RM_HW5

Padmini

May 29, 2016

Question 1: The effect of loan concentration on Credit VaR of a portfolio of loans

Consider a portfolio of 1,000 loans of $10,000 each with LGD=45% and PD=0.19%, rho=0.15.

1. Apply the Normal Copula (Vasicek) closed-form formula to find VaR at 99.9%.

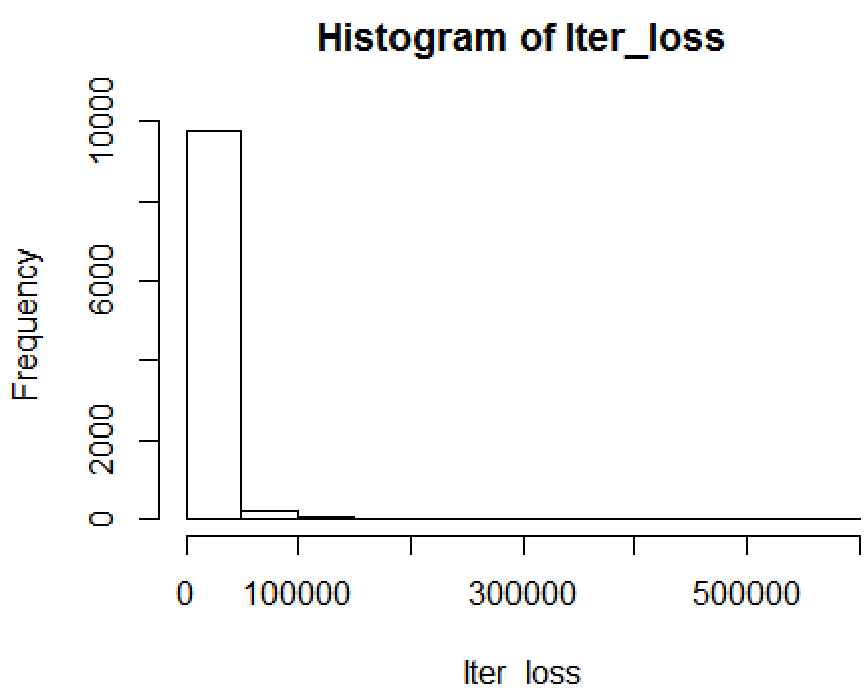
2. Simulate a Normal Copula Model, by running 10,000 iterations. For each one, draw F, the common factor, from a standard normal distribution. For each loan draw the idiosyncratic risk, and compute Ui. Determine whether the loan defaulted by comparing the Normal copula to the PD. For each iteration sum up all default losses. To find VaR use the relevant quantile of the total losses per iteration. Compare the simulation results to the closed-form formula.

3. Now, consider a portfolio of 100 loans of $100,000 each, with the same PD and LGD. Simulate VaR at 99.9%. Compare the result to the previous results.

4. How do your results change when you consider a portfolio of 10 loans of $1M each with the same PD and LGD?

```
loansNum = 1000 #Number of Loans
loanAmt = 10000 #Dollar size of each Loan
lgd = 0.45
pd = 0.0019 # pd of Loan
rho = 0.15 #Correlation between latent variables

################################
#######   Question 1.1   #######
################################
# Normal Copula (Vasicek) closed-form formula
varp =0.999
# Worst Case Default Rate
num = qnorm(pd) + sqrt(rho)*qnorm(varp)  #N-inverse is calculated using qnorm
denom = sqrt(1-rho)
wcdr = pnorm(num/denom) #N: Normdist cdf is calculated using pnorm
wcdr

## [1] 0.03279933

# Credit Var
cvar1.1 = loansNum*loanAmt*lgd*(wcdr - pd)
cvar1.1
```

```
## [1] 139047

################################
#######   Question 1.2   #######
################################
# Simulate a Normal Copula Model

iter = 10000 #number of iterations
Iter_loss = array (0, dim=c(iter)) #Distribution of losses per iteration
for (i in 1:iter) {
    F = matrix(rnorm(1),1,loansNum) #One F Factor value per iteration
        Z = matrix(rnorm(loansNum,mean=0,sd=1), 1, loansNum) #idiosyncratic
errors for all loans
        U = sqrt(rho)*F + sqrt(1-rho)*Z #Generate the U_i for all loans
        Default = (U < qnorm(pd)) #Every loan, every iteration, did it
default
        loan_loss = Default*loanAmt*lgd #Total loss on each loan for this
iteration
        Iter_loss[i] = sum(loan_loss) #Total loss for this iteration
}

options("scipen"=100, "digits"=4) #r format instructions

hist(Iter_loss) #Histogram of losses over iterations
```

