SC4/SM8 Advanced Topics in Statistical Machine Learning

Chapter 5: Latent Variable Models and EM Algorithm

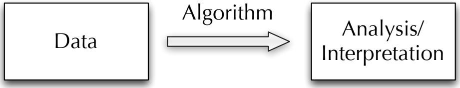
Yee Whye Teh
Department of Statistics
Oxford

https://github.com/ywteh/advml2020

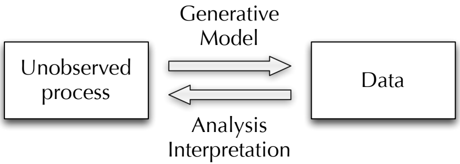
Probabilistic Unsupervised Learning

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**).
- Samples in population k can be modelled using a distribution F_{μ_k} with density $f(x|\mu_k)$, where μ_k is the **model parameter** for the k-th component. 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 the assignment variable independently for each data item i:

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

    where **mixing proportions** are $\pi_k \geq 0$ for each k and $\sum_{k=1}^K \pi_k = 1$.
  - Given the assignment $Z_i = k$, then $X_i = (X_i^{(1)}, \ldots, X_i^{(p)})^\top$ is sampled (independently) from the corresponding k-th component:

    $$X_i | Z_i = k \sim f(x|\mu_k)$$

- We observe $X_i = x_i$ for each i but not Z_i's (**latent variables**), and would like to infer the parameters $\{\mu_k\}_{k=1}^K$ and $\{\pi_k\}_{k=1}^K$ (σ^2 can also be estimated).

Mixture Models

- Unknowns to learn given data are
  - **Parameters**: $\theta = (\pi_k, \mu_k)_{k=1}^K$, where $\pi_1, \ldots, \pi_K \in [0,1]$, $\mu_1, \ldots, \mu_K \in \mathbb{R}^p$, **and**
  - **Latent variables**: $z_1, \ldots, z_n$.
- The joint probability over all cluster indicator variables $\{Z_i\}$ are:

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

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

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

Mixture Models: Joint pmf/pdf of observed and latent variables

- Unknowns to learn given data are
  - **Parameters**: $\theta = (\pi_k, \mu_k)_{k=1}^K$, where $\pi_1, \dots, \pi_K \in [0, 1]$, $\mu_1, \dots, \mu_K \in \mathbb{R}^p$, **and**
  - **Latent variables**: $z_1, \dots, z_n$.
- The joint probability mass function/density[1] is:

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

- And the marginal density of x_i (resulting model on the observed data) is:

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

Mixture Models: Gaussian Mixtures with Unequal Covariances

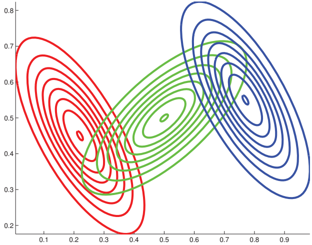


figure from Murphy, 2012, Ch. 11.

Here $\theta = (\pi_k, \mu_k, \Sigma_k)_{k=1}^K$ are all the model parametes and

$$
\begin{aligned}
f\left(x \middle| \left(\mu_k, \Sigma_k\right)\right) &= (2\pi)^{-\frac{p}{2}} |\Sigma_k|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\right), \\
p(x) &= \sum_{k=1}^{K} \pi_k f\left(x \middle| \left(\mu_k, \Sigma_k\right)\right)
\end{aligned}
$$

Mixture Models: Responsibility

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

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

- The conditional probability Q_{ik} is called the **responsibility** of mixture component k for data point x_i.
- These conditionals **softly partitions** the dataset among the k components: $\sum_{k=1}^K Q_{ik} = 1$.

