University of Karlsruhe
Institute of Stochastics

Prof. Dr. P. R. Parthasarathy
Dipl.-Math. D. Gentner

Applied Stochastic Models (SS 09)

Problem Set 12

Problem 1

(a) Two random variables X and Y are identically distributed. Show that

$$Var((X+Y)/2) \leq Var(X).$$

Conclude that the use of antithetic variables can never increase variance.
(b) If the random variables $(X_1, X_2, ..., X_n) =: X$ are independent and f and g are increasing functions of these n variables, use induction to prove that

$$\mathbb{E}(f(X)g(X)) \geq \mathbb{E}(f(X))\mathbb{E}(g(X)).$$

Solution: (a) According to the Cauchy-Schwartz-inequality we have

$$|\mathrm{Cov}(X,Y)| \leq \sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)},$$

which implies

$$\mathrm{Var}(\frac{X+Y}{2}) = \frac{1}{4}(\mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X,Y)) \leq \frac{1}{4}(4\mathrm{Var}(X)) = \mathrm{Var}(X).$$

(b) First let $n = 1$. Consider an independent copy X' of X (i.e. $X \stackrel{d}{=} X'$). Then

$$\begin{aligned}&\mathrm{Cov}(f(X) - f(X'), g(X) - g(X')) \\ &\quad = \mathrm{Cov}(f(X), g(X)) - \mathrm{Cov}(f(X), g(X')) - \mathrm{Cov}(f(X'), g(X)) + \mathrm{Cov}(f(X'), g(X')) \\ &\quad = 2\mathrm{Cov}(f(X), g(X)).\end{aligned}$$

Since $\mathbb{E}(f(X) - f(X')) = 0 = \mathbb{E}(g(X) - g(X'))$ we have on the other hand

$$\mathrm{Cov}(f(X) - f(X'), g(X) - g(X')) = \mathbb{E}((f(X) - f(X'))(g(X) - g(X'))),$$

and since f, g are increasing the two factors are always either both positive or both negative. Hence

$$\mathrm{Cov}(f(X), g(X)) = \frac{1}{2}\mathbb{E}((f(X) - f(X'))(g(X) - g(X'))) \geq 0,$$

which is equivalent to the assertion in the problem for $n = 1$.

Now assume the assumption is valid for all $k \leq n-1$. Then for $X = (X_1, ..., X_n)$ by independence

$$\begin{aligned}\mathbb{E}[f(X)g(X)] &= \int \mathbb{E}[f(X_1, ..., X_{n-1}, x_n)g(X_1, ..., X_{n-1}, x_n)]\mathbb{P}^{X_n}(dx_n) \\ &\overset{k=n-1}{\geq} \int \mathbb{E}[f(X_1, ..., X_{n-1}, x_n)]\mathbb{E}[g(X_1, ..., X_{n-1}, x_n)]\mathbb{P}^{X_n}(dx_n).\end{aligned}$$

where we applied the case $k = n-1$ to the monotonic functions $f(\cdot, ..., \cdot, x_n)$ and $g(\cdot, ..., \cdot, x_n)$ for arbitrary fixed x_n. The functions

$$m(x_n) := \mathbb{E}[f(X_1, ..., X_{n-1}, x_n)], \quad k(x_n) := \mathbb{E}[g(X_1, ..., X_{n-1}, x_n)]$$

are clearly monotonic, hence we may apply the case $n = 1$ as follows:

$$\begin{aligned}\int \mathbb{E}[f(X_1, ..., X_{n-1}, x_n)]\mathbb{E}[g(X_1, ..., X_{n-1}, x_n)]\mathbb{P}^{X_n}(dx_n) &= \mathbb{E}[m(X_n)k(X_n)] \\ &\geq \mathbb{E}[m(X_n)]\mathbb{E}[k(X_n)],\end{aligned}$$

and the last term is nothing but $\mathbb{E}[f(X)]\mathbb{E}[g(X)]$ by independence and Fubini.

Problem 2

Generate 300 pairs of random numbers and use them to simulate an $M/\Gamma(2,2)/1$ queue. Arrivals are exponential with mean 2 and service times are gamma with parameters (2,2). Obtain the average waiting time of the customers.

Solution: An implementation will be shown in the exercise session. Here is the simulation of the queue (i.e. the simulation of all interarrival times T_i and all service times S_i) and the derivation of the mean empirical waiting time of a customer in such a simulation. Generate independent $U_1, ..., U_n \sim U(0,1)$ ($n = 300$ here). In our case we use the 300 variables for a simulation for 100 customers since for each customer we need to simulate one interarrival time (1 variable is enough to simulate an exponential variable) and one service time (here we take two U_i´s since $\Gamma(2,2)$ is most conveniently simulated as the sum of two independent $Exp(2)$ variables). Use the first 100 to simulate the interarrival times T_i ($T_i :=$ time between the arrival of the i^{th} and the $(i-1)^{th}$ customer for $i > 1$, $T_1 :=$ arrival time of first customer) and the next 200 to simulate the service times S_i ($S_i =$ time needed to serve the i^{th} customer). In our case this means

---

for $i = 1..100$ do $\{T_i := -2 \ln U_i;\}$
for $i = 1..100$ do $\{S_i := -\frac{1}{2} \ln(U_{i+100}U_{i+200});\}$

The waiting times Q_i are then given by

$Q_1 := 0;$
for $i = 2..100$ do $\{Q_i := [Q_{i-1} + S_{i-1} - T_i]^+;\}$

---

while their average is $\overline{Q} = \frac{1}{100}\sum_{i=1}^{100} Q_i$. Note that since $\mathbb{E}[T_i] = 2 > 1 = \mathbb{E}[S_i]$ a stationary distribution exists.

