

SABER AMINI

CREDIT RISK MODELING AND SIMULATION

UNIVERSITY OF TORONTO PRESS

1 Credit Risk Modeling

1.1 Introduction

The general approach in portfolio modeling is to assume some kind of distribution for your financial instruments, then do some kind of portfolio modeling to get the distribution of your overall portfolio.

Credit risk modeling involves modeling of assets that have potential to go to zero. The obvious example is bonds but can also included derivatives. Bonds do not appear much in news and certainly are not as hot as stocks, but in reality, they play a major part in finance. This section looks in more detail on how we model bonds.

To start, let's look at the bond prices of some major companies during the 2008-2009 financial crisis.

Issue	March 2008		March 2009		Δ Price (\$)
	Price (\$)	Yield (%)	Price (\$)	Yield (%)	
Ford 6.5% 08/01/18	70	11.5	16	45.6	-54
GM 7.7% 04/15/16	82	11.0	13	66.4	-69
Target 6.0% 01/15/18	103	5.6	100	6.0	-3
Walmart 5.4% 04/05/17	103	4.9	105	4.5	2

What can observe from this table?

- Automotive bonds lost about 80% of their value in one year
- Bonds of discount retailers retained their value

We can therefore say that the market was less confident that automotive companies would be able to make the required interest and principle payments. This is therefore an example of credit risk, the expectation of default for that counterparty that you are holding.

Based on the above example, the formal of definition of credit risk is as follows. **Credit risk** is the risk of monetary loss due to the default

or a change in the perceived likelihood of default, of a counterparty to a contract.

So clearly, knowing the likelihood of default is very important for us in terms of modeling credit risk. How can we model the likelihood of default?

One way to assess the likelihood is through **credit ratings** of the counterparties. These credit ratings are assigned by various credit rating agencies such as S&P, Moody's, Fitch, DBRS etc. The lower the rating is, the more compensation is required from that company or government either in form of *more interest or more collateral in getting a loan*. So generally, if there is more perceived likelihood of default, you must provide more collateral or pay more interest on a loan, or a combination of both.

Generally, even when there is a default on a bond, we can get some of our capital back through collateral, coupons, bankruptcy proceedings etc. On average, after a default, bond holders get 40-60% of their original loan back, so defaults do not mean zero return.

1.2 Credit Rating Agency Designations

The following table shows the rating scales among major agencies. The ratings can be divided into investment grade, and non-investment grade.

Moody's	S&P and Fitch
Aaa	AAA
Aa	AA
A	A
Baa	BBB
Ba	BB
B	B
Caa	CCC
Ca	CC
C	C

Investment grade bonds

Non-investment grade bonds

For us, in the examples given, we will ignore CC and C ratings because

these are very close to default anyways.

1.3 Credit Risk Transition

Now the previous section gave us a good start on how we can model the credit risk of a counterparty credit risky financial instrument. However, in general, the current risk rating alone is not sufficient for modeling.

So credit loss value is not only due to the perceived default but also to the change in the credit rating. So if your company is downgrade from let's say AA to A, then the value of the bond in that case goes down. In the same way, we can improve if the value of the credit rating is upgraded.

Therefore, we need to quantify not just the credit rating of the bond, but also the probability of the credit rating changing from one rating to another. For this, we form what is called an **Transition Matrix** which specifies the likelihood of migrating from one credit rating (state) to another over a fixed time horizon usually a year.

An example of an annual transition matrix is shown below:

From \ To	AAA	AA	A	BBB	BB	B	CCC	Default
AAA	92.18	7.06	0.73	0.00	0.02	0.00	0.00	0.01
AA	1.17	90.84	7.63	0.26	0.07	0.01	0.00	0.02
A	0.05	2.39	91.83	5.07	0.50	0.13	0.01	0.02
BBB	0.05	0.24	5.20	88.49	4.88	0.80	0.16	0.18
BB	0.01	0.05	0.50	5.45	85.12	7.05	0.55	1.27
B	0.01	0.03	0.13	0.43	6.52	83.20	3.04	6.64
CCC	0.00	0.00	0.00	0.58	1.74	4.18	68.00	25.50
Default	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00

What this is showing is that for example, if our current rating is AAA, then the probability of our rating staying at AAA one year from now is about 93%, the probability of it going down to AA is 7.06% and so on. Notice that the highest probability for any rating is to stay at the same rating for next year. Also notice that if you are in default, then you will stay at default, there is no chance of a company that is in default getting back to any rating.

1.4 Calculating Credit Losses

So now that we have the general preliminaries, we want to see how we can calculate and model credit losses. So as before, since credit losses, just like any other financial instrument future pricing, is an uncertainty, we will "measure" the loss through a probability distribution.

So we need to essentially come up with a relationship between the credit risk and the probability of change in that credit risk and the underlying price of the credit risky financial instrument, for example a bond.

Our basic model is quite simple. First of all, we assume for simplicity that bond value only depends on credit rating. We then assume that associated with each future credit state there is an accompanying change in the monetary value of the contract. For example a BBB-rated bond that is worth \$100 today may, one year from now, be worth \$92 if the issuer is downgraded to BB or \$104 if the issuer is upgraded to AA rating.

So the the **counterparty loss (L)** has a **discrete distribtuion (F_L)**. For example, assume we have a bond with a current BBB rating, then the future loss can be modeled as shown below ¹

¹ positive values mean losses

	AAA	AA	A	BBB	BB	B	CCC	Default
Loss per \$1	-0.07	-0.06	-0.04	0.00	0.08	0.20	0.45	1.00
Probability (%)	0.05	0.24	5.20	88.49	4.88	0.80	0.16	0.18

Notice that if the credit rating does not change, then we lose nothing. If it improves, then we gain in our price and if it goes down we lose on the bond price. Note that in this model, we assume 100% loss for a default but you can adjust for that if you expect to salvage some value through collateral or other means.

1.5 Real world examples of Credit Loss Data

To give you an idea of how you normally calculate credit loss, we will look at the data sets that companies use when evaluating portfolios of credit risky instruments. The following is a the data for some credit risky instruments which gives the general characteristic of the instru-

ments. We have highlighted one company. We can see that the current rating for this company is BB. We can also see that in case of default, we can recover 70% of the value of the bond. Standard deviation of recovery rate gives you likelihood of getting that 70 percent. We will not recover 70 percent for sure, that is just a mean, so standard deviation gives you what is the 63% probability to recover within that mean.

