Expected Shortfall about Modified Sampling

Jingjie Zhou

1.Introduction

In this project, I created a constant correlation matrix with parameter ρ on N entities is a correlation matrix with every off-diagonal element equal to ρ. Such matrices are used as a way of dealing with the essentially unknowable correlations between underlying normal variables of copula models. And I also compared the portfolio shortfall using importance sampling shift and non-importance sampling. Comparing the difference of two random sampling methods.

2.Function Used

The assignment specifications require the following 6 functions:

- portfolioCorrelationMatrix(rhoDefault,corrEq, corrVol, corrHzd,rhoSame, rhoUnrelated,) capable of generating a covariance matrix
- cointegratedUniform(rhoDefault,corrEq, corrVol, corrHzd,rhoSame, rhoUnrelated,samples) capable of generating an array of M simulated uniformly distributed random variables using a multivariate gaussian copula model
- defaultTimes(hazardRates, samples) capable of generating an array of M simulated default time arrays ∼τT on N securities using a copula model
- equityPrices(muE, T, S0, sigmaE, defTimes,samples) capable of generating an array of M simulated equity price arrays S∼T on N securities using a multivariate gaussian copula model
- hazardRates(muH, T, h0, sigmaH, samples) capable of generating an array of M simulated equity price arrays ∼hT on N securities using a multivariate gaussian copula model
- ZCBValues(bondTenors, bondRiskFreeRates, defTimes, h, recoveryRates, T)capable of generating an array of M simulated zero coupon bond values B∼ on N securities

- volatilities(eta,T, sigma0, varvol,samples)capable of generating an array of M simulated equity price arrays ~h on N securities using a multivariate gaussian copula model

- optionPrices(callput, ST, K, r, optionTenors,sigmaT, q, defTimes,T)capable of generating an array of M option price arrays S~on N options

- portfolioValueElements(T,eqPositions, bondPositions, optionPositions,S0, h0, sigma0,muE, muH, eta,sigmaH, varvol,bondTenors, bondRiskFreeRates, recoveryRates,callput, K, r, optionTenors, q,rhoDefault,corrEq, corrVol, corrHzd,rhoSame, rhoUnrelated,samples)capable of generating a two-dimensional M × 3N array of instrument valuations. The first N are for equity values, the next for bonds and the final N for options

- portfolioValue(T,eqPositions, bondPositions, optionPositions,S0, h0, sigma0,muE, muH, eta,sigmaH, varvol,bondTenors, bondRiskFreeRates, recoveryRates,callput, K, r, optionTenors, q,rhoDefault,corrEq, corrVol, corrHzd,rhoSame, rhoUnrelated,samples)capable of generating an array of M portfolio valuations

- expectedShortfall(valuations, level=0.05, weights=1)that returns estimated expected shortfall from a set of samples at the specified level

- importantSampleDetailes(originalUniformSamples,correlation_in,shift_in=0)

3.Test Suite

I pass all the tests.

