Iterative Solvers for Portfolio Optimization

Anandadeep Mandal

Abstract

In this paper we propose iterative solvers for portfolio optimization in a two dimensional domain. To put the modeled equations into practice we provide Jacobi and Gauss-Seidal algorithms. In order to improve the efficiency of portfolio optimization iterative solvers we study the convergence rate and introduce successive over relaxation scheme to the developed algorithms. Further to overcome the domain bias of this relaxation scheme we propose a symmetric successive relaxation model. This is demonstrated through a Chebyshev acceleration technique. We conclude by stating that iterative solvers are more superior and consistent techniques for portfolio optimization.

Keywords: Iterative solvers, portfolio optimization, successive over relaxation, Chebyshev accelaration, Jacobi and Gauss-Seidel algorithms

1. Introduction

Markowitz pioneered portfolio optimization in his seminal paper in 1952. Since then there has been lot of research in his field related to the development of portfolio optimization models. The basic fundamentals of portfolio optimization are: a) modeling of risk and utility constraints and b) addressing efficiency parameter of the model as the portfolio is subjected to wide variety of instrument and scenario. Authors developed several models for addressing and modeling the above issues related to portfolio optimization. Konno and Wijayanayake (1999) developed the mean absolute deviation approach, Dembo and Rosen (1999) formulated the regret optimization approach and Young (1998) devised the minimax technique. The efficiency of the models using linear programming techniques showed better performance than the quadratic programming approach proposed by Markowitz.

Rockafellar and Uryasev (2000) stated that the linear programming methodology can be used for Tail Value-at-Risk or termed as Conditional Value-at-Risk (CVaR). Uryasev (2000) highlighted in his paper the need to generate equation of constraints for optimization of problems using CVaR. Various authors including Duffie and Pan (1997), Pritsker (1997), Simons (1996) and Stambaugh (1996) have evaluated VaR through linear estimation of portfolio risk under log-normal distribution of underlying constraints. In case of portfolios with optians, other techniques involving Monte Carlo simulations can be effectively used. Bucay and Rosen (1999), Mausser and Rosen (1991) and Stublo Beder (1995) have highlighted the use of simulation in portfolio optimization.

* **Anandadeep Mandal**, Doctoral Scholar, Cranfield School of Management, Cranfield University, England, United Kingdom

Even though VaR is very commonly used, it fails to explain high losses issues related to sub-additivity. Artzner et. al. (1997 and 1999) has shown in their work the various undesirable properties related to VaR method of approximation of risk. The researchers have shown that CVaR has its edge over the other method in estimating portfolio risk and is consistent with sub-additive properties. However Rockafellar and Uryasev (2000) have shown that in case of normal distribution for returns the optimal portfolio estimation for both CVaR and VaR are same.

In this regard the paper proposes iterative solvers as a portfolio optimization technique. The paper further provides Jacobi and Gauss-Seidel algorithm for enabling these iterative solvers to practice and proves its consistency. Further the paper introduces relaxation of the solver to improve the convergence rate of the algorithms employing Chebyshev acceleration technique for gaining more stability and consistent results in multistage schemes for its application in various market environments.

The rest of the paper unfolds as follows: Section 2 develops the iterative solvers as a portfolio optimization tool. Section 3 provides a brief discussion on the algorithms of the above formulated models and section 4 concludes the study.

2. Modeling Iterative Solvers

We can represent an optimization equation with homogeneous boundary conditions with a known function $\Theta_0(x,y)$ and $q(x,t)$:

$$\nabla^2\Theta + q = 0 \text{ on } \Omega \tag{3.1}$$

$$\Theta(x,y) = 0 \text{ on } \partial\Omega \tag{3.2}$$

In order to compute the sequence $\Theta_{ij}^0, \Theta_{ij}^1, \Theta_{ij}^2, ..., \Theta_{ij}^{n-1}, \Theta_{ij}^n, ..., \Theta_{ij}^\infty$ we introduce pseudo-time derivative to the above equations to obtain:

$$\frac{\partial\Theta}{\partial t} = \nabla^2\Theta + q \text{ on } \Omega, \Theta(x,y) = 0 \text{ on } \Omega \tag{3.3}$$

Here Θ_j^∞ is a steady state solution different from the exact solution due to error in formulation of the initial value problem. In this particular case, the steady state error lies due to spatial discretization.