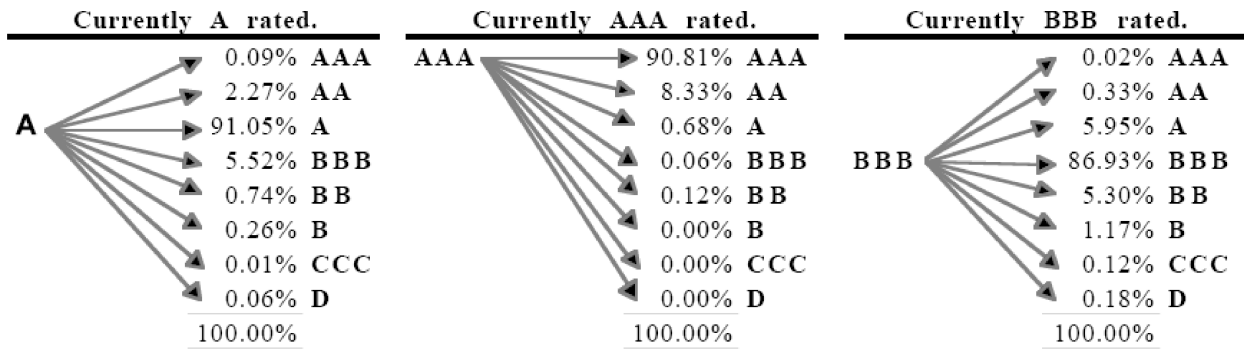
Generally we do not take into account standard deviation in examples given, but you can take it into account when doing more precise modeling or when a lot of money is at stake.

Counterparty		Credit Driver				Recovery after Default			Mkt. Return
Group #	CP ID	Credit Driver	CDIDX	Sensitivity Vector	Credit State	Recovery Mean	Recovery Standard Dev	Value	E[Return]
1	USBUSIN1234	DJUSBM	1	0.5568	A	0.62	0.3567	10,255,741	7.0315%
1	USPAPER1234	DJUSBM	1	0.5568	A	0.6	0.3600	9,997,200	7.0435%
129	FRANSTEEL1234	E2BSC	21	0.5399	BB	0.7	0.3368	49,911,974	10.4248%
203	JAPAHOTEL1234	P1CYC	32	0.5154	CCC	0.62	0.3567	5,466,820	36.1575%
271	AUSTREAL1234	P2FIN	44	0.5380	BBB	0.66	0.3481	4,993,662	8.2542%

The following is the transition matrix for the above assets.

Counterparty		Credit State Migration Probabilities							
Group #	CP ID	Pr(Default)	Pr(CCC)	Pr(B)	Pr(BB)	Pr(BBB)	Pr(A)	Pr(AA)	Pr(AAA)
1	USBUSIN1234	0.0002	0.0001	0.0013	0.005	0.0507	0.9183	0.0239	0.0005
1	USPAPER1234	0.0002	0.0001	0.0013	0.005	0.0507	0.9183	0.0239	0.0005
129	FRANSTEEL1234	0.0127	0.0055	0.0705	0.8512	0.0545	0.005	0.0005	0.0001
203	JAPAHOTEL1234	0.255	0.68	0.0418	0.0174	0.0058	0	0	0
271	AUSTREAL1234	0.0018	0.0016	0.008	0.0488	0.8849	0.052	0.0024	0.0005

We can also show the above graphically:



Finally, below you can see the loss matrix for the assets if the ratings changes from the current to one of the various ratings shown.

Counterparty		Losses in Credit States							
Group #	CP ID	L(Default)	L(CCC)	L(B)	L(BB)	L(BBB)	L(A)	L(AA)	L(AAA)
1	USBUSIN1234	10,255,741	503,640	201,280	59,625	11,484	0	-500	-1,500
1	USPAPER1234	9,997,200	939,798	398,956	124,925	23,465	0	-3,250	-5,000
129	FRANSTEEL1234	49,911,974	3,571,706	1,283,666	0	-533,837	-677,873	-698,786	-710,178
203	JAPAHOTEL1234	5,466,820	0	-230,494	-375,416	-440,986	-458,134	-461,911	-463,096
271	AUSTREAL1234	4,993,662	328,776	169,824	54,867	0	-15,392	-18,599	-19,623

1.6 Factor Modeling of Credit Risk

As we mentioned, we will assume for our model that bond prices only depend on credit risk. And to model credit risk, we will use factor models. Specifically, a single factor model. For our model, the only factor will be in which industry and in which country the coun-

terparty issuing the credit risky instrument is located in. So for our example above that was highlighted in red, we have the credit driver "E2BSC". This means that the company is in the European Union (E) and in the basic material sector (BSC). The value "sensitivity vector" is essentially the β factor and this is basically saying that 54% of the attributes this company can be attributed to movements in the European basic material sector. ²

² using a single factor model for bond losses is actually an industry standard. You can do multifactor models if you like.

1.7 Correlation Matrix Between Credit Drivers

So what else do we need? We also need the correlation matrix between the credit drivers, those country industry indices. This is because if we are going to use random numbers to create scenarios, then we need to make sure that if there is a relationship between different counterparties, that these are taken into account. For example, it would be wrong to assume that one counterparty's rating is downgraded and another is upgraded, if there is any correlation between these two counterparties.

So for example assume you have three credit drivers: US chemicals, German Insurance, German banking. Then you can say perhaps the following correlation between these drivers. So for example, the correlation between US chemicals and german banking is only 8% but for german banking and german insurance it is 34% which makes sense since they are in the same company and similar industries.

Index	Correlations		
	U.S. Chemicals	Germany Insurance	Germany Banking
U.S.: Chemicals	1.00	0.16	0.08
Germany: Insurance	0.16	1.00	0.34
Germany: Banking	0.08	0.34	1.00

1.8 Structural model of portfolio credit risk

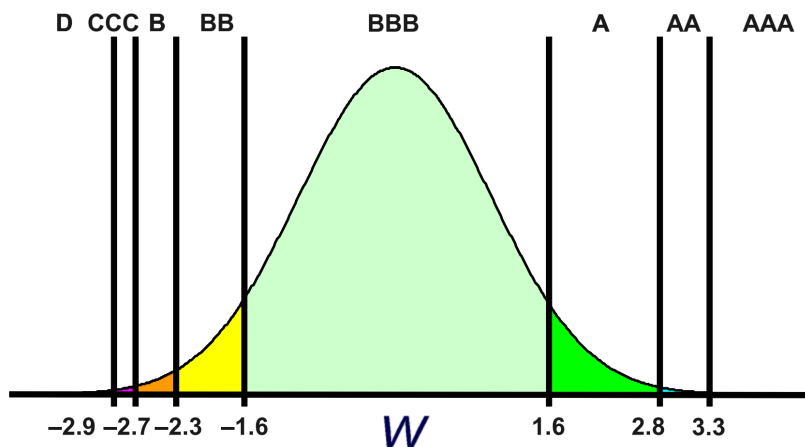
So first of all let's recap, what do we want? We want to know if we have a portfolio of credit risky assets such as bonds, what will the loss distribution of that portfolio look like. Note that previously, we never

looked at credit risk, only at market risk so this is somewhat new.³

³ Our underlying assumption was that credit risk was zero

Now the model for our portfolio credit risk will be a factor model. And we want to know what happens with the migration of credit worthiness (ratings) 1 year (or whatever time period you choose) from today. Since we don't know the future, as usual we model the future with a random variable w , which we call **the creditworthiness index**. What does that random variable represent?

