Betalpha Technology Research 2017

Portfolio VaR and CVaR Optimization

via Multi-Factors Model

Niyu Jia

Introduction

We have been talking about mean-variance portfolio optimization methods under Markowitz theory. While in practice, people may focus more on the downside risk, that is, when the portfolio makes loss. Thus it's a good idea to replace the variance with value at risk (VaR) and conditional value of risk (CVaR). In this project, we will try to optimize portfolio by minimize portfolio VaR and CVaR minus a penalty constant times portfolio expected return, sum of allocation is 1 and max allocation is 30%, we won't accept short sell. We use multi-factors model to simulate out of sample stock return. We based on S&P 100 stocks and using the data from 2012 to 2016 to simulate and optimize, then using data in 2016 to do backtesting on our models' performance.

2 Model Construction

Based on the Fundamental-factor model, we denote stock return as:

$$r_i = \alpha + \sum_{i=1}^{n} \beta_i F_i + \varepsilon$$

WhereF_i is i-th factor exposure, β_i is the coefficient of Factor i.

2.1 Choice of Factors

Ideally we want to simulate with minimum error therefore one of the best features of these factors is that they can be used with every stock. For this purpose, we choose the factors bellow as something that exert influence over the whole market:

RM: Market return

YIELD: Interest rate

LIQ: Daily turnover rate over the whole market, can be computed as $\frac{TransactionVolume}{TotalVolume}$ (A factor reflects liquidity of market)

SZ: Natural daily logarithm of market capitalization (A factor reflects size of market)

VOL: Daily VIX index (A factor reflects market volatility)

This model means that we put all idiosyncratic return of specific stock return into the residual terms.

2.2 Data Source and Time Horizon

The whole market data is used for acquiring factor exposure and factor loading. But we only consider the portfolio of SPY100 as our target portfolio.

As for time horizon, we use factor data from 2012-2015 to compute the coefficients and use data of 2016 to do back test.

Our turnover data comes from CRDS, Volume data comes from Kenneth R. French website and yield data comes from government site.

2.3Model details

Multi Factor Regression

With factors described as above, we daily regress the stock return against these factor and acquire the β.

We are expecting that the residuals are iid distributed.

Simulate factors

Since we have already noticed that portfolio return VaR is not the simple sum of single stock return VaR, we have to firstly know the distribution of the portfolio return. One way is to simulate this distribution.

Monte Carlo Simulation

The common way to simulate portfolio return is to simulate the return of every stock and then do the summation according to the weights. However, this method involves Monte Carlo Error:

$$MCE_i = Var(r_i)$$

Which is the simulation variance of i-th stock. Since we assume that every Monte Carlo simulation is independent therefore the overall variance of portfolio return simulation is:

$$MCE_p = \sum_{i=1}^{n} w_i^2 Var(r_i)$$

Where the wi is the weight of i-th stock.

Although we do have some methods to reduce the variance such as Control Variates or Antithetic Variables .But let us assume that these factors are multivariate distributed with covariance matrix Σ. Then we can simulate these factors with Monte Carlo method to get stock return.

For a multivariate normal distribution with covariance matrix Σ, we do Cholesky decomposition:

$$\Sigma = AA^T$$

We want to simulate multivariate normal $XN(\mu, \Sigma)$, where $\mu = (\mu_1, \mu_2, \ldots \mu_n)$, n is the number of the factors； μ_n is the mean of factor n

Then draw a standard multivariate normal $YN(0, I)$, we can simulate X as:

$$X = AY + \mu$$

Bootstrapping

We use bootstrapping to obtain portfolio return distribution

The portfolio return is just simply the weighted sum of stock return:

$$r_p = w^T \vec{r}$$

Apparently, after the simulation we can theoretically write down the VaR of portfolio return at the 95% quantile:

Sort the simulated portfolio return rp from the lowest to the highest.

Get the exactly the lowest 5% of the return as the VaR.

$$\mathrm{VaR_p} = \mathrm{q\ r_p}$$

q = N×5%

2.4 Optimization Approach

We are using the following approaches to minimize portfolio VaR:

Direct Method: Genetic Algorithm

Indirect Method: CVaR method

3. Genetic Algorithm

3.1. Genetic Algorithm

Apparently, the portfolio VaR is not smooth or differentiable wrt weight vector. It is hard for us to implement gradient-based optimization method.

And Since it is difficult to write explicit form of probability density function as a function of unknown weight w, we can use Genetic Algorithm to evolve for the optimal weight.

