**The Expectation Maximization (EM) Algorithm**

gustavoc@msu.edu

**Brief description of the algorithm**

The Expectation-Maximization (EM) algorithm is often used to obtained maximum likelihood (ML) estimates in cases where the likelihood function does not have a closed form. The algorithm can also be used for estimation of the posterior mode in Bayesian models; however, we will be focused on ML estimation. The algorithm was formalized by Dempster, Laird, and Rubin (J. Royal Soc. B, 1977).

The ***likelihood function*** is the joint distribution of the data ($\boldsymbol{y}_o$), given the parameters ($\boldsymbol{\theta}$), viewed as a function of the model parameters. In many cases the likelihood function has a closed form; therefore, we can obtain ML estimates using standard analytical or numerical approaches such as the ones discussed in the first half of the course. In other cases, the likelihood of the observed data may not have a closed form; in these cases the EM-algorithm may be useful. The nature of the 'unobserved' random variables (or missing data) may include random effects, censored data points or latent variables (e.g., hidden states in a hidden Markov model).

The ***complete-data likelihood*** is the joint distribution of the observed and the missing data given the parameters, viewed as a function of the parameters, that is, $L_C(\boldsymbol{\theta}) = p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta})$. Since $\boldsymbol{y}_m$ is unobserved, we cannot evaluate the complete likelihood directly. However, this object will become useful in one of the steps of the EM-algorithm.

The EM-algorithm is an iterative procedure; at each iteration, we complete two steps: The E (or 'expectation') step, followed by the M (or maximization) step.

---

**Outline of the EM-algorithm**

- **Initialize**: set $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$ where $\boldsymbol{\theta}^{(0)}$ is a value within the parameter space,
- **Iterate** until convergence the following steps:
  - **E-step**: find $Q(\boldsymbol{\theta}, \boldsymbol{y}_o) = E_{\boldsymbol{y}_m|\boldsymbol{y}_o,\boldsymbol{\theta}_{t-1}}\{\log\{L_c(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta}_{t-1})\}$.
  - **M-step**: maximize $Q(\boldsymbol{\theta}, \boldsymbol{y}_o)$ with respect to $\boldsymbol{\theta}$ and set

$$\widehat{\boldsymbol{\theta}}^{(t)} = \underset{argmax}{\boldsymbol{\theta}} \; Q(\boldsymbol{\theta})$$

---

The expression involved in the E-step is the expected value of the logarithm of the complete-data. The expectation is taken with respect to the conditional distribution of the missing data given the observed data and the current parameter values. The resulting expression, $Q(\boldsymbol{\theta}, \boldsymbol{y}_o)$ only depends on the parameters and the observed data. In the M-step we maximize this function with respect to the parameters.

## Example 1: Maximum likelihood estimation with right-censored data

Suppose that $y_i \sim Exponential(\lambda)$ is a time-to-event variable following an exponential distribution. We have a sample consisting of n pairs of the form $(y_i, d_i)$ where $y_i$ is time to event or time to censoring and $d_i$ is a dummy variable indicating whether $y_i$ is time-to-event ($d_i = 1$) or time to censoring ($d_i = 0$). Thus, our observed data consist of all the times to events and our missing data consists of all the unknown times to event ($\tilde{y}_{mi}$) of the censored observations, that is: $\boldsymbol{y}_o = \{y_i : d_i = 1\}$ **&** $\tilde{\boldsymbol{y}}_m = \{\tilde{y}_{mi} : d_i = 0, \}$**.** For the unobserved time to events, since data is right-censored, the unobserved times must be greater than the censoring times: $\tilde{y}_{mi} > y_i$. Thus, complete data likelihood takes the form

$$p(\boldsymbol{y}_o, \tilde{\boldsymbol{y}}_m | \boldsymbol{d}, \lambda) = \prod\nolimits_{d_i=1} \lambda e^{-\lambda y_i} \prod\nolimits_{d_i=0} \lambda e^{-\lambda \tilde{y}_{mi}}$$

The first term in the right-hand side is the likelihood of the observed data and the second component is the component of the complete data-likelihood involving the censored observations (recall that $\tilde{y}_i$ is missing).

Simplifying the right-hand-side we get

$$p(\boldsymbol{y}_o, \tilde{\boldsymbol{y}}_m | \boldsymbol{d}, \lambda) = \lambda^{n_o + n_m} e^{-\lambda\left(\sum_{d_i=1} y_i + \sum_{d_i=0} \tilde{y}_{mi}\right)}$$

where $n_o = \sum_i d_i$ is the number of observed time-to-event and $n_c = \sum_i 1 - d_i$ is the number of right-censored data points. The logarithm of the complete-data likelihood becomes

$$l_c = (n_o + n_m) \log(\lambda) - \lambda \left( \sum\nolimits_{d_i=1} y_i + \sum\nolimits_{d_i=0} \tilde{y}_{mi} \right)$$

