PM520–Homework 5

Chengliang Dong Thursday 23rd April, 2015

1 Part I

1.1 Introduction

In the current homework, we were asked to repeat the Urn model assignment to find the posterior distribution of the weight of the black ball given that we observe just one color in an urn containing 5 balls. We sample the mutation rate θ' from $\xi \sim exp(\lambda)$ using importance Bayesian sampling methods. In statistics, importance sampling is a general technique for estimating properties of a particular distribution, while only having samples generated from a different distribution from the distribution of interest. It is related to umbrella sampling in computational physics. Depending on the application, the term may refer to the process of sampling from this alternative distribution, the process of inference, or both.
The algorithm is as follows. Suppose we have parameter θ, prior π and importance sampling distribution ξ and observed data D.
1. Sample θ' from ξ and simulate data D' using θ'.
2. Accept θ' if $D' = D$. Otherwise reject θ' and return to 1.
3. Add a mass of $\pi(\theta)/\xi(\theta)$ to the posterior at θ'.
4. Rejection method: sample from prior $\pi(\theta)$ and construct posterior.
On the other hand, we need to compare the result with that from Rejection method that just samples directly from a $Uniform[0, 20]$ distribution.
The algorithm is as follows. Suppose we have parameter θ and observed data D.
1. Sample θ' from $Uniform[0, 20]$ and simulate data D' using θ'.
2. Accept θ' if $D' = D$. Otherwise reject θ' and return to 1.
3. Construct posterior based on these sampled θ'.
Kernel density estimation is a non-parametric way to estimate the probability density function of a random variable. This method with Gaussian kernel is used in the current homework to estimate the posterior distribution of θ.

1.2 Method

1.2.1 Bayesian importance sampling

Using the algorithm described above, we applied the importance sampling distribution ξ $exp(\lambda)$ (note that here we investigated a group of λ, including 0.0000001,0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1 and 10) and observed data D, which is "just 1 color in an urn containing 5 balls" and simulated 10000 acceptance to see how many simulations is needed to generate these acceptance and what does the posterior distribution look like from these 10000 acceptance.

1.2.2 Rejection sampling

Using the algorithm described above, we applied the observed data D, which is "just 1 color in an urn containing 5 balls" and also simulated 10000 acceptance to see how many simulations is needed to generate these acceptance and what does the posterior distribution look like from these 10000 acceptance.

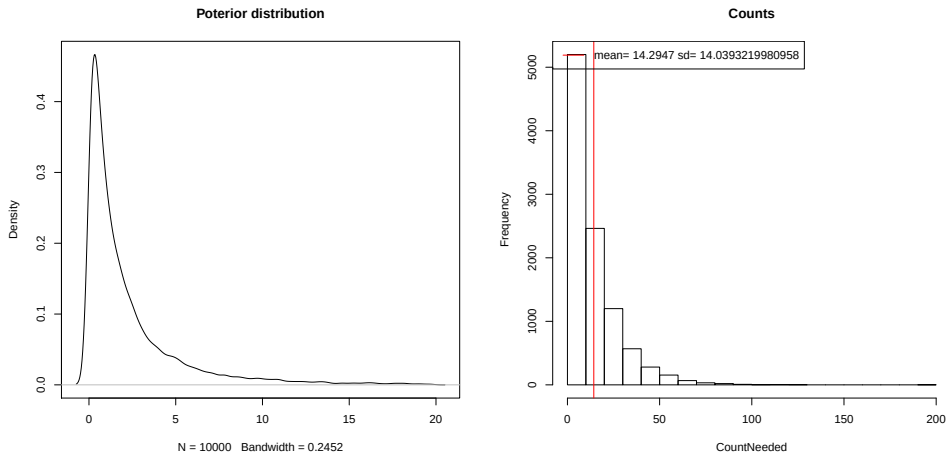
1.2.3 Kernel smoothing

Kernel smoothing method with Gaussian kernel was used to estimate the posterior distribution in both Bayesian importance sampling cases and Rejection sampling cases. Note that in Bayesian importance sampling methods, weight (mass for each estimated θ) was also taken into account to generate smoothed density plot. I used bandwidth of 0.9 times the minimum of the standard deviation and the interquartile range divided by 1.34 times the sample size to the negative one-fifth power derived from Silverman's 'rule of thumb' from Silverman, B. W. (1986) Density Estimation. London: Chapman and Hall.

