Stat 601 - Lab 2

Anh Le

January 24, 2013

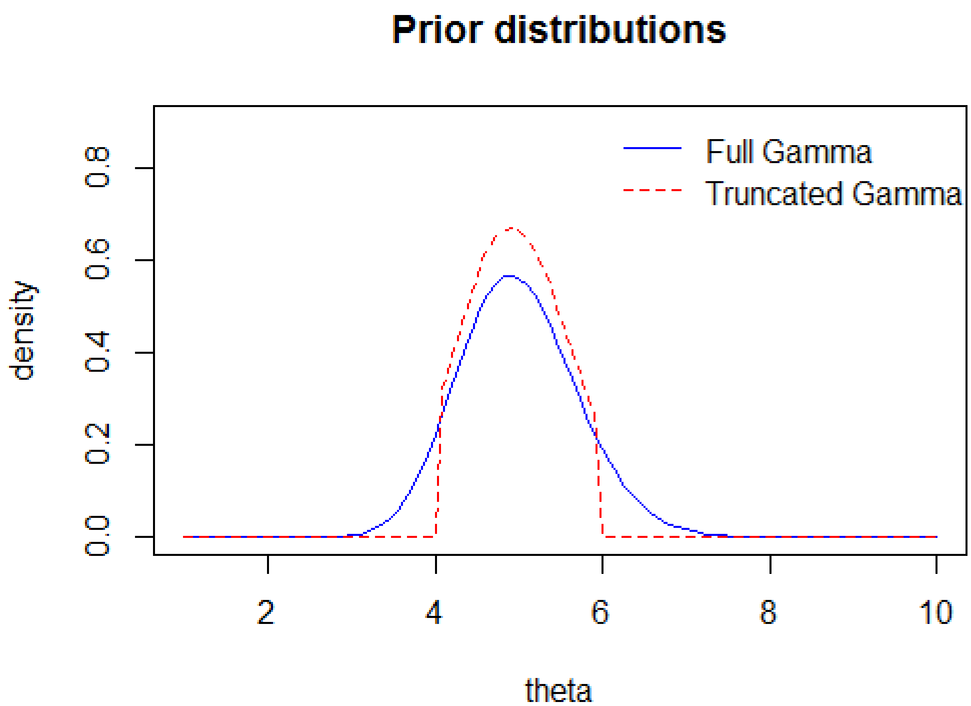

Figure 1: We see that both priors have the same shape. The truncated gamma is shifted up, compared to the full gamma, so that it integrates to 1 over support [4,6]. Outside of [4,6], the density of the truncated gamma is 0.

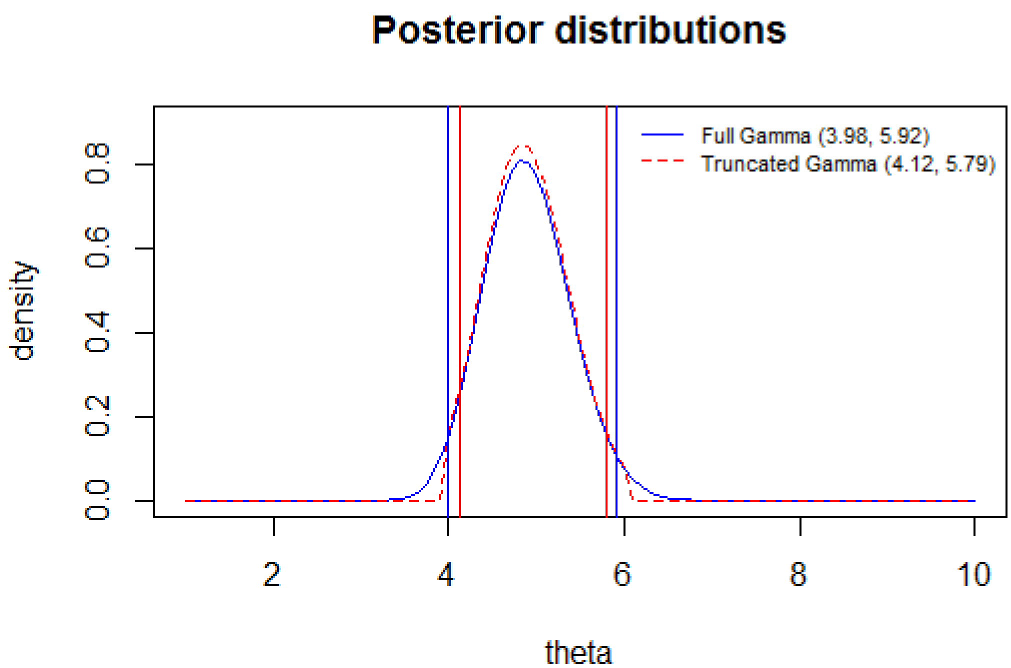

Figure 2: After being updated, both priors become almost identical. The 95% credible interval of the truncated gamma is slightly smaller because it has 0 density outside [4,6], thus less spread out.

