Realized Risk Reduction in Portfolio Optimization through Iterative Solutions of the MVDR Filter

Antoine Bosselut
260365970

Yannis Psaromiligkos

Abstract:
The goal of this project is to gauge the effectiveness of using signal processing techniques for portfolio optimization problems. The equation for computing the weight vector of the minimum variance portfolio of a group of stocks is mathematically identical to the minimum output variance of the MVDR filter. By applying an iterative algorithm for the computation of the minimum output variance of the MVDR filter to historical stock return data, we hope to reduce the effect of the bias inherent in historical data, and achieve lower portfolio variances. We will also expand the scope of our research to other iterative noise-reducing algorithms that rely on auxiliary vector filtering, specifically the block matrix algorithm, in the hopes that they will yield similar results that allow us to reduce the realized risks of our portfolios. The motivation for this experiment is to investigate whether the mathematics that is the foundation of common engineering problems can be applied to problems in different fields to achieve optimal solutions. In this case, I will be testing this hypothesis with respect to financial portfolios and investigating whether these signal-processing techniques allow us to limit the bias of historical sample data when formulating future portfolios, thereby achieving less risky portfolios. In terms of deliverables, this project will lead to a practical algorithm that achieves an optimal portfolio through signal processing techniques.

Acknowledgements:
I would like to thank Professor Yannis Psaromiligkos for his help and unyielding support throughout the development of my thesis. His commitment to helping me formulate the direction of my research, suggesting potential papers of interest, and following up on papers that I suggested to him were essential factors in the success of this project.

I would also like to thank Professor Matthieu Bouvard from the faculty of management for helping me gain access to certain papers relevant to my research.

Finally, I would like to thank Professor Roni Khazaka for helping me structure my thesis.

Table of Contents:

List of Tables:

List of Figures:

List of Abbreviations:

MVDR – Minimum Variance Distortionless Response
AV – Auxiliary vector
MVP – Minimum Variance Portfolio

Introduction, Motivation and Objective:

The risk of a portfolio is defined as the variance of the potential returns of that portfolio. Because returns are uncertain, they are usually modeled as continuous random variables. The risk of a stock is the variance of that return. The risk of a portfolio can be calculated by multiplying a matrix filled with the covariances of returns between the stocks in the portfolio with vectors that contain the ratio of overall investment put into a particular stock in the portfolio (known as the weight of each stock in the portfolio). In order to predict the future risk of a portfolio, we are forced to rely on either economic estimates of stock return variance or rely on historical pricing data, both of which expose risk predictions to estimation bias, thereby increasing the probability that the realized risk of the portfolio will be much higher than the risk we have predicted. By using different techniques to calculate these weights, we can limit the effect of estimation bias and reduce the realized risk of our portfolio.

In this project, we will be looking at new ways in which we can compile low-risk portfolios using new techniques for calculating the investment weights in the Markowitz portfolio problem. Our strategy is to examine whether iterative approaches for computing the final solution of the MVDR filter used to reduce noise in beamforming are relevant when applied in a financial portfolio context. By using filters normally utilized in signal processing applications to reduce signal interference, we will calculate portfolio weights that, while potentially increasing the biased predicted risks of our portfolio, will diminish the realized risks of our portfolio.

This project is important because we are investigating strategies by which investors can reduce the bias of historical sample data when computing the risk of their portfolios. While stock-specific (or idiosyncratic) risks can be reduced to ignorable amounts by diversifying a portfolio across a wide range of stocks, systematic risk is always present in a portfolio, even one perfectly spread across the entire market (i.e. the market portfolio). By using auxiliary vector filtering techniques we can reduce the realized systematic risk of a portfolio, thereby identifying a safer investment portfolio.

This is particularly important to pension funds and mutual funds – large passive funds designed to achieve continuous mid-level returns over a long period of time. Passive funds buy and hold stocks for long periods of time. They rely on stock price returns due to economic growth to fuel their returns. Active funds, conversely, rely more on taking advantage of shorter-term stock price movements to achieve their returns. People who invest in passive funds often are investing for their future financial security. They are investors with an appetite for low-risk, stable returns that do not fluctuate and that over time will yield large savings. As such, they are often less focused on the absolute return of stocks, as much as how these stocks will move in relation to one another. With this research, we hope to extend our knowledge of signal processing filters to the Markowitz portfolio problem in order to see whether we can safely reduce the risk of these types of portfolios.

This thesis is structured as follows. The background section will give an overview of portfolio theory, specifically with regards to the mathematics behind the Markowitz Portfolio problem, as well as describing some of the limitations of the Markowitz portfolio, and various solutions that have been developed to counter these limitations. The Design and Implementation section will give an overview of the two iterative algorithms for computing the MVDR filter that we are applying to the Markowitz portfolio problem, as well as how our data is prepared before our simulations. The Results section will describe our sampling methodology for back-testing historical data, as well as expose the results of our simulations. The Impact on Society and the Environment Section will depict how our research and results could affect society and any potential environmental implications they may have. Finally, the conclusion section will summarize our results, and provide ideas for future research directions within this field.

Background:

(i) Stock Returns

In order to calculate the return on an investment in a stock over any month, day or year, one must calculate the increase in price of that stock over the course of the month. This can be done using the price of that stock at the beginning of the time range we are examining and the price at the end of the time period and applying the formula:

$$E(r_n) = \frac{P(t_o+T) - P(t_o)}{P(t_o)} \tag{1}$$

where $P(t_o + T)$ and $P(t_o)$ correspond to the price of the stock at the end and beginning of the time period whose return we are calculating, respectively [6]. In this research, we are far more interested in how stock prices move, grow and vary as opposed to their absolute values and thus must calculate the returns on different stocks.

(ii) The Markowitz Portfolio Problem

Using the returns of these stocks, we can attempt to solve the Markowitz Portfolio Optimization problem. This optimization problem attempts to address the fundamental issue of minimizing the overall risk of a portfolio for a certain return. This is done by picking the *N×1* weight vector, $\boldsymbol{w}$ (the weights correspond to the percentage of the total investment assigned to each stock in the portfolio), of the portfolio in order to achieve:

$$\arg\min \ \sigma_p^2 = \boldsymbol{w}^T C \boldsymbol{w} \tag{2}$$

subject to:

$$\mathrm{r}_p = \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{\mu} \tag{3}$$

$$\boldsymbol{w}^{\boldsymbol{T}} \mathbf{1} = 1 \tag{4}$$

where C is a *N×N* matrix of the pairwise covariances between the stocks in the portfolio, σ_p^2 is the variance of the portfolio, $\mathbf{1}$ is a *N×1* vector of 1's, μ is a *N×1* vector of individual stock returns and r_p is the target return of the portfolio [6].

Due to multiple market factors, it can often be very difficult to estimate future returns for stocks. Macroeconomic conditions can often have a great influence on how stocks perform. For example, an economic recession will lower the returns of most stocks. What is more constant, however, is how stocks perform in relation to one another. Two stocks that perform similarly in a recession should perform similarly in a growing economy. We can therefore make the assumption that stocks will perform similarly in relation to one another in the near future as they have in the recent past. It is thus easier to try to build portfolios with the goal of minimizing overall risk than of maximizing return. As a result, most research on the subject tends to focus on the former goal.

In order to find the minimum variance portfolio (also known as the Markowitz portfolio) of a portfolio of stocks, one must try to minimize (2) subject to only (4). This optimization problem yields the solution:

$$\boldsymbol{w} = \frac{C^{-1}\mathbf{1}}{\mathbf{1}^{\boldsymbol{T}} C^{-1} \mathbf{1}} \tag{5}$$

as long as the covariance matrix is invertible [6].