The algorithm can be summarized as follow:

Initialization

Randomly initialize some arrays with population size of P. Regard these vectors as possible choice of solution weights.

Target function

Use the target function as:

$$target = VaR_p - \lambda r_p$$

r_p and VaR_p are same as above, λ is the penalty coefficient which can be regarded as risk aversion. The meaning of this target is that although we have a VaR, we think our expected return can compensate part of the risk to some degree.

Selection

Selection principal is to select the good v with higher probability to generate next generation. Some common selection methods are stochastic acceptance, roulette-wheel selection, tournament selection etc.

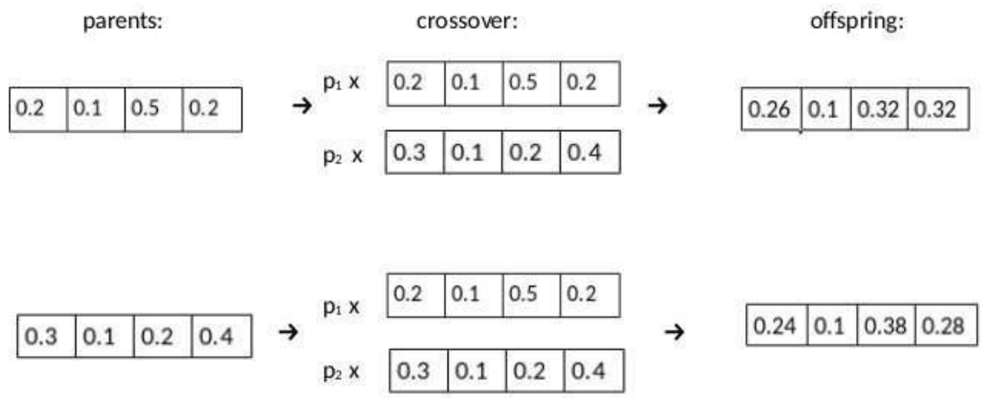
Recombination and Mutation

We define crossover rule as follow:

For the selected v in the previous step, randomly make pairs as parents and then recombine some parts of their genes with certain probability p1 in one of the parent, p2 in another parent, with random starting point and random length.

Here is an example of recombination:

p1=0.4, p2=0.6



As for the mutation, we assume that every gene has a small probability to mutate. The value of the gene may shift upwards or downwards by a number between (0,1). As the result of mutation may violate the constraint that sum of the weights is 1, we have to normalize the vector after mutation

Here is an example of mutation:

Original weight

0.25	0.07	0.34	0.11	0.23

↓

Mutation

0.25(+0.05)	0.07	0.34	0.11(-0.2)	0.23

↓

New weight

0.3	0.07	0.34	-0.09	0.23

↓

Normalized new weight

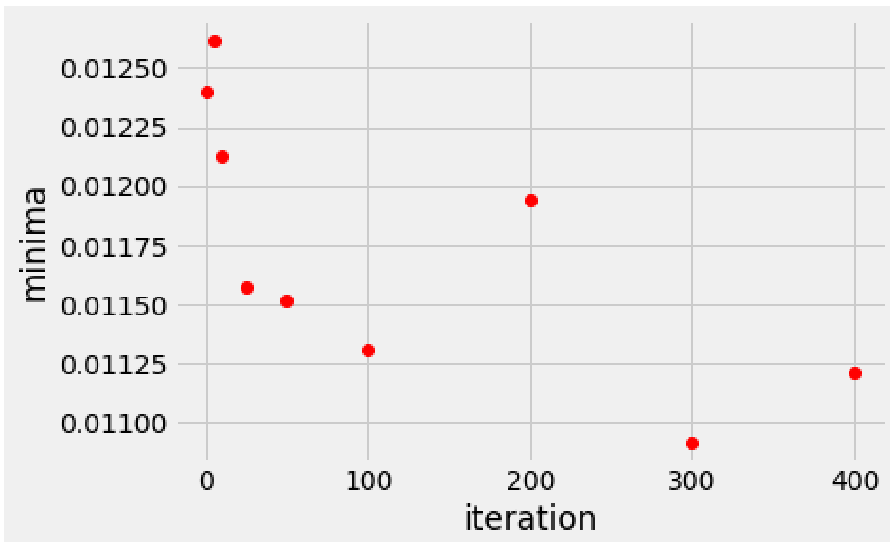
0.353	0.082	0.4	-0.106	0.271

For the new generation, check whether they satisfy our goal, if not, back to step (2)