2.1 Developing the Algorithm Equation

Applying Euler-forward discretization[1] in a two-dimensional Ω-domain and assuming $\Delta x = \Delta y$ we get:

$$\nabla^2\Theta + q = 0 \text{ on } \Omega \tag{3.4}$$

$$\Theta(x,y) = 0 \text{ on } \partial\Omega \tag{3.5}$$

Where $D = \Delta t / \Delta x^2$ is termed as the diffusion number. To evaluate the convergence we write:

$$\frac{\partial\Theta}{\partial} = \nabla^2\Theta + q \tag{3.6}$$

Where, ε_j^n is deviation from steady state. The corresponding differential equation which Θ_j^∞ satisfies is:

$$\frac{1}{\Delta x^2}\left(\Theta_{i+1,j}^n + \Theta_{i,j+1}^n - 4\Theta_{i,j}^n + \Theta_{i,j-1}^n + \Theta_{i-1,j}^n\right) + q_{ij} = 0 \tag{3.7}$$

By subtracting equation (3.7) from equation (3.5) we get:

$$\varepsilon_{ij}^{n+1} = \varepsilon_{ij}^n + D\left(\varepsilon_{i+1,j}^n + \varepsilon_{i,j+1}^n - 4\varepsilon_{i,j}^n + \varepsilon_{i,j-1}^n + \varepsilon_{i-1,j}^n\right) \tag{3.8}$$

To imply absolute stability[2] convergence is obtained when $|\varepsilon^n| \to 0$. Thus, we require $D \le 1/4$ and for most efficient possible convergence we take $D = 1/4$. Therefore we obtain the basic Algorithm equation as:

$$\Theta_{ij}^{n+1} = \frac{1}{4}\left(\Theta_{i+1,j}^n + \Theta_{i,j+1}^n - \Theta_{i,j}^n + \Theta_{i,j-1}^n + \Theta_{i-1,j}^n + \Delta x^2 q_{ij}\right) \tag{3.9}$$

The different algorithms which can be developed based on the above formulated equation for real-time application is discussed in the next section.

2.2 Convergence Rate

Considering $\Theta(x,t) = e^{-k\pi^2 t}\sin\pi x$, where '$k$' is the correction diffusion rate over time 't' and $\Theta_n^j = \sum_{k=1}^{N-1} a_k^n \sin\pi kx$ we attain the von-Neumann stability equation as:

$$a_k^{n+1} = a_k^n\left(\frac{1 - D(1-\theta)\sigma_k}{1 + D\theta\sigma_k}\right) \tag{3.10}$$

[1] Euler-forward is a one-sided backward finite difference equation: $\left(\Theta_j^{n+1} - \Theta_j^n\right)/\Delta t = f\left(t, \Theta^n\right)$

[2] Absolute Stability condition $|\varepsilon^{n+1}| \le |\varepsilon^n|$, all the constituents of errors have to be uniformly bounded.

Where a_k^n is the Fourier sine coefficient, $\sigma_k \equiv 2(1-\cos\pi k\Delta x)$ and $D \equiv \frac{k\Delta t}{\Delta x^2}$. The convergence error ε^n therefore satisfies the above equation and hence we get:

$$\varepsilon_{kl}^{n+1} = \varepsilon_{kl}^n \left[\frac{1-\frac{1}{4}(1-\theta)\sigma_{kl}}{1+\frac{\theta}{4}\sigma_{kl}} \right] \quad (3.11)$$

