Computing the MLE and the EM Algorithm

If $X \sim p(x|\theta)$, $\theta \in \Theta$, then the MLE is the solution to the equations \( \frac{\partial \log p(x|\theta)}{\partial \theta} = 0 \). Sometimes these equations have a simple closed form solution, and other times they do not and we must use computational methods to find $\hat{\theta}$.

**Example 1** In some cases, the MLE is computed by taking a simple average. Suppose $X_i \overset{i.i.d.}{\sim} \text{Poisson}(\lambda)$. Then the MLE is $\hat{\lambda}_n = \frac{1}{n} \sum X_i$.

**Example 2** The MLE sometimes requires solving a system of linear equations. Suppose that $X \sim N(H\theta,I)$, where $H$ is $n \times k$ and known and $\theta$ is $k \times 1$ and unknown. Then the MLE is $\hat{\theta} = (H^T H)^{-1} H^T X$.

**Example 3** The MLE can also be the solution to a nonlinear system of equations. Suppose that $X_i \overset{i.i.d.}{\sim} pN(\mu_0,\sigma_0^2) + (1-p)N(\mu_1,\sigma_1^2)$, $i = 1,...,n$, and let $\theta = [p \ \mu_0 \ \sigma_0^2 \ \mu_1 \ \sigma_1^2]^T$

$$
p(x_i|\theta) = \frac{p}{\sqrt{2\pi\sigma_0^2}} e^{-\frac{(x_i-\mu_0)^2}{2\sigma_0^2}} + \frac{1-p}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}}
$$

**Figure 1:** Two-dimensional Gaussian mixture density.

The likelihood is a complicated nonlinear function. Moreover, it is non-convex in $\theta$.

$$
p(x|\theta) = \prod_{i=1}^{n} p(x_i|\theta), \text{ a product of sums of exponentials.}
$$

Taking the logarithm doesn’t simplify things:

$$
\log p(x|\theta) = \text{a sum of logs of a sum of exponentials.}
$$
Also recall that the sufficient statistic in this case is the whole set of data \((X_1, X_2, ..., X_n)\); i.e., there is no small sufficient statistic that summarizes them.

What can we do in such situations? We need a computational method to maximize the likelihood function. There are two common approaches:

1. **Gradient/Newton methods**
   \[
   \theta^{(t+1)} = \theta^{(t)} + \Delta \frac{\partial}{\partial \theta} \log p(x|\theta) |_{\theta=\theta^{(t)}}, \text{ where } \Delta > 0 \text{ is a step size.}
   \]

2. **Expectation-Maximization Algorithm (EM algorithm)**

Gradient ascent methods should be familiar to most readers. The EM algorithm is a specialized approach designed for MLE problems, and it has some attractive properties, namely it doesn’t require specification of a step size and under mild conditions it is guaranteed to converge to a local maximum of the likelihood function. If the likelihood function is concave (i.e., negative log-likelihood is convex), then convergence to a global maximum likelihood point is possible using gradient methods or EM. The rest of the lecture discusses the EM algorithm.

### 1 The EM Algorithm

In many problems MLE based on observed data \(X\) would be greatly simplified if we had additionally observed another piece of data \(Y\). \(Y\) is called the hidden or latent data.

**Example 4** \(X \sim \mathcal{N}(H\theta, I)\) can be modeled as:

\[
\begin{align*}
Y_{1 \times 1} & = \theta + W_1 \\
X_{n \times 1} & = H_{n \times k}Y + W_2
\end{align*}
\]

such that \(HW_1 + W_2 \sim \mathcal{N}(0, I)\).

If we just have \(X\), then we must solve a system of equations to obtain the MLE. If the dimension is large, then computing the MLE is quite expensive (i.e. the inversion is at least \(O(\max(nk^2, k^3))\)). But if we also have \(Y\), then the MLE can be computed with \(O(k)\) as we know \(\hat{\theta} = Y\).

