Lecture 3(Clustering)

TLDR : we mostly talk about GMM from the perspective of hidden variables and derive the EM steps. Next important thing to understand is Mean Shift algorithm, followed by model selection (selection of hyperparameters)

Clustering

Assign one common label to similar samples in unsupervised fashion. There are 3 specific algorithms: K-Means, GMM, Mean Shift

K-Means

K-means minimizes Euclidean **within-cluster distance**:

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N} \underbrace{\sum_{k=1}^{K} r_k^{(n)} \left\| \mathbf{m}_k - \mathbf{x}^{(n)} \right\|^2}_{\substack{\text{distance between } \mathbf{x}^{(n)} \\ \text{and its assigned cluster centre}}}$$

where m_k is the cluster center and $x^{(n)}$ are data points.
Steps:

1. Initialize K cluster centers randomly
2. Calculate distances to each cluster center from each point and choose the minimum distanced cluster center as cluster assignment
3. Calculate mean of the newly assigned samples in each cluster and update the cluster center
4. Go back to 2 until convergence.
   K-means is locally optimal method. K-Means always reduces the cost at each iteration and whenever an assignment doesn't change, we have converged to a local minimum.
   Model selection is done via Gap statistics (What the fuck is that?)

GMM

The new perspective here is that we should look at GMM as a parametric density estimation method (as opposed to usual clustering method). GMM with gaussians is a universal density estimator.
Each data point is expressed as a weighted sum of Gaussians:

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k)$$

where $0 < \pi_k < 1$ and $\sum_{k=1}^{K} \pi_k = 1$
We cannot directly fit MLE into GMMs because:

1. Some Gaussian components might collapse to single data points, leading to extremely high likelihood values for those points and very low variance estimates.
2. Non-convexity
   That's why we use EM. Hidden variables can be introduced :

   $$p(x, z) = p(z)p(x|z)$$

   We can find $p(x)$ by marginalizing over z:

   $$p(x) = \sum_z p(x, z)$$

   Replacing the value of joint distribution we get:

   $$p(x) = \sum_z p(x, z) = \sum_z p(z)p(x|z)$$

   Compare this to the above (weighted sum of Gaussians) and you immediately can see the correspondence between values $p(z)$ and π_k and $p(x|z)$ and $N(x|\mu_k, \sigma_k)$

z here is a one hot vector, only one of the values is 1, others are 0, thus always summing to 1.

MLE fitting to GMM

Recall that the maximum likelihood for a dataset $D = [x^{(n)}]$ for n data samples has the following form:

$$argmax \log p(D|\mu_k, \sigma_k, \pi_k) = argmax \sum_{i=1}^{N} log(\sum_{k=1}^{K} \pi_k N(x^{(i)}|\mu_k, \sigma_k)$$

Responsibilities:

$$r_{nk} = p(z_k = 1|x_n) = \frac{\pi_k N(x_n|\mu_k, \sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \sigma_j)}$$

What is the probability that a data point is generated by k-th component of the mixture. Or how much do we think a cluster is responsible for generating a datapoint.
$r_n = [r_{n1}, r_{n2}, \ldots, r_{nK}]$ is a probability vector. Total responsibility of the k-th mixture component for the entire dataset is defined as :

$$N_k = \sum_{n=1}^{N} r_{nk}$$

If you have one data point and responsibilities $[0.2, 0.8]$, then there are 20 % chance it comes from the first Gaussian and 80 % from the second Gaussian.

Steps :

1. Initialize μ_k, σ_k, π_k to random values. We have to be careful in initialization, otherwise, we might get stuck over outliers, and converge to a bad local minima.
2. **E-step**: using the current values of μ_k, σ_k, π_k, evaluate responsibilities r_{nk} (posterior distribution) for each component:

$$r_{nk} = \frac{\pi_k N(x_n|\mu_k, \sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \sigma_j)}$$

3. Using responsibilities, evaluate μ_k, π_k, σ_k.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n$$

$$\sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)^2$$

$$\pi_k = \frac{N_k}{N}$$

   EM algorithm:
4. E-step: compute posterior probability over z given our current model
5. M-step: maximize the probability that it would generate the data it is currently responsible for.

GMM is only locally optimal unless operating on Gaussian distributions.