Considering $\sigma_{kl} = 2[(1-\cos k\pi\Delta x)+(1-\cos\pi l\Delta y)]$ and setting $\theta = 0, \delta = 1$ since $\Delta x = \Delta y$ we get:

$$\varepsilon_{kl}^{n+1} = \varepsilon_{kl}^n \left(1-\frac{1}{4}\sigma_{kl}\right) \quad (3.12)$$

We get the minimum estimate for σ_{kl} when $k = l = 1$. Hence the error estimate is obtained as:

$$\varepsilon^{n+1} = \varepsilon^n \left[1-\frac{1}{2}[(1-\cos\pi\Delta x)+(1-\cos\pi\Delta x)]\right]$$

$$\Rightarrow \varepsilon^{n+1} = \varepsilon^n [1-(1-\cos\pi\Delta x)]$$

$$\Rightarrow \varepsilon^{n+1} \approx \varepsilon^n \left(1-\frac{\pi^2\Delta x^2}{2}\right) \quad (3.13)$$

Now we introduce z_j as the convergence rate and we define it for the Jacobi algorithm as:

$$z_j = \frac{\varepsilon^{n+1}}{\varepsilon^n} = \left(1-\frac{\pi^2\Delta x^2}{2}\right) \quad (3.14)$$

Therefore we can represent the convergence error as:

$$\varepsilon^n = z_j^n \varepsilon^0 \quad (3.15)$$

To obtain accuracy defined as $z_j^n = e^{-d}$ we have to compute the following number of iterations:

$$n \ln z_j = -d \Rightarrow -n\frac{\pi^2\Delta x^2}{2} \sim -d \Rightarrow n \sim \frac{2}{\pi^2}\frac{d}{\Delta x^2} \quad (3.16)$$

2.3 Introducing Relaxation for Iterative Solvers

For the Gauss-Seidel method discussed in the next section the convergence can be efficiently improved by establishing a relaxation procedure. It is defined as:

$$\Theta_{ij}^{n+1} = \Theta_{ij}^n + \omega d_{ij}^n \quad (3.17)$$

Where, $d_{ij}^n = \hat{\Theta}_j^{n+1} - \Theta_j^n$. The term $\hat{\Theta}_j^{n+1}$ is computed by equation (4.5). Employing predictor-corrector[3], we get the displacement d_{ij}^n as:

$$d_{ij}^n = \frac{1}{4}\left(\Theta_{i+1,j}^n + \Theta_{i,j+1}^n - 4\Theta_{i,j}^n + \Theta_{i,j-1}^n + \Theta_{i-1,j}^n + \Delta x^2 q_{ij}\right) \quad (3.18)$$

Substituting the value of d_{ij}^n in equation (3.10) we obtain:

$$\Theta_{ij}^{n+1} = \frac{\omega}{4}\left(\Theta_{i+1,j}^n + \Theta_{i,j+1}^n + \Theta_{i,j-1}^{n+1} + \Theta_{i-1,j}^n + \Delta x^2 q_{ij}\right) + (1-\omega)\Theta_{ij}^n \quad (3.19)$$

Assuming that the matrix 'A' defined in the equation (4.1) is positively definite and symmetric the equation (3.12) converges for $0 < \omega < 2$. The various situations that can generate from the above equation are:

For $\omega = 1$, we gat a standard Guass-Seidel equation

For $\omega > 1$, we get successive over-relaxation (SOR)

For $\omega < 1$, we get under-relaxation

From the equation (3.12) we get the optimized value of ω for maximizing convergence as:

$$\omega_{opt} = \frac{2}{\left(1+\sqrt{1-z_j^2}\right)} \text{ And } z_{opt} = \frac{z_j^2}{\left(1+\sqrt{1-z_j^2}\right)^2} \quad (3.20)$$

Here z_j is the convergence rate and for $\Delta x = \Delta y$ the Jacobi convergence rate is $z_j = \cos\pi\Delta x$. Thus we derive:

$$\omega_{opt} = \frac{2}{1+\sin\pi\Delta x} \approx 2(1-\pi\Delta x) \quad (3.21)$$

And correspondingly we get:

$$z_{opt} = \frac{\cos^2\pi\Delta x}{(1+\sin\pi\Delta x)^2} \approx 1-2\pi\Delta x \quad (3.22)$$

Therefore for an accuracy of 'd' correct units and considering $\Delta x = 1/N$, the number of iterations required to achieve convergence is:

$$\omega_{SOR} iteration : n \approx \frac{-d}{\ln z_{opt}} \approx N \quad (3.23)$$

---

[3] Predictor-corrector involves correction of intermediate computed solutions. In this case we apply this method to correct Θ_j^{n+1}.

