Stockholm University 1. November 2017

Bayesian Methods
Exam
Petra Poklukar

I have collaborated mostly with the german guys Luca, Maximillian and Timo by comparing and discussing the results. Note also that the knitr template was provided by Theresa Stocks.

Exercise 1

a. We assume that we want to infer the probability of defection θ $(0 \leq \theta \leq 1)$ for a product manufactured from a factory. From a random sample of 30 products, it is found that 3 of them are defected. We also assume that the probability for a product to be defected is independent of the others. In order to find the posterior of θ, we first define random variables

$$Y_i = \begin{cases} 1 & \text{if the } i\text{-th product is defected,} \\ 0 & \text{otherwise.} \end{cases}$$

for $i = 1, \dots, 30$. The random sample of 30 products thus forms a sequence of Bernoulli trials. The random variable

$$X = \sum_{i=1}^{30} Y_i$$

then counts the number of defected products in the random sample of size 30. Therefore, $X|\theta$ is distributed binomial with parameters 30 and θ, so the corresponding probability mass function is

$$f(x|\theta) = \binom{30}{x} \theta^x (1-\theta)^{30-x}.$$

Using the uniform prior $\pi(\theta) = 1$, we then compute the posterior

$$\begin{aligned} p(\theta|x) &\propto f(x|\theta) \cdot \pi(\theta) \\ &= \binom{30}{x} \theta^x (1-\theta)^{30-x} \\ &\propto \text{Beta}(x+1, 31-x) \end{aligned}$$

Since the number x of observed defected products is 3, the posterior is distributed $\text{Beta}(4, 28)$.

b. We assume now that we keep on sampling the products randomly until 3 defected products are seen. It now happens that the 30th product sampled is the third defected one we found. In this new sampling scheme, we now assume that a random variable X counts the number of perfect products preceding the third defected product, where the probability of a defected product is θ. Then $X|\theta$ is distributed negative binomial with probability mass function

$$f(x|\theta) = \binom{x+3-1}{x}\theta^3(1-\theta)^x.$$

Using the uniform prior $\pi(\theta) = 1$ again, the posterior becomes

$$\begin{aligned} p(\theta|x) &\propto f(x|\theta) \cdot \pi(\theta) \\ &= \binom{x+3-1}{x}\theta^3(1-\theta)^x \\ &\propto \mathrm{Beta}(3+1, x+1) \end{aligned}$$

Since the number x of observed perfect products preceding the third defected product is 27, the posterior is again distributed $\mathrm{Beta}(4, 28)$ which is the same as in part (a). This means that the design of the sampling scheme does not affect the posterior distribution if we are using the same prior distributions. Then the obtained posterior distributions are the same because the likelihood functions in (a) and (b) have the same shape (proportional to a constant) as functions of the parameter θ.

c. We first recall that Jeffrey's prior $\pi(\theta) \propto \sqrt{I(\theta)}$ where

$$I(\theta) = -\,\mathrm{E}_{X|\theta}\left(\frac{\partial^2}{\partial\theta^2}\log f(x|\theta)\right)$$

is the expected Fisher information in the model $X|\theta \sim \mathrm{Bin}(30, \theta)$. Using the second derivative of the log likelihood function

$$\begin{aligned} \frac{\partial^2}{\partial\theta^2}\log f(x|\theta) &= \frac{-x + 2x\theta - 30\theta^2}{\theta^2(1-\theta)^2} \\ &= -\frac{x(1+2\theta) + 30\theta^2}{\theta^2(1-\theta)^2}, \end{aligned}$$

we compute the expected Fisher information

$$\begin{aligned} -\,\mathrm{E}_{X|\theta}\left(\frac{\partial^2}{\partial\theta^2}\log f(x|\theta)\right) &= -\sum_{x=0}^{30}\binom{30}{x}\theta^x(1-\theta)^{30-x} \cdot \left[-\frac{x(1-2\theta)+30\theta^2}{\theta^2(1-\theta)^2}\right] \\ &= \frac{(1-2\theta)}{\theta^2(1-\theta)^2} \cdot \mathrm{E}(X) + \frac{30\theta^2}{\theta^2(1-\theta)^2} \\ &= \frac{(1-2\theta)\cdot 30\theta + 30\theta^2}{\theta^2(1-\theta)^2} \\ &= \frac{30}{\theta(1-\theta)}, \end{aligned}$$

since $\mathrm{E}(X) = 30\theta$. Therefore, the Jeffrey's prior is $\pi(\theta) \propto \sqrt{\dfrac{30}{\theta(1-\theta)}} \propto$

Beta$(0.5, 0.5)$ and the posterior becomes

$$\begin{aligned}
p(\theta|x) &\propto f(x|\theta) \cdot \pi(\theta) \\
&\propto \theta^x (1-\theta)^{30-x} \cdot \theta^{-\frac{1}{2}} (1-\theta)^{-\frac{1}{2}} \\
&= \theta^{x-\frac{1}{2}} (1-\theta)^{30-x-\frac{1}{2}} \\
&\propto \text{Beta}(x + \frac{1}{2}, 30 + \frac{1}{2} - x)
\end{aligned}$$

Since $x = 3$, the posterior $p(\theta|x)$ is distributed Beta$(3.5, 27.5)$.

d. We again need to compute the expected Fisher information $I(\theta)$ but this time in the model $X|\theta \sim$ NegBin$(3, \theta)$. The second derivative of the log likelihood function is

$$\begin{aligned}
\frac{\partial^2}{\partial \theta^2} \log f(x|\theta) &= \frac{\partial^2}{\partial \theta^2} \log \left[\binom{x+3-1}{x} \theta^3 (1-\theta)^x \right] \\
&= -\left(\frac{3}{\theta^2} + \frac{x}{(1-\theta)^2} \right),
\end{aligned}$$

which yields the expected Fisher information

$$\begin{aligned}
-\text{E}_{X|\theta} \left(\frac{\partial^2}{\partial \theta^2} \log f(x|\theta) \right) &= -\sum_{x=0}^{30} \binom{x+3-1}{x} \theta^3 (1-\theta)^x \cdot \left[-\left(\frac{3}{\theta^2} + \frac{x}{(1-\theta)^2} \right) \right] \\
&= \frac{3}{\theta^2} + \frac{1}{(1-\theta)^2} \cdot \text{E}(X) \\
&= \frac{3}{\theta^2} + \frac{1}{(1-\theta)^2} \cdot \frac{3(1-\theta)}{\theta} \\
&= \frac{3}{\theta^2(1-\theta)},
\end{aligned}$$

since $\text{E}(X) = \dfrac{3(1-\theta)}{\theta}$. Therefore, the Jeffrey's prior $\pi(\theta) \propto \sqrt{\dfrac{3}{\theta^2(1-\theta)}} \propto$ Beta$(0, 0.5)$, so the posterior becomes

$$\begin{aligned}
p(\theta|x) &\propto f(x|\theta) \cdot \pi(\theta) \\
&\propto \theta^3 (1-\theta)^x \cdot \theta^{-1} (1-\theta)^{-\frac{1}{2}} \\
&= \theta^2 (1-\theta)^{x-\frac{1}{2}} \\
&\propto \text{Beta}(3, x + \frac{1}{2})
\end{aligned}$$

Since in this case $x = 27$, the posterior $p(\theta|x) \sim$ Beta$(3, 27.5)$.

e. The plot of all four posterior distributions is shown in Figure 1. As mentioned in part (b), the design of the sampling scheme does not affect the posterior distribution as long as we are using the same prior. In both (a) and (b) we used the uniform prior and computed the same posterior distributions since the likelihood functions had similar shapes. However, in (c) and (d) we computed Jeffrey's priors which depend on the chosen likelihood function of the model. Thus, the obtained posterior distributions differ. It might also be worth mentioning that

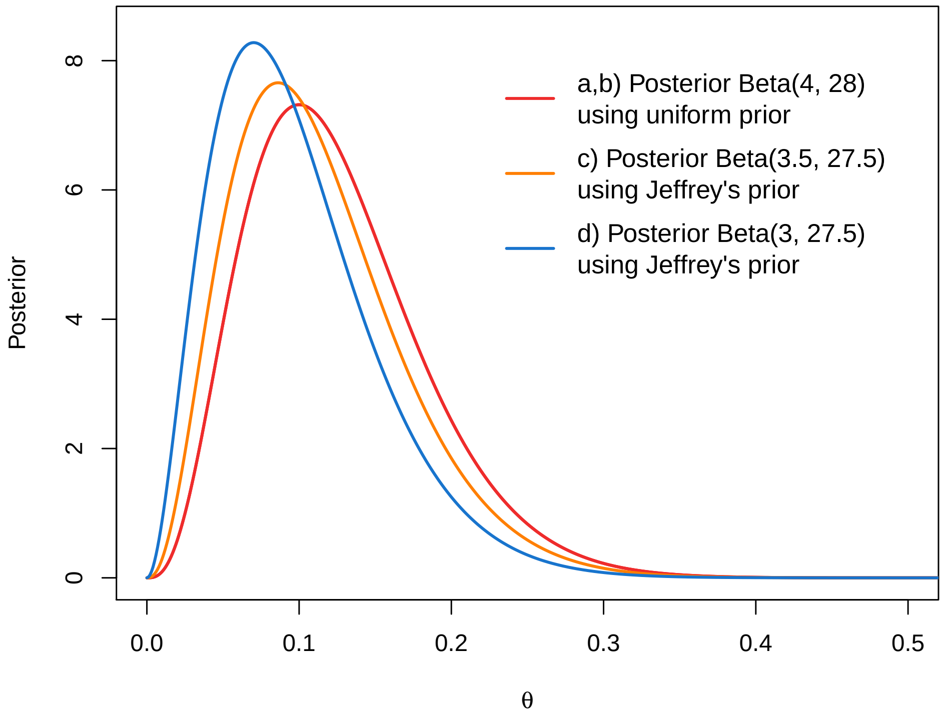

Figure 1: Plots of the four posterior distributions from parts (a), (b), (c) and (d).

the Jeffrey's prior $\text{Beta}(0, 0.5)$ is improper even though this does not affect the parameter inference. We conclude with an observation that all three graphs look fairly similar so we will get approximately the same results using any of the sampling scheme and any of the two priors.

