Eliot Paisley
STA 250
11-27-13
Prof. Baines
U.C. Davis

Assignment 3

EM and Optimization Module

1. In this problem you will implement the most basic optimization algorithms presented in class: bisection and Newton Raphson.

   a. Write a general function to implement the bisection algorithm.
      Answer: Code included at the end of this report.

   b. Write a general function to implement the Newton-Raphson algorithm.
      Answer: Code included at the end of this report.

   c. Consider the classic linkage problem from Rao (1969) with probabilities and counts:

Phenotype	Probability	Count
AB	$(3-2\theta+\theta^2)/4$	125
Ab	$(2\theta-\theta^2)/4$	18
aB	$(2\theta-\theta^2)/4$	20
ab	$(1-2\theta+\theta^2)/4$	34

Defining $\lambda = 1 - 2\theta + \theta^2$, the likelihood is seen to be:

$$L(\lambda) \propto (2+\lambda)^{125}(1-\lambda)^{18+20}\lambda^{34} \tag{1}$$

Use both of your functions to find the MLE for λ in the linkage example.

Answer: In Fig. 1 we see the plot of the likelihood given above in (1) over two different ranges. Visually, we identify a local maximum somewhere near $\lambda = -0.6$, and a global maximum near $\lambda = 0.6$. Our goal is to identity the global maximum. In Fig. 2, we plot the first derivative of the log-likelihood. Here we note two possible roots; one between $(-1, 0)$, and the other between $(0, 1)$. This agrees with our observation of the likelihood.

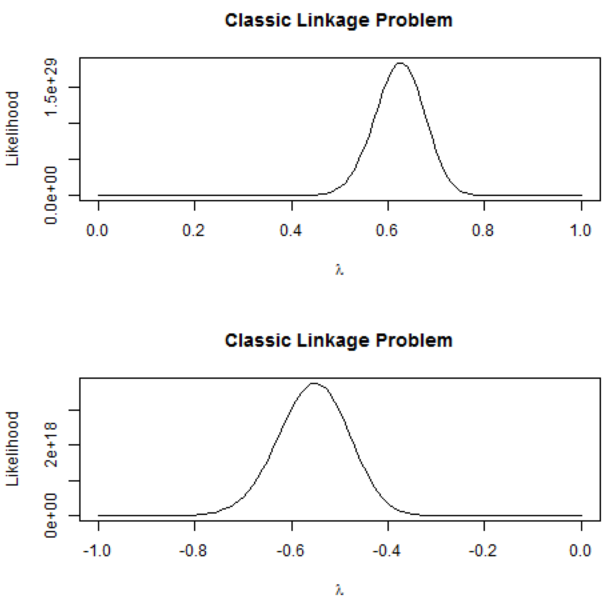

Figure 1: Plot of the likelihood function over two disjoint ranges.

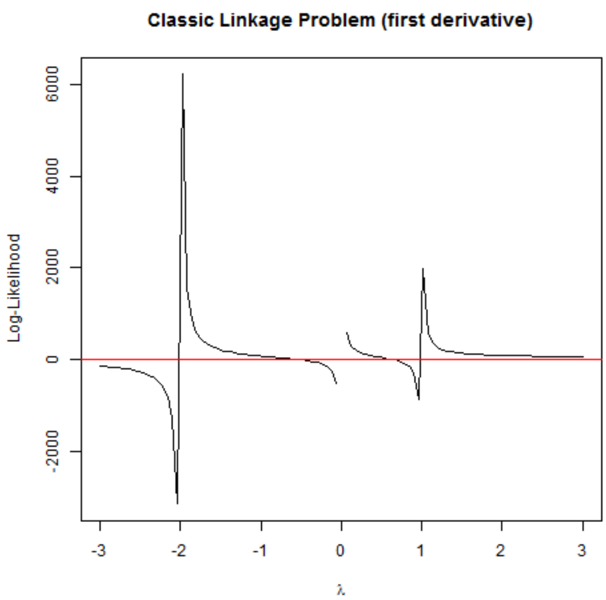

Figure 2: Plot of the first derivative of the logarithm of the likelihood function.

For the bisection method, using a tolerance of $1E07$ and a maximum of 1000 iterations, we identified two roots depending on the initial interval. Table 1 summarizes these results. When applying the Newton-Raphson method, the convergence of the algorithm is very sensitive to the choice of the initial value. In Table 2 we summarize the results of the NR method over three different starting values. Here we see that when starting at nearly the exact value of the local maximum ($\lambda \sim -0.55$), the NR method fails to converge. Also, when we choose an inital value near the global maximum, $\lambda = 0.4$, the algorithm fails to converge as well. Only when we start much closer to the actual maximum, $\lambda = 0.6$, do we see convergence.

Root	Value	Iterations	Int. Low	Int. Upp
-0.5507	0.0000	27	-1.9999	-0.0001
0.6268	0.0000	26	0.0001	0.9999

Table 1: Results from applying the bisection method to two different starting intervals.

In Figures 3 and 4, we show the plot of the original likelihood function, along with the maximums as found by the bisection method, and Newton-Raphson method.

□

Root	Value	Iteration	Initial
-1.e-26	-2e+27	10	-0.55
-1.e-22	-3e+22	8	0.40
0.6268	-2e-05	80	0.60

Table 2: Results from applying the Newton-Raphson method to three different starting values.