3.2. Parameter choosing

We use 4 year history data from history to obtain a single optimized weight for some stocks in the portfolio. In the following instance we choose the first 15 stocks in the SP100 list to construct our portfolio. (If no information of portfolio is given, the program will randomly choose some stocks for the portfolio)

In order to avoid the local minima, we set a relatively high mutation probability to 0.05. As for the iteration, we tried different numbers and record their minima (of first day):

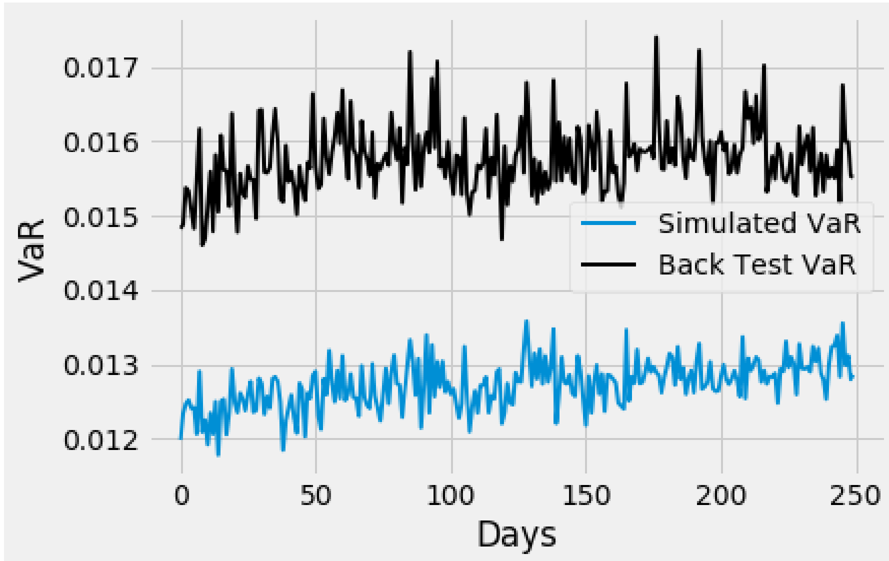


We see that the minimum value drops quickly when iteration grows larger but has fluctuation in finding the solution. Finally, considering the CPU workload, we choose 300 as our generation number.

Moreover, we find the size of initial population won't exert too much influence on the convergence of the optimization.

3.3 Back Test

For the optimized weight we get above, we calculate the rolling historical return for past 4 years and get the 95% quantile as the real historical VaR value. Then we compare it with the optimized VaR in terms of our model.



This plot shows that we cannot fully believe the result of genetic algorithm for the expected VaR. There are possible reasons:

(1) We are choosing inefficient factors and they may not be multivariate normally distributed

(2) Simulation error in factors and residuals

(3) Genetic Algorithm does not converge, we may miss the global minimum value

As a result, we are expecting a higher VaR when we refer to our genetic algorithm result.

4. CVaR Algorithm

4.1 Model construction

Firstly we put forward the portfolio return loss function $f(x, y)$, where x is the weight vector, *y* is the future returns of the stocks. *f (x, y) =<x , -y>*

Then denote $\varphi(x,\alpha) = \int_{f(x,y)<\alpha} p(y)dy$ as the probability that *f(x,y)* won't exceed the cutoff value α.

To figure out what is the value of the cutoff α, we can firstly consider α as a function of *x* and β, where beta is the probability level, we usually set it as 0.95. Accordingly, α denotes the value of β(usually 95%) cutoff of loss function.

According to the definition of CVaR, we can express it as $\emptyset(x, \beta)$, where

$$\emptyset(x, \beta) = \frac{1}{1-\beta} \int_{f(x,y)>\alpha} f(x, y)p(y)dy$$

The problem to be solved is minimizing CVaR to control on the downside risk.

To make the objective function able to optimize, we rearrange CVaR formula as:

$$\emptyset(x, \beta) = \frac{1}{1-\beta} \int_{f(x,y)>\alpha} f(x, y)p(y)\, dy$$

$$\frac{1}{1-\beta}\left[\int_{f(x,y)>\alpha} (f(x, y) - \alpha)p(y)dy + \int_{f(x,y)>\alpha} \alpha p(y)\, dy\right]$$

$$\frac{1}{1-\beta} \int_{f(x,y)>\alpha} (f(x, y) - \alpha)p(y)dy + \frac{\alpha}{1-\beta}\big(1 - \varphi(x, \alpha)\big)$$