**E-step**

In the E-step, we take the expected value of $l_c$ with respect to the missing data ($\tilde{y}_{mi}$) conditional on the parameters and the observed data. For right-censored data point we need to condition on the fact that the missing data must be greater than the censoring time ($\tilde{y}_{mi} > y_i$), thus

$$E[l_c | \boldsymbol{y}, \boldsymbol{d}] = (n_o + n_m) \log(\lambda) - \lambda \sum\nolimits_{d_i=0} E[\tilde{y}_{mi} |\, \tilde{y}_{mi} > y_i]$$

The conditional mean of an exponential random variable can be shown to be

$$E[\tilde{y}_{mi} |\, \tilde{y}_{mi} > y_i] = y_i + \lambda \qquad [1]$$

, thus

$$Q(\lambda) = E[l_c|\boldsymbol{y}, \boldsymbol{d}] = (n_o + n_m)\log(\lambda) - \lambda\left(\sum_{d_i=i} y_i + \sum_{d_i=0} y_i^*\right)$$

where $y_i^* = E[\tilde{y}_{mi}|\, \tilde{y}_{mi} > y_i] = y_i + \lambda$.

We can see that $Q(\lambda)$ is equivalent to the log-likelihood of exponential data, with the right-censored data 'imputed' with its conditional expectation.

**M-step**

Differentiating $Q(\lambda)$ with respect to $\lambda$ gives

$$\frac{\partial Q(\lambda)}{\partial \lambda} = \frac{(n_o + n_m)}{\lambda} - (n_o + n_m)\bar{y}$$

where $\bar{y}=\frac{\left(\sum_{d_i=i} y_i + \sum_{d_i=0} y_{mi}^*\right)}{(n_o+n_m)}$ is the mean of the observed and 'imputed' data. Thus, the first order condition is

$$\frac{(n_o+n_m)}{\hat{\lambda}} = (n_o + n_m)\bar{y} \Leftrightarrow \hat{\lambda} = \frac{1}{\bar{y}} \qquad [2]$$

Therefore, the EM-algorithm consists of the following steps:

- Initialize $\lambda$ to a positive value (e.g., the inverse of the mean of the non-censored data).
- Iterate until convergence:
  - **E-step**: Impute the censored data with the conditional expectation [1],
  - **M-step**: Update $\lambda$ using [2]

**Simulating right-censored data**

```
set.seed(195021)
 n=1000
 lambda0=4
 y=rexp(n=n,rate=lambda0)
 # let's consider fixed censoring time
 d=y<0.3 # TRUE here indicate event and FALSE right-censored

 yCen=y; yCen[!d]=0.3 # this is the data we observe
```

**EM-algorithm**

```
lambda=rep(NA,100) # a vector to store estimates iterations
lambda[1]=1/mean(y[d]) # initial value (estimate ignoring censoring)
completeData=yCen  # this vector stores the 'complete' data

for(i in 2:length(lambda)){
   # E-step (imputing the right-censored data)
   completeData[!d]=yCen[!d]+1/lambda[i-1]
   # M-step
   lambda[i]=1/mean(completeData)
}

# Trace plot to check convergence
 plot(lambda,type='o')
 tail(lambda)

#MLE if there were no censoring
 1/mean(y)
```

## Example 2: Mixture of Gaussians

In a finite mixture model, the density of a RV is modeled as the weighted sum of a finite number of densities. Here, we consider finite mixtures with Gaussian components. The density function of a mixture with 2 Gaussian components is:

$$p(x_i|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\alpha_1) = \alpha_1(2\pi\sigma_1^2)^{-\frac{1}{2}}e^{-\frac{(x_i-\mu_1)^2}{2\sigma_1^2}} + (1-\alpha_1)\,(2\pi\sigma_2^2)^{-\frac{1}{2}}e^{-\frac{(x_i-\mu_2)^2}{2\sigma_2^2}}$$

Above, $\mu_.$ and $\sigma_.^2$ and the means and variances of each of the components and $0 < \alpha_1 < 1$ is a mixture proportion. The model can be naturally extended to K components and multivariate data (i.e., when $x_i$ is a vector following a multivariate normal distribution).

**Likelihood Function**

Assuming IID data, the joint density of the data is

$$p(x_1,\dots,x_n|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\alpha_1) = \prod_{i=1}^{n} \alpha_1 dnorm(x_i|\mu_1,\sigma_1^2) + (1-\alpha_1)\,dnorm(x_i|\mu_2,\sigma_2^2)$$

Where $dnorm(x_i|\mu_*,\sigma_*^2) = (2\pi\sigma_*^2)^{-\frac{1}{2}}e^{-\frac{(x_i-\mu_*)^2}{2\sigma_*^2}}$.

The model parameters, $\boldsymbol{\theta} = \{\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \alpha_1\}$, can be estimated via maximum likelihood. Maximization could be done using general purpose optimization algorithms (e.g., those implemented in `optim()` ) or, alternatively, we can maximize the likelihood using the EM-algorithm.

