Stat 8003: Homework 5

Group B: El Moustaid, Fadoua and Kandadai, Venkatesh

October 2, 2014

Solution to Problem 1

We have that $x_i \sim f(x_i) = \pi_0 f_0(x_i) + \pi_1 f_1(x_i)$ where $f_0(x_i) = \mathbb{1}(0 \le x_i \le 1)$ with a uniform density and $f_1(x_i) = \beta(1-x_i)^{\beta-1} = Beta(1,\beta)$. The complete data is represented by $y_i = (x_i, z_i)$ and $\theta = (\pi_0, \beta)$, the vector of parameters.

(a) Derive the completely likelihood function $L_n(\theta|x_i, z_i)$

$$L_n(\theta|x_i,z_i) = \prod_{i=1}^{n}\sum_{j=0}^{1}\pi_j f_j(x_i)\mathbb{1}(z_i = j)$$

the completely log-likelihood function can be represented by,

$$l_n(\theta|x_i,z_i) = \sum_{i=1}^{n}\sum_{j=0}^{1}\mathbb{1}(z_i = j)log(\pi_j f_j(x_i))$$

(b) Using the EM Algorithm to derive the estimators for π_0 and β

<u>Expectation Step</u>:

$$Q(\theta,\theta^t) = E[l_n(\theta|x_i,z_i)] = E[\sum_{i=1}^{n}\sum_{j=0}^{1}\mathbb{1}(z_i = j)log(\pi_j f_j(x_i))] = \sum_{i=1}^{n}\sum_{j=0}^{1}P(z_i = j|x_i,\theta)log(\pi_j f_j(x_i))$$

where: $P(z_i = j|x_i,\theta) = \dfrac{P(x_i|z_i = j)P(z_i = j)}{\sum_{j=0}^{1}P(x_i|z_i = j)P(z_i = j)} = \dfrac{\pi_j f_j(x_i)}{\pi_0 f_0(x_i) + \pi_1 f_1(x_i)} = T_{ji}^t$

therefore:

$$Q(\theta,\theta^t) = \sum_{i=1}^{n}\sum_{j=0}^{1}T_{ji}^t log(\pi_j f_j(x_i))$$

Maximization Step:

(Maximize π_0):

if $j = 0$: $\sum_{i=1}^{n} T_{0i}^{t} log(\pi_0 f_0(x_i)$

if $j = 1$: $\sum_{i=1}^{n} T_{1i}^{t} log((1-\pi_0) f_1(x_i))$

therefore:

$$\Delta = \sum_{i=1}^{n} T_{0i}^{t} log(\pi_0 f_0(x_i) + \sum_{i=1}^{n} T_{1i}^{t} log((1-\pi_0) f_1(x_i))$$

$$\frac{\partial \Delta}{\partial \pi_0}(\sum_{i=1}^{n} T_{0i}^{t} log(\pi_0 f_0(x_i) + \sum_{i=1}^{n} T_{1i}^{t} log((1-\pi_0) f_1(x_i))) = 0$$

$$= (\sum_{i=1}^{n} T_{0i}^{t}) \frac{f_0(xi)}{\pi_0 f_0(xi)} + (\sum_{i=1}^{n} T_{1i}^{t}) \frac{-f_1(x_i)}{f_1(x_i) \pi_0 f_1(x_i)} = 0$$

$$(\sum_{i=1}^{n} T_{0i}^{t}) \frac{f_0(xi)}{\pi_0 f_0(xi)} = (\sum_{i=1}^{n} T_{1i}^{t}) \frac{f_1(x_i)}{(1-\pi_0) f_1(x_i)}$$

$$(\sum_{i=1}^{n} T_{0i}^{t}) \frac{1}{\pi_0} = (\sum_{i=1}^{n} T_{1i}^{t}) \frac{1}{1-\pi_0}$$

Thus,

$$\boxed{\pi_0^{t+1} = \frac{1}{n} \sum_{i=1}^{n} T_{0i}^{t}}$$

(Maximize β):

