Due: Wednesday, March 24, 1:30 pm
Submission by email, send completed Jupyter notebook.

Consider the portfolio loss incurred from defaults,

\[ L(X) = c_1 1\{X_1 > x_1\} + \ldots + c_n 1\{X_n > x_n\} \]

where the \( c_i \)'s are the size of the outstanding loan to obliger \( i \), and \( 1\{X_i > x_i\} \) represents the random indicator for whether or not obliger \( i \) will default on their loan.

Here \( X_i \) somewhat represents the level of financial strain for obliger \( i \). This reflects the individual (“idiosyncratic”) situation for each obliger, but all obligers are equally affected by broad economic swings. We model these separately, so

\[ X_i = \sqrt{(1 - \rho)}\eta_i + \sqrt{\rho}Z \]

where \( \rho \in (-1, 1) \) specifies the \( \text{Corr}(X_i, X_j) \), \( Z \sim \text{Normal}(0, 1) \) are shared between all obligers, and \( \eta_i \overset{\text{i.i.d.}}{\sim} \text{Normal}(0, 1) \) is the idiosyncratic variable.

This problem is inspired by Section 5 of Chan and Kroese (2011), though I have simplified it a bit. This paper is extremely well written, I’d recommend taking a look for more explanation & context.

The main goal is to use crude Monte Carlo, cross-entropy method, and the improved cross-entropy method (with MCMC samples) to estimate

\[ \ell = \mathbb{P}(L(X) > \gamma) \]

We start by importing some packages and defining the constants for our particular problem.

```python
import arviz as az
import matplotlib.pyplot as plt
import numpy as np
import numpy.random as rnd
from scipy import stats
from tqdm import trange, tqdm
%config InlineBackend.figure_format = 'retina'
```
\[
\begin{align*}
n &= 10 \\
\rho &= 0.5 \\
c_s &= \exp(0.2 \times \text{arange}(n)) \\
x_s &= \exp(0.2 \times \text{arange}(n)) \\
\gamma &= 0.75 \times \text{sum}(c_s) \\
\text{print}("c:", \text{list}(c_s)) \\
\text{print}("x:", \text{list}(x_s)) \\
\text{print}("\gamma:", \gamma)
\end{align*}
\]

I’ll supply the code for a (very) crude Monte Carlo run using a small number of \( R \) replications.

\[
\begin{align*}
\text{rng} &= \text{rnd.default_rng}(1) \\
R &= 10^6 \\
\eta_s &= \text{rng.normal(size=(R, n))} \\
Z_s &= \text{rng.normal(size=(R, 1))} \\
X_s &= \sqrt{1-\rho} \times \eta_s + \sqrt{\rho} \times Z_s \\
defaults &= X_s > x_s \\
losses &= \text{np.dot}(defaults, c_s) \\
est &= losses > \gamma \\
\ell_{Hat} &= \text{est}.mean() \\
\sigma_{Hat} &= \text{est}.std() \\
width\text{CI} &= 1.96 \times \sigma_{Hat} / \sqrt{R} \\
\text{print}(f"CMC estimate: \{\ell_{Hat} \pm width\text{CI}\})") \\
\text{print}(f"CMC low bound: \{\text{np.maximum}(\ell_{Hat}-width\text{CI}, 0)\})") \\
\text{print}(f"CMC upp bound: \{\ell_{Hat}+width\text{CI}\})")
\end{align*}
\]

**Big crude Monte Carlo**

1) Run repeated crude Monte Carlo tests, as in the code demonstrations, so that in total you have a combined CMC test with \( R = 10^9 \) iterations.

\[
\begin{align*}
\text{rng} &= \text{rnd.default_rng}(2) \\
R &= 10^9 \\
\# ADD CODE HERE
\end{align*}
\]

\[
\begin{align*}
\text{print}(f"CMC estimate: \{\ell_{Hat} \pm width\text{CI}\})") \\
\text{print}(f"CMC low bound: \{\text{np.maximum}(\ell_{Hat}-width\text{CI}, 0)\})") \\
\text{print}(f"CMC upp bound: \{\ell_{Hat}+width\text{CI}\})")
\end{align*}
\]
Improved cross-entropy method

2) Use MCMC to sample \( \eta = (\eta_1, \ldots, \eta_n) \) and \( Z \) conditionally on the event that \( L(X) > \gamma \) where \( X \equiv X(\eta, Z) \). Use the random walk sampler where each jump is an \( n + 1 \) dimensional vector of i.i.d. normal random variables.