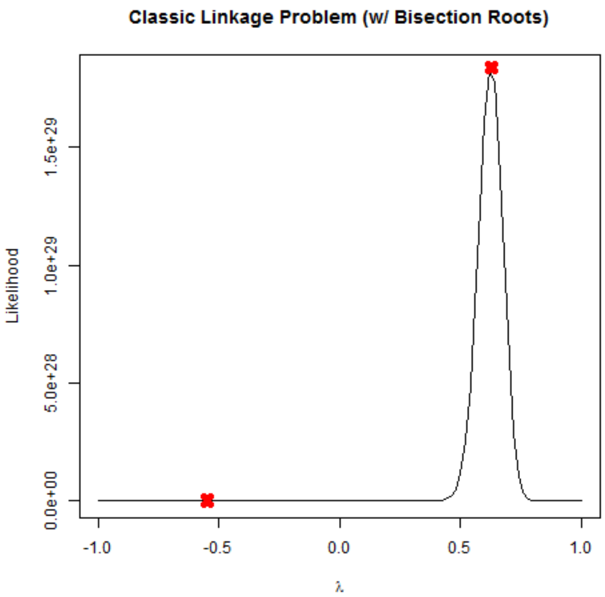

Figure 3: Plot of the likelihood function and locations of the two maximums found by the bisection method.

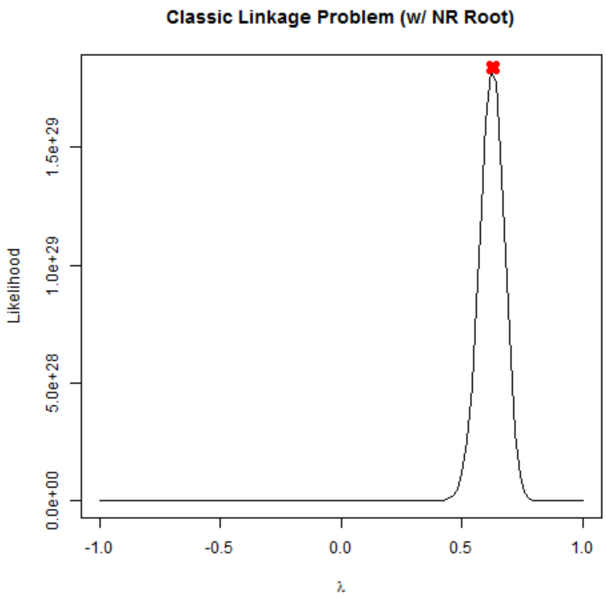

Figure 4: Plot of the likelihood function and location of the maximum found by the Newton-Raphson method.

---

2. In this question, you will implement EM for a multivariate-t regression model:

$$Y_i \sim t_{\mu}, \Psi, \nu), \quad i = 1, \ldots, n \tag{2}$$

where t_p is a multivariate t-distribution in p dimensions. It turns out that the t-distribution has a convenient representation as a ratio of a normal and χ^2 distribution. Therefore, we can write the model in(2) as:

$$Y_i|\tau_i, \theta \sim N_p\left(\mu, \frac{1}{\tau_i}, \Psi\right), \tag{3}$$

$$\tau_i \sim \frac{\chi^2_\nu}{\nu}, \tag{4}$$

where $\theta = (\mu, \Psi)$ is the parameter to be estimated (ν is fixed) with $Y_{obs} = (Y_1, \ldots, Y_n)$ and $Y_{mis} = (\tau_1, \ldots, \tau_n)$. For convenience, define

$$\tau_i^{(t+1)} = \mathbf{E}\left[\tau_i | Y_{obs}, \theta^{(t)}\right] = \frac{\nu + p}{\nu + \left(Y_i - \mu^{(t)}\right)^T \left[\Psi^{(t)}\right]^{-1}\left(Y_i - \mu^{(t)}\right)}. \tag{5}$$

a. Derive the EM algorithm for estimating θ i.e. find $\mu^{(t+1)}$ and $\Psi^{(t+1)}$.
Answer: We begin by writing down the likelihood functions for $Y_i|\tau_i, \theta \sim N_p\left(\mu, \frac{1}{\tau_i}, \Psi\right)$, and $\tau_i \sim \frac{\chi^2_\nu}{\nu}$.

$$\begin{aligned}
p(\vec{Y}|\vec{\tau}, \theta) &= \prod_{i=1}^n p(Y_i|\tau_i, \theta) \\
&= \prod_{i=1}^n (2\pi)^{-p/2} \left|\frac{1}{\tau_i}\Psi\right|^{-1/2} \exp\left\{-\frac{1}{2}(Y_i - \mu)^T \left(\frac{1}{\tau_i}\Psi\right)^{-1}(Y_i - \mu)\right\} \\
&= \prod_{i=1}^n (2\pi)^{-p/2} \tau_i^{p/2} |\Psi|^{-1/2} \exp\left\{-\frac{\tau_i}{2}(Y_i - \mu)^T (\Psi)^{-1}(Y_i - \mu)\right\} \\
&= (2\pi)^{-np/2} \prod_{i=1}^n \left(\tau_i^{p/2}\right) |\Psi|^{-n/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^n \tau_i (Y_i - \mu)^T (\Psi)^{-1}(Y_i - \mu)\right\}
\end{aligned}$$

and, noting that $\tau_i \sim \frac{\chi^2_\nu}{\nu} \sim \Gamma\left(k = \nu/2, \theta = 2/\nu\right)$

$$p(\vec{\tau}) = \prod_{i=1}^{n} \frac{\tau_i^{\nu/2-1} e^{-\nu\tau_i/2}}{\left(\frac{2}{\nu}\right)^{\nu/2} \Gamma(\frac{\nu}{2})} \mathbf{1}_{\{\tau_i \geq 0\}} = \frac{\prod_{i=1}^{n} \tau_i^{\nu/2-1} e^{-\frac{1}{2}\sum_{i=1}^{n} \nu\tau_i}}{\left(\frac{2}{\nu}\right)^{n\nu/2} \Gamma(\frac{\nu}{2})^n} \mathbf{1}_{\{\prod_{i=1}^{n} \tau_i \geq 0\}}$$

