Computational Statistics
Monte-Carlo methods

georges.tod@oultook.com

May 2020

1 Introduction

In this report, some Monte-Carlo based methods are illustrated for the numerical estimation of integrals and to sample from *less friendly* distributions.

2 Numerical estimation of an integral

We start by an application arising in Bayesian inference where the following ratio might need to be evaluated,

$$r_m = \frac{\displaystyle\int_{-\infty}^{+\infty} x^m \cdot \frac{1}{b\pi[1+(\frac{x-a}{b})^2]} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} dx}{\displaystyle\int_{-\infty}^{+\infty} \frac{1}{b\pi[1+(\frac{x-a}{b})^2]} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} dx}$$

We will discuss how to approximate r_1 when choosing for the sake of simplicity, $a = 0, b = 1, \mu = 0$ and $\sigma = 1/\sqrt{2}$. Our focus will be on r_1,

$$r_1 = \frac{\displaystyle\int_{-\infty}^{+\infty} x \cdot \frac{1}{\pi(1+x^2)} \cdot \frac{1}{\sqrt{\pi}} e^{-x^2} dx}{\displaystyle\int_{-\infty}^{+\infty} \frac{1}{\pi(1+x^2)} \cdot \frac{1}{\sqrt{\pi}} e^{-x^2} dx} \tag{1}$$

for which there is no easily accessible analytical solution.

2.1 Rejection sampling

From theory we know, we can generate data X from a distribution proportional to,

$$f_X(x) = K \cdot \frac{1}{1+x^2} \cdot g_X(x)$$

where $g_X(x)$ is the normal density function and it then holds that $r_1 = E(X)$. By generating $X_1, ..., X_n$, we introduce the estimator,

$$\hat{\theta}_1 = \frac{1}{n}\sum_{i=1}^{n} X_i = \bar{X}$$

By the law of large numbers,

$$\hat{\theta}_1 = \frac{1}{n}\sum_{i=1}^{n} X_i \to E(X) = r_1$$

Therefore $\hat{\theta}_1$ is asymptotically unbiased.

2.2 Basic Monte-Carlo integration

To use the basic Monte-Carlo integration, we call decompose the problem of estimating r_1 into the estimation of its numerator and its denominator. Two estimators of r_1 can then be introduced.

2.2.1 using a normal distribution

By generating normal data from $Z \sim N\Big(0, \frac{1}{2}\Big)$, a second estimator is,

$$\hat{\theta}_2 = \frac{\overline{Z \cdot I_Z}}{\overline{I_Z}}$$

where $I_Z = h(Z)$ and $h(x) = \frac{1}{\pi(1+x^2)}$. Applying the law of large numbers to the numerator leads to,

$$\overline{Z \cdot I_Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i \cdot h(Z_i) \to E(Z \cdot h(Z)) = \int_{-\infty}^{+\infty} x \cdot \frac{1}{\pi(1+x^2)} \cdot \frac{1}{\sqrt{\pi}} e^{-x^2} dx$$

Similarly,

$$\overline{I_Z} \to E(h(Z)) = \int_{-\infty}^{+\infty} \frac{1}{\pi(1+x^2)} \cdot \frac{1}{\sqrt{\pi}} e^{-x^2} dx$$

So as soon as both $\overline{Z \cdot I_Z}$ and $\overline{I_Z}$ are finite and non null, for $n \to +\infty$,

$$\hat{\theta}_2 \to r_1$$

therefore $\hat{\theta}_2$ is asymptotically unbiased.

2.2.2 using a Cauchy distribution

In this case, by generating data from $T \sim \text{Cauchy}(0, 1)$, a third estimator is,

$$\hat{\theta}_3 = \frac{\overline{T \cdot I_T}}{\overline{I_T}}$$

where $I_T = g(T)$ and $g(x) = \frac{1}{\sqrt{\pi}} e^{-x^2}$. Invoking the law of large numbers and the steps of the previous subsection allows to conclude $\hat{\theta}_3$ is also an asymptotically unbiased estimator of r_1.

2.3 Numerical experiments

On figure 1, some numerical estimations of r_1 are computed for varying n. For each sample size n, the experiment is repeated 100 times. A more detailed analysis of the variance is proposed on figure 2 where a Levene's test was performed.

3 Markov Chain Monte Carlo sampling

This section illustrates the application of the *Metropolis-Hastings (MH) algorithm* to generate samples from a Gamma distribution with parameters α and λ,

$$f_X(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x} \tag{2}$$

where $x > 0$, $\alpha > 0$, $\lambda > 0$ and $\Gamma(\alpha) = \int_0^\infty y^{\alpha-1} e^{-y} dy$.