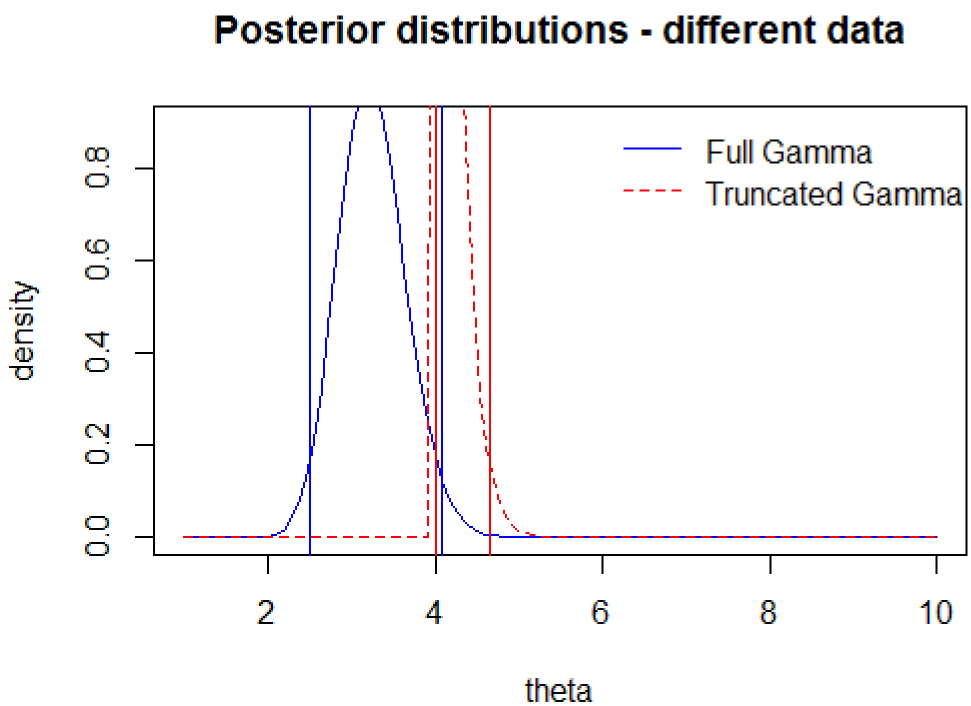

Figure 3: Given the information provided by the second set of data, the full gamma posterior shifts greatly to the left, centering at the left of 4. However, the truncated gamma is bounded at 4. Hence, we see a vertical drop at 4. This shows the danger of mistakenly truncating a prior. If the true value lies outside of the prescribed bound, the truncated posterior will fail to capture it.

Listing 1: My R Code

```
rm(list=ls())

data.1 = c(2,1,9,4,3,3,7,7,5,7)
data.2 = c(2,1,0,4,1,1,0,1,1,4)

p = seq(1,10,length=100)
d1 = dgamma(p, shape=50, scale=0.1)
plot(p, d1, type='l', col='blue',
      ylim=c(0,0.9), ylab='density', xlab='theta')
title(main='Prior distributions')
legend(x='topright', legend=c('Full Gamma','Truncated Gamma'),
        lty=c(1,2), col=c('blue','red'), bty='n')
```

```
F = function(x) {pgamma(x, shape=50, scale=0.1)}

d2 = dgamma(p, shape=50, scale=0.1)*as.numeric(p>4 & p<6) / (F(6) - F(4)) # Trun
lines(p, d2, lty=2, col='red')

######################################
### Update the prior with data.1 ###
######################################

shape.post = 50+sum(data.1) # Calculate the shape and scale of the posterior Gam
scale.post = 1/(10+length(data.1))
d1.post = dgamma(p, shape=shape.post, scale=scale.post)

# Plot the posterior density
plot(p, d1.post, type='l', col='blue',
     ylim=c(0,0.9), xlab='theta', ylab='density')
title(main='Posterior distributions')
legend(x='topright', legend=c('Full Gamma','Truncated Gamma'),
       lty=c(1,2), col=c('blue','red'), bty='n')

Fpost = function(x) {pgamma(x, shape=shape.post, scale=scale.post)}

# Find the truncated posterior
d2.post = dgamma(p, shape=shape.post, scale=scale.post)*as.numeric(p>=4 & p<=6)
  (Fpost(6) - Fpost(4))
# Plot the truncated density
lines(p, d2.post, lty=2, col='red')

# Find the quantile function for truncated posterior
qgamma.trun.post = function(p) {
  qgamma((p*(Fpost(6)-Fpost(4))+Fpost(4)), shape=shape.post, scale=scale.post)
}
# Plot the 95 interval for both posteriors
abline(v=qgamma(0.025, shape=shape.post, scale=scale.post), col='blue')
abline(v=qgamma(0.975, shape=shape.post, scale=scale.post), col='blue')
abline(v=qgamma.trun.post(0.025), col='red')
abline(v=qgamma.trun.post(0.975), col='red')
```