4.Range of Input

sample=(10000,6*4)
level=0.05
I tried several shift to see the difference between samples and important sampling samples.

```
shift=[0,-0.1,-0.2,-0.3,-0.4,-0.5,-0.6,-0.7,-0.8,-0.9,-1]
```

```
corrEq = np.array([[1,0.11,0.22],[0.11,1,0.33],[0.22,0.33,1]])
corrVol = np.array([[1,0.101,0.202],[0.101,1,0.303],[0.202,0.303,1]])
corrHzd = np.array([[1,0.01,0.02],[0.01,1,0.03],[0.02,0.03,1]])
rhoDefault=0.45
rhoSame=0.1
rhoUnrelated=0.05
hazardRates = np.array([0.01,0.0001,0.91,0.055])
samples = np.abs(np.sin( np.cumsum( np.cumsum( np.ones((10000,6*4)),axis=0), axis=1) ))
samples[1:4,1] = 1e-9
corrEq = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrVol = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrHzd = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
T = 1.5/52.
S0 = np.array([40., 50., 250.,135.,130.,55.])
h0 = np.array([0.024, 0.035, 0.025,0.035,0.015,0.035])
sigma0 = np.array([0.1,0.2,0.3,0.2,0.4,0.3])
muE = np.array([0.02, 0.04, 0.02,0.04,0.03,0.02])
muH = np.array([0.0, 0, 0,0,0,0])
eta = np.array([0.0, 0, 0,0,0,0])
sigmaH = np.array([0.23,0.25,0.13,0.29,0.35,0.27])
varvol = np.array([0.36,0.25,0.57,0.21,0.22,0.13])
bondTenors = np.array([4,8,3.7,4,3,7])
bondRiskFreeRates = np.array([0.052, 0.042, 0.022,0.024,0.033,0.021])
recoveryRates = np.array([0.12, 0.15, 0.13,0.25,0.22,0.15])
callput = np.array([-1,-1,1,-1,1,-1])
K = np.array([90., 55., 150, 35,35,45])
r = np.array([0.014, 0.023, 0.032,0.016,0.032,0.019])
optionTenors = np.array([0.21, 0.26, 0.32,0.28,0.32,0.6])
q = np.array([0.0015, 0.0022, 0.0023,0.0016,0.0023,0.0023])
eqPositions = np.array([8,6,10,3,6,10])
bondPositions = np.array([0,0,0,0,0,0])
optionPositions = np.array([0,0,0,0,0,0])
rhoDefault = 0.45
rhoSame = 0.1
rhoUnrelated = 0.05
```