Taking the log of the product we get:

$$\begin{aligned}
&\log\left(p(\vec{Y}|\vec{\tau}, \theta) \cdot p(\vec{\tau})\right) \\
&= \log\left((2\pi)^{-np/2} \prod_{i=1}^{n} \left(\tau_i^{p/2}\right) |\Psi|^{-n/2} \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \tau_i \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \right\} \cdot \frac{\prod_{i=1}^{n} \tau_i^{\nu/2-1} e^{-\frac{1}{2}\sum_{i=1}^{n} \nu\tau_i}}{\left(\frac{2}{\nu}\right)^{n\nu/2} \Gamma(\frac{\nu}{2})^n} \mathbf{1}_{\{\prod_{i=1}^{n} \tau_i \geq 0\}} \right) \\
&= \frac{p}{2} \sum_{i=1}^{n} \log(\tau_i) - \frac{n}{2} \log|\Psi| - \frac{1}{2} \sum_{i=1}^{n} \tau_i \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) + \left(\frac{\nu}{2} - 1\right) \sum_{i=1}^{n} \log(\tau_i) - \frac{1}{2} \sum_{i=1}^{n} \nu\tau_i + constants \\
&= \left(\frac{p}{2} + \frac{\nu}{2} - 1\right) \sum_{i=1}^{n} \log(\tau_i) - \frac{n}{2} \log|\Psi| - \frac{1}{2} \sum_{i=1}^{n} \tau_i \left[\left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) + \nu \right] + constants
\end{aligned}$$

where the constants we have omitted are not functions of θ, Y_i, or τ_i. Now, we are able to form our Q function for the E step of the algorithm. At this point we now drop all terms not containing θ, and multiply by 2, as this will not affect the maximization.

$$\begin{aligned}
Q\left(\theta|\theta^{(t)}\right) &= \mathbb{E}\left[\log\left(p(\vec{Y}|\vec{\tau}, \theta) \cdot p(\vec{\tau})\right) \middle| Y_i, \theta^{(t)} \right] \\
&= \mathbb{E}\left[-\log|\Psi| - \sum_{i=1}^{n} \tau_i \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \middle| Y_i, \theta^{(t)} \right] \\
&= n \log|\Psi^{-1}| - \mathbb{E}\left[\sum_{i=1}^{n} \tau_i \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \middle| Y_i, \theta^{(t)} \right] \\
&= n \log|\Psi^{-1}| - \sum_{i=1}^{n} \mathbb{E}\left[\tau_i \middle| Y_i, \theta^{(t)} \right] \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \\
&= n \log|\Psi^{-1}| - \sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right)
\end{aligned}$$

Here, the second term can be written as:

$$\begin{aligned}
\sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) &= \sum_{i=1}^{n} \tau_i^{(t+1)} tr\left[\left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \right] \\
&= \sum_{i=1}^{n} \tau_i^{(t+1)} tr\left[\left(Y_i - \mu\right) \left(Y_i - \mu\right)^T \left(\Psi\right)^{-1} \right] \\
&= \sum_{i=1}^{n} \tau_i^{(t+1)} tr\left[\left(\Psi\right)^{-1} \left(Y_i - \mu\right) \left(Y_i - \mu\right)^T \right] \\
&= tr\left[\sum_{i=1}^{n} \tau_i^{(t+1)} \left(\Psi\right)^{-1} \left(Y_i - \mu\right) \left(Y_i - \mu\right)^T \right] \\
&= tr\left[\left(\Psi\right)^{-1} \sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu\right) \left(Y_i - \mu\right)^T \right]
\end{aligned}$$

Thus,

$$Q\left(\theta|\theta^{(t)}\right) = n \log|\Psi^{-1}| - tr\left[\left(\Psi\right)^{-1} \sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu\right) \left(Y_i - \mu\right)^T \right]$$

Following the hint given that

$$\underset{A}{argmax}\{n \log|A^{-1}| - tr\left(A^{-1}B\right)\} = \frac{1}{n} B$$

we conclude that

$$\Psi^{(t+1)} = \underset{\Psi}{argmax}\ Q\left(\Psi, \mu^{(t)} \middle| \Psi^{(t)}, \mu^{(t)}\right) = \frac{1}{n}\sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu^{(t)}\right)\left(Y_i - \mu^{(t)}\right)^T.$$

To maximize over our other parameter, μ, we investigate the Q function found above, however now we can drop all terms not containing μ.

