

## Solution to Assignment #9

1. A sample  $s = (753.0, 1396.9, 1528.2, 1646.9, 717.6, 446.5, 848.0, 1222.0, 748.5, 610.1)$  is modeled as an i.i.d sample from a  $\text{Gamma}(\alpha, \lambda)$  distribution. You may think of this data as a sample of lifetimes measured in hours.

We wish to analyze this data with a Bayesian model with the joint prior distribution for  $\theta = (\alpha, \lambda)$  having density

$$\pi(\alpha, \lambda) = \frac{3\alpha^3(4 - \alpha)}{32(400)^2\lambda^3} e^{-\alpha/(400\lambda)}, \quad 0 < \alpha < 4, \lambda > 0$$

using MCMC. (This prior density arises from  $\alpha/4 \sim \text{Beta}(2, 2)$ ,  $\lambda^{-1} | \alpha \sim \text{Gamma}(2, \alpha/800)$ .)

- (a) For a fixed  $\alpha_0$ , find an expression for  $\lambda$  that maximizes the likelihood and evaluate this for  $\alpha_0 = 2$ .

Solution: The likelihood for an i.i.d. sample from a  $\text{Gamma}(\alpha, \lambda)$  distribution for fixed  $\alpha = \alpha_0$  is

$$L(\lambda | s) = \prod_{i=1}^n \frac{\lambda_0^{\alpha_0}}{\Gamma(\alpha_0)} x_i^{\alpha_0-1} e^{-\lambda x_i} = \frac{\lambda^{n\alpha_0}}{(\Gamma(\alpha_0))^n} \left( \prod_{i=1}^n x_i \right)^{\alpha_0-1} e^{-\lambda \sum_{i=1}^n x_i}.$$

The maximum is found by taking the logarithm, setting the derivative to zero, and solving.

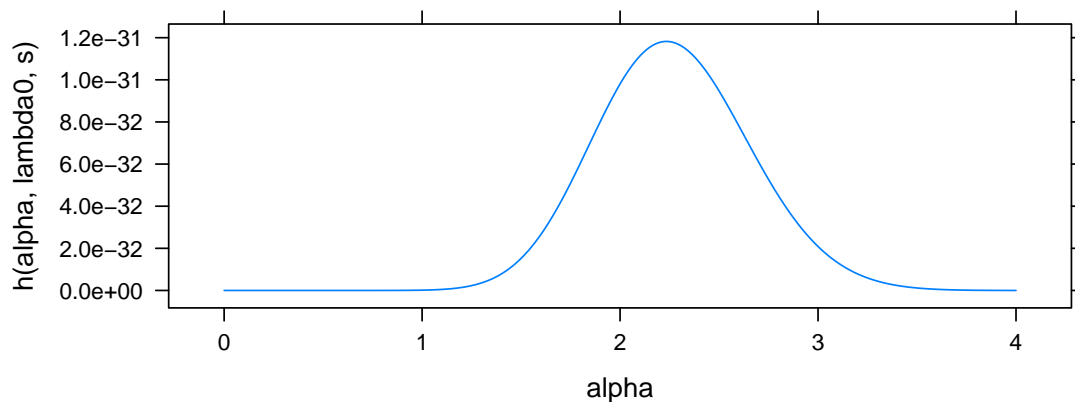
$$\begin{aligned} \frac{\partial \log L(\lambda | s)}{\partial \lambda} &= \frac{n\alpha_0}{\lambda} - \sum_{i=1}^n x_i = 0 \\ \frac{n\alpha_0}{\lambda} &= \sum_{i=1}^n x_i \\ \hat{\lambda} &= \frac{n\alpha_0}{\sum_{i=1}^n x_i} \\ &= \frac{\alpha_0}{\bar{x}} \end{aligned}$$

(This is a maximum as the second derivative is  $-n\alpha_0/\lambda^2 < 0$  for all  $\lambda$ .) Setting  $\alpha_0 = 2$ , the expression that maximizes the likelihood is  $2/\bar{x}$ .

- (b) Plot  $h(\alpha, \lambda_0)$  versus  $\alpha$  for  $\lambda_0$  equal to the value from (a) where  $h(\alpha, \lambda) = \pi(\alpha, \lambda)f_\theta(s)$ . By eye, what is an approximate good initial value to choose for  $\alpha$  where  $h$  is high? For this  $\alpha$ , which value of  $\lambda$  will maximize the likelihood? This pair is an excellent candidate to start your MCMC.