**Example 5**

\[
\begin{align*}
x_i & \overset{iid}{\sim} p\mathcal{N}(\mu_0, \sigma^2_0) + (1-p)\mathcal{N}(\mu_1, \sigma^2_1) \\
y_i & \overset{iid}{\sim} \text{Bernoulli}(p) = p^{y_i}(1-p)^{1-y_i} \\
x_i|y_i = l & \sim \mathcal{N}(\mu_l, \sigma^2_l)
\end{align*}
\]

Given \(\{(x_i, y_i)\}_{i=1}^n\), we have:

\[
\begin{align*}
\hat{\mu}_l & = \frac{1}{\sum 1_{y_i = l}} \sum 1_{y_i = l} x_i \\
\hat{\sigma}^2_l & = \frac{1}{\sum 1_{y_i = l}} \sum 1_{y_i = l} (x_i - \hat{\mu}_l)^2 \\
\hat{p} & = \frac{\sum 1_{y_i = l}}{n}
\end{align*}
\]

MLE’s are easy to compute here. However, if we only have \(\{x_i\}_{i=1}^n\), the computation of MLE is a complicated, non-convex optimization, where we can apply EM algorithm to compute. The application of EM algorithm in this situation is shown in **Example 4**.
Main Idea

Let \( L(\theta) = \log p(x|\theta) \) and also define the complete data log-like:

\[
L_c(\theta) = \log p(x, y|\theta) = \log p(y|x, \theta)p(x|\theta) = \log p(y|x, \theta) + \log p(x|\theta) = \log p(y|x, \theta) + L(\theta)
\]

Suppose our current guess of \( \theta \) is \( \theta(t) \) and that we would like to improve this guess. Consider

\[
L(\theta) - L(\theta(t)) = L_c(\theta) - L_c(\theta(t)) + \log \frac{p(y|x, \theta(t))}{p(y|x, \theta)}
\]

Now take expectation of both sides with respect to \( y \sim p(y|x, \theta(t)) \), we have:

\[
L(\theta) - L(\theta(t)) = \mathbb{E}_{y \sim p(y|x, \theta(t))} [L_c(\theta) - L_c(\theta(t))] + D(p(y|x, \theta(t)) \parallel p(y|x, \theta))
\]

Since \( D(p(y|x, \theta(t)) \parallel p(y|x, \theta)) \geq 0 \), we have the following inequality:

\[
L(\theta) - L(\theta(t)) \geq \mathbb{E}_{y \sim p(y|x, \theta(t))} [L_c(\theta) - L_c(\theta(t))] = Q(\theta, \theta(t)) - Q(\theta(t), \theta(t))
\]

where \( Q(\theta, \theta') := \mathbb{E}_{y \sim p(y|x, \theta')} [\log p(x, y|\theta)] \) is the expectation of complete data log-likelihood. We choose \( \theta(t+1) \) as the solution of the following optimization problem:

\[
\theta(t+1) = \arg \max_{\theta} Q(\theta, \theta(t))
\]

The EM algorithm is an attractive option if the \( Q \) function is easily computed and optimized. The relationship between \( \log p(x, \theta) \), \( Q(\theta, \theta(t)) \), \( \theta(t) \) and \( \theta(t+1) \) are depicted in the following figure:

![Figure 2: Graphical show of EM algorithm](image)

The process of EM algorithm is as follows:

**Init:** \( t = 0 \), \( \theta(0) = 0 \) or random value

**for** \( t=0,1,2,\ldots \)

**E step:**

\[
Q(\theta, \theta(t)) = \mathbb{E}_{p(y|x, \theta(t))} [\log p(x, y|\theta)]
\]

**M step:**

\[
\theta(t+1) = \arg \max_{\theta} Q(\theta, \theta(t))
\]

The E-step and M-step repeat until convergence. The two key properties of the EM algorithm are:
1. \( \log p(x|\theta^{(0)}) \leq \log p(x|\theta^{(1)}) \leq \ldots \)

2. It converges to stationary point (e.g., local max)

Now let’s look at a few applications of the EM algorithm. The EM algorithm is especially attractive in cases where the \( Q \) function is easy to compute and optimize. There is a bit of art involved in the choice of the hidden or latent data \( Y \), and this needs to be worked out on a case-by-case basis.

**Example 6**

Original model \( X = H\theta + W \):

**Complete model:**

\[
\begin{align*}
Y &= \theta + W_1 \quad W_1 \sim \mathcal{N}(0, \alpha^2 I_{k \times k}) \\
X &= H_{n \times k} Y + W_2 \quad W_2 \sim \mathcal{N}(0, I_{n \times n} - \alpha^2 HH^T)
\end{align*}
\]

Then we construct the complete log-likelihood:

\[
\log p(x, y|\theta) = \log p(x|y|\theta) + \log p(y|\theta) = \text{constant} - \frac{||y - \theta||^2}{2\alpha^2}
\]

\[
= \frac{1}{2\alpha^2} (2\theta^T y - \theta^T \theta - y^T y) + \text{constant}
\]

As the part left after taking away the constant is proportional to \( y \), so we only need to calculate \( \mathbb{E}_{p(y|x|\theta^{(t)})}[y] \).