$$\begin{aligned}
Q\left(\theta|\theta^{(t)}\right) &= n\log|\Psi^{-1}| - \sum_{i=1}^{n} \tau_i^{(t+1)} (Y_i - \mu)^T (\Psi)^{-1} (Y_i - \mu) \\
&= -\sum_{i=1}^{n} \tau_i^{(t+1)} (Y_i - \mu)^T (\Psi)^{-1} (Y_i - \mu) \\
&= -\sum_{i=1}^{n} (Y_i - \mu)^T \left(\frac{\Psi}{\tau_i^{(t+1)}}\right)^{-1} (Y_i - \mu)
\end{aligned}$$

Appealing to the fact that for a symmetric matrix W, $\frac{d}{ds}\sum(y-s)^T W(y-s) = \sum 2W(y-s)$, we have

$$\frac{d}{d\mu} Q\left(\theta|\theta^{(t)}\right) = \sum_{i=1}^{n} 2\cdot\left(\frac{\Psi}{\tau_i^{(t+1)}}\right)^{-1} (Y_i - \mu)$$

Setting equal to 0, we find

$$\begin{aligned}
&\sum_{i=1}^{n} 2\cdot\left(\frac{\Psi}{\tau_i^{(t+1)}}\right)^{-1} (Y_i - \mu) = 0 \\
\implies &\sum_{i=1}^{n} \tau_i^{(t+1)} Y_i - \sum_{i=1}^{n} \tau_i^{(t+1)} \mu = 0 \\
\implies &\sum_{i=1}^{n} \tau_i^{(t+1)} Y_i = \sum_{i=1}^{n} \tau_i^{(t+1)} \mu \\
\implies &\mu^{(t+1)} = \underset{\mu}{argmax}\ Q\left(\Psi^{(t+1)}, \mu|\Psi^{(t)}, \mu^{(t)}\right) = \frac{\sum_{i=1}^n \tau_i^{(t+1)} Y_i}{\sum_{i=1}^n \tau_i^{(t+1)}}
\end{aligned}$$

To summarize, we conclude that

$$\Psi^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \tau_i^{(t+1)} \left(Y_i - \mu^{(t)}\right)\left(Y_i - \mu^{(t)}\right)^T.$$

$$\mu^{(t+1)} = \frac{\sum_{i=1}^n \tau_i^{(t+1)} Y_i}{\sum_{i=1}^n \tau_i^{(t+1)}}$$

□

---

3. In this question you will implement two different EM algorithms for a Hierarchical Poisson model:

$$Y_i|\lambda_i \sim Pois(\lambda_i), \qquad \lambda_i|\beta \sim Gamma(1, \beta), \tag{6}$$

where the pdf of a $Gamma(1, b)$ random variable is defined to be

$$p(x) = I_{\{x>0\}} b\exp\{-bx\}, \tag{7}$$

i.e. $\lambda_i \sim Expo(\beta)$.

a. Derive the EM algorithm for β in the model (6).
Answer: We begin by writing down the likelihood function for $Y_i|\lambda_i \sim Pois(\lambda_i)$ and $\lambda_i|\beta \sim Gamma(1, \beta)$.

$$p(\vec{Y}|\vec{\lambda}) = \prod_{i=1}^{n} \frac{\lambda_i^{Y_i} e^{-\lambda_i}}{Y_i!} = \frac{\prod_{i=1}^n \lambda_i^{Y_i} e^{-\sum_{i=1}^n \lambda_i}}{\prod_{i=1}^n Y_i!}$$

$$p(\vec{\lambda}|\beta) = \prod_{i=1}^{n} \beta\exp\{-\beta\lambda_i\} = \beta^n \exp\left\{-\beta\sum_{i=1}^{n}\lambda_i\right\}$$

Taking the log of the product we get the complete-data log-likelihood:

$$\log(p(\vec{Y},\vec{\lambda}|\beta)) = \log(p(\vec{Y}|\vec{\lambda})\cdot p(\vec{\lambda}|\beta)) = \log\left(\frac{\prod_{i=1}^n \lambda_i^{Y_i} e^{-\sum_{i=1}^n \lambda_i}}{\prod_{i=1}^n Y_i!}\cdot \beta^n \exp\left\{-\beta\sum_{i=1}^n \lambda_i\right\}\right) \tag{8}$$

$$= \sum_{i=1}^n Y_i\log(\lambda_i) - \sum_{i=1}^n \log(Y_i!) - \sum_{i=1}^n \lambda_i + n\log(\beta) - \beta\sum_{i=1}^n \lambda_i \tag{9}$$

$$\tag{10}$$

$$\begin{aligned}
Q\left(\theta|\theta^{(t)}\right) &= \mathbb{E}\left[\log\left(p(\vec{Y}|\vec{\lambda})\cdot p(\vec{\lambda}|\beta)\right)\middle|Y_i,\beta^{(t)}\right]\\
&= \mathbb{E}\left[\sum_{i=1}^n Y_i\log(\lambda_i) - \sum_{i=1}^n \log(Y_i!) - \sum_{i=1}^n \lambda_i + n\log(\beta) - \beta\sum_{i=1}^n \lambda_i\middle|Y_i,\beta^{(t)}\right]\\
&= n\log(\beta) - \beta\sum_{i=1}^n \mathbb{E}\left[\lambda_i\middle|Y_i,\beta^{(t)}\right]\\
&= n\log(\beta) - \beta\sum_{i=1}^n \frac{Y_i+1}{\beta^{(t)}+1}\\
&= n\log(\beta) - \frac{\beta}{\beta^{(t)}+1}(n\bar{Y}+n)
\end{aligned}$$