So assume that we generate a random number and it falls somewhere between -1.6 to 1.6, then this bond will stay at it's current rating, for now assumed to be BBB.



However, before generating that random variable, we need a model for it. So as it is done in general for factor models, we divide the total credit risk into systemic and specific risk. Systemic risk depends on which country and which industry that company belongs. This is modeled by the credit drivers that we just talked about in the previous section. And the second part of the model is the specific risk ⁴ which is independent on any of the risk factors.

⁴ which we have seen in the capital asset pricing model

So if we want to compute w_j where j is the index of our company, we must compute:

$$w_j = \beta_j y_{j(k)} + \sigma_j z_j$$

where we have:

- β_j sensitivity of counterparty j to the credit driver. For the example

above it is 0.54. So this means 54% of the changes in credit worthiness of this company can be attributed to being in that country or industry (ie related to our factor)

- $y_{j(k)}$ is the one factor in our model. Note that j is the counterparty and k is the industry. So for our example above of the french steel company, j is 129 and k is the CDIDX number 21. This random variable depends on how good the situation is in that country and in that index. So the *larger this value* the bigger the chance that our company is performing well.

Y is generated assuming a random distribution with a correlation that you get from your data (again you cannot assume that systematic risk is independent across all companies)

Note that y and z are random variables so we need to make some assumptions on the distribution of those random variables. In this model the assumption is that z is a *standard normal random variable*. This means that it has a mean of 0 and standard deviation of 1. We denote this usually with $N(0,1)$. Y is also a standard normal variable but y 's are correlated to each other because we know that there is a correlation matrix between credit drivers. Different countries and different industries are related to each other (some move together, some opposite) and we capture that through the distribution of y .

We now are in a process to start our simulation modeling. So we do not know w unless we sample. So we generate random numbers for z and y , multiply by β and σ and compute w_j . We then need to look in which range that value w_j belongs and that tells us how does the credit worthiness of our counterparty changes in that specific scenario. So we need to generate random numbers as before, but process is a bit more involved.

Let's go over the process. Assume we are in scenario 1. Then we generate random numbers for z for company j assume here it is the French company we have in examples above. In this case, then we

$$\beta_{129} = 0.54$$

$$\sigma_{129} = \sqrt{1 - \beta_{129}^2} = 0.842$$

So for the above scenario, we then generate random variable z_j so assume it is equal to 0.1. Now we generate random variable for y_i but remember that the random variable is correlated random numbers so we need to take into account the correlation matrix.

So let's say you calculated $y_{j(k)} = 1.2$. After you substitute everything into the formula and you get a w equals to some value -0.5.

Then you look at our distribution and see that since this falls in the fat part of our curve, there is no change in credit rating and therefore there is zero loss on the credit risky asset we are holding (ie a bond). Therefore, after every scenario generation you know the associated loss.

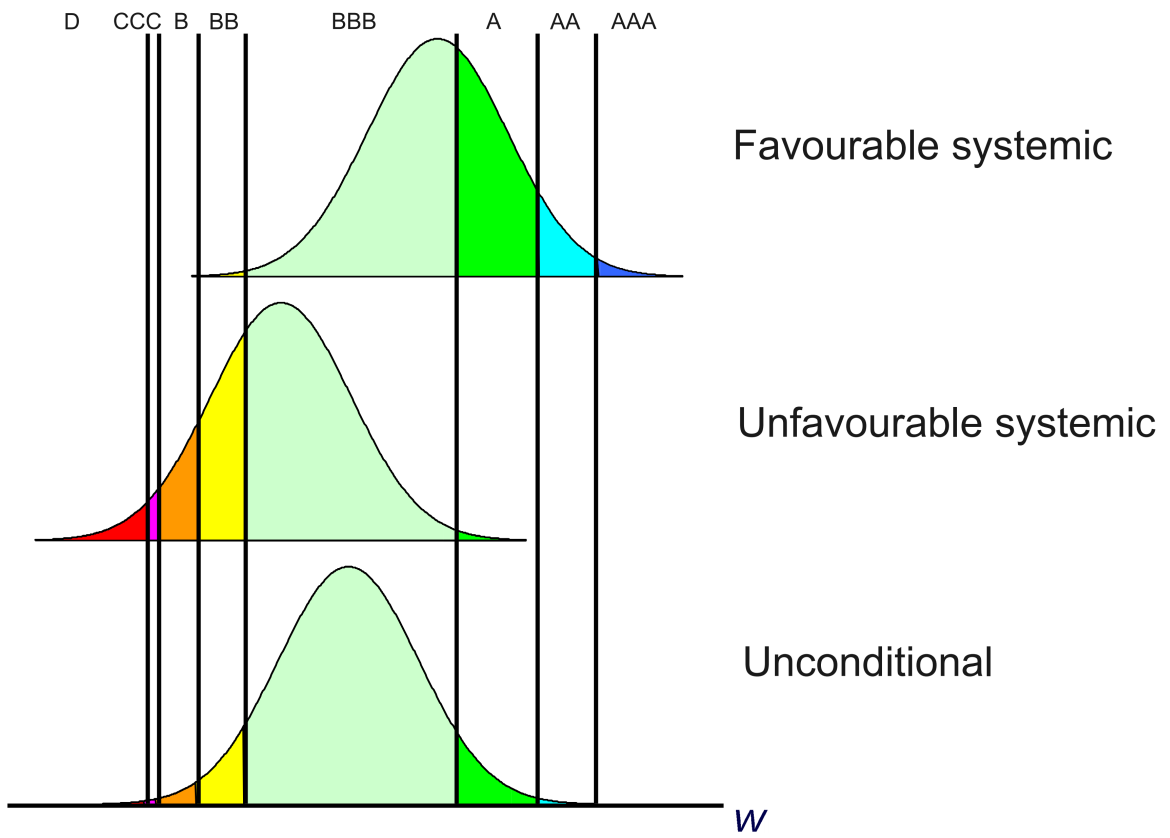
1.8.1 *Meaning of random variable y*

Now there is something subtle that we did in the previous section that we need to emphasize further here. y is a normal random variable with the mean zero. If the random variable y is greater than zero, what does it tell us about that country and industry? Is that country doing good or doing bad? Well the bigger the y value, then the bigger the w value will be and more chance that our change in credit rating will be an improvement and the price will increase.

So what this means is that as soon as we generate y , the distribution of in this formula is shifted to the right if y is greater than zero (favourable systematic risk) and to the left if y is less than zero (unfavorable systematic risk).

So during the 2008-2009 crisis the systematic risk was substantially unfavorable, hence the chance of default went up. Curves that only have z are called conditional transition probabilities - conditional because they are only dependent on value of y .

