CFRM 505 Homework 1

Eunki Chung
eunkich@uw.edu

January 10, 2023

1 Problem 1

Consider the random variables X and Y with the following joint distribution:

$$f_{XY}(x,y) = \begin{cases} \frac{xy^2+2xy}{Z} & 0 < x, y < 1 \\ 0 & \text{otherwise} \end{cases}$$

where Z is some constant. Calculate the following quantities analytically (**not** by simulation).

By the Probability Axiom,

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{xy^2+2xy}{Z}\,dx\,dy = 1$$

$$\begin{aligned}
\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{xy^2+2xy}{Z}\,dx\,dy &= \int_0^1\int_0^1 \frac{y^2+2y}{Z}x\,dx\,dy \\
&= \int_0^1 \frac{y^2+2y}{Z}\left[\frac{1}{2}x^2\right]_0^1 dy \\
&= \int_0^1 \frac{y^2+2y}{2Z}\,dy \\
&= \left.\frac{\frac{1}{3}y^3+y^2}{2Z}\right|_0^1 \\
&= \frac{\frac{1}{3}+1}{2Z} \\
&= \frac{\frac{4}{3}}{2Z} \\
&= \frac{2}{3Z}
\end{aligned}$$

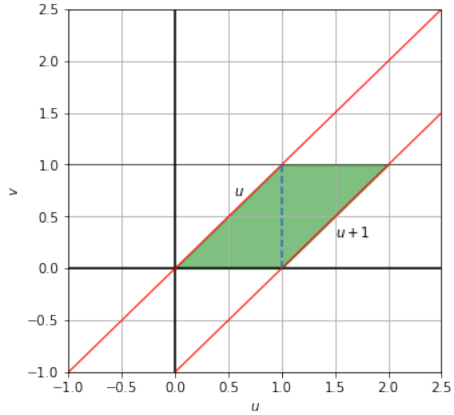
Thus,

$$\begin{aligned}
\frac{2}{3Z} &= 1 \\
Z &= \frac{2}{3}
\end{aligned}$$

$$f_{XY}(x,y) = \begin{cases} \frac{3}{2}x(y^2+2y) & 0 < x, y < 1 \\ 0 & \text{otherwise} \end{cases}$$

1)

Consider the following variable transformation $\begin{array}{l} U = X + Y \\ V = Y \end{array} \Rightarrow \begin{array}{l} X = U - V \\ Y = V \end{array}$,
with support $S' = \{(u,v) \in \mathbb{R}^2 : v < u < 1 + v,\ 0 < v < 1\}$



The Jacobian of the Transformation is given by, $|J| = \left| \det \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix} \right| = \left| \det \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \right| = 1$

$$\begin{aligned} f_{U,V}(u,v) &= \frac{3}{2}(u-v)(v^2+2v) \cdot |J| \\ &= \begin{cases} \frac{3}{2}(u-v)(v^2+2v) & (u,v) \in S' \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Notice that the support S' has the following region.

Thus,

$$\begin{aligned} f_U(u) &= \int_{-\infty}^{\infty} f_{U,V}(u,v)\, dv \\ &= \begin{cases} \int_0^u f_{U,V}(u,v)\, dv & 0 < u < 1 \\ \int_{u-1}^1 f_{U,V}(u,v)\, dv & 1 < u < 2 \end{cases} \end{aligned}$$

Since we're only interested in the region of $U = X + Y < 0.5$, we'll just use the first case.

$$\begin{aligned}
f_U(u) &= \int_0^u \frac{3}{2}(u-v)(v^2+2v)\,dv \\
&= \frac{3}{2}\int_0^u uv^2 + 2uv - v^3 - 2v^2\,dv \\
&= \frac{3}{2}\int_0^u -v^3 + (u-2)v^2 + 2uv\,dv \\
&= \frac{3}{2}\left[-\frac{1}{4}v^4 + \frac{u-2}{3}v^3 + uv^2\right]_0^u \\
&= \frac{3}{2}\left(-\frac{1}{4}u^4 + \frac{u-2}{3}u^3 + u^3\right) \\
&= \frac{3}{2}\left(-\frac{1}{4}u^4 + \frac{1}{3}u^4 - \frac{2}{3}u^3 + u^3\right) \\
&= \frac{3}{2}\left(\frac{1}{12}u^4 + \frac{1}{3}u^3\right) \\
&= \frac{1}{8}u^4 + \frac{1}{2}u^3
\end{aligned}$$

