Computational Stochastic Processes
Problem Sheet 1

U. Vaes

with problems from G. Pavliotis, A. Duncan and S. Gomes

February 2020

You are free to return a selection of your work to me for marking. This is entirely optional and the mark will not count for assessment.

Generating non-uniform random variables

Problem 1 (Generalized Bernoulli Distribution)**.** Suppose X is a discrete valued random variable taking values i with probability p_i for $i \in \{1, \ldots, k\}$ where $\sum_{i=1}^{k} p_i = 1$.

1. Write down the CDF $F(x)$ for the probability distribution of this r.v..
2. Write down an expression for the generalised inverse of the CDF $F(x)$.
3. Use the inverse transform method to derive an algorithm to sample from this distribution.
4. Implement a sampler based on this scheme using a programming language of your choice.
5. For $k = 4$, $p_1, p_2, p_3, p_4 = 0.125, 0.125, 0.375, 0.375$ generate $N = 10^3$ samples and generate a normalized histogram from this sample to verify that each value is generated with the correct probability.

Problem 2 (Sample from Gamma(k, λ) distribution)**.** When $k \in \mathbb{N}$, it is known that

$$X_1 + \ldots + X_k \sim \text{Gamma}(k, \lambda),$$

where $X_1, \ldots, X_k$ are iid Exp(λ) distributed random variables.

1. Based on this observation, write a scheme to generate Gamma(k, λ) distributed samples, where $k \in \mathbb{N}$.
2. Suppose now that $k \in \mathbb{R}_{\geq 0}$. We wish to implement a rejection sampler for Gamma(k, λ) using proposal density of the form

$$g(x) = \frac{\lambda_0^{k_0}}{\Gamma(k_0)} x^{k_0 - 1} e^{-\lambda_0 x},$$

where $k_0 = \lfloor k \rfloor$, and $\lambda_0 > 0$. Calculate the upper bound $M = \sup_{x \geq 0} f(x)/g(x)$, and show that necessarily $\lambda < \lambda_0$ and

$$M = \frac{\lambda^k}{\Gamma(k)} \frac{\Gamma(k_0)}{\lambda_0^{k_0}} \left(\frac{k - k_0}{\lambda - \lambda_0} \right)^{k-k_0} e^{-(k-k_0)}.$$

3. Optimise over λ_0 to minimise M.

4. Implement a rejection sampling scheme to sample from Gamma distribution. Implement also a naive rejection sampler based on a proposal distributed according to the standard Cauchy distribution, and compare the performance of both.

Monte Carlo simulation

Problem 3 (Monte Carlo Simulation)**.** We want to estimate the area inside the Batman curve via Monte Carlo simulation. To this end, let (X, Y) be a uniformly distributed random variable in the rectangle $[-L_x, L_x] \times [-L_y, L_y]$, with $L_x = 7.25$ and $L_y = 4$, and let $f(\cdot, \cdot)$ denote the indicator function of the surface inside the Batman curve. For the student's convenience, this function is implemented in *Python* below:

```
def batman_indicator(x, y):

    # We'll initialize at one and remove parts one by one
    result = np.ones(x.shape)

    # Ellipse
    ellipse = (x/7)**2 + (y/3)**2 - 1 >= 0
    result[np.where(ellipse)] = 0

    # Bottom curve on [-3, 3]
    bottom = (abs(x) < 4) * \
             (y <= abs(x/2) - ((3*np.sqrt(33)-7)/112)*x**2 - 3
              + np.sqrt(np.maximum(0, 1-(abs(abs(x)-2) - 1)**2)))
    result[np.where(bottom)] = 0

    # Top curve
    top = (abs(x) > .75) * (abs(x) < 1) * (y > 9 - 8*abs(x)) \
          + (abs(x) > .5) * (abs(x) < .75) * (y > 3*abs(x) + .75) \
          + (abs(x) < .5) * (y > 2.25) \
          + (abs(x) > 1) * (abs(x) < 3) * \
            (y > (6*np.sqrt(10)/7+(1.5-.5*abs(x))-(6*np.sqrt(10)/14)*\
                 np.sqrt(np.maximum(0, 4-(abs(x)-1)**2))))
    result[np.where(top)] = 0
    return result
```