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 maximise the marginal probability of the data over the parameters

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

Mixture Models: Maximum Likehood

- Marginal log-likelihood:

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

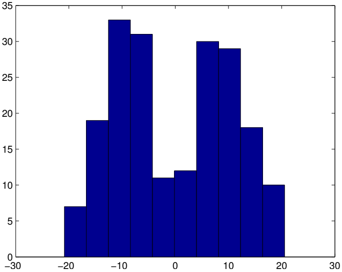
- The gradient w.r.t. μ_k:

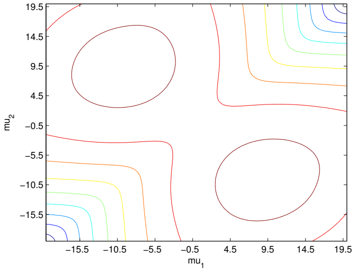
$$\nabla_{\mu_k} \ell((\pi_k, \mu_k)_{k=1}^K) = \sum_{i=1}^n \frac{\pi_k f(x_i|\mu_k)}{\sum_{j=1}^K \pi_j f(x_i|\mu_j)} \nabla_{\mu_k} \log f(x_i|\mu_k)$$
$$= \sum_{i=1}^n Q_{ik} \nabla_{\mu_k} \log f(x_i|\mu_k).$$

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

Likelihood Surface for a Simple Example

If latent variables z_i's were all observed, we would have a unimodal likelihood surface but when we marginalise out the latents, the likelihood surface becomes multimodal: no unique MLE.





(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) Observed data log likelihood surface $\ell(\mu_1, \mu_2)$, all the other parameters being assumed known.

Mixture Models: Maximum Likehood

Recall we would like to solve:

$$\nabla_{\mu_k} \ell((\pi_k, \mu_k)_{k=1}^K) = \sum_{i=1}^n Q_{ik} \nabla_{\mu_k} \log f(x_i | \mu_k) = 0$$

- 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}.
- Soft K-Means algorithm?

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: $\sum_{k=1}^K \sum_{i=1}^n Q_{ik} = \sum_{i=1}^n \underbrace{\sum_{k=1}^K Q_{ik}}_{=1}$ $\quad \boxed{\pi_k^{\text{ML?}} = \frac{\sum_{i=1}^n Q_{ik}}{n}}$

- 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**:

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

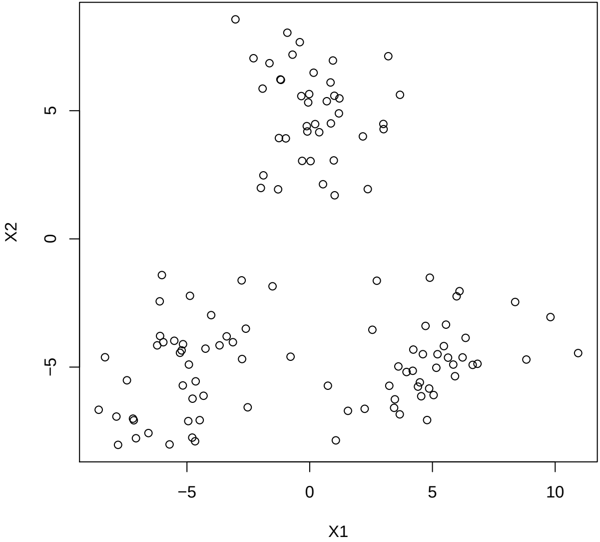
  - **Maximization Step**:

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

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

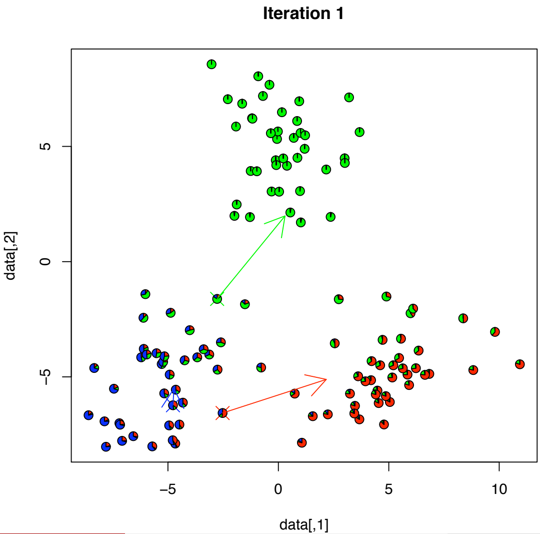
Example: Mixture of 3 Gaussians

An example with 3 clusters.