Solution: The following R code will create a graph. Assume that you have read in functions `prior()`, `likelihood()`, `h()`

```
> library(lattice)
> source("hw09.R")
> s = c(753, 1396.9, 1528.2, 1646.9, 717.6, 446.5, 848, 1222, 748.5, 610.1)
> lambda0 = 2/mean(s)
> alpha = seq(0, 4, 0.01)
> fig = xyplot(h(alpha, lambda0, s) ~ alpha, type = "l")
> print(fig)
```



We see that  $\alpha$  may be a bit higher than 2, say about 2.2. For this value of  $\alpha$ , the best  $\lambda$  is  $2.2/\bar{x} \doteq 0.002$ . This pair is a good starting value for the MCMC.

- (c) Run an initial run of MCMC of 1000 steps with the starting value from the previous part and some choices for tuning parameters using `mcmc()` in `hw09.R`. Find the mean and standard deviation of each column of the output. Ideally, the mean acceptance probability will be between 0.1 and 0.4 (say close to 0.25). Experiment with selecting values of the tuning parameters until you achieve a mean acceptance probability in this range. Which parameter values do you find work? *Hint: using a window whose width is a few posterior sd's is a good idea.*
- (d) Take a large MCMC sample using good starting values and proposal densities. On the basis of this sample, find approximate 95% credible regions for  $\alpha$  and for  $\lambda$ . *Hint: Use a command like the following: `apply(out.2[,1:2], 2, quantile, probs=c(0.025, 0.975))` to find the empirical quantiles of columns 1 and 2 from the matrix `out.2`.*

Solution: Summaries of a short MCMC run are as follows:

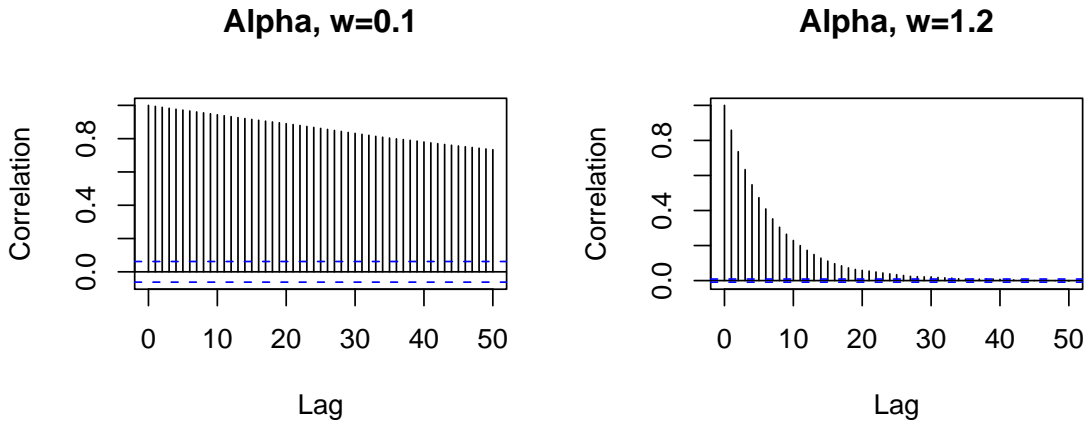
```
> set.seed(4352)
> out.1 = mcmc(1000, 2, 0.001, s, w.alpha = 0.1, w.lambda = 5e-04)
> print(apply(out.1, 2, mean))
      alpha      lambda      h  accept.prob
1.578249e+00 1.576495e-03 4.059286e-32 6.952130e-01
> print(apply(out.1, 2, sd))
      alpha      lambda      h  accept.prob
4.583983e-01 5.131521e-04 3.954406e-32 3.289919e-01
```

We see that the standard deviations of the posterior distributions for  $\alpha$  and  $\lambda$  are about 0.5 and 0.001 respectively, and that the mean acceptance rate is about 0.695. We can do better by adjusting the proposal tuning parameters.

