**MIE1622**

**Assignment 3**

**REPORT**

**By:**

**Qian Zhang**

**1000062736**

**March 24, 2019**

1. **Introduction**

This report investigates three different ways of modelling a credit-risky portfolio consisted of 100 corporate bonds, over a period of 1 year. The three methods are as follows:

1. Monte Carlo approximation using 5000 in-sample scenarios with 1000 systemic scenarios and 5 idiosyncratic scenarios for each systemic
2. Monte Carlo approximation using 5000 in-sample scenarios with 5000 systemic scenarios and 1 idiosyncratic scenario for each systemic
3. True distribution using 100,000 out-of-sample scenarios, which are all systemic

As part of this report, the corresponding VaR and CVaR values for each model is calculated. The normal distribution of each model is also calculated.

1. **MATLAB Outputs**

======= Credit Risk Model with Credit-State Migrations =======

============== Monte Carlo Scenario Generation ===============

Number of out-of-sample Monte Carlo scenarios = 100000

Number of in-sample Monte Carlo scenarios = 5000

Number of counterparties = 100

Portfolio 1:

Out-of-sample: VaR 99.0% = $55133563.30, CVaR 99.0% = $62248780.93

In-sample MC1: VaR 99.0% = $54387175.08, CVaR 99.0% = $61696195.62

In-sample MC2: VaR 99.0% = $54701896.62, CVaR 99.0% = $60321119.57

In-sample No: VaR 99.0% = $42361941.91, CVaR 99.0% = $46955344.15

In-sample N1: VaR 99.0% = $41313173.98, CVaR 99.0% = $41313173.98

In-sample N2: VaR 99.0% = $42369186.15, CVaR 99.0% = $42369186.15

Out-of-sample: VaR 99.9% = $70020037.02, CVaR 99.9% = $76957299.36

In-sample MC1: VaR 99.9% = $69507513.26, CVaR 99.9% = $77282400.23

In-sample MC2: VaR 99.9% = $66842299.20, CVaR 99.9% = $70821320.45

In-sample No: VaR 99.9% = $52716550.75, CVaR 99.9% = $56469414.14

In-sample N1: VaR 99.9% = $51433984.91, CVaR 99.9% = $51433984.91

In-sample N2: VaR 99.9% = $52769252.09, CVaR 99.9% = $52769252.09

Portfolio 2:

Out-of-sample: VaR 99.0% = $42967121.46, CVaR 99.0% = $48311218.15

In-sample MC1: VaR 99.0% = $55435439.45, CVaR 99.0% = $61209553.23

In-sample MC2: VaR 99.0% = $55644839.40, CVaR 99.0% = $61431796.71

In-sample No: VaR 99.0% = $32510601.14, CVaR 99.0% = $35868757.57

In-sample N1: VaR 99.0% = $41278614.14, CVaR 99.0% = $41278614.14

In-sample N2: VaR 99.0% = $42388588.71, CVaR 99.0% = $42388588.71

Out-of-sample: VaR 99.9% = $55317439.76, CVaR 99.9% = $59116827.98

In-sample MC1: VaR 99.9% = $69483520.73, CVaR 99.9% = $73504879.83

In-sample MC2: VaR 99.9% = $66220243.97, CVaR 99.9% = $71570354.26

In-sample No: VaR 99.9% = $40080675.42, CVaR 99.9% = $42824328.52

In-sample N1: VaR 99.9% = $51401993.91, CVaR 99.9% = $51401993.91

In-sample N2: VaR 99.9% = $52761666.61, CVaR 99.9% = $52761666.61

1. **Plots and Graphs**

A screenshot of a cell phone

Description automatically generated

Figure 1. True distribution of portfolio 1

A close up of a map

Description automatically generated

Figure 2. True distribution of portfolio 2

A close up of a map

Description automatically generated

Figure 3. Monte Carlo 1 distribution of portfolio 1

A screenshot of a map

Description automatically generated

Figure 4. Monte Carlo 2 distribution of portfolio 1

A close up of a map

Description automatically generated

Figure 5. Monte Carlo 1 distribution of portfolio 2

A close up of a map

Description automatically generated

Figure 5. Monte Carlo 2 distribution of portfolio 2

1. **Code Overview**

**4.1. Out-of-sample Scenarios (True Distribution)**

% define y with 100,000 systemic scenarios each for 50 correlated credit drivers

y\_MC1 = ones(Nout,Ncd); % size = 100,000 by 50

% define w as 100,000 systemic scenarios each for 100 counterparties

w\_MC1 = ones(Nout,K); % size = 100,000 by 100

% define out\_of\_sample losses as 100,000 systemic scenarios each for

% the losses of 100 counterparties

Losses\_out = ones(Nout,K); % size = 100,000 by 100

i = 1;

for s = 1:Nout

y = (randn(1,Ncd) \* sqrt\_rho)';