Now we set α as $\alpha(x, \beta)$, where $\alpha(x, \beta)$is the βquantile of loss distribution, so it's obvious $\varphi\big(x, \alpha(x, \beta)\big) = \beta$

Thus, $\emptyset(x, \beta)$at $\alpha(x, \beta)$ is$\frac{1}{1-\beta} \int_{f(x,y)>\alpha} (f(x, y) - \alpha)p(y)dy + \frac{\alpha}{1-\beta} \varphi\big(x, \alpha(x, \beta)\big)$ = $\frac{1}{1-\beta} \int_{f(x,y)>\alpha} (f(x, y) - \alpha)p(y)dy + \alpha$. We denote this function as $F_\beta(x, a)$.

$$\frac{dF_\beta(x, a)}{d\alpha} = \frac{1}{1-\beta}(\varphi(x, \alpha) - 1) + 1$$

F.O.C:$\frac{dF_\beta(x,a)}{d\alpha} = 0, thus \alpha \quad = \alpha(x, \beta)$

It shows that $\alpha(x,\beta)$ minimizes the function $F_\beta(x,a)$

From the derivation above, here is an equation holds:

$$\emptyset(x,\beta) at \alpha(x,\beta) \stackrel{\text{def}}{=} F_\beta\big(x, a(x,\beta)\big) = \min_{\alpha} F_\beta(x,a)$$

The αthat minimizes the function is just $\alpha(x,\beta)$, the quantile of loss function, thus we set it as *quantile(f(x,y),β)*. Finally we can minimize portfolio CVaR as:

$$\big(f(x,y) - quantile(f(x,y),\beta)\big)^{+p(y)dy} + quantile(f(x,y),\beta)\ \frac{1}{1-\beta}\int$$

Discretization

Suppose there are J days data to be used, we can approximate the formula above into discrete form.

The integral can be seen as the expectation of $+(f(x,y) - \alpha)$, where we can replace with sample mean, thus the formula can be written as:

$$F_\beta(x,a) \approx quantile(< -r, x >, 0.95)$$
$$+ \frac{1}{1-\beta}\Big(\frac{1}{J}\sum_{j=1}^{J}(f(x,yj) - quantile(< -r, x >, 0.95))^{+}\Big)$$

As a result, the formula above is obviously a feasible objective function to be optimized.

4.2 Efficient Frontier

Equivalent to the definition of efficient frontier in Markowitz theorem, the only different thing is variance is replaced with CVaR of portfolio. To draw the frontier, we set the following optimizing problem.

$$\min_x F_\beta(x,a)$$

$$\text{s.t.} \sum \; xi = 1$$

$$\sum \; xiri = \text{R}$$

$$0 < x_i < 0.3$$

We variate R — the portfolio expected return, and get corresponding CVaR.

Then draw the graph of R v.s. CVaR to get the efficient frontier.

The result shows as following:

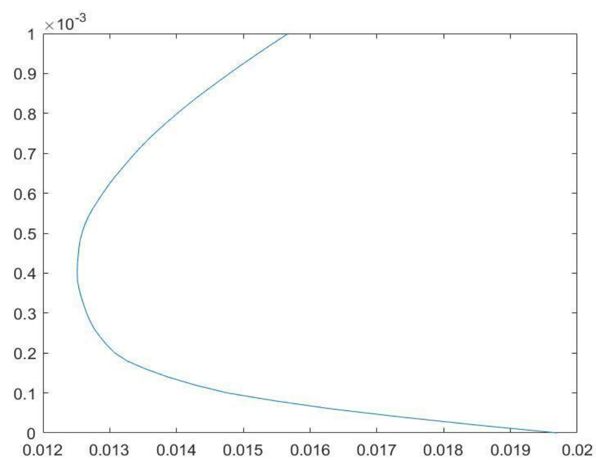


Figure 1 α=0.95

4.3 Optimization problem setting

In our optimization problem, the objective function is *CVaR – a *E(R)*, where a is risk aversion parameter, here we set it as 0.5.

Then the optimizing problem is:

$$\min_x F_\beta(x,a) - \sum xiri$$

$$\text{s.t.} \sum xi = 1$$

$$0 < x_i < 0.3$$

To set problem more realistic and get more reasonable results, stocks cannot be sold short and the maximum allocated weight of a single stock is 30%.

4.3.1 Bootstrapping

When using factor model to simulate the portfolio return distribution for next day, in this part, we use bootstrapping to simulate distribution of factors tomorrow from history data, then plug simulated factors in the regression equation with adding a simulated idiosyncratic risk of companies, we can get a bunch of simulated returns used in our optimization problem.