By either keeping covariances between any two stocks low, or balancing them with negatively correlated stocks, we can reduce the risk of the portfolio. Unfortunately, negatively correlated stocks are usually overpriced as most investors are willing to pay more to have stocks in their portfolio that reduce that portfolio's overall risk. Additionally, holding stocks whose returns vary negatively with other stocks will induce large positive weights in these negative covariance stocks when optimizing the portfolio. This goes against the principle of diversification defined before and will re-introduce idiosyncratic risk to the portfolio. Because our model does not take idiosyncratic risk into account, we must diversify our portfolio sufficiently so that it can be ignored. Shorting stocks also can help us reduce the overall risk of the portfolio by holding a negative weight of a particular stock. In practice, shorting stocks involves borrowing a stock from its owner, selling it, and then buying the stock back at a later date. Mathematically, because we have sold a stock we do not own, we can consider shorting a stock to be a negative investment, and we assign a negative weight to this stock in the portfolio. This approach is limited by multiple legal factors, however. It is an option for active-traded funds (e.g., hedge funds and industry-specific funds) whose goals are to maximize the returns of their clients by taking advantage of short-term price movements, but most mutual funds and pension funds are legally not allowed to short stocks when forming their portfolios. Furthermore, shorting stocks as a mechanism to reduce portfolio weights tends to induce large negative weights for certain stocks with high covariances with multiple other stocks [2]. Similarly to the consequences of having negatively covariant stocks in the portfolio, this practice goes against the idea of well diversifying one's portfolio and reintroduces idiosyncratic risk into the portfolio. In our models, we will conduct simulations for case where stocks are allowed to be shorted.

(iii) The Capital Asset Pricing Model

As can be expected, most stocks do not have negative covariances with one another. While certain industries can thrive as others falter, the general view is that most stock prices move along similar lines as the overall market, albeit at different scales. This is the foundation of the Capital Asset Pricing Model (a single-factor model, where all stock returns are dependent on only one factor: the market return), which states that the expected value of a stock's (or portfolio's) return should be of the function:

$$E(r_s) = r_f + \beta(r_M + r_f) \tag{6}$$

where $E(r_s)$ corresponds to the expected return of a particular stock (or portfolio), r_M corresponds to the return of the market portfolio, r_f corresponds to the risk-free rate (usually the interest rate on a 90-day treasury bill as this is the financial asset that is considered to have the lowest risk) and β is a measure of the volatility of that particular stock relative to the market portfolio (also known as the systematic risk of a stock) [6]. Stocks with high β are more exposed to variability in the overall market's return and have a higher systematic risk. β can be calculated with the equation:

$$\beta = \frac{cov(r_M, r_s)}{\sigma_M^2} \tag{7}$$

where r_s is the return of the stock (or portfolio) whose β is being calculated, and σ_M^2 is the variance of returns of the market [6]. The CAPM does not consider idiosyncratic risk in its formula. It assumes that investors should not be rewarded for risk that can be eliminated through diversification [6].

(iv) Covariance Matrices

In order to avoid introducing idiosyncratic risk to our model, the best option for reducing overall portfolio risk is to keep any covariance between two particular stocks as low as possible, or to keep the weight of stocks that have high covariances with multiple other stocks as low as possible. Even though these covariances can be extracted from historical stock return data, they can be manipulated if we accept that the historical sample is inherently biased. Covariance matrices are populated by the pair-wise covariances of every stock in the portfolio (for a well-diversified portfolio this must theoretically be at least a minimum 30 stocks, but most portfolios distribute weights to far more stocks), and thus the covariance matrix of a portfolio can have anywhere from $\sim 10^3$ to $\sim 10^7$ unique entries. Seeing as portfolios are calculated using monthly or daily historical data (a shorter sampling rate exposes pricing data to microstructure effects), there is a limited historical data set from which we can sample. Additionally, if we assume only historical data from a recent time period should be sampled, this historical data set becomes even smaller. If we go back too far in history to increase the size of sample data set, we can no longer make the assumption that stocks will behave the same way in relation to each other. Because the number of indices in the covariance matrix is much larger than the number of data points available to us, the sample covariance matrix is necessarily exposed to estimation bias. Additionally, there is another major flaw with using only the sample covariance matrix of historical prices as an estimator for the covariance matrix. When the number of historical data points, T, which we sample from, is less than the number of stocks in the portfolio, N, the sample covariance matrix will have null eigenvalues. This is because we are constructing approximately N^2 indices from only NT data points. If $T<N$, this will mean we are extrapolating a larger number of data points than the number of data points we have available to make these predictions. Mathematically, this factor will make the covariance matrix non-invertible, and we will not be able to calculate the minimum variance portfolio using the formula outlined above. For shorter data sets that form singular sample covariance matrices, we can use the Moore-Penrose pseudoinverse to estimate the Markowitz portfolio, but these portfolios tend to be riskier and less diversified than the other portfolios calculated using longer sample data time windows [4].

(v) Existing Techniques for reducing the bias of the Sample Covariance Matrix

In order to counter the flaws of the sample covariance matrix as an estimator for the covariance matrix, multiple techniques to reduce the effect of estimation error on covariance matrices have been developed to lower portfolio risk. Although these techniques potentially introduce new types of error to our model, they reduce the influence of the sample covariance matrix on our final covariance matrix and therefore reduce the impact of estimation error on our model. Shrinkage estimators, spectral estimators, clustering estimators, multi-factor models, portfolio norm constraints and market similarity weightings are such techniques.

Multi-factor models involve defining the covariance matrix in terms of a stock's exposure to different market factors. These factors include the market return, the return of different industries, as well as the return of certain classifications of companies [13]. This allows the number of covariance matrix indices to be a function of the number of stocks in the portfolio and the number of factors considered. For a large portfolio, this number will be much less than the number of entries in the sample covariance matrix. A multi-factor can be modeled by the following two equations:

$$\boldsymbol{r} = \boldsymbol{\alpha} + \boldsymbol{F}\boldsymbol{r}_{\boldsymbol{F}} + \boldsymbol{e} \tag{8}$$

$$\boldsymbol{C} = \boldsymbol{F}\boldsymbol{C}_{\boldsymbol{f}}\boldsymbol{F}^{\boldsymbol{T}} + \boldsymbol{C}_{\boldsymbol{s}} \tag{9}$$

where **r** is a $N \times 1$ (where N is the number of stocks in the portfolio) vector of expected returns for the stocks in the portfolio, **α** is a $N \times 1$ vector of stock-specific returns, **F** is a $N \times M$ (where M is the number of factors considered in the model) matrix of each stocks exposure to the factor returns, $\mathbf{r_F}$ is a $M \times 1$ vector of factor-specific returns and **e** is an error vector of length N. **C** represents the $N \times N$ covariance matrix of the returns, $\mathbf{C_f}$ is the $M \times M$ covariance matrix of the factor returns and $\mathbf{C_s}$ is the $N \times N$ covariance matrix of stock risks (a diagonal matrix where each entry is the risk of the particular stock in the portfolio) [1]. As can be seen in the equations above, this is a function where the majority of the data points normally prone to error with the sample covariance matrix are now derived from market factors. This model reduces estimation error because most multi-factor models consider far fewer factors, M, than stocks in the portfolio, N [8]. We will use a single-factor model based on the CAPM as a comparison for our results.

There are numerous estimators that allow us to modify the sample covariance matrix, as well. Spectral estimators involve manipulating the eigenvalues of the sample covariance matrix. The fundamental idea behind these estimators is that the eigenvalues of the covariance matrix carry different economic information depending on their value [4]. Eliminating or adjusting groups of certain eigenvalues can reduce the exposure of the portfolio to estimation error. We will use two spectral estimators as a comparison for our results. Their application will be described in a later section.

Shrinkage estimators involve weighing the influence of the sample covariance matrix and another covariance matrix estimation model, in order to reduce the effect of the estimation error from the sample covariance matrix when calculating our predicted covariance matrix [17]. The relationship, outlined in [10], can be expressed with the equation:

$$\boldsymbol{C} = \alpha\boldsymbol{T} + (1-\alpha)\boldsymbol{S}^{(\boldsymbol{s})} \tag{10}$$

where $\boldsymbol{S}^{(S)}$ is the sample covariance matrix (the covariance matrix formulated from past historical data), $\boldsymbol{T}$ is a target covariance matrix (in our case, this will usually be a multi-factor model), $\boldsymbol{C}$ is a more robust covariance matrix estimator and α is the shrinkage intensity, a multiplier whose range is [0,1], that allows us to balance the influences of the target matrix and sample covariance matrix on the final covariance matrix.