1.3 Result

1.3.1 Rejection sampling

From posterior distribution of θ, we can see that the value of θ smoothly spread from 0 to 20 with a peak at around 2 at density about 0.45. It takes on average 14.3 times (sd = 14.03) to generate each acceptance from all of these 10000 acceptance.



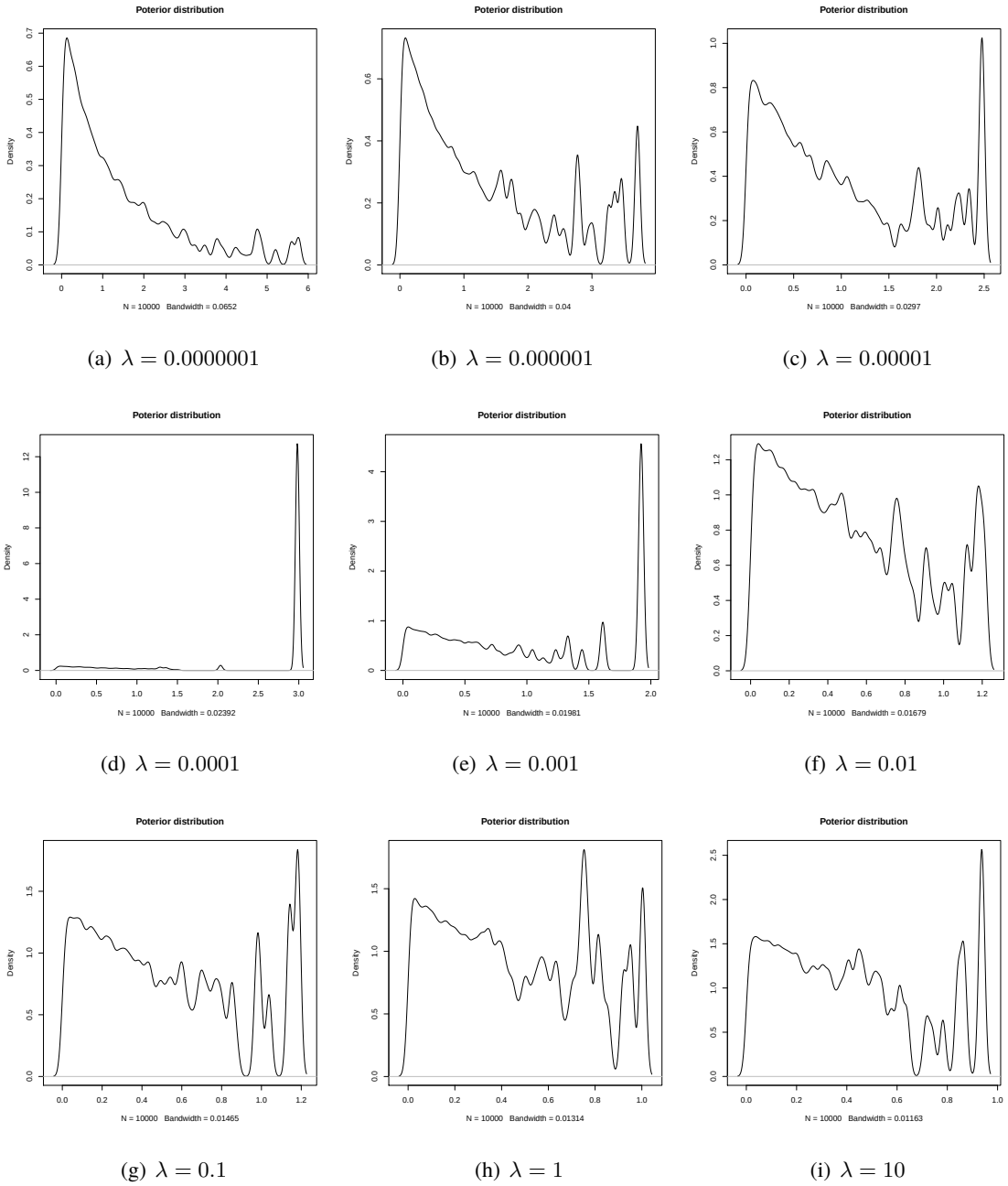
(a) Density (b) Count

Figure 1: Result for rejection sampling

1.3.2 Importance sampling

From posterior distribution of θ, we can see that the smaller the λ in the importance sampling distribution, the more spread out the value of θ is, and the more it looks like the posterior distribution generated from Rejection sampling method. For example, when $\lambda = 0.0000001$, θ ranges from -0.1956 to 5.9588 with