Introduce \( Z_1 = Y, Z_2 = X - HY \), then we have the joint distribution of \( Z_1, Z_2 \) as:

\[
\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \theta \\ 0 \end{bmatrix}, \begin{bmatrix} \alpha^2 I_{k \times k} & 0 \\ 0 & I_{n \times n} - \alpha^2 H H^T \end{bmatrix} \right)
\]

As we know \( \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} H & I_{n \times n} \\ I_{k \times k} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \), we know:

\[
\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \theta \\ \theta \end{bmatrix}, \begin{bmatrix} I_{n \times n} & \alpha^2 H \\ \alpha^2 H^T & \alpha^2 I_{k \times k} \end{bmatrix} \right)
\]

Make a linear transformation, we have:

\[
\begin{bmatrix} X \\ Y - \alpha^2 H^T X \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \theta \\ -\alpha^2 H^T \theta \end{bmatrix}, \begin{bmatrix} I_{n \times n} & 0 \\ 0 & \alpha^2 I_{k \times k} - \alpha^4 H^T H \end{bmatrix} \right)
\]

So we have:

\[
\mathbb{E}_{p(y|x|\theta^{(t)})}[y] = \alpha^2 H^T x + \theta^{(t)} - \alpha^2 H^T \theta^{(t)} = y^{(t)}
\]

As \( Q(\theta, \theta^{(t)}) = \frac{1}{2\alpha^2} (2\theta^T y^{(t)} - \theta^T \theta) + \text{constant} \), set \( \frac{\partial Q}{\partial \theta} = 0 \), we have:

\[
\theta^{(t+1)} = y^{(t)}
\]

It is easy to calculate the stationary point in this iteration, let \( \theta^{(t+1)} = \theta^{(t)} \), we have:

\[
\theta_{\text{stationary}} = (H^T H)^{-1} H^T x
\]

which is the answer we are familiar with.
Example 7 Suppose:

\[ X_1, X_2, \ldots, X_n \overset{iid}{\sim} \sum_{j=1}^{m} p_j N(\mu_j, \sigma_j^2) \]

We have:

\[ p(x, y | \theta) = \prod_{i=1}^{n} \sum_{j=1}^{m} p_j \frac{1}{\sqrt{2\pi} \sigma_j} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}} 1_{y_i = j} \]

Thus,

\[ \log p(x, y | \theta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \log \left( \frac{p_j}{\sqrt{2\pi} \sigma_j} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}} \right) 1_{y_i = j} \]

\[ \mathbb{E}_{p(y|x|\theta^{(t)})}[\log p(x, y | \theta)] = \sum_{i=1}^{n} \sum_{j=1}^{m} \log \left( \frac{p_j}{\sqrt{2\pi} \sigma_j} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}} \right) \mathbb{E}_{p(y|x|\theta^{(t)})}[1_{y_i = j}] \]

Denote \( p^{(t)}(y_i = j) = \frac{p_j^{(t)} N(x_i; \mu_j^{(t)}, \sigma_j^{(t)}^2)}{\sum_{l=1}^{n} p_l^{(t)} N(x_i; \mu_l^{(t)}, \sigma_l^{(t)}^2)} \), we have the expression of \( Q(\theta, \theta^{(t)}) \):

\[ Q(\theta, \theta^{(t)}) = \sum_{i=1}^{n} \sum_{j=1}^{m} p^{(t)}(y_i = j) \log (p_j^{(t)} N(x_i; \mu_j, \sigma_j^2)) \]

\[ = \sum_{i=1}^{n} \sum_{j=1}^{m} p^{(t)}(y_i = j) \log (N(x_i; \mu_j, \sigma_j^2)) + \text{constant} \]

Set \( \frac{\partial Q}{\partial \theta} = 0 \), we have:

\[ \mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} p^{(t)}(y_i = j) x_i}{\sum_{i=1}^{n} p^{(t)}(y_i = j)} \]

\[ (\sigma_j^{(t+1)})^2 = \frac{\sum_{i=1}^{n} (x_i - \mu_j^{(t+1)})^2 p^{(t)}(y_i = j)}{\sum_{i=1}^{n} p^{(t)}(y_i = j)} \]