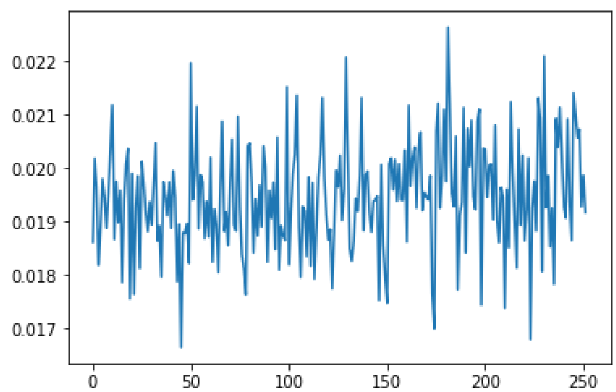
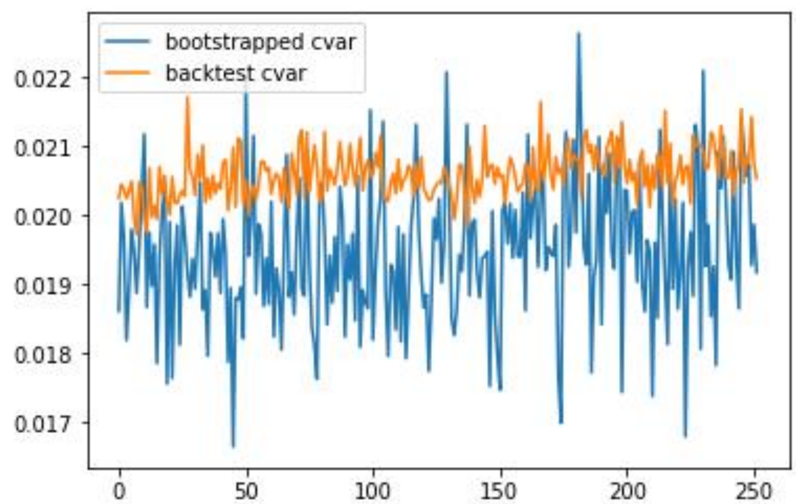


Figure 2 minimized cvar from bootstrapping

4.3.2 Backtest on CVaR

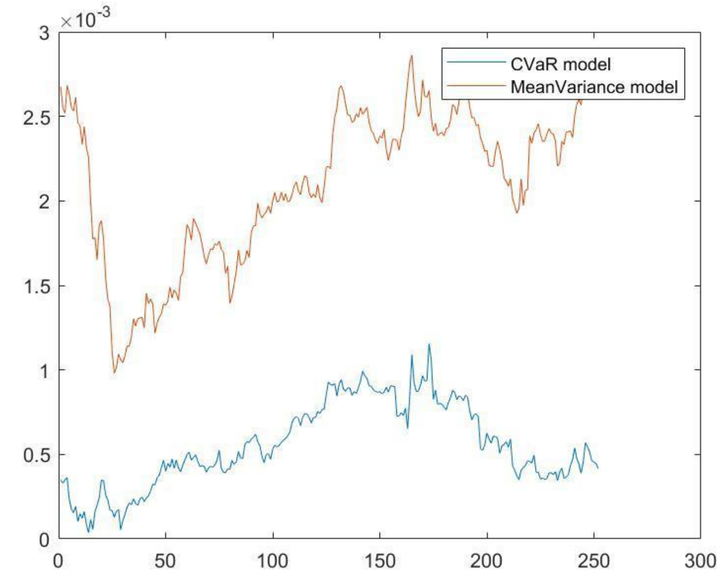
We use the daily data from 2012 to 2015 to bootstrap, and get daily simulated portfolio return distribution of year 2016. To backtest on it, we use actual stock return data to compute CVaR again.



Because bootstrapping is to resample from history data, it won't surely be able to simulate a precise distribution of tomorrow returns, thus the backtested cvar is slightly bigger than using bootstrapped data.

4.3.3 Comparison with Mean-Variance Model

To verify we do have the optimal portfolio CVaR, we compare our model with mean-variance model. Use the same historical data, firstly we minimize portfolio variance minus 0.5 times portfolio expected return, like classical method, get the weights, and calculate the portfolio CVaR. We also use the model above to minimize portfolio CVaR. Compare these two CVaRs, the graph is as following:



As the graph shows, our portfolio CVaR is actually smaller than the value of CVaR in mean-variance models. It succeeds to get an optimal CVaR.