Exercise 2 Throughout this exercise, we always refer to the paper [1].

a. The idea of the Bayes factor is to produce a number which quantifies the evidence for one of the two models in consideration. Based on that number one then decides (or not) which of the two compared models is better described by the data. It is computed as the ratio of posterior and prior distributions

$$\begin{aligned} BF_{21} &= \frac{P(H_2|D)/P(H_1|D)}{P(H_2)/P(H_1)} \\ &= \frac{P(D|H_2)}{P(D|H_1)}, \end{aligned}$$

if one uses the Bayes' Theorem. As in the paper we assume that both models are equally probable $P(H_1) = P(H_2)$, so

$$BF_{21} = \frac{P(H_2|D)}{P(H_1|D)} = \frac{P(D|H_2)}{P(D|H_1)}.$$

In order to make truthful statements about the model selection, one needs to compute a precise Bayes factor, and not only proportional to a constant as it is usually done when using Bayesian inference. This means that one needs to calculate the marginal likelihoods

$$P(D|H_i) = \int p(D|\lambda_i, H_i)p(\lambda_i|H_i)\,\mathrm{d}\lambda_i$$

which depend on the parameters λ_i of each of the two models $i = 1, 2$. Now let us assume we use improper prior distributions $p(\lambda_i|H_i)$ for parameters λ_i in each model, i.e.,

$$\int_{-\infty}^{\infty} p(\lambda_i|H_i)\,\mathrm{d}\lambda_i = \infty.$$

If we pretend, for the sake of the further formulation, that these two distributions are normalizable, then there are constants $M_{\lambda_1}, M_{\lambda_2}$ such that

$$\int_{-\infty}^{\infty} M_{\lambda_i}p(\lambda_i|H_i)\,\mathrm{d}\lambda_i = 1,$$

or equivalently,

$$M_{\lambda_i} = \left(\int_{-\infty}^{\infty} p(\lambda_i|H_i)\,\mathrm{d}\lambda_i\right)^{-1}.$$

Clearly, in case of improper priors, the constants $M_{\lambda_i} = \dfrac{1}{\infty} \approx 0$. Using the normalized priors, the formula for the Bayes factor then becomes

$$\begin{aligned} BF_{21} &= \frac{P(D|H_2)}{P(D|H_1)} \\ &= \frac{M_{\lambda_2}}{M_{\lambda_1}} \frac{\int p(D|\lambda_2, H_2)p(\lambda_2|H_2)\,\mathrm{d}\lambda_2}{\int p(D|\lambda_1, H_1)p(\lambda_1|H_1)\,\mathrm{d}\lambda_1}. \end{aligned}$$

The obvious problem here is that the obtained formula contains an undefined factor $\dfrac{M_{\lambda_2}}{M_{\lambda_1}}$ which makes it impossible to make any truthful conclusions.

However, in the paper, hypothesis H_1 has one model parameter λ whereas hypothesis H_2 has two independent model parameters λ_1, λ_2. If we repeat the above calculation of the Bayes factor for this case, we get that

$$BF_{21} = \frac{M_{\lambda_2}M_{\lambda_1}}{M_\lambda} \frac{\int p(D|\lambda_1, \lambda_2, H_2)p(\lambda_1|H_2)p(\lambda_2|H_2)\,\mathrm{d}\lambda_1\,\mathrm{d}\lambda_2}{\int p(D|\lambda, H_1)p(\lambda|H_1)\,\mathrm{d}\lambda}.$$

Therefore, we can conclude that the nominator term $M_{\lambda_2}M_{\lambda_1}$ is converging to 0 faster than the denominator hence producing $BF_{21} \approx 0$ which means that we are favoring the model in the denominator, H_1.

We could also have a situation where the denominator M_λ converges faster, which would yield the Bayes factor $BF_{21} \approx \infty$ and we would favor model H_2. However, in both cases, the use improper priors boils down to automatically favoring one of the hypotheses in consideration making them inappropriate for calculating the Bayes factor.

b. In the paper they state that the prior is flat in the region of the data and thus the likelihood will be sharp compared to the prior so it can therefore be thought of as the Dirac δ function centered at MLE $\hat{\lambda}_i$. In this case we would have that

$$\int p(D|\lambda_i, H_i)p(\lambda_i|\zeta)\,\mathrm{d}\lambda_i \approx \int \delta(\lambda_i - \hat{\lambda}_i)p(\lambda_i|\zeta)\,\mathrm{d}\lambda_i = p(\hat{\lambda}_i|\zeta)$$

by the properties of the Dirac δ function. Therefore, we get the idea to Taylor expand the exponential function in the prior distribution, $g(\lambda_i) := e^{-\frac{\lambda_i^2}{2\zeta^2}}$, around the MLE $\hat{\lambda}_i = \dfrac{C_i}{N_i}$ (we note here that there is a typo in the paper since there should be a $-$ in the exponential function). We are going to Taylor expand it up to the second order derivative and we therefore compute

$$\begin{aligned} g^{(1)}(\lambda_i) &= g(\lambda_i)\cdot\frac{-\lambda_i}{\zeta^2} =: \frac{1}{\zeta^2}g(\lambda_i)\delta_1(\lambda_i) \\ g^{(2)}(\lambda_i) &= g(\lambda_i)\left(-\frac{1}{\zeta^2}+\frac{\lambda_i^2}{\zeta^4}\right) =: \frac{1}{\zeta^2}g(\lambda_i)\delta_2(\lambda_i). \end{aligned}$$

The Taylor series for $g(\lambda_i)$ is then

$$\begin{aligned} g(\lambda_i) &= g(\hat{\lambda}_i) + g^{(1)}(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i) + \frac{g^{(2)}(\hat{\lambda}_i)}{2!}(\lambda_i-\hat{\lambda}_i)^2 + \frac{g^{(3)}(\hat{\lambda}_i)}{3!}(\lambda_i-\hat{\lambda}_i)^3 + \cdots \\ &= g(\hat{\lambda}_i)\left(1 + \frac{1}{\zeta^2}\delta_1(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i) + \frac{1}{2!\cdot\zeta^2}\delta_2(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i)^2 + \cdots\right). \end{aligned}$$

Let us define $f(D|\lambda_i) := \lambda_i^{C_i}e^{-N_i\lambda_i}$. Since $\lambda_i > 0$ is the parameter of the Poisson distribution,

$$\int_{-\infty}^{\infty} f(D|\lambda_i)\,\mathrm{d}\lambda_i = \int_0^{\infty}\lambda_i^{C_i}e^{-N_i\lambda_i}\,\mathrm{d}\lambda_i = \int_0^{\infty}\left(\frac{a}{N_i}\right)^{C_i}e^{-a}\frac{\mathrm{d}a}{N_i} = \frac{\Gamma(C_i+1)}{N_i^{C_i+1}}.$$

Similarly, we get that

$$\int_{-\infty}^{\infty} f(D|\lambda_i)\lambda_i^k\,\mathrm{d}\lambda_i = \frac{\Gamma(C_i+k+1)}{N_i^{C_i+k+1}}$$

for every $k \in \mathbb{N}\cup 0$. We now plug the Taylor series for g in the desired approximation

$$\begin{aligned} I &= \int \mathrm{d}\lambda_i f(D|\lambda_i)p(\lambda_i|\zeta) = \int \mathrm{d}\lambda_i f(D|\lambda_i)\frac{2}{\sqrt{2\pi\zeta^2}}g(\lambda_i) \\ &= \int \mathrm{d}\lambda_i f(D|\lambda_i)\frac{2}{\sqrt{2\pi\zeta^2}}g(\hat{\lambda}_i)\Bigg(1 + \frac{1}{\zeta^2}\delta_1(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i) + \\ &\quad + \frac{1}{2!\cdot\zeta^2}\delta_2(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i)^2 + \cdots\Bigg) \\ &= p(\hat{\lambda}_i|\zeta)\Bigg(\int \mathrm{d}\lambda_i f(D|\lambda_i) + \int \mathrm{d}\lambda_i f(D|\lambda_i)\frac{1}{\zeta^2}\delta_1(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i) \\ &\quad + \int \mathrm{d}\lambda_i f(D|\lambda_i)\frac{1}{2!\cdot\zeta^2}\delta_2(\hat{\lambda}_i)(\lambda_i-\hat{\lambda}_i)^2 + \cdots\Bigg) \end{aligned}$$

We further expand the integrals and use the above calculations to obtain

$$I = p(\hat{\lambda}_i|\zeta)\Bigg(\int \mathrm{d}\lambda_i f(D|\lambda_i) + \frac{\delta_1(\hat{\lambda}_i)}{\zeta^2}\Bigg[\int \mathrm{d}\lambda_i f(D|\lambda_i)\lambda_i - \hat{\lambda}_i \int \mathrm{d}\lambda_i f(D|\lambda_i)\Bigg] +$$
$$+ \frac{\delta_2(\hat{\lambda}_i)}{2! \cdot \zeta^2}\Bigg[\int \mathrm{d}\lambda_i f(D|\lambda_i)\lambda_i^2 - 2\hat{\lambda}_i \int \mathrm{d}\lambda_i f(D|\lambda_i)\lambda_i + \hat{\lambda}_i^2 \int \mathrm{d}\lambda_i f(D|\lambda_i)\Bigg] + \cdots \Bigg)$$
$$= p(\hat{\lambda}_i|\zeta)\Bigg(\frac{\Gamma(C_i+1)}{N_i^{C_i+1}} + \frac{\delta_1(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{\Gamma(C_i+2)}{N_i^{C_i+2}} - \hat{\lambda}_i \frac{\Gamma(C_i+1)}{N_i^{C_i+1}}\Bigg] +$$
$$+ \frac{\delta_2(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{\Gamma(C_i+3)}{N_i^{C_i+3}} - 2\hat{\lambda}_i \frac{\Gamma(C_i+2)}{N_i^{C_i+2}} + \hat{\lambda}_i^2 \frac{\Gamma(C_i+1)}{N_i^{C_i+1}}\Bigg] + \cdots \Bigg).$$

By taking out the term $\frac{\Gamma(C_i+1)}{N_i^{C_i+1}}$ we get that

$$I = \Bigg[p(\hat{\lambda}_i|\zeta)\frac{\Gamma(C_i+1)}{N_i^{C_i+1}} \Bigg] \cdot \Bigg(1 + \frac{\delta_1(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{\Gamma(C_i+2)}{\Gamma(C_i+1)N_i} - \hat{\lambda}_i \Bigg] +$$
$$+ \frac{\delta_2(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{\Gamma(C_i+3)}{\Gamma(C_i+1)N_i^2} - 2\hat{\lambda}_i \frac{\Gamma(C_i+2)}{\Gamma(C_i+3)N_i} + \hat{\lambda}_i^2 \Bigg] + \cdots \Bigg).$$