Where we have dropped all terms not involving $\theta = \beta$.
So,

$$\frac{dQ}{d\theta} = \frac{n}{\beta} - \frac{n\bar{Y}+n}{\beta^{(t)}+1}$$

and setting equal to zero, we arrive at

$$\beta^{(t+1)} = \frac{\beta^{(t)}+1}{\bar{Y}+1}$$

□

b. Construct an Ancillary Augmentation (AA) that preserves the observed data log-likelihood of the Hierarchical model in 6.
Answer: Suppose our model is:

$$\begin{aligned}
Y_i|\lambda_i,\beta &\sim Pois(\lambda_i/\beta)\\
\lambda_i &\sim \Gamma(1,1)
\end{aligned}$$

The complete data density for a single observation is then:

$$p(Y_i,\lambda_i|\beta) = p(Y_i|\lambda_i,\beta)\cdot p(\lambda_i) = \frac{\left(\frac{\lambda_i}{\beta}\right)^{Y_i} e^{-\lambda_i/\beta}}{Y_i!}\cdot e^{-\lambda_i}$$

and it follows that the observed data density for a single observation is:

$$\begin{aligned}
p(Y_i|\beta) &= \int_0^\infty p(Y_i,\lambda_i|\beta)\,d\lambda_i\\
&= \int_0^\infty \frac{\left(\frac{\lambda_i}{\beta}\right)^{Y_i} e^{-\lambda_i/\beta}}{Y_i!} e^{-\lambda_i}\,d\lambda_i\\
&= \frac{1}{Y_i!\beta^{Y_i}}\int_0^\infty \lambda_i^{(Y_i+1)-1} e^{-\lambda_i(1+1/\beta)}\,d\lambda_i\\
&= \frac{\Gamma(Y_i+1)}{(1+1/\beta)^{Y_i+1}}\frac{1}{Y_i!\beta^{Y_i}}\int_0^\infty \frac{(1+1/\beta)^{Y_i+1}}{\Gamma(Y_i+1)}\lambda_i^{(Y_i+1)-1} e^{-\lambda_i(1+1/\beta)}\,d\lambda_i\\
&= \frac{\beta^{Y_i+1}}{(\beta+1)^{Y_i+1}}\frac{1}{\beta^{Y_i}}\\
&= \frac{\beta}{(\beta+1)^{Y_i+1}}
\end{aligned}$$

which matches what is found below in part (d), and confirms that the observed data log-likelihood has been preserved.
Taking our AA to be

$$Y_i|\lambda_i, \beta \sim Pois(\tilde{\lambda}_i/\beta)$$
$$\tilde{\lambda}_i \sim \Gamma(1,1),$$

we proceed in constructing the EM algorithm for β in this model.
The complete-data log-likelihood is

$$\begin{aligned}
\log(p(\vec{Y}, \vec{\tilde{\lambda}}|\beta)) = \log(p(\vec{Y}|\vec{\tilde{\lambda}}, \beta) \cdot p(\vec{\tilde{\lambda}})) &= \log\left(\prod_{i=1}^n \frac{\left(\frac{\tilde{\lambda}_i}{\beta}\right)^{Y_i} e^{-\tilde{\lambda}_i/\beta}}{Y_i!} e^{-\tilde{\lambda}_i}\right) \\
&= \log\left(\prod_{i=1}^n \frac{\tilde{\lambda}_i^{Y_i}}{\beta^{Y_i} Y_i!} e^{-\tilde{\lambda}_i(1+1/\beta)}\right) \\
&= \log\left(\frac{\prod_{i=1}^n \tilde{\lambda}_i^{Y_i}}{\beta^{\sum_{i=1}^n Y_i} \prod_{i=1}^n Y_i!} e^{-(1+1/\beta)\sum_{i=1}^n \tilde{\lambda}_i}\right) \\
&= \sum_{i=1}^n Y_i \log\tilde{\lambda}_i - \sum_{i=1}^n Y_i \log\beta - \sum_{i=1}^n \log Y_i! - (1+1/\beta)\sum_{i=1}^n \tilde{\lambda}_i
\end{aligned}$$

Dropping all terms not involving β, we can write our Q function as

$$Q(\beta|\beta^{(t)}) = \mathbb{E}\left[-\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\sum_{i=1}^n \tilde{\lambda}_i | Y_i, \beta^{(t)}\right]$$

In order to compute this expectation, we note that $p(\tilde{\lambda}|Y_i, \beta) \propto p(Y_i, \tilde{\lambda}_i|\beta) \approx \lambda_i^{(Y_i+1)-1} e^{-\lambda_i(1+1/\beta)}$, which we recognize as a $\Gamma(Y_i + 1, 1/\beta + 1)$ distribution. Thus, $\mathbb{E}(\tilde{\lambda}|Y_i, \beta^{(t)}) = \frac{Y_i+1}{1/\beta^{(t)}+1}$.
Thus,

$$\begin{aligned}
Q(\beta|\beta^{(t)}) &= \mathbb{E}\left[-\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\sum_{i=1}^n \tilde{\lambda}_i | Y_i, \beta^{(t)}\right] \\
&= -\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\sum_{i=1}^n \mathbb{E}\left[\tilde{\lambda}_i | Y_i, \beta^{(t)}\right] \\
&= -\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\sum_{i=1}^n \frac{Y_i + 1}{1/\beta^{(t)} + 1} \\
&= -\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\alpha
\end{aligned}$$

where $\alpha = \sum_{i=1}^n \frac{Y_i+1}{1/\beta^{(t)}+1} = \frac{\beta^{(t)}}{1+\beta^{(t)}} \sum_{i=1}^n Y_i + 1 = \frac{n\beta^{(t)}}{1+\beta^{(t)}}(\bar{Y} + 1)$ is not a function of β. To maximize the Q function, we take the derivative with respect to β and set equal to zero.

