Robust Portfolio Optimization With Uncertainty in Risk Measure based on Worst Case Value-at-Risk

A Project Report by

Sidharth Gulati and Tushar Sudhakar Jee

June 8, 2016

1 Introduction

A classical problem in computational finance is the financial decision-making under model uncertainty, guaranteed (robust) Portfolio optimization deals with optimal portfolio selection of finitely many risky assets to maximize expected return of the investment subject to controlled "risks". Developed by Harry Markowitz (1952) five decades prior,this approach quantifies the trade-off between the expected return and the risk of portfolios of financial assets using mathematical techniques. The approach is also called Mean Variance Optimization (MVO) as risk is measured by the variance of random portfolio return. Despite being an elegant model, it has been subjected to much skepticism by investment practitioners on its practicality due to the fact that optimal portfolios are often sensitive to changes in the input parameters of the problem(expected returns and the covariance matrix) indicating inputs to the MVO model to be very accurately estimated.

Robust optimization, an emerging branch in the field of optimization, offers vehicles to incorporate estimation risk into the decision making process in portfolio allocation. Generally speaking, robust optimization refers to finding solutions to given optimization problems with uncertain input parameters that will achieve good objective values for all, or most, realizations of the uncertain input parameters.

2 Problem description

The main goal of this project is to present an efficient computational framework for robust portfolio selection in the situation of asset returns described by an ambiguous discrete joint probability distribution. In the Markowitz approach site, it is assumed that the mean r and covariance matrix Γ of the return vector are both known, and risk is defined as the variance of the return. Minimizing the risk subject to a lower bound on the mean return leads to the familiar problem given by :

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & x^\top \Gamma x \\
\text{subject to} \quad & x \in \boldsymbol{\chi}
\end{aligned}
$$

where, x is the allocation vector for the stocks considered in the portfolio and the set, χ, is given by :

$$
\boldsymbol{\chi} = \{x : x \succeq 0, 1^\top x = 1\} \tag{1}
$$

In this project, we consider risk measure given by the Value-at-Risk (VaR) framework. The VaR looks at the probability of losses and is defined as the minimal level $\boldsymbol{\gamma}$ such that the probability that the portfolio loss **-l(x,r)** exceeds $\boldsymbol{\gamma}$ is below $\boldsymbol{\epsilon}$:

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & \gamma \\
\text{subject to} \quad & \mathbf{Prob}\{\gamma \le -l(x,r)\} \le \epsilon
\end{aligned}
$$

where $\boldsymbol{\epsilon} \in (0, 1]$ is given. In contrast to the Markowitz framework, that requires knowledge of the first and second moments of the distribution of returns only, the VaR above assumes that the entire distribution is perfectly known. With given distribution as Gaussian,mean as r and covariance matrix as Γ , the VaR can be expressed as:

$$
V(x) = \kappa(\epsilon)\sqrt{x^\top \Gamma x} - r^\top x \tag{2}
$$

where

$$
\kappa(\epsilon) = -\boldsymbol{\Phi}^{-1}(\epsilon) \tag{3}
$$

is an appropriate *risk factor* depending on prior assumptions on distribution of returns(Gaussian, arbitrary with given moments, etc.). When $\kappa \geq 0$(which in the Gaussian case is true if and only if $\epsilon \in (0, 1/2]$, *V(x)* is a convex function of x, making it easy to solve globally using, for example, interior-point techniques for convex or second-order cone programming. In this project, our approach is to assume that the true distribution of returns is only partially known. We denote the set of allowable distributions by π which can be chosen as per the requirement and based on the dataset. For a given loss probability level $\epsilon \in (0, 1]$, and a given portfolio $x \in \boldsymbol{\chi}$, we define the *worst-case Value-at-Risk* with respect to the set of probability distributions π as

$$
V_\pi(x) = \inf\{\gamma \mid \lambda(\gamma, x, r) \le \boldsymbol{\epsilon}\}
$$

where,

$$
\lambda(\gamma, x, r) = \sup\{\mathbf{Prob}\{\boldsymbol{\gamma} \le -l(x,r)\}\} \tag{4}
$$

where the **sup** in the above expression is taken with respect to all probability distributions in π . The corresponding ***robust portfolio optimization*** problem is to solve :

$$
V_\pi^{opt}(x) = \inf\{\ V_\pi(x) \mid x \in \boldsymbol{\chi}\ \} \tag{5}
$$

The key point in this project is to consider the probability distribution π to be imprecisely known. In particular, we assume that a nominal value η for the distribution is given, but that the actual π is only known to lie in a region at distance no larger than d from its nominal value, where d is a user-definable parameter that quantifies the (lack of) confidence in the nominal probability. To measure the distance among distributions, we use the standard metric given by the *Kullback–Leibler divergence*.

Assume that a nominal return probability distribution η is given, for instance, as a result of estimation from samples. If π, η are two probability vectors in $\mathbb{R}^T$, with η >0 describing the nominal probability, the KL distance between π and η is defined as

$$\mathrm{KL}(\pi, \eta) = \sum_{k=1}^{T} \pi_k \log \frac{\pi_k}{\eta_k} \tag{6}$$

We shall henceforth assume that the true probability π is only known to lie within KL distance $d \geq 0$ from η, i.e., $\pi \in \mathrm{K}(\eta,\, d)$, where

$$\mathrm{K}(\eta, d) = \{\pi \in \Pi : \mathrm{KL}(\pi, \eta) \leq d\}, \tag{7}$$

where,

$$\Pi = \{\pi : \pi \succeq 0, 1^\top \pi = 1\}, \tag{8}$$

Specifically, we assume that the given the ambiguity model K(η,d) follows Gaussian Distribution that is , η and π follows Gaussian Distribution and η has a mean r and covariance Γ. Under this assumption, it has been already shown that the *worst-case VaR* is given by [1] :

$$V_\pi(x) = \kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2 - r^\top x \tag{9}$$

where $\kappa(\epsilon, d)$ is given by :

$$\kappa(\epsilon, d) = -\mathbf{\Phi}^{-1}(f(\epsilon, d)) \tag{10}$$

$$f(\epsilon, d) = \sup_{\nu > 0} \frac{e^{-d}(\nu + 1)^{\epsilon} - 1}{\nu} \tag{11}$$

