Variance Reduction for Estimating Value at Risk

STAT 428 FA2018 - Group #18

Haowen Zhou (haowenz4)
Qilan Zhang (qilanz2)
Jiahui Zhao (jzhao71)
Shanxin Zhou (shanxin2)
Shuaiqi Zheng (szheng25)

Abstract

|This project is focusing on applying methods learned from class in real life financial case. (need adding more sentences)"

Contents

1 Introduction

It is important for banks and insurance companies to estimate and manage the risk to avoid the situation when there is capital deficiency. Value at Risk (VAR) is a measure for the risk of investments loss, which estimates how much value a investment portfolio might lose at given market conditions. Two types of problems are often encompassed by acompany when measuring the risk.The first one is how the value of market risk will change according to risk factors (e.g. interest rate). The other one is what in fluence the risk will pose on the value of a portfolio. This project is intended to develope a model to revalue the portfolio with the interest rate by starting from the simple linear assumption to quadratic model and make use of revalued portfolio to estimate Value at Risk(VAR). The technique problem we are facing is how to make our estimate much more precise,which means we should do variance reduction for our estimator of VAR.

About the data

2 Methods

2.0.0.0.1 part 1

$$\Delta V \approx \frac{\partial V}{\partial t}\Delta t + \delta^\top \Delta S + \frac{1}{2}\Delta S^\top \Gamma \Delta S$$

, (9.2)

where

$$\delta_i = \frac{\partial V}{\partial S_i}, \Gamma_{ij} = \frac{\partial^2 V}{\partial S_i \partial S_j}$$

$\Delta S = CZ$ with $CC^\top = \sum_S$ $L = -\Delta V$

2.0.0.0.2 Part 2

$\Delta S \sim N(0, \sum_S)$ $a = -\frac{\partial V}{\partial t}\Delta t$

$$L \approx a - (C^\top \delta)^\top - \frac{1}{2}Z^\top (C^\top \Gamma C) Z$$

(9.3)

$$-\frac{1}{2}\tilde{C^\top}\Gamma\tilde{C^\top} = U\Lambda U^\top$$

set $C = \tilde{C}U$, $CC^\top = \tilde{C}UU^\top\tilde{C^\top} = \sum_S$

$$-\frac{1}{2}C^\top \Gamma C = -\frac{1}{2}U^\top(\tilde{C}\Gamma\tilde{C})U = U^{top}(U\Lambda U^\top)U = \Lambda$$

$b = -C^\top \delta$

$$L \approx a + b^\top Z + Z^\top \Lambda Z$$
$$= a + \sum_{j=1}^{m}(b_j Z_j + \lambda_j Z_j^2 \equiv Q)$$

(9.4)

2.0.0.0.3 Part 3

$\frac{\partial P_\theta}{\partial P} = e^{\theta Q - \phi(\theta)}$

$$\mathcal{P}(Q > x) = E_\theta[(\frac{\partial P}{\partial P_\theta})\infty(Q > x)] = E_\theta[e^{-\theta Q + \phi(\theta)}\infty(Q > x)]$$

$Z \sim P_\theta$, $Z \sim N(\mu(\theta), \sum(\theta))$, $\mu_j(\theta) = \frac{\theta b_j}{1-2\lambda_j\theta}$, $\sigma_j^2(\theta) = \frac{1}{1-2\lambda_j\theta}$ $2\lambda\theta < 1$, so that $\phi(\theta) < \infty$ $\frac{\partial P_\theta}{\partial \mathcal{P}} = e^{\theta Q - \phi(\theta)}$ $\mathcal{P}(Q > x) = \mathcal{E}[(\frac{\partial P}{\partial P_\theta})\infty(\theta > x)] = E_\theta[e^{-\theta Q}\mathcal{Q} > \S]$ $Z \sim \mathcal{P}_\theta$, $Z \sim N(\mu(\theta), \sum(\theta))$, $\mu_j(\theta) = \frac{\theta b_j}{1-2\lambda_j\theta}$, $\sigma_j^2(\theta) = \frac{1}{1-2\lambda_j\theta}$ $2\lambda_j\theta < 1$, so that $2P(\theta) < \infty$ $\phi(\theta) \equiv a\theta + \sum_{j=1}^m \phi_j(\theta) = a\theta + \frac{1}{2}\sum\sum_{j=1}^m(\frac{\theta^2 b_j^2}{1-2\theta\lambda_j} - log(1-2\theta\lambda_j))$

$$\mathcal{P}(L > x) = \mathcal{E}[e^{-\theta Q + \phi(\theta)}\infty(L > x)]$$

```
# Multinormal Distribution
Multi_Normal <- function(u, Sigma){
  N <- 5000
  burn <- 3000
  X <- matrix(0, N, 2)

  rho <- Sigma[1,2]
  mu1 <- u[1,1]
  mu2 <- u[2,1]
  sigma1 <- Sigma[1,1]
  sigma2 <- Sigma[2,2]
  s1 <- sqrt(1 - rho^2) * sigma1
  s2 <- sqrt(1- rho^2) * sigma2

  X[1,] <- c(mu1, mu2)

  for(i in 2:N){
    x2 <- X[i-1,2]
    m1 <- mu1 + rho * (x2 -mu2) * sigma1/sigma2
    X[i,1] <- rnorm(1, m1, s1)
    x1 <- X[i,1]
    m2 <- mu2 + rho * (x1 -mu1) * sigma2/sigma1
    X[i,2] <- rnorm(1, m2, s2)
  }

  b <- burn+1
  x <- X[b:N,]
  return(x)
}
```

```
# General way percentile is p
VAR <- function(a0, a, A, Sigma,delta_S, percentile){

  L <- a0 + delta_S %*% a + 0.5* rowSums(delta_S %*% A * delta_S)
  L <- -L
  L <- sort(L)

  p_index <- ceiling(length(L) * percentile)
  x_quantile <- L[length(L) - p_index]

  return( x_quantile)
}
```

```r
General <- function(a0, a, A, Sigma, x){
  B <- 1000
  Prob <- numeric(B)

  for(i in 1:B){
    delta_S <- Delta_S[sample(1:nrow(Delta_S),500),]
    #delta_S <-  Multi_Normal(a, Sigma)
    L <- -(a0 + delta_S %*% a + 0.5 *rowSums(delta_S %*% A * delta_S))
    Prob[i] <- mean(L > x)
  }
  return(c(mean(Prob) , var(Prob)))
}
```

```r
# Importance Sampling
phi <- function(theta,a, b, lambda){
  phi_value <- a * theta + 0.5 *
    sum((theta^2 * b^2)/(1 - 2 * theta * lambda) - log(1 - 2 * theta * lambda))
  return(phi_value)
}

IS <- function(a0, a, A, Sigma, x){
  N <- 1000
  loss <- numeric(N)
  for(i in 1:N){
  # get lambda, b
  CC <- eigen(Sigma)
  CC <- CC$vectors %*% diag(sqrt(CC$values), nrow=2, ncol=2)
  matrix_trans <- t(CC) %*% A %*% CC
  C <- eigen( -0.5 * matrix_trans)

  lambda <- C$values

  C <- CC %*% C$vectors

  b <- -t(C) %*% a

  #IS distribution
  theta <- 0.25
  #theta <- uniroot(phi, a=a, b=b, lambda= lambda, lower = -100, upper = 100)

  mu_new <- theta * b /(1- 2 * lambda *theta)
  sigma_new <- 1/(1 - 2 * theta * lambda )

  # get Z, S
  u <- matrix(mu_new,nrow = 2, ncol = 1)
  sig <- diag(sigma_new, nrow = 2, ncol = 2)

  Z <- mvrnorm(3000, u, sig)
  delta_S <- Z %*% t(C)

  # estimate Q, L
  Q <- -a0 + Z %*% b + rowSums(lambda * Z^2)
```

```r
  L <- a0 + delta_S %*% a + 0.5 *rowSums(delta_S %*% A * delta_S)
  L <- -L

  phi <- -a0 * theta + 0.5 * sum( (theta^2 * b^2)/(1 - 2* theta * lambda) - log(1- 2*theta * lambda) )
  loss[i] <-  mean( exp(theta * Q + phi) *( L > x) )
  }

  prob <- mean(loss)
  var <- var(loss)

  return(c(prob, var))
}
```

```r
# Stratified Sampling
SS <- function(a0, a, A, Sigma, x){
  N <- 1000
  loss <- numeric(N)

  # get lambda, b
  CC <- eigen(Sigma)
  CC <- CC$vectors %*% diag(sqrt(CC$values), nrow=2, ncol=2)
  matrix_trans <- t(CC) %*% A %*% CC
  C <- eigen( -0.5 * matrix_trans)

  lambda <- C$values

  C <- CC %*% C$vectors

  b <- -t(C) %*% a

  #IS distribution
  theta <- 0.01
  mu_new <- theta * b /(1- 2 * lambda *theta)
  sigma_new <- 1/(1 - 2 * theta * lambda )

  u <- matrix(mu_new,nrow = 2, ncol = 1)
  sig <- diag(sigma_new, nrow = 2, ncol = 2)

  for(i in 1:N){
    # get Z, S
    #Z <- mvrnorm(3000, u, sig)
    Z <- Multi_Normal(u, sig)
    delta_S <- Z %*% t(C)

    # estimate Q, L
    Q <- -a0 + Z %*% b + rowSums(lambda * Z^2)
    Q_sort <- sort(Q)

    # strata 1
    point1 <- quantile(Q, 0.2)
    Z_1 <- Z[which(Q <= point1),]
    delta_S_1 <- Z_1 %*% t(C)
```

```
    # strata 2
    point2 <- quantile(Q, 0.5)
    Z_2 <- Z[which(Q >point1 & Q <= point2),]
    delta_S_2 <- Z_2 %*% t(C)

    # strata 3
    point3 <- quantile(Q, 0.8)
    Z_3 <- Z[which(Q > point2 & Q <= point3),]
    delta_S_3 <- Z_3 %*% t(C)

    # strata 4
    Z_4 <- Z[which(Q >point3),]
    delta_S_4 <- Z_4 %*% t(C)

    phi <- -a0 * theta + 0.5 * sum( (theta^2 * b^2)/(1 - 2* theta * lambda) - log(1- 2*theta * lambda) )

    Q1 <- -a0 + Z_1 %*% b + rowSums(lambda * Z_1^2)
    L1 <- -(a0 + delta_S_1 %*% a + 0.5 *rowSums(delta_S_1 %*% A * delta_S_1))
    loss1 <- mean(exp(theta * Q1 + phi) * (L1 >x))

    Q2 <--a0 + Z_2 %*% b + rowSums(lambda * Z_2^2)
    L2 <- -(a0 + delta_S_2 %*% a + 0.5 *rowSums(delta_S_2 %*% A * delta_S_2))
    loss2 <- mean(exp(theta * Q2 + phi) * (L2 >x))

    Q3 <--a0 + Z_3 %*% b + rowSums(lambda * Z_3^2)
    L3 <- -(a0 + delta_S_3 %*% a + 0.5 *rowSums(delta_S_3 %*% A * delta_S_3))
    loss3 <- mean(exp(theta * Q3 + phi) * (L3 >x))

    Q4 <--a0 + Z_4 %*% b + rowSums(lambda * Z_4^2)
    L4 <- -(a0 + delta_S_4 %*% a + 0.5 *rowSums(delta_S_4 %*% A * delta_S_4))
    loss4 <- mean(exp(theta * Q4 + phi) * (L4 >x))

    loss[i] <- (loss1+loss2+loss3+loss4)/4
  }

  prob <- mean(loss)
  var <- var(loss)

  return(c(prob,var))

}
```

```
Controlvariate <- function (L, Q, x) {
  # input: (Li, Qi), x
  # output: estimation of P(L>x), variance of L after/before implement control variate
  # Using bootstrap method to estimate P(Q > x)

  ProbQ <- mean(rep((sample(Q, size = length(Q), replace = TRUE) > x), 100000))
  beta <- cov(L, Q)/var(Q)
  ProbL <- mean(L > x) - beta * (mean(Q > x) - ProbQ)
  varL <- var(L) - 2*beta*sd(L)*sd(Q)*cor(L, Q) + beta^2*var(Q)
  result <- as.matrix(c(ProbL, varL, var(L)))
  rownames(result) <- c("Est of P(L>x) (quantile of x)", "variance after CV", "variance before CV")
```

```r
  return(c(ProbL, varL))
}

CV <- function(a0, a, A, Sigma, x){

  # get lambda, b
  CC <- svd(Sigma)
  CC <- CC$u %*% diag(sqrt(CC$d), nrow=2, ncol=2)
  matrix_trans <- -0.5 * t(CC) %*% A %*% CC
  C <- svd(matrix_trans)
  lambda <- C$d
  C <- CC %*% C$u

  b <- -t(C) %*% a

  u <- matrix(0,nrow = 2, ncol = 1)
  sig <- diag(1, nrow = 2, ncol = 2)
  Z <- mvrnorm(3000, u, sig)
  delts_S <- Z %*% t(C)

  Q <- -a0 + Z %*% b + rowSums(lambda * Z^2)

  L <- a0 + delts_S %*% a + 0.5 *rowSums(delts_S %*% A * delts_S)
  L <- -L
  return(Controlvariate(L, Q, x_var))
}
```

```r
# Data Precess
Interest <- read.csv("Interest.csv", header = TRUE)
Bond <- read.csv("Bond.csv", header = TRUE)

Bond <- Bond[c(625:688),2]
Interest <- Interest[c(745:808),2]

delta_Bond <- numeric(length(Bond))
delta_Interest <- numeric(length(Interest))

for(i in 2 : length(Bond) ){
  delta_Bond[i-1] <- Bond[i] - Bond[i-1]
  delta_Interest[i-1] <- Interest[i] - Interest[i-1]

}

# Input parameter
Sigma <- cov(data.frame(delta_Bond,delta_Interest))
Sigma <- as.matrix(Sigma)
a0 <- 0
a <- matrix(c(1,1), ncol = 1)
A <- matrix(c(1,0,0,1), ncol = 2, nrow = 2, byrow = TRUE)
percentile <- 0.9
Delta_S <- Multi_Normal(matrix(c(0,0),ncol=1), Sigma)
# Var
x_var <- VAR(a0, a, A, Sigma, Delta_S, percentile)
```

```r
# Estimate precision and variance
variance_general <- General(a0, a, A, Sigma, x_var)

variance_is <- IS(a0, a, A, Sigma, x_var)

variance_ss <- SS(a0, a, A, Sigma, x_var)

variance_cv <- CV(a0, a, A, Sigma, x_var)
```

3 Result

4 Discussion

5 Conclusion

6 Appendix

7 Bibliography

delete the below code when finish