$$\begin{aligned}
\frac{d}{d\beta}\left(-\sum_{i=1}^n Y_i \log\beta - \frac{1}{\beta}\alpha\right) &= -\frac{\sum_{i=1}^n Y_i}{\beta} + \frac{\alpha}{\beta^2} = 0 \\
&\implies \beta^2 \sum_{i=1}^n Y_i - \beta\alpha = 0 \\
&\implies \beta(\beta \sum_{i=1}^n Y_i - \alpha) = 0 \\
&\implies \beta = 0 \quad \text{or} \quad \beta = \frac{\alpha}{\sum_{i=1}^n Y_i}
\end{aligned}$$

The choice of $\beta = 0$ is not allowed, and so we have

$$\beta^{(t+1)} = \frac{\frac{n\beta^{(t)}}{1+\beta^{(t)}}(\bar{Y} + 1)}{n\bar{Y}} = \frac{\beta^{(t)}}{1+\beta^{(t)}}\left(1 + \frac{1}{\bar{Y}}\right)$$

□

c. Using the SA and AA from (a) and (b), derive the corresponding Interwoven EM algorithm. You may select either the SA or AA as A1.

Answer: For the IEM, we will follow the following 5 steps, where we have chosen SA to be A1.

i. Update β from the AA=A2 scheme, label it $\beta^{(t+0.5)}$.

$$\beta^{(t+0.5)} = \frac{\beta^{(t)}}{1+\beta^{(t)}}\left(1+\frac{1}{\bar{Y}}\right)$$

ii. Write down the Q function from the SA = A1 scheme

$$Q(\beta|\beta^{(t+0.5)}) = \mathbb{E}_{AA}\left[\mathbb{E}_{SA}\left[n\log(\beta) - \beta\sum_{i=1}^{n}\lambda_i\middle|Y_i, \tilde{\lambda}_i^{(t+0.5)}, \beta^{(t+0.5)}\right]\middle|Y_i, \beta^{(t)}\right]$$

iii. Replace $\lambda_i = H(\tilde{\lambda_i}, \beta^{(t+0.5)}) = \frac{\tilde{\lambda}_i}{\beta^{(t+0.5)}}$

$$\begin{aligned}
Q(\beta|\beta^{(t+0.5)}) &= \mathbb{E}_{AA}\left[\mathbb{E}_{SA}\left[n\log(\beta) - \beta\sum_{i=1}^{n}\frac{\tilde{\lambda}_i}{\beta^{(t+0.5)}}\middle|Y_i, \tilde{\lambda}_i^{(t+0.5)}, \beta^{(t+0.5)}\right]\middle|Y_i, \beta^{(t)}\right]\\
&= \mathbb{E}_{AA}\left[n\log(\beta) - \beta\sum_{i=1}^{n}\frac{\tilde{\lambda}_i}{\beta^{(t+0.5)}}\middle|Y_i, \beta^{(t)}\right]
\end{aligned}$$

iv. Compute expectation:

$$\begin{aligned}
Q(\beta|\beta^{(t+0.5)}) &= \mathbb{E}_{AA}\left[n\log(\beta) - \beta\sum_{i=1}^{n}\frac{\tilde{\lambda}_i}{\beta^{(t+0.5)}}\middle|Y_i, \beta^{(t)}\right]\\
&= n\log(\beta) - \frac{\beta}{\beta^{(t+0.5)}}\sum_{i=1}^{n}\mathbb{E}_{AA}\left[\tilde{\lambda}_i\middle|Y_i, \beta^{(t)}\right]\\
&= n\log(\beta) - \frac{\beta}{\beta^{(t+0.5)}}\sum_{i=1}^{n}\frac{\beta^{(t)}(Y_i+1)}{\beta^{(t)}+1}
\end{aligned}$$

v. Maximize, by taking the derivative of β and setting equal to zero.

$$\begin{aligned}
\frac{d}{d\beta}Q(\beta|\beta^{(t+0.5)}) &= \frac{d}{d\beta}\left(n\log(\beta) - \frac{\beta}{\beta^{(t+0.5)}}\sum_{i=1}^{n}\frac{\beta^{(t)}(Y_i+1)}{\beta^{(t)}+1}\right)\\
&= \frac{n}{\beta} - \frac{1}{\beta^{(t+0.5)}}\sum_{i=1}^{n}\frac{\beta^{(t)}(Y_i+1)}{\beta^{(t)}+1} = 0\\
&\implies \frac{n}{\beta} = \frac{n\beta^{(t)}}{\beta^{(t+0.5)}(\beta^{(t)}+1)}(\bar{Y}+1)\\
&\implies \beta = \frac{\beta^{(t+0.5)}(\beta^{(t)}+1)}{\beta^{(t)}(\bar{Y}+1)}\\
&\implies \beta = \frac{\frac{\beta^{(t)}}{1+\beta^{(t)}}\left(1+\frac{1}{\bar{Y}}\right)(\beta^{(t)}+1)}{\beta^{(t)}(\bar{Y}+1)}\\
&\implies \beta = \frac{\left(1+\frac{1}{\bar{Y}}\right)}{(\bar{Y}+1)}\\
&\implies \beta = \frac{1}{\bar{Y}}
\end{aligned}$$

and we see that we have converged to the MLE in one step (using the result from part (d), below).