Thus, the **robust portfolio optimization problem** is given by :

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad \kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2 - r^\top x \\ &\text{subject to} \quad x \in \boldsymbol{\chi} \\ \iff \quad &\underset{x}{\text{minimize}} \quad \kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2 - r^\top x \\ &\text{subject to} \quad 1^\top x = 1 \\ &\qquad\qquad\quad\; x \succeq 0 \end{aligned}$$

3 Approach

In our project, we have implemented large scale optimization algorithms that were taught in the class that is, in EE236C (Optimization Methods for Large Scale Systems). The aim would be to provide a numerical comparison between the algorithms based on the number of iterations used in respective algorithm to provide an optimal value and also, comparing the time taken by the algorithms. The methods implemented for robust portfolio optimization are :

3.1 Douglas Rachford Algorithm

The robust portfolio optimization problem given by :

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2 - r^\top x \\
\text{subject to} \quad & 1^\top x = 1 \\
& x \succeq 0
\end{aligned}$$

can be written as :

$$\underset{x}{\text{minimize}} \quad \underbrace{\kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2}_{\text{g(x)}} \underbrace{-r^\top x + \delta_{\boldsymbol{\chi}}(x)}_{\text{h(x)}} \tag{12}$$

where $\delta_{\boldsymbol{\chi}}(x)$ is the indicator function of the set $\boldsymbol{\chi}$.

As, it is evident both $g(x)$ and $h(x)$ are closed, non differentiable convex functions. Hence, we will use Douglas Rachford Algorithm to solve the above optimization problem. The proximal operator for both $g(x)$ and $h(x)$ are given below :

3.1.1 Proximal Operator of $g(x)$

$$g(x) = \kappa(\epsilon, d) \left\|\Gamma^{1/2} x\right\|_2 = \kappa(\epsilon, d) \sup_{\{z^\top \Gamma^{-1} z \leq 1\}} z^\top x \tag{13}$$

As it is evident that, the function, given by,

$$\delta^*_{\chi}(x) = \sup_{\{z^\top \Gamma^{-1} z \leq 1\}} z^\top x \tag{14}$$

is the conjugate function of the set C given by :

$$C = \{z : z^\top \Gamma^{-1} z \leq 1\} \tag{15}$$

From Lecture 8 slides, we know from Mareau Decomposition,

$$x = \text{prox}_{tg}(x) + t\text{prox}_{t^{-1}g*}(x) \tag{16}$$

Thus,

$$\begin{aligned}
\text{prox}_{tg}(x) &= x - t\text{prox}_{t^{-1}g*}(x) \\
&= x - tP_C(x/t)
\end{aligned} \tag{17}$$

The projection on Ellipse can be solved by analytically solving below optimization problem :

$$P_C(x/t) = \inf\{(1/2t) \left\|u - x\right\|_2^2 \mid u^\top \Gamma^{-1} u \leq 1\}$$

$$\Longleftrightarrow P_C(x/t) = \inf\{(1/2t) \left\|u - x\right\|_2^2 \mid u^\top \Lambda^{-1} u \leq 1\}$$

where, Λ is the Eigenvalue decomposition of Γ.

Let $P_C(x)$ be the projection on the Ellipse defined by equation 15. Then, the projection, $P_C(x)$ is given by :

$$P_C(x) = \begin{cases} x & x^\top \Lambda^{-1} x < 1 \\ (I + t\Lambda^{-1})^{-1} x & \text{otherwise} \end{cases} \tag{18}$$

where t is the solution of the nonlinear equation defined by :

$$f(t) = x^\top (I + t\Lambda^{-1})^{-1} \Lambda^{-1} (I + t\Lambda^{-1})^{-1} x - 1$$

3.1.2 Proximal Operator of $h(x)$

The proximal operator of $h(x)$ is calculated using Proximal Gradient method. and is given by :

$$\mathrm{prox}_{th}(x) = \underset{u \in \boldsymbol{\chi}}{\mathrm{argmin}} \quad (1/2t) \left\| u - x \right\|_2^2 - r^\top u \tag{19}$$

Thus, now we know proximal operator of both the convex functions $f(x)$ and $g(x)$ respectively, hence, the steps for Douglas Rachford Algorithm can be referenced from EE236C Lecture Slides.

3.2 Proximal gradient Algorithm

The robust portfolio optimization problem given by :

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \kappa(\epsilon, d) \left\| \Gamma^{1/2} x \right\|_2 - r^\top x \\ \text{subject to} & 1^\top x = 1 \\ & x \succeq 0 \end{array}$$

$$\Longleftrightarrow \begin{array}{ll} \underset{x}{\text{minimize}} & \kappa(\epsilon, d) x^\top \Gamma x - r^\top x \\ \text{subject to} & 1^\top x = 1 \\ & x \succeq 0 \end{array}$$

$$\Longleftrightarrow \underset{x}{\text{minimize}} \quad \underbrace{\kappa(\epsilon, d) x^\top \Gamma x - r^\top x}_{\text{g(x)}} \quad + \underbrace{\delta_{\boldsymbol{\chi}}(x)}_{\text{h(x)}} \tag{20}$$

where $\delta_{\boldsymbol{\chi}}(x)$ is the indicator function of the set $\boldsymbol{\chi}$.