3.1 The Metropolis-Hastings algorithm

The MH algorithm [1] is a specific MCMC method that works as follows. Let $q(y|x)$ be an arbitrary, friendly distribution (i.e., we know how to sample from $q(y|x)$). The conditional density $q(y|x)$ is called the *proposal distribution*. The MH algorithm creates a sequence of observations $X_1, ..., X_n$ as follows,
Choose X_1 arbitrarily, and suppose we have generated, $X_2, X_3, ..., X_i$. To generate X_{i+1} do the following,

1. generate a proposal candidate $Y \sim q(y|X_i)$
2. evaluate $r = r(X_i, Y)$ where,

$$r(x, y) = \min\left\{\frac{f(y) \cdot q(x|y)}{f(x) \cdot q(y|x)}, 1\right\} \tag{3}$$

3. set

$$X_{i+1} = \begin{cases} Y & \text{with probability } r \\ X_i & \text{with probability } 1 - r \end{cases} \tag{4}$$

A very simple way to execute step (3) is to generate $U \sim (0, 1)$. If $U < r$ set $X_i + 1 = Y$ otherwise set $X_{i+1} = X_i$.

[1] This section comes from Wasserman, L. (2013). All of statistics: a concise course in statistical inference. Springer Science & Business Media.

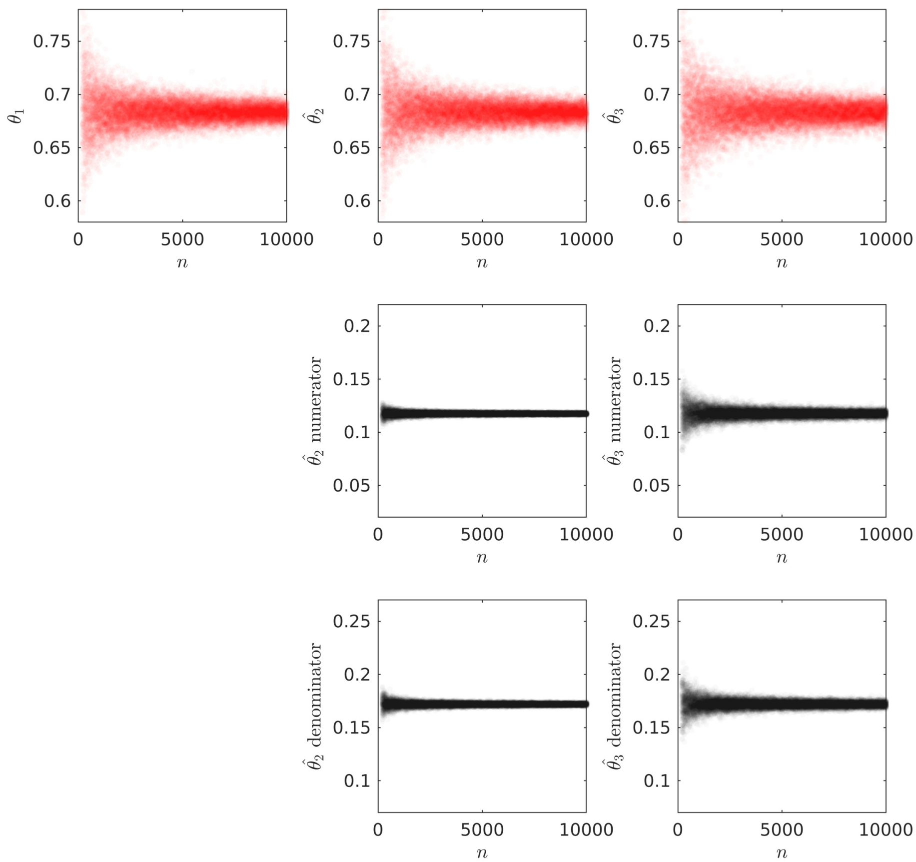

Figure 1: Numerical estimations of r_1, $\hat{\theta}_1$ by rejection sampling, $\hat{\theta}_2$ by Monte-Carlo method from Normal distribution and $\hat{\theta}_3$ by Monte-Carlo method from Cauchy distribution. For each sample size n, the experiment is repeated 100 times. In this particular case, $\hat{\theta}_1$ and $\hat{\theta}_2$ seem to give quite similar results while $\hat{\theta}_3$ seems to converge at a slower rate. Interestingly, there is a large difference in the speed of convergence in the case one would only need to estimate the numerator or the denominator: see the four subplots at the bottom. It seems to be a much better idea to sample from a Normal than a Cauchy distribution (in this particular case).