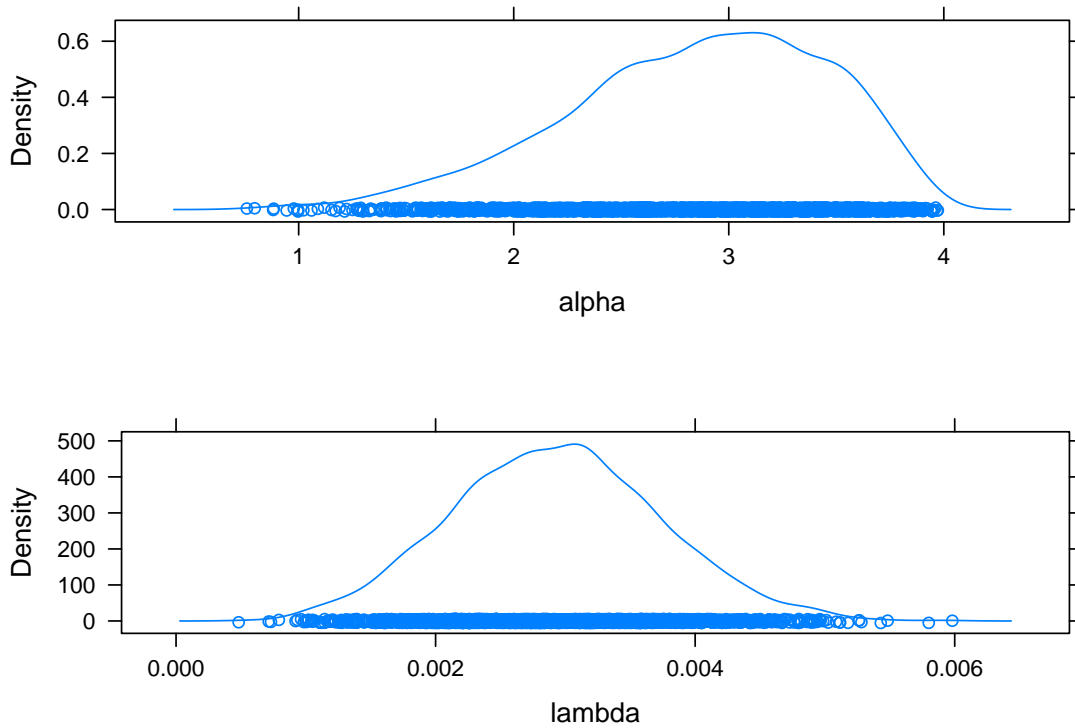
```
> out.2 = mcmc(50000, 2.2, 2.2/mean(s), s, w.alpha = 1.2, w.lambda = 0.002)
> out.alpha = out.2[, 1]
> out.lambda = out.2[, 2]
> print(apply(out.2, 2, mean))
      alpha      lambda      h  accept.prob
2.837451e+00 2.920680e-03 1.027093e-31 2.838350e-01
```

```
> print(apply(out.2, 2, sd))
      alpha      lambda      h  accept.prob
6.010594e-01 8.037033e-04 5.222167e-32 3.779096e-01
```

Here, the mean acceptance probability is closer to the sweet spot of 0.25. A graphical display of the autocorrelation function demonstrates the improved mixing with the new parameters.



Notice how the dependence drops close to zero much faster for the right. The density plot of the posterior density for  $\alpha$  shows a fair amount of skewness, so credible region intervals based on normality is inaccurate. The posterior density for  $\lambda$  is closer to normal.



Here are the quantile estimates.

```
> apply(out.2[, 1:2], 2, quantile, probs = c(0.025, 0.975))
```

```

      alpha      lambda
2.5%  1.563806  0.001395895
97.5%  3.802742  0.004520530

```

- (e) Summarize the sample to estimate the posterior mean and sd for both  $\alpha$  and  $\lambda$ . Are the intervals you found in the previous part close to the the mean plus and minus 1.96 times the sd?