We now use the following property of Γ function

$$\frac{\Gamma(C_i+1+k)}{\Gamma(C_i+1)} = (C_i+1)(C_i+2)\cdots(C_i+k)$$

and the equality $\hat{\lambda}_i = \frac{C_i}{N_i}$ to get that

$$I = \Bigg[p(\hat{\lambda}_i|\zeta)\frac{\Gamma(C_i+1)}{N_i^{C_i+1}} \Bigg] \cdot \Bigg(1 + \frac{\delta_1(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{C_i+1}{N_i} - \hat{\lambda}_i \Bigg] +$$
$$+ \frac{\delta_2(\hat{\lambda}_i)}{\zeta^2}\Bigg[\frac{(C_i+2)(C_i+1)}{N_i^2} - 2\hat{\lambda}_i \frac{C_i+1}{N_i} + \hat{\lambda}_i^2 \Bigg] + \cdots \Bigg)$$
$$= \Bigg[p(\hat{\lambda}_i|\zeta)\frac{\Gamma(C_i+1)}{N_i^{C_i}} \Bigg] \cdot \Bigg(1 + \frac{\delta_1(\hat{\lambda}_i)}{\zeta^2 N_i} + \frac{\delta_2(\hat{\lambda}_i)}{\zeta^2}\frac{C_i+2}{N_i^2} + \cdots \Bigg).$$

Using the definition of δ_i we finaly get

$$I = \Bigg[p(\hat{\lambda}_i|\zeta) \int \mathrm{d}\lambda_i \lambda_i^{C_i} e^{-N_i\lambda_i} \Bigg] \cdot \Bigg(1 - \frac{C_i}{\zeta^2 N_i^2} + \frac{C_i+2}{\zeta^4 N_i^4}\left[C_i^2 - N_i^2\zeta^2\right] + \cdots \Bigg).$$

From here we finally see that the leading correction term is $-\frac{C_i}{\zeta^2 N_i^2}$ and the perturbation parameter is $\frac{1}{N\zeta^2}$ since it is increasing exponentially in the residual series. According to Wikipedia [2], computing the order of the magnitude of the leading correction term amounts to writing it as

$$\frac{C_i}{N_i^2\zeta^2} = a \cdot 10^b$$

where $0.5 \leq a \leq 5$. The number b then represents the order of the magnitude. Therefore, we calculate that

$$\log \frac{C_i}{N_i^2 \zeta^2} = \log a + b \cdot \log 10$$

which yields that the order of the magnitude

$$b \approx \log \frac{C_i}{N_i^2 \zeta^2} - \log a$$

since $\log_{10} 10 \approx 1$. We could also neglect the $\log a$ term since it is close to 0 as $0.5 \leq a \leq 5$ and we would get that

$$b \approx \log \frac{C_i}{N_i^2 \zeta^2}.$$

c. **COMPARING TRAJECTORY SEGMENTS**

Suppose the proposed algorithm detected the following 4 change points.

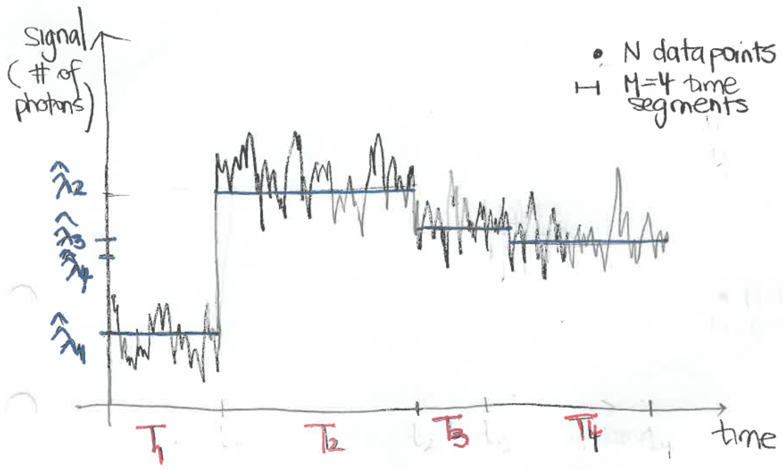

Figure 2: Detected change points in the signal.

QUESTION: How many different states of the system does the above signal represent?

(1) Sort the obtained trajectory segments by increasing MLE of the mean counts $\hat{\lambda}_i$.

Consider the following hypothesis test:

H_1 : there is no change point = there is one system state

H_2 : there is a change point = there are two system states.

(2) Generate all possible partitions of the trajectory segments into two system states.

In the example above we get three possible partitions:

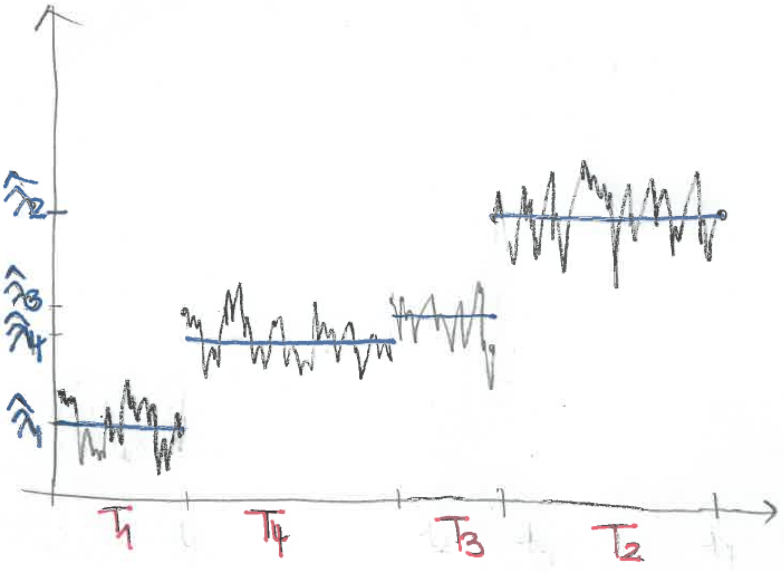

Figure 3: Sorted trajectory segments.

1. state	2. state	partition
T_1	$T_2 \cup T_3 \cup T_4$	π_1
$T_1 \cup T_2$	$T_3 \cup T_4$	π_2
$T_1 \cup T_2 \cup T_3$	T_4	π_3

Then

$$\sum_{\text{partitions } \pi_i} P(D|t_s, H_2) = \underbrace{P(D|H_2)}_{\text{likelihood of the second hypothesis}}$$

(3) Compute Bayes factor.

$$BF = \frac{\sum_{\text{partitions } \pi_i} P(D|H_2, t_s)}{P(D|H_1)}$$

Note the limited choice of possible change points compared to the change point detection algorithm!

(4) Determine state dividers (if any).
Let τ_{cutoff} be a predetermined cutoff value. Then

- if $BF < \tau_{\text{cutoff}} \Rightarrow$ there is no change point, i.e., there is only one system state.
- if $BF > \tau_{\text{cutoff}} \Rightarrow$ determine the dividing $\hat{\lambda}$.

(5) Divide and conquer. Repeat steps (2), (3), (4) on T_1 and $T_2 \cup T_2 \cup T_3$, see Figure 4. The algorithm then detects 3 different system states shown in Figure 5.

(6) Optional cleanup.

1. Increase skepticism by increasing the τ_{cutoff}.

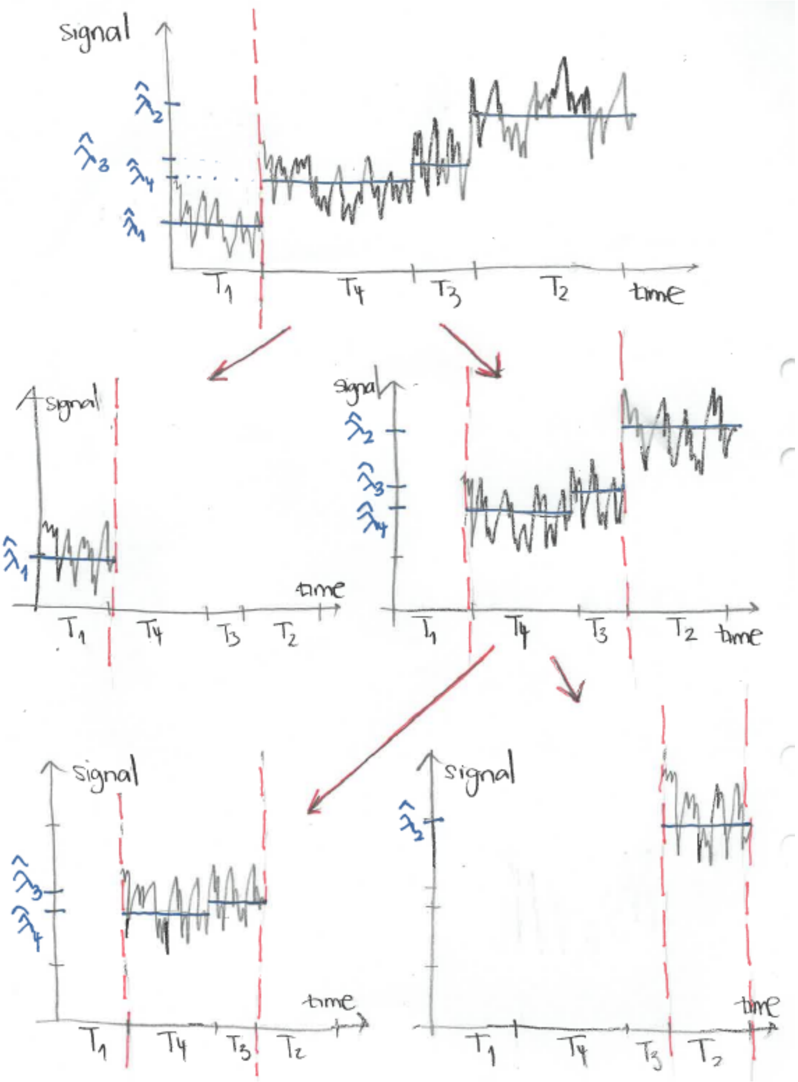

Figure 4: Sorted trajectory segments.

2. Compute BF for the new cutoff values for each of the detected state dividers.

d. I believe there are two possible criticisms of the proposed method: lack of tracking uncertainty of detected change points and lack of discussion on how to determine an appropriate Bayes factor cutoff value.