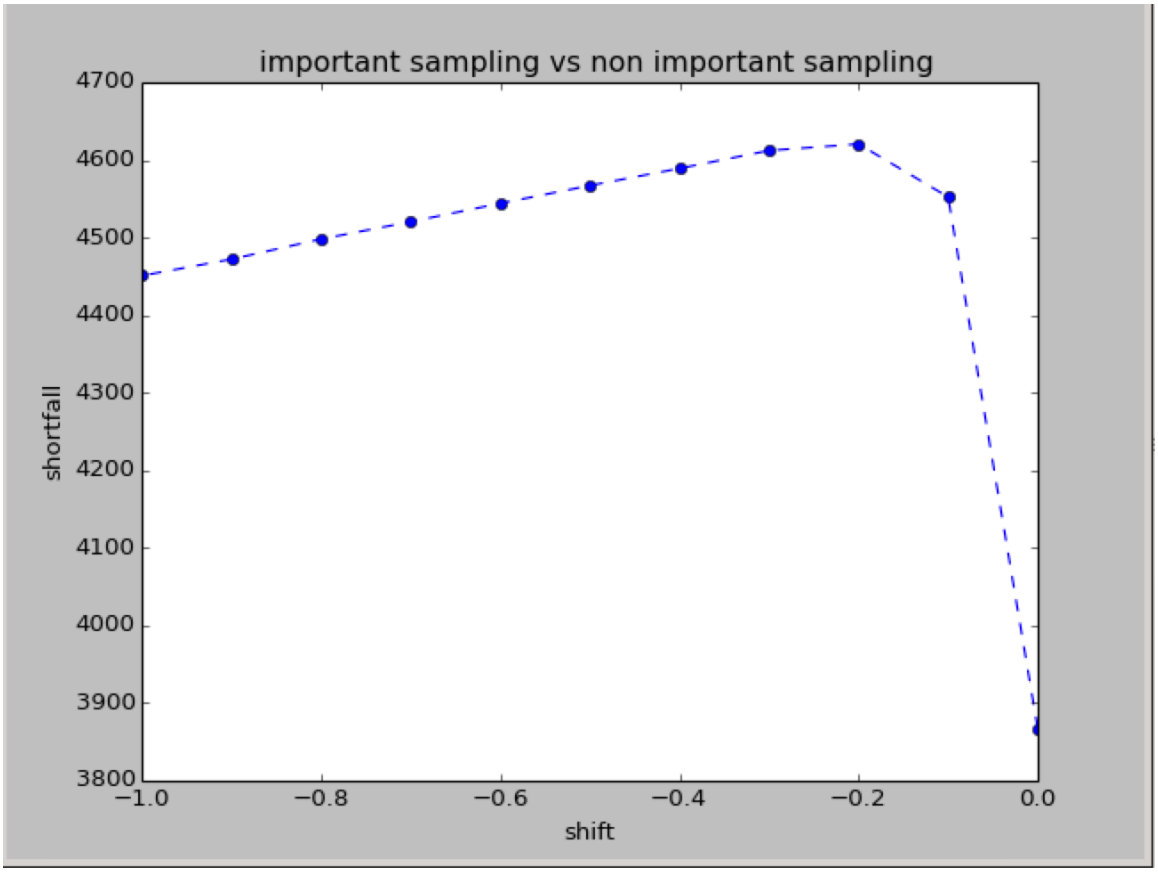
```
corrEq = np.array([[1,0.11,0.22],[0.11,1,0.33],[0.22,0.33,1]])
corrVol = np.array([[1,0.101,0.202],[0.101,1,0.303],[0.202,0.303,1]])
corrHzd = np.array([[1,0.01,0.02],[0.01,1,0.03],[0.02,0.03,1]])
rhoDefault=0.45
rhoSame=0.1
rhoUnrelated=0.05
hazardRates = np.array([0.01,0.0001,0.91,0.055])
samples = np.abs(np.sin( np.cumsum( np.cumsum( np.ones((10000,6*4)),axis=0), axis=1) ))
samples[1:4,1] = 1e-9
corrEq = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrVol = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrHzd = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
T = 1.5/52.
S0 = np.array([70., 60., 150.,75.,60.,55.])
h0 = np.array([0.024, 0.035, 0.025,0.065,0.015,0.035])
sigma0 = np.array([0.1,0.8,0.2,0.2,0.3,0.4])
muE = np.array([0.05, 0.04, 0.06,0.05,0.03,0.02])
muH = np.array([0.0, 0, 0,0,0,0])
eta = np.array([0.0, 0, 0,0,0,0])
sigmaH = np.array([0.23,0.35,0.53,0.39,0.65,0.27])
varvol = np.array([0.36,0.35,0.67,0.31,0.26,0.63])
bondTenors = np.array([6,8,3.6,4,7,6])
bondRiskFreeRates = np.array([0.052, 0.062, 0.022,0.064,0.063,0.031])
recoveryRates = np.array([0.12, 0.15, 0.13,0.65,0.22,0.65])
callput = np.array([-1,-1,1,-1,1,-1])
K = np.array([90., 55., 150, 65,35,85])
r = np.array([0.014, 0.023, 0.032,0.026,0.022,0.039])
optionTenors = np.array([0.21, 0.26, 0.23,0.68,0.32,0.62])
q = np.array([0.0015, 0.0022, 0.0023,0.0036,0.0036,0.0033])
eqPositions = np.array([0,0,0,0,0,0])
bondPositions = np.array([0,0,0,0,0,0])
optionPositions = np.array([6,3,5,7,11,4])
rhoDefault = 0.45
rhoSame = 0.1
