

MATH 598 - PROJECT 1 : SOLUTIONS

(a) The likelihood for a sample $y_1, \dots, y_n$ takes the form

$$\mathcal{L}_n(\theta) = \prod_{i=1}^n \theta \exp\{-\theta y_i\} = \theta^n \exp\left\{-\theta \sum_{i=1}^n y_i\right\} = \theta^n \exp\{-\theta s_n\}$$

say.

(i) If $\pi_0(\theta) \equiv \text{Gamma}(\alpha, \beta)$ for $\alpha, \beta > 0$, then for the posterior we have

$$\begin{aligned} \pi_n(\theta) \propto \mathcal{L}_n(\theta) \pi_0(\theta) &= \theta^n \exp\{-\theta s_n\} \theta^{\alpha-1} \exp\{-\theta \beta\} \quad \theta > 0 \\ &= \theta^{n+\alpha-1} \exp\{-\theta(s_n + \beta)\} \\ &\equiv \text{Gamma}(n + \alpha, s_n + \beta) \end{aligned}$$

(ii) If $\pi_0(\theta) \equiv \text{Uniform}(0, a)$ for $a > 0$, then we have

$$\begin{aligned} \pi_n(\theta) \propto \mathcal{L}_n(\theta) \pi_0(\theta) &= \theta^n \exp\{-\theta s_n\} \mathbb{1}_{(0,a)}(\theta) \quad \theta > 0 \\ &\propto \frac{s_n^{n+1}}{\Gamma(n+1)} \theta^{(n+1)-1} \exp\{-\theta s_n\} \mathbb{1}_{(0,a)}(\theta). \end{aligned}$$

This is the $\text{Gamma}(n+1, s_n)$ distribution, truncated to the interval $(0, a)$, and we have that the unspecified normalizing constant is the integral

$$\int_0^a \frac{s_n^{n+1}}{\Gamma(n+1)} \theta^{(n+1)-1} \exp\{-\theta s_n\} d\theta \equiv P(X \leq a)$$

if $X \sim \text{Gamma}(n+1, s_n)$. This integral can be computed using the `pgamma` function in R.

(iii) If $\pi_0(\theta)$ takes the mixture form

$$\pi_0(\theta) = \frac{1}{2} \exp\{-\theta\} + \exp\{-2\theta\} \quad \theta > 0$$

then we have from first principles that

$$\begin{aligned} \pi_n(\theta) \propto \mathcal{L}_n(\theta) \pi_0(\theta) &= \theta^n \exp\{-\theta s_n\} \frac{1}{2} (\exp\{-\theta\} + 2 \exp\{-2\theta\}) \quad \theta > 0 \\ &\propto \left(\theta^{(n+1)-1} \exp\{-\theta(s_n + 1)\} + 2\theta^{(n+1)-1} \exp\{-\theta(s_n + 2)\} \right). \end{aligned}$$

The normalizing constant is therefore

$$\int_0^\infty \left(\theta^{(n+1)-1} \exp\{-\theta(s_n + 1)\} + 2\theta^{(n+1)-1} \exp\{-\theta(s_n + 2)\} \right) d\theta$$

which we can deduce is equal to

$$\frac{\Gamma(n+1)}{(s_n + 1)^{n+1}} + 2 \frac{\Gamma(n+1)}{(s_n + 2)^{n+1}}$$

as each term in the integrand is proportional to a Gamma pdf. Thus we have a mixture posterior

$$\pi_n(\theta) = w_n \pi_n^{(1)}(\theta) + (1 - w_n) \pi_n^{(2)}(\theta)$$

where $\pi_n^{(1)}(\theta) \equiv \text{Gamma}(n+1, s_n + 1)$ and $\pi_n^{(2)}(\theta) \equiv \text{Gamma}(n+1, s_n + 2)$ and