$$f_U(u) = \frac{1}{8}u^4 + \frac{1}{2}u^3 \quad 0 < u < 1$$

$$\begin{aligned}
P[X+Y<0.5] &= \int_0^{0.5} f_U(u)du \\
&= \int_0^{0.5} \frac{1}{8}u^4 + \frac{1}{2}u^3 du \\
&= \frac{1}{40}u^5 + \frac{1}{8}u^4\Big|_0^{\frac{1}{2}} \\
&= \frac{1}{2^3\cdot 5}\cdot\frac{1}{2^5} + \frac{1}{2^3}\cdot\frac{1}{2^4} \\
&= \frac{1}{2^8\cdot 5} + \frac{1}{2^7} \\
&= \frac{1}{2^7}\cdot\frac{11}{10} \\
&= \frac{11}{1280}
\end{aligned}$$

2)

$$f_{X,Y}(x,y) = \begin{cases} \frac{3}{2}x(y^2+2y) & 0 < x, y < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$
\begin{aligned}
f_X(x) &= \int_0^1 f_{X,Y}(x,y)\,dy \\
&= \int_0^1 x\left(\frac{3}{2}y^2 + 3y\right)\,dy \\
&= x\left[\frac{1}{2}y^3 + \frac{3}{2}y^2\right]_0^1 \\
&= 2x
\end{aligned}
$$

$$
f_X(x) = \begin{cases} 2x & 0 < x < 1 \\ 0 & o.w. \end{cases}
$$

$$
\begin{aligned}
E[X] &= \int_0^1 x \cdot f_X(x)\,dx \\
&= \int_0^1 2x^2\,dx \\
&= \frac{2}{3}x^3\Big|_0^1 \\
&= \frac{2}{3}
\end{aligned}
$$

3)

$$
Var[Y] = E[Y^2] - E[Y]^2
$$

$$
f_{X,Y}(x,y) = \begin{cases} \frac{3}{2}x(y^2+2y) & 0 < x, y < 1 \\ 0 & \text{otherwise} \end{cases}
$$

Notice that the joint pdf on the support can be factorized into two parts $f_{X,Y}(x,y) = g(x) \cdot h(y)$, which implies that X, Y are independent. Thus, by the definition of independence, we can get the marginal pdf of Y, f_Y by deviding the joint pdf $f_{X,Y}$ with the marginal pdf of X, f_X, which we calculated above.

$$
\begin{aligned}
f_Y(y) &= \frac{f_{X,Y}(x,y)}{f_X(x)} \\
&= \frac{\frac{3}{2}x(y^2+2y)}{2x} \\
&= \frac{3}{4}(y^2+2y) \\
&= \begin{cases} \frac{3}{4}(y^2+2y) & 0 < y < 1 \\ 0 & o.w. \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
E[Y] &= \int_0^1 y \cdot f_Y(y)\,dy \\
&= \int_0^1 \frac{3}{4}y^3 + \frac{3}{2}y^2\,dy \\
&= \frac{3}{16}y^4 + \frac{1}{2}y^3\Big]_0^1 \\
&= \frac{3}{16} + \frac{1}{2} \\
&= \frac{11}{16}
\end{aligned}
$$

$$
\begin{aligned}
E\left[Y^2\right] &= \int_0^1 y^2 f_Y\left(y\right)\,dy \\
&= \int_0^1 \frac{3}{4}y^4 + \frac{3}{2}y^3\,dy \\
&= \frac{3}{20}y^5 + \frac{3}{8}y^4\Big]_0^1 \\
&= \frac{3}{20} + \frac{3}{8} \\
&= \frac{6}{40} + \frac{15}{40} \\
&= \frac{21}{40}
\end{aligned}
$$

$$
\begin{aligned}
Var\left[Y\right] &= \frac{21}{40} - \left(\frac{11}{16}\right)^2 \\
&= \frac{21}{40} - \frac{121}{256} \\
&= \frac{67}{1280}
\end{aligned}
$$

4)

Since $X \perp\!\!\!\perp Y,\ E[XY] = E[X]E[Y]$.

$$Cov(X,Y) = E[XY] - E[X] \cdot E[Y] = 0$$

5)

$$Corr(X,Y) = 0 \quad (\because\ Cov(X,Y) = 0)$$

6)