3.2.1 Proximal Operator of $h(x)$

The proximal operator of $h(x)$, $\mathrm{prox}_h(x)$,is the projection on Probability Simplex defined by the set $\boldsymbol{\chi}$ and the projection $P_{\boldsymbol{\chi}}(x)$is given by :

$$P_{\boldsymbol{\chi}}(x) = \mathrm{prox}_{th}(x) = \underset{u \in \boldsymbol{\chi}}{\mathrm{argmin}} \quad (1/2t) \left\| u - x \right\|_2^2 \tag{21}$$

$$\Longleftrightarrow P_{\boldsymbol{\chi}}(x) = (x - \lambda \mathbf{1})_+ \tag{22}$$

where, λ is the solution of the equation :

$$\mathbf{1}^\top (x - \lambda \mathbf{1})_+ = 1 \tag{23}$$

3.3 Accelerated Proximal gradient method

The robust portfolio optimization problem given by :

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \kappa(\epsilon, d) \left\| \Gamma^{1/2} x \right\|_2 - r^\top x \\ \text{subject to} \quad & 1^\top x = 1 \\ & x \succeq 0 \end{aligned}$$

$$\begin{aligned} \Longleftrightarrow \underset{x}{\text{minimize}} \quad & \kappa(\epsilon, d) x^\top \Gamma x - r^\top x \\ \text{subject to} \quad & 1^\top x = 1 \\ & x \succeq 0 \end{aligned}$$

$$\Longleftrightarrow \underset{x}{\text{minimize}} \quad \underbrace{\kappa(\epsilon, d) x^\top \Gamma x - r^\top x}_{\text{g(x)}} \quad + \underbrace{\delta_{\boldsymbol{\chi}}(x)}_{\text{h(x)}} \tag{24}$$

where $\delta_{\boldsymbol{\chi}}(x)$ is the indicator function of the set $\boldsymbol{\chi}$.

3.3.1 Proximal Operator of $h(x)$

The proximal operator of $h(x)$ is calculated using Proximal Gradient method and is given by :

$$\text{prox}_{th}(x) = \underset{u \in \boldsymbol{\chi}}{\text{argmin}} \quad (1/2t) \left\| u - x \right\|_2^2 \tag{25}$$

The $\text{prox}_h(x)$ is the projection on Probability Simplex defined by the set $\boldsymbol{\chi}$ and the projection $P_{\boldsymbol{\chi}}(x)$is given by :

$$P_{\boldsymbol{\chi}}(x) = (x - \lambda \mathbf{1})_+ \tag{26}$$

where, λ is the solution of the equation :

$$\mathbf{1}^\top (x - \lambda \mathbf{1})_+ = 1 \tag{27}$$

3.4 FISTA Algorithm

The robust portfolio optimization problem given by :

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \kappa(\epsilon, d) \left\| \Gamma^{1/2} x \right\|_2 - r^\top x \\ \text{subject to} \quad & 1^\top x = 1 \\ & x \succeq 0 \end{aligned}$$

$$
\begin{aligned}
\iff & \underset{x}{\text{minimize}} & & \kappa(\epsilon, d) x^\top \Gamma x - r^\top x \\
& \text{subject to} & & 1^\top x = 1 \\
& & & x \succeq 0
\end{aligned}
$$

$$
\iff \underset{x}{\text{minimize}} \quad \underbrace{\kappa(\epsilon, d) x^\top \Gamma x - r^\top x}_{\text{g(x)}} \quad + \underbrace{\delta_{\boldsymbol{\chi}}(x)}_{\text{h(x)}} \tag{28}
$$

where $\delta_{\boldsymbol{\chi}}(x)$ is the indicator function of the set $\boldsymbol{\chi}$.

3.4.1 Proximal Operator of $h(x)$

The proximal operator of $h(x)$ is calculated using Proximal Gradient method and is given by :

$$
\text{prox}_{th}(x) = \underset{u \in \boldsymbol{\chi}}{\text{argmin}} \quad (1/2t) \left\| u - x \right\|_2^2 \tag{29}
$$

The $\text{prox}_h(x)$ is the projection on Probability Simplex defined by the set $\boldsymbol{\chi}$ and the projection $P_{\boldsymbol{\chi}}(x)$is given by :

$$
P_{\boldsymbol{\chi}}(x) = (x - \lambda \mathbf{1})_+ \tag{30}
$$

where, λ is the solution of the equation :

$$
\mathbf{1}^\top (x - \lambda \mathbf{1})_+ = 1 \tag{31}
$$

4 Result

4.1 Algorithmic Comparison

For Algorithmic Comparison, we performed our experiment on random data generated using MATLAB. As described in previous Section, we implemented following 4 algorithms :

1. Douglas Rachford Algorithm
2. Proximal Gradient Method
3. Simplest Nestrov's Method
4. FISTA Algorithm

The list of parameters used in each of the Large Scale Optimization Methods implemented is given below :

1. **Douglas Rachford Algorithm**
   (a) Stepsize, $t_k = 1/\lambda_{max}(\Gamma^{1/2})$
   (b) Stopping Criteria, $y = F(y)$

2. **Proximal Gradient Method**
   (a) Stepsize, $t_k = 1/\lambda_{max}(\Gamma^{1/2})$
   (b) Stopping Criteria, $0 \in G_t(u)$
3. **Simplest Nestrov's Method**
   (a) Stepsize, $t_k = 1/\lambda_{max}(\Gamma^{1/2})$
   (b) Stopping Criteria, $|f(x^k) - f(x^*)| \leq \epsilon$
4. **FISTA Algorithm**
   (a) Stepsize, $t_k = 1/\lambda_{max}(\Gamma^{1/2})$
   (b) Stopping Criteria, $|f(x^k) - f(x^*)| \leq \epsilon$
   (c) $\gamma_0 = m$, where, $m = \lambda_{min}(\Gamma^{1/2})$

The details of the parameters can also be found in the corresponding lecture slides of the course EE236C. We illustrate our findings based on 2 comparison parameters namely, Error Convergence and Time taken for optimization by each method. The plot for Error Convergence, that is, $(f(x^k) - f(x^*))/f(x^*)$ for each of the method are plotted below.

