High Frequency Minimum-variance Portfolios

Jared Fisher and David Puelz

Jared Fisher

e-mail: jared.fisher@utexas.edu

David Puelz

e-mail: david.puelz@utexas.edu

Abstract: We construct minimum-variance portfolios using high-frequency realized covariance matrices. Our contributions are two-fold. First, we derive a penalized minimum variance loss function and ADMM procedure for calculating sparse-changing optimal portfolio weights. Second, we calculate high frequency realized covariance matrices to be used as inputs into the optimization procedure.

1. Introduction

As an alternative to mean-variance investing, the minimum variance portfolio is appealing. From a financial perspective, it boasts lower volatility than its predecessor, and doesn't "chase winners." Jagannathan and Ma (2003) and DeMiguel et al. (2009) have shown, among many others, that the minimum variance portfolio performs better out-of-sample than its mean-variance counterpart. An additional consideration is that expected returns are much harder to estimate than covariances, and this is widely discussed in the portfolio optimization literature, see Merton (1980) for a seminal paper. Thus, it is statistically desirable to consider an optimization that avoids specification of the first moment of returns.

We construct a minimum variance portfolio while reducing the amount of trivial changes to the investment strategy. This objective is relevant to both individual investors as well as large institutions since it penalize changes in the portfolio that constribute to transaction costs. This can be solved by penalizing changes to the strategic holdings. This is illustrated in our problem of interest,

$$\min_{\boldsymbol{w}_t} \quad \frac{1}{2}\boldsymbol{w}_t'\Sigma_t\boldsymbol{w}_t + \lambda||\boldsymbol{w}_t - \boldsymbol{w}_{t-1}||_1 \tag{1}$$

$$\text{subject to} \quad \boldsymbol{w}_t'\mathbf{1} = 1. \tag{2}$$

where $\boldsymbol{w}_t$ and Σ_t are the portfolio weights and stock return covariance matrix at time t, respectively. We subdivide this into two problems. First, we solve the above optimization problem. Second, we model Σ_t.

Our first contribution to existing portfolio optimization literature involves developing an alternating direction method of multipliers (ADMM) procedure tailored to the penalized minimum variance problem. The covariance modeling is a separate but important input to our ADMM procedure and represents the second contribution of this paper. We utilize high frequency trading data to estimate Σ_t using a *realized kernel* approach. This is informed by Windle et al. (2014), who demonstrate that the high fidelity estimation of Σ_t with high frequency data can aid significantly in improved portfolio performance. The coupling of a realistic portfolio objective with high quality optimization inputs provides a realistic framework to study minimum variance portfolios and the benefits of penalizing changes in portfolio allocations.

2. Optimization

Assuming Σ_t is known (at least up to an estimate), our problem can be solved with the alternating direction method of multipliers, or ADMM (Boyd et al., 2010). First, let $\boldsymbol{w}_{t-1} = \tilde{\boldsymbol{w}}$, $\boldsymbol{w}_t = \boldsymbol{w}$, $\boldsymbol{z} = \boldsymbol{w}_t - \boldsymbol{w}_{t-1}$.

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{z}} \quad & \frac{1}{2}\boldsymbol{w}'\Sigma_t\boldsymbol{w} + \lambda||\boldsymbol{z}||_1 && (3)\\ \text{subject to} \quad & \mathbf{1}'\boldsymbol{w} = 1, \boldsymbol{w} - \boldsymbol{z} = \tilde{\boldsymbol{w}} && (4) \end{aligned}$$

and combining the constraints we can rewrite the problem as

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{z}} \quad & \frac{1}{2}\boldsymbol{w}'\Sigma_t\boldsymbol{w} + \lambda||\boldsymbol{z}||_1 && (5)\\ \text{subject to} \quad & \begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z} = \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}. && (6) \end{aligned}$$

This yields the augmented Lagrangian

$$L_\rho(\boldsymbol{w},\boldsymbol{z},\boldsymbol{y}) = \frac{1}{2}\boldsymbol{w}'\Sigma_t\boldsymbol{w} + \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}'\left(\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z} - \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}\right) + \frac{\rho}{2}\left|\left|\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z} - \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}\right|\right|_2^2.$$

The ADMM iterates over the following updates until convergence is reached. The criteria for convergence is addressed in the next section.

$$\begin{aligned} \boldsymbol{w}_{k+1} &= arg\min_{w} \quad \frac{1}{2}\boldsymbol{w}'\Sigma_t\boldsymbol{w} + \boldsymbol{y}_k'\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w} + \frac{\rho}{2}\left|\left|\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z}_k - \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}\right|\right|_2^2 && (7)\\ \boldsymbol{z}_{k+1} &= arg\min_{z} \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k'\begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z} + \frac{\rho}{2}\left|\left|\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w}_{k+1} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z} - \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}\right|\right|_2^2 && (8)\\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \rho\left(\begin{bmatrix}\mathbf{1}'\\ I\end{bmatrix}\boldsymbol{w}_{k+1} + \begin{bmatrix}\mathbf{0}'\\ -I\end{bmatrix}\boldsymbol{z}_{k+1} - \begin{bmatrix}1\\ \tilde{\boldsymbol{w}}\end{bmatrix}\right) && (9) \end{aligned}$$

Each of the above equations can be simplified, and the step-by-step derivations are in the appendix. Here, we summarize. We can solve the first analytically by finding the gradient.

$$\begin{aligned}
0 &\overset{set}{=} \nabla_w L_\rho(w, z_k, y_k) \\
\Rightarrow \boldsymbol{w}_{k+1} &= -\left(\Sigma_t + \rho(I_n + J_n)\right)^{-1} \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k - \rho(\boldsymbol{z}_k + \tilde{\boldsymbol{w}} + \mathbf{1}) \right)
\end{aligned}$$

The second uses the soft thresholding function $S_\lambda(x) = sign(x) * (|x| - \lambda)_+$, as well as completing the square.

$$\begin{aligned}
\boldsymbol{z}_{k+1} &= arg\min_z \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \left|\left| \begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right|\right|_2^2 \\
&= arg\min_z \quad \frac{\lambda}{\rho}||\boldsymbol{z}||_1 + \frac{1}{2} \left|\left| \boldsymbol{z} - \left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right) \right|\right|_2^2 \\
&= S_{\lambda/\rho}\left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right)
\end{aligned}$$

The last equation only needs algebra to simplify which we show here.

$$\begin{aligned}
\boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \rho \left(\begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z}_{k+1} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= \boldsymbol{y}_k + \rho \left(\begin{bmatrix} \mathbf{1}'\boldsymbol{w}_{k+1} \\ I\boldsymbol{w}_{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{0}'\boldsymbol{z}_{k+1} \\ -I\boldsymbol{z}_{k+1} \end{bmatrix} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= \boldsymbol{y}_k + \rho \left(\begin{bmatrix} \sum_i \boldsymbol{w}_{k+1,i} \\ \boldsymbol{w}_{k+1} \end{bmatrix} + \begin{bmatrix} 0 \\ -\boldsymbol{z}_{k+1} \end{bmatrix} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= \boldsymbol{y}_k + \rho \begin{bmatrix} \sum_i \boldsymbol{w}_{k+1,i} - 1 \\ \boldsymbol{w}_{k+1} - \boldsymbol{z}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix}
\end{aligned}$$

Thus, our ADMM simplifies to

$$\begin{aligned}
\boldsymbol{w}_{k+1} &= -\left(\Sigma_t + \rho(I_n + J_n)\right)^{-1} \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k - \rho(\boldsymbol{z}_k + \tilde{\boldsymbol{w}} + \mathbf{1}) \right) && (10) \\
\boldsymbol{z}_{k+1} &= S_{\lambda/\rho}\left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right) && (11) \\
\boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \rho \begin{bmatrix} \sum_i \boldsymbol{w}_{k+1,i} - 1 \\ \boldsymbol{w}_{k+1} - \boldsymbol{z}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix} && (12)
\end{aligned}$$

2.1. Stopping Rules

We consider that convergence is reached once all residuals are sufficiently small. Again following Boyd et al. (2010), we define the "primal residual" as r and the dual residual s as follows

$$\begin{aligned} r_{k+1} &= \begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z}_{k+1} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \\ &= \begin{bmatrix} (\sum_i w_{i,k+1}) - 1 \\ \boldsymbol{w}_{k+1} - \boldsymbol{z}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix} \\ s_{k+1} &= \rho \begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix}' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} (z_{k+1} - z_k) \\ &= \rho(-I)(z_{k+1} - z_k) \\ &= -\rho(z_{k+1} - z_k). \end{aligned}$$

Hence, stop iterating and claim convergence when both of the following criteria are met. For $\epsilon_r, \epsilon_s > 0$,

$$\begin{aligned} ||r_{k+1}||_2 &< \epsilon_r \\ ||s_{k+1}||_2 &< \epsilon_s \end{aligned}$$

In practice, we augment the original algorithm with the method prosed by Tansey et al. (2016) (page 14 of version 1, not clarified in version 2), where the step size ρ is adjusted if $||r_{k+1}||_2$ and $||s_{k+1}||_2$ sufficiently different in size.

3. Covariance

In this section, we discuss how to compute the main (in fact, only) parameter of the optimization procedure: Σ_t.

Suppose a stock price is given by the continuous time stochastic process $\{S_t\}_{t\geq 0}$ where t is measured in days. Let $R_t = \log S_t - \log S_0$ be the cumulative log return of the stock and $r_t(\delta) = R_t - R_{t-\delta}$ be the "δ-log-returns." The realized variance for day t is then defined as:

$$RV_t(\delta) \equiv \sum_{\delta_i \in (t-1,t]} r^2_{\delta_i}(\delta) \tag{13}$$

In words, a stock's realized variance for day t is just the sum of intraday squared returns. As $\delta \to 0$, the realized variance converges to the daily quadratic variation of the stock.

This definition can be extended to many stocks to construct the realized covariance:

$$RC_t(\delta) \equiv \sum_{\delta_i \in (t-1,t]} \vec{r}_{\delta_i}(\delta)\vec{r}_{\delta_i}(\delta)^T \tag{14}$$

Due to microstructure noise, finite time-scale effects and asynchronous price data observed in practice, summing directly over the intraday squared returns will not give the appropriate realized covariance. As a result, we follow the *realized kernel* method of Barndorff-Nielsen et al. (2011) to compute realized kernels $\{RK_t\}$ for estimates of the realized covariance. In our optimization procedure above, RK_t will take the place of the covariance matrix Σ_t.

Since we are forming portfolios at time t for investment at time $t+1$, one-step ahead predictions are also relevant for this exercise. Windle and Carvalho (2012) and Windle et al. (2014) showed that exponentially smoothed realized kernels produce better forecasts than factor stochastic volatility models. As a second step, we hope to incorporate smoothed predictions for the covariance into our optimization procedure.

4. Data

We use data from the New York Stock Exchange Trade and Quote (NYSE TAQ) database from 1999 through 2012. This data includes all intraday transactions for securities listed on NYSE and AMEX (American Stock Exchange).

It is big data. When uncompressed, the data takes up approximately six terabytes of space[1]. The data must be processed before calculating realized covariance. There are three steps:

1. Constructing a dataframe of prices indexed by seconds after midnight.
2. Cleaning price data.
3. Synchronizing and refreshing prices.

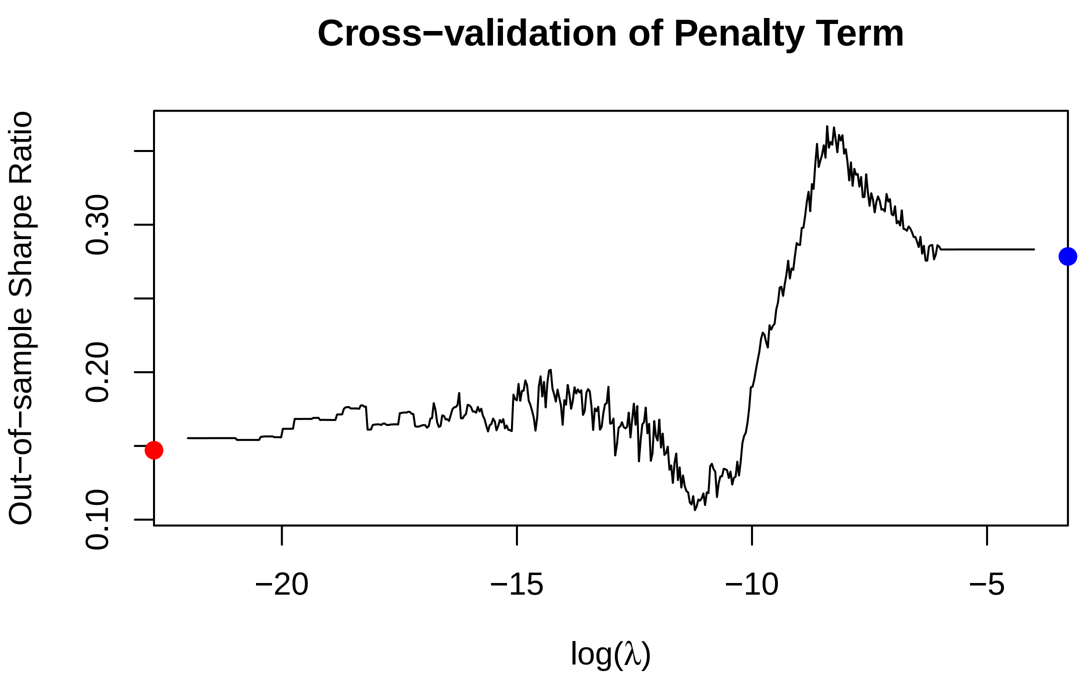
After these steps are completed, we compute daily realized kernels $\{RK_t\}$ of the most liquid stocks that vary in dimension from roughly 700×700 to 1000×1000. Thus, we transform this big data set into a *slightly* less big (although more structured and higher quality) data set of realized kernels for only the 30 Dow Jones Industrial Average stocks.

We supplement this dataset with daily returns (with dividends) from these 30 firms obtained from The Center for Research in Security Prices (CRSP). Both CRSP and TAQ can be accessed on the Wharton Research Data Services (WRDS) website.

[1] There are roughly two gigabytes of transaction data for each trading day.

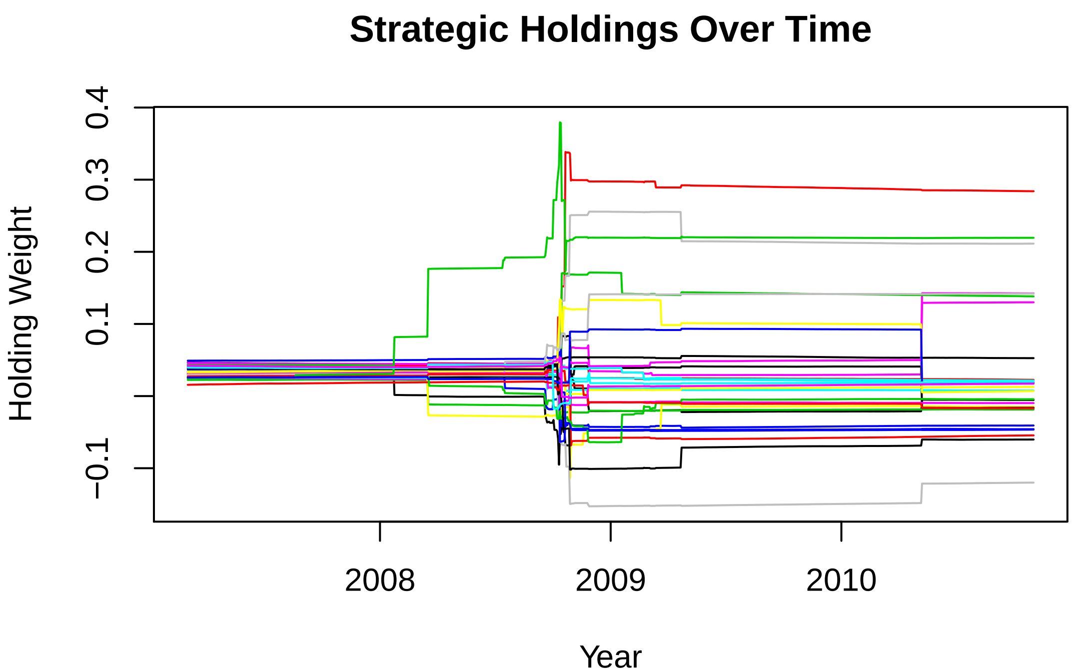
5. Choice of λ Penalty

Clearly, the choice of λ will have a significant impact on the performance of the portfolio. Below we present a grid search for the ideal penalty value, in the spirit of cross-validation. The out-of-sample Sharpe ratio is our measurement of interest. The blue dot represents the Sharpe ratio of an equal weighted portfolio, holding each of the 30 stocks in equal proportion. This is the limiting case as λ increases, if your initial holdings are even across all stocks. The red dot represents the Sharpe ratio unpenalized minimum variance portfolio, namely when $\lambda = 0$. The highest Sharpe ratio occurs at $\lambda \approx 0.00022 \approx exp(-8.4)$.



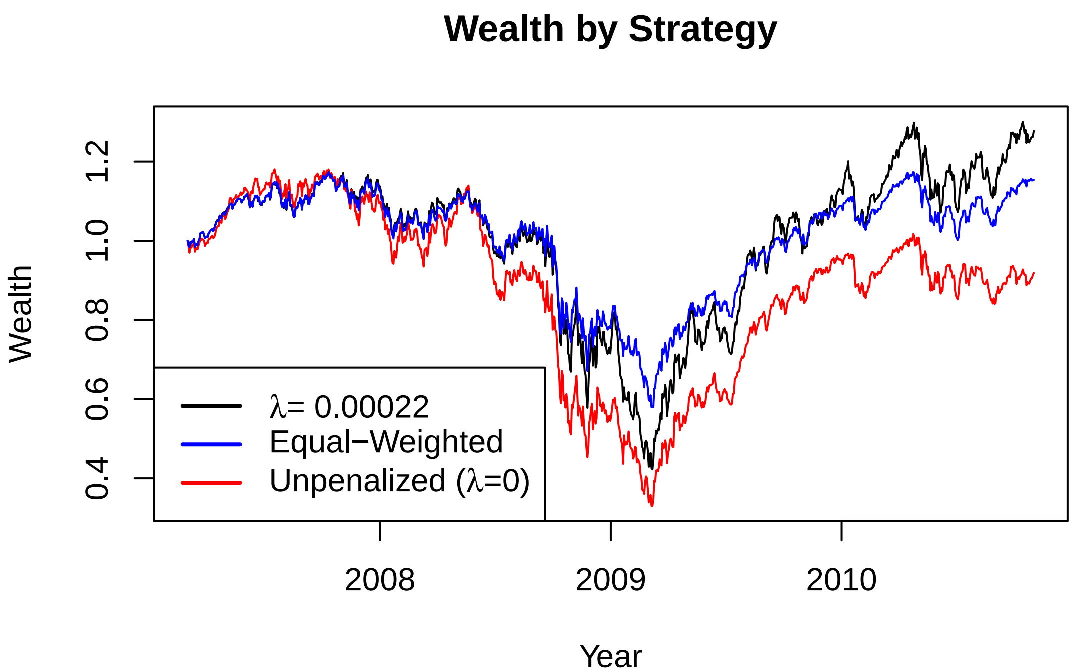
6. Results