Conditional transition probabilities for BBB



1.8.2 Conditional Independence Framework

So what happens when you start generating scenarios for this model? So you start with generating random numbers for y and y_j 's are correlated to each other. So it's usually beneficial to generate all the y_j 's in one shot. So in the our example you have 50 different credit drivers - so your index is $k = 1$ to 50. So usually what you do is generate y for those 50 values with a normal distribution ⁵ and correlation matrix C . What happens next, when you start generating vector w for each instrument j , you need to know which y to take for that vector w . For instance, let's say that our counterparty $j = 10$ and it belongs to the country-industry 3 (this is the credit driver factor that affects our counter party $j = 10$). In that case then what you do is take the third coordinate of y and substitute it in your formula. Because your counter party j depends on factor k . You then generate a random variable z , this is uncorrelated so easy to generate.

⁵ $N(0,1)$

So first thing we observe is that it is much more computationally intensive to generate random values for y than z because y values are correlated with each other. So one way to reduce our computational method is ⁶ to generated it once and afterwards, for each y you generate z 5 times, and that gives you 5 different values of w . So this is one thing that you often need to do in any field involving simulations. Can you simplify the computational time and get the precise result.

⁶ And this is what we do in the example below

So for scenario one, we went and generated vector y of correlated random numbers. We then need to calculate w_1 for scenario 1. So we just get y and z and get a value for w . But now in scenario 2, we keep the same y but get new z . We repeat this 5 times, each time keeping the same y but generating new z .

Then in scenario 6, what you do is generate a new vector y which you use in the next 5 scenarios. So this simplifies your computation. And in the example, that's what exactly we will test. What happens if you generate a new y in every scenario and what happens if you generate a new y every 5 scenarios. How less precise your results become. Think of simulations as a way for you to test your hypothesis, if you are waiting three days for simulations to finish and you then realize that you misplaced a matrix somewhere, then that's a big waste

of time. So you need to have the skill to reduce simulation times. Once you have a model that you are fairly confident in, you can run the simulation for few days just to get a final result.

So in the example below, we need to generate two sets of scenarios. One set of scenarios contains 100,000 samples and we call this **out of sample scenario**⁷ because 100,000 is quite a large number, we can see it as a true distribution of portfolio losses. In this case, we generate y 100,000 times and z 100,000 times.

⁷ Also known as "in reality"

The second set of scenarios is called **in sample scenario** and contains 5,000 scenarios. Why we call it "in-sample"? The word "in-sample" means we are not generating enough scenarios (samples) to get a true distribution. We do this cause we want to reduce computational time. We also go a little further to reduce computational intensity. For those 5,000 scenarios, we generate y only 1000 times, and for each y we generate z 5000 times, so total is 5000 scenarios. However, we will likely lose something in the quality of our distribution and this is something you need to compare.

Using the above two techniques, we will be able to evaluate what we call the "sampling error". We will also look at what the error would be if we do "normal" distribution rather the distribution given by the scenario generations.

So the purpose of the example is two fold, measure risk through $CvaR$ and VaR and calculate

1. How much risk is underestimated or overestimated with small samples and
2. How much risk is underestimated or overestimated assuming normal distribution.

1.8.3 Finding true distribution

Note that **portfolio loss distribution conditional on y_m** is the convolution of conditional counterparty loss distributions:

$$F_{L(x)|y} = F_{L^1x_1|y} * F_{L^2x_2|y} * \dots * F_{L^Jx_J|y}$$

This means that if we have 8 credit states, J counterparties, then

there is 8^J possible portfolio losses for each y .

The unconditional portfolio loss distribution is the mixture of the conditional portfolio loss distributions:

$$F_{L(x)}(l) = \frac{1}{M} \sum_{m=1}^M F_{L(x)|y}(l)$$

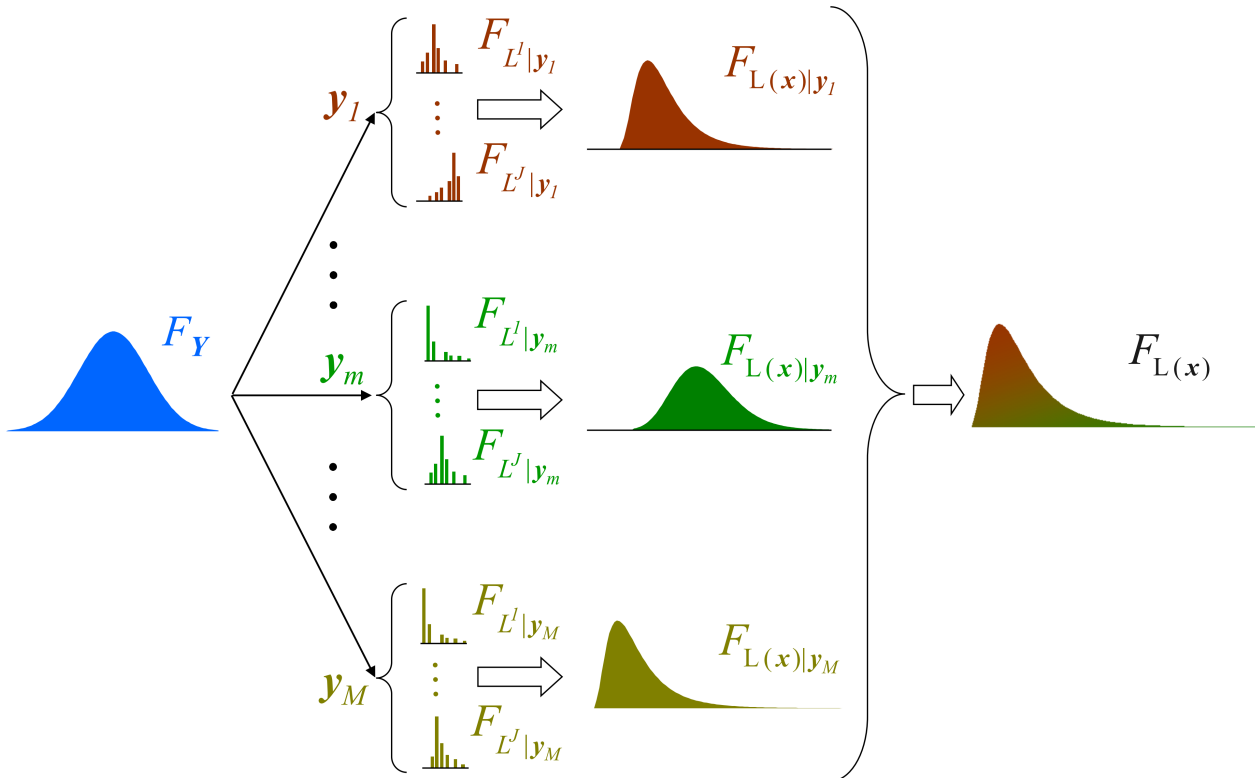
And as we said, the **approximations to conditional loss distribution** $F_{L(x)|y}$ is to use either 1) Monte Carlo sampling (MC) or Conditional Normal (CN) \rightarrow ie assume a normal distribution and accept that the error is not that big.