mean of 2.8816 and the peak occurs around 0.2 at density about 0.68. At relatively larger λ, the density of θ around 0 to 1 decreases. For example, when $\lambda = 0.0001$, there was almost no mass around 0 to 1. As λ increases, the less smooth the density plot looks like, the more spikes occurs in the density plot and less spread the θ is. For example, when $\lambda = 10$, θ ranges from -0.03487 to 0.97196 with mean of 0.46854 and a peak occurs around 0.9 at density about 2.5. From the distribution for count needed



(a) $\lambda = 0.0000001$ (b) $\lambda = 0.000001$ (c) $\lambda = 0.00001$

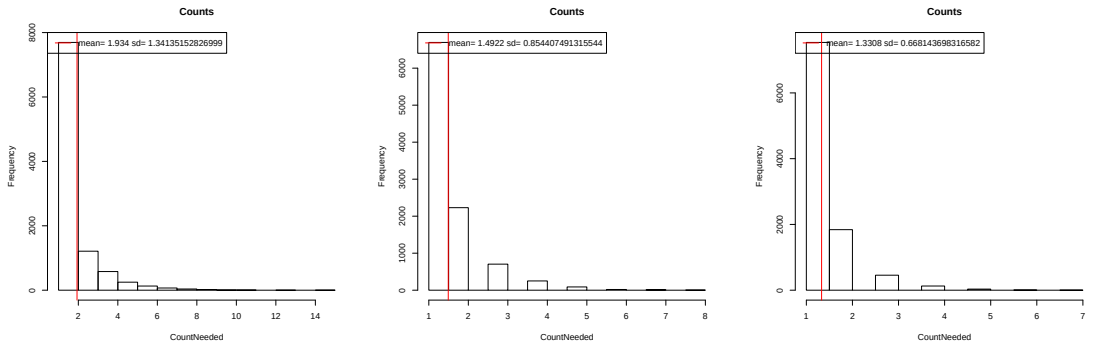
(d) $\lambda = 0.0001$ (e) $\lambda = 0.001$ (f) $\lambda = 0.01$

(g) $\lambda = 0.1$ (h) $\lambda = 1$ (i) $\lambda = 10$

Figure 2: Density generated from importance sampling

to generate 10000 acceptance, we can see that in general Importance sampling is more efficient than Rejection sampling method as it takes less runs to generate each acceptance. Indeed, the mean count needed to generate 10000 acceptance is less than 2, which is far less than that of Rejection sampling (count = 14.3). Moreover, the higher the λ, the less count it is needed to generate 10000 acceptance. This is because $1/\lambda$ is the expected value of exponential random variable, therefore, the higher the λ, the

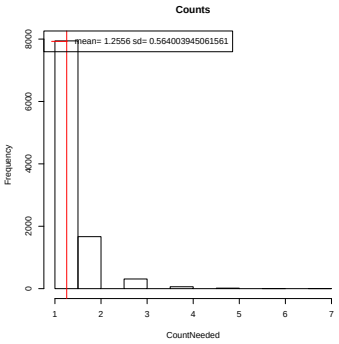
higher the tendency of generating smaller weight for mutation ball (black ball), therefore, the more likely to generate 5 balls of the same color because we tend to not get black ball in each draw. Vice versa.