Problem 3

A point process consisting of randomly occurring points in the plane is said to be a two-dimensional Poisson process having rate λ if the number of points in any given area A is Poisson distributed with mean $\lambda|A|$ ($|\cdot|$ denotes area or 2-dim Lebesgue measure) and the numbers of points in disjoint regions are independent.
Write an algorithm to simulate points of this process in a circular region of radius r centered around a fixed point O.
(Hint: Let $R_i, i = 1, 2, 3, ...$ denote the distance between O and its i^{th} nearest Poisson point. Then $P(\pi R_1^2 > b) = e^{-\lambda b}, P(\pi R_2^2 - \pi R_1^2 > b|R_1) = e^{-\lambda b}$, ...)

Solution:
We are looking for characteristics of the process which are easy to simulate. Label the points of the process in increasing order with respect to their distance R_i to the origin. Next describe a point x_i of the process in terms of its polar coordinates (R_i, φ_i). Following the hint, we note that

$$P(\pi R_1^2 > b) = P(\text{no points are in the circle around 0 with area } b) = e^{-\lambda b},$$

and

$$\begin{aligned} P(\pi R_{i+1}^2 - \pi R_i^2 > b|R_i) &= P(\text{no points are in the anulus around 0 with inner radius } R_i \text{ and area } b) \\ &= e^{-\lambda b}, \quad i > 1. \end{aligned}$$

This means that we may simulate the R_i's as follows: Start with $R_1 := \sqrt{-\frac{1}{\pi\lambda}\ln(U_1)}$, $U_1 \sim U(0,1)$, and simulate recursively $R_{i+1} := \sqrt{R_i^2 - \frac{1}{\lambda\pi}\ln(U_i)}$, $U_i \sim U(0,1)$ (the U_is must be independent).
We are left with simulating the angles φ_i. Since $P(\varphi_i \leq \vartheta) = \frac{\vartheta}{2\pi}$, φ_i is $U(0, 2\pi)$-distributed and can be simulated via $\varphi_i = 2\pi V_i$ where $V_i \sim U(0,1)$ are independent. This yields the following algorithm:

$R_0 := 0; n := 0; r := 10;$
While $R_n \leq r$ do
{
$U := Unif(0,1);$
$n := n + 1;$
$R_n := \sqrt{R_{n-1}^2 - \frac{1}{\lambda\pi}\ln(U)};$
}
delete(R_n);
$n := n - 1;$
For $i = 1, ..., n$ do
{
$V_i := Unif(0,1);$
$\varphi_i := 2\pi V_i;$
}
plot$\{(R_i\cos(\varphi_i), R_i\sin(\varphi_i)), i = 1, ..., n\};$

Problem 4

Use the Gibbs sampler to generate n random points in the unit circle conditional on the event that no two points are within a distance d of each other ($d < \frac{2\pi}{n}$), where

$$P(\text{no two points are within } d \text{ of each other})$$

is assumed to be a small positive number.

Solution: The assumption that

$$P(\text{no two points are within } d \text{ of each other}) =: p$$

is a small positive number is not essential for the Gibbs sampling algorithm to work, but it gives the reason, why it is preferable to apply the Gibbs method in contrast to just keep simulating n points in each step and accept the realization iff the distance condition is fulfilled. The latter algorithm would need in average $1/p$ (mean of a geometric distribution with parameter p) steps until an acceptable realization comes out, which is a large number of iterations under the assumtion that p is very small (The result would be an exact (!) simulation of the desired conditional distribution though.). Instead if one does not have the time and is satisfied with an approximate simulation of this conditional distribution, the Gibbs sampler comes into play.
We describe the n points by n-tuples $x = (x_1, ..., x_n)$, where $x_i \in (0, 2\pi)$. Start with some configuration x of n points on the circle that fulfills the distance condition (for example choose $x = (0, \frac{d}{2\pi}, \frac{2d}{2\pi}, \frac{3d}{2\pi}, ..., , \frac{nd}{2\pi})$). Then choose one coordinate x_I at random, i.e. I takes every value in $\{1, 2, ..., n\}$ with equal probability $\frac{1}{n}$. Next generate a random point q on the circle (i.e. a random number q in $(0, 2\pi)$) and check if this point is not within d of any of the remaining points. If it is not, the new configuration is x with x_I replaced by q, otherwise the configuration x remains unchanged. After a large number of repetitions the distribution of x will be close to the conditional distribution we wanted to simulate.