1.8.4 Overall Summary

So the following bullet summarize the steps we take for finding the structural model of portfolio:

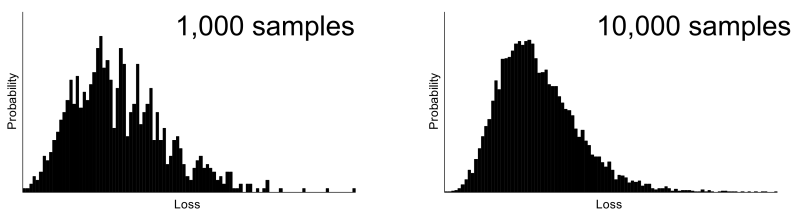
- we generate scenarios for the random variable y from the normal distribution with the correlation matrix C . Remember that y is a random vector of random variables corresponding to each of the credit drivers. In the figure below the number of times you generate y is given by M . For each M , you will get a vector of size k , k being the number of drivers.
- We generate a random variable z . This variable is not conditional.
- compute w_i for *each* credit risky asset i that can be in our portfolio. We do this by taking the credit driver factor that corresponds to the company, multiplying by β_i and adding z_i .
- Now realistically, we need to generate a new y vector every time we calculate a new w_i . But this is computationally intensive, so we have some tricks that lead to what is known as sampling error. We explore this in the examples at the end of this chapter.
- Either way, once we have generated all our losses based on our model for all our credit risky assets, we multiply these losses for each company by vector x which represents our weights in these assets. Based on this we get our *portfolio of conditional losses on a given y* . This is denoted by $F_{L(x)|y_1}$ where x is your portfolio weights.

- Once we generate enough separate conditional losses, we sum up all the conditional portfolio distribution losses to get the unconditional distribution of losses for our portfolio



1.8.5 A note on sampling error and number of samples needed

Generally, sampling error gets worse as $\alpha \rightarrow 1$. For common credit risk analysis, $\alpha \geq 0.995$ so to get good estimates of VaR_α and $CVaR_\alpha$ we would need a lot of samples.



1.8.6 Calculating Losses

As we mentioned previously, finding the true loss distribution requires convolution. Monte Carlo gives you an approximation. So given vector x which represents the positions we are taking in each counterparty⁸, then in order to calculate the losses for each portfolio you need to do is sum up for each counterparty the number of units, multiplied by the per unit loss for that counter party that you get from the scenarios. So how is that going to be done, that is going to be done first we generate all these random numbers z and y , we create the creditworthiness variable w , after we check in which interval that w belongs, that tells you in which credit rating you are in now, and for each of the credit ratings you look up what are the losses in that credit rating. You multiply that loss by the number of units of the counterparty in your portfolio, sum up over all portfolios⁹ and you will get *an approximation* to the portfolio of loss distributions.

⁸ In the example, this is fixed, we are not doing any optimization, just analysis on the portfolio

⁹ ie all your scenarios

Now the formula we have is as follows:

$$F_{L(x)}(l) \approx F^{MC}(l; x) = \frac{1}{MN} \sum_{m,n} 1_{L(y_m, z_{mn}; x) \leq l}$$

This formula looks complex, you may implement it differently. But what does it tell us? 1_B is an indicator function and it is equal to 1 when w is in the range of c where c is a specific credit rating and it is equal to zero if w_j is not in that range. So remember you must sum up over all credit states (Default, CCC, B, BB, BBB, A, AA, AAA). So c takes eight different values and we know what is the loss in each of those credit stats given by vector l_c^j . So we need to determine in which range our w belongs and so we will have a vector that will be zeros and only 1 in the position that the credit loss occurs.

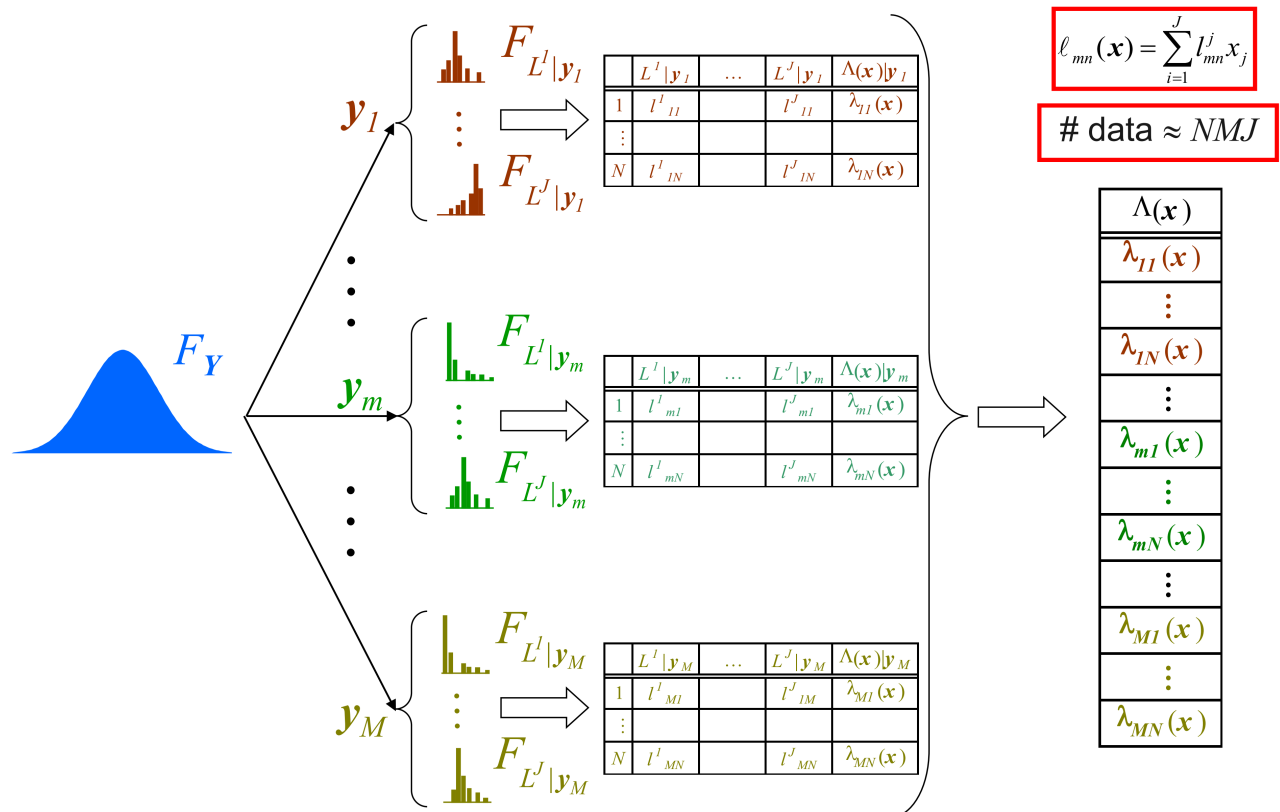
So for example, let's say in one scenario w is equal to -2.5 . So now, we check that -2.5 is in range B, so our bond has migrated from BB to B. So now, we must do the sum. The default is $-\$49$ million, multiplied by indicator function which would be zero plus, triple C loss ($\$3.5$ million) times indicator function (again zero), and so on. The only non-zero is in the B-rating where the loss $-\$1.2$ million.