rhoUnrelated = 0.05
```

```
corrEq = np.array([[1,0.11,0.22],[0.11,1,0.33],[0.22,0.33,1]])
corrVol = np.array([[1,0.101,0.202],[0.101,1,0.303],[0.202,0.303,1]])
corrHzd = np.array([[1,0.01,0.02],[0.01,1,0.03],[0.02,0.03,1]])
rhoDefault=0.45
rhoSame=0.1
rhoUnrelated=0.05
hazardRates = np.array([0.01,0.0001,0.91,0.055])
samples = np.abs(np.sin( np.cumsum( np.cumsum( np.ones((10000,6*4)),axis=0), axis=1) ))
samples[1:4,1] = 1e-9
corrEq = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrVol = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
corrHzd = 0.25*np.ones((6,6))+(1-0.25)*np.eye(6)
T = 1.5/52.
S0 = np.array([40., 60., 150.,135.,60.,55.])
h0 = np.array([0.024, 0.035, 0.025,0.035,0.015,0.035])
sigma0 = np.array([0.1,0.8,0.2,0.2,0.4,0.4])
muE = np.array([0.05, 0.04, 0.06,0.04,0.03,0.02])
muH = np.array([0.0, 0, 0,0,0,0])
eta = np.array([0.0, 0, 0,0,0,0])
sigmaH = np.array([0.23,0.35,0.13,0.29,0.65,0.27])
varvol = np.array([0.36,0.35,0.57,0.21,0.26,0.63])
bondTenors = np.array([6,8,3.6,4,3,5])
bondRiskFreeRates = np.array([0.052, 0.042, 0.022,0.064,0.033,0.031])
recoveryRates = np.array([0.12, 0.15, 0.13,0.25,0.22,0.25])
callput = np.array([-1,-1,1,-1,1,-1])
K = np.array([90., 55., 150, 25,35,65])
r = np.array([0.014, 0.023, 0.032,0.026,0.032,0.039])
optionTenors = np.array([0.21, 0.26, 0.23,0.28,0.32,0.62])
q = np.array([0.0015, 0.0022, 0.0023,0.0036,0.0023,0.0033])
eqPositions = np.array([0,0,0,0,0,0])
bondPositions = np.array([9,6,5,6,7,10])
optionPositions = np.array([0,0,0,0,0,0])
rhoDefault = 0.45
rhoSame = 0.1
rhoUnrelated = 0.05
```