The exact area of the Batman sign is given below:

```
# Exact area
I = (955/48) - (2/7) * (2*np.sqrt(33) + 7*np.pi + 3*np.sqrt(10) * (np.pi - 1)) \
    + 21 * (np.arccos(3/7) + np.arccos(4/7))
```

1. Using a sample of size $n = 1000$, generate a 95% confidence interval for the area of the Batman sign, $I := 4\, L_x\, L_y\, \mathbb{E}[f(X, Y)]$ based on

   (a) Chebychev's inequality;
   (b) The central limit theorem (CLT);
   (c) Bikelis' theorem.

2. Perform this previous simulation 1000 times independently, and measure how many of the reported confidence intervals actually contained I.

3. Use Hoeffding's inequality to derive $100(1-\alpha)\%$-confidence intervals and compare with the ones obtained via Chebychev and the CLT.

   Theorem 1. *Let $Z_1, \dots, Z_n$ be iid random variables supported within the interval $[0,1]$ with mean μ. Then*
   $$\mathbb{P}\left[|S_n - \mu| \geq a\right] \leq 2e^{-2na^2},$$
   where $S_n = n^{-1}\sum_{i=1}^{n} Z_i$ and $a > 0$.

4. Find a control variate that enables a reduction of the variance of the estimator.

Problem 4 (Density estimation using Histograms)**.** A very common problem in computational stochastic methods is the estimation of density, i.e. given a stream of iid samples of some rv X we wish to accurately recover the density $p(x)$ of X. The MC approach to density estimation is to express the density $p(x)$ as a limit of expectations of random variables, namely

$$p(x) = \lim_{h\to 0} \frac{\mathbb{E}\mathbf{1}\left[x < X < x+h\right]}{h}.$$

Thus, for small h, we use the following MC estimator

$$\hat{p}_n(x) = \frac{1}{hn}\sum_{i=1}^{n} \mathbf{1}\left[x < X_i < x+h\right]$$

1. Assume that $p \in C^1(\mathbb{R})$, use Taylor's theorem to show that $\hat{p}_n(x)$ is asymptotically biased (and thus biased). Show that the bias goes to zero as $h \to 0$.

2. Compute the variance of the estimator $\hat{p}(x)$. Show that $\mathrm{Var}[\hat{p}_n(x)] \to \infty$ as $h \to 0$.

Taking $h \to 0$, the bias goes to zero, while the variance blows up. This suggests that a good choice of h must involve some kind of "trade-off" between variance and bias.

3. Assuming again that $p \in C^1(\mathbb{R})$. Write an expression for the MSE of $\hat{p}_n$ which is correct to $o(h)$.

4. Find the value of h which minimises the MSE. Conclude that MSE is minimized taking $h = O(N^{-1/3})$, in which case the MSE goes to zero with rate $N^{-2/3}$.

Variance reduction techniques

Problem 5 (Importance Sampling)**.** Consider the problem of estimating the moments of the distribution

$$p(x) = \frac{1}{2}e^{-|x|},$$

called the double exponential density. The CDF of this function is

$$F(x) = \frac{1}{2}e^x \mathbf{1}[x \le 0] + \frac{1}{2}(1 - e^{-x}/2)\mathbf{1}[x > 0],$$

which is a piecewise function and difficult to invert. Indeed, we cannot "easily" sample from this distribution. Suppose we wish to estimate the second moment of the distribution $\mathbb{E}[X^2]$.

1. Using importance sampling distribution $\mathcal{N}(0, 4)$ construct an importance sampler for computing $\mathbb{E}[X^2]$.

2. Implement this sampler in a programming language of your choice, and generate 10^5 samples, and compute the mean. The true value of this expectation should be 2.

3. Can you use the expression for the variance of the importance sampler (or some other method) to find a better choice of σ^2 for a proposal distribution $\mathcal{N}(0, \sigma^2)$? Implement this scheme and compare the performance computationally.

