ISYE 6420: Bayesian Statistics **Spring 2020**

Midterm Exam

Instructor: Roshan Vengazhiyil, Brani Vidakovic *Name:* Nick Korbit, *gtID:* 903263968

Problem 1

We can think about the neurons structure as a Bayes net. Here we have six binary variables each of which can take the values of "Fire" and "Stop". We could tackle the problem the following ways:

- Analytical solution;
- Full enumeration;
- Simulation.

As the number of variable is small we'll start with the **full enumeration**. Six binary variables give us $2^6 = 64$ states that system can take. Since N_1 is given a stimulus, $P(N_1 = Fire) = 0.9$. Neurons N_2 to N_5 are conditionally dependent on the previous neuron. In case the previous neuron is fired $P(N_i = Fire|N_{i-1} = Fire) = 0.9$ otherwise $P(N_i = Fire|N_{i-1} = Stop) = 0.05$. Finally, the last neuron is fired in case either N_4 or N_5 is fired $P(N_6 = Fire|N_4 = Fire \vee N_5 = Fire) = 0.9$ and $P(N_6 = Fire|N_4 = Stop \wedge N_5 = Stop) = 0.05$.

We develop *run_full_enumeration()* procedure in *neurons.py*. The result of the enumeration is a table specifying probabilities for each of the system's state:

Figure 1: First 7 entries of the enumeration table.

	A	B	C	D	E	F	G
1	N1	N2	N3	N4	N5	N6	prob
2	True	True	True	True	True	True	0.531441
3	True	True	True	True	True	False	0.059049
4	True	True	True	True	False	True	0.059049
5	True	True	True	True	False	False	0.006561
6	True	True	True	False	True	True	0.059049
7	True	True	True	False	True	False	0.006561
8	True	True	True	False	False	True	0.0003645

To calculate the desired probabilities we need to sum up the states' probabilities where the neurons in question are set to *Fire* (True) or *Stop* (False). For example, to calculate $P(N_6 = Fire)$ we sum up probabilities from the table where $N6 = True$. Let's calculate the probabilities for parts a, b and c:

a) $P(N_6 = Fire) \approx 0.8038$

b) $P(N_6 = Fire|N_4 = Stop) \approx 0.1353$

c) $P(N_2 = Fire|N_6 = Stop) \approx 0.0946$

Note. We interpret "N_5 received stimulus" as "previous neuron – N_2 – is fired".

It's interesting to contrast full enumeration to **simulation**. Let's develop a *run_simulation()* procedure in *neurons.py* and compare results. Running the simulation for 100000 paths will yield the following results:

a) $P(N_6 = Fire) \approx 0.8039$

b) $P(N_6 = Fire|N_4 = Stop) \approx 0.1354$

c) $P(N_2 = Fire|N_6 = Stop) \approx 0.0941$

Problem 2

Since our likelihood is Gamma and our prior is Gamma, we have a conjugate prior problem. For our Gamma-Gamma case the posterior is also a Gamma, distributed as:

$$\lambda|\mathbf{X} \sim \mathcal{G}a\left(\alpha + nr, \beta + \sum_{i=1}^{n} X_i\right)$$

Substituting $r = 4$, $\alpha = 3$, $\beta = 5$, $n = 23$ and $\sum_{i=1}^{n} X_i = 50$ we obtain:

$$\lambda|\mathbf{X} \sim \mathcal{G}a\,(95, 171.148)$$

We could calculate Bayes estimator as a mean of the resulting Gamma distribution:

$$\hat{\lambda}_b^{mean} = \frac{\alpha}{\beta} = \frac{95}{171.148} \approx 0.555$$

Let's find the equitailed credible set. We start with the lower bound L:

$$\int_{-\infty}^{L} \pi(\lambda|x)d\lambda = \alpha/2$$

We can substitute the integral with the Gamma cdf:

$$F_X(L) = \alpha/2$$

We solve for L numerically using the Brent solver implemented via scipy *root_scalar* function.

$$F_X(L) - 0.05/2 = 0; L \approx 0.4491$$

Let's turn to the upper bound U:

$$\int_{-\infty}^{U} \pi(\lambda|x)d\lambda = 1 - \alpha/2$$

Alternatively

$$F_X(U) = 1 - \alpha/2$$

Solving for U gives us:

$$F_X(U) - 1 + 0.05/2 = 0; U \approx 0.6721$$

Resulting in the credible set $[0.4491, 0.6721]$ of length $l = 0.223$.