% one randomly generated idiosyncratic scenario for each systemic

z\_out = randn(K,1); % size = 100 by 1

% for each of the 100 counterparties

for k = 1:K

% calculate creditworthiness index for each scenario and counterparty

w(k) = beta(k) \* y(driver(k)) + sqrt(1-beta(k)^2) \* z\_out(k);

% combine w with CS boundaries as a vector in ascending order

cs\_axis = sort([w(k) CS\_Bdry(k,:)]);

% returns the position of w amongst the CS boundaries

cs\_position = find(cs\_axis == w(k));

% calculate losses for each counterparty based on the position

% of w in exposure

Losses\_out\_temp(k) = exposure(k,cs\_position);

end

% store out-of-sample losses, size = 100,000 by 100

Losses\_out(i,:) = Losses\_out\_temp;

i = i+1;

end

Since systemic scenarios require the generation of random numbers according a pre-defined correlation between the 50 credit drivers, matrix y is of size 100,000 by 50. For each y, a random value of z is generated, which is not correlated. Credit worthiness w is calculated using the formula given in class and is calculated for each counterparty. After w is calculated, it is sorted into ascending order alongside the credit-state boundaries, amongst which the location of w is found. The location of w is then returned to the exposure matrix, and the corresponding loss for each counterparty is determined.

**4.2. In-sample Scenarios (Monte Carlo 1)**

for s = 1:ceil(Nin/Ns)

y\_MC1 = (randn(1,Ncd) \* sqrt\_rho)';

% 5 idiosyncratic scenarios for each systemic

for si = 1:Ns

z\_MC1 = randn(K,1);

% 100 counterparties

for k = 1:K

% calculate creditworthiness index for each scenario and counterparty

w\_MC1(k) = beta(k) \* y\_MC1(driver(k)) + sqrt(1-beta(k)^2) \* z\_MC1(k);

% combine w with CS boundaries as a vector in ascending order

cs\_axis = sort([w\_MC1(k) CS\_Bdry(k,:)]);

% returns the position of w amongst the CS boundaries

cs\_position = find(cs\_axis == w\_MC1(k));

% calculate losses for each counterparty based on the position

% of w in exposure

Losses\_inMC1\_temp(k) = exposure(k,cs\_position);

end

% store MC1 losses, size = 5,000 by 100

Losses\_inMC1(i,:) = Losses\_inMC1\_temp;

i = i + 1;

end

end

Similar to the calculation for the true distribution, y is generated using 50 correlations. However, in this case, 5 idiosyncratic scenarios (z) is calculated for each y, and the counter party losses for each combination of scenarios is calculated. The total number of scenarios is 5000, where y = 1000, z = 5.

**4.3. In-sample Scenarios (Monte Carlo 2)**

i = 1;

% 5000 systemic scenarios (1 idiosyncratic scenario for each systemic)

for s = 1:Nin

y\_MC2 = (randn(1,Ncd) \* sqrt\_rho)';

z\_MC2 = randn(K,1);

% for each of the 100 counterparties

for k = 1:K

% calculate creditworthiness index for each scenario and counterparty

w\_MC2(k) = beta(k) \* y\_MC2(driver(k)) + sqrt(1-beta(k)^2) \* z\_MC2(k);

% combine w with CS boundaries as a vector in ascending order

cs\_axis = sort([w\_MC2(k) CS\_Bdry(k,:)]);

% returns the position of w amongst the CS boundaries

cs\_position = find(cs\_axis == w\_MC2(k));

% calculate losses for each counterparty based on the position

% of w in exposure

Losses\_inMC2\_temp(k) = exposure(k,cs\_position);

end

% store MC2 losses, size = 5,000 by 100

Losses\_inMC2(i,:) = Losses\_inMC2\_temp;

i = i+1;

end

In MC2, 1 idiosyncratic scenario (z) is generated for each of the 5000 systemic scenarios (y), and the corresponding counterparty loss is calculated for each scenario.

Both MC1 and MC2 are calculated in 100 trials, where 100 unique sets of scenarios are generated for each of the two Monte Carlo distributions.

1. **Questions and Discussions**

**5.1. Analyzing Sampling and Model Errors.**

NOTE: all VaR and CVaR values for MC1 and MC2 are the averaged results over 100 trials.