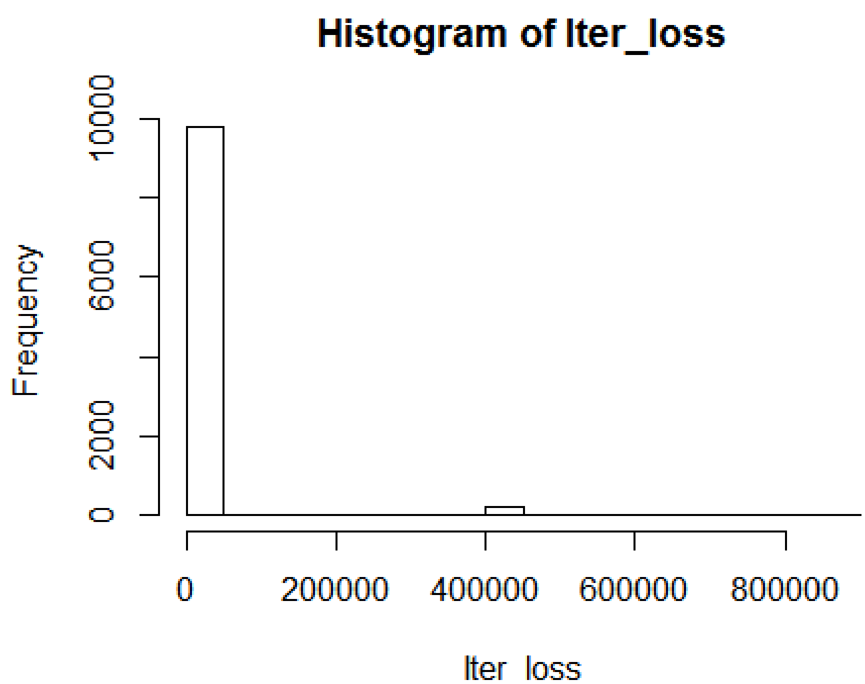



```
EL = mean(Iter_loss) #Expected loss
EL
```

```
## [1] 8607

cvar1.2 = quantile(Iter_loss, varp) #VaR
cvar1.2

##  99.9%
## 157514

# Comparing simulation results to the closed-form formula
cvar1.1

## [1] 139047

cvar1.2

##  99.9%
## 157514

# Credit Var from simulation is higher than that from closed form solution


################################
#######   Question 1.3   #######
################################
# Simulate a Normal Copula Model - new loan values
loansNum = 100 #Number of loans
loanAmt = 100000 #Dollar size of each loan

iter = 10000 #number of iterations
Iter_loss = array (0, dim=c(iter)) #Distribution of losses per iteration
for (i in 1:iter) {
    F = matrix(rnorm(1),1,loansNum) #One F Factor value per iteration
        Z = matrix(rnorm(loansNum,mean=0,sd=1), 1, loansNum) #idiosyncratic
errors for all loans
        U = sqrt(rho)*F + sqrt(1-rho)*Z #Generate the U_i for all loans
        Default = (U < qnorm(pd)) #Every loan, every iteration, did it
default
        loan_loss = Default*loanAmt*lgd #Total loss on each loan for this
iteration
        Iter_loss[i] = sum(loan_loss) #Total loss for this iteration
}

options("scipen"=100, "digits"=4) #r format instructions

hist(Iter_loss) #Histogram of losses over iterations
```


```
EL = mean(Iter_loss) #Expected loss
EL

## [1] 8334

cvar1.3 = quantile(Iter_loss, varp) #VaR
cvar1.3

##  99.9%
## 225000

# Comparing simulation results to the closed-form formula
cvar1.2

##  99.9%
## 157514

cvar1.3

##  99.9%
## 225000

# The var in (c) is higher than var in (b)


################################
#######   Question 1.4   #######
################################
```

```r
# Simulate a Normal Copula Model - new loan values
loansNum = 10 #Number of loans
loanAmt = 1000000 #Dollar size of each loan

iter = 10000 #number of iterations
Iter_loss = array (0, dim=c(iter)) #Distribution of losses per iteration
for (i in 1:iter) {
    F = matrix(rnorm(1),1,loansNum) #One F Factor value per iteration
        Z = matrix(rnorm(loansNum,mean=0,sd=1), 1, loansNum) #idiosyncratic
errors for all loans
        U = sqrt(rho)*F + sqrt(1-rho)*Z #Generate the U_i for all loans
        Default = (U < qnorm(pd)) #Every loan, every iteration, did it
default
        loan_loss = Default*loanAmt*lgd #Total loss on each loan for this
iteration
        Iter_loss[i] = sum(loan_loss) #Total loss for this iteration
}

options("scipen"=100, "digits"=4) #r format instructions

hist(Iter_loss) #Histogram of losses over iterations
```




```r
EL = mean(Iter_loss) #Expected loss
EL

## [1] 9360
```

```
cvar1.4 = quantile(Iter_loss, varp) #VaR
cvar1.4

##  99.9%
## 450000

# Comparing simulation results to the closed-form formula
cvar1.2

##  99.9%
## 157514

cvar1.3

##  99.9%
## 225000

cvar1.4

##  99.9%
## 450000

# The var in (d) is a lot higher than var in (c). We observe that as the
value of individual loans goes up, the credit var increases
```

