Clustering

Given a training set \( T = \{x_1, \ldots, x_n\} \), the \( K \)-means clustering algorithm aims to find cluster centers in such a way that points assigned to a given cluster are close to that cluster’s center. Note that \( K \)-means makes no reference to a statistical model for the data. This imposes constraints on the kinds of clusters that \( K \)-means detects.

We can create a simple but powerful statistical model for a single cloud of points by introducing a multivariate Gaussian \( N(x \mid \mu, \Sigma) \) over \( \mathbb{R}^D \). The maximum likelihood estimators for the parameters parameters of a Gaussian are given by

\[
\mu = \frac{1}{n} \sum_{i=1}^{n} x_i \in \mathbb{R}^D,
\]

\[
\Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T \in \mathbb{R}^{D \times D}.
\]

Mixture of Gaussians

Suppose we have some data set \( T = \{x_1, \ldots, x_n\} \) that we assume can be partitioned into \( K \) clusters. We will model each cluster as being characterized by one of \( K \) Gaussians. Define the indicator random variables

\[
\xi_{ij} = \begin{cases} 
1 & \text{if the } j\text{th Gaussian generated the } i\text{th point} \\
0 & \text{otherwise},
\end{cases}
\]

where \( i \in \{1, \ldots, n\} \) and \( j \in \{1, \ldots, K\} \). We can write the joint density as

\[
p(x, \xi) = p(\xi) \cdot p(x \mid \xi) = \pi_j \cdot N(x \mid \mu_j, \Sigma_j)
\]

Then we can sum over the latent variable \( \xi \) to find the density on \( X \),

\[
p(x) = \sum_{\xi} p(x, \xi) = \sum_{j=1}^{k} \pi_j N(x \mid u_j, \Sigma_j) \quad \text{(mixture of gaussians)}
\]
where $\xi$ is an indicator vector for 1 of $K$ gaussians, $\pi_j$ is the probability that a sample comes from cluster $j$, where $1 \leq j \leq k$, and $N(x \mid \mu_j, \Sigma_j)$ is the density associated with each cluster.

The parameters of the model are

$$\theta = \{ \pi_1, \ldots, \pi_k, \mu_1, \ldots, \mu_k, \Sigma_1, \ldots, \Sigma_k \}$$

**Maximum Likelihood**

Given $T = \{x_1, \ldots, x_n\}$, we want to find $\theta$ that achieves $\max_\theta P(T \mid \theta)$. As usual, we can instead work with the log likelihood.

$$L(\theta) = \log P(T \mid \theta) = \sum_{i=1}^n \log \left( \sum_{j=1}^k \pi_j N(x_i \mid \mu_j, \Sigma_j) \right) \tag{1}$$

We can also compute

$$\gamma_{ij}(\theta) = P(\xi_{ij} = 1 \mid x_i, \theta) = \frac{P(x_i \mid \xi_{ij} = 1, \theta)P(\xi_{ij} = 1 \mid \theta)}{P(x_i \mid \theta)} = \frac{N(x_i \mid \mu_j, \Sigma_j)\pi_j}{?}$$

where the “?” is a normalizing factor such that the constraint

$$\sum_j \gamma_{ij} = 1$$

is satisfied.

Returning to (1), we can set the derivative of $L(\theta)$ with respect to $\mu_j$ equal to 0, and after some algebra conclude

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^n \gamma_{ij}(\theta) x_i \tag{2}$$

$$N_j = \sum_{i=1}^n \gamma_{ij}(\theta)$$

This gives a formula for $\mu_j$ in terms of $\gamma_{ij}$ which itself depends on $\mu_j$, so we still don’t get a closed form solution for $\mu_j$.

We can similarly get equations for the $k$ covariance matrices,

$$\Sigma_j = \frac{1}{N_j} \sum_{i=1}^n \gamma_{ij}(x_i - \mu_j)(x_i - \mu_j)^T \tag{3}$$
Similarly,

\[ \pi_j = \frac{N_j}{n} \quad (4) \]

Since these formulae for the parameters depend themselves on the parameters, we first pick a set of parameters and update the parameter values iteratively. This leads us to the EM Algorithm.

**EM Algorithm**

We alternate estimation of \( \gamma \)'s and \( \theta \).

1. **Initialize** \( \mu_j, \Sigma_j, \pi_j \) for all \( j \).

2. **Update** \( \gamma_{ij} = \frac{N(x_i \mid \mu_j, \Sigma_j) \pi_j}{p(x_i \mid \theta)} \).

3. **Estimate** \( \mu_j, \Sigma_j, \pi_j \) for all \( j \) using the boxed expressions (2), (3), and (4) and the updated value of \( \gamma_{ij} \).

This may look reminiscent of the \( k \) means algorithm. In fact, if we let \( \Sigma_j = \epsilon I \), and take the limit of the \( \gamma_{ij} \) update as \( \epsilon \to 0 \), we indeed recover the \( k \) means algorithm.

**Jensen’s Inequality:** If \( f \) is a random variable and \( \phi \) is a convex function, then

\[ \phi(E(f)) \leq E(\phi(f)) \]

**Claim:** \( L(\theta) \) increases over iterations of this algorithm.

**Proof Sketch.** Recall that the likelihood is of the form

\[ L(\theta) = \sum_{i=1}^{n} \log \sum_{j=1}^{k} \pi_j N(x_i \mid \mu_j, \Sigma_j) \quad \left( = \sum_{1}^{n} \phi(E(f)) \right) \]

By Jensen’s,

\[ L(\theta) \geq \sum_{i=1}^{n} \sum_{j=1}^{k} r_{ij} \pi_j N(x_i \mid \mu_j, \Sigma_j) \quad \left( = \sum_{1}^{n} \phi(\phi(f)) \right) \]

for any collection of \( \{r_{ij}\} \) satisfying \( \sum_j r_{ij} = 1 \). (Recall that \( - \log \) is convex implies \( \log \) is concave, so Jensen’s is actually flipped). If we define \( B(\theta) \) as

\[ B(\theta) = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_{ij} \log \pi_j N(x_i \mid \mu_j, \Sigma_j) \]
By Jensen’s $L(\theta) \geq B(\theta)$. We can show that $B(\theta^t) = L(\theta^t)$, i.e. these two values are equal at the model parameters of the current step in the algorithm. This means that $\theta^{t+1}$ maximizes $B(\theta)$.

Then

$$L(\theta^{t+1}) \geq B(\theta^{t+1}) \geq B(\theta^t) = L(\theta^t)$$

Since $B$ is increasing in $t$ and assumed to be bounded above by the likelihood (we must assume the likelihood is bounded), the algorithm eventually converges.

**Remark:** It is possible to choose covariance matrices of the form $\varepsilon I$ and let $\varepsilon \to 0$ so that the gaussian concentrates all mass on a single point, and at that point the likelihood would go to infinity. To remedy this issue, we can restrict the search space of parameters to exclude such covariance matrices by using MAP estimation so that $P(\theta \mid T) \propto P(T \mid \theta)P(\theta)$ and have the $P(\theta)$ equal 0 for these parameters.