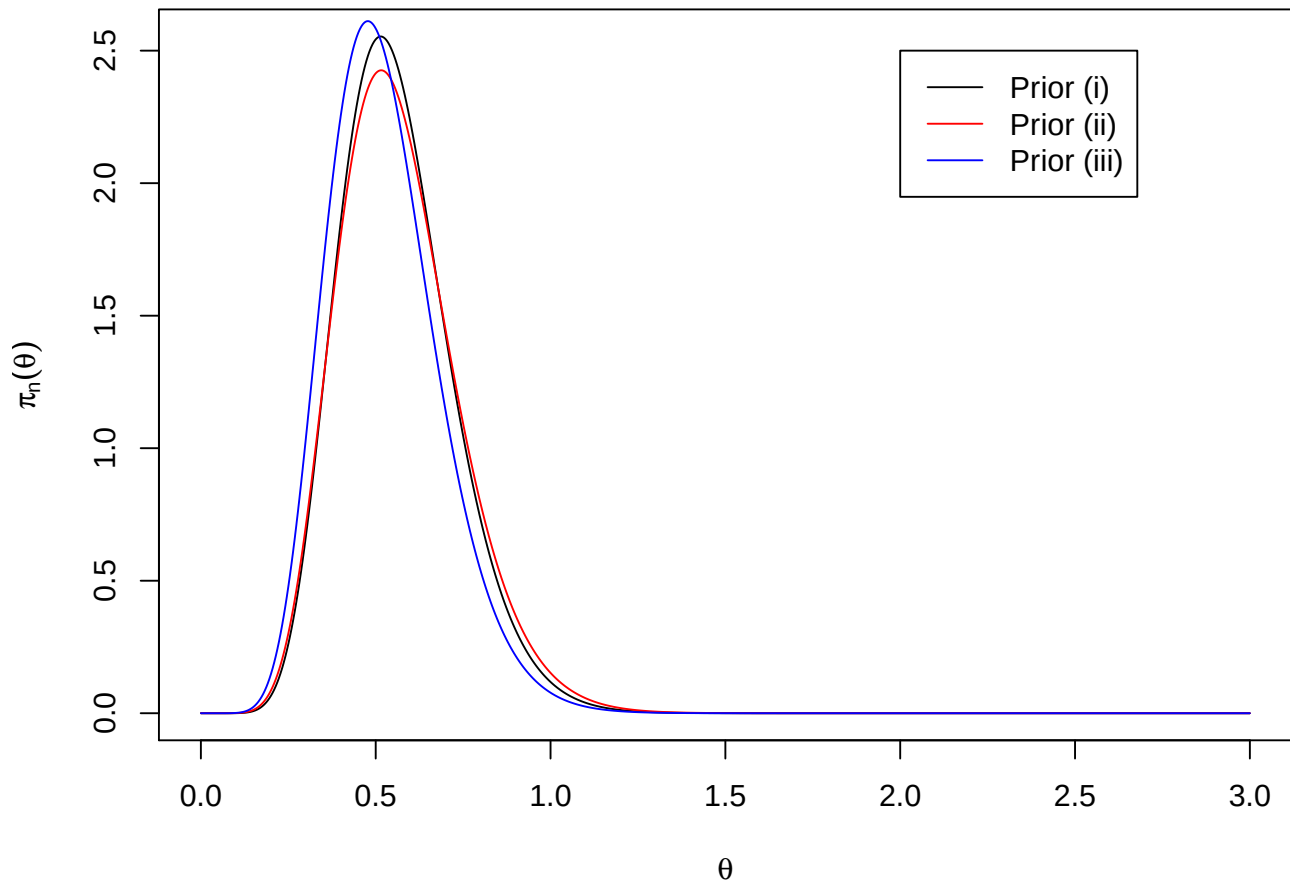
$$w_n = \frac{\frac{\Gamma(n+1)}{(s_n + 1)^{n+1}}}{\frac{\Gamma(n+1)}{(s_n + 1)^{n+1}} + 2 \frac{\Gamma(n+1)}{(s_n + 2)^{n+1}}}$$