**Augmented likelihood**

To facilitate the implementation of the EM-algorithm we introduce a latent variable $z_i \in \{1,2\}$ which indicates whether the ith observation comes from 1st or 2nd component of the mixture. Parameter $\alpha_1$ can be interpreted as the proportion of the observations coming from the first component, that is $p(z_i = 1|\alpha_1) = \alpha_1$ and $p(z_i = 2|\boldsymbol{\theta}) = 1 - \alpha_1$. Clearly, $p(z_i|\alpha_1) = \alpha_1^{1(z_i=1)}(1 - \alpha_1)^{1(z_i=2)}$ where $1(z_i = 1) = \{1\ if\ z_i = 1\ ;\ 0\ otherwise\}$ and $1(z_i = 2) = \{1\ if\ z_i = 2\ ;0\ otherwise\}$

The augmented (or complete-data) likelihood is the joint distribution of the observed (x) and the missing (z) data given the parameters ($\boldsymbol{\theta}$). For the ith-data point the augmented likelihood is:

$$p(x_i, z_i|\boldsymbol{\theta}) = p(x_i|z_i, \boldsymbol{\theta}) \times p(z_i|\boldsymbol{\theta})$$
$$= dnorm(x_i|\mu_1, \sigma_1^2)^{1(z_i=1)} dnorm(x_i|\mu_2, \sigma_2^2)^{1(z_i=2)} \times \alpha_1^{1(z_i=1)}(1 - \alpha_1)^{1(z_i=2)}$$

Therefore, assuming IID data the complete-data likelihood becomes

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \prod_{i=1}^{n} dnorm(x_i|\mu_1, \sigma_1^2)^{1(z_i=1)} dnorm(x_i|\mu_2, \sigma_2^2)^{1(z_i=2)} \times \alpha_1^{1(z_i=1)}(1 - \alpha_1)^{1(z_i=2)}$$

**E-step**

To perform the E-step we must derive the expected value of the logarithm of the complete-data likelihood with respect to the distribution of the missing data (z) given the observed data (x) and the parameters, that is $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$. The logarithm of the complete likelihood takes the form

$$l(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \sum_{i=1}^{n} 1(z_i = 1) \log\left[dnorm(x_i|\mu_1, \sigma_1^2)\ \right] + 1(z_i = 2) \log\left[dnorm(x_i|\mu_2, \sigma_2^2)\ \right] + 1(z_i = 1) \log[\alpha_1] + 1(z_i = 2) \log[1 - \alpha_1]$$

The expected value the above expression is

$$l(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta}) = \sum_{i=1}^{n} w_i \log\left[dnorm(x_i|\mu_1, \sigma_1^2)\ \right] + (1 - w_i) \log\left[dnorm(x_i|\mu_2, \sigma_2^2)\ \right] + w_i \log[\alpha_1] + (1 - w_i) \log[1 - \alpha_1] \quad [1]$$

Where $w_i = E(z_i = 1|x_i, \boldsymbol{\theta}) = p(z_i = 1|x_i, \boldsymbol{\theta})$.

Using Bayes' rule

$$p(z_i = 1|x_i) = \frac{p(x_i|z_i=1)p(z_i=1)}{p(x_i)} = \frac{p(x_i|z_i=1)p(z_i=1)}{p(x_i|z_i=1)p(z_i=1)+p(x_i|z_i=2)p(z_i=2)} = \frac{A}{A+B}$$

Where $A = dnorm(x_i|\mu_1, \sigma_1^2) \times \alpha_1$ and $B = dnorm(x_i|\mu_2, \sigma_2^2) \times (1 - \alpha_1)$, thus

$$w_i = \frac{dnorm(x_i|\mu_1,\sigma_1^2) \times \alpha_1}{dnorm(x_i|\mu_1,\sigma_1^2) \times \alpha_1 + dnorm(x_i|\mu_2,\sigma_2^2) \times (1-\alpha_1)} \quad [2]$$

**M-Step:**

In the M-step we maximize [1] with respect to each of the parameters of the mixture. Note that [1] is a weighted log-likelihood. It can be shown that the ML estimates of the parameters are given by the following weighted means and weighted variances:

$$\hat{\mu}_1 = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \; ; \; \hat{\mu}_2 = \frac{\sum_{i=1}^n (1-w_i) y_i}{\sum_{i=1}^n (1-w_i)}; \; \hat{\sigma}_1^2 = \frac{\sum_{i=1}^n w_i (y_i - \hat{\mu}_1)^2}{\sum_{i=1}^n w_i} \; ; \; \hat{\sigma}_2^2 = \frac{\sum_{i=1}^n (1-w_i)(y_i - \hat{\mu}_2)^2}{\sum_{i=1}^n (1-w_i)}; \; \hat{\alpha}_1 = \frac{\sum_{i=1}^n w_i}{n} \quad [3]$$

Thus, EM-algorithm iterates using the following steps

> **E-step**: Compute the weights using [2] evaluated at the current parameter values
> **M-step**: update parameters using the expressions in [3]

An implementation of this algorithm is given in the github repository.