Now let's turn to the hypothesis $H_0 : \lambda \leq 0.5$. In order to find the probability of H_0 we integrate the posterior with respect to the parameter space:

$$p_0 = \int_{\Lambda_0} \pi(\lambda|x)d\lambda = \mathbb{P}^{\lambda|X}\left(H_0\right)$$

We use posterior $\mathcal{G}a\,(95, 171.148)$ cdf to calculate p_0. So that

$$p_0 = F_X(0.5) \approx 0.1669$$

Problem 3

Let Y_i be duration observations, μ_i – the mean and τ_i – the precision of the Normal distribution:

$$\begin{aligned} Y_1, Y_2, \ldots, Y_n &\sim \mathcal{N}(\mu, 1/\tau) \\ \mu &\sim \mathcal{N}(0.6, 1) \\ \tau &\sim \mathcal{G}a(20, 0.5) \end{aligned}$$

We now find the joint distribution $f(y, \mu, \tau)$;

$$f(y, \mu, \tau) = \left\{ \prod_{i=1}^{n} f(y_i|\mu, \tau) \right\} \pi(\mu)\pi(\tau) =$$

$$= \left\{ \prod_{i=1}^{n} \sqrt{\frac{\tau}{2\pi}} e^{-\tau(y_i-\mu)^2/2} \right\} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\mu-0.6)^2} \frac{0.5^{20}}{\Gamma(20)} \tau^{19} e^{-0.5\tau}$$

Removing constant terms will yield us:

$$f(y, \mu, \tau) \propto \left\{ \prod_{i=1}^{n} \sqrt{\tau} e^{-\tau(y_i-\mu)^2/2} \right\} e^{-\frac{1}{2}(\mu-0.6)^2} \tau^{19} e^{-0.5\tau}$$

$$f(y, \mu, \tau) \propto \tau^{\frac{n}{2}} e^{-\frac{\tau}{2}\sum_i^n (y_i-\mu)^2} e^{-\frac{1}{2}(\mu-0.6)^2} \tau^{19} e^{-0.5\tau}$$

We now find the full conditionals for μ and τ. If we look at the priors independently, we notice that both Normal-Normal and Normal-Gamma are conjugate priors. Although Normal-Normal-Gamma is not strictly a conjugate triplet we can recognize a semi-conjugate case. We derive the following posteriors for μ [2]:

$$\mu|\tau, y_{1:n} \sim \mathcal{N}\left(\frac{\tau_0\mu_0 + \tau\sum_{i=1}^{n} y_i}{\tau_0 + n\tau}, (\lambda_0 + n\tau)^{-1} \right)$$

$$p(\mu_1|\tau_1, y_1) = \mathcal{N}\left(\frac{0.6 + 27.95\tau}{1 + 43\tau}, (1 + 43\tau)^{-1} \right)$$

$$p(\mu_2|\tau_2, y_2) = \mathcal{N}\left(\frac{0.6 + 6.48\tau}{1 + 12\tau}, (1 + 12\tau)^{-1} \right)$$

Knowing that

$$\sum_i^n y = n\bar{y}$$

And posterior for τ:

$$\tau|\mu, y_{1:n} \sim \mathcal{G}a\left(a + \frac{n}{2}, b + \frac{1}{2}\sum (y_i - \mu)^2 \right)$$

$$p(\tau_1|\mu_1, y_1) = \mathcal{G}a\left(41.5, 0.5 + 0.5\left(1.3608 + 43(0.65 - \mu)^2\right)\right)$$

$$p(\tau_2|\mu_2, y_2) = \mathcal{G}a\left(26, 0.5 + 0.5\left(0.2156 + 12(0.54 - \mu)^2\right)\right)$$