In the paper they even state that "a tool for detecting a change point should be impartial and systematic, and it should measure the uncertainty associated with change point detection", however, their proposed method clearly does not fulfill these requirements, which is my first criticism. Including and displaying a 95% credibility intervals associated with detected change points would have at least two advantages. Firstly, we could critically evaluate the accuracy of the method,

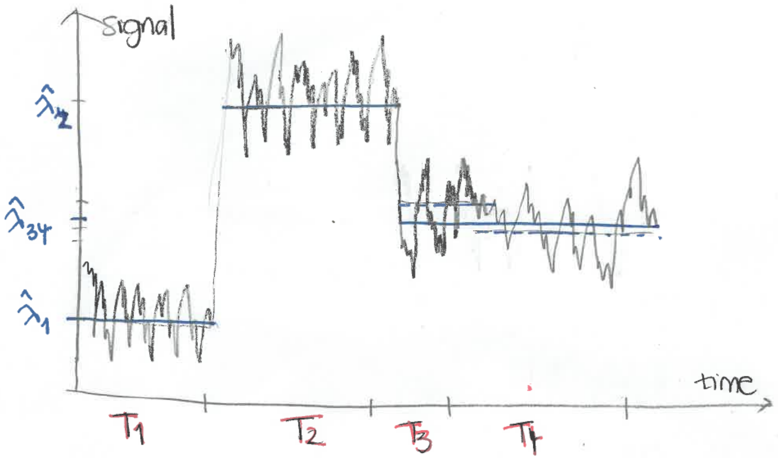

Figure 5: Original trajectory with marked 3 detected system states with mean counts λ_1, λ_2 and λ_{34}.

and secondly, we could determine a better cutoff value for the Bayes factors. This could also be a solution for my second criticism which is that they do not discuss how the Bayes factor cutoff value is determined. In Section 3 they test the proposed method on a simulated trajectories which they know to have 6 system states by construction. Using the Bayes factor cutoff values of 10, the algorithm performs well on the 10000-point trajectory and detects 6 different system states. However, running it on the 100000-point trajectory detects 15 different system states. Their solution is simply to significantly increase the cutoff value, namely from 10 to 10^{29} which is a very big range! I believe that this solution simply does not work in practice since we do not know the number of system states in advance - this is nevertheless what we want to estimate. So how should one decide on the cutoff value? If the credibility intervals were included in the change point detection method we could argue that if the intervals are very wide then we need to be more skeptical and hence increase the cutoff value. We would then reach a cutoff value that produces sufficiently narrow intervals, which would at the end help us decide on the number of system states.

Another possible criticism could be the performance of the method in terms of speed which they did not discuss either. They state that "the evaluation of the Bayes factor for the Poisson case scales as $O(N)$ since the number of putative change points scales as N" so a large number of points (i.e., a long trajectory) slows down the algorithm significantly. However, this is hard to judge without experience in data analysis.

Exercise 3 We consider the following Poisson-Gamma model

$$\begin{aligned} y|\lambda &\sim \mathrm{Po}(t \cdot \lambda) \\ \lambda &\sim \mathrm{Gamma}(a, b), \end{aligned}$$

where $a > 0, b > 0$ are hyperparameters and $t > 0$ a known value. Note first that we are going to use the notation t instead of e for the theoretical computations and the following parametrization of the Gamma distribution

$$\pi(\lambda) = \frac{\lambda^{a-1}e^{-\frac{\lambda}{b}}}{\Gamma(a)b^a}.$$

a. Since the prior Gamma distribution is proper for $\lambda \in [0, \infty)$, we compute the posterior distribution as

$$\begin{aligned} p(\lambda|y) &\propto f(y|\lambda) \cdot \pi(\lambda) \\ &= \frac{(t\lambda)^y e^{-(t\lambda)}}{y!} \cdot \frac{\lambda^{a-1}e^{-\frac{\lambda}{b}}}{\Gamma(a)b^a} \\ &= \frac{t^y}{y!\Gamma(a)b^a} \cdot \lambda^{y+a-1}e^{-\lambda(t+1/b)} \\ &\propto \text{Gamma}\left(a+y, \frac{1}{t+1/b}\right) \\ &= \text{Gamma}\left(a+y, \frac{b}{tb+1}\right) \end{aligned}$$

b. Since $\lambda|y \sim \text{Gamma}\left(a+y, \dfrac{b}{tb+1}\right)$ we know that the mean is computed as the product

$$\mathrm{E}(\lambda|y) = (a+y)\frac{b}{tb+1}. \tag{0.1}$$

Next, we compute the first derivative of the likelihood function (thought of as a function of the parameter λ)

$$\begin{aligned} f'(y|\lambda) &= \left(\frac{(t\lambda)^y e^{-(t\lambda)}}{y!}\right)' \\ &= \frac{t^y}{y!}\left[y\lambda^{y-1}e^{-(t\lambda)} + \lambda^y e^{-(t\lambda)}(-t)\right] \\ &= \frac{t^y}{y!}\lambda^{y-1}e^{-(t\lambda)}(y - \lambda t). \end{aligned}$$

Thus, the maximum likelihood estimator for λ is $\hat{\lambda} = \dfrac{y}{t}$. Since the prior mean of λ is $\mathrm{E}(\lambda) = ab$, we can write

$$\begin{aligned} \mathrm{E}(\lambda|y) &= (a+y)\frac{b}{tb+1} \\ &= ab \cdot \frac{1}{tb+1} + \frac{y}{t} \cdot \frac{bt}{tb+1} \\ &= \mathrm{E}(\lambda) \cdot \frac{1}{tb+1} + \hat{\lambda} \cdot \frac{bt}{tb+1} \end{aligned}$$

which is exactly a weighted average of the prior mean of λ and the maximum likelihood estimator for λ.

c. We compute the marginal distribution for y in the following way

$$\begin{aligned}
m(y|a,b) &= \int_0^\infty \frac{(t\lambda)^y e^{-(t\lambda)}}{y!} \cdot \frac{\lambda^{a-1}e^{-\frac{\lambda}{b}}}{\Gamma(a)b^a}\,\mathrm{d}\lambda \\
&= \frac{t^y}{y!\Gamma(a)b^a}\int_0^\infty \lambda^{y+a-1}e^{-\lambda(t+1/b)}\,\mathrm{d}\lambda \\
&= \frac{t^y}{y!\Gamma(a)b^a}\Gamma(y+a)\left(\frac{b}{bt+1}\right)^{y+a}
\end{aligned}$$

since $\displaystyle\int_0^\infty \frac{\lambda^{y+a-1}e^{-\lambda(t+1/b)}\,(t+1/b)^{y+a}}{\Gamma(y+a)}\,\mathrm{d}\lambda = 1$ as it is the integral of a probability density function. Then we further compute that

$$\begin{aligned}
m(y|a,b) &= \frac{\Gamma(y+a)}{y!\Gamma(a)}(bt)^y\left(\frac{1}{bt+1}\right)^{y+a} \\
&= \frac{\Gamma(y+a)}{y!\Gamma(a)}\left(\frac{bt}{bt+1}\right)^{y}\left(\frac{1}{bt+1}\right)^{a} \\
&= \frac{\Gamma(y+a)}{y!\Gamma(a)}\left(\frac{1}{bt+1}\right)^{a}\left(1-\frac{1}{bt+1}\right)^{y} \\
&= \binom{y+a-1}{y}\left(\frac{1}{bt+1}\right)^{a}\left(1-\frac{1}{bt+1}\right)^{y}.
\end{aligned}$$

Hence, the marginal distribution is exactly $\text{NegBin}\left(a, \dfrac{1}{bt+1}\right)$. To convert this result to the expected one we transform the parameter b to $\dfrac{1}{b}$ (i.e., we use different parametrization of the prior Gamma distribution) and we obtain $\text{NegBin}\left(a, \dfrac{1}{t/b+1}\right) = \text{NegBin}\left(a, \dfrac{b}{b+t}\right)$ which is what we wanted to prove.

d. Assume now that $y = (y_1, \dots, y_n)'$ and $e = (e_1, \dots, e_n)'$ are both n-dimensional and that $y_1, \dots, y_n$ are independent given η. Since we know from (c) that $y_i|a,b \sim \text{NegBin}\left(a, \dfrac{1}{bt+1}\right)$, we have that the marginal likelihood of y given η is

$$\begin{aligned}
m(y|a,b) &= \prod_{i=1}^n m(y_i|a,b) \\
&= \prod_{i=1}^n \binom{y_i+a-1}{y_i}\left(\frac{1}{bt_i+1}\right)^{a}\left(1-\frac{1}{bt_i+1}\right)^{y_i}.
\end{aligned}$$

Therefore, the R function is implemented in the following way

```r
# ------------------------------------------------------------- #
# mloglik: returns the log marginal likelihood for the
# Poisson-Gamma model
#
# Parameters:
# eta - vector of hyperparameters (a, b) for prior Gamma distribution
# y - n-dimensional vector of observations
# t - n-dimensional correction vector in Poisson distribution
# ------------------------------------------------------------- #
mloglik <- function(eta, y, t) {
  sum(dnbinom(y, size = eta[1], prob = 1/(eta[2] * t + 1), log = TRUE))
}
```

e. We write an R function psummary(y, e, a, b, alpha=0.05) according to the instructions. We note here that the basic idea is taken from [3].

```
# ------------------------------------------------------------- #
# psummary: for given Poisson-Gamma model with parameters 'lambda',
# 'e', 'a' and 'b', this function returns a matrix of dimensions
# n x 4 where n is the number of observations. For each pair (yi, ei),
# the ith row consists of posterior mean of lambda, an equitailed
# 100x(1-alpha)% credibility region for lambda, and the posteriror
# probability of the hypothesis H0: lambda <= 1
#
# Parameters:
# y - vector of observations (n-dimensional)
# t - correction vector in Poisson distribution (n-dimensional)
# a - shape parameter of the prior Gamma distribution
# b - scale parameter of the prior Gamma distribution
# alpha - credibility region parameter
# ------------------------------------------------------------- #
psummary <- function(y, t, a, b, alpha = 0.05) {
    n <- length(y)
    res <- matrix(ncol = 4, nrow = n,
                  dimnames=list(NULL, c("posterior mean ",
                                        "2.5% tail ",
                                        "97.5% tail ",
                                        "H_0: lambda <= 1")))

    for (i in 1:n) {
        # posterior mean is the weighted average
        res[i, 1] <- (a + y[i]) * (b/(b * t[i] + 1))

        # 95% credibility interval2
        res[i, 2:3] <- qgamma(c(alpha/2, 1 - alpha/2),
                              shape = y[i] + a,
                              scale = b/(b * t[i] + 1))

        # probability that lambda <= 1
        res[i, 4] <- pgamma(1, shape = y[i] + a,
                            scale = b/(b * t[i] + 1))
    }
    return(res)
}
```