Given:

$$Q(\theta, \theta^t) = \sum_{i=1}^{n} T_{0i}^{t} log(\pi_0 f_0(x_i) + \sum_{i=1}^{n} T_{1i}^{t} log(\pi_1 \beta (1-x_i)^{\beta-1})$$

Let

$$\Delta = \sum_{i=1}^{n} T_{1i}^{t} (log(\pi_1) + log(\beta) + (\beta-1) log(1-x_1))$$

$$\frac{\partial \Delta}{\partial \beta} = \sum_{i=1}^{n} T_{1i}^{t}(\frac{1}{\beta} + log(1 - x_i)) = 0$$

$$\sum_{i=1}^{n} T_{1i}^{t}\frac{1}{\beta} = -\sum_{i=1}^{n} T_{1i}^{t} log(1 - x_i)$$

$$\boxed{\beta^{t+1} = -\frac{\sum_{i=1}^{n} T_{1i}^{t}}{\sum_{i=1}^{n} T_{1i}^{t} log(1 - x_i)}}$$

(c) Our R-program to estimate π_0^{t+1} and β^{t+1} using EM algorithm has been attached to this homework assignment. Our computed estimates are:

$$\pi_0^{t+1} = 0.696794$$

$$\beta^{t+1} = 11.093275$$

Using these estimates, we fit our beta-uniform mixture model to the pvalues dataset:

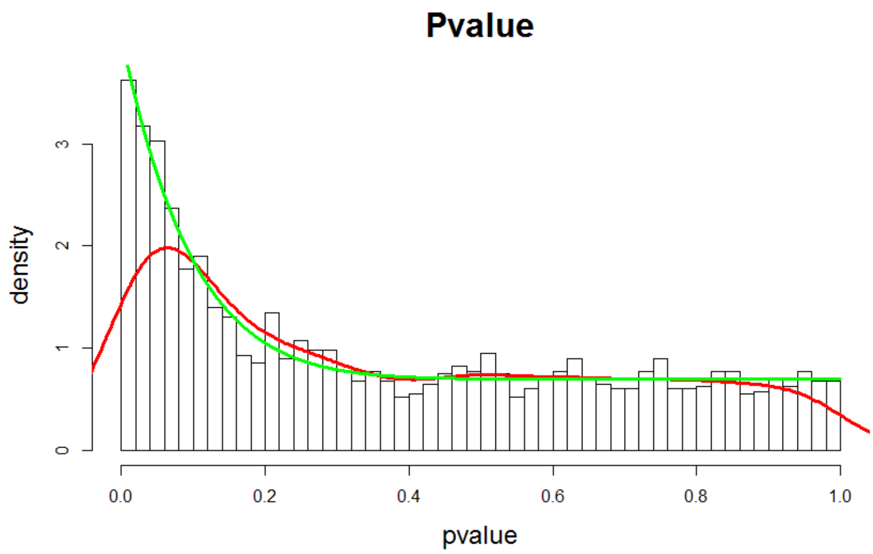

Figure 1: Beta-uniform mixture fit to pvalues; generated in R. Note: the green curve represents the fitted density

Given:

$$fdr_i(x_i) = P(Z_i = 0|x_i) = \frac{\pi_0 f_0(x_i)}{\pi_0 f_0(x_i) + \pi_1 f_1(x_i)} = \frac{\pi_0 f_0(x_i)}{\pi_0 f_0(x_i) + (1 - \pi_0)\beta(1 - x_i)^{\beta - 1}}$$

we can substitute our computed estimates, π_0^{t+1} and β^{t+1} for π_0 and β and compute the local fdr score for each x_i using the following R-script:

```
###X is a vector of pvalues
fdrlocal<-(pi0*dunif(X,0,1)) / (pi0*dunif(X,0,1)+pi1*dbeta(X,1,beta))
```

(d) Our R-script attached to this homework yields **321** $(117 + 204)$ falsely classified data points when $fdr_i(x_i) > 0.5$