For illustration, we choose $\alpha = \beta = 2$, and $a = 2$ in the following calculations

```

n<-10
y<-c(3.33,2.04,1.96,4.72,1.13,0.12,4.06,0.13,1.79,0.11)
s.n<-sum(y)
al<-be<-2
a<-2
xv<-seq(0,3,by=0.001)
yv1<-dgamma(xv,n+al,s.n+be)
par(mar=c(4,4,1,0))
plot(xv,yv1,type='l',xlab=expression(theta),ylab=expression(pi[n](theta)))
yv2<-(xv<a)*dgamma(xv,n+1,s.n)/pgamma(a,n+1,s.n)
lines(xv,yv2,col='red')
const<-gamma(n+1)/(s.n+1)^(n+1) + 2*gamma(n+1)/(s.n+2)^(n+1)
w.n<-gamma(n+1)/(s.n+1)^(n+1)/const
#yv3<-(exp(n*log(xv)-(s.n+1)*xv)+2*exp(n*log(xv)-(s.n+2)*xv))/const
yv3<-w.n*dgamma(xv,n+1,s.n+1)+(1-w.n)*dgamma(xv,n+1,s.n+2)
lines(xv,yv3,col='blue')
legend(2.0,2.5,c('Prior (i)', 'Prior (ii)', 'Prior (iii)'),col=c('black','red','blue'),lty=1)