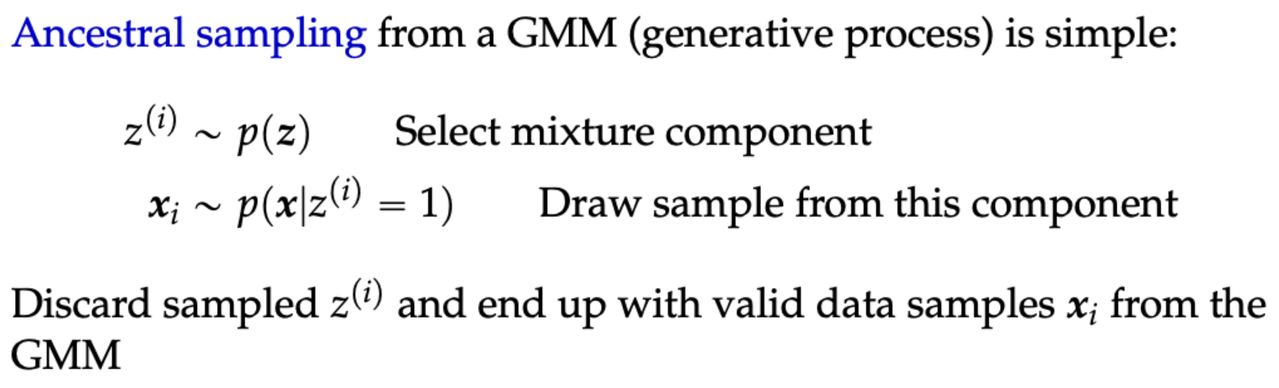
Model selection

We discussed Variational Inference and MCMC for deciding number of clusters for GMM.

Ancestral Sampling

How to draw a sample from a GMM:

1. Randomly choose a component k with prior probability π_k
2. Make a random draw from that k-the Gaussian.
   First sample from the prior, and then from the likelihood given the prior. Sampling from a density forest also works identically



Ancestral sampling from a GMM (generative process) is simple:

$$z^{(i)} \sim p(z) \qquad \text{Select mixture component}$$

$$\boldsymbol{x}_i \sim p(\boldsymbol{x}|z^{(i)} = 1) \qquad \text{Draw sample from this component}$$

Discard sampled $z^{(i)}$ and end up with valid data samples $\boldsymbol{x}_i$ from the GMM

Mean Shift Clustering

Non-parametric approach to density estimation. How to estimate a PDF in a non-parametric way?
Identifies clusters by iteratively shifting data points towards the mode of the density distribution.
It uses a smoothing kernel (Parzen Window Technique):

$$f(x) = \sum_m K(x - x_m) = \sum_m k(\frac{x - x_m}{h})$$

Standard choice for a kernel K is a Gaussian:

$$k(x) = (2\pi)^{-d/2} exp(\frac{-1}{2}|x|^2)$$

The kernel has to be **radially symmetric**. Remember: a kernel is radially symmetric if its value only depends on the distance from the center and not the direction. In other words, K is radially symmetric if $K(x) = K(||x||)$, where $||x||$ is the Euclidean norm of x and k is a function of this norm.

A radially symmetric kernel ensures that **1)** the influence of a point in the feature space is uniform in all directions from the center. If not radially symmetric, the kernel would introduce directional bias, leading to incorrect identification of these regions. **2)** Since Mean Shift is doing gradient ascent on the density estimate, such kernels ensure that gradient ascent moves towards the nearest mode without directional bias, making the convergence more stable and predictable.

Other radially symmetric kernels:

- Epanechnikov kernel $K(x) = 1 - x$ for $|x| \leq 1$ and $K(x) = 0$ if $x > 1$
- Uniform kernel
- Gaussian kernel
  Steps:

1. Compute mean shift vector
2. Translate the Kernel window

Mean Shift clusters result from a few local kernel density estimates, thus being in between K-Means and GMM.

Steps:

1. Calculate the mean shift vector $m^{(t)}(\mathbf{x})$ for iteration t:

$$m^{(t)}(\mathbf{x}) = \frac{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2)} - \mathbf{x}^{(t)} \tag{11}$$

2. Update position $\mathbf{x}^{(t)}$:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + m^{(t)}(\mathbf{x}) = \frac{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} k'(\|\mathbf{x}_i - \mathbf{x}^{(t)}\|_2^2)} \tag{12}$$

   (note that $\mathbf{x}^{(t)}$ cancels, since it also occurs in $m^{(t)}(\mathbf{x})$)

3. Goto 1) until convergence (i.e., at a mode where gradient is 0)

Required parameters for mean shift clustering include the kernel parameters (window size) and cluster linking parameters for postprocessing.

Remarks:

Large kernels result in less clusters, while small kernels result in more clusters. The euclidean distance is sensitive to scaling differences -> use normalization of the dimension of samples.

Why can the Mean Shift algorithm be seen as the middle way between K-Means and GMM?
Answer: K-means based on minimizing distances between samples. GMM clusters based on full density estimation. Mean Shift is based on a few local kernel density estimates.