	$group_{EM}$	
	0	1
$group$		
0	1182	204
1	117	497

Solution to Problem 2

Given the local fdr score as:

$$fdr_i = \frac{\pi_0 f_0(x_i)}{f(x_i)}$$

where $f(x_i)$ is the marginal density of x_i and assuming $\pi_0 = 0.7$

(a) Given: $\hat{f}_h(X) = \frac{1}{nh}\sum_{i=1}^{n} k(\frac{x-x_i}{h})$ and $k(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}$:

results in:

$$\hat{f}_h(X) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\phi(\frac{x-x_i}{h})$$

using Silverman's h:

$$h = 1.06\hat{\sigma}n^{\frac{-1}{5}}$$

Below is a density plot generated in R using the Gaussian-Kernel method with Silverman's h to estimate $f(x_i)$ from the pvalue dataset:

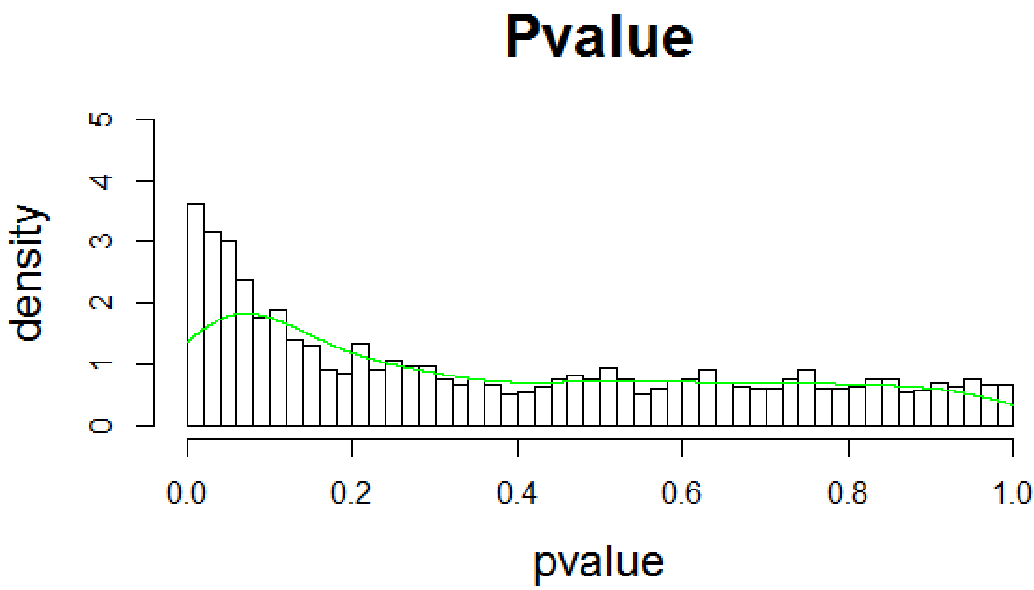

Figure 2: Gaussian-Kernel with Silverman's h, fit to pvalues generated in R. Note: the green curve represents the fitted density

(b) We can estimate the local fdr score using the Gaussian-Kernel with Silverman's h method with the following R-script:

```
#estimate local fdr score using Gaussian-Kernel method
#X is a vector of pvalues
n <- length(X)
h <- 1.06 * sqrt( var(X) ) / (n^(1/5)) # Silverman's h
fnorm.hat.h<-X
for( i in 1:length( X ) )
{
  fnorm.hat.h[i] <-  mean(dnorm( X[i]-X, 0, h ))
}

fdr_local_kernel<-(0.7*dunif(pvalue$X,0,1)) / fnorm.hat.h #yields a 2000x1 vector of local fdr scores

plot(X,fnorm.hat.h) #plots pvalue against Gaussian-Kernel density estimate
```

(c) Our R-script attached to this homework yields **335** $(117 + 218)$ falsely classified data points when $fdr_i(x_i) > 0.5$