```



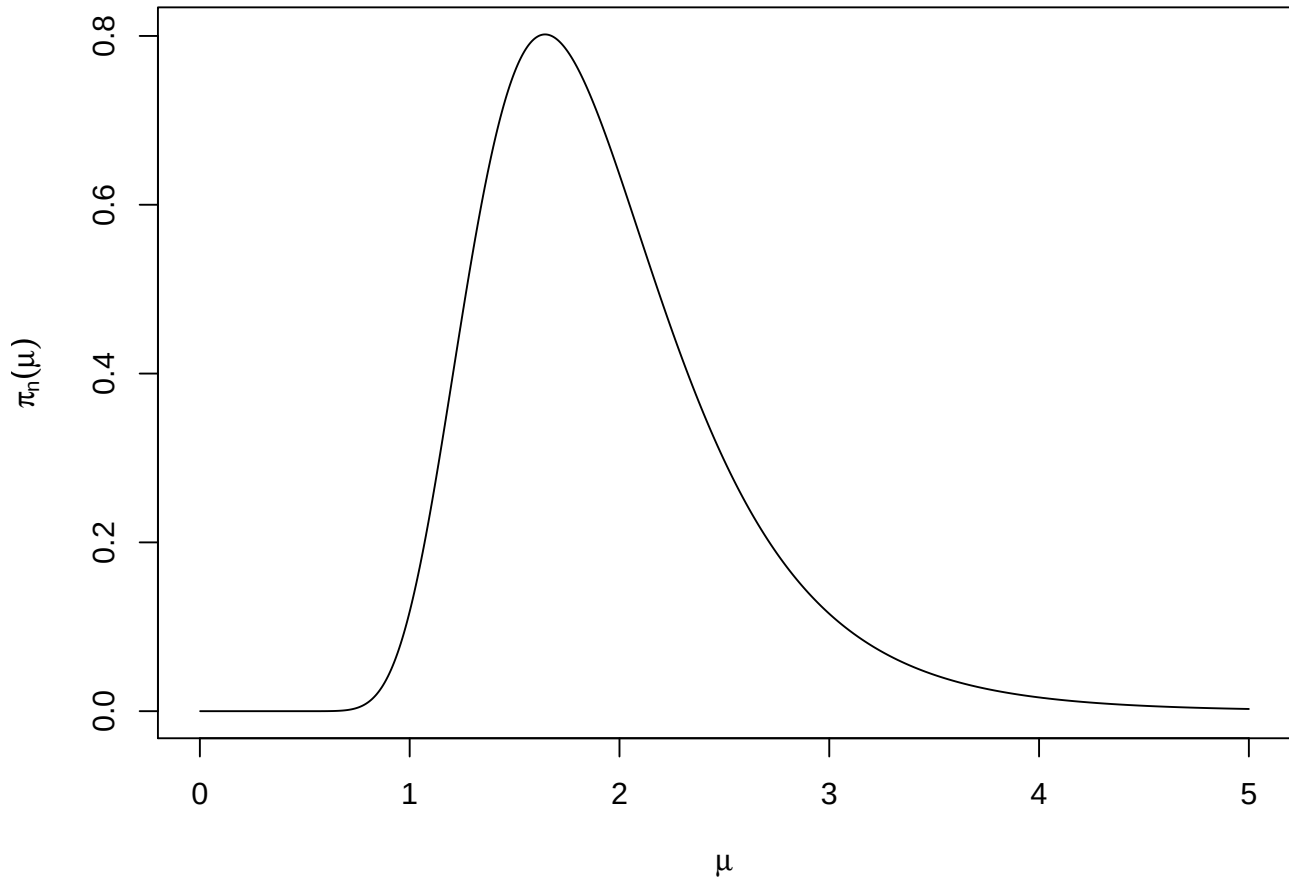
15 MARKS

(b) If $\mu = 1/\theta$, then the Jacobian of the transformation is $1/\mu^2$, so we have that

$$\pi_n(\mu) = \frac{(s_n + \beta)^{n+\alpha}}{\Gamma(n + \alpha)} \left(\frac{1}{\mu}\right)^{(n+\alpha)+1} \exp\left\{-\frac{(s_n + \beta)}{\mu}\right\} \quad \mu > 0$$

which is the so-called Inverse Gamma distribution $InvGamma(n + \alpha, s_n + \beta)$.

```
dinvgamma<-function(x,a,b){return(exp(a*log(b)-lgamma(a)-(a+1)*log(x)-b/x))}
xv<-seq(0,5,by=0.001)
yv4<-dinvgamma(xv,n+al,s.n+be)
par(mar=c(4,4,1,0))
plot(xv,yv4,type='l',xlab=expression(mu),ylab=expression(pi[n](mu)))
```



5 MARKS

- (c) The joint predictive distribution is computed as per the de Finetti theorem: for $y_{11}, y_{12} > 0$, if $\alpha_n = n + \alpha$ and $\beta_n = s_n + \beta$, we have

$$\begin{aligned} p_n(y_{11}, y_{12}) &= \int_0^\infty f_Y(y_{11}; \theta) f_Y(y_{12}; \theta) \pi_n(\theta) d\theta = \int_0^\infty \theta^2 \exp\{-\theta(y_{11} + y_{12})\} \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \theta^{\alpha_n-1} \exp\{-\beta_n \theta\} d\theta \\ &= \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \int_0^\infty \theta^{(\alpha_n+2)-1} \exp\{-(\beta_n + y_{11} + y_{12})\theta\} d\theta \\ &= \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \frac{\Gamma(\alpha_n + 2)}{(\beta_n + y_{11} + y_{12})^{\alpha_n+2}} \end{aligned}$$

This formula shows that the joint pdf depends on the arguments only via the sum $y_{11} + y_{12}$.