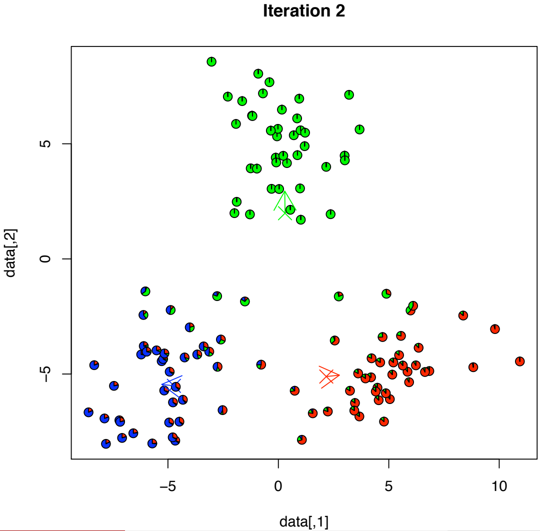
Example: Mixture of 3 Gaussians

After 1st E and M step.



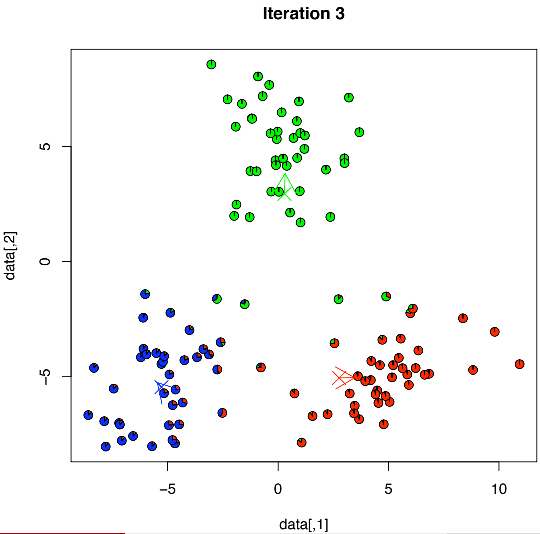
Example: Mixture of 3 Gaussians

After 2nd E and M step.



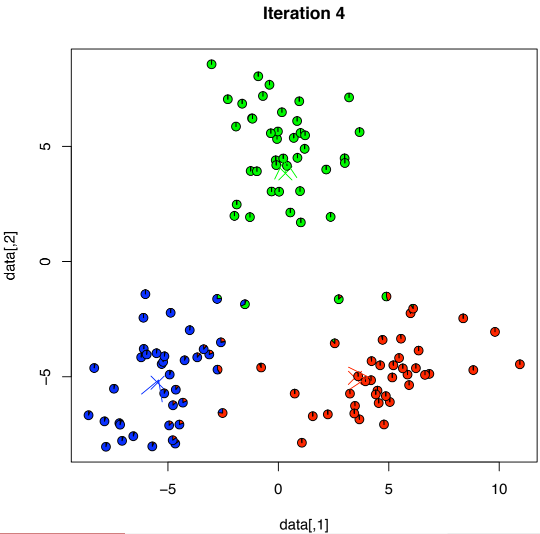
Example: Mixture of 3 Gaussians

After 3rd E and M step.