Question 2: The effect of marking to market on Credit VaR of a portfolio of loans

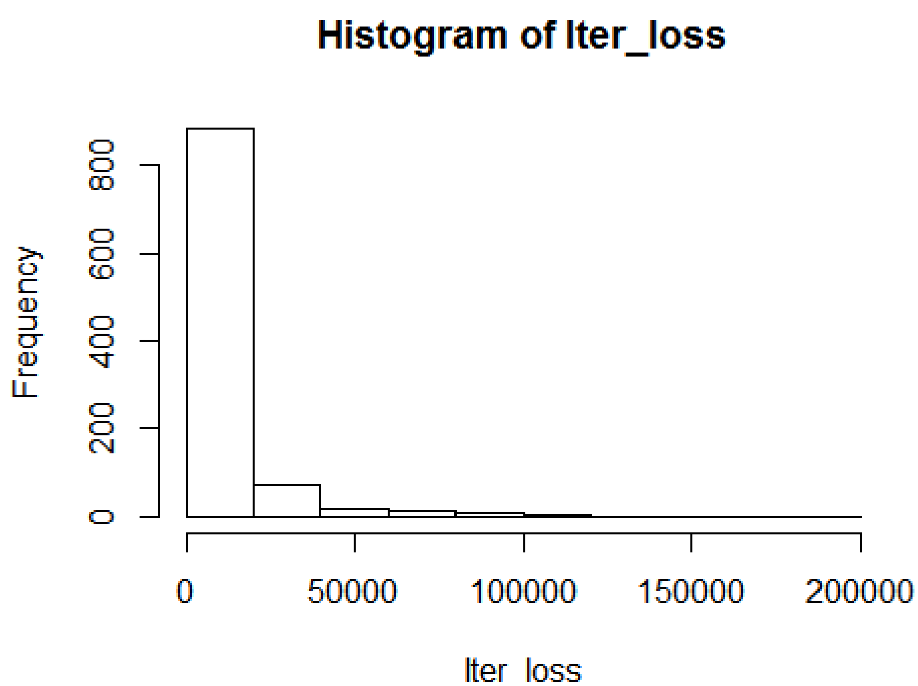
1. Simulate the Normal Copula Default-Only model on a portfolio of 1,000 loans of \$10,000 with LGD=45% to find Credit VaR at 99.9%. The loans are rated BBB and have PD=0.19%, rho=0.15.

2. Assume the loans are all 2-year loans, so at the end of the first year they will be one year from maturity when the principal will be paid along with a LIBOR+210bp coupon. Assume LIBOR is 0.5% for all time periods. The loans are rated BBB. The corresponding 1-year ratings transition matrix, and credit spreads for the different ratings are given.