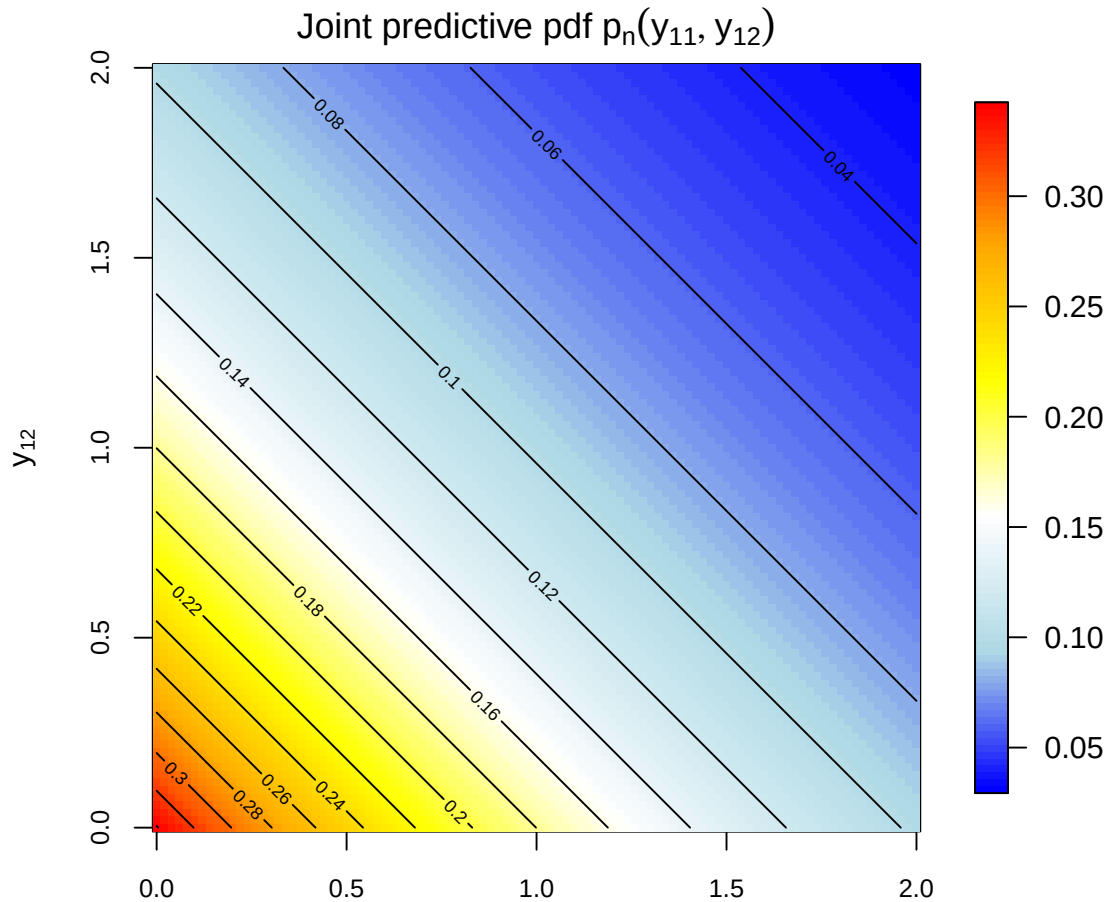
```
joint.pred<-function(y1,y2,a,b){
  jpdf<-a*log(b)-lgamma(a)+lgamma(a+2)-(a+2)*log(b+y1+y2)
  return(exp(jpdf))
}
f <- Vectorize(joint.pred,vectorize.args=c("y1","y2"))
y1v<-seq(0,2,length=101)
```

```

y2v<-seq(0,2,length=101)
jmat<-outer(y1v,y2v,f,a=n+al,b=s.n+be)

#Plot
library(fields,quietly=TRUE)
par(pty='s',mar=c(2,3,2,2))
colfunc <- colorRampPalette(c("blue","lightblue","white","yellow","orange","red"))
image.plot(y1v,y2v,jmat,col=colfunc(100),
           xlab=expression(y[11]),ylab=expression(y[12]),cex.axis=0.8)
contour(y1v,y2v,jmat,add=T,levels=seq(0,1,by=0.02))
title(expression(paste('Joint predictive pdf ',p[n](y[11],y[12]))))

```



4 MARKS

(d) There are a number of different ways to define a Bayesian credible interval (θ_L, θ_U) ; if $\Pi_n(\theta)$ represents the posterior cdf, then we could define

(I) a $\gamma/2$ **equi-tailed interval**: here we compute the two values θ_L and θ_U such that

$$\Pi_n(\theta_L) = \frac{\gamma}{2} \quad \Pi_n(\theta_U) = 1 - \frac{\gamma}{2}.$$