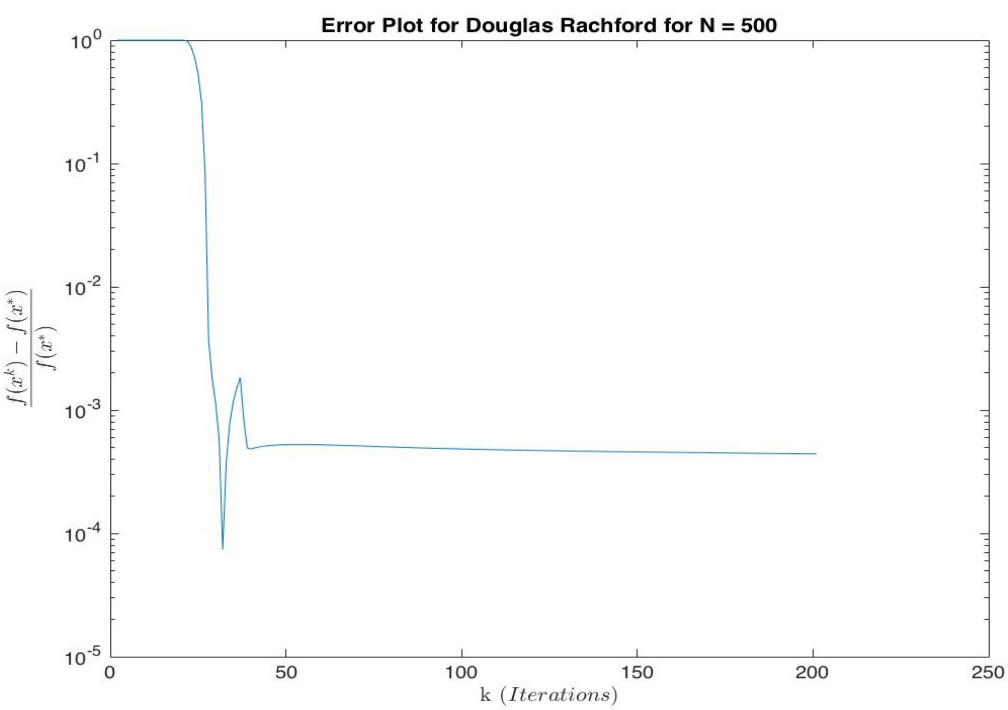

Figure 1: Error Convergence for Douglas Rachford Algorithm for N=500

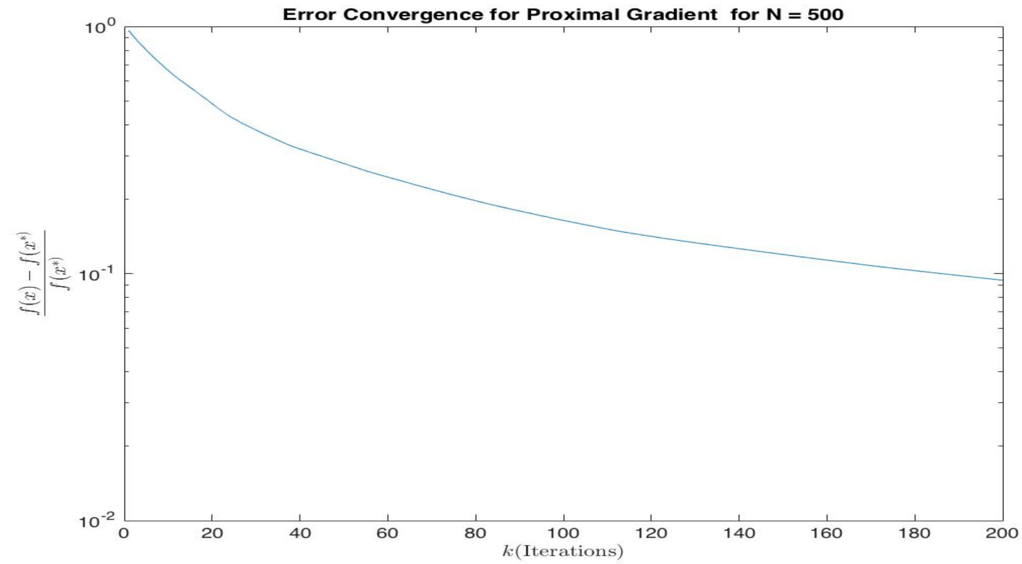

Figure 2: Error Convergence for Proximal Gradient Algorithm for N=500

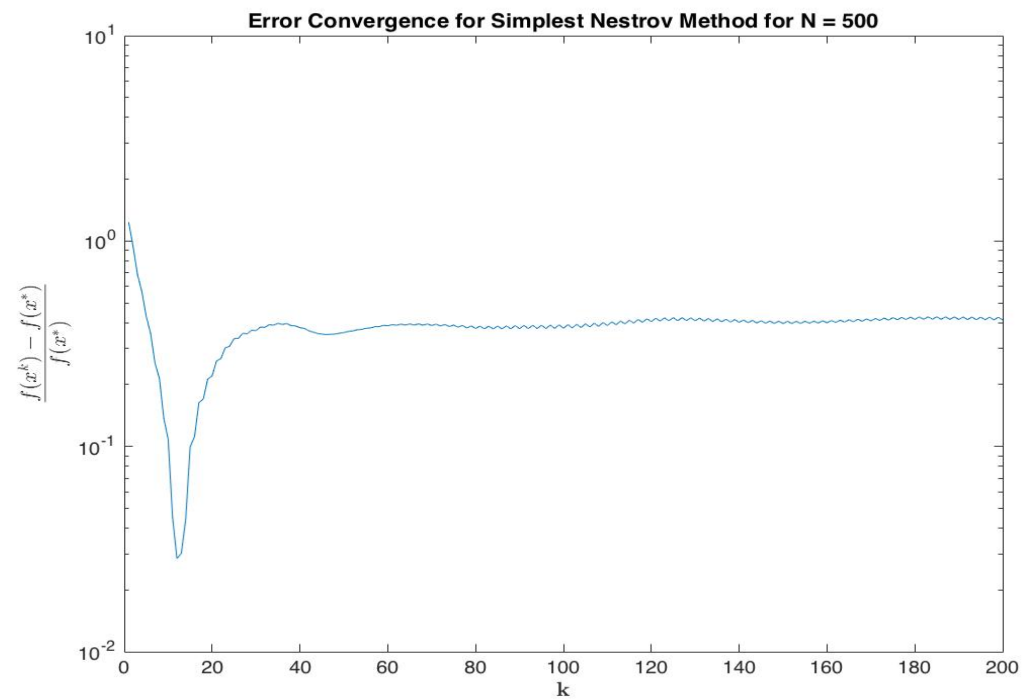

Figure 3: Error Convergence for Nestrov's Algorithm for N=500

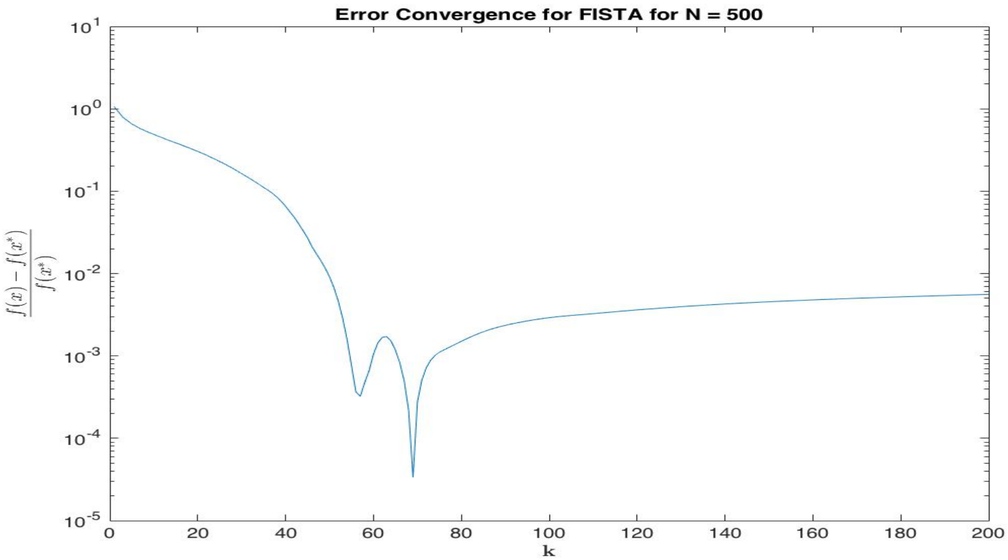

Figure 4: Error Convergence for FISTA Algorithm for N=500

The comparison between all the Large Scale Optimization methods is plotted below.