Note that the function B gives you the vector of numbers corresponding to the ranges.

1.8.7 Monte Carlo Sampling Approximation

So you start off by generating random variable y in scenario $M = 1$ which is a vector of 50 random variables

Next you need to determine loss for instrument 1 to whatever. In order to calculate loss, you need to use the factor model, generate w , figure out what is your migration, look up the loss in that credit state. Remember that you may generate z multiple times for each y because computing independent random variables is much easier computational than generating correlated random numbers. So this is a simplification and this is shown in the Fig. 1.1.



In the example, capital N is either 1 or 5 which means that we can use the same y up to 5 times to make our algorithm more computation-

Figure 1.1: Monte Carlo Simulation with reduced computational intensity by reusing y variables

ally efficient. So for counterparty one, you will have 1 – 5 losses for different z . We denote this by l_{11}^1 which mean the loss for counterparty one for $M = 1$ and $N = 1$.

So the portfolio loss in one scenario y_M is the loss for each counterparty times the portfolio weights and given by:

$$\Lambda(x)|y_M, N = \lambda_{MN}(x) = \sum_{i=1}^J l_{MN}^i x_j$$

We are using this notation because as we said, for simplification, we may generate M y random variables and for y then generate an additional N z random variables. So the total number of scenarios would be $M \times N$.

Now the distribution of losses then would give us an approximate true distribution (had we used convolution). And as soon as you have the distribution of portfolio losses we can plot it and get VaR and CVaR as we did previously from a distribution of portfolio losses.

$$\Lambda(x) = [\lambda_{MN}(x)]$$

As soon as you have all the losses, your work is done.

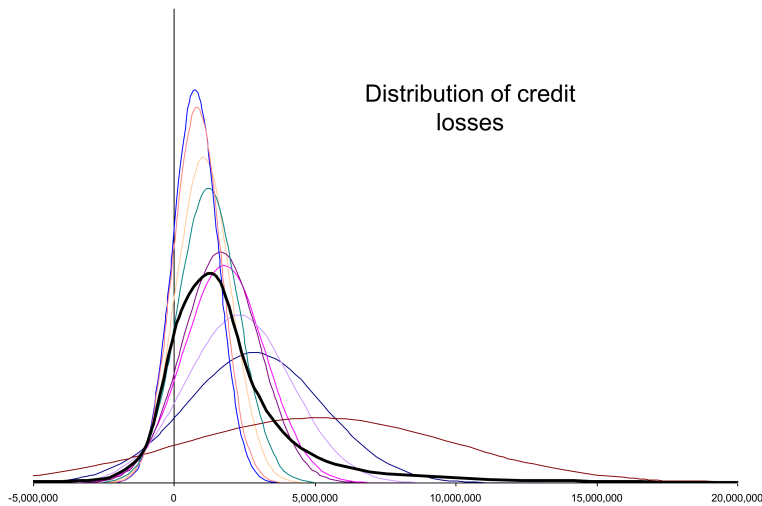
1.8.8 Distribution of Unconditional Framework

The following shows 10 conditional normal distributions. When we put these together, we get the unconditional portfolio loss distribution. notice that the portfolio distribution is not normal ¹⁰. So this is an illustration that even when we use random variables that are normally distributed, we get portfolios that are not normally distributed. So this is one of the caveats of the conditional dependence framework which we have used.

So conditional independence framework is not a normal distribution. Conditional on y , each one of the distributions is a normal distribution . But unconditionally, the portfolio distribution is a fat line. So this shows that with the help of normal distributed variables we can still model portfolios and counterparties that do not have normally distributed losses - even though each of the random variables is normally distributed.

¹⁰ we say that it has a fat tail

Conditional Normal approximation



1.8.9 Summary and use in Industry

Conditional Dependence Framework is the most complex model we talk about in this book and in fact is used in industry. The attributes in industry are as follows:

- **Typical portfolio size:** 5,000 counterparties
- **Typical number of credit drivers per party:** 1
- **Typical Beta:** 0.4 – 0.5
- **Typical number of systematic samples M:** 10,000
- **Typical number of specific samples N:** 1,000 (for risk measurement, *not* for optimization.)
- **Typical total number of scenarios** 10,000,000 ($M \times N$)

So in the example, we are essentially solving an industry problem with much smaller number of counterparties, and a 100 times less out of sample scenarios (100,000 versus 10,000,000). Note that the code provided ran around 30 seconds for 100,000 scenarios so you can use this as a benchmark ¹¹

¹¹ maybe different for your system but should not take hours!

2 *Matlab Functions*

2.1 *Vector Based Programming and simulation time*

All codes used in the in-sample and out of sample analysis are vector based for speed. The code for 100,000 samples run for less than 30 seconds. The 100 trials take a few minutes to run but can be optimized more to reduce the time.

Overall, the 100,000 out of sample scenarios and MC₁ and MC₂ scenarios with 100 trials take 4 minutes to run.

3 Results and analysis

3.1 Period Output

Portfolio 1:

Out-of-sample: VaR 99.0% = \$15574388.00, CVaR 99.0% = \$23498580.80

In-sample MC1: VaR 99.0% = \$13899085.00, CVaR 99.0% = \$20527116.78

In-sample MC2: VaR 99.0% = \$11928212.00, CVaR 99.0% = \$17855827.94

In-sample No: VaR 99.0% = \$8215656.80, CVaR 99.0% = \$9617205.94

In-sample N1: VaR 99.0% = \$7235465.67, CVaR 99.0% = \$8503896.23

In-sample N2: VaR 99.0% = \$6921917.20, CVaR 99.0% = \$8142676.01

Out-of-sample: VaR 99.9% = \$24590588.00, CVaR 99.9% = \$42627487.00

In-sample MC1: VaR 99.9% = \$27782539.00, CVaR 99.9% = \$34046393.40

In-sample MC2: VaR 99.9% = \$27701140.00, CVaR 99.9% = \$30968822.00

In-sample No: VaR 99.9% = \$11375078.14, CVaR 99.9% = \$12520160.18

In-sample N1: VaR 99.9% = \$10094806.44, CVaR 99.9% = \$11131129.06

In-sample N2: VaR 99.9% = \$9673794.64, CVaR 99.9% = \$10671168.88

Portfolio 2:

Out-of-sample: VaR 99.0% = \$14452443.67, CVaR 99.0% = \$23064292.72

In-sample MC1: VaR 99.0% = \$12888880.03, CVaR 99.0% = \$20181297.05

In-sample MC2: VaR 99.0% = \$10546353.42, CVaR 99.0% = \$16921141.93

In-sample No: VaR 99.0% = \$7605223.57, CVaR 99.0% = \$8927654.97

In-sample N1: VaR 99.0% = \$6757345.86, CVaR 99.0% = \$7963000.61

In-sample N2: VaR 99.0% = \$6285256.40, CVaR 99.0% = \$7424018.26

Out-of-sample: VaR 99.9% = \$25833032.68, CVaR 99.9% = \$38565450.43

In-sample MC1: VaR 99.9% = \$28944736.31, CVaR 99.9% = \$34157176.74

In-sample MC2: VaR 99.9% = \$26721076.74, CVaR 99.9% = \$31485366.18

In-sample No: VaR 99.9% = \$10586294.92, CVaR 99.9% = \$11666736.85

In-sample N1: VaR 99.9% = \$9475175.18, CVaR 99.9% = \$10460209.22

In-sample N2: VaR 99.9% = \$8852293.42, CVaR 99.9% = \$9782675.18

3.2 Distributions Plots

In the following pages the distribution plots for the three scenarios are shown. Although the losses do follow the expected log-normal shape for the loss side, there also seems to be log-normal distribution for gains which is expected. We only care about losses in the following analysis however.

3.3 Analysis

3.3.1 Analyzing Sample Error and Model Error

These results show that sampling error is significant.

The in sample VaR and CVaR **are not as accurate** and sometime overstate or underestimate the the more accurate out of sample loss. **The difference can be as large as 30% which is quite unacceptable.** For example, out of sample VaR for 99.9% indicates \$14.2Million loss but the in sample VaR indicates a \$12.9 Million loss (MC₁) and \$10.5 Million loss (MC₂). Therefore both in-sampling techniques underestimate loss in this case.

On the other hand, for VaR 99.9%, the out of sample loss is \$25.8 Million but the in-sample VaR is \$28.9 Million (MC₁) and \$26.7 Million (MC₂). So in this case, the in-sample scenarios overestimate the loss. Similar conditions hold for CVaR.

Model error (using normal distribution) has much worse results than in-sample errors. Model errors are **more than 50% off in most cases and sometimes more.** So I would only use model error for quick "back of the envelope" simulations but they should not be relied upon for decision making.

3.3.2 Reporting in Sample VaR and CVaR

Based on the above analysis, reporting in sample VaR and CVaR can either underestimate or overestimate risk. So essentially, reporting in-sample VaR/CVaR can cause you to either to take too much risk (losses are too small) or be too conservative (losses are too large) and therefore suffer on returns.

3.3.3 Possible Strategies

Apart from using a higher speed equipment, one method I can think of to improve results is regression. So we take the results that we achieve using out of samples and try to regress as to how much error there is when doing in-sample scenarios and make an adjustment.

A second approach is to use a more hardware oriented programming language. Matlab has quite a bit of overhead. Looking at Python, C/C++ or even machine language (Assembly) can speed up programs by a few factors.

A third approach is to use FPGA programming to do this, this is quite simple using Verilog code.

Finally, I would only do this if there is a clear reason for speeding up the simulations (for example if you are looking to test/improve on new models). Overall, I believe if you are going to plan for a 1 year risky asset model, then you are better off spending a few days running your simulations. Cutting corners can have catastrophic results.