Since $0 < x < 1,\ 0 < y < 1$, and $X \perp\!\!\!\perp Y$,

$$P(X > 0.5 | Y > 0.5) = \frac{P\left(X > 0.5,\ Y > 0.5\right)}{P\left(Y > 0.5\right)}$$

$$\begin{aligned} P(X > 0.5,\ Y > 0.5) &= P(X > 0.5) \cdot P(Y > 0.5) \\ &= (1 - P(X \leq 0.5)) \cdot (1 - P(Y \leq 0.5)) \end{aligned}$$

where

$$\begin{aligned} P(X \leq 0.5) &= \int_0^{0.5} 2x \, dx \\ &= \left. x^2 \right|_0^{0.5} \\ &= \frac{1}{4} \end{aligned}$$

$$\begin{aligned} P\left(Y \leq 0.5\right) &= \int_0^{0.5} \frac{3}{4} y^2 + \frac{3}{2} y \, dy \\ &= \left. \frac{1}{4} y^3 + \frac{3}{4} y^2 \right|_0^{\frac{1}{2}} \\ &= \frac{1}{32} + \frac{3}{16} \\ &= \frac{7}{32} \end{aligned}$$

$$\begin{aligned} P(X > 0.5,\ Y > 0.5) &= \left(1 - \frac{1}{4}\right) \cdot \left(1 - \frac{7}{32}\right) \\ &= \frac{3}{4} \cdot \frac{25}{32} \\ &= \frac{75}{128} \end{aligned}$$

$$P(X > 0.5 | Y > 0.5) = \frac{75}{128} \cdot \frac{32}{25} = \frac{3}{4}$$

7)

$$\begin{aligned} f_{X|Y}(x|y) &= \frac{f_{X,Y}(x,y)}{f_Y(y)}, \quad 0 < x, y < 1 \\ &= \frac{f_X(x) f_Y(y)}{f_Y(y)} \\ &= f_X(x) \\ &= 2x \end{aligned}$$

$$f_{X|Y=0.25}(x) = 2x$$

$$\begin{aligned} E[X|Y=0.25] &= E[X] \\ &= \frac{2}{3} \end{aligned}$$

2 Problem 2

Use Monte Carlo integration to estimate

$$\int_{-2}^{3} 3x^2 + 2x \,\mathrm{d}x$$

(The exact answer is 40. You don't need to rederive it.)

Use a sample size of at least $n = 100,000$.

Let $u = \frac{1}{5}x + \frac{2}{5} \Leftrightarrow x = 5u - 2, \quad \frac{dx}{du} = 5$

$$\begin{aligned} \int_{-2}^{3} 3x^2 + 2x\,dx &= \int_0^1 (3(5u-2)^2 + 2(5u-2))\frac{dx}{du}\,du \\ &= \int_0^1 (3(25u^2 - 20u + 4) + 10u - 4) \cdot 5\,du \\ &= \int_0^1 \left(75u^2 - 60u + 12 + 10u - 4\right) \cdot 5\,du \\ &= \int_0^1 375u^2 - 250u + 40\,du \\ &= E[f(U)] \end{aligned}$$

where $U \sim \text{Unif}(0,1)$ and $f(u) = 375u^2 - 250u + 40$

By default, `np.random.random()` method uses Mersenne Twister algorithm (`MT19937`) for its Pesudo Random Number Generator(PRNG). However, the official numpy documentation recommends to use other modern PRNG instead of `MT19937`, since `MT19937` fails some statiscal tests and does not have any significant advantage in its generation speed.

Therefore, for the generation of Uniform random samples, I used Permutation Congruential Generator (`PCG64`) as recommended.

In addition, I used parallel processing to generate very large numbers of sample. As shown in below, multi-threaded generator made a significant improvement in speed compared to a single-threaded generator.

Ref: https://numpy.org/doc/stable/reference/random/performance.html

[5]:
```
import numpy as np
rng = np.random.default_rng(12)
# Use Permutation Congruential Generator(PCG64) instead of Mersenne␣
↪Twister(MT19937)
```

[6]:
```python
from numpy.random import default_rng, SeedSequence
import multiprocessing
import concurrent.futures
import numpy as np

class MultithreadedRNG:
    def __init__(self, n, seed=None, threads=None):
        if threads is None:
            threads = multiprocessing.cpu_count()
        self.threads = threads

        seq = SeedSequence(seed)
        self._random_generators = [default_rng(s)
                                   for s in seq.spawn(threads)]

        self.n = n
        self.executor = concurrent.futures.ThreadPoolExecutor(threads)
        self.values = np.empty(n)
        self.step = np.ceil(n / threads).astype(np.int_)

    def fill(self):
        def _fill(random_state, out, first, last):
            random_state.random(out=out[first:last])

        futures = {}
        for i in range(self.threads):
            args = (_fill,
                    self._random_generators[i],
                    self.values,
                    i * self.step,
                    (i + 1) * self.step)
            futures[self.executor.submit(*args)] = i
        concurrent.futures.wait(futures)

    def __del__(self):
        self.executor.shutdown(False)
```