To test if my computations were correct I also wrote an alternative psummary.alt(y, t, a, b, alpha=0.05) function using JAGS library.

```
# ------------------------------------------------------------- #
# psummary.alt: returns exactly the same as psummary. The
# only difference is that it uses JAGS library for
# computations instead of exact formulas from parts a, b, c.
#
# Parameters: see psummary description
# ------------------------------------------------------------- #
psummary.alt <- function(y, t, a, b, alpha = 0.05) {
    n <- length(y)
    res <- matrix(ncol = 4, nrow = n,
                  dimnames=list(NULL, c("posterior mean ",
                                        "2.5% tail ",
                                        "97.5% tail ",
                                        "H0: lambda <= 1")))

    for (i in 1:n) {
        res[i, ] <- createJagsModel(y[i], t[i], a, b, alpha = 0.05)
    }
    res
}

# -------------------------------------------------------------- #
# createJagsModel: compiles a jags model and returns a list
# consisting of the posterior mean of lambda, an equitailed
# 100x(1-alpha)% credibility region for lambda, and the
# posterior probability of the hypothesis H0: lambda <= 1.
#
# Parameters:
# yi- the ith observation
# ti - the ith correction scalar
# a - shape parameter of the prior Gamma distribution
# b - scale parameter of the prior Gamma distribution
# alpha - credibility region parameter
# -------------------------------------------------------------- #
createJagsModel <- function(yi, ti, a, b, alpha = 0.05) {
    PLmodel.string <- "model{
        mu <- ti*lambda
        yi ~ dpois(mu)

        # Prior
        lambda ~ dgamma(a, 1/b)
    }"

    PLmodel <- jags.model(textConnection(PLmodel.string),
                          data = list('yi' = yi, 'ti' = ti,
                                      'a' = a, 'b' = b),
                          n.chains = 1, n.adapt = 100, quiet = TRUE)

    samples.i <- window(coda.samples(PLmodel,
                                     variable.names = c("lambda"),
                                     n.iter = 20000), start = 9000)
    lambda <- as.numeric(unlist(samples.i))

    postMean.i <- mean(lambda)
    ci95.i <- quantile(lambda, c(alpha/2, 1 - alpha/2))
    h0.i <- mean(lambda <= 1)

    return(c("posterior mean" = postMean.i, ci95.i, h0.i))
}
```

f. Let us now assume that $y = (13, 5, 36)$ and $t = (5.7219, 8.9395, 40.8851)$. We first find the global maximum of the marginal log likelihood function `mloglik` implemented in (d) by using the R function `optim` with parameter `control=list(fnscale=-1)` which turns a minimization problem into a maximization problem. We use `c(1, 1)` as the vector of initial values for the parameters (a, b) to be optimized over.

Table 1: Output of the `psummary` function.

posterior mean	2.5% tail	97.5% tail	H_0: lambda <= 1
1.7837338	1.0578792	2.696111	0.0153748
0.7531968	0.3618424	1.285582	0.8521038
0.9064435	0.6505987	1.204052	0.7553363

Table 2: Output of the `psummary.alt` function.

posterior mean	2.5% tail	97.5% tail	H_0: lambda <= 1
1.7904689	1.0456307	2.736970	0.0168167
0.7467432	0.3609658	1.269741	0.8573766
0.9055504	0.6508271	1.205799	0.7592037

```
# optimizing  marginal log likelihood with given values
y <- c(13, 5, 36)
e <- c(5.7219, 8.9395, 40.8851) # we use e instead of t
max.lik <- optim(c(1, 1), mloglik, y = y, t = e,
                 control = list(fnscale = -1))$par
```

The result of the optimization is $\hat{\eta} = (5.0417832, 0.2276496)$. We now use the obtained parameters as the input in functions `psummary` and `psummary.alt`

```
# output of psummary with optimized parameters
psummary.res <- psummary(y, e, max.lik[1], max.lik[2])
psummary.alt.res <- psummary.alt(y, e, max.lik[1], max.lik[2])
```

The results of the `psummary` and `psummary.alt` functions are shown in Table 1 and Table 2 respectively. Comparing the output we see that both functions produce fairly similar results.

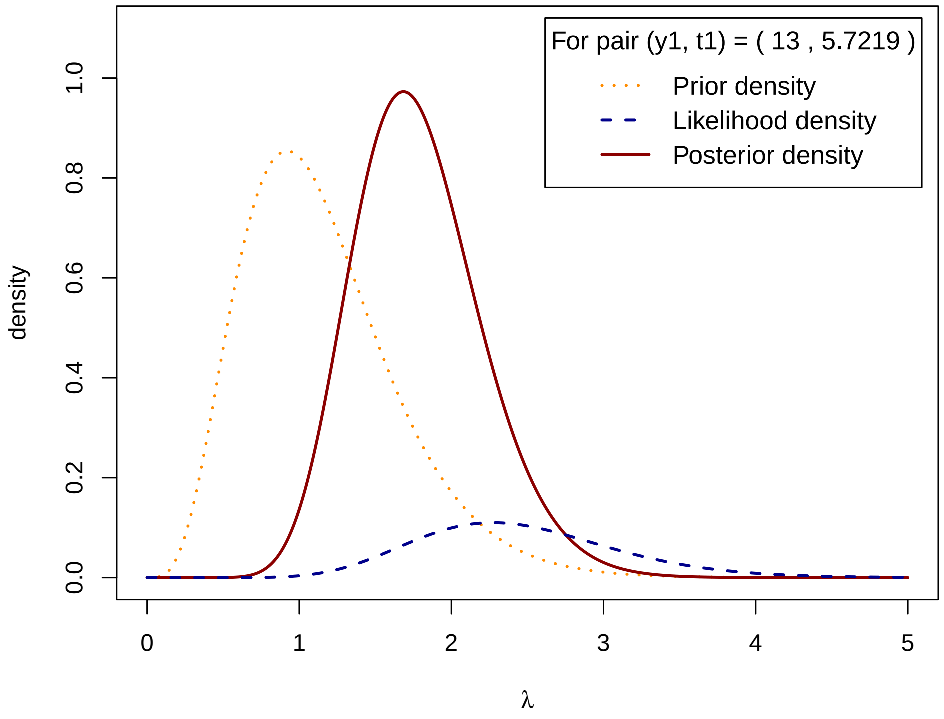

Figure 6: Plots of prior, likelihood and posterior densities as functions of λ for the pair (y_1, t_1).

Next, we show the information update by plotting prior, likelihood and posterior density as functions of λ for the first data pair (y_1, e_1). The results are shown in Figure 6. We first observe that the posterior distribution has a similar shape as the prior but it is shifted to the right. Although we plot the likelihood function as a function of λ and not y (which means it is not a probability density function), we can still look at the supports of the prior and likelihood function The support of the earlier is roughly in the interval $(0, 3)$ whereas the support of the later is roughly in the interval $(0, 4)$ with the maximum between 2 and 3. Therefore, the posterior distribution, which is computed approximately as the product of the prior and likelihood distributions, is shifted in the direction of the maximum of the likelihood, i.e., right of the prior.

g. Since $Y|\lambda \sim \text{Po}(t\lambda)$ we have that $\text{E}_{Y|\lambda} = t\lambda$ and $\text{Var}_{Y|\lambda} = t\lambda$. We have also computed that the posterior mean $\text{E}(\lambda|y, t) = (y + a)\dfrac{b}{bt + 1} =: \hat{\lambda}_{PEB}$. We now compute its MSE using the formula from lecture [4]

$$\begin{aligned}
\mathrm{MSE}(\hat{\lambda}_{PEB}) &= \mathrm{Var}_{Y|\lambda}(\hat{\lambda}_{PEB}) + \mathrm{E}_{Y|\lambda}(\hat{\lambda}_{PEB} - \lambda)^2 \\
&= \mathrm{Var}_{Y|\lambda}\left(\frac{(y+a)\cdot b}{bt+1}\right) + \mathrm{E}_{Y|\lambda}\left(\frac{(y+a)\cdot b}{bt+1} - \lambda\right)^2 \\
&= \left(\frac{b}{bt+1}\right)^2 \cdot t\lambda + \left(\frac{b}{tb+1}\cdot t\lambda + \frac{ab}{tb+1} - \lambda\right)^2 \\
&= \left(\frac{b}{bt+1}\right)^2 \cdot t\lambda + \left(\frac{b\cdot(a-\lambda/b)}{tb+1}\right)^2 \\
&= \left(\frac{b}{bt+1}\right)^2 \cdot \left[t\lambda + \left(a - \frac{\lambda}{b}\right)^2\right] \\
&= \left(\frac{b}{bt+1}\right)^2 \cdot \left[\frac{\lambda^2}{b^2} + \lambda\left(t - \frac{2a}{b}\right) + a^2\right].
\end{aligned}$$

We also compute MSE for the maximum likelihood estimator $\hat{\lambda}_{ML} = \dfrac{\lambda}{t}$ of λ which we computed in part (b). In this case

$$\begin{aligned}
\mathrm{MSE}(\hat{\lambda}_{ML}) &= \mathrm{Var}_{Y|\lambda}(\hat{\lambda}_{ML}) + \mathrm{E}_{Y|\lambda}(\hat{\lambda}_{ML} - \lambda)^2 \\
&= \mathrm{Var}_{Y|\lambda}\left(\frac{y}{t}\right) + \mathrm{E}_{Y|\lambda}\left(\frac{y}{t} - \lambda\right)^2 \\
&= \frac{1}{t^2}\cdot t\lambda + \left(\frac{1}{t}\cdot t\lambda - \lambda\right) \\
&= \frac{\lambda}{t}.
\end{aligned}$$

We then implement function `mse.lambdapeeb(a, b, t, lambda)` in R that returns the mean squared error for the posterior mean $\mathrm{E}(\lambda|y,t)$.

```r
# ------------------------------------------------------------ #
# mse.lambdapeeb: returns MSE for the posterior mean
#
# parameters:
# a, b - hyperparameters of the prior Gamma distribution
# t - correction scalar for the Poisson likelihood distribution
# lambda - observed value of the parameter
# ------------------------------------------------------------ #
mse.lambdapeeb <- function(a, b, t, lambda) {
  (b/(b * t + 1))^2 * ((lambda/b)^2 + lambda * (t - 2 * a/b) + a^2)
}
```