2.4 Symmetric Successive Over-Relaxation

In successive over-relaxation technique the solver takes a preferential direction and hence is biased. For instance if the solver sweeps the domain from left to right, this may lead to accumulation of error and will generate a complex eigenvalue matrix $'G'$. To nullify this issue we use the symmetric successive over-relaxation technique. Here we allow the solver to sweep the domain in two phases: a) first left to right and then b) right to left. For this, we first generate a standard Guass-Seidel model (defined in the next section) to get the intermediate value of $\hat{x}^{n+1}$ as:

$$(D+\omega L)\check{x}^{n+1} = \left[(1-\omega)D - \omega U\right]x^n + \omega b \tag{3.24}$$

The corresponding iteration for the reverse sweep is:

$$(D+\omega U)\check{x}^{n+1} = \left[(1-\omega)D - \omega L\right]x^n + \omega b \tag{3.25}$$

Considering and approximation proposed by Miellou and Spiteri (2004) we have:

$$\omega_{opt} = \frac{2}{\left(1+\sqrt{2\left[1-\rho(G)\right]}\right)} \tag{3.26}$$

With $\rho(G) = 1-\pi/N$, this leads to double the convergence rate of successive over-relaxation iteration, which further leads to double number of operations. The convergence rate of symmetric successive over-relaxation model becomes:

$$z \approx 1-\pi\Delta x \approx 1-\frac{\pi}{N} \tag{3.27}$$

3. Putting the Iterative Algorithms in Practice

3.1 Jacobi Algorithm

The Jocobi algorithm developed in equation (3.9) can be employed to love linear systems of the form:

$$Ax = b \tag{4.1}$$

Now, we decompose matrix A as:

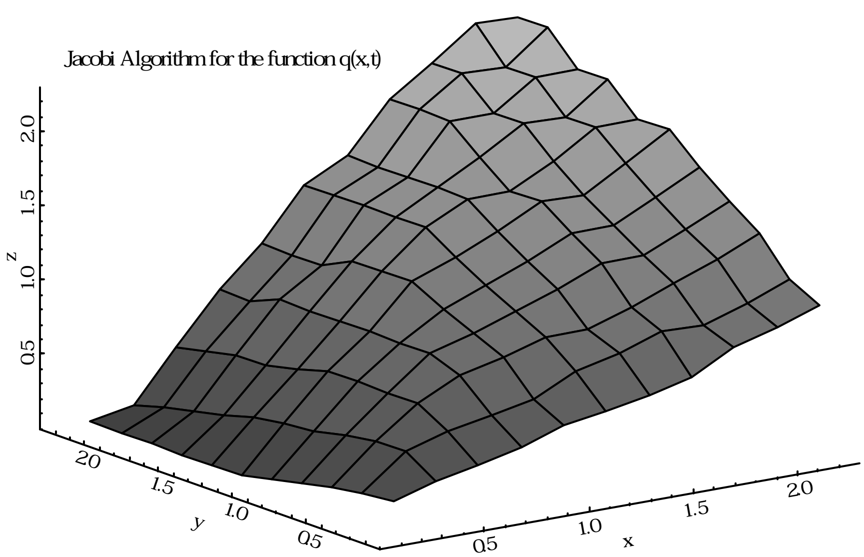
$$A = \underbrace{L}_{lower} + \underbrace{D}_{diagonal} + \underbrace{U}_{upper} \tag{4.2}$$

We can formulate the following iteration procedure by denoting $'n'$ as the iteration number.

$$Dx^{n+1} = b-(L+U)x^n \tag{4.3}$$

In the above Jacobi algorithm, matrix inversion is not required assuming non-zero entities as $D^{-1} = 1/D$, i.e. the diagonal matrix is trivially inverted. The figure below illustrates a Jacobi algorithm for the domain $'\Omega'$.