[7]:
```python
mrng = MultithreadedRNG(100_000_000, seed=12)
```

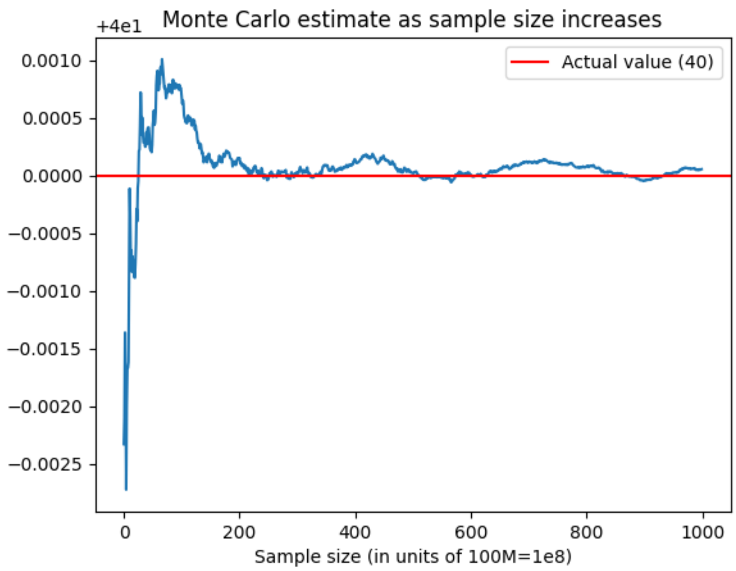
[8]:
```python
from timeit import timeit
print("Single Thread")
value = np.empty(100_000_000)
%timeit rng.random(out=value)
print(f"Multi Thread (num_threads={mrng.threads})")
%timeit mrng.fill()
```

```
Single Thread
```

```
437 ms ± 19.5 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
Multi Thread (num_threads=8)
75.1 ms ± 1.5 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

[9]:
```
from tqdm import tqdm
val = 0.
ests = []
for i in tqdm(range(1000)):
    mrng.fill()
    smpls = mrng.values
    val += np.sum(375 * (smpls ** 2) - 250 * smpls + 40)
    ests.append(val / ((i+1) * int(1e8)))
```

[10]:
```
import matplotlib.pyplot as plt
plt.plot(ests)
plt.axhline(40, color='red', label="Actual value (40)")
plt.xlabel("Sample size (in units of 100M=1e8)")
plt.legend()
plt.title("Monte Carlo estimate as sample size increases")
plt.show()
```



[11]:
```
print(f"{'Monte Carlo estimate ': <21} = {ests[-1]}")
print(f"{'Theoretical value ': <21} = {40}")
```

```
Monte Carlo estimate  = 40.0000533971071
Theoretical value     = 40
```

3 Problem 3

Use Monte Carlo integration to estimate

$$\int_0^{\infty} \int_0^1 \frac{xy}{1+x^4}\,\mathrm{d}y\mathrm{d}x$$

(The exact answer is $\pi/8$. You don't need to rederive it.)

Use a sample size of at least $n = 100,000$.

Let $z = \frac{x}{1+x} = \frac{1}{\frac{1}{x}+1} \Leftrightarrow x = \dfrac{z}{1-z}, \quad \dfrac{dx}{dz} = \dfrac{1}{(1-z)^2}$

Notice that $0 < x < \infty \Leftrightarrow 0 < z < 1$ since

$$\lim_{x\to 0+} \frac{1}{\frac{1}{x}+1} = 0, \quad \lim_{x\to\infty} \frac{1}{\frac{1}{x}+1} = 1$$

$$\begin{aligned}
\int_0^{\infty} \int_0^1 \frac{xy}{1+x^4}\,\mathrm{d}y\mathrm{d}x &= \int_0^1 \int_0^1 \frac{\frac{z}{1-z}y}{1+\left(\frac{z}{1-z}\right)^4} \cdot \frac{1}{(1-z)^2}\,dx\,dz \\
&= \int_0^1 \int_0^1 \frac{\frac{z}{(1-z)^3}}{1+\frac{z^4}{(1-z)^4}} y\,dz\,dy \\
&= \int_0^1 \int_0^1 \frac{(1-z)\cdot z}{(1-z)^4+z^4} y\,dz\,dy \\
&= E[f(Z)\cdot Y]
\end{aligned}$$

where $Y,\ Z \overset{\text{iid}}{\sim} \text{Unif}(0,1)$ and $f(z) = \dfrac{(1-z)\cdot z}{(1-z)^4+z^4}$

[12]:
```python
u1 = mrng.values.copy()
mrng.fill()
u2 = mrng.values.copy()
est = np.mean((((1-u1) * u1) / ((1-u1) ** 4 + u1 ** 4)) * u2)
print(f"{'Monte Carlo estimate ' : <21} = {est}")
print(f"{'Theoretical value ' : <21} = {np.pi / 8}")
```

