Solutions to Stat 701 HW2 due 2/23/15
Problem 4 Information Matrix has been corrected.

(1). (a) For $\alpha$ restricted to be 3.0, the density is $(b^3/2)x^2e^{-bx}$ and the MLE is $3n/\bar{X} = 120\cdot3/282.0035$

> 120*3/282.0035
[1] 1.27658

#### compare to 1.2 = correct answer.

(b) For $\beta$ restricted = 1.1, the density is $1.1^a/\Gamma(a))x^{a-1}\exp(-1.1x^a)$, Fisher Information is $\text{trigamma}(a)$, and Newton-Raphson algorithm gives:

> a0 = 1.5
> logLik(gdat, a0, 1.1)
[1] -238.8549

> tmplik = avec = numeric(10)
for(i in 1:10) {
    a0 = a0 + (log(1.1)-digamma(a0))/trigamma(a0)
    avec[i] = a0
    tmplik[i] = logLik(gdat,a0,1.1)
}

rbind(avec = avec[1:5], llk =tmplik[1:5])

avec 1.562923 1.564712 1.564714 1.564714 1.564714

## Convergence in 3 iterations !! Note that the log-likelihoods are
## fairly poor compared to the maximized loglik of -196.2 because
## 1.1 is far away from the MLE 1.17 for beta.

(2). Put $J_k = \int z^k (f_0'(z))^2/f_0(z) \, dz$ for $k = 0,1,2$. Then direct differentiation, followed by the change of variables $z = (x-a)/b$ and the identities

$\int f_0'(z) \, dz = \int z f_0'(z) \, dx = 0,$

and $\int z f_0'(z) \, dz = -1$, shows that the Fisher information matrix is given by

$I(a,b) = \frac{1}{b^2} \begin{pmatrix} J_0 & J_1 \\ J_1 & J_2 - 1 \end{pmatrix}$

Note that this matrix is positive-definite because of the Cauchy-Schwarz inequality applied to the integral of $(f_0'(z)/f_0(z))(z-1)$ against $f_0(z) \, dz$. 1
(3). (a). Note first that, as in Problem 2, the Fisher Information matrix for a location-scale parameter family does not depend at all on the location parameter value, and depends on the scale parameter only through the proportionality constant $1/\sigma^2$, that is,

$$I(\mu, \sigma) = \begin{pmatrix} J_0 & J_1 \\ J_1 & J_2 - 1 \end{pmatrix} = \begin{pmatrix} 1/3 & 0 \\ 0 & 1.4300 \end{pmatrix}$$

where we have substituted the values $J_k$ found by numerical integration in R. (The values $J_0, J_1$ are also easy to integrate in closed form.)

(b). Computational comparison in R of NR versus Fisher scoring.

```r
xdat = seq(.06,.63, by=.03)
DlgLik = function(th, xdat) {## gradient of log-likelihood
  xst = (xdat-th[1])/th[2]
  aux = 1-2*plogis(xst)
  (-1/th[2])*c(sum(aux), sum(1+xst*aux)) }
D2lgLik = function(th, xdat) {## Hessian of log-likelihood
  xst = (xdat-th[1])/th[2]
  aux = -1+2*plogis(xst)
  ax2 = 2*dlogis(xst)
  (-1/th[2]^2)*array(c(sum(ax2), rep(sum(aux + xst*ax2),2),
    sum(-1+2*xst*aux + xst^2*ax2)), c(2,2)) }

> rbind(100*(DlgLik(c(.26,.3),xdat)-DlgLik(c(.25,.3),xdat)),
    D2lgLik(c(.25,.3),xdat)
   [,1]
[1,] -100.9080 -56.42681 ## difference quotient wrt mu
[2,] -100.9098 -56.42866 ## Check of deriv wrt mu

> rbind(100*(DlgLik(c(.25,.31),xdat)-DlgLik(c(.25,.3),xdat)),
    D2lgLik(c(.25,.305),xdat)
   [,2]
[1,] -56.92238 94.66893 ## difference quotient wrt sigma
[2,] -56.89821 94.70706 ## Check of deriv wrt sigma

> tmp = rlogis(1000) # simulated logistic(.,0,1) data, n=1000
> -D2lgLik(c(0,1),tmp)/1000
   [,1]      [,2]
[1,] 0.33703251 0.009981602
[2,] 0.009981602 1.370897541 ### Check Information vs D2lgLik
```