In machine learning one often needs to compute expectations with respect to the *Gumbel distribution*

$$p(x) = \exp(x - \exp(x)).$$

4. Show that $\mathbb{E}[\exp(X)] < \infty$, where $X \sim p$.

5. Using a standard Gaussian importance distribution, implement a regular importance sampler approximating $\mathbb{E}[\exp(X)]$ in a programming language of your choice.

6. Similarly implement a self-normalized version of the importance sampler.

7. Compare the performance of both by computing the variance of both estimators, approximated over 10^3 independent runs.

Problem 6 (Gambler's ruin)**.** Here we consider again a problem that was discussed in the workbook on variance reduction techniques. Assume that $\{Z_i\}_{i=0}^{N-1}$ are independent $\mathcal{N}(0, \sigma^2)$ random variables and define

$$S_k = s_0 + \sum_{i=0}^{k-1} Z_i, \qquad k = 1, \dots, N.$$

we want to calculate the probability of ruin within the first N games, given by

$$I = \mathbb{P}\left(\min_{k \in \{1, \dots, N\}} S_k \le 0\right),$$

In order to better estimate I with a Monte Carlo method, we will use importance sampling with an important distribution given by the PDF of the $\mathbb{R}^N$-valued random variable V obtained by

$$V_k = s_0 + \sum_{i=0}^{k-1} b(V_i) + \sum_{i=0}^{k-1} Z_i, \qquad k = 1, \dots, N, \tag{1}$$

where $b(\cdot)$ is real-valued function.

1. Calculate the likelihood ratio $g(v)$ between the PDF of $S = (S_1, \dots, S_N)^T$ and that of $V = (V_1, \dots, V_N)^T$.

2. Show that, if $V = (V_1, \ldots, V_N)^T$ is obtained from (1), then the likelihood ratio evaluated at V admits the following expression:

$$g(V) = \exp\left(-\frac{1}{\sigma^2}\left(\sum_{k=0}^{N-1} b(V_k)\, Z_k + \frac{1}{2}\sum_{k=0}^{N-1} |b(V_k)|^2\right)\right), \tag{2}$$

where we used the notation $V_0 = s_0$.

3. Calculate the expectation $\mathbb{E}[g(V)]$. Was the result expected?

4. For the parameters $N = 10$ and $\sigma = .2$, calculate I by using importance sampling using the modified dynamics (1). Can you find a choice of the function $b(\cdot)$ that produces better results than the constant function $b(\cdot) = -.1$?

Problem 7 (Control Variates)**.** Let $X \sim p$ and suppose we want to evaluate

$$\mathbb{P}(X > a) = \int_a^\infty p(x)\, dx.$$

Suppose that p is symmetric around zero, so that $\mathbb{P}(X > 0) = \frac{1}{2}$. Form a control variate estimator

$$\hat{I}_n^c = \frac{1}{n}\sum_{i=1}^{n}\left[\mathbf{1}(X_i > a) + \alpha\left(\mathbf{1}(X_i > 0) - \frac{1}{2}\right)\right]$$

1. Compute the variance $\mathrm{Var}[\hat{I}_n^c]$.

2. Find the optimal value of α for which there is maximum reduction in variance. Is it computable in practice? Find a range of α over which there will be some improvement in variance.

3. Suppose now that p is the standard Gaussian distribution, and $a = 3$. Implement both a standard MC estimator $\hat{I}_n$ and appropriately tuned control variate estimator $\hat{I}_n^c$. Plot the 95% intervals for both as a function of n.

4. Employing the fact that there there is an explicit formula for the moments of p, construct a control variate that produces a better variance reduction. Is it worth doing?

Problem 8 (Importance sampling with Gaussian mixture)**.** Consider the function

$$f(x, y) = e^{-\beta V_1(x,y)} + e^{-\beta V_2(x,y)}$$

where $\beta > 0$ and

$$V_1(x, y) = (x - 0.5)^2 + \frac{1}{2}(y + 0.1)^4, \quad V_2(x, y) = 0.75(x + 0.4)^2 + (y - 0.5)^4.$$

1. Use a Monte-Carlo algorithm to estimate the integral

$$Z = \int_{-1}^{1}\int_{-1}^{1} f(x, y)\, dx\, dy,$$

for $\beta = 100$.