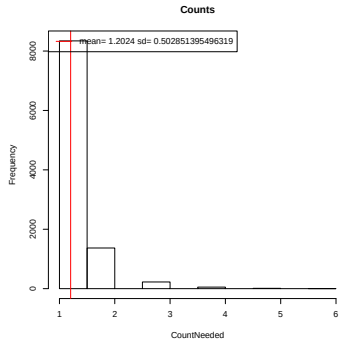
(a) $\lambda = 0.0000001$

(b) $\lambda = 0.000001$

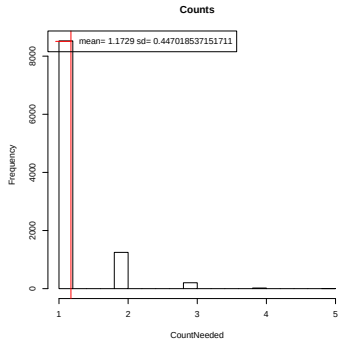
(c) $\lambda = 0.00001$



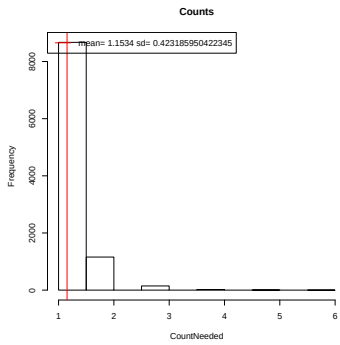
(d) $\lambda = 0.0001$



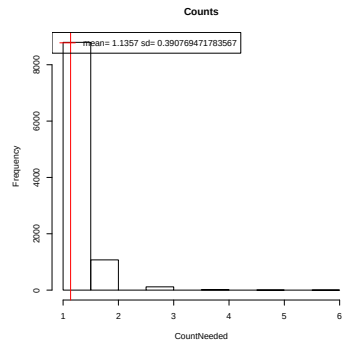
(e) $\lambda = 0.001$



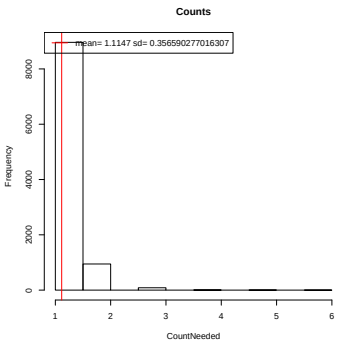
(f) $\lambda = 0.01$



(g) $\lambda = 0.1$



(h) $\lambda = 1$



(i) $\lambda = 10$

Figure 3: Count to generate 10000 acceptance from importance sampling

1.4 Conclusion

Bayesian importance sampling method is more efficient than Rejection sampling method and that different choices of importance sampling distribution parameters could effect the efficiency as well as the posterior distribution.

2 Part II

2.1 Introduction

Nucleic acid sequencing is a method for determining the exact order of nucleotides present in a given DNA or RNA molecule. In the past decade, the use of nucleic acid sequencing has increased exponentially as the ability to sequence has become accessible to research and clinical labs all over the world. The first major foray into DNA sequencing was the Human Genome Project, a 3 billion, 13-year-long endeavor, completed in 2003. The Human Genome Project was accomplished with first-generation sequencing, known as Sanger sequencing. Sanger sequencing (the chain-termination method), developed in 1975 by Edward Sanger, was considered the gold standard for nucleic acid sequencing for the subsequent two and a half decades.

In the process of sequencing, to increase efficiency, people always sequence 96 samples in a single run, with different barcodes (a sequence of nucleotides) marking different individuals. In the current homework, we are going to use Monte Carlo simulation (same technique as in toy homework 1) to calculate the minimal length of barcode needed to identify these 96 sample with the probability of encountering the same barcode for two different samples less than 0.05. We consider two cases, case I, we treat AA, AB, BB, BA four alleles distinctively and case II, we treat AB and BA as same allele, resulting in AA, AB(or BA), BB two distinctive alleles.

2.2 Method

In both cases, we started from length of 1 and steadily increase length till the probability of randomly drawing 96 barcodes from this given length with at least 1 same barcode from 10000 trails is less or equal to 0.05. In case I, we generate 4 different alleles at each position with same weight, and in case II, we generate 3 different alleles at each position with weight of 1,1 and 2.

2.3 Result

From our result, we can see that for case I, we need at least length of 9 to ensure the probability of countering at least 1 same barcode for these 96 samples to be less 0.05 and in case II, we need length of at least 12.

Table 1: Probability for 10000 simulations in both cases

Case	I	II
L=1	1	1
L=2	1	1
L=3	1	1
L=4	1	1
L=5	0.9901	1
L=6	0.6749	1
L=7	0.2497	0.9891
L=8	0.0643	0.8283
L=9	0.0164	0.4714
L=10	0.0038	0.2183
L=11	0.00006	0.0893
L=12	0.00005	0.0349
L=13	0.00001	0.0148

2.4 Conclusion

We need at least length of 9 to ensure the probability of countering at least 1 same barcode for these 96 samples to be less 0.05 in case I and length of 12 in case II.