2
## now find out what the actual ML will be.
> optim(c(.3,-1), function(x)
>  -sum(log(dlogis(xdat,x[1],exp(x[2])))))$par
[1] 0.3450245 -2.2519633
> c(.345, exp(-2.252))
[1] 0.3450000 0.1051886  ## this is the MLE

## Ten iterations of Newton-Raphson:
th=c(.3,.14); thmat = array(0, c(2,10))
for(i in 1:10) {
    th = th - solve(D2lgLik(th,xdat), DlgLik(th,xdat))
    thmat[,i] = th }
> round(thmat,5)[,1:7]
[1,] 0.38625 0.35838 0.34849 0.34553 0.34502 0.34500 0.34500
[2,] 0.05610 0.07344 0.09028 0.10138 0.10491 0.10519 0.10519
### converged in 6

## Ten iterations of Fisher scoring:
th=c(.3,.14); thmat = array(0, c(2,10))
for(i in 1:10) {
    th = th + (th[2]^2/20)*c(1/.333333, 1/1.430)*DlgLik(th,xdat)
    thmat[,i] = th }
> round(thmat,5)[,1:7]
[1,] 0.38625 0.35838 0.34849 0.34553 0.34502 0.34500 0.34500
[2,] 0.05610 0.07344 0.09028 0.10138 0.10491 0.10519 0.10519
### converged a little quicker than NR

th=c(.3,.16); thmat = array(0, c(2,10))
for(i in 1:10) {
    th = th - solve(D2lgLik(th,xdat), DlgLik(th,xdat))
    thmat[,i] = th }  ## NR badly non-convergent

th=c(.2,.2); thmat = array(0, c(2,10))
for(i in 1:10) {
    th = th + (th[2]^2/20)*c(1/.333333, 1/1.430)*DlgLik(th,xdat)
    thmat[,i] = th }  # scoring works fine, in 6 iterations
As indicated in class, the log-likelihood in this problem is
\[ \sum_{i=1}^{n} \left\{ D_i \log((1 - T_i/2) \lambda T_i^{\alpha-1} e^{-\lambda T_i^\alpha}) + (1 - D_i) \log((1/2) e^{-\lambda T_i^\alpha}) \right\} \]
which immediately implies that for fixed and known \( \alpha = \alpha_0 \), the MLE for \( \lambda \) is
\[ \hat{\lambda} = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{n} T_i^{\alpha_0}} \]
Next, we calculate the method-of-moments estimator of \( \lambda \) based on \( E(D_i) \) by equating (for \( \alpha = \alpha_0 \))
\[ n^{-1} \sum_{i=1}^{n} D_i = P(X < C) = 1 - \frac{1}{2} \int_{0}^{2} e^{-\lambda x^\alpha} dx \equiv g(\lambda) \quad (\circ) \]
Solving the last equation uniquely for \( \lambda = \hat{\lambda} \) is evidently possible by the Intermediate Value Theorem but not in closed form, since the right-hand side is monotone increasing in \( \lambda \) and ranges from 0 to 1 as \( \lambda \) ranges from 0 to \( \infty \).

The Delta Method immediately implies, with \( g'(\lambda) = 0.5 \int_{0}^{2} x^{\alpha_0} e^{-\lambda x^\alpha} dx \), that
\[ \sqrt{n} (\hat{\lambda} - \lambda) \xrightarrow{D} \mathcal{N}(0, \frac{g(\lambda)(1 - g(\lambda))}{(g'(\lambda))^2}) \]
The asymptotic variance of the MLE is obtained as the reciprocal of the Fisher Information matrix (with \( \alpha_0 \) substituted for \( \alpha \)), where the requested \( 2 \times 2 \) Fisher Information matrix is given by
\[ I(\lambda, \alpha) = E \left( \begin{array}{c} D/\lambda^2 \\ \{\log(T)\} T^\alpha \\ D/\alpha^2 + \lambda \{\log(T)\}^2 T^\alpha \end{array} \right) \]
So actually, the Fisher Information is finite as long as \( \alpha > 0 \), and as long as this is true, we can write \( E(D) = g(\lambda) \), and for \( j = 1, 2 \),
\[ E(\{\log(T)\}^j T^\alpha) = \lambda \alpha \int_{0}^{2} (\log(t))^j t^{2\alpha-1} e^{-\lambda t^\alpha} (1 - t/2) dt \]
In the setting with fixed \( \alpha = \alpha_0 \), the ARE for \( \hat{\lambda} \) with respect to \( \lambda \), is \( g^2(\lambda) (1 - g(\lambda))/\{\lambda g'(\lambda)\}^2 \). Since \( g(0) = 0 \) and \( g \) is increasing and concave, it follows that \( \lambda g'(\lambda) > g(\lambda) \) for all \( \lambda > 0 \), and that this ARE is < 1.
As indicated in #4, MLE is
\[
\text{sum(Dvec)/sum(Tvec^2)} \quad \#\# = 0.7239591
\]
\[
a0=2; \ lam = 1 \quad \# \text{arbitrary starting value}
\]
\[
\text{aux = sum(Tvec^2); aux2 = sum(1-Dvec)}
\]
\[
\text{for(i in 1:20) \{lam = 60/(aux + aux2/lam )
\}
\]
\[
0.831009 0.770264 0.744855 0.733563 0.728409 0.726029
0.724924 0.724409 0.724169 0.724057 0.724005 0.72398
0.723969 0.723964 0.723961 0.72396 0.72396 0.723959
0.723959 0.723959
\]