```
Monte Carlo estimate  = 0.39275503717035704
Theoretical value     = 0.39269908169872414
```

4 Problem 4

Let $U \sim U(0,1)$ and $X = 1-U$. Use Monte Carlo simulation to confirm that

$$\mathbb{E}[X] = \frac{1}{2} \text{ and } \mathrm{Var}[X] = \frac{1}{12} \text{ and } \mathrm{Cov}[U,X] = -\frac{1}{12}$$

Use a sample size of at least $n = 100,000$.

[13]:
```python
u = mrng.values
x = 1 - u
```

[14]:
```python
print("Monte Carlo estimates")
print("---" * 15)
print(f"{'E[X] ' : <10} = {np.mean(x)}")
print(f"{'Var[X] ': <10} = {np.cov(u,x)[0,0]}")
print(f"{'Cov[U, X] ': <10} = {np.cov(u,x)[0,1]}")
print("---" * 15, "\n")
print("Theoretical values")
print("---" * 15)
print(f"{'E[X] ' : <10} = {1/2}")
print(f"{'Var[X] ': <10} = {1/12}")
print(f"{'Cov[U, X] ': <10} = {-1/12}")
```

```
Monte Carlo estimates
---------------------------------------------
E[X]       = 0.4999481238364926
Var[X]     = 0.08332875314388954
Cov[U, X]  = -0.08332875314388954
---------------------------------------------

Theoretical values
---------------------------------------------
E[X]       = 0.5
Var[X]     = 0.08333333333333333
Cov[U, X]  = -0.08333333333333333
```

5 Problem 5

Let U_1 and U_2 be i.i.d. uniform random variables on $(0, 1)$. Define $X = \max(U_1, U_2)$. Derive the following by hand and then use Monte carlo simulation to estimate them:

1) $\mathbb{E}[X]$
2) $\mathbb{P}[X < 0.25]$

Use a sample size of at least $n = 100,000$ for each part.

$$X = \max(U_1, U_2), \quad \text{where } U_1, U_2 \overset{\text{iid}}{\sim} \text{Unif}\,(0, 1)$$

$$
\begin{aligned}
P(X \leq x) &= P(\max{(U_1, U_2)} \leq x) \\
&= P(U_1 \leq x) \cdot P\left(U_2 \leq x\right) \\
&= \int_0^x 1 \, du_1 \cdot \int_0^x 1 \, du_2 \\
&= x^2
\end{aligned}
$$

By definition of cdf,

$$
F_X(x) = P(X \leq x) = \begin{cases} 1 & 1 \leq x \\ x^2 & 0 < x < 1 \\ 0 & x \leq 0 \end{cases}
$$

and the pdf is

$$
f_X(x) = F'_X(x) = \begin{cases} 2x & 0 < x < 1 \\ 0 & o.w. \end{cases}
$$

1)

$$
\begin{aligned}
E[X] &= \int_0^1 x f_X(x) \, dx \\
&= \int_0^1 2x^2 \, dx \\
&= \frac{2}{3} x^3 \Big|_0^1 \\
&= \frac{2}{3}
\end{aligned}
$$

2)

$$
P(X \leq 0.25) = F_X\left(\frac{1}{4}\right) = \frac{1}{16}
$$

[15]:
```
x = np.max([u1, u2], axis=0)
```

[16]:
```
print(f"{'Monte Carlo estimate ': <21}" + f"= {np.mean(x)}")
print(f"{'Theoretical value ': <21}" + f"= {2/3}")
```

```
Monte Carlo estimate = 0.66668394507005
Theoretical value    = 0.6666666666666666
```

[17]:
```
print(f"{'Monte Carlo estimate ': <21}" + f"= {np.mean(x < 0.25)}")
print(f"{'Theoretical value ': <21}" + f"= {1/16}")
```

```
Monte Carlo estimate = 0.06248674
Theoretical value    = 0.0625
```