(II) the **shortest interval**: here we compute the two values θ_L and θ_U such that

$$(\theta_L, \theta_U) = \arg \min_{t_1, t_2} (t_2 - t_1)$$

subject to the constraint

$$\Pi_n(\theta_U) - \Pi_n(\theta_L) = 1 - \gamma.$$

(III) any interval (θ_L, θ_U) such that

$$\Pi_n(\theta_U) - \Pi_n(\theta_L) = 1 - \gamma.$$

In all three cases, if the cdf is available, then the interval can be computed using grid-search. In part (a), we have that $\Pi_n(\theta)$ can be computed using the `pgamma` function in R. The range of possible credible intervals can be constructed using the `qgamma` function. For the Gamma posterior in (i):

```
gam<-0.05
a.n<-n+al; b.n<-s.n+be
p.L<-0; p.U<-gam

#Equi-tailed
theta.L.1<-qgamma(gam/2,a.n,b.n)
theta.U.1<-qgamma(1-gam/2,a.n,b.n)
c(theta.L.1,theta.U.1,theta.U.1-theta.L.1)

+ [1] 0.2898820 0.9201514 0.6302694

#Shortest & Any
pv<-seq(p.L,p.U,length=21)
ci.mat<-matrix(0,nrow=length(pv),ncol=5)
for(i in 1:length(pv)){
  p1<-pv[i]
  p2<-p1+(1-gam)
  th.L<-qgamma(p1,a.n,b.n)
  th.U<-qgamma(p2,a.n,b.n)
  ci.mat[i,1]<-p1
  ci.mat[i,2]<-p2
  ci.mat[i,3]<-th.L
  ci.mat[i,4]<-th.U
  ci.mat[i,5]<-th.U-th.L
}
colnames(ci.mat)<-c('Lower.p','Upper.p','Lower','Upper','Length')
print(ci.mat)

+      Lower.p Upper.p      Lower      Upper      Length
+ [1,]  0.0000  0.9500 0.0000000 0.8512162 0.8512162
+ [2,]  0.0025  0.9525 0.2114114 0.8565039 0.6450925
+ [3,]  0.0050  0.9550 0.2310948 0.8620419 0.6309472
+ [4,]  0.0075  0.9575 0.2439485 0.8678581 0.6239095
+ [5,]  0.0100  0.9600 0.2537719 0.8739848 0.6202129
+ [6,]  0.0125  0.9625 0.2618424 0.8804610 0.6186187
+ [7,]  0.0150  0.9650 0.2687569 0.8873334 0.6185765
+ [8,]  0.0175  0.9675 0.2748462 0.8946587 0.6198125
+ [9,]  0.0200  0.9700 0.2803137 0.9025073 0.6221935
+ [10,] 0.0225  0.9725 0.2852943 0.9109672 0.6256729
+ [11,] 0.0250  0.9750 0.2898820 0.9201514 0.6302694
+ [12,] 0.0275  0.9775 0.2941454 0.9302080 0.6360625
+ [13,] 0.0300  0.9800 0.2981363 0.9413362 0.6431999
+ [14,] 0.0325  0.9825 0.3018944 0.9538139 0.6519195
+ [15,] 0.0350  0.9850 0.3054513 0.9680455 0.6625942
+ [16,] 0.0375  0.9875 0.3088322 0.9846532 0.6758210
+ [17,] 0.0400  0.9900 0.3120578 1.0046709 0.6926131
+ [18,] 0.0425  0.9925 0.3151453 1.0300176 0.7148723
+ [19,] 0.0450  0.9950 0.3181092 1.0649489 0.7468397
+ [20,] 0.0475  0.9975 0.3209614 1.1228071 0.8018456
+ [21,] 0.0500  1.0000 0.3237126      Inf      Inf

#Shortest interval
print(ci.mat[which.min(ci.mat[,5]),])

+      Lower.p Upper.p      Lower      Upper      Length
+ 0.0150000 0.9650000 0.2687569 0.8873334 0.6185765
```