$\square$

d. Derive the observed data log-likelihood according to model 6 and compute the MLE for β.

Answer: The complete data density for the SA model is given by

$$p(Y_i, \lambda_i|\beta) = p(Y_i|\lambda_i)\cdot p(\lambda_i|\beta) = \frac{\lambda_i^{Y_i}e^{-\lambda_i}}{Y_i!}\cdot\beta e^{-\beta\lambda_i} = \frac{\beta}{Y_i!}\lambda_i^{Y_i}e^{-\lambda_i(\beta+1)}$$

The observed-data density is then

$$
\begin{aligned}
p(Y_i|\beta) = \int_0^\infty p(Y_i, \lambda_i|\beta)\, d\lambda_i &= \int_0^\infty \frac{\beta}{Y_i!} \lambda_i^{Y_i} e^{-\lambda_i(\beta+1)}\, d\lambda_i \\
&= \frac{\beta}{Y_i!} \int_0^\infty \lambda_i^{(Y_i+1)-1} e^{-\lambda_i(\beta+1)}\, d\lambda_i \\
&= \frac{\Gamma(Y_i+1)}{(\beta+1)^{Y_i+1}} \frac{\beta}{Y_i!} \int_0^\infty \frac{(\beta+1)^{Y_i+1}}{\Gamma(Y_i+1)} \lambda_i^{(Y_i+1)-1} e^{-\lambda_i(\beta+1)}\, d\lambda_i \\
&= \frac{\beta}{(\beta+1)^{Y_i+1}}
\end{aligned}
$$

Computing the observed data log-likelihood we have:

$$
\begin{aligned}
l(\beta|Y_i) &= \log\left(\prod_{i=1}^n \frac{\beta}{(\beta+1)^{Y_i+1}}\right) \\
&= \log\left(\frac{\beta^n}{(\beta+1)^{n\bar{Y}+n}}\right) \\
&= n\log(\beta) - (n\bar{Y}+n)\log(\beta+1)
\end{aligned}
$$

Then, taking the derivative and setting equal to zero:

$$
\begin{aligned}
\frac{dl(\beta|Y_i)}{d\beta} &= \frac{n}{\beta} - \frac{n\bar{Y}+n}{\beta+1} = 0 \\
&\implies \beta + 1 = \beta(\bar{Y}+1) \\
&\implies \beta_{MLE} = \frac{1}{\bar{Y}}
\end{aligned}
$$

□

e. Compare the linear rate of convergence of each of the EM algorithms, and discuss when each algorithm will perform well (or not).
Answer: As we have already noted, the IEM algorithm converges to the MLE in only a single step. Overall this would appear to be the preferred algorithm of choice.
For the individual EM algorithms, SA and AA, we derive their convergence rates and compare. Note: we are denoting $\beta^* = \beta_{MLE} = 1/\bar{Y}$.

SA:

$$
\left.\frac{d}{d\beta^{(t)}}\beta^{(t+1)}\right|_{\beta^{(t)}=\beta^*} = \left.\frac{d}{d\beta^{(t)}}\frac{\beta^{(t)}+1}{\bar{Y}+1}\right|_{\beta^{(t)}=\beta^*} = \frac{1}{\bar{Y}+1}
$$

Here we see that the SA scheme will have a slow rate of convergence ~ 1 when $\bar{Y}$ is small, and will increase (become faster) as $\bar{Y}$ increases. Thus, when we have a data set with a relatively large mean, then the SA will perform well.

AA:

$$
\begin{aligned}
\left.\frac{d}{d\beta^{(t)}}\beta^{(t+1)}\right|_{\beta^{(t)}=\beta^*} &= \left.\frac{d}{d\beta^{(t)}}\frac{\beta^{(t)}}{1+\beta^{(t)}}\left(1+\frac{1}{\bar{Y}}\right)\right|_{\beta^{(t)}=\beta^*} \\
&= \left.\frac{d}{d\beta^{(t)}}\frac{\beta^{(t)}}{1+\beta^{(t)}}\left(1+\frac{1}{\bar{Y}}\right)\right|_{\beta^{(t)}=\beta^*} \\
&= \left.\left(1+\frac{1}{\bar{Y}}\right)\left(\frac{1}{1+\beta^{(t)}} - \frac{\beta^{(t)}}{(1+\beta^{(t)})^2}\right)\right|_{\beta^{(t)}=\beta^*} \\
&= \left.\left(1+\frac{1}{\bar{Y}}\right)\left(\frac{1}{(1+\beta^{(t)})^2}\right)\right|_{\beta^{(t)}=\beta^*} \\
&= \left(1+\frac{1}{\bar{Y}}\right)\left(\frac{1}{(1+1/\bar{Y})^2}\right) \\
&= \frac{\bar{Y}}{1+\bar{Y}}
\end{aligned}
$$

For the AA scheme, we have the opposite convergence performance compared to the SA. When $\bar{Y}$ is very large, $\frac{\bar{Y}}{1+\bar{Y}}$ will be close to 1, indicating a slow convergence rate. Hence, the AA would be the method of choice when using a data set with a relatively small mean.