We then plot it for $\lambda \in [0,2]$, $a = 5$, $\dfrac{1}{b} = 5$ (due to the different choice of the Gamma parametrization), and $t = 10$. The result is shown in Figure 7. We see that the lines intersect in two points, approximately at $\lambda_1 \approx 0.6$ (shown in the figure) and at $\lambda_2 \approx 2.5$ (not shown in the figure). We note that these are not the exact values but only estimates based on the shown plots. We see that the posterior mean estimate has a smaller mean square error for $\lambda \in (0.6, 2.5)$ and it is hence in this case more appropriate for further inference. Since the given a and b values are approximately the same as the ones we got by maximizing the

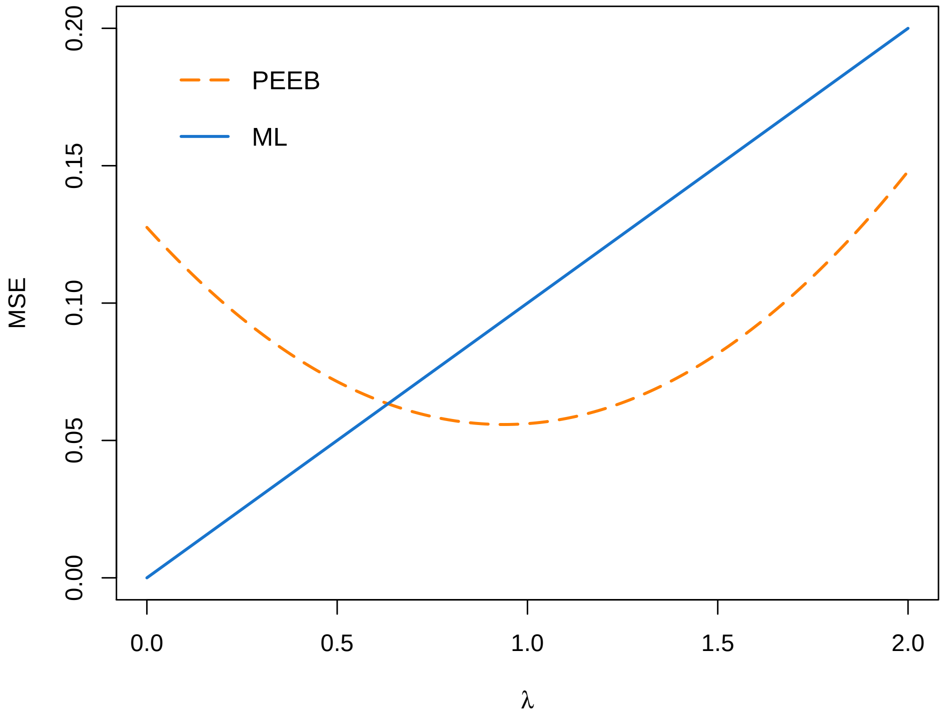

Figure 7: Plots of MSE for the posterior mean and the MLE for λ.

marginal likelihood, we can return back to Figure 6. From here we see that the support of the prior is approximately within the interval $(0, 3)$ with maximum at around 1, which means that, according to our calculations of MSE, λ_{PEB} would be a better choice.

Exercise 4 In this exercises we always refer to the paper [5].

a. In health and social sciences, one wishes to obtain an accurate estimation of provider-specific Standardized Mortality Ratio (SMR) and their percentiles, and then use the estimates to evaluate performance of each provider, i.e., to identify excellent and poor providers. In order to obtain a valid comparison of providers, one needs to keep track of both estimated values as well as their statistical uncertainty. The goal of the paper is to use Bayesian hierarchical models to obtain SMR estimates based on the observed and expected number of deaths, and use them to compare providers. They also compared their results based on the posterior mean SMRs estimates with the ones based on the ML estimates of SMRs.

First, they used the Poisson-Normal model (Section 3) to infer expected number of deaths for each provider. Using a simple Poisson model for observed number of deaths (Section 4) they obtained the ML estimates of SMRs as the ratio of observed/expected number of deaths. One then observes that when a provider has a small expected number of deaths, its MLE is going to be large. From here we can see the motivation for using Bayesian hierarchical models: they enable us to include all relevant uncertainties which help stabilize the estimates. Another advantage of this approach is that it enables us to interpret rank as a random variable. Computing the mean of the rank samples then gives a more acurrate ranking of providers, in the sense that two providers can have ranks 1.0 and 1.3 instead of for example 1 and 2. This essentially leads to a better comparison of providers.

Using MCMC methods for the hierarchical model in Section 5 they obtained posterior estimates for means, credibility intervals, ranks, percentiles, etc. They obtained a reduced variability but they also shrank posterior means toward 1 and unfairly upgraded/downgraded certain providers. By including Figure 5 they also demonstrated the uncertainty of a provider being in the bottom 20%, i.e., how uncertain it is to claim that a particular provider is excellent or poor. They conclude with a remark that dealing with such diverse data and determining the optimal method is hard since one method can unfairly penalize large providers yet another can unfailry penalize small providers.

b. We first read the whole data file, available at [6], restrict it to all facilities in the state New York using the `subset` function and then generate the `dia` subset. To extract the correct values we looked in the corresponding data dictionary [7] which contains descriptions of variables used in the original data. We recompute the missing `rho_ml` values using equation (1) in the paper for the MLE estimator, and we discard the two providers for which `rho_ml` becomes `NA`.

```r
# reading the file, converting extracted values to numeric type
dfrdata <- read.csv("DFR_Data_FY2017.csv", header = TRUE)
dataNY <- subset(dfrdata, state == "NY")

dia <- subset(dataNY, select=c(dyy4_f, exdy4_f, deay4_f, smry4_f))
colnames(dia) <- c("yrs_at_risk", "mu", "y", "rho_ml")

dia$yrs_at_risk <- as.numeric(paste(dia$yrs_at_risk))
dia$mu <- as.numeric(paste(dia$mu))
dia$y <- as.numeric(paste(dia$y))
dia$rho_ml <- as.numeric(paste(dia$rho_ml))

# recalculating missing rhos
missingRho <- which(is.na(dia$rho_ml))
for (i in 1:length(missingRho)) {
  dia$rho_ml[missingRho[i]] <- dia$y[missingRho[i]]/dia$mu[missingRho[i]]
}
# eliminating NA values
dia <- dia[complete.cases(dia), ]
write.csv(dia, file = "dia.csv")
```

Since we are going to work with the generated `dia.csv` file, we read it again

Table 3: Percentage of providers in small, moderate and large MLE SMR category compared to the whole *dia* dataset.

Provider size	**Lower 25%**	**Middle 50%**	**Upper 25%**	**No. of providers**
< 25	50	12	38	26
25 - 50	23	54	23	35
50 - 100	16	49	34	79
100 - 150	16	58	25	55
$>= 150$	35	57	8	60

```
dia <- read.csv("dia.csv", row.names = 1, header = TRUE)
```

c. Table 3 shows the percentage of providers in the patient-year category (i.e., `yrs_at_risk` category) with MLE-estimated percentiles in the bottom 25%, middle 50% and upper 25%. We see that 50% of providers in the < 25 patient-years stratum had small SMRs, which is similar to the results in the paper. Middle size providers seem to be pretty balanced but there is a difference in the ≥ 150 patient-years stratum, where we get that 35% of providers had small SMRs whereas only 8% of providers had large SMRs. The differences we obtain (compared to the results in the paper) can be explain by looking at the number of providers in each stratum. We have a lot less providers since we restricted our analysis only to the New York facilities and hence we get different results. However, based on the Table 3 we can similarly conclude that small providers usually had very small SMRs whereas the percentage of providers with moderate SMRs is increasing with their size.

d. We first implement a function `createSMRmodel` which returns a MCMC list of samples.

```
# ------------------------------------------------------------- #
# createSMRmodel: returns mcmc.list of samples of SMRs for the
# Poisson-Normal-Gamma model from the paper.
#
# Parameters:
# sigma2 - squared variance of zeta hyperparameter
# a - the Gamma parameter for lambda hyperparameter
# nChains - desired number of parallel chains
# nSamples - desired number of samples
# burnin - desired size of the burnin
# ------------------------------------------------------------- #
createSMRmodel <- function(sigma2, a, nChains = 1, nSamples, burnin) {
  SMRmodel.string <- "model{
    for (i in 1:n) {
      par[i] <- mu[i] * rho[i]
      y[i] ~ dpois(par[i])
      rho[i] <- exp(theta[i])
      theta[i] ~ dnorm(zeta, lambda)
    }
    # Priors
    zeta ~ dnorm(0, 1/sigma2)
    lambda ~ dgamma(a, a)
  }"


  SMRmodel <- jags.model(textConnection(SMRmodel.string),
                         data = list('n' = nrow(dia), 'y' = dia$y,
                                     'mu' = dia$mu, 'sigma2' = sigma2,
                                     'a' = a),
                         n.chains = nChains, n.adapt = 100, quiet = TRUE)

  samples <- window(coda.samples(SMRmodel,
                                 variable.names = c('rho'),
                                 n.iter = nSamples + burnin),
                    start = burnin + 101)
  return(samples)
}
```

e. To investigate the convergence of the chains of the model we implement a function `runDiagnostic`.

```
# -------------------------------------------------------------- #
# runDiagnostic: returns gelman.diag object evaluated on the
# mcmc.list from the above JAGS model.
#
# Parameters:
# sigma2 - squared variance of zeta hyperparameter
# a - the Gamma parameter for lambda hyperparameter
# nChains - desired number of parallel chains
# nSamples - desired number of samples
# burnin - desired size of the burnin
# -------------------------------------------------------------- #
runDiagnostic <- function(sigma2, a, nChains = 1, nSamples = 2500, burnin) {
  mysamples <- createSMRmodel(sigma2, a, nChains, nSamples, burnin)
  # Attempt of using trace and gelman plots - cumbersome hence not used.
  # gr <- gelman.plot(mysamples)
  # shrink <- gr$shrink[1, , ]
  # psrf <- gelman.diag(window(mysamples, end=gp$last.iter[1]), multivariate=FALSE)
  # medMean <- mean(shrink[, 1] - psrf$psrf[, 1])
  # uppCIMean <- mean(shrink[, 1] - psrf$psrf[, 1])

  gd <- gelman.diag(mysamples)
  return(gd)
}
```