Knowing that

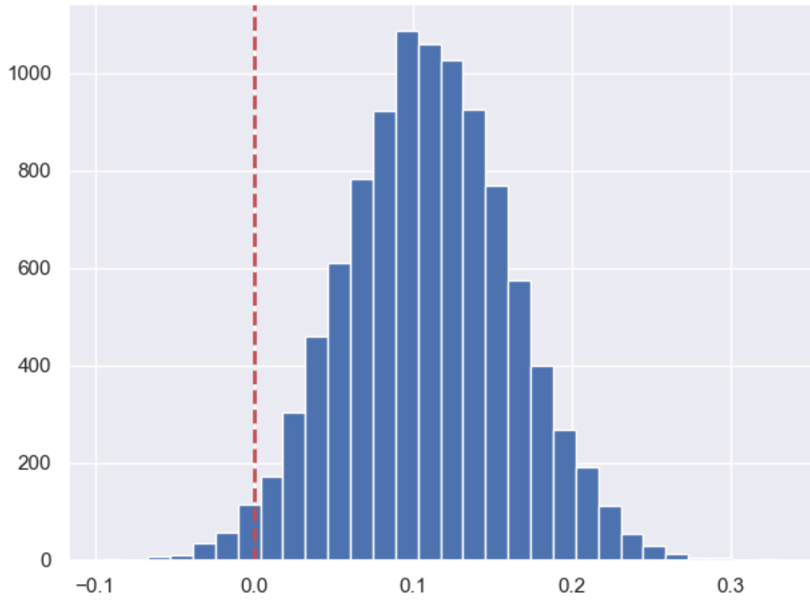
$$\sum_i^n (y_i - \mu)^2 = (n-1)s^2 + n(\bar{y} - \mu)^2$$

Let's develop a Gibbs sampling procedure for μ and τ:

1. For $i = 1, 2$:
2. Start with $\mu_0 = 0$, $\tau_0 = 1$
3. Sample
   - μ_{n+1} from $\mu|\tau, y_{1:n} \sim \mathcal{N}\left(\frac{\tau_0\mu_0 + \tau_n\sum_{i=1}^{n} y_i}{\tau_0 + n\tau_n}, (\lambda_0 + n\tau_n)^{-1} \right)$
   - τ_{n+1} from $\tau|\mu, y_{1:n} \sim \mathcal{G}a\left(a + \frac{n}{2}, b + \frac{1}{2}\sum (y_i - \mu_n)^2 \right)$
4. Set $n = n + 1$ and go to Step 3.

We now simulate 11000 samples for μ and τ for both species. After that we calculate the difference $\mu_1 - \mu_2$. Plotting the distribution of the resulting sequence will yield us:

Figure 2: Empirical distribution of $\mu_1 - \mu_2$.



The mean of the distribution is 0.1092 with 95% credible set of $[0.006; 0.2136]$. We also see the red dashed line ($\mu_1 - \mu_2 = 0$) is visually located far from the distribution peak 2. Based on the analysis we can reject the hypothesis $H_0 : \mu_1 = \mu_2$ and conclude that the length of call is indeed a discriminatory characteristic.

Note. In order to run the code that solves problems 1 to 3 sequentially, run the command *python runner.py*.

References

[1] Engineering Biostatistics: An Introduction using MATLAB and WinBUGS. Brani Vidakovic - Wiley Series in Probability and Statistics.

[2] Bayesian and Modern Statistics. Course material for STA 360/601, Jeff Miller, Spring 2015, Duke University. Chapter 6: Gibbs Sampling. `https://jwmi.github.io/BMS/chapter6-gibbs-sampling.pdf`