2. Plot the variance of the estimator as a function of the number of random samples that you are generating. Use a deterministic numerical method to estimate Z and calculate the error as a function of the number of random samples.

3. Choose an appropriate distribution $\psi(x, y)$ and estimate Z by using importance sampling. Justify the choice of $\psi(x, y)$ and plot the variance and the error of the estimator as a function of the number of samples.

Simulation of continuous-time Gaussian processes

Problem 9 (Simulation of Markovian Gaussian processes)**.** A very useful property of multivariate Gaussian random variables is that if we condition on part of the random vector, the resulting distribution remains Gaussian. To see this, suppose that

$$\mathbf{X} = (X_1, X_2)^\top \sim \mathcal{N}(\mathbf{m}, \Sigma), \qquad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

Then we know that

$$\mathbf{X}_1 \sim \mathcal{N}(\mathbf{m}_1, \Sigma_{11}), \quad \text{and} \quad \mathbf{X}_2 \sim \mathcal{N}(\mathbf{m}_2, \Sigma_{22}).$$

Furthermore, the conditional distribution of $\mathbf{X}_2$ conditional on $\mathbf{X}_1$ is a multivariate normal with

$$\mathbb{E}[\mathbf{X}_2 \,|\, \mathbf{X}_1] = \mathbf{m}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{X}_1 - \mathbf{m}_1). \tag{3}$$

and

$$\mathrm{Var}(\mathbf{X}_2 \,|\, \mathbf{X}_1) = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \tag{4}$$

Using these properties we can develop a more efficient scheme to simulate Gaussian processes, more specifically to interpolate between already simulated points of a Gaussian process.

1. Suppose we wish to generate X_{n+1} at time t_{n+1} given that we have already generated $X_0, \ldots, X_n$.

   (a) Specify the conditional distribution of X_{n+1} given $X_0, \ldots, X_n$.

   (b) Use this to construct a numerical scheme to simulate a Gaussian stochastic process.

2. Suppose additionally that the Gaussian process $X(t)$ is Markovian, so that in particular, you only need to know the value of $X(t_n)$ to generate $X(t_{n+1})$. Construct a scheme to iteratively sample $X(t_i)$ over a sequence of points $t_0 < t_1 < t_2 < \ldots$.

   (a) In the case of Brownian motion, show that the update formula can be written as:

   $$X(t_{i+1}) = X(t_i) + \left(\sqrt{t_{i+1} - t_i}\right) Z,$$

   where $Z \sim \mathcal{N}(0, 1)$.

   (b) Derive a similar update formula for the stationary Ornstein-Uhlenbeck process with mean 0 and covariance $C(s, t) = \exp(\alpha|t - s|/2)$.

Problem 10 (Karhunen–Loève expansion)**.** Consider the Gaussian random field $X(x)$ in $\mathbb{R}$ with covariance function

$$\gamma(x, y) = e^{-a|x-y|}$$

where $a > 0$.

1. Simulate this field: generate samples and calculate the first four moments.
2. Consider $X(x)$ for $x \in [-L, L]$. Calculate analytically the eigenvalues and eigenfunctions of the integral operator $\mathcal{K}$ with kernel $\gamma(x, y)$,

$$\mathcal{K}f(x) = \int_{-L}^{L} \gamma(x, y) f(y)\, dy.$$

   Use this in order to obtain the Karhunen-Loéve expansion for X. Plot the first five eigenfunctions when $a = 1$, $L = 0.5$. Investigate (either analytically or by means of numerical experiments) the accuracy of the KL expansion as a function of the number of modes kept.
3. Develop a numerical method for calculating the first few eigenvalues/eigenfunctions of $\mathcal{K}$ with $a = 1$, $L = -0.5$. Use the numerically calculated eigenvalues and eigenfunctions to simulate $X(x)$ using the KL expansion. Compare with the analytical results and comment on the accuracy of the calculation of the eigenvalues and eigenfunctions and on the computational cost.