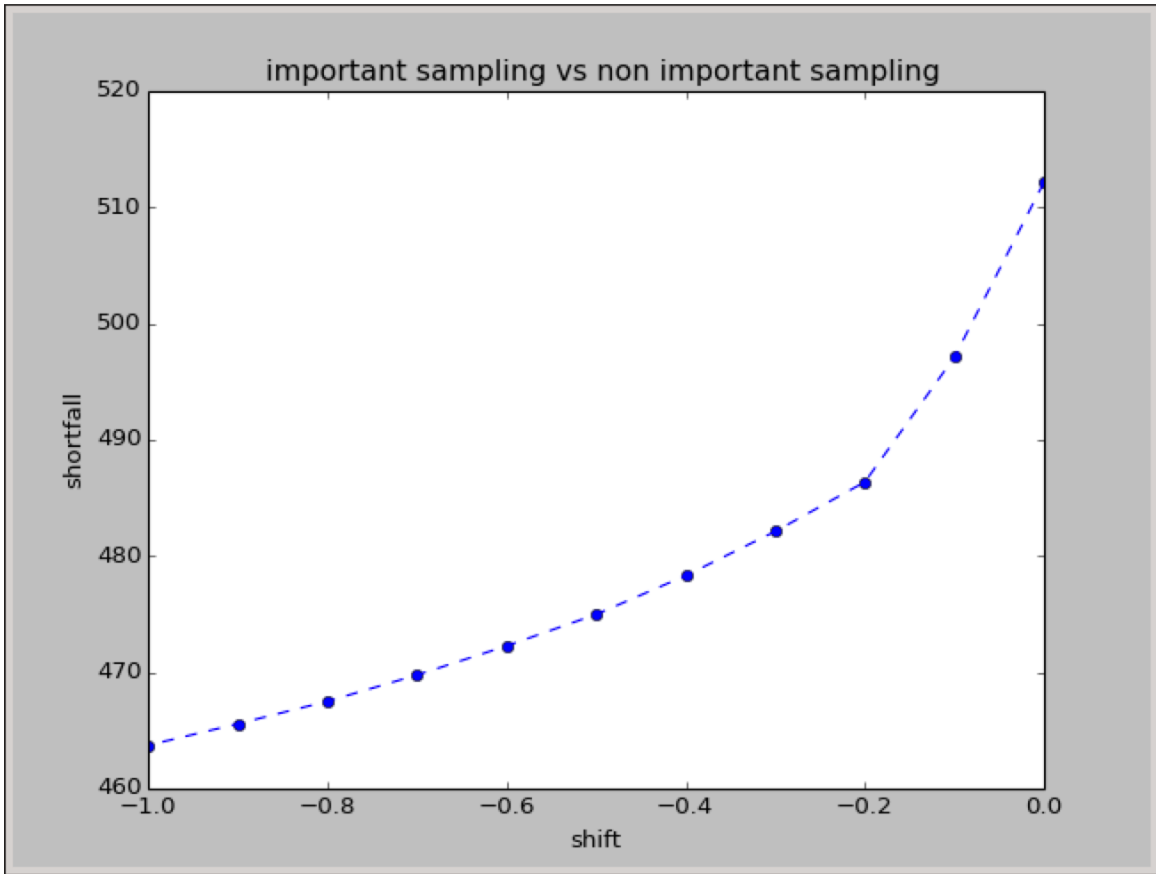
5.Analysis

I tried equation, option and bond:
(1)equation

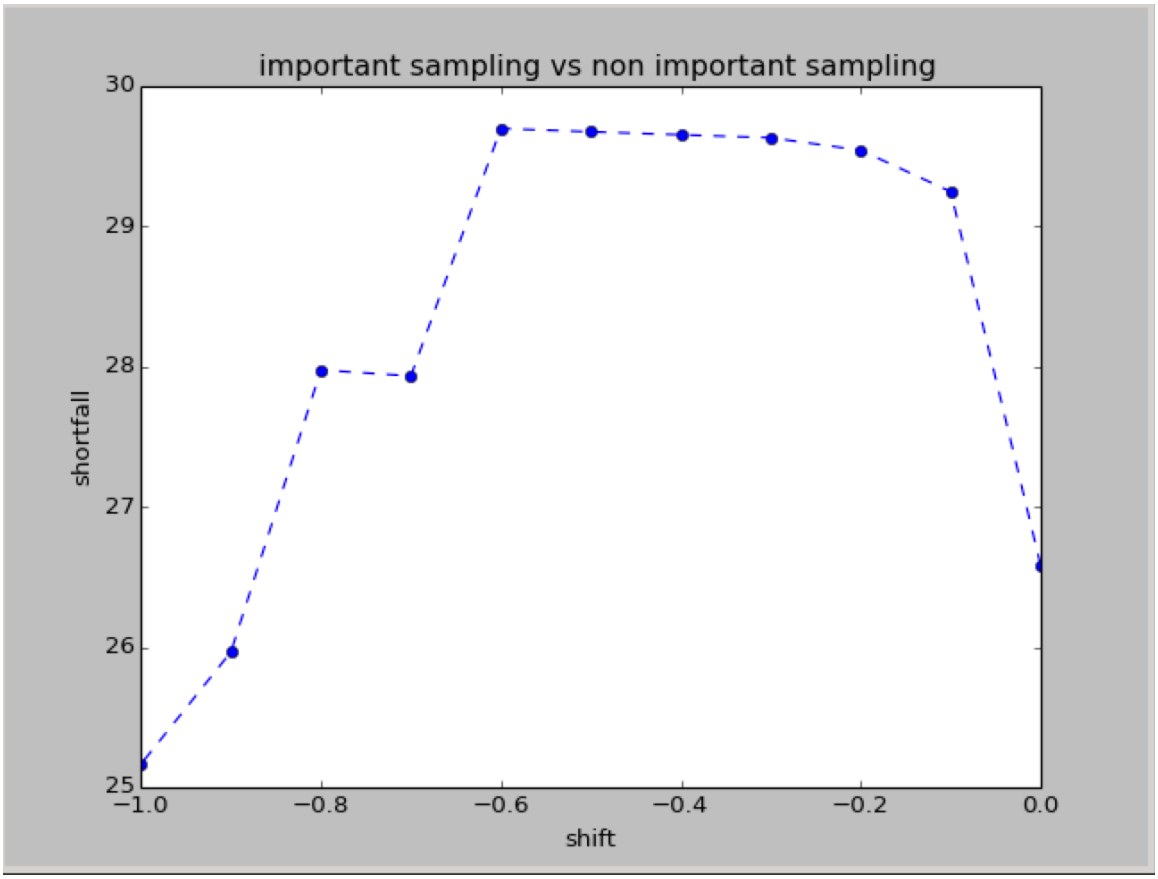
and the expectedshortfalls is showed below:


important sampling vs non important sampling
4700
4600
4500
4400
4300
4200
4100
4000
3900
3800
shortfall
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
shift

(2) option


important sampling vs non important sampling
520
510
500
490
480
470
460
shortfall
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
shift

(3)bond



With the shift goes to -1, the equation graph seems much more stable and convergence to 4500, the option continues goes down, and the bond is close to non-shift shortfall around shift -0.8. When shift is equal to -0.8, it seems that bond and equation performed quite well.

6.Conclusion

It seems that not all the portfolio shortfall is perform better using importance sampling shift.