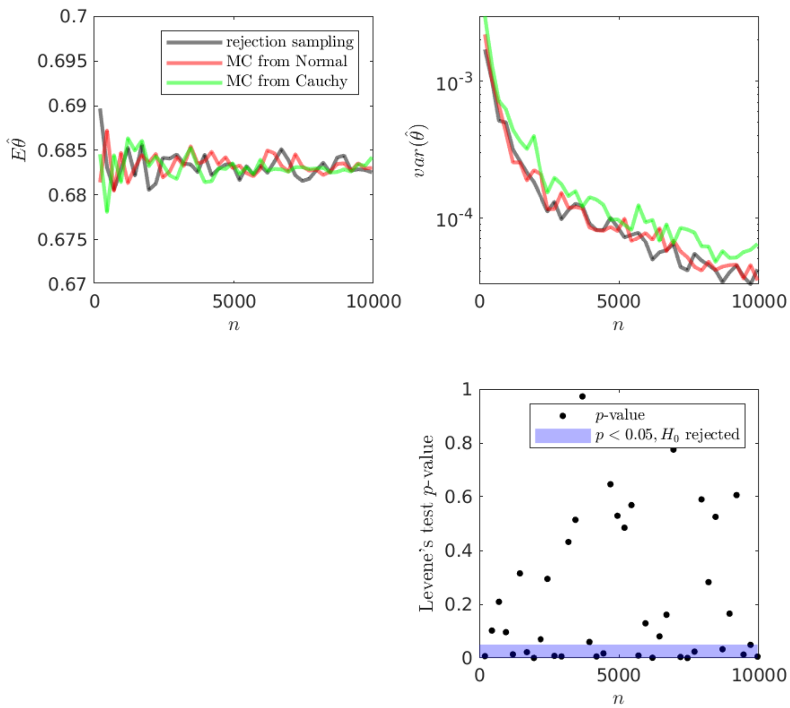

Figure 2: Here, estimates of bias and variance of the estimators are computed by repeating the experiment 100 times for each sample size n. The three estimators variance converges very quickly to zero for small sample sizes but then slows down. Even though it seems the rate of convergence of $var(\hat{\theta}_3)$ is slower, a Levene's test shows the points contained in the rejection zone of H_0 is very small and therefore, the variances are more often significantly equal (up to $\alpha = 0.05$).

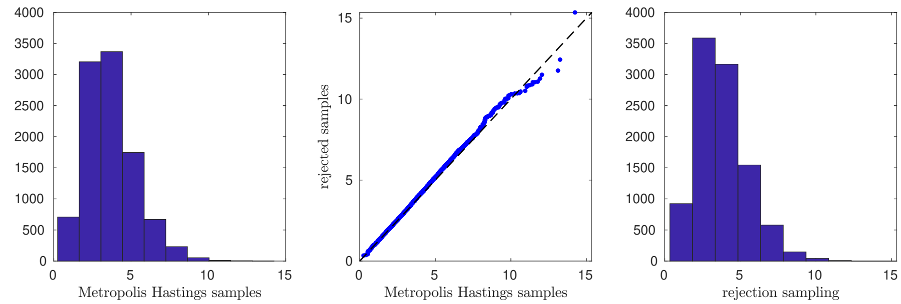

Figure 3: Metropolis-Hastings algorithm versus rejection sampling - sample size $n = 10000, \lambda = 1.4, \alpha = 5.3$. The distributions agree quite well except at the tails of the distribution: close to 0 and above 10 in this case.

3.2 Some implementation remarks

It is proposed to use,

$$q(x|y) = \frac{x^{r-1}\lambda^r e^{-\lambda x}}{(r-1)!}$$

where r is the greatest integer less than or equal to α. Therefore the ratio in equation 3 becomes,

$$\frac{f_X(y)\cdot q(x|y)}{f_X(x)\cdot q(y|x)} = \frac{y^{\alpha-1}e^{-\lambda y}}{x^{\alpha-1}e^{-\lambda x}} \cdot \frac{x^{r-1}e^{-\lambda x}}{y^{r-1}e^{-\lambda y}} = \left(\frac{y}{x}\right)^{\alpha-r}$$

Explaining the 3^{rd} line of the pseudo-code requested. The 4^{th} and 5^{th} being explained by equation 4. To explain the first two lines of pseudo-code, we can invoke that at that point r is an integer and that as in slide 40, X can be rewritten as a sum of samples from an exponential distribution.

3.3 A numerical experiment

Finally, sampling results are reported on figure 3 in which we compare both MCMC sampling and rejection sampling. A drawback of the HM algorithm is one needs to make a good choice of the proposal distribution q: this might not always be straightforward. In addition, it is reported in literature[2] that rejection sampling does not perform well in higher dimensions and that MCMC sampling overcomes this limitation.

[2] Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. Machine learning, 50(1-2), 5-43.