5 Model Comparison

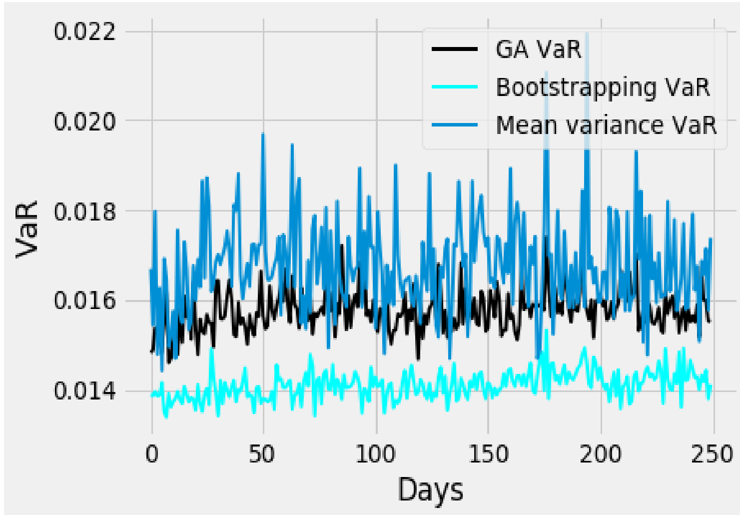
We are doing the following models to get the optimized weight of portfolio:

A) Minimizing the CVaR with bootstrapping method and then calculate the VaR under the minimum CVaR, get the weight w_1

B) Minimizing VaR directly through simulation and genetic algorithm , get the weight w_2

C) Using the typical Mean-Variance model, get the weight w_3

For each weight, we calculate the portfolio distribution using rolling window of historical data and use 95% quantile as VaR, then we can acquire VaR_1, VaR_2, VaR_3respectively for weight w_1, w_2, w_3. The time horizon is one year of 252 observations back test.



From this plot we find that mean variance model has the highest VaR with its weight and genetic algorithm would decrease our VaR to some degree. And the indirect method to first minimize CVaR by bootstrapping actually achieved better optimization of minimizing the VaR.

Moreover, VaR of mean variance model has greater volatility compared with other two models which means bigger risk of extremely large VaR.

6 Model deficiency

6.1 For genetic algorithm:

(1) Genetic algorithm is gradient free method therefore we cannot promise a convergence of minima in our method

(2) The model is based on factor simulation therefore cost a lot when simulate thousands of trials to get portfolio return distribution. One of the solution for acceleration is to do parallel computation.

6.2 For CVaR algorithm:

(1) When using bootstrapping to simulate factors value, we resample from the history. Here we made an assumption that the future situations are the same as historical, which might be problematic.

7 Conclusion

We actually control downside risk by minimizing VaR and CVaR.

As VaR is not a convex function, it's hard to find the global minimum. So when thinking about minimizing VaR, we can firstly minimize CVaR, which turns out to have a better performance.

8 Reference

[1] Alexei A. Gaivoronski, Georg Pflug Value-at-Risk in Portfolio Optimization: Properties and Computational Approach. Journal of Risk, Vol 7, No.2, PP. 1-31, Winter 2004-2005.

[2] Onur Babat. Essays on risk management in portfolio optimization and gas supply networks. Lehigh University, May, 2017

[3] Nataˇsa Kreji´, Miles Kumaresan, Andrea Roznjik. VaR optimal portfolio with transaction costs. University of Novi Sad, February, 2015

[4] J.Samnel Baixauli-Soler, Eva Alfaro-Cid, Matilde O. Fernandez-Blanco. Mean-VaR portfolio Selection Under Real Contraints. Computational Economics (2011) 37: 113-131

[5] Vinay Kaura. Portfolio Optimization Using Value at Risk. Imperial University. June, 2018

[6] Pavlo Krokhmal, Jonas Palmquist, Stanislav Uryasev. Portfolio optimization with conditional value-at-risk objective and constraints. University of Florida& McKinsey. Sep, 2001

[7] Lobo, Miguel Sousa, Maryam Fazel, and Stephen Boyd. Portfolio optimization with linear and fixed transaction costs. Annals of Operations Research 152.1 (2007): 341-365.

[8] Sarykalin, Sergey, Gaia Serraino, and Stan Uryasev. Value-at-risk vs. conditional value-at-risk in risk management and optimization. State-of-the-art decision-making tools in the information-intensive age. Informs, 2008. 270-294.