Overview of the Expectation Maximization (EM) algorithm

gustavoc@msu.edu

Expectation-Maximization is an algorithm for maximum likelihood and maximum a posteriori estimation. Here, we concentrate on the use of EM in the context of maximum likelihood. The algorithm is particularly useful for cases where the likelihood function does not have a closed form but an augmented version of it can be maximized easily. The algorithm was formalized by Dempster, Laird and Rubin (1977); however, earlier versions appeared before in Baum, Petrie Soules and Weiss (1970). The algorithm formalizes an idea earlier used to deal with missing data. Suppose we have a likelihood function of the form $L(\boldsymbol{\theta}) = p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta})$. Where $\boldsymbol{y}_.$ represents data and $\boldsymbol{\theta}$ is a parameter vector. The maximum likelihood estimate of $\boldsymbol{\theta}$ is

$$\widehat{\boldsymbol{\theta}} \overset{=}{_{argmax}} \; p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta})$$

Now, suppose that we only observe $\boldsymbol{y}_o$ and $\boldsymbol{y}_m$is missing (i.e., un-observed). Technically, the likelihood is the marginal distribution of the data given the parameters, viewed as function of the parameters, that is $p(\boldsymbol{y}_o|\boldsymbol{\theta}) = \int p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta}) d\,\boldsymbol{y}_m$. The EM-algorithm is particularly useful in dealing with cases where this integral does not have a closed form

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

- **Initialize**: set $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ where $\boldsymbol{\theta}_0$ is a value within the parameter space,
- For *t=1,…* iterate until convergence the following steps
    1. **E-step**: Impute the missing data with its conditional expectation, that is set $\boldsymbol{y}_m^{(t)} = E(\boldsymbol{y}_m|\boldsymbol{\theta}_{t-1}, \boldsymbol{y}_o)$
    2. **M-step**: Maximize the likelihood treating the imputed data as observed, that is set
    $$\boldsymbol{\theta}_t \overset{=}{_{argmax}} \; p\left(\boldsymbol{y}_o, \boldsymbol{y}_m^{(t)}|\boldsymbol{\theta}_{t-1}\right)$$

Above, $p\left(\boldsymbol{y}_o, \boldsymbol{y}_m^{(t)}|\boldsymbol{\theta}\right)$ is referred as to the ‘complete' data likelihood, that is the likelihood that we would use if there were no complete data. The EM-algorithm alternates between an Expectation and Maximization steps.

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 $(d_i = 0)$. Technically, the time-to-event variable is missing for all the data points with $d_i = 0$. The following scripts simulates right-censored exponential data.

```
set.seed(195021)
n=100
y=rexp(n=n,rate=4)
# 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
```

The mean of an exponential random variable with rate λ is $E[y_i] = \frac{1}{\lambda}$, in this case 1/4. Ignoring the right-censored data leads to serious downward-bias:

```
# estimate with the 'complete' data
   mean(y)
[1] 0.2193996

# estimate ignoring censoring
  mean(yCen[d])
[1] 0.08742043
```

The exponential distribution has the following (memoryless) property $E[y_i|\tau\,] = \tau + 1/\lambda$. Furthermore, in absence of missing values (complete-data likelihood) the maximum likelihood estimate of the rate can be shown to be: $\hat{\lambda} = \frac{1}{\bar{y}}$ where $\bar{y}$ is the sample mean. We use these two results for implementing the EM-algorithm below.

```
lambda=rep(NA,10) # a vector to store estimates iterations
lambda[1]=1/mean(y[d]) # initial value (estimate ignoring censoring)
completeData=y  # this vector stores the 'complete' data
for(i in 2:length(lambda)){
   # E-step
   completeData[!d]=y[!d]+1/lambda[i-1]
   # M-step
   lambda[i]=1/mean(completeData)
}
round(1/lambda,3)

[1] 0.087 0.247 0.299 0.315 0.320 0.322 0.322 0.323 0.323 0.323
```

Formal Definition of the EM-Algorithm

More formally the EM-algorithm consists of the following steps:

- **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 $Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}) = E_{\boldsymbol{y}_m|\boldsymbol{y}_o,\boldsymbol{\theta}_{t-1}}\{\log[p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta}_{t-1})]\}$
    - **M-step**: set $\boldsymbol{\theta}_t = argmax\ Q(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$

When the distribution $p(\boldsymbol{y}_o, \boldsymbol{y}_m|\boldsymbol{\theta}_{t-1})$ belongs to the Exponential family the E-step reduces to impute the missing data to its conditional expectation; however, in other cases the E-step may have a different form.