Let’s start with the brute-force method, based on log-likelihood
\[
\sum_{i=1}^{n} \log(a \cdot I_{X_i=0}) + (1-a) \cdot \exp(-\lambda) \cdot \lambda^{X_i}/\Gamma(X_i + 1))
\]

With the dataset \(n = 80\) in \(Wvec\), this leads to:
\[
dist = \text{table(Wvec)}
\]
\[
\begin{align*}
0 & \quad 31 & 1 & 2 & 3 & 4 & 5 \\
1 & \quad & 7 & 1 & 1
\end{align*}
\]
\[
\text{logLik} = \text{function}(x,d) \quad d[1]*\log(x[1]+(1-x[1])*\exp(-x[2])) + \\
(80-d[1])*\log(1-x[1])-x[2]) + \\
\text{sum((1:5)*d[2:6])*\log(x[2])}
\]
\[
> \text{logLik(c(.2,1.4),dist) \quad \#\# = -78.87767}
\]
\[
\text{optim(c(.5,1), function(x,.d) \text{-logLik(x,.d), .d=dist,}
\text{method="L-BFGS-B", lower=c(0.01,0.01))}
\]
\[
$\text{par}$

\[
\begin{align*}
\text{[1]} & \quad 0.1528073 \quad 1.2836542 \quad \text{### ahat and lamhat}
\end{align*}
\]
\[
$\text{value}$

\[
\begin{align*}
\text{[1]} & \quad 78.68902 \quad \text{### converged in 10 steps}
\end{align*}
\]

Now we do the EM calculation, based on ‘complete data’ consisting of mixture-component \(\epsilon_i\) plus \(X_i\): here \(\epsilon_i \sim \text{Binom}(1,1-a)\) and \(X_i = \text{...
0 whenever \( \epsilon_i = 0 \). Then for the EM iterations, the complete-data log-likelihood is (apart from terms not involving parameters)

\[
\sum_{i=1}^{n} X_i \log \lambda + n (\log(1 - a) - \lambda) + \sum_{i=1}^{n} (1 - \epsilon_i) (\lambda + \log(a/(1 - a)))
\]

and \( \epsilon_i = 1 \) whenever \( X_i > 0 \), while

\[
E_{\theta_0}(1 - \epsilon_i \mid X_i = 0) = \frac{a_0}{a_0 + (1 - a_0) e^{-\lambda_0}}
\]

So the EM iteration consists of finding \( \theta_1 = (a, \lambda) \) to maximize

\[
\sum_{i=1}^{n} X_i \log \lambda + n (\log(1 - a) - \lambda) + \frac{M_0 a_0 (\lambda + \log(a/(1 - a)))}{a_0 + (1 - a_0) e^{-\lambda_0}}
\]

where \( M_0 = \sum_{i=1}^{n} I[X_i = 0] \), which implies

\[
\hat{a} = \frac{a_0 M_0/n}{a_0 + (1 - a_0) e^{-\lambda_0}}, \quad \hat{\lambda} = \bar{X}/(1 - \hat{a})
\]

### EM algorithm iteration:

```r
th = c(.5,1); Wbar = mean(Wvec); M0 = 31/80 ## = prop obs'd 0's
llkEM = numeric(100)
thmat = array(0,c(2,100))
for(i in 1:100) {
  thmat[,i]=th
  llkEM[i] = logLik(th,dist) }
> round(th,5)
[1] 0.15281 1.28365
```

Loglik converged up to 4 decimal places in 30 iterations, and \( \theta \) converged to 4 decimal place accuracy in about 50 iterations.