Figure 3.1 Jacobi Algorithm



3.2 Gauss-Seidel Algorithm

In Gauss-Seidal algorithm we iterate the result by using the most recent computed value. This is done to improve the convergence of the equation. Here we take the matrix version of the Jacobi algorithm and consider the lower matrix. The Gauss-Seidal algorithm can be represented as:

$$(L+D)x^{n+1} = b-Ux^n \tag{4.4}$$

Using this algorithm we obtain:

$$\frac{1}{\Delta x^2}\left(\Theta^{n+1}_{i-1,j}+\Theta^{n+1}_{i,j-1}-4\Theta^{n+1}_{i,j}\right)=-q_{ij}-\frac{1}{\Delta x^2}\left(\Theta^{n}_{i+1,j}+\Theta^{n}_{i,j+1}\right) \quad (4.5)$$

The Gauss-Seidal (GS) algorithm has its edge over the Jacobi (J) algorithm when we compare their respective convergence rate. We have $z_G = z_J^2$.

3.3 Symmetric Successive Over-Relaxation (SSOR): Chebyshev Acceleration

Through SSOR, described in the above section, we can convert a system $Ax = b$ to:

$x^{n+1} = Gx^n + c$. Employing a Chebyshev accelerator we can converge the combination of iterates $\left[x^n\right]$ faster than the last iterate. For this purpose we define a linear combination as:

$$y_n = \sum_{q=0}^{n} \gamma_n^q x^q \quad (4.6)$$

Considering the linear combination the exact solution for 'x' is represented as:

$$y_n - x = \sum_{q=0}^{n} \gamma_n^q \left(x^q - x\right) = \sum \gamma_n^q G^q \left(x^0 - x\right) = p_n(G)(x_0 - x) \quad (4.7)$$

In the above equation the estimation polynomial 'p_n' of degree 'n' is defined with $p_n(1) = 1$, since the interpolation coefficient satisfies $\sum_{q=0}^{n}\gamma_n^q = 1$. To minimize the computation we need to minimize the maximum eigen value for the matrix $p_n(G)$. To define the Chebyshev polynomial that satisfies the above condition we assume $x \in [-\rho, \rho]$, where ρ is the spectral radius of the eigen value matrix.

$$p_n(x) = \frac{T_n(x/\rho)}{T_n(1/\rho)} = \frac{T_n(x/\rho)}{C_n} \quad (4.8)$$

Where, $T_n(x)$ is the nth Chevyshev polynomial and C_n is the recursive normalization constant which can be obtained by:

$$C_n = \frac{2}{\rho}C_{n-1} - C_{n-2} \quad (4.9)$$

For the error term $(y^n - x)$ we have:

$$y^n - x = p_n(G)$$
$$\Rightarrow y^n - x = C_n^{-1}T_n(G/\rho)(x^0 - x)$$
$$\Rightarrow y^n - x = C_n^{-1}\left[2(G/\rho)T_{n-1}(G/\rho)(x^0 - x) - T_{n-2}(G/\rho)(x^0 - x)\right]$$
$$\Rightarrow y^n - x = C_n^{-1}\left[2(G/\rho)C_{n-1}p_{n-1}(G/\rho)(x^0 - x) - C_{n-2}p_{n-2}(G/\rho)(x^0 - x)\right]$$
$$\Rightarrow y^n - x = C_n^{-1}\left[2(G/\rho)C_{n-1}(y^{n-1} - x) - C_{n-2}(y^{n-2} - x)\right]$$

Therefore, the recursive relation can be written as:

$$y^n = 2\frac{C_{n-1}}{C_n}\frac{G}{\rho}y^{n-1} - \frac{C_{n-2}}{C_n}y^{n-2} + D^n \quad (4.10)$$

Where, $D^n = \frac{2}{\rho}\frac{C_{n-1}}{C_n}c$ for the iteration equation $x^{n+1} = Gx^n + c$. Now by setting: $C_0 = 1$ and $C_1 = 1/\rho$, we can initiate the first iteration for $y^1 = Gx^0 + c$.