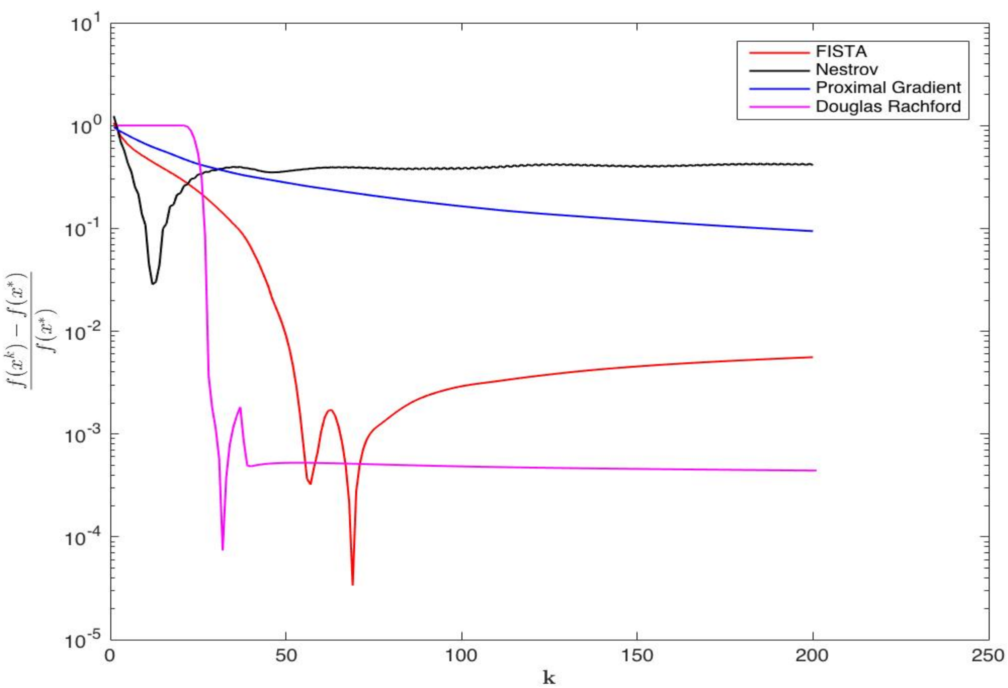

Figure 5: Error Convergence Comparison for N=500

From the above comparison plot, we found that Douglas Rachford Algorithm has the best convergence followed by FISTA algorithm.

The next parameter of comparison between the optimization methods implemented in this experiment is the comparison between the time taken by each optimization method and time taken by CVX to implement the optimization problems. The total time plot for each of the method compared with CVX are plotted below.