For the Truncated Gamma posterior in (ii): we have that if $f(y)$ represents the non-truncated density, and $f(y; a)$ represents the truncated density, then

$$F(y; a) = \frac{F(y)}{F(a)} \quad 0 < y < a$$

so therefore for any target probability p , the corresponding quantile q for the truncated distribution is

$$q = F^{-1}(F(a)p)$$

where $F(a)$ can be computed using `pgamma` and $F^{-1}(\cdot)$ can be computed by `qgamma`.

```
gam<-0.05;p.L<-0; p.U<-gam
const<-pgamma(a,n+1,s.n)
#Equi-tailed
theta.L.1<-qgamma(const*gam/2,n+1,s.n);theta.U.1<-qgamma(const*(1-gam/2),n+1,s.n)
c(theta.L.1,theta.U.1,theta.U.1-theta.L.1)

+ [1] 0.2831955 0.9484452 0.6652497

#Shortest & Any
pv<-seq(p.L,p.U,length=21)
ci.mat<-matrix(0,nrow=length(pv),ncol=5);
colnames(ci.mat)<-c('Lower.p','Upper.p','Lower','Upper','Length')
for(i in 1:length(pv)){
  p1<-pv[i];p2<-p1+(1-gam)
  th.L<-qgamma(const*p1,n+1,s.n)
  th.U<-qgamma(const*p2,n+1,s.n)
  ci.mat[i,1]<-p1
  ci.mat[i,2]<-p2
  ci.mat[i,3]<-th.L
  ci.mat[i,4]<-th.U
  ci.mat[i,5]<-th.U-th.L
}
print(ci.mat)

+      Lower.p Upper.p      Lower      Upper      Length
+ [1,] 0.0000 0.9500 0.0000000 0.8747920 0.8747920
+ [2,] 0.0025 0.9525 0.2028089 0.8804368 0.6776279
+ [3,] 0.0050 0.9550 0.2228653 0.8863497 0.6634844
+ [4,] 0.0075 0.9575 0.2360049 0.8925604 0.6565556
+ [5,] 0.0100 0.9600 0.2460674 0.8991040 0.6530366
+ [6,] 0.0125 0.9625 0.2543470 0.9060220 0.6516750
+ [7,] 0.0150 0.9650 0.2614495 0.9133644 0.6519149
+ [8,] 0.0175 0.9675 0.2677107 0.9211922 0.6534816
+ [9,] 0.0200 0.9700 0.2733375 0.9295808 0.6562433
+ [10,] 0.0225 0.9725 0.2784671 0.9386248 0.6601576
+ [11,] 0.0250 0.9750 0.2831955 0.9484452 0.6652497
+ [12,] 0.0275 0.9775 0.2875924 0.9592010 0.6716086
+ [13,] 0.0300 0.9800 0.2917106 0.9711060 0.6793955
+ [14,] 0.0325 0.9825 0.2955906 0.9844586 0.6888680
+ [15,] 0.0350 0.9850 0.2992646 0.9996927 0.7004281
+ [16,] 0.0375 0.9875 0.3027585 1.0174766 0.7147181
+ [17,] 0.0400 0.9900 0.3060933 1.0389207 0.7328274
+ [18,] 0.0425 0.9925 0.3092867 1.0660869 0.7568002
+ [19,] 0.0450 0.9950 0.3123533 1.1035489 0.7911956
+ [20,] 0.0475 0.9975 0.3153055 1.1656547 0.8503492
+ [21,] 0.0500 1.0000 0.3181541 2.0000000 1.6818459

#Shortest interval
print(ci.mat[which.min(ci.mat[,5]),])

+ Lower.p Upper.p      Lower      Upper      Length
+ 0.012500 0.962500 0.254347 0.906022 0.651675
```

For the mixture prior posterior in (iii): we have that

$$\Pi_n(\theta) = w_n \Pi_n^{(1)}(\theta) + (1 - w_n) \Pi_n^{(2)}(\theta)$$

where $\Pi_n^{(1)}(\theta)$ and $\Pi_n^{(2)}(\theta)$ are the cdfs of $\text{Gamma}(n+1, s_n+1)$ and $\text{Gamma}(n+1, s_n+2)$ respectively, and

$$w_n = \frac{\frac{\Gamma(n+1)}{(s_n+1)^{n+1}}}{\frac{\Gamma(n+1)}{(s_n+1)^{n+1}} + 2 \frac{\Gamma(n+1)}{(s_n+2)^{n+1}}}.$$

