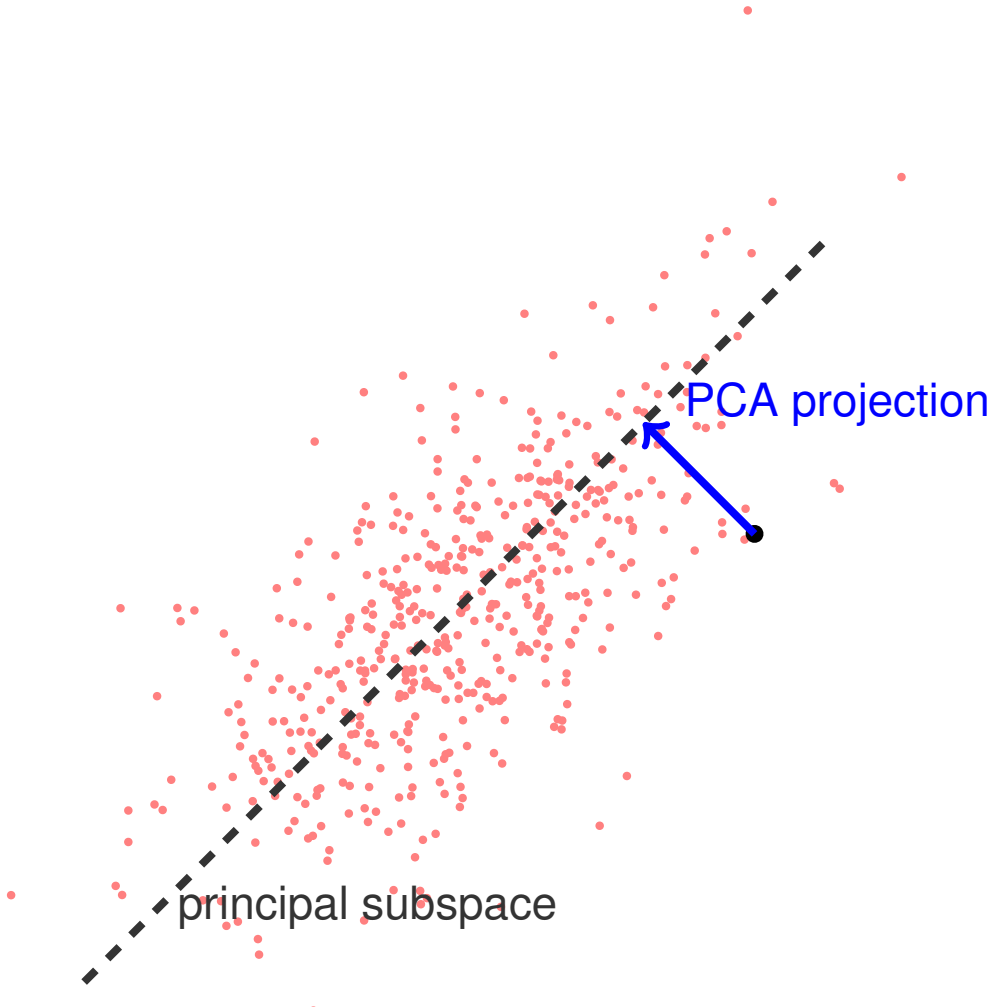
PPCA latents



figures by M. Sahani

Probabilistic PCA

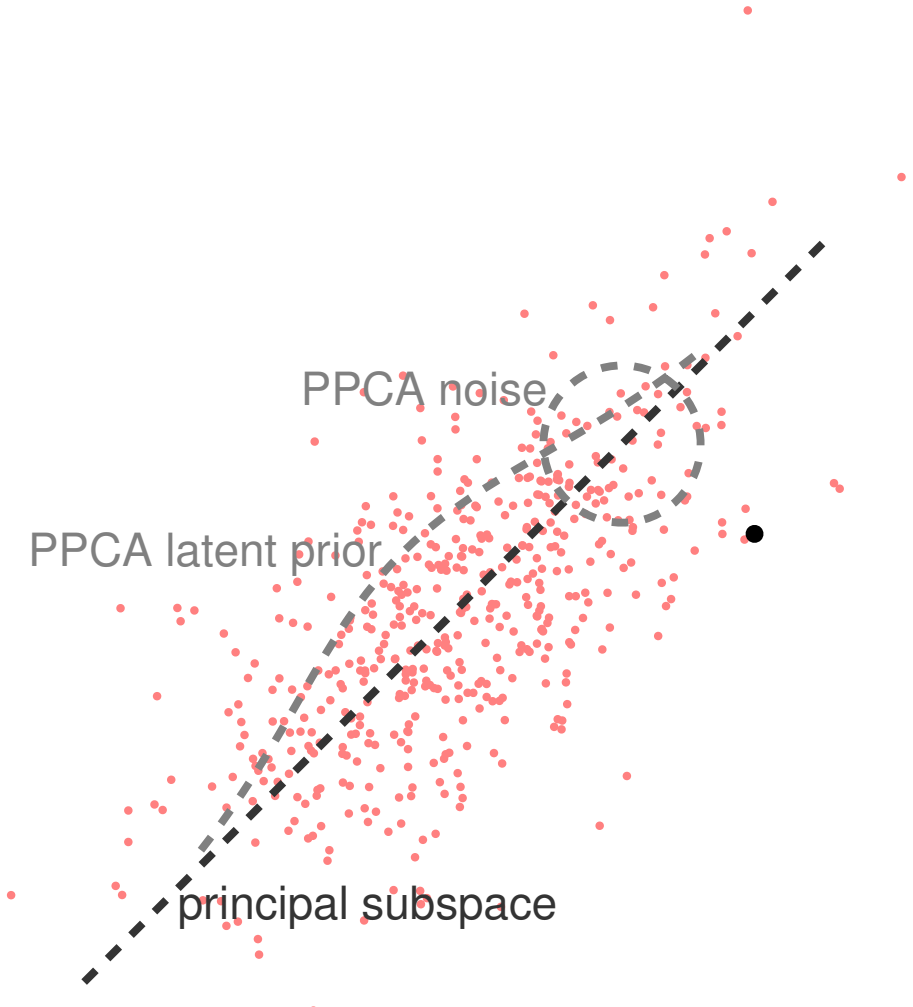
PPCA latents



figures by M. Sahani

Probabilistic PCA

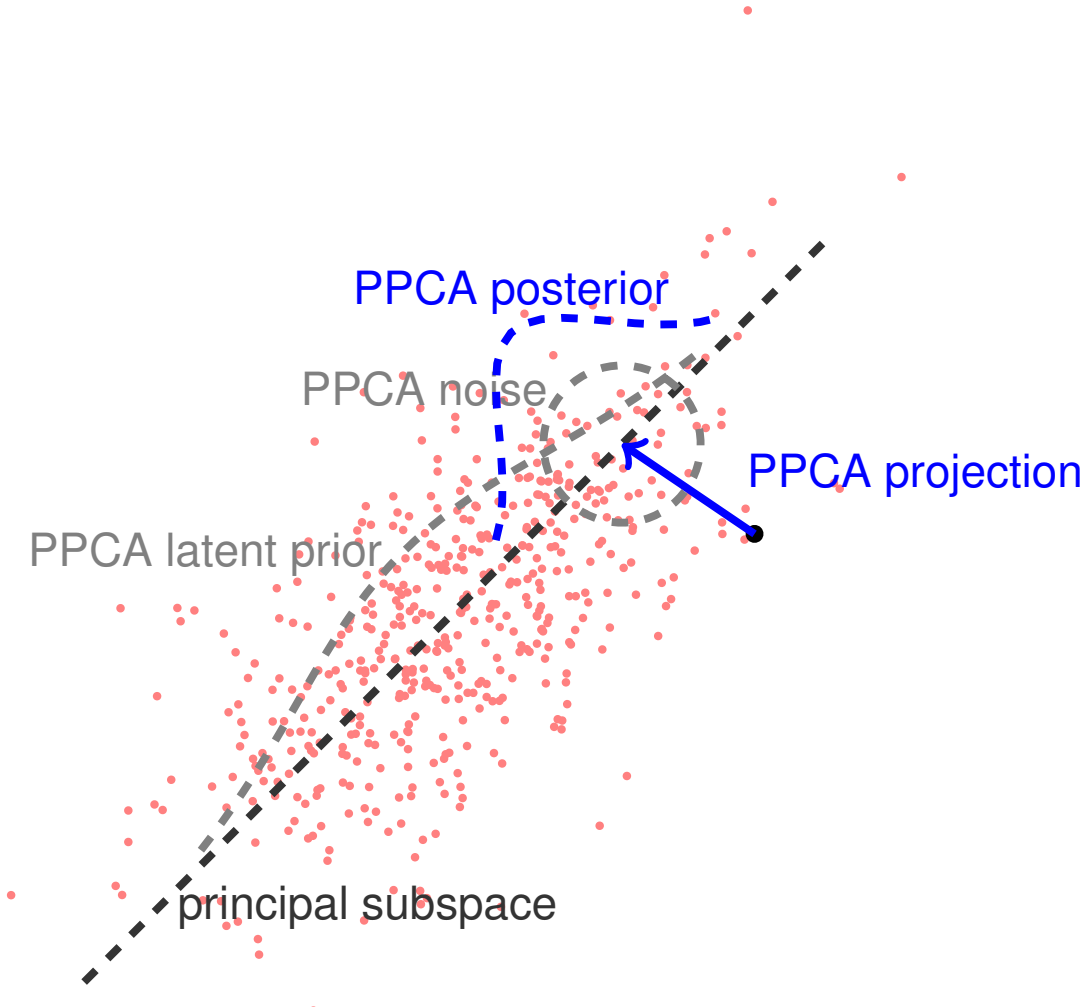
PPCA latents



figures by M. Sahani

Probabilistic PCA

PPCA latents



figures by M. Sahani

Mixture of Probabilistic PCAs

- We have learnt two types of unsupervised learning techniques:
  - Dimensionality reduction, e.g. PCA, MDS, Isomap.
  - Clustering, e.g. K-means, linkage and mixture models.
- Probabilistic models allow us to construct more complex models from simpler pieces.
- Mixture of probabilistic PCAs allows both clustering and dimensionality reduction at the same time.

$$
\begin{aligned}
Z_i &\sim \text{Discrete}(\pi_1, \ldots, \pi_K) \\
Y_i &\sim \mathcal{N}(0, I_d) \\
X_i | Z_i = k, Y_i = y_i &\sim \mathcal{N}(\mu_k + L y_i, \sigma^2 I_p)
\end{aligned}
$$

- Allows flexible modelling of covariance structure without using too many parameters.

Ghahramani and Hinton 1996

Further Reading—Unsupervised Learning

- Hastie et al, Chapter 14.
- James et al, Chapter 10.
- Ripley, Chapter 9.
- Tukey, John W. (1980). We need both exploratory and confirmatory. The American Statistician 34 (1): 23-25.