	$group_{Kernel}$	
	0	1
$group$		
0	1168	218
1	117	497

(d) Using maximum likelihood cross-validation method:

$$\hat{f}_h(X) = \frac{1}{nh}\sum_{i=1}^{n} k(\frac{x - x_i}{h})$$

$$\hat{f}_h(x_j) = \frac{1}{nh}\sum_{i=1}^{n} k(\frac{x_j - x_i}{h})$$

Likelihood function:

$$\prod_{j=1}^{n} \hat{f}_h(x_j) = \frac{1}{nh}\prod_{j=1}^{n}\sum_{i=1}^{n} k(\frac{x_j - x_i}{h})$$

using the "leave-one-out" method:

$$\hat{f}_{h,i}(x_j) = \frac{1}{(n-1)h}\sum_{i \neq j} k(\frac{x_j - x_i}{h})$$

therefore:

$$MLCV = \frac{1}{n}\sum_{i} log(\sum_{i \neq j} k(\frac{x_j - x_i}{h})\frac{1}{(n-1)h})$$

The following R-script using the "kedd" package will generate a density curve using the MLCV method:

```
#X is a vector of pvalues
library(kedd)
h.cv <- h.mlcv(X)$h
xaxis_new <- seq( min(X), max(X), 0.00001 )
fnorm.cv.hat <- xaxis_new

for( i in 1:length( xaxis_new ) )
{
fnorm.cv.hat[i] <- mean( dnorm( xaxis_new[i]-X, 0, h.cv ))
}

#plot density
hist( X, freq=F, br=40,  main="Pvalue", xlab="pvalue", ylab="density", cex.n
points( xaxis_new, fnorm.cv.hat, main=paste("h=",h.cv, sep=""), xlab="pvalue
```

Figure 3 depicts densities of 3 methods fit to pvalues: 1) EM-algorithm 2) Gaussian-Kernel with Silverman's h 3) Maximum likelihood cross-validation

Our R-script attached to this homework yields **501** $(334 + 167)$ falsely classified data points when compared to the original group classification, and when $fdr_i(x_i) > 0.5$

	$group_{MLCV}$	
	0	1
$group$		
0	1219	167
1	334	280

(e) The EM-algorithm worked the best in terms of having the lowest classification error.

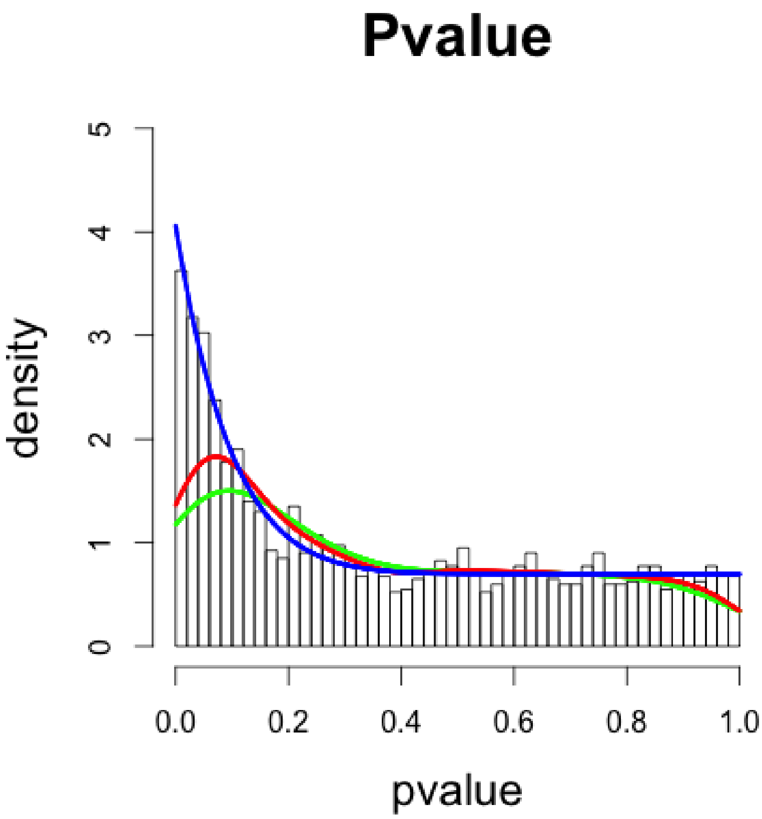

Figure 3: Note: Blue = EM-algorithm, Red= Gaussian-Kernel w Silverman's h, Green=MLCV