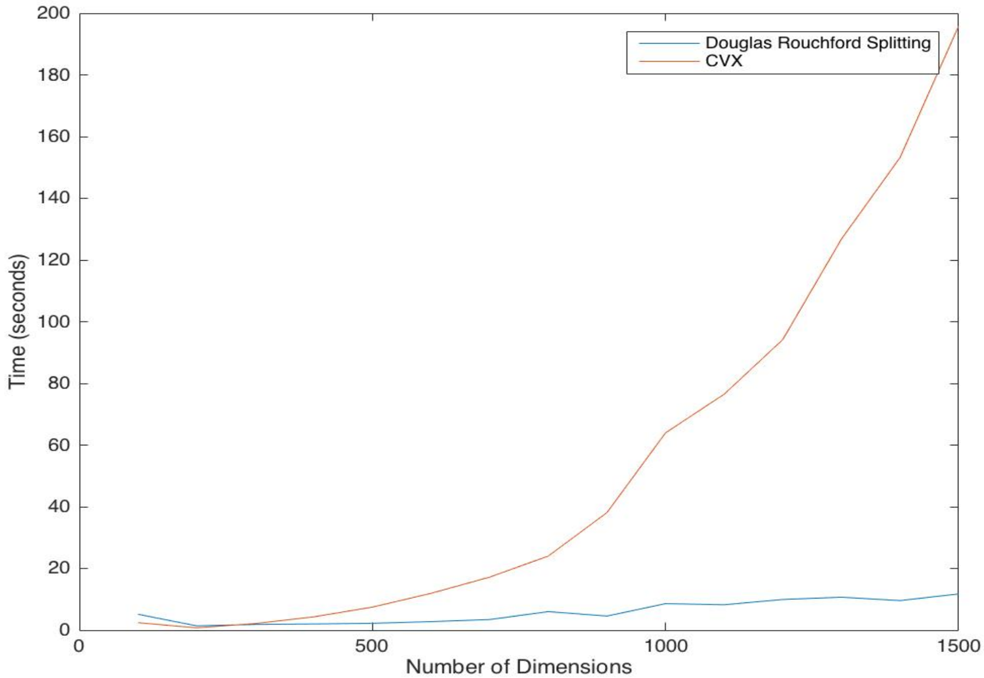

Figure 6: Time Comparison for Douglas Rachford Algorithm

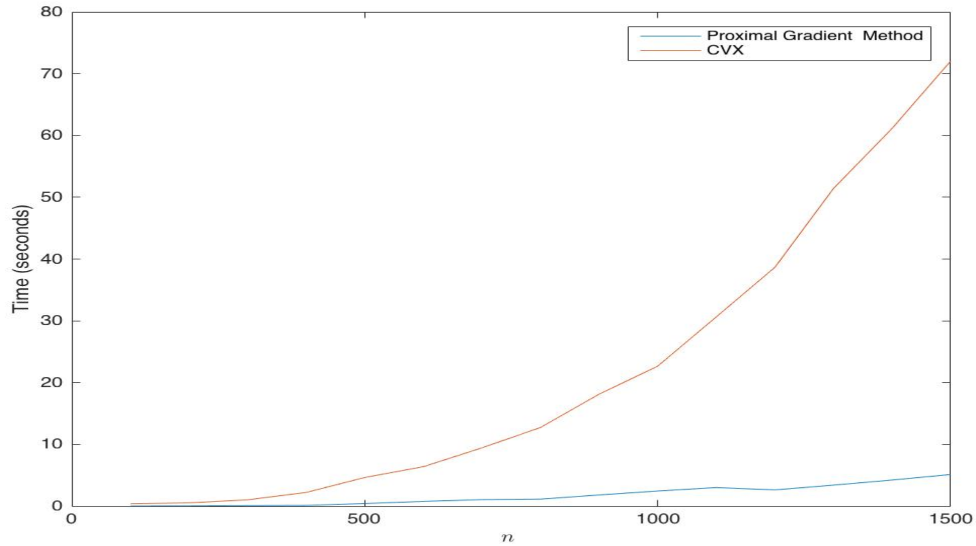

Figure 7: Time Comparison for Proximal Gradient Algorithm

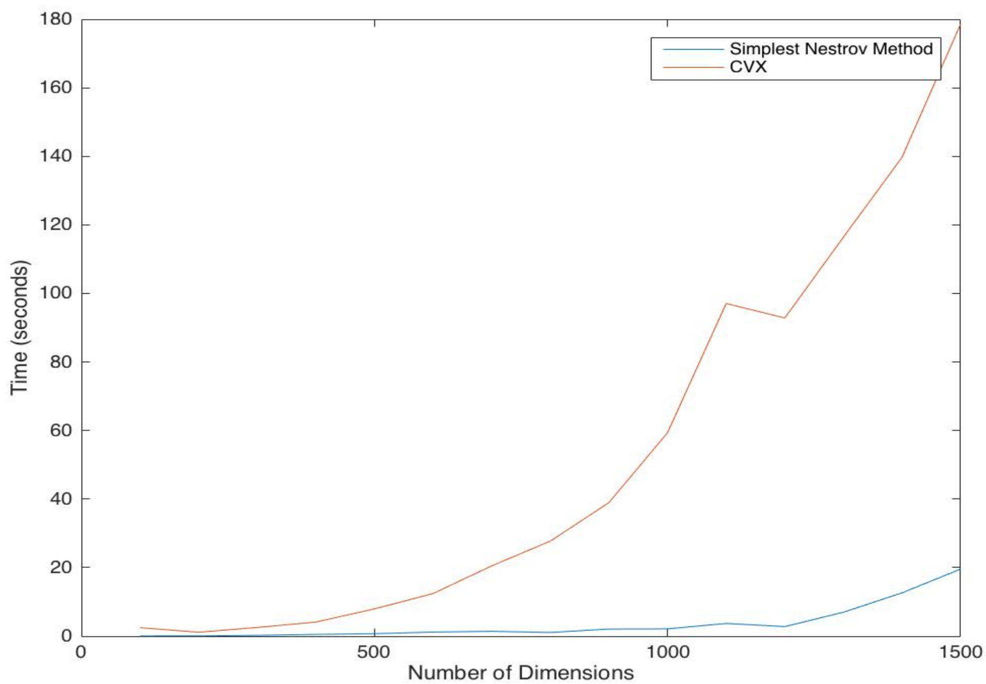

Figure 8: Time Comparison for Simplest Nestrov Algorithm

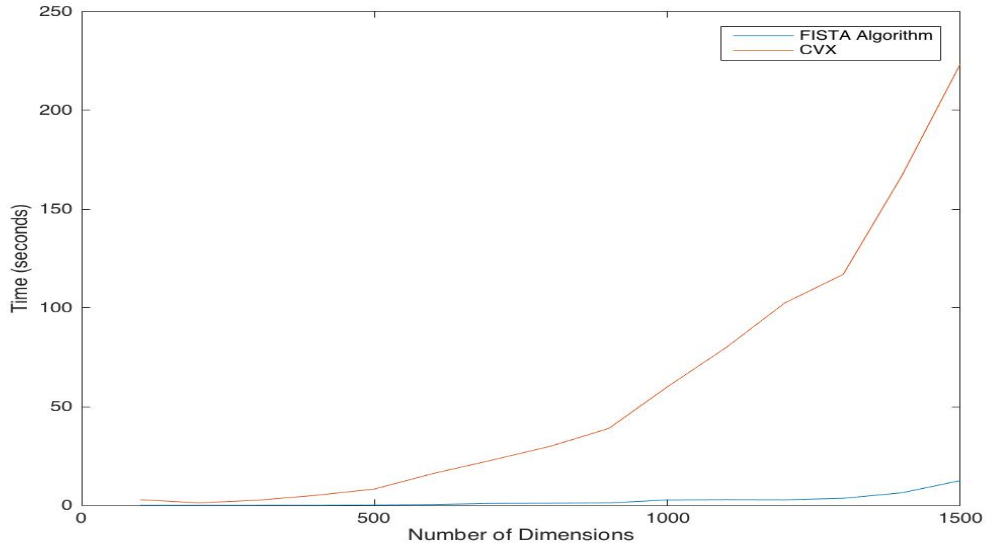

Figure 9: Time Comparison for FISTA Algorithm

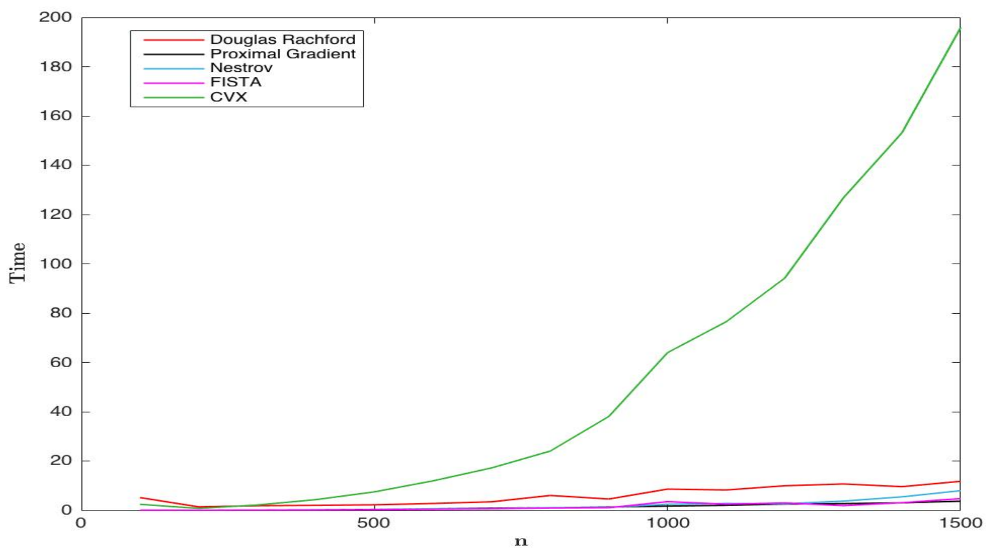

Figure 10: Performance Comparison based on Time taken

As it is evident from the above plots, each optimization provides a lower bound on the time taken by CVX to perform the optimization. The performance comparison based on time taken, shows that FISTA and Proximal Gradient Method have a better time performance as compared to the other optimization methods implemented.

4.2 Numerical Example

We downloaded the adjusted closing price for different stocks ($n = 28$) from 31-Dec-2008 to 31-Dec-2013 from Yahoo! Finance. We then calculated the returns for each stock and calculated their Covariance Matrix (Γ). As a next step, we implemented the robust Value-at-Risk to get the optimal composition and compared the portfolio possibility curve of robust Value-at-Risk model with the classical Markowitz Model. The below table, illustrates the optimal composition for robust Value-at-Risk model and classical Markowitz problem.

Company Name(Ticker)	**Optimal Compositions**	
	Robust Value at Risk	**Markowitz model**
Oil & Natural Gas Corporation L (ONGC.NS)	3.77E-07	1.39E-09