Instead of figuring out for each loan whether it defaulted or not, now compute the loss/gain in value as result of rating migration. Compare this MTM VaR to the Default-Only VaR Result. How big is the effect? Do you think this effect would have been larger or smaller if the loans were rated AA? CCC? Why?

```
################################
#######   Question 2.1   #######
################################

# Normal Copula Default-Only model
loansNum = 1000 #Number of loans
```

```
loanAmt = 10000 #Dollar size of each loan
lgd = 0.45
pd = 0.0019 # pd of loan
rho = 0.15 #Correlation between latent variables
varp =0.999

iter = 1000 #number of iterations

Iter_loss = array (0, dim=c(iter)) #Distribution of losses per iteration
for (i in 1:iter) {
    F = matrix(rnorm(1),1,loansNum) #One F Factor value per iteration
        Z = matrix(rnorm(loansNum,mean=0,sd=1), 1, loansNum) #idiosyncratic errors for all loans
        U = sqrt(rho)*F + sqrt(1-rho)*Z #Generate the U_i for all loans
        Default = (U < qnorm(pd)) #Every loan, every iteration, did it default
        loan_loss = Default*loanAmt*lgd #Total loss on each loan for this iteration
        Iter_loss[i] = sum(loan_loss) #Total loss for this iteration
}

options("scipen"=100, "digits"=4) #r format instructions

hist(Iter_loss) #Histogram of losses over iterations
```



```
EL1 = mean(Iter_loss) #Expected loss
EL1

## [1] 8896

cvar2.1 = quantile(Iter_loss, varp) #VaR
cvar2.1

##  99.9%
## 135054





################################
#######   Question 2.2   #######
################################
T = 2 #loan years
libor = 0.005
coupon = (libor + 0.0210)*loanAmt
loanValue = data.frame(matrix(nrow = loansNum, ncol = iter))

# Std Normal Cutoffs
ratings <- data.frame(matrix(nrow = 8, ncol=2))
ratings[,1] <- c(0.05,0.19,4.79,89.41,4.35,0.82,0.2,0.19)/100
ratings[,2] <- c(0.7,0.88,1.19,2.1,3.39,4.56,8.17,0)
colnames(ratings) <- c('prob','spread')
ratings$cumprob <- 1
ratings$lb[1] <- qnorm(ratings$cumprob[1]- ratings[1,1])
ratings$ub[1] <- qnorm(ratings$cumprob[1])
for (i in 2:8){
  ratings[i,3] <- ratings[i-1,3] - ratings[i-1,1]
  ratings[i,4] <- qnorm(ratings[i,3]- ratings[i,1])
  ratings[i,5] <- ratings[i-1,4]
}

## Warning in qnorm(ratings[i, 3] - ratings[i, 1]): NaNs produced

ratings$lb[8] <- qnorm(0)

# Creating fwd curve
ratings$fwdrate <- libor + ratings$spread/100
ratings$fwdrate[8] <- Inf
ratings

##     prob spread cumprob     lb     ub fwdrate
## 1 0.0005   0.70  1.0000  3.291    Inf  0.0120
## 2 0.0019   0.88  0.9995  2.820  3.291  0.0138
## 3 0.0479   1.19  0.9976  1.642  2.820  0.0169
## 4 0.8941   2.10  0.9497 -1.593  1.642  0.0260
## 5 0.0435   3.39  0.0556 -2.254 -1.593  0.0389
```

```
## 6 0.0082   4.56  0.0121 -2.661 -2.254  0.0506
## 7 0.0020   8.17  0.0039 -2.894 -2.661  0.0867
## 8 0.0019   0.00  0.0019   -Inf -2.894     Inf
```

```
for (k in 1:iter) {
  ### Simulating Ui for every loan
  F = matrix(rnorm(1),loansNum,1) #One F Factor value per iteration
    Z = matrix(rnorm(loansNum,mean=0,sd=1), loansNum, 1) #idiosyncratic
errors for all loans
    U = data.frame(sqrt(rho)*F + sqrt(1-rho)*Z) #Generate the U_i for all
loans

  ### Determing loan's new ratings based on std norm cdf cutoffs
  U$oldrating <- 4
  U$newrating <- 4

  for (i in 1:loansNum) {
    for(j in 1:8){
      if( U[i,1] >= ratings[j,4] && U[i,1] < ratings[j,5]){
        U[i,3] = j
        break
      }
    }
  }

  ### Repricing the loan with the credit spreads for the new rating
  loanValue[c(1:loansNum),k] <- coupon +
(loanAmt+coupon)/(1+ratings$fwdrate[U[c(1:loansNum),3]])
}


### MTM Var Calculation

# Sum of all the iterations
new <- colSums(loanValue)

# MEan Horizon Value - average of loan values of all iterations
MHV = sum(new)/iter
MHV
```

```
## [1] 10238569
```

```
# X% Worst Case
new <- sort(new, decreasing = FALSE)
XWC = new[(1-varp)*iter]
XWC
```

```
##    X924
## 9932730
```

```r
# Expected loss
EL2 = loanAmt*loansNum - MHV
EL2

## [1] -238569

# unexpected VAR
cvar2.2 <- MHV - XWC
cvar2.2

##   X924
## 305839

### Comparison with 2.1 answer
cvar2.1

##  99.9%
## 135054

cvar2.2

##   X924
## 305839

# MTM VaR is higher compared to the Default-Only VaR Result. The default only
underestimates var
```

Effect if the loans were rated AA? CCC?

The major differences will be the coupons used to find the value of the loan and Probability (%) of Transition.

Spread for BBB loans was 2.10%. Probability (%) of Transition was 89.41% which is high.

(1) For AA, spread is only 0.88% so the loan values will be way lesser. At the same time, probability of transition is only 0.19%. Hence the difference in var value by default only and MTM methods will not be very different.

(2) For CCC, spread is 8.17% so the loan values will be a lot higher than 2.2. At the same time, probability of transition is only 0.2%. Hence the difference in var value by default only and MTM methods will not be very different.