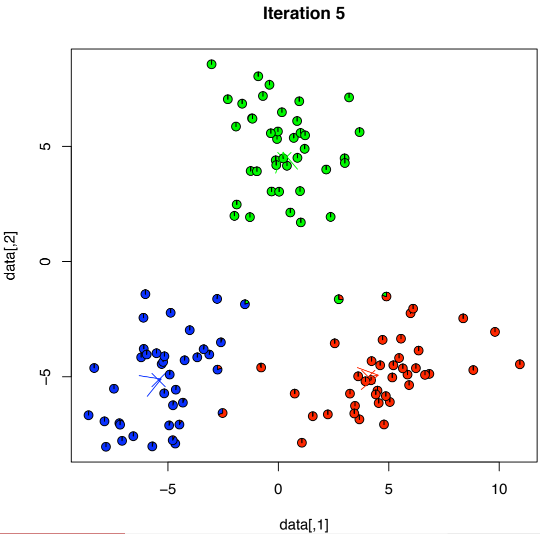
Example: Mixture of 3 Gaussians

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 maximisation is not feasible.

- Consider another objective function $\mathcal{F}(\theta, q)$, where q is any probability distribution on latent variables z, such that:

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

  $\mathcal{F}(\theta, q)$ is a **lower bound on the log likelihood**.

- We can construct an alternating maximisation 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 $\mathbf{z} := (z_i)_{i=1}^n$.

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

Lemma

$\mathcal{F}(\theta, q) \leq \ell(\theta)$ for all q and for all θ.

EM Algorithm - Solving for q

Lemma

$\mathcal{F}(\theta, q) = \ell(\theta)$ *for* $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \theta)$.

In combination with previous Lemma, this implies that $q(\mathbf{z}) = p(\mathbf{z}|\mathbf{x}, \theta)$ maximizes $\mathcal{F}(\theta, q)$ for fixed θ, i.e., the optimal q^* is simply the conditional distribution given the data and that fixed θ.

- In mixture model,

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

$$= \prod_{i=1}^n p(z_i|x_i, \theta).$$

EM Algorithm - Solving for θ

- Setting derivative with respect to μ_k to 0,

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

- 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 (**generalized EM**):

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

- 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**:

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

  - **Maximization Step**:

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

Theorem

EM algorithm does not decrease the log likelihood.

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

- 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:

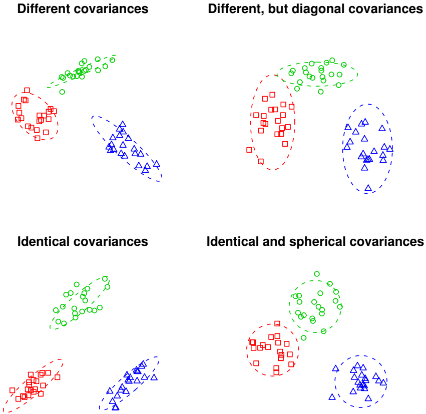
- Guaranteed convergence to locally optimal parameters.
- Formal reasoning of uncertainties, using both Bayes Theorem and maximum likelihood theory.
- Rich language of probability theory to express a wide range of generative models, and straightforward derivation of algorithms for ML estimation.

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 to enable greater flexibility in the model.



Probabilistic PCA

- A probabilistic model related to PCA (also known as sensible 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 $\sigma^2 > 0$.

Tipping and Bishop, 1999

Probabilistic PCA: EM vs MLE

- EM algorithm can be used for ML estimation (lecture notes), but PPCA can more directly give an MLE (which is not unique).
- Let $\lambda_1 \geq \cdots \geq \lambda_p$ be the eigenvalues of the sample covariance and $V_{1:k} \in \mathbb{R}^{p\times k}$ the top k eigenvectors as before. Let $Q \in \mathbb{R}^{k\times k}$ be any orthogonal matrix. Then an MLE is given by:

$$\mu^{\text{MLE}} = \bar{x} \qquad (\sigma^2)^{\text{MLE}} = \frac{1}{p-k}\sum_{j=k+1}^{p} \lambda_j$$