In this case, the most straightforward computational method is to compute and invert $\Pi_n(\theta)$ numerically using the uniroot function

```
gam<-0.05;p.L<-0; p.U<-gam
cont<-gamma(n+1)/(s.n+1)^(n+1) + 2*gamma(n+1)/(s.n+2)^(n+1)
w.n<-(gamma(n+1)/(s.n+1)^(n+1))/const
pmix<-function(q,wv,a1,b1,a2,b2){
  return(wv*pgamma(q,a1,b1)+(1-wv)*pgamma(q,a2,b2))
}
qmix<-function(p,wv,a1,b1,a2,b2){
  f<-function(x){return(wv*pgamma(x,a1,b1)+(1-wv)*pgamma(x,a2,b2)-p)}
  s<-uniroot(f,lower=0,upper=100)
  return(s$root)
}
#Equi-tailed
theta.L.1<-qmix(gam/2,w.n,n+1,s.n+1,n+1,s.n+2);theta.U.1<-qmix((1-gam/2),w.n,n+1,s.n+1,n+1,s.n+2)
c(theta.L.1,theta.U.1,theta.U.1-theta.L.1)

+ [1] 0.2567171 0.8597642 0.6030470

#Shortest & Any
pv<-seq(p.L,p.U,length=21)
ci.mat<-matrix(0,nrow=length(pv),ncol=5);
colnames(ci.mat)<-c('Lower.p','Upper.p','Lower','Upper','Length')
for(i in 1:length(pv)){
  p1<-pv[i];p2<-p1+(1-gam)
  th.L<-qmix(p1,w.n,n+1,s.n+1,n+1,s.n+2)
  th.U<-qmix(p2,w.n,n+1,s.n+1,n+1,s.n+2)
  ci.mat[i,1]<-p1
  ci.mat[i,2]<-p2
  ci.mat[i,3]<-th.L
  ci.mat[i,4]<-th.U
  ci.mat[i,5]<-th.U-th.L
}
print(ci.mat)

+      Lower.p Upper.p      Lower      Upper      Length
+ [1,] 0.0000 0.9500 0.0000000 0.7929983 0.7929983
+ [2,] 0.0025 0.9525 0.1838436 0.7981165 0.6142729
+ [3,] 0.0050 0.9550 0.2020271 0.8034800 0.6014529
+ [4,] 0.0075 0.9575 0.2139435 0.8091202 0.5951767
+ [5,] 0.0100 0.9600 0.2230532 0.8150362 0.5919830
+ [6,] 0.0125 0.9625 0.2305347 0.8213380 0.5908033
+ [7,] 0.0150 0.9650 0.2370086 0.8279380 0.5909295
+ [8,] 0.0175 0.9675 0.2426695 0.8350444 0.5923749
+ [9,] 0.0200 0.9700 0.2477686 0.8426600 0.5948914
+ [10,] 0.0225 0.9725 0.2524030 0.8508619 0.5984589
+ [11,] 0.0250 0.9750 0.2567171 0.8597642 0.6030470
+ [12,] 0.0275 0.9775 0.2606985 0.8695143 0.6088157
```

```

+ [13,] 0.0300 0.9800 0.2644391 0.8803062 0.6158670
+ [14,] 0.0325 0.9825 0.2679500 0.8923914 0.6244414
+ [15,] 0.0350 0.9850 0.2712627 0.9062353 0.6349726
+ [16,] 0.0375 0.9875 0.2744240 0.9223194 0.6478954
+ [17,] 0.0400 0.9900 0.2774822 0.9417903 0.6643081
+ [18,] 0.0425 0.9925 0.2803530 0.9664167 0.6860637
+ [19,] 0.0450 0.9950 0.2831481 1.0003444 0.7171963
+ [20,] 0.0475 0.9975 0.2857977 1.0566601 0.7708624
+ [21,] 0.0500 1.0000 0.2884186 100.0000000 99.7115814

```

```
#Shortest interval
```

```
print(ci.mat[which.min(ci.mat[,5]),])
```

```

+ Lower.p Upper.p Lower Upper Length
+ 0.0125000 0.9625000 0.2305347 0.8213380 0.5908033

```

6 MARKS