Green Dragon Gas (GDG.L)	0.005606225	0.00514701
NMDC Limited (NMDC.NS)	0.011485115	2.86E-09
Asian Paints Limited (ASIANPAINT.NS)	0.034289486	0.120327172
E. I. du Pont de Nemours and Company (DD.MX)	1.66E-07	6.57E-09
Oracle Corporation (ORCL)	4.47E-07	1.72E-09
HP Inc. (HPQ.MX)	0.011568426	1.09E-09
Electronics for Imaging, Inc. (EFII)	1.28E-07	1.88E-09
AT&T, Inc. (T)	5.08E-07	1.71E-09
Verizon Communications Inc. (VZ)	6.49E-07	3.45E-09
The Dun & Bradstreet Corporation (DNB)	0.055838821	0.025711526
Lancashire Holdings Limited (LRE.L)	0.162493694	0.133687455
London Stock Exchange Group plc (LSE.L)	2.86E-07	3.18E-09
HDFC Bank Limited (HDB)	1.66E-07	9.65E-10
The Goldman Sachs Group, Inc. (GS.MX)	0.018314789	2.45E-09
Jardine Lloyd Thompson Group plc (JLT.L)	0.037283524	0.05918331
Shire plc (SHP.L)	0.048481839	0.130189584
Sun Pharmaceutical Industries Ltd. (SUNPHARMA.NS)	2.59E-07	0.038351156
Johnson & Johnson (JNJ.MX)	0.168829101	0.083411261
Pfizer Inc. (PFE.MX)	0.139986142	0.107862109
UnitedHealth Group Incorporated (UNH)	0.040897382	0.121298822
Agilent Technologies, Inc. (A)	3.12E-06	0.01392153
The Boeing Company (BA.MX)	2.70E-07	0.012214118
General Electric Company (GE.MX)	2.46E-07	1.27E-08
3M Company (MMM.MX)	4.20E-07	0.004624611
Cheung Kong Infrastructure Holdings Ltd. (1038.HK)	0.235415273	0.144070294
Compagnie de Saint-Gobain S.A. (COD.L)	0.029502634	1.21E-09
CRH plc (CRH.L)	5.07E-07	1.48E-09

The plots for the portfolio possibility curve and portfolio performance are plotted below.