Solution: The estimates based on approximate normality of the posterior distribution are not good for  $\alpha$ , but are close for  $\lambda$ .

```

> mu = apply(out.2[, 1:2], 2, mean)
> sigma = apply(out.2[, 1:2], 2, sd)
> ci.a = mu - 1.96 * sigma
> ci.b = mu + 1.96 * sigma
> print(rbind(ci.a, ci.b))

```

```

      alpha      lambda
ci.a  1.659375  0.001345422
ci.b  4.015528  0.004495939

```

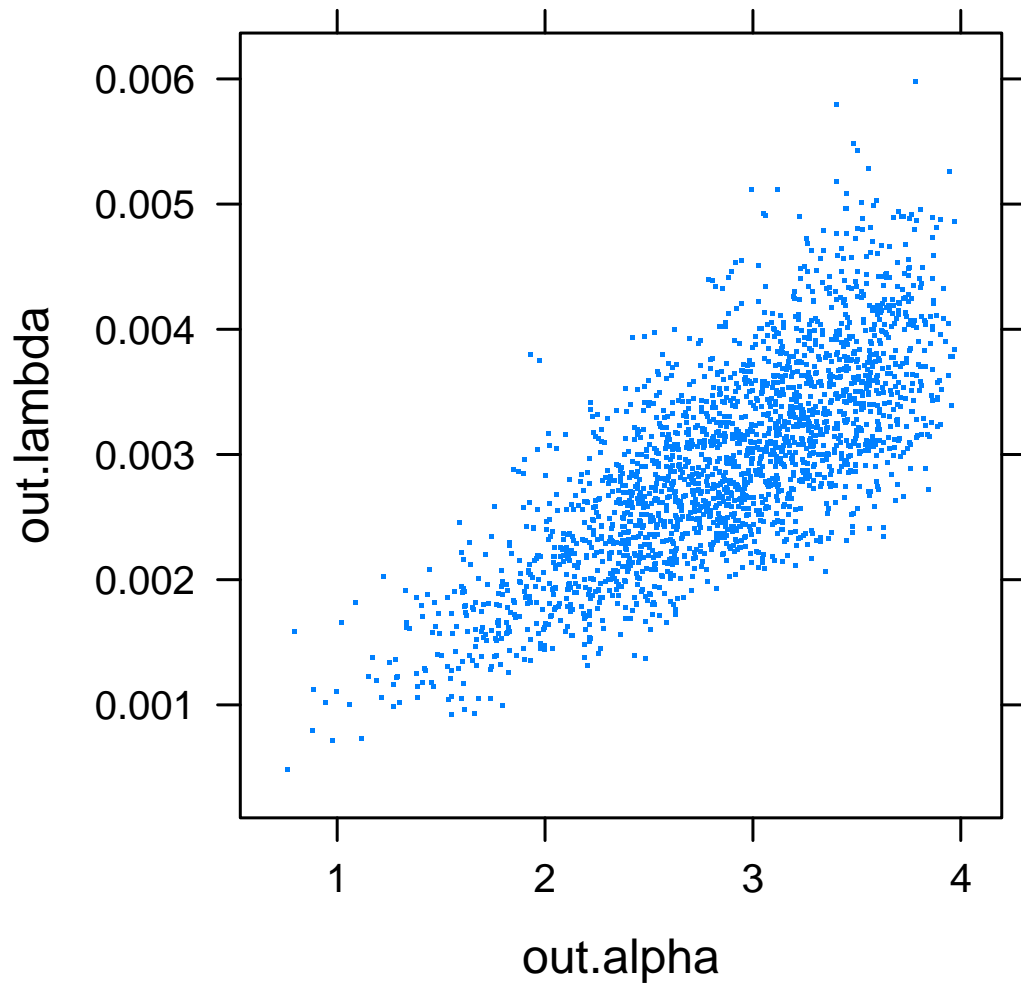
- (f) Use `xyplot()` in the `lattice` package to graph the sample  $\lambda$  versus  $\alpha$ , (`xyplot(out.2[,2] ~ out.2[,1])`). Are  $\alpha$  and  $\lambda$  nearly independent in the posterior distribution, or does information about one parameter say something about the other?

Solution: This plot graphs a sample of 2500 of the sampled pairs, and uses a small dot for the plotting character.

```

> cor(out.alpha, out.lambda)
[1] 0.7624948
> print(xyplot(out.lambda ~ out.alpha, subset = subs, pch = ".", cex = 1.5))

```



Notice that there is a positive correlation between  $\alpha$  and  $\lambda$  in the posterior distribution. When one variable is high, the other also tends to be high.