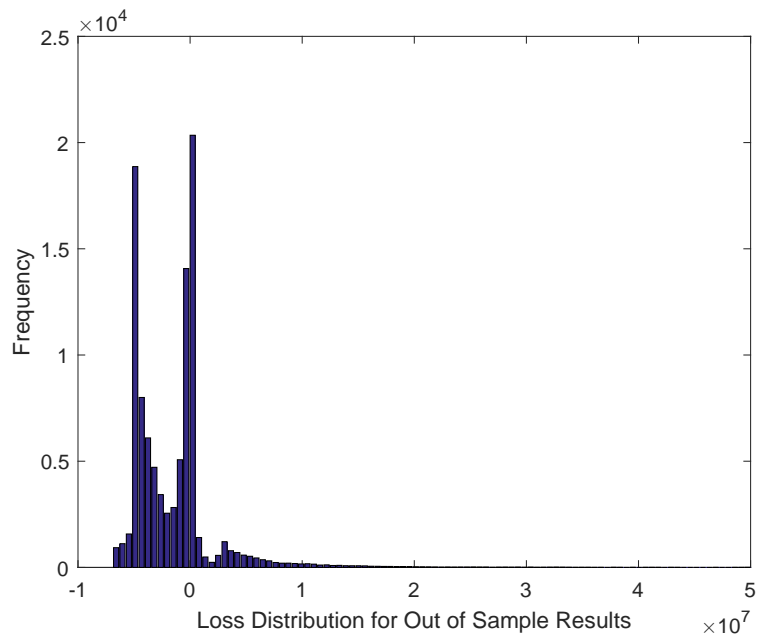


Figure 3.1: Out of sample loss distribution for portfolio 1

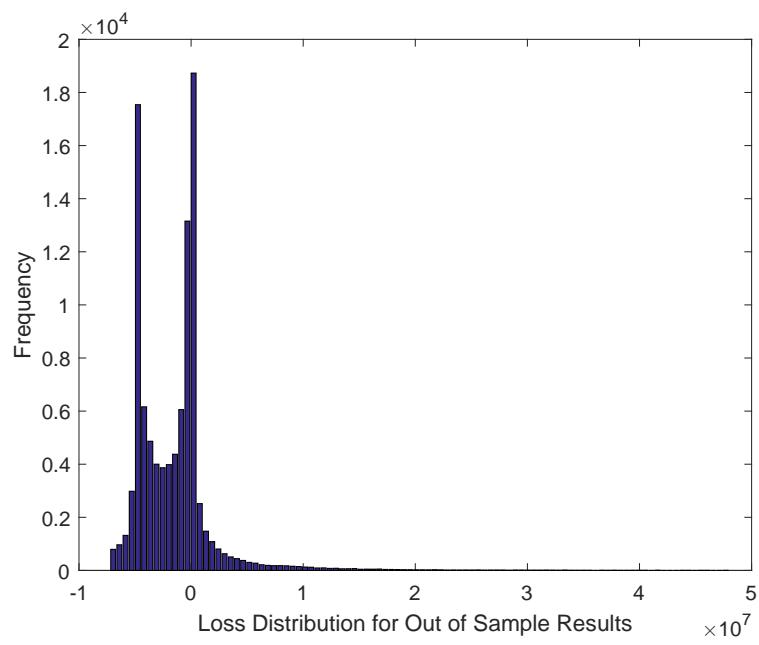


Figure 3.2: Out of sample loss distribution for portfolio 2

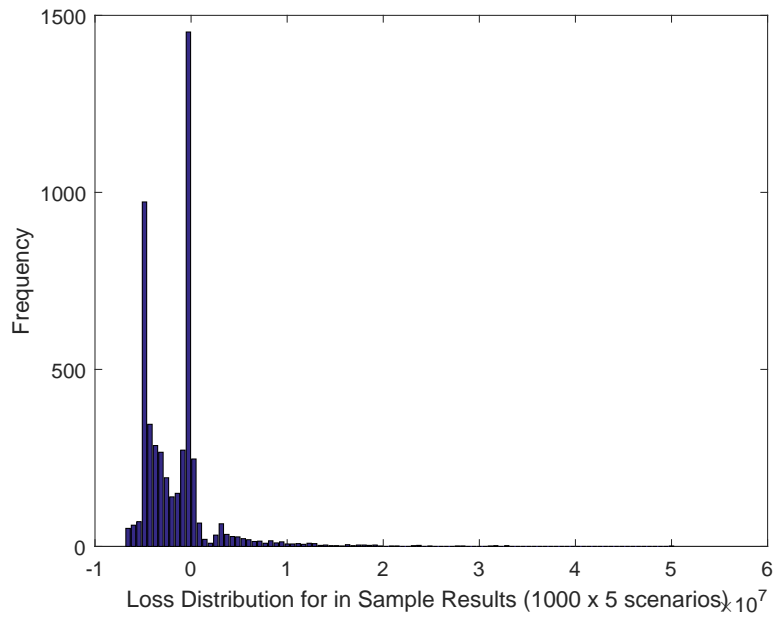


Figure 3.3: In sample loss distribution for portfolio 1

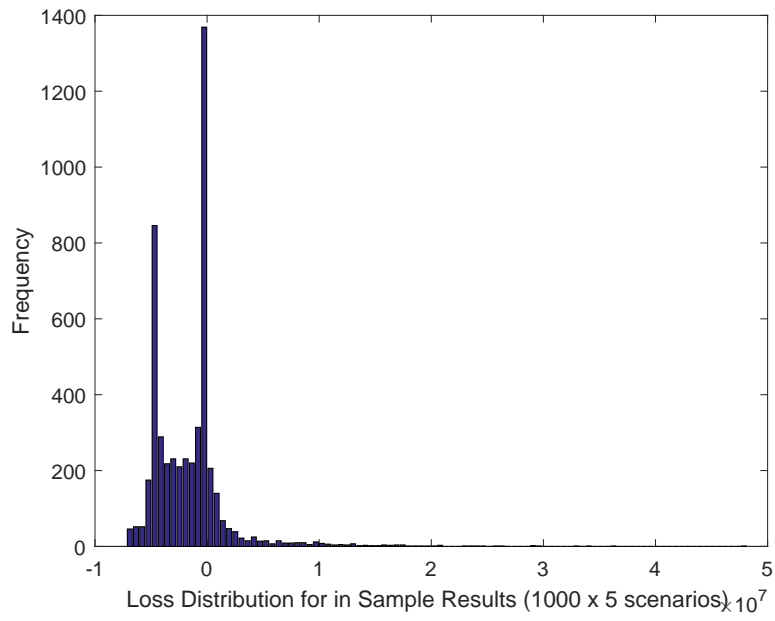


Figure 3.4: In sample loss distribution for portfolio 2

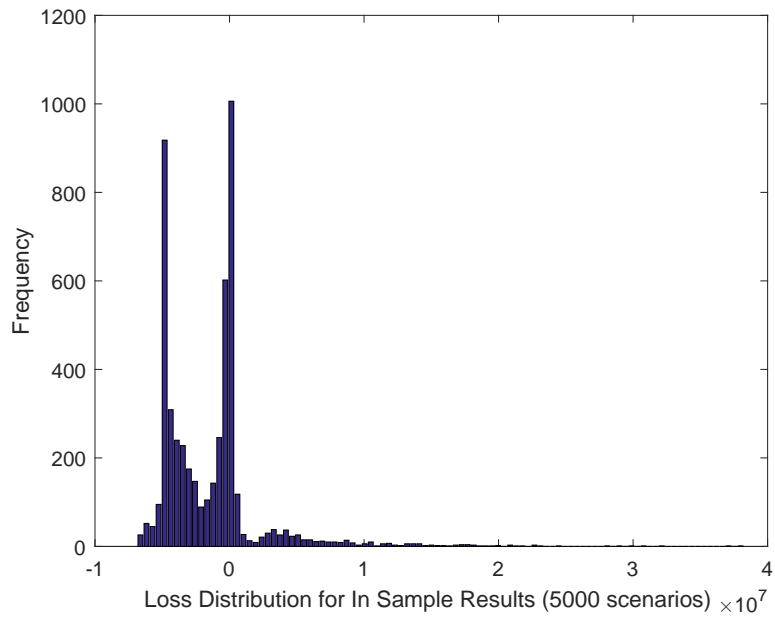


Figure 3.5: In sample loss distribution for portfolio 1

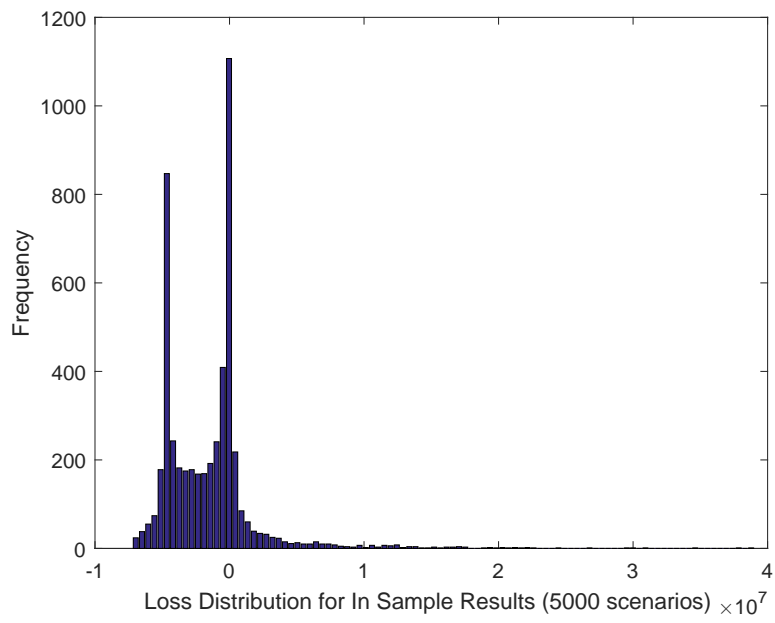


Figure 3.6: In sample loss distribution for portfolio 2