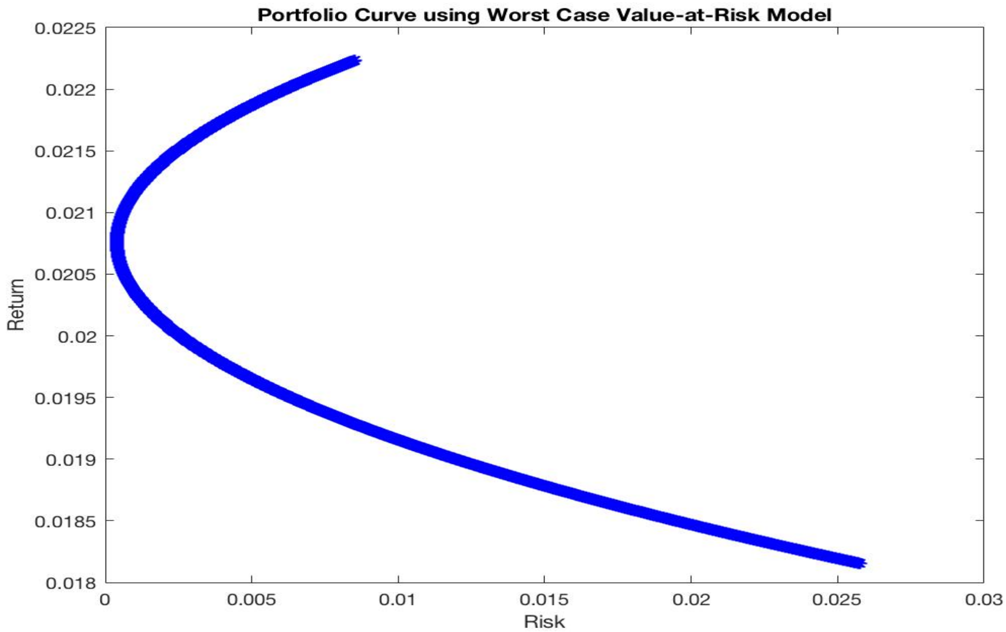

Figure 11: Portfolio Possibility Curve for Value-at-Risk Model

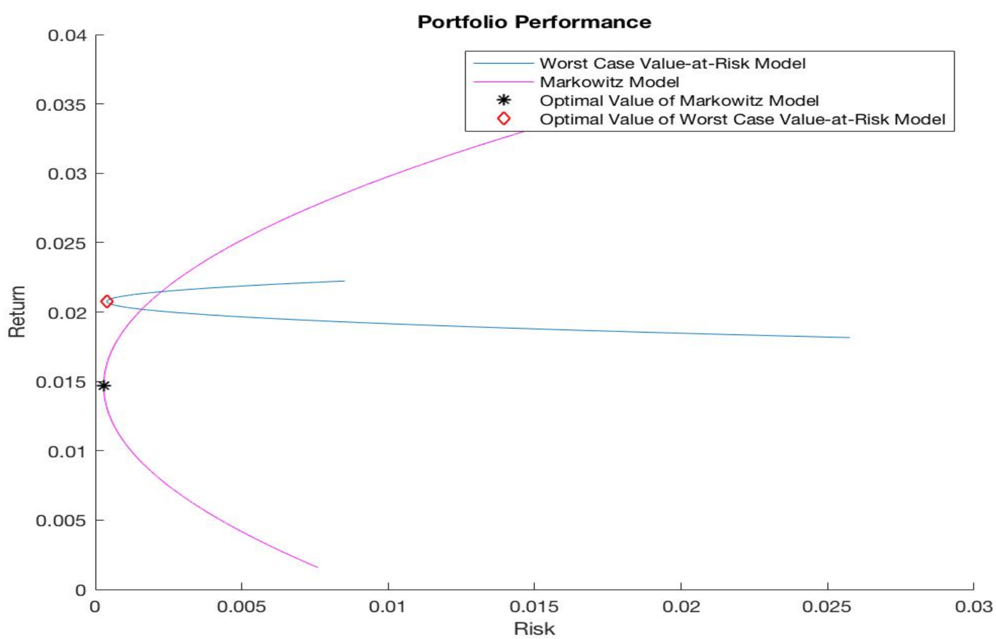

Figure 12: Portfolio Performance

The optimal value of Risk and Expected Return for Markowitz Model and Robust Value-at-Risk Model is tabulated below.

Model	Risk	Expected Return
Markowitz Model	2.8836e-04	0.0147
Robust Value-at-Risk Model	3.9514e-04	0.0208

As it is clear from the above table, Robust Value-at-Risk Model provides a higher expected return at comparable risk of the classical Markowitz Model. This, corroborates our problem statement that robust Value-at-Risk is a better measure for portfolio optimization as compared to classical Markowitz Model.

5 Hardware Specification

We implemented the above algorithms in MATLAB (2015b) on a 2.7GHz i5 Intel processor, OS X El Capitan, with 8GB RAM and 250GB disk storage space.

6 Conclusion

Our method finds reasonable measure of portfolio optimization as compared with the classical Markowitz problem. Our method robustly handles real prices by accounting for the combined effect of value-at-risk measure for the loss and expected return. The KL distance provides a good estimate of the divergence of the probability distribution of stock prices and gives a realistic estimation of the probability distribution of the stock prices. The risk factor acts as a regularization parameter in the optimization problem providing a trade off between risk and expected return. This was also witnessed in the results shown above of expected return and risk for both robust value-at-risk model and classical Markowitz problem.

We also found that the convergence of Douglas Rachford is high as compared to the other methods implemented at a cost ho high time require to do the optimization. Overall, FISTA algorithm finds a good method to implement by observing both the time and convergence parameters.

7 Work Distribution

The work distribution for this experiment is given below :

1. **Paper Review** - Sidharth and Tushar
2. **Problem Formulation** - Sidharth
3. **Douglas Dachford Algorithm** - Sidharth
4. **Proximal Gradient Algorithm** - Tushar
5. **Simplest Nestrov Algorithm** - Tushar
6. **FISTA Algorithm** - Sidharth
7. **Numerical Analysis** - Sidharth
8. **Reporting** - Sidharth and Tushar

8 Acknowledgement

The authors gratefully acknowledge Professor Lieven Vandenberghe for valuable discussions and suggestions on the project which helped us to substantially improve the quality of our work.

References

[1] Laurent El Ghaoui , Maksim Oks and Francois Oustry, *Worst Case Value-at-Risk and Robust Portfolio Optimization: A Conic Programming Approach*, (Operations Research, INFORMS,Vol. 51, No. 4 (2003), pp. 543–556).

[2] Giuseppe C. Calafiore, *Ambiguous risk measures and optimal robust portfolios*, (SIAM J. Optim.,Vol. 18 (2007), No. 3, pp. 853–877).

[3] R. H. Tutuncu and M. Koenig, *Robust Asset Allocation*, (September 2003).

[4] Harry Markowitz, *Portfolio Selection* , (The Journal of Finance, Vol. 7, No. 1. (Mar., 1952), pp. 77–91)

[5] L. Vandenberghe, *EE236C Lecture Slides* , (Spring 2016)