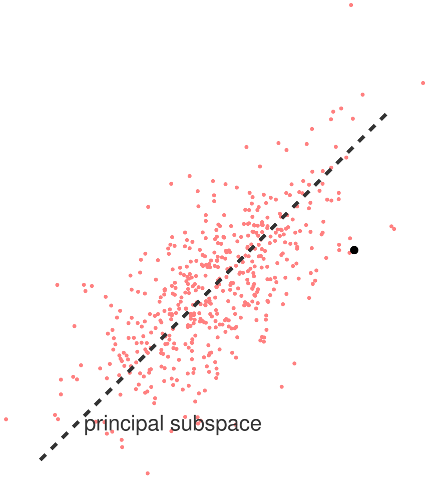
$$L^{\text{MLE}} = V_{1:k}\,\mathrm{diag}((\lambda_1 - (\sigma^2)^{\text{MLE}})^{\frac{1}{2}}, \ldots, (\lambda_k - (\sigma^2)^{\text{MLE}})^{\frac{1}{2}})Q$$

- However, EM can be faster, can be implemented online, can handle missing data and can be extended to more complicated models!

Tipping and Bishop, 1999

Probabilistic PCA

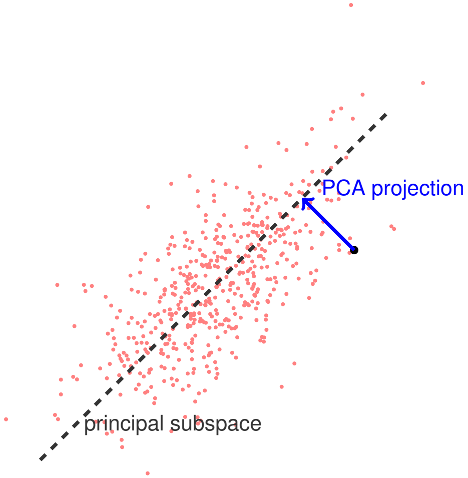
PPCA latents



figures from M. Sahani's UCL course on Unsupervised Learning

Probabilistic PCA

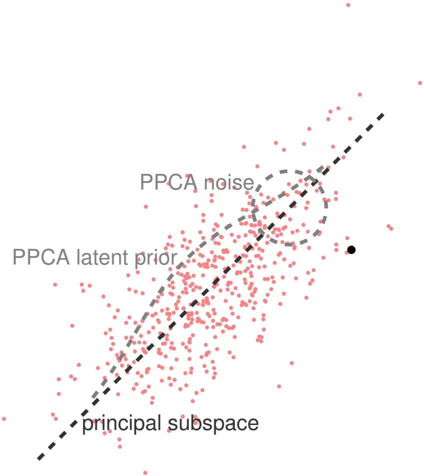
PPCA latents



figures from M. Sahani's UCL course on Unsupervised Learning

Probabilistic PCA

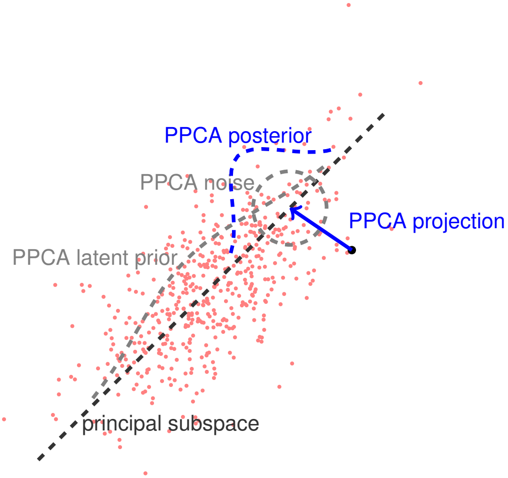
PPCA latents



figures from M. Sahani's UCL course on Unsupervised Learning

Probabilistic PCA

PPCA latents



figures from M. Sahani's UCL course on Unsupervised Learning

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.

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

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

Ghahramani and Hinton 1996

Further reading

- Hastie et al, 8.5
- Bishop, Chapter 9
- Roweis and Ghahramani: A unifying review of linear Gaussian models