Note, if we say \( X \equiv X(\eta, Z) \), then the target density in terms of \( \eta \) and \( Z \) is

\[
\pi(\eta, Z) = \frac{1}{\ell} \mathbb{1}\{L(X(\eta, Z)) > \gamma\} \phi(Z) \prod_{i=1}^{n} \phi(\eta_i)
\]

where \( \phi \) is the p.d.f. of a standard normal distribution. Since we don’t need proportionality constants, we can instead use

\[
\pi(\eta, Z) = \mathbb{1}\{L(X(\eta, Z)) > \gamma\} \exp\left\{-\frac{Z^2}{2}\right\} \prod_{i=1}^{n} \exp\left\{-\frac{\eta_i^2}{2}\right\}
\]

\[
3): \text{%%%time}
\]

\[
\text{rng} = \text{rnd.default_rng}(3)
\]
\[
R = 10**6
\]
\[\text{# ADD CODE HERE}\]

3) Print out the traceplots for \( \eta_1 \) and \( Z \). Throw away some burn in samples if you decide it is necessary.

\[\text{[ ]:}\]

\[\text{[ ]:}\]

4) Calculate the effective sample size (ESS) for your \( Z \) samples and make sure that your \( R \) is large enough so that this ESS is at least 1000. If it is too small, go back and update the previous cells until this constraint is reached. If this takes a long time, try playing with the scale parameter for the MCMC jumps.

\[\text{[ ]:}\]

5) Use the arviz \texttt{plot_posterior} function to visualise the \( \eta_0 \) samples, again for the \( \eta_n \) samples, and again for the \( Z \) samples.

\[\text{[ ]:}\]

\[\text{[ ]:}\]

\[\text{[ ]:}\]
6) Calculate the overall sample mean $\overline{\eta}$ of the $\eta$ samples (i.e. just one number for the mean of the $R \times n$ matrix of samples), and the sample mean $\overline{Z}$ of the $Z$ samples.

```
7) Run importance sampling with $R = 10^6$ samples where we sample each $\eta_i$ from a Normal($\overline{\eta}, 1$) distribution and each $Z$ from a Normal($\overline{Z}, 1$) distribution. This is the improved cross-entropy estimate. Print the result and the confidence interval.

```

```
%%time
rng = rnd.default_rng(5)
R = 10**6

# ADD CODE HERE

print(f"ICE estimate: \t {\ell_Hat} (+/- {width_CI})")
print(f"ICE low bound: \t {\ell_Hat-width_CI}")
print(f"ICE upp bound: \t {\ell_Hat+width_CI}")
```

Cross entropy

The problem above is particularly hard for the traditional CE method (this is probably why the authors chose it to compare their ‘improved’ version against it). Let’s consider the same problem except the loss will instead be

$$L(X) = X_11\{X_1 > x_1\} + \ldots X_n1\{X_n > x_n\}.$$ 

Here are the constants we’ll use for this question.

```
8) Use the original cross-entropy algorithm to find a good proposal distribution. Look inside the family of distributions where $\eta_n \sim$ Normal($\overline{\eta}_n, 1$) (note, the other $\eta_i$’s are unchanged) and where $Z \sim$ Normal($\overline{Z}, 1$).

```

```
%%time
rng = rnd.default_rng(6)
R = 10**6
maxIter = 20
rho = 0.05
```

```
\begin{code}
\begin{verbatim}
v = (0, 0)

for iterNum in range(maxIter):
    print(v)

    # ADD CODE HERE

9) Run importance sampling with this proposal to get the cross-entropy estimate.

[ ]: %%time
rng = rnd.default_rng(1234)
R = 10**6

# ADD CODE HERE

print(f"CE estimate:\t \{\ellHatCE}\ (+/‐ \{widthCICE\})")
print(f"CE low bound:\t \{\ellHatCE‐widthCICE\}")
print(f"CE upp bound:\t \{\ellHatCE+widthCICE\}")
\end{verbatim}
\end{code}