Table 1. Sampling errors between MC approximations and true distribution, for each portfolio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Portfolio 1** | **MC1** | **MC2** | **True** | **MC1 Sampling Error** | **MC2 Sampling Error** |
| **VaR (99%)** | $54364850 | $55238947 | $54706871 | -0.62% | 0.97% |
| **CVaR (99%)** | $63004601 | $63202287 | $62074665 | 1.50% | 1.82% |
| **VaR (99.9%)** | $71091163 | $70148783 | $70457223 | 0.90% | -0.44% |
| **CVaR (99.9%)** | $79392084 | $77292967 | $76967676 | 3.15% | 0.43% |
| **Portfolio 2** | **MC1** | **MC2** | **True** | **MC1 Sampling Error** | **MC2 Sampling Error** |
| **VaR (99%)** | $56024568 | $53066195 | $43106302 | 29.97% | 23.11% |
| **CVaR (99%)** | $65602726 | $59977148 | $48577143 | 35.05% | 23.47% |
| **VaR (99.9%)** | $78213731 | $66799114 | $55145994 | 41.83% | 21.13% |
| **CVaR (99.9%)** | $84794700 | $70502700 | $58896062 | 43.97% | 19.77% |

Table 2. VaR and CVaR model errors between normal models and true distributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Portfolio 1** | **True (Normal)** | **True** | **Model Error** |
| **VaRN (99%)** | $42215165 | $54706871 | -22.83% |
| **CVaRN (99%)** | $46794697 | $62074665 | -24.61% |
| **VaRN (99.9%)** | $52538507 | $70457223 | -25.43% |
| **CVaRN (99.9%)** | $56280038 | $76967676 | -26.88% |
| **Portfolio 2** | **True (Normal)** | **True** | **Model Error** |
| **VaRN(99%)** | $32523967 | $43106302 | -24.55% |
| **CVaRN (99%)** | $35885920 | $48577143 | -26.13% |
| **VaRN (99.9%)** | $40102601 | $55145994 | -27.28% |
| **CVaRN (99.9%)** | $42849356 | $58896062 | -27.25% |

As seen in Tables 1, the sampling errors between the two Monte Carlo approximations and the true distribution are close to zero for portfolio 1. MC1 is observed to have slightly larger sampling errors as compared to MC2. This is expected since MC1 uses less systemic scenarios (1,000) than the MC2 (5,000), whereas the true distribution uses 100,000 systemic scenarios. The sampling errors are small because the MC approximations are calculated using a relatively large number of scenarios (5,000). In the case of MC2, it is half the amount of true distribution in terms of the number of scenarios.

On the other hand, for portfolio 2, the sampling errors are larger and varies between approximately 20% and 44%, which means that the VaR and CVaR obtained by the MC approximations are significantly worse than that of the true distributions (shifted to the tail-end as seen in Figures 2, 3, and 6). The loss of portfolio 2 is in fact much less than that of portfolio 1, but this is inversely reflected in the MC approximations. This means that although the equally weighted portfolio performs better than the equal-valued portfolio, it cannot be accurately depicted by MC approximations due to sampling shortcomings.

As seen in Table 2, the model errors of the true distribution for both portfolios are around the same range (approximately -25%). This is expected since normal distributions are known to underestimate the actual VaR and CVaR values, shifting to the left on the loss distribution plot as a result (seen in figures 1 to 6).

**5.2. If you report the in-sample VaR and CVaR to decision-makers in your bank, what consequences for the bank capital requirements it may have?**

Since the in-sample VaR and CVaR are closely aligned with the true distribution for the first portfolio, decisions made on the capital requirements based on the in-sample values will be relatively accurate. However, the in-sample VaR and CVaR values for portfolio 2 greatly over-estimates portfolio loss, especially in the case of MC1. Therefore, in the case of portfolio 2, the in-sample VaR and CVaR losses may cause the bank to over-allocate assets to its capital requirements, taking less risk than it could have.

**5.3. Can you suggest techniques for minimizing impacts of sampling and model errors?**

The impact of sampling error and/or model error can be reduced by the following techniques:

1. Reducing the error itself
   1. Sampling error can be reduced by increasing the number of scenarios, preferably systemic scenarios, even though this can lead to longer calculation times.
   2. Modelling error in the case of the normal distribution model is induced because the true distribution is not normal. Although normal distribution comes close to the true distribution especially for short-term investments, the normal model tends to under estimate VaR and CVaR. As such, we increase VaR and CVaR by increasing the standard deviation of the normal distribution, making the tail-end “fatter” as a result.
2. Since sampling and model errors are imminent with any type of statistical and modelling analysis, an investor would know better not to entirely trust the results generated. To reduce the impact of these errors, the investor should treat the VaR and CVaR values as a range of values instead of a single number. This would allow for better tolerance against errors.