Conclusion

This paper focuses on portfolio optimization by employing iterative solvers on a two dimensional domain 'Ω'. The paper also provides Jacobi and Gauss-Seidal algorithm for their application in real time. For better convergence of the models, the paper proposes successive over-relaxation of the iterative solvers. Since this relaxation assumes a preferential direction and suffers from domain bias, they lead to accumulation of error.

Further, to overcome this undesirable property we provide algorithm for the symmetric successive algorithm technique: Chebyshev acceleration. To conclude we state and show that iterative solvers act as far superior tools for portfolio optimization in a multidimensional domain and shows better stability and convergence rate.

References

A. Kreinin, Merkoulovitch, L., Rosen, D., and Z. Michael (1998), "Measuring Portfolio Risk Using Quasi Monte Carlo Methods", *ALGO Research Quarterly*, 117-25

A. Lucas, and P. Klaassen (1998), "Extreme Returns, Downside Risk, and Optimal Asset Allocation", *Journal of Portfolio Management*, 25,71-79.

D. Duffie and J. Pan(1997), "An Overview of Value-at-Risk", *Journal of Derivatives*, 4, 7-49.

F. Stambaugh (1996), "Risk and Value-at-Risk", *European Management Journal*, 14, 612-621.

H. Konno and H. Yamazaki (1991), "Mean Absolute Deviation Portfolio Optimization Model and Its Application to Tokyo Stock Market", *Management Science*, 37, 519-531.

H.M. Markowitz (1952), "Portfolio Selection", *Journal of Finance*, 7, 77-91.

Mausser, H. and D. Rosen (1991), "Beyond VaR: From Measuring Risk to Managing Risk", *ALGO Research Quarterly,* 5-20.

K. Simons (1996), "Value-at-Risk New Approaches to Risk Management", *New England Economic Review*, 3-13.

M. Pritsker (1997), "Evaluating Value at Risk Methodologies", *Journal of Financial Services Research*, , 201-242.

M.R. Young (1998), "A Minimax Portfolio Selection Rule with Linear Programming Solution", *Management Science*, 445, 673-683.

N. Bucay and D. Rosen (1999), "Credit Risk of an International Bond Portfolio: a Case Study", *ALGO Research Quarterly*, 29-29.

P. Artzner, Delbaen F., Eber, J.M., and D. Heath (1997), "Thinking Coherently", *Risk*, 10, 68-71.

P. Embrechts (1999), "Extreme Value Theory as a Risk Management Tool", *North American Actuarial Journal*, 3.

P. Artzner, Delbaen F., Eber, J.M., and D. Heath (1999), "Coherent Measures of Risk", *Mathematical Finance*, 9, 203-228.

Dembo, R.S. and D. Rosen (1999), "The Practice of Portfolio Replication: A Practical Overview of Forward and Inverse Problems", *Annals of Operations Research*, 85, 267-284.

R.T. Rockafellar and S. Uryasev (2000), "Optimization of Conditional Value-at-Risk", *The Journal of Risk*,

R. Litterman (1997), "Hot Spots and Hedges", *Risk*, 10, 42-45.

S. Uryasev (2000), "Conditional Value-at-Risk", *Optimization Algorithms and Applications Financial Engineering News*, 14.

T. Stublo Beder (1995), "VAR: Seductive but Dangerous", *Financial Analysts Journal*, 12-24.

W.T. Ziembaand and J.M. Mulvey (1998), "Worldwide Asset and Liability Modeling", *Cambridge University Press*.