Clustering estimators involve invoking procedures that iteratively merge pair groupings of stocks into clusters of increasing size based on their degree of similarity in order to simplify the values of the covariance matrix [16]. While many variables can be used as the measure of similarity between data points, for our purpose, correlation between stock returns is usually used as the similarity basis [4].

Market similarity weighing is a technique that involves weighing each value of the sample covariance matrix by how similar the current correlations of any two stocks are to a specific period in the past. This process, outlined in [11] can be represented by the equation:

$$\zeta^L(t_1, t_2) = ||C^L(t_1) - C^L(t_2)||_2 \tag{11}$$

where $\zeta^L(t_1, t_2)$ represents the weighing of market similarity between times t_1 and t_2, $\boldsymbol{C^L}$ are matrices outlining the correlations between stocks at time t_n, and $|| \,\boldsymbol{.}\, ||_2$ represents the Frobenius norm of the difference between these correlation matrices [11]. By normalizing these market similarity weightings we can find the adaptive similarity weightings which will adjust the values in covariance matrix based on stock return similarity to a past time window.

Portfolio Norm Constraints involve constraining a norm of the weight vector so that we restrict the size of the investment in the stocks that make up the portfolio. Although we do not constrain the weight of any particular stock in the portfolio, constraining the norm of the weight vector will theoretically yield more evenly distributed portfolios. This ascertains that idiosyncratic risk is not introduced to the portfolio and that any particular stock whose performance in relation to other stocks differs from as it has in the past will not adversely affect the realized risk of the portfolio. Multiple norms such as the 1-norm, 2-norm (also known as the Frobenius norm) or other custom norms can be used to set this constraint [5].

Other solutions such as stochastic models [3], power mapping [7], penalized normal likelihoods [9], and covariance thresholding [12] can also be used as estimators the covariance matrix, but they will not be discussed in this paper.

(vi) Iterative Algorithms for the Computation of the MVDR Filter

While it would not be expected that signal processing filters could be easily applied to a portfolio, an analysis of past literature [18]-[21] clearly shows that the mathematics behind the output of many interference-reducing beamformers is analogous the optimization problem outlined in (4). While these filters often impose more conditions on the optimization problems due to their application setting (e.g. most signal filters must process data measured in both time and space, compared to financial stock price data which is only a function of time), we can apply these filters to portfolio problems by simplifying their application. For the purpose of our research, we chose to apply the MVDR AV filter outlined in [19] and the Block Matrix AV filter, one of the algorithms for AV filtering defined in [18].

Minimum variance distortionless response (MVDR) filtering refers to the problem of identifying a linear filter that minimizes the variance at its output, while at the same time maintaining a distortionless response toward a specific input vector direction of interest [19]. In simpler terms, this translates to reducing the variance of the output signal by balancing the influence of multiple input signals without changing them. Mathematically, this process can be described by:

$$\arg\min \ \sigma_O^2 = \boldsymbol{w}^H \boldsymbol{R} \boldsymbol{w} \tag{12}$$

subject to:

$$\boldsymbol{w}^H \mathbf{1} = \rho \tag{13}$$

where σ_O^2 refers to the variance of the output signal, $\boldsymbol{w}^H$ refers to the conjugate transpose of the weight vector (since we are dealing with only real numbers in a portfolio context, however, this is the same as the real transpose), $\boldsymbol{R}$ is the input covariance matrix and ρ is the desired response of the filter. In a portfolio optimization context, ρ is equal to 1, as the weights of our stocks must add up to 1. This optimization problem yields the recognizable solution:

$$\boldsymbol{w}_{MVDR} = \frac{\boldsymbol{R}^{-1}\mathbf{1}}{\mathbf{1}^T \boldsymbol{R}^{-1}\mathbf{1}} \tag{14}$$

where $\boldsymbol{R}^{-1}$ is the inverse of the input covariance matrix. By applying the MVDR filter to portfolio data, we would find that the minimum variance portfolio is the output signal of the filter. In [19], however, the authors developed an iterative auxiliary vector approach to calculating the minimum-variance output signal. This process converges to the MVDR filter solution in (14), but uses a conditional statistical optimization procedure. Beginning with the output vector:

$$\boldsymbol{w}_{\boldsymbol{o}} = \frac{\rho^*}{\|\mathbf{1}\|^2} \tag{15}$$

which is the solution to the MVDR filter when the input is white noise (in a portfolio context, this corresponds to the $1/N$ portfolio, where each stock has an equal weight in the final portfolio), the authors developed an iterative algorithm that converges to the MVDR solution [19]. Continuing with:

$$\boldsymbol{w}_1 = \boldsymbol{w}_0 - \mu_1 \boldsymbol{g}_1 \tag{16}$$

where $\boldsymbol{w}_1$ is the second output vector calculated, the authors calculated the value of μ_1 that minimizes the variance at the output of $\boldsymbol{w}_1$ and the value of auxiliary vector $\boldsymbol{g}_1$ that maximizes the cross-correlation between $\boldsymbol{w}_0^H \boldsymbol{r}$ and $\boldsymbol{g}_1^H \boldsymbol{r}$ subject to:

$$\boldsymbol{g}^H \mathbf{1} = 0 \text{ and } \boldsymbol{g}^H \boldsymbol{g} = 1 \tag{17}$$

where $\boldsymbol{r}$ is the input vector of interest [19]. This optimization yields specifically:

$$\boldsymbol{g}_1 = \frac{\boldsymbol{R}\boldsymbol{w}_0 - \frac{\boldsymbol{1}^H\boldsymbol{R}\boldsymbol{w}_0}{\|\boldsymbol{1}\|^2}\boldsymbol{1}}{\left\|\boldsymbol{R}\boldsymbol{w}_0 - \frac{\boldsymbol{1}^H\boldsymbol{R}\boldsymbol{w}_0}{\|\boldsymbol{1}\|^2}\boldsymbol{1}\right\|} \tag{18}$$

$$\mu_{\boldsymbol{1}} = \frac{\boldsymbol{g}_1^H\boldsymbol{R}\boldsymbol{g}_1}{\boldsymbol{g}_1^H\boldsymbol{R}\boldsymbol{w}_0} \tag{19}$$

for the first auxiliary vector case, and more generally:

$$\boldsymbol{g}_{k+1} = \frac{\boldsymbol{R}\boldsymbol{w}_k - \frac{\boldsymbol{1}^H\boldsymbol{R}\boldsymbol{w}_k}{\|\boldsymbol{1}\|^2}\boldsymbol{1}}{\left\|\boldsymbol{R}\boldsymbol{w}_k - \frac{\boldsymbol{1}^H\boldsymbol{R}\boldsymbol{w}_k}{\|\boldsymbol{1}\|^2}\boldsymbol{1}\right\|} \tag{20}$$

$$\mu_{\boldsymbol{k}} = \frac{\boldsymbol{g}_{k+1}^H\boldsymbol{R}\boldsymbol{g}_{k+1}}{\boldsymbol{g}_{k+1}^H\boldsymbol{R}\boldsymbol{w}_k} \tag{21}$$

for each successive output vector calculated by the equation:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \mu_{k+1}\boldsymbol{g}_{k+1} \tag{22}$$

For the purpose of our simulation, we calculated AVs out to $K = 150$. By this point, the algorithm had usually converged the MVDR filter (or was close to doing so). The strength of this auxiliary vector algorithm lies in the fact that the input covariance matrix, $\boldsymbol{R}$, needed to calculate the output variance is unknown and must be estimated from historical data samples, $\widehat{\boldsymbol{R}}$. The authors observed that some output vectors, $\widehat{\boldsymbol{w}}_{\boldsymbol{k}}$, that occurred before the algorithm converged to the MVDR filter (and that were calculated using $\widehat{\boldsymbol{R}}$) yielded output variances closer to that which was predicted by the MVDR filter when they were applied to the actual input covariance matrix, $\boldsymbol{R}$ [19]. By applying this algorithm to stock price return data, we hope to witness the same result: weights vectors that yield realized portfolio risks closer to the sample predicted minimum variance portfolio. In order to assess the impact of weights calculated using different numbers of AVs, we grouped all weight vectors, $\boldsymbol{w}_k$, into a matrix $\boldsymbol{W}_M = [\boldsymbol{w}_1 \quad \boldsymbol{w}_2 \quad \dots \ \boldsymbol{w}_k]$ to be able to compute the realized risk for each weight vector. This would allow us to pinpoint the optimal number of AVs that were needed to achieve the lowest realized risk.