□

```
### STA 250 - HW 3
### Eliot Paisley
### 11/27/13
##################

setwd("C:/Users/EliotP/Google Drive/sta 250 - f13/") #laptop
library(xtable)

### bisection method.
### arguments are:
### function to find roots of, initial lower and upper bounds, tolarance, and max. number of iterations

bisect= function(func, lower, upper, tol=1e-07, niter = 1000, ...) {
        l = lower
        u = upper
        func_low = func(lower, ...)
        func_high = func(upper, ...)
        f_eval = 2

        if (func_low * func_high > 0) stop
        change = upper - lower

        while (abs(change) > tol) {
          x_new = (lower + upper) / 2
          f_new = func(x_new, ...)
          if (abs(f_new) <= tol) break
          if (func_low * f_new < 0) upper = x_new
          if (func_high * f_new < 0) lower = x_new
          change = upper - lower
          f_eval = f_eval + 1
        } # end while loop
        list(x = x_new, value = f_new, fevals=f_eval, l =l,u = u )
} # end bisect function

### Newton Raphson method/
### arguments are:
### first derivative, second derivative, initial guess, tolerance, and max. number of iterations

newton = function(D, DD, initial, tol=1e-6, niter=20){
          int = initial # inital guess to report at end
          theta = initial # initial theta
          out = matrix(NA, nrow=niter+1,ncol=4) #create matrix for the output
          val = D(initial) # goal is to arrive at val = 0
          out[1,] = c(initial,val,niter,int)

          i = 1
          continue = T
          while (continue) {
            i = i+1
            theta.old = theta
            theta = theta - D(theta)/DD(theta) ## NR update

            val = D(theta)
            out[i,] = c(theta,val,i,int)
            continue = (abs(theta-theta.old) > tol) &&
              (i <= niter)
          }
          if (i > niter) {
            warning("Maximum number of iterations reached")
            return(out)
          }
          out = out[!is.na(out[,1]),]
          out
} # end NR function


### classic linkage problem
### approximate likelihood

like = function(x){ (2+x)^125*(1-x)^38*x^34}
# take log to simplify
# max will still occur at same point
```

```
log_like = function(x){ 125*log(2+x) + 38*log(1-x) + 34*log(x)}
# take first derivative
log_like_D = function(x){ 125/(2+x)- 38/(1-x) + 34/x}
# take second derivative
log_like_DD = function(x){ -125/(2+x)^2- 38/(1-x)^2 - 34/x^2}

# max visually found somewhere near 0.6, local max around -0.5

png("like.png")
par(mfrow=c(2,1))
plot(like,xlim=c(0,1),ylab="Likelihood",xlab=expression(paste(lambda)),main="Classic Linkage Problem")
plot(like,xlim=c(-1,0),ylab="Likelihood",xlab=expression(paste(lambda)),main="Classic Linkage Problem")
dev.off()

# roots visually identified somewhere in (-1, 0) and (0,1)
png("loglikeD.png")
plot(log_like_D, xlim = c(-3,3),ylab="Log-Likelihood",xlab=expression(paste(lambda)),
main="Classic Linkage Problem (first derivative)")
abline(h=0,col="RED")
dev.off()


# set very small offset to avoid infinities
epsilon = 0.0001

###### search for roots using bisection
#######################################

# first interval, results in root at x = -0.5506794
one = bisect(log_like_D, -2+epsilon,0-epsilon)
# first interval, results in root at x = 0.6268215
two = bisect(log_like_D, 0 +epsilon, 1-epsilon)

png("bisect.png")
plot(like,xlim=c(-1,1),ylab="Likelihood",xlab=expression(paste(lambda)),
main="Classic Linkage Problem (w/ Bisection Roots)")
points(x = one$x, y = like(one$x), lwd =5, col="RED", pch =4 )
points(x = two$x, y = like(two$x), lwd =5, col="RED", pch =4 )
dev.off()
t = t(as.table(unlist(one),digits=3))
tt = t(as.table(unlist(two),digits=3))
colnames(t) = c("Root", "Value","Iterations","Int. Low", "Int. Upp")
colnames(tt) = c("Root", "Value","Iterations","Int. Low", "Int. Upp")
print(xtable(rbind(t,tt)),include.rownames=FALSE)

##### search for roots using Newton-Raphson
###########################################

newt1 = newton(log_like_D, log_like_DD,initial= -0.55,niter=500,tol=1e-7)
newt2 = newton(log_like_D, log_like_DD,initial=  0.40,niter=500,tol=1e-7)
newt3 = newton(log_like_D, log_like_DD,initial=  0.60,niter=500,tol=1e-7)

colnames(newt1) = c("Root", "Value", "Iteration", "Initial")
ttt = rbind(t(as.table(newt1[nrow(newt1),])),t(as.table(newt2[nrow(newt2),])),t(as.table(newt3[nrow(newt3),])))
print(xtable(ttt),include.rownames=FALSE)

png("newton.png")
plot(like,xlim=c(-1,1),ylab="Likelihood",xlab=expression(paste(lambda)),main="Classic Linkage Problem (w/ NR Root)")
points(x = newt3[nrow(newt3),1], y = like(newt3[nrow(newt3),1]), lwd =5, col="RED", pch =4 )
dev.off()
```