To decide whether the chains have converged or not we compare the point estimate of the multivariate potential scale reduction factor in the output of the `gelman.diag` function. We accept the convergence if `mpsrf` is approximately less than 1.05. We then run

```
# Testing the convergence of chains
testConv1 <- runDiagnostic(10^6, 10^(-4), nChains = 2,
                           nSamples = 2500, burnin = 1000)
```

```
testConv2 <- runDiagnostic(10^6, 10^(-4), nChains = 2,
                           nSamples = 2500, burnin = 5000)
testConv3 <- runDiagnostic(10^6, 10^(-4), nChains = 3,
                           nSamples = 2500, burnin = 1000)
testConv4 <- runDiagnostic(10^6, 10^(-4), nChains = 3,
                           nSamples = 2500, burnin=5000)
```

which return `mpsrf` equal to 1.1176047, 1.0823789, 1.073032, and 1.0472107, respectively. Therefore, we accept the samples that correspond to specifications of the variable `textConv4`.

```
# Samples with accepted convergence
rhoSamples <- createSMRmodel(10^6, 10^(-4), 3, 2500, 5000)
```

We could also look at the trace plots or gelman plots of the samples (the commented area of `runDiagnostic` function) but since ρ is 255-dimensional I think this approach a bit too cumbersome. We could however just extract a representative subset and analyze the convergence based on their plots.

f. We compute $\mathrm{E}(\rho_k|\mathbf{y}, \boldsymbol{\mu})$ and an 95% credibility interval for each facility k directly from the obtained samples. The results are sorted by their posterior means and shown in Figure 8 (which corresponds to the Figure 3 in the paper).

```
# plotting PM and their 95% CI
xaxis <- seq(1, nrow(dia))
posteriorMeans <- summary(rhoSamples)$statistics[, "Mean"]
posteriorCI <- summary(rhoSamples)$quantiles[, c(1, 5)]
posteriorMeansCI <- data.frame(posteriorMeans, posteriorCI)
posteriorMeansCI <- posteriorMeansCI[order(posteriorMeans), ]

plot(xaxis,posteriorMeansCI$posteriorMeans, ylim = c(0, 3),
     pch = 18, ylab = "PM SMR", xlab = "Provider") # means
arrows(xaxis, posteriorMeansCI$X2.5., xaxis, posteriorMeansCI$X97.5.,
       length = 0.05, angle = 90, code = 3) # CIs
abline(h = 1)
```

We use the R function `poisson.test` to compute confidence intervals of MLE SMRs. The results are again sorted but this time by their MLE SMRs and shown in Figure 9. The results are again comparable to the results in the paper. The posterior mean estimates are shrunken towards 1.0 compared to MLEs which means that the variability is improved. The credibility intervals are also significantly more narrow than the confidence intervals of the MLEs meaning that the precision has also been improved using sampling approach. We see from Figure 9 that the first few providers have large confidence intervals.

```
# plotting MLE and their 95% CI
mlLowCI <- c()
mlUppCI <- c()
for (i in 1:nrow(dia)) {
  PoissonTest <- poisson.test(dia$y[i], T = dia$mu[i], r = dia$rho_ml[i],
                              alternative = "two.sided")$conf.int
```

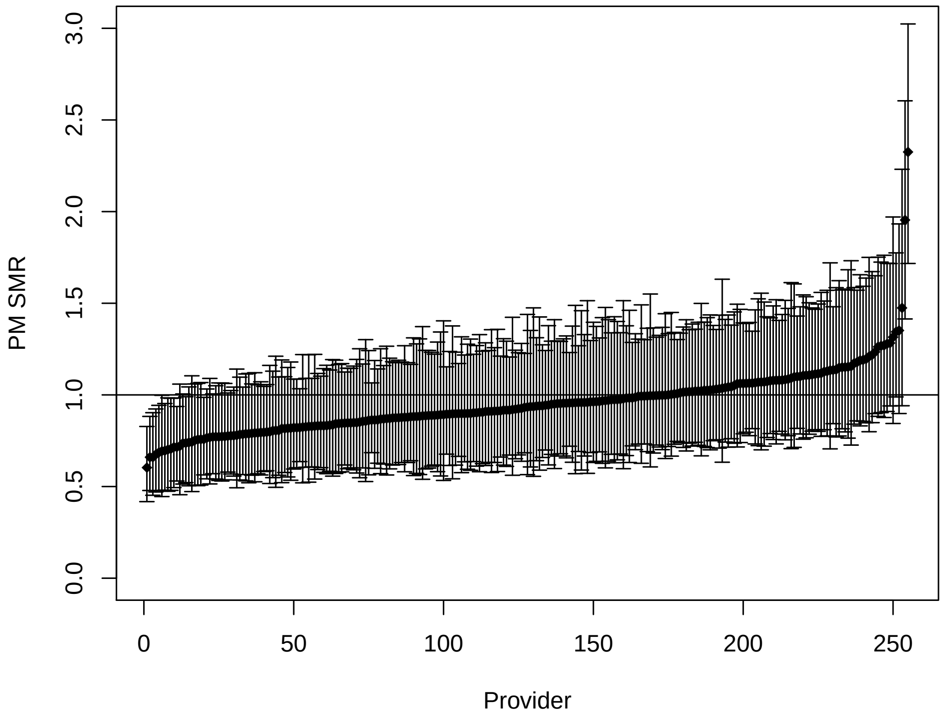

Figure 8: Posterior means of SMR and their 95% credibility intervals.

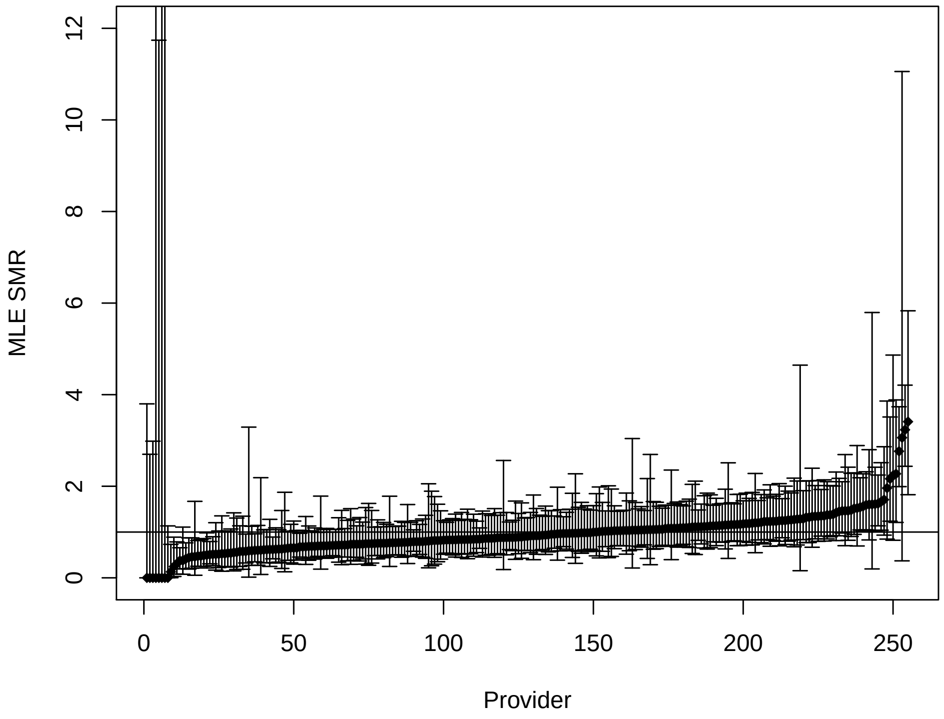

Figure 9: MLE SMRs and their 95% confidence intervals.

```r
  mlLowCI[i] <- PoissonTest[1]
  mlUppCI[i] <- PoissonTest[2]
}

mlMeansCI <- data.frame(mlMeans = dia$rho_ml, mlLowCI, mlUppCI)
mlMeansCI <- mlMeansCI[order(mlMeansCI$mlMeans), ]

plot(xaxis,mlMeansCI$mlMeans, pch = 18, ylab = "MLE SMR",
     xlab = "Provider", ylim = c(0, 12)) # means
arrows(xaxis, mlMeansCI$mlLowCI, xaxis, mlMeansCI$mlUppCI,
       length = 0.05, angle = 90, code = 3) # CIs
abline(h = 1)
```

We selected providers with `rho_ml` smaller than 0.01 and calculated that the mean of their `yrs_at_risk` values is barely 6.472625 which explains their large confidence intervals since they hardly treated any patients.

g. We first extract provider names from the original data and write it to another file to prevent reading the original (large) file again in the part (i).

```r
# Extracting provider names
dfrdata <- read.csv("DFR_Data_FY2017.csv", header = TRUE)
```

```r
dfrdataProviders <- subset(dfrdata, state == "NY", select = c(provname) )
dfrdataProviders$provname <- as.character(dfrdataProviders$provname)
colnames(dfrdataProviders) <- c("providers")
write.csv(dfrdataProviders, file = "provnames.csv")
```

We then find the row number of the provider *Westchester center for rental care* according to the original data using the `grep` function, and use that to extract the data from the `dia` dataset.

```r
dfrdataProviders <- read.csv("provnames.csv", row.names = 1,
                              header = TRUE)

# Extracting data for the Westchester provider
westIndex <- grep("WESTCHESTER CENTER FOR RENAL CARE",
                  dfrdataProviders$providers)
westProvider <- dia[westIndex, ]
```

Next we compute its MLE and its posterior mean from the above JAGS model

```r
# computing ML and PM of the Westchester provider
westML <- westProvider$rho_ml # MLE
westPM <- posteriorMeans[westIndex] # PM
```

and we obtain that and $\rho_{211}^{ml} = 3.40992$ and $\mathrm{E}(\rho_{211}|\mathbf{y}, \boldsymbol{\mu}) = 1.4744945$. To compute the posterior mean from the empirical Bayes analysis from Exercise 3, we first optimize `mloglik` for the current *dia* dataset. We then use the obtained parameters for the Gamma distribution of the hyperparameter λ and calculate the corresponding posterior mean from equation (0.1)

```r
# Computing EBPM for the Westchester provider
priorA <- optim(c(1,1), mloglik, y = dia$y, t = dia$mu,
                control = list(fnscale = -1))$par
westEBmean <- (priorA[1] + westProvider$y)*
  (priorA[2]/(westProvider$mu * priorA[2] + 1))
```

and we obtain that $\rho_{211}^{peeb} = 1.3883375$. The three different means are shown in Table 4. We notice that PM and PM from empirical Bayes model (PMEB) are quite close since the models used are quite similar. One could argue that more complicated model of Liu et al. with additional levels of hyperparameters does not contribute much to improving the value of the posterior mean. To explain why MLE is much greater than the other two we take a closer look of the Westchester Center for Renal care provider

```r
print(westProvider)

##      yrs_at_risk     mu  y  rho_ml
## 3604      15.871 3.8124 13 3.40992
```

and we see that they observed a lot more deaths than expected. Taking into account the patient-years we can conclude that most of their treated patients died which led to a more extreme value of MLE.