The block matrix AV algorithm is very similar to the iterative MVDR algorithm. It also converges to the MVDR filter. The difference between it and the algorithm outlined above is that it recalculates the factors μ that minimize the output variance of the filter once every auxiliary vector, $\boldsymbol{g}_i$, has been calculated. The behavior of the filter (outlined in [18]) can be approximated by the equation:

$$\boldsymbol{w}_{\boldsymbol{B}} = \boldsymbol{1}_{\boldsymbol{0}} - \boldsymbol{B}\boldsymbol{\mu} \tag{23}$$

where $\mathbf{1_0}$ is a normalized vector of 1's, $\boldsymbol{\mu}$ is a vector of μ_k factors that minimize the output variance, $\boldsymbol{B}$ is a blocking matrix composed of the auxiliary vectors, $\boldsymbol{g_i}$, that have been orthonormalized according the Gram-Schmidt process. Each auxiliary vector, $\boldsymbol{g_k}$, is calculated using the formula [18]:

$$\boldsymbol{g}_{k+1} = \frac{\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)-\left[\mathbf{1}_0^H\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)\right]\mathbf{1}_0-\sum_{j=1}^{k}\left[\boldsymbol{g_j}^H\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)\right]\boldsymbol{g_j}}{\left\|\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)-\left[\mathbf{1}_0^H\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)\right]\mathbf{1}_0-\sum_{j=1}^{k}\left[\boldsymbol{g_j}^H\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)\right]\boldsymbol{g_j}\right\|} \quad (24)$$

where $\boldsymbol{R}$ is the input covariance matrix and c_i is a series of coefficients that minimize the output variance for the weight vector computed with their corresponding auxiliary vectors. These coefficients are calculated using [18]:

$$c_{k+1} = \frac{\boldsymbol{g}_{\boldsymbol{k+1}}^H\boldsymbol{R}\left(\mathbf{1}_0-\sum_{i=1}^{k} c_i\boldsymbol{g_i}\right)}{\boldsymbol{g}_{k+1}^H\boldsymbol{R}\boldsymbol{g}_{\boldsymbol{k+1}}} \quad (25)$$

Once K auxiliary vectors had been calculated, which in our simulations was decided to be K = 20 (after 25 or so iterations, the algorithm begins to have stability issues), the auxiliary vectors, $\boldsymbol{g_k}$, were grouped sequentially in the blocking matrix $\boldsymbol{B}$. In order to test blocking matrices of different lengths, 20 blocking matrices were calculated with corresponding k auxiliary vectors according to:

$$\boldsymbol{B} = [\boldsymbol{g_1} \quad \boldsymbol{g_2} \quad \boldsymbol{g_3} \quad \ldots \quad \boldsymbol{g_k}] \quad (26)$$

where k = 1,…, 20. From there, the vector $\boldsymbol{\mu}$ for each $\boldsymbol{B}$ was calculated using [18]:

$$\boldsymbol{\mu} = [\boldsymbol{B}^{\boldsymbol{H}}\boldsymbol{R}\boldsymbol{B}]^{-1}\boldsymbol{B}^H\boldsymbol{R}\mathbf{1_0} \quad (27)$$

and (23) was applied to yield the weight vector of choice. Because we used 20 blocking matrices of lengths k = 1,…, 20 auxiliary vectors, our solution to this algorithm was a matrix $\boldsymbol{W_B}$, with 20 different weight vectors. The block matrix AV approach also converges to the MVDR filter (until it becomes unstable) and as such, similar properties to the MVDR iterative AV algorithm can be observed with regards to output variances from sample data sets.

Design and Implementation:

(i) Data Organization

In order to implement this research, we will be using historical monthly time series stock price data taken from the Wharton Research Data Services [23]. This data will then be organized by stock ID and into a large matrix that lists the price of every stock at a certain day in time from the range t_o to $t_o + T$ in every row as depicted below:

$$\begin{bmatrix} Price\ of\ Stock\ i_1\ at\ Date\ t_o & \cdots & Price\ of\ Stock\ i_N at\ Date\ t_o \\ \vdots & \ddots & \vdots \\ Price\ of\ Stock\ i_1\ at\ Date\ t_o + T & \cdots & Price\ of\ Stock\ i_N at\ Date\ t_o + T \end{bmatrix}$$

Figure 1: Data Organization Matrix

Once the data has been organized into this matrix, price data for different time windows can be extracted and converted into return data using (1).

Using the returns over time of a large subset of stocks, it is possible to calculate the average return of a stock over any time window desired (called the sample window). And, more importantly, it is possible to calculate the covariance of stock returns between any two stocks over this same time window. These pair-wise stock covariances can then be organized into an $N \times N$ matrix (N representing the number of stocks in the sample set):

$$\begin{bmatrix} Variance\ of\ stock\ \ i_1 & Cov\ of stocks\ i_1\ and\ \ i_2 & \cdots & Cov\ of stocks\ i_1\ and\ \ i_N \\ Covariance\ of stocks\ i_1\ and\ \ i_2 & Var\ of\ stock\ \ i_2 & \cdots & Cov\ of stocks\ i_2\ and\ \ i_N \\ \vdots & \vdots & \ddots & \vdots \\ Covariance\ of stocks\ i_1\ and\ \ i_N & Cov\ of stocks\ i_2\ and\ \ i_N & \cdots & Var\ of\ stock\ \ i_N \end{bmatrix}$$

Figure 2: Sample Covariance matrix

Using this covariance matrix, it is now possible to calculate the Markowitz portfolio, and apply the different Auxiliary Vector filters (Iterative MVDR and Block Matrix) to see if we can construct portfolios with lower realized risks.

(ii) Cross-Validation

As outlined in the background section, the iterative MVDR and Block Matrix algorithms return solutions of K = 150 or K = 20 different weight vectors, respectively, for the sample window we are looking at. This leaves us with the task of finding which of these 150 or 20 weight vectors (depending on the algorithm) yields the best realized risk over the prediction window. In order to do this, we use cross-validation. Cross-validation involves partitioning the data set (in our case, daily stock return data) into multiple subsets, which are all re-grouped into two different sets – the training set and the validation set [22].

Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Validation Set	Training Set			

Figure 3: Partitioning the Cross-Validation Data Set – First Step

Once we have partitioned the data set into these two subsets, we re-apply the Iterative MVDR and Block Matrix algorithms to the training set, which yielded weight matrices, $\boldsymbol{W_M}$ and $\boldsymbol{W_B}$, of K = 150 and K = 20 weight vectors, respectively. We then individually substitute each of these weight vectors (in both weight matrices) into (2) with the covariance matrix of the validation set and see which weight vector (identified by its index, k) for both techniques yielded the lowest output portfolio variance. This gives us the indices, k, corresponding to a weight vector in $\boldsymbol{W_M}$ and $\boldsymbol{W_B}$ that yields the lowest output variance for a particular training set and validation set.

` We repeat this process by shifting the validation set across every subset in our data set:

Subset 1	Subset 2	Subset 3	Subset 4	Subset 5
Training Set	Validation Set	Training Set		

Figure 4: Partitioning the Cross-Validation Data Set – Second Step

and then recomputing the new indices, k, that yield the lowest output variance for both techniques. Once every subset in our data set has served as the validation set, we average the values of k from each validation step to compute indices for the predicted optimal weight vectors in both $\boldsymbol{W_M}$ and $\boldsymbol{W_B}$. This allows us to predict the index in both algorithms whose corresponding weight vector will yield the least risky portfolio. If the average value of k was not an integer, we rounded to the nearest whole number.

While cross-validation is a straightforward process, it is necessary to define the size of the validation set in comparison to the training set. Different sizes for these sets will yield different indices, k, for the optimal weight vector in both weight matrices. This decision on how big to make the validation set invokes an interesting problem of its own. A smaller validation set allows us to keep our training set as close to our original sample set as possible. Conversely, using a validation set that is too small will not closely reflect the variability and price movements of our stocks over time, and will yield a validation covariance matrix that is not indicative of the covariance matrix of the data set we are trying to predict. An analysis of the ideal validation set size will be performed in the next section.

Results and Tests:

(i) Simulation Parameters

For the purpose of testing our hypotheses, we have chosen $N = 84$ stocks from the *NYSE 100* index that were part of this index between January 1st, 1997 to December 31st, 2007. The ticker symbols for these stocks are: ABT, AIG, ALL, APA, AXP, BA, BAC, BAX, BEN, BK, BMY, BNI, BUD, C, CAT, CCL, CL, COP, CVS, CVX, D, DD, DE, DIS, DOW, EMC, EMR, EXC, FDX, FNM, GD, GE, GLW, HAL, HD, HIG, HON, HPQ, IBM, ITW, JNJ, JPM, KMB, KO, LEH, LLY, LMT, LOW, MCD, MDT, MER, MMM, MO, MOT, MRK, MRO, MS, OXY, PCU, PEP, PFE, PG, RIG, S, SGP, SLB, SNS, SO, T, TGT, TRV, TWX, TXN, UNH, UNP, USB, UTX, VZ, WAG, WB, WFC, WMT, WYE, and XOM. These are the ticker symbols these stocks had at the end of the 11-year period. Some of these stocks changed ticker symbols mid-way through the period due to acquisitions, name changes, or other business-specific factors. Their stock prices, however, are continuous despite the name changes as we identify stocks by their PERMCO, a unique permanent company identification number, assigned by the CRSP, the Center for Research in Security Prices.

This sample set gives us exactly 2767 stock price data points for each of the 84 stocks. By calculating the daily returns of these stocks using (1), we have 2766 stock return data points for each stock. Using a similar methodology as in [4], we calculate the predicted risk and realized risk for various portfolios using AV filters of different lengths. First, we set a time window, T, to be either of length of 60, 90, 120, 180, or 240 trading days. Starting at time, t_o, we sample the time data for the length of our desired window for all 84 stocks and then construct the sample covariance matrix for the data in this window (we will refer to this window as the sample window, and its length as the sample length). Using the techniques and filters outlined in previous sections, we construct multiple competing portfolios, and then assess their performance over the next time windows of 30, 60, 90, 120, 180 and 240 days (this test window will be known as the prediction window, and its length as the prediction length). After this iteration, we advance to the next sample window. Because we test sample windows of multiple lengths with predictions windows of different lengths, how we advance through the 10-year data set depends whether the prediction length is greater than the sample length, and vice versa. If the sample length is smaller or equal to the prediction length, we start the next sample window at the date immediately following the last date included in the previous sample window. If the prediction length is smaller than the sample length, the start of the new sample window is $T =$ the prediction length days later than the start of the previous sample window. This allows us to construct portfolios with a greater variety of data points from the sample. For the purpose of presenting these results, we will only be presenting the results from sample lengths of 90 days (equivalent to rebalancing a portfolio every 4 months) and 240 days (equivalent to rebalancing a portfolio every year approximately). The reason we have chosen these two sample lengths is that a 90-day sample length allows us to analyze the behavior of these filters when N and T are close to one another and a 240-day sample length allows us to examine this behavior when $N <<$ T. All data for other sample lengths and prediction lengths can be found in the appendices, however. All results are reported as annual results (~250 trading days) *(ii) Comparative Estimators*

For the purpose of comparing our data to previous data, we will be comparing our results to those of the sample covariance matrix, two spectral estimators (RMT-0 and RMT-M) and a CAPM-driven single-index model (SI). The RMT-0 method involves diagonalizing the sample correlation matrix, setting all eigenvalues smaller than:

$$\lambda_{max} = \sigma^2(1 + \frac{N}{T} + 2\sqrt{\frac{N}{T}}) \tag{28}$$

(where σ^2 = 1 for correlation matrices) equal to 0, reverting the modified diagonal matrix back to the standard basis, setting all diagonal elements to 1 and then multiplying each index by the square root of the variances of the stock returns of the two stocks that correspond to that particular index [4]. The RMT-M method is very similar to the RMT-0 method, but instead of setting all eigenvalues of the diagonalized correlation matrix smaller than λ_{max} equal to 0, they are averaged, and this average value replaces each individual eigenvalue. This preserves the trace of the matrix [4]. The rest of the procedure is the same. The SI model involves forming the covariance matrix from each individual stock's exposure to the market portfolio (using the value-weighted daily returns of the S&P 500 as a market indicator). This relationship is expressed by the equation:

$$\boldsymbol{C}^{(SI)} = \sigma_M^2 \boldsymbol{\beta}\boldsymbol{\beta}^{\boldsymbol{T}} \tag{29}$$

where $\boldsymbol{\beta}$ is a vector of β_i terms derived using (7) and $\boldsymbol{C}^{(SI)}$ is the covariance matrix estimator using the SI model.

(iii) The Behavior of the iterative MVDR and Block Matrix Filters

The first step to understanding how the iterative MVDR and block matrix filters can help us reduce the realized risk of portfolios is to examine the behavior of the filter when applied to portfolio data for both sampling windows, starting with the 90-day window.

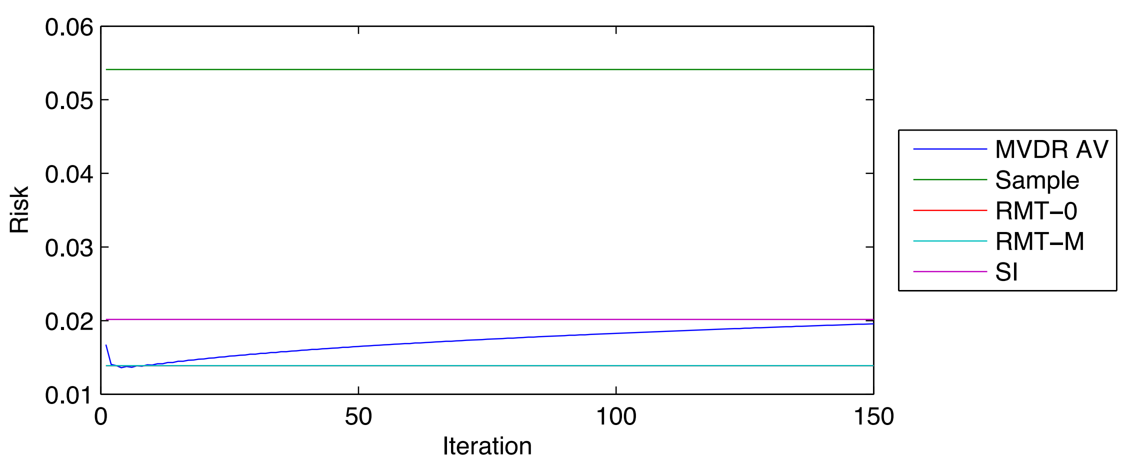

Figure 5: Iterative MVDR Realized Risk Behavior for 90-day sampling window

As can be seen for the iterative MVDR case, we achieve portfolios whose realized risks are far inferior to that of the minimum variance portfolio of the sample covariance matrix up until 150 iterations. Additionally, we can see that the minimum risk achievable happens on average around the 5th iteration. This minimum is the index we are trying to discover using cross-validation. Although the cross-validation exercise is not as necessary for the 90-day sample window because every portfolio achieved through the MVDR iterative algorithm is on average a vast improvement over the sample covariance matrix MVP, it will be more important for the longer sample windows.

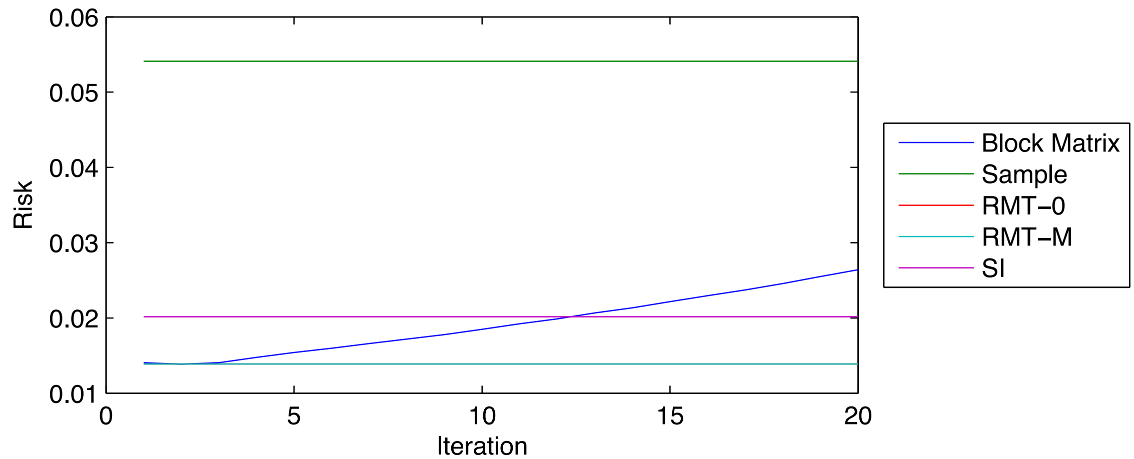

Figure 6: Block Matrix Realized Risk Behavior for 90-day sampling window

For the block matrix algorithm for the 90-day sample window, we see a similar overall pattern, except that there is less of a noticeable trough for the first few iterations. Both filters eventually converge to the sample covariance matrix MVP if they remain stable. Once again, however, it is clear that the block matrix algorithm is an improved estimator compared to the sample covariance matrix MVP.

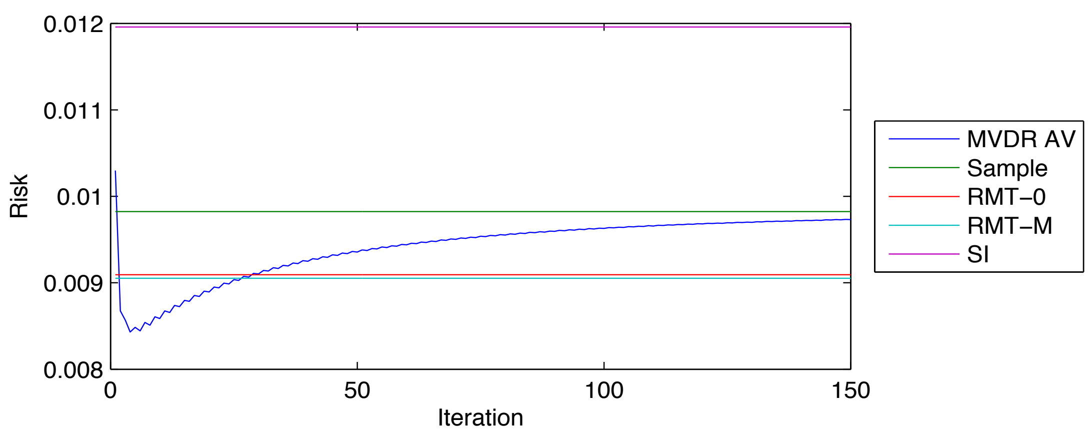

Figure 7: Iterative MVDR Realized Risk Behavior for 240-day sampling window

For the 240-day sample window, the regular behavior of the MVDR iterative filter (i.e. when $T >> N$) is a lot clearer. We can witness the trough that corresponds to the lowest possible realized risk using the iterative MVDR filter, and it yields a superior portfolio in terms of realized risk than both the sample covariance matrix MVP and almost every other comparative estimator, as well. For a 240-day sample window it becomes much more important to achieve a good cross-validation estimate of the minimum possible realized risk because the filter's performance in relation to other comparative estimators is clearly closely linked to correctly identifying the minimum realized risk iteration number.

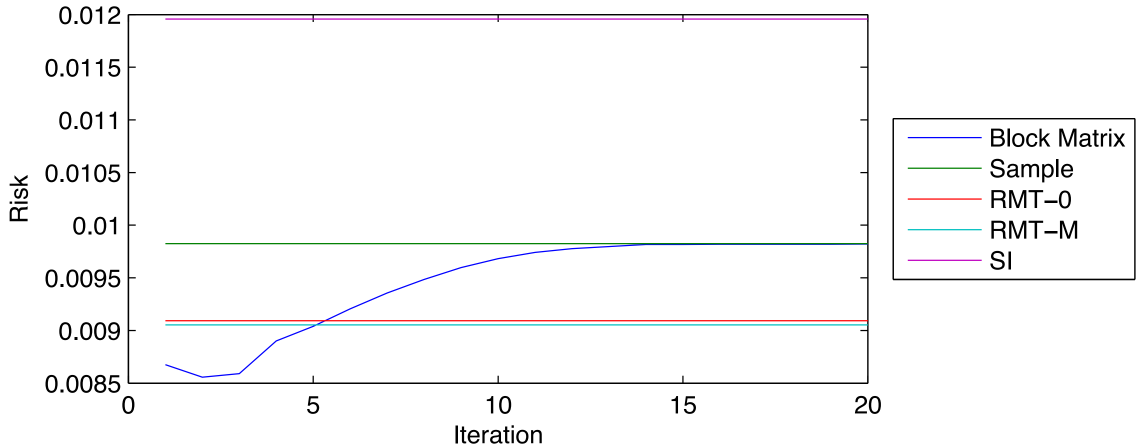

Figure 8: Block Matrix Realized Risk Behavior for 240-day sampling window

In Figure 8, we see that the same differences between the two sampling windows for the iterative MVDR filter also apply to the Block Matrix Filter. The minimum possible realized risk is clearly superior to that of even the closest comparative estimator and finding this minimum index using cross-validation is essential.

(iv) Cross-Validation Results

One of the questions we must answer before moving forward in our analysis of whether our estimators are useful predictors of realized risk is whether or not we can consistently identify the weight vector for the iterative MVDR and Block Matrix filters that yields the lowest risk through cross-validation. This is necessary because the knowledge that we can reduce our realized risk using these two estimators is not enough. We must be able to pinpoint the exact weight vector for which the realized risk will be minimized. In order to test this hypothesis, we applied the cross-validation algorithm to our simulation using validation sets whose size were 3.33% (known as 30-fold cross-validation as there will be 30 iterations of the cross-validation algorithm), 6.67% (15-fold cross-validation), 10% (10-fold cross-validation), 16.67% (6-fold cross-validation), 20% (5-fold cross-validation), and 33.33% (3-fold cross-validation) of the length of overall sample window. After computing the predicted index for the optimal weight vector and substituting this vector of portfolio weights into (2) to compute our anticipated optimal realized risk, we pinpointed the weight vector that actually corresponded to the lowest realized risk over the course of the prediction window. We did this for every sample and prediction window pair over the course of 11-year data set and computed the average absolute difference between each cross-validation estimate and the actual minimum achievable realized risk.

	MVDR Iterative Filter		Block Matrix Filter	
Validation Set Size	Average Difference	Standard Deviation	Average Difference	Standard Deviation
30-fold	.002322	.001799	.002654	.001914
15-fold	.001920	.001697	.001818	.001792
10-fold	.001312	.001369	.001417	.001546
6-fold	.001058	.001111	.001218	.001203
5-fold	.001009	.001014	.001116	.001162
3-fold	**.000640**	**.000680**	**.000631**	**.000729**

Table 1: Cross-Validation Results for 90-day sample and prediction window

For the case where the sampling and prediction windows are 90 days long, it is clear that a 3-fold cross-validation (this corresponds to 30 days) provides an estimate of the realized risk that is more accurate and more precise than any other validation set size. The average difference and the standard deviation of that difference are both lowest for the 3-fold cross-validation for both filters. And the difference is considerable. For the MVDR iterative filter, the 3-fold cross-validation provides an average difference that is 36.5% smaller than the 5-fold cross-validation estimate, the next smallest average difference. For the Block Matrix filter, this discrepancy is even greater, at 43.1%. Additionally, it is important to note that as we decrease the size of the validation set, these average differences become greater and their standard deviations increase as well. Therefore, we can infer that the validation set must be big enough so that its covariance matrix is an adequate representation of the one we are trying to predict.

	MVDR Iterative Filter		Block Matrix Filter	
Validation Set Size (as % of sample window)	Average Difference	Standard Deviation	Average Difference	Standard Deviation
30-fold	.000661	.000630	.000532	.000759
15-fold	.000505	.000625	.000428	.000781
10-fold	.000517	.000577	.000342	.000517
6-fold	.000333	.000673	.000411	.000770
5-fold	.000305	**.000453**	.000342	.000691
3-fold	**.000244**	.000461	**.000188**	**.000287**

Table 2: Cross-Validation Results for 240-day sample and prediction window

We can witness a similar pattern when examining the cross-validation data for a 240-day sample and prediction window, although it is less distinguished for the iterative MVDR estimate. For the block matrix, the 3-fold cross-validation still has the smallest average difference between its cross-validation estimate and the actual minimum realized risk achievable, as well as the smallest standard deviation for this average difference. The percentage difference between the 3-fold cross-validation estimate and the next smallest average difference estimate is 45%. Similarly, the 3-fold cross-validation estimate is also far more precise than any other estimate, with a standard deviation that is 44.4% smaller than the next smallest standard deviation for any estimate. For the iterative MVDR filter, the 3-fold cross-validation estimate is still clearly the best cross-validation estimate, but this conclusion is not as obvious. While the 3-fold estimate still gives the most accurate estimate, as its average difference from the optimal minimum realized risk is 20% smaller than the next best cross-validation estimate, it is not the most precise estimate. The standard deviation of the 5-fold estimate is slightly smaller. This difference is not significant, however, and the 3-fold estimate is a superior cross-validation estimate due to its lower average difference.

As such, it can be seen that the 3-fold estimate is the superior cross-validation estimate for the cases where the sample window and prediction window are both 90 days or both 240 days long. As it turns the 3-fold cross-validation estimate is also the best estimate for almost every sample length and prediction length combination. The only exception is for 30-day prediction windows. In these cases, the best estimate is the 30-fold cross-validation estimate.

(v) Comparative Results

In order to practically assess how our filters performed compared to existing strategies, we decided to compare our 3-fold cross-validation results to the comparative estimators outlined in *(ii)*. Because we will never be able to know which index in both filters corresponded to the weight vectors that achieved the lowest realized risk until after the fact, we chose to compare these baseline estimators with what we believe is the most robust estimate of the minimum achievable realized risk: the 3-fold cross-validation estimate. In order to report these results, we have outlined the average percentage difference between the 3-fold cross-validation estimate of realized risk and the sample covariance, RMT-0, RMT-M, and single index realized risks, and the standard deviation of these percentage differences.

	MVDR Iterative Filter		Block Matrix	
Estimator	Average Difference (%)	Standard Deviation (%)	Average Difference (%)	Standard Deviation (%)
Sample Covariance	**-68.4694**	17.2278	**-68.3988**	17.0042
RMT-0	0.1896	10.2984	0.8708	10.9947
RMT-M	0.0815	10.7721	**-0.0623**	8.6768
Single Index Model	**-26.6931**	17.4361	**-26.2005**	17.7261

Table 3: 90-day sample and prediction window comparative results

For a 90-day sample window (which is roughly the case where $T \approx N$), we can clearly see that the iterative MVDR and block matrix algorithms are significantly superior estimates for reducing the realized risk of a portfolio than the sample covariance matrix. Both estimators on average yield a percentage difference between the estimator and the sample covariance matrix MVP of greater than 68%. While the standard deviations of this average are around 17% for both estimators, this is still a huge significant discrepancy. This huge discrepancy could be a result of the fact that $T \approx N$ for a 90-day sample length, which could cause the data to behave exceptionally (it is important to note that when $T < N$, the sample covariance matrix is singular). Conversely, the iterative MVDR and block matrix algorithms do not perform particularly well in relation to the RMT-0 and RMT-M techniques. On average, the 3-fold cross-validation realized risk estimate yields a realized risk that is greater than the realized risk of both of these comparative estimators. This average difference is to the order of 10^{-2} of a percentage, however, and is practically insignificant. But because the standard deviation of these differences is around 8% to 10%, we conclude that the iterative MVDR and block matrix estimators are weaker tools for this sample length as the greater spread of possible realized risks makes it a less predictable estimator.

	MVDR Iterative Filter		Block Matrix	
Estimator	Average Difference (%)	Standard Deviation (%)	Average Difference (%)	Standard Deviation (%)
Sample Covariance	**-12.3734**	7.8807	**-12.0477**	7.5310
RMT-0	**-5.2286**	6.0561	**-4.7355**	7.4591
RMT-M	**-4.8335**	6.7994	**-4.7344**	11.2662
Single Index Model	**-24.0429**	14.5680	**-23.8739**	13.5693

Table 4: 240-day sample and prediction window comparative results

For the 240-day sample length, our estimators are more consistently superior than the other comparative estimators, but less drastically so (for the case of the sample covariance matrix MVP). On average, both estimators are only roughly 12% superior to the sample covariance matrix MVP (as opposed to 68% in the case of the 90-day sample length), but they are also on average superior to the comparative estimators. Both the iterative MVDR and block matrix algorithms achieve a realized risk (through 3-fold cross-validation) that is on average about 5% lesser than that of both the RMT-M and RMT-0 and 24% lesser than the single index model. Seeing as the standard deviation of these average differences ranges between 6% and 11% (for the spectral estimators and the sample covariance matrix MVP) and 14% and 15% for the single index model, we can conclude that these average differences are significant for each estimator.

Impact on Society and the Environment:

My project is based on the theoretical application of robust minimum variance and low rank filters to a covariance matrix of stock returns in order to reduce portfolio risk. As such, there is no real environmental benefit to its application.

The purpose of my research is centered on lowering the risk of investment portfolios, thereby increasing the financial security of the owners of said portfolios. This pursuit has huge potential effects on society as a whole. The majority of the populations of Canada and the United States are in some way an investor. Some are active investors whose careers involve continuously trading different securities and living off of their commissions, capital gains and bonuses. Most people, however, are passive investors who invest money into retirement plans, mutual funds, or pension funds, with the hope of earning long-term returns on their savings. As the goal of these investors is to enlarge their life savings over time, they often prefer lower risk gains as opposed to high risk, high return investment strategies. As such, the formation of low-risk portfolios by the funds they invest in is of paramount importance. The safety of passive investors' financial assets is a valid concern that must be addressed and I hope my research will lead to portfolio formation strategies that resolve some of these issues.

In a country like the United States, company pensions are often invested in large trillion dollar pension funds until the days where they must be handed out. These pension funds cannot afford large losses or corporations and governments could default under the pressure of having to pay out more money in pensions than they could afford. While the only true way to assure than an asset is low-risk is to invest in short-term treasury bills (whose return rate is usually assumed to be the risk-free), the returns afforded off of this approach are often not high enough to secure the money that will be needed to finance worker pensions and retirements. As such pension and mutual funds look to build portfolios that they predict will offer significantly larger returns at marginally higher risk thresholds. While it is often very difficult to predict future returns due to changing market conditions, the risks of future portfolios can be more accurately calculated due to previous knowledge of how stocks have moved in relation to each other. While the Capital Asset Pricing Model predicts that the market portfolio should in theory be the best balance between systematic risk and return, the results in practice can often be different. Due to the inherent estimation error in formulating covariance matrices, it is possible to reduce the overall risk of a portfolio by balancing this estimation error with other sources of possible error in different models. Thus, once we have formulated an optimal covariance matrix, we can assign our capital to different stocks in the portfolio in order to achieve the lowest overall risk for this portfolio. As such, we will potentially be able to increase the security of pension funds, mutual funds and other passive investment bodies without resorting to minimal return strategies.

This is not a certainty, however, and it is important to recognize that a model is constrained to the assumptions it is built on. Investors who do not recognize potential limitations of the model they are using, and who use a model outside of the assumptions used to formulate it could possibly suffer huge losses in the stock market and ruin the future financial security of many of their clients.

Conclusion:

The iterative MVDR and block matrix filters are clear improvements over the sample covariance matrix estimator minimum variance portfolio in terms of realized risk. This improvement becomes greater as we approach the limit where the sample covariance matrix is still not singular (i.e. as T approaches N from the right). Their limitations lie in the fact that we cannot know at which iteration the minimum realized risk is achieved using these techniques. This shortcoming makes us rely on cross-validation to estimate the ideal iteration number that will yield this minimum realized risk. Unfortunately, while on average our cross-validation estimate is very close to the minimum achievable realized risk, there is still a considerable spread in how close the cross-validation estimate can be. This fact makes our estimate weaker than if we were able to pinpoint the minimum possible realized risk at every application of the filter.

Another shortcoming of both of these techniques is that they do not necessarily outperform other risk-controlling techniques. As T approaches N from the right, the comparative estimators tend to outperform both of these iterative algorithms, and the spread of these iterative algorithms makes them less attractive to more risk-averse investors. As our sample length becomes greater, the iterative algorithms outperform the other estimators (as well as the sample covariance matrix MVP), but the average difference is less than the standard deviation of this average difference (for the spectral estimators). While this does not necessarily mean that the average difference between our algorithms and the comparative estimators is no significant, it does point to the fact that the spread of our results may be unattractive to a more risk-averse investor who is looking for a more stable realized risk.

Risk analysis in portfolios is an inherently flawed practice. Because if we assume that the market behaves randomly, it is theoretically impossible to predict future returns. Even if we could accurately apply a fundamental analysis of companies' financial information in order to predict their performance, any model we design would still sensitive to behavioral finance considerations. We are therefore constrained to using historical data to augment our risk models, which leaves us susceptible to large estimation error as the number of data points available from which to sample is often smaller than or similar to the number of data points we are extrapolating. As such, identifying the risk of a portfolio, and attempting to reduce it is really an exercise of balancing the multiple types of error that can infiltrate our calculations. It is therefore incredibly important to outline the assumptions that accompany research on this subject, as all models to identify the risk of a portfolio can only be applied in light of these assumptions.

There is still, much research that can be done on these two risk estimators. First and foremost, altering the algorithms for calculating the block matrix and the iterative MVDR weight matrices in order to impose a no-shorting condition is a very interesting problem on its own that I will continue to analyze. Many mutual funds, pension funds, and other institutional investors are prohibited from shorting stocks in their portfolios. If similar results could be achieved for a model with this additional constraint, the model could be extended to the investing practices of all these other institutional investors. Furthermore, it could be beneficial to research the types of returns we can expect over these sample periods using these two risk estimators. If we are achieving consistent negative returns with these methods, it may not be worth it to continue applying resources its research. Finally, it would be interesting to see if these two filters can be applied to other covariance matrix estimators, such as the RMT-0 and RMT-M spectral estimators, as well as many others outlined in the Background section, to reduce realized risk even further. One of the advantages of these AV filters is that from a mathematical point of view, they do not alter our covariance matrix estimator. Therefore, they can be applied to any covariance matrix estimation estimator in order to find the optimal portfolio in terms of realized risk for that particular estimator.

References:

[1] G. Ganesan, “A Subspace Approach to Portfolio Analysis,” *IEEE Signal Processing Magazine*, vol. 28, no. 5, pp. 49–60, Sep. 2011.

[2] D. J. Disatnik, and S. Katz, “Portfolio Optimization Using a Block Structure for the Covariance Matrix.” *SSRN Electronic Journal*, 2011.

[3] T. L. Lai, H. Xing, and Z. Chen, “Mean–variance portfolio optimization when means and covariances are unknown,” *The Annals of Applied Statistics*, vol. 5, no. 2A, pp. 798–823, Jun. 2011.

[4] E. Pantaleo, M. Tumminello, F. Lillo, and R. N. Mantegna, “When do improved covariance matrix estimators enhance portfolio optimization? An empirical comparative study of nine estimators,” *Quantitative Finance*, vol. 11, no. 7, pp. 1067–1080, Jul. 2011.

[5] V. DeMiguel, L. Garlappi, F. J. Nogales, R. Uppal, “A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms.” *Management Science*. 55:798–812. 2009

[6] M. Capiński, and T. Zastawniak. *Mathematics for Finance, An Introduction to Financial Engineering*, London: Springer, 2003.

[7] R. Schafer, N.F. Nilsson, T. Guhr, “Power mapping with dynamical adjustment for improved portfolio optimization”. *Quant. Finance*, vol. 10, pp. 107–119, 2010

[8] J. Fan, Y. Fan, and J. LV, “High dimensional covariance matrix estimation using a factor model” *J. Econometrics* 147 186–197. MR2472991. 2008

[9] J. Z. Huang, N. Liu, M. Pourahmadi and L. Liu, “Covariance matrix selection and estimation via penalized normal likelihood.” *Biometrika* 93 85–98. MR2277742. 2006

[10] O. Ledoit, and M. Wolf, “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection.” *Journal of Empirical Finance*, vol. 10, no. 5, pp. 603-621. 2003

[11] M.C. Münnix , R. Schäfer & O. Grothe. “Estimating correlation and covariance matrices by weighting of market similarity”, *Quantitative Finance*, DOI:10.1080/14697688.2011.605075, 2011

[12] T. Ma, and R. Jagannathan, “Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps”. NBER Working Paper No. w8922, 2002

[13] "U.S. Fundamental Equity Risk Model". Northfield Info. Serv., Inc., Boston, MA, Final Report, Mar. 3, 2005.

[14] P. J. Bickel and E. Levina, "Covariance regularization by thresholding." *Ann. Statist.* 36, 2577–2604. 2008

[15] I. Kondor, S. Pafka, and G. Nagy, "Noise sensitivity of portfolio selection under various risk measures." *J. Bank. Finance*, 31, 1545–1573. 2007.

[16] V. Tola, F. Lillo, M. Gallegati, and R. N. Mantegna, "Cluster analysis for portfolio optimization." *Journal of Economic Dynamics and Control*, vol. 32, no. 1, pp. 235-258, 2008.

[17] J. Schäfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statistical applications in genetics and molecular biology*, vol. 4, no. 2005.

[18] D. A. Pados and S. N. Batalama, "Joint space-time Auxiliary-Vector filtering for DS/CDMA systems with antenna arrays," *IEEE* Trans. Comm., vol. 47, pp. 1406-1415, Sept. 1999.

[19] D. A. Pados and G. N. Karystinos, "An iterative algorithm for the computation of the MVDR filter." *IEEE Transactions on Signal Processing*, vol. 49, pp. 290–300, Feb. 2001.

[20] R. G. Lorenz and S.P. Boyd, "Robust minimum variance beamforming." *Signal Processing, IEEE Transactions on*, vol. 53, no. 5, pp. 1684-1696. 2005.

[21] S. A. Vorobyov, A. B. Gershman, and Z. Q. Luo, "Robust adaptive beamforming using worst-case performance optimization: A solution to the signal mismatch problem." *Signal Processing, IEEE Transactions on*, vol. 51 no. 2, 313-324. 2003.

[22] Y. S. Abu-Mostafa, M. Magdon-Ismail, H. Lin, *Learning from Data*, 2012.

[23] Wharton Research Data Services. Wrds-web.wharton.upenn.edu/wrds.