Additional problems

Problem 11 (Sampling uniformly on spheres and balls)**.** A rv has uniform distribution on the d–dimensional ball $D = \{x \in R^d : |x|^2 \leq 1\}$ if the rv takes values almost surely in D and has distribution

$$p(x)\, dx = \frac{dx}{\int_D 1\, dx}.$$

Similarly, a rv has uniform distribution on the sphere $C = \{x \in \mathbb{R}^d : |x|^2 = 1\}$ if the rv takes values almost surely in C and has distribution,

$$q(dx) = \frac{1}{\lambda(C)} \lambda(dx)$$

where $\lambda(dx)$ is the spherical measure on C.

1. Given $U \sim U(0, 1)$, show that $(\cos(2\pi U), \sin(2\pi U))$ is uniformly distributed on the 2D circle.
2. Suppose we want to sample uniformly from the 3D sphere. We use the spherical coordinate transformation $\psi \in [0, \pi]$, and $\theta \in [0, 2\pi]$,

$$(x(\theta, \psi), y(\theta, \psi), z(\theta, \psi)) = (\sin(\psi) \cos(\theta), \sin(\psi) \sin(\theta), \cos(\psi)) \,.$$

   (a) Let $\mathbf{X}$ be a uniformly distributed random variable on the sphere. Writing the spherical measure in spherical coordinates, write down the marginal densities of the random variables ψ and θ, and write down the CDF of ψ.

(b) Based on the previous step, generate samples ψ and θ using an appropriate method, and construct a sampler generate samples from the 2–sphere.

3. Given $(X, Y) \sim U(D)$, show that

$$\left(\frac{X}{\sqrt{X^2+Y^2}}, \frac{Y}{\sqrt{X^2+Y^2}}\right) \sim U(C),$$

where C is the 1-sphere in $\mathbb{R}^2$. Construct a rejection based sampler to generate samples $U(C)$ using proposals with distribution $U([-1,1] \times [-1,1])$. This algorithm can be readily generalised sample from spheres in arbitrary dimensions. How do you expect the average performance to depend on dimension?

4. Suppose we can generate samples $X, Y \sim \mathcal{N}(0,1)$ iid. Show that

$$\left(\frac{X}{\sqrt{X^2+Y^2}}, \frac{Y}{\sqrt{X^2+Y^2}}\right) \sim U(C).$$

Problem 12 (Sampling Gaussian random variables)**.** First we consider the problem of generating Gaussians using rejection sampling

1. The standard Cauchy distribution is a continuous probability distribution having pdf:

$$f(x) = \frac{1}{\pi(1+x^2)}.$$

Use the inverse transform method to derive an algorithm to sample from this distribution.

2. Using the Cauchy distribution as proposal, use the rejection algorithm to generate samples from the standard Gaussian distribution $e^{-x^2/2}/\sqrt{2\pi}$. Implement a function in a programming language of your choice to sample Gaussian random variables using this scheme.

3. Would it be possible to work the other way round, i.e., use rejection sampling to produce Cauchy distributed draws from using a Gaussian proposal distribution?

4. Write code to implement the Box-Muller sampling algorithm described in class. Frequently the BM-algorithm is cited as being slow due to the necessity to compute cosines and sines. Use the rejection based method described in the previous problem to obtain samples which have the same distribution as $(\cos(2\pi U_2), \sin(2\pi U_2))$. Implement code to sample Gaussian random variables using this scheme.

5. Using timing functions provided in your language (or use the shell `time` command), calculate the time of execution for all three methods, after generating 10^6 samples. Which is the fastest? Is this what you expected?

6. Use the Kolmogorov–Smirnov test as a check that the generated random numbers follow the standard normal distribution.

7. Suppose we wish to sample a pair of Gaussian random variables X_1 and X_2 having means μ_i, variances σ_i^2 and correlation ρ. By assuming that the Cholesky decomposition of the covariance matrix is of the form

$$C = \begin{pmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{pmatrix},$$

find expressions for a_{11}, a_{21} and a_{22}, and solve them to generate samples from X_1, X_2.