Table 4: MLE, PM and PMEB for the Westchester Center for Renal Care facility.

MLE	**PM**	**PMEB**
3.40992	1.474495	1.388338

h. Calculation of the posterior probability for the hypothesis $\rho_{211} > 1$ is straightforward from the obtained samples

```
# Posterior probability for the hypothesis
posteriorHyp <- mean(unlist(rhoSamples[ , westIndex]) > 1)
```

which gives the output $P(\rho_{211} > 1) = 0.9604$. Next, we analytically compute the prior probability for this hypothesis under the model from the paper. This very nice solution is Luca's idea.

$$\begin{aligned}
P(\rho_{211} > 1) = P(\theta_{211} > 0) &= \int_0^\infty p(\theta_{211})\,\mathrm{d}\theta_{211} \\
&= \int_{-\infty}^\infty \mathrm{d}\lambda \int_{-\infty}^\infty \mathrm{d}\zeta \int_0^\infty p(\theta_{211}|\zeta,\lambda)p(\zeta|\lambda)p(\lambda)\,\mathrm{d}\theta_{211} \\
&= \int_{-\infty}^\infty \mathrm{d}\lambda p(\lambda)\Bigg[\int_{-\infty}^0 \mathrm{d}\zeta p(\zeta)\int_0^\infty p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} + \\
&\quad + \int_0^\infty \mathrm{d}\zeta p(\zeta)\int_0^\infty p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211}\Bigg].
\end{aligned}$$

We now use the change of variable $\zeta \to -\zeta$ and reverse the integration bounds in the following integral

$$\int_{-\infty}^0 \mathrm{d}\zeta p(\zeta)\int_0^\infty p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} = \int_0^\infty \mathrm{d}\zeta p(-\zeta)\int_0^\infty p(\theta_{211}|-\zeta,\lambda)\,\mathrm{d}\theta_{211}.$$

Now we use the facts that $p(-\zeta) = p(\zeta)$ since $\zeta \sim N(0,\sigma^2)$, and hence symmetric, and that $p(\theta_{211}|-\zeta,\lambda) = p(-\theta_{211}|\zeta,\lambda)$ since $\theta_{211}|\zeta,\lambda \sim N(\zeta,\lambda^{-1})$ and we can therefore write $(\theta_{211}+\zeta)^2 = (-\theta_{211}-\zeta)^2$ in the exponent. We also change the variable $-\theta_{211} \to \theta_{211}$ and we get that

$$\begin{aligned}
\int_0^\infty \mathrm{d}\zeta p(-\zeta)\int_0^\infty p(\theta_{211}|-\zeta,\lambda)\,\mathrm{d}\theta_{211} &= \int_0^\infty \mathrm{d}\zeta p(\zeta)\int_0^\infty p(-\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} \\
&= \int_0^\infty \mathrm{d}\zeta p(\zeta)\int_0^{-\infty} p(\theta_{211}|\zeta,\lambda)(-\,\mathrm{d}\theta_{211}).
\end{aligned}$$

Putting all together and using the fact the the integrals of the densities are 1 we

Table 5: Prior and posterior probabilities for the hypothesis $\rho_{211} > 1$ under the two models.

	Prior rho_211 > 1	**Posterior rho_211 > 1**
Liu et al.	0.50	0.9540000
Empirical Bayes	0.39	0.9491128

get that

$$
\begin{aligned}
P(\rho_{211} > 1) = \int_{-\infty}^{\infty} \mathrm{d}\lambda p(\lambda) \Bigg[\int_{0}^{\infty} \mathrm{d}\zeta p(\zeta) \int_{-\infty}^{0} p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} + \\
+ \int_{0}^{\infty} \mathrm{d}\zeta p(\zeta) \int_{0}^{\infty} p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} \Bigg] \\
= \int_{-\infty}^{\infty} \mathrm{d}\lambda p(\lambda) \int_{0}^{\infty} \mathrm{d}\zeta p(\zeta) \int_{-\infty}^{\infty} p(\theta_{211}|\zeta,\lambda)\,\mathrm{d}\theta_{211} \\
= \frac{1}{2}
\end{aligned}
$$

since we integrate normal distribution with mean 0 on the positive real axis.

To compute the prior probability for this hypothesis under the empirical Bayes model we use hyperparameters values `priorA` obtained from the above optimization to calculate the following integrals

```r
# EB prior and posterior probabilities of the hypothesis
priorEB <- integrate(function(x)
  dgamma(x, shape = priorA[1], scale = priorA[2]), 1, Inf)

postEB <- integrate(function(x)
  dgamma(x, shape = (priorA[1] + westProvider$y),
         scale = (priorA[2]/(westProvider$mu * priorA[2] + 1))), 1, Inf)
```

Therefore, the prior probability for the hypothesis $\rho_k > 0$ is in this case `priorEB` $=$ 0.3882539. To compare the information update in both models we also computed the posterior probability of the hypothesis in the empirical Bayes model. The final results are stated in Table 5. We can see that the posterior probability under both models are approximately the same and both very high, meaning that information update is big and that again adding levels of hyperparameters in Liu et al. model does not make a difference at all. Of course, we need to make it clear that this is only true in this particular case using this particular data, and that we are not claiming that it is generally better to use more simple models.

i. We add the rank and percentile rank calculations in the JAGS model but we only sample the rank since only this is needed in further calculations.

```
# ------------------------------------------------------------ #
# createRankmodel: returns mcmc.list of samples of ranks for the
# Poisson-Normal-Gamma model from the paper.
#
# Parameters:
# sigma2 - squared variance of zeta hyperparameter
# a - the Gamma parameter for lambda hyperparameter
# nChains - desired number of parallel chains
# nSamples - desired number of samples
# burnin - desired size of the burnin
# ------------------------------------------------------------ #
createRankmodel <- function(sigma2, a, nChains = 1, nSamples, burnin) {
  Rankmodel.string <- "model{
    for (i in 1:n) {
      par[i] <- mu[i] * rho[i]
      y[i] ~ dpois(par[i])
      rho[i] <- exp(theta[i])
      theta[i] ~ dnorm(zeta, lambda)
    }

    r <- rank(rho[])
    pr <- r*100/n

    # Priors
    zeta ~ dnorm(0, 1/sigma2)
    lambda ~ dgamma(a, a)
  }"

  Rankmodel <- jags.model(textConnection(Rankmodel.string),
                         data = list('n' = nrow(dia), 'y' = dia$y,
                                     'mu' = dia$mu, 'sigma2'=sigma2,
                                     'a' = a),
                         n.chains = nChains, n.adapt = 100, quiet = TRUE)

  samples <- window(coda.samples(Rankmodel,
                                 variable.names = c('r'),
                                 n.iter = nSamples + burnin),
                    start = burnin + 101)
  return(samples)
}
```

For each facility we then compute the posterior probability of being in the top 3 dialysis facilities in state New York by running the updated JAGS program and extracting the facilities with the lowest three rank values.

```
# Computing top3 facilities from the updated JAGS program
rankSamples <- createRankmodel(10^6, 10^(-4), 3, 2500, 5000)

top3Probs <- c()
for (i in 1:255) {
  top3Probs[i] <- mean(unlist(rankSamples[, i]) <= 3)
}
# indices of top3 facilities in dia.csv
top3ProbIndex <- order(top3Probs, decreasing = TRUE)[1:3]

# determining the names of top3 facilities
```

Table 6: Top 3 facilities.

provname	Probability of being top3	MLE
NORTH SHORE UNIVERSITY HOSPITAL	0.2693333	0.37592
SOUTHERN WESTCHESTER DIALYSIS CTR	0.1368000	0.41518
BROADWAY DIALYSIS CENTER AT ELMHURST HOS	0.1180000	0.30657

```r
diaRows <- as.numeric(rownames(dia)[top3ProbIndex])
top3Names <- dfrdataProviders[paste(diaRows[1:3]), ]

# top3 posterior probabilities and MLEs
top3Prob <- top3Probs[top3ProbIndex]
top3MLE <- dia$rho_ml[top3ProbIndex]

# generating data.frame
top3 <- data.frame(top3Prob, top3MLE, stringsAsFactors = FALSE)
colnames(top3) <- c("Probability of being top3", "MLE")
topp3 <- cbind("provname" = top3Names, top3)
```

The results for the three facilities having the highest probability to belong to the top 3 are shown in Table 6.

As a patient, the best choice seems to be the first provider, *North Shore University Hospital*, because it has a very high posterior probability that it belongs to the top 3 facilities in New York compared to the other two facilities. However, one would also think that the third provider, *Broadway Dialysis Center at Elmhurst Hos*, is a good choice since it has the lowest MLE. We again take a closer look at those three providers

```r
print(cbind("provname" = top3Names, dia[top3ProbIndex, ]))

##                                       provname yrs_at_risk      mu  y  rho_ml
## 3636           NORTH SHORE UNIVERSITY HOSPITAL     230.992 31.9218 12 0.37592
## 3447         SOUTHERN WESTCHESTER DIALYSIS CTR     165.684 24.0857 10 0.41518
## 3535 BROADWAY DIALYSIS CENTER AT ELMHURST HOS     120.164 13.0477  4 0.30657
```

and we see that the *North Shore University Hospital* provider has a significantly higher patient-years than the *Broadway Dialysis Center at Elmhurst Hos* one which is why it is the prefer option.

References

[1] Daniel L Ensign and Vijay S Pande. "Bayesian detection of intensity changes in single molecule and molecular dynamics trajectories". In: *The Journal of Physical Chemistry B* 114.1 (2009), pp. 280–292.

[2] *Order of magnitude*. 2017. URL: https://en.wikipedia.org/wiki/Order_of_magnitudef.

[3] Brian Reich. *Poisson/Gamma model*. URL: http://www4.stat.ncsu.edu/~reich/st590/code/PoissonGammal.

[4] Michael Höhle. *Empirical Bayes*. URL: http://staff.math.su.se/hoehle/teaching/bayesmethht2017/empirical-handout.pdf.

[5] Jiannong Liu et al. "Methods for estimating and interpreting provider-specific standardized mortality ratios". In: *Health Services and Outcomes Research Methodology* 4.3 (2003), pp. 135–149.

[6] *FY2017 Dialysis Facility Report Data*. 2017. URL: https://www.dialysisdata.org/content/dialysis-facility-report-data.

[7] *FY 2017 Dialysis Facility Report Data Dictionary*. 2017. URL: https://www.dialysisdata.org/sites/default/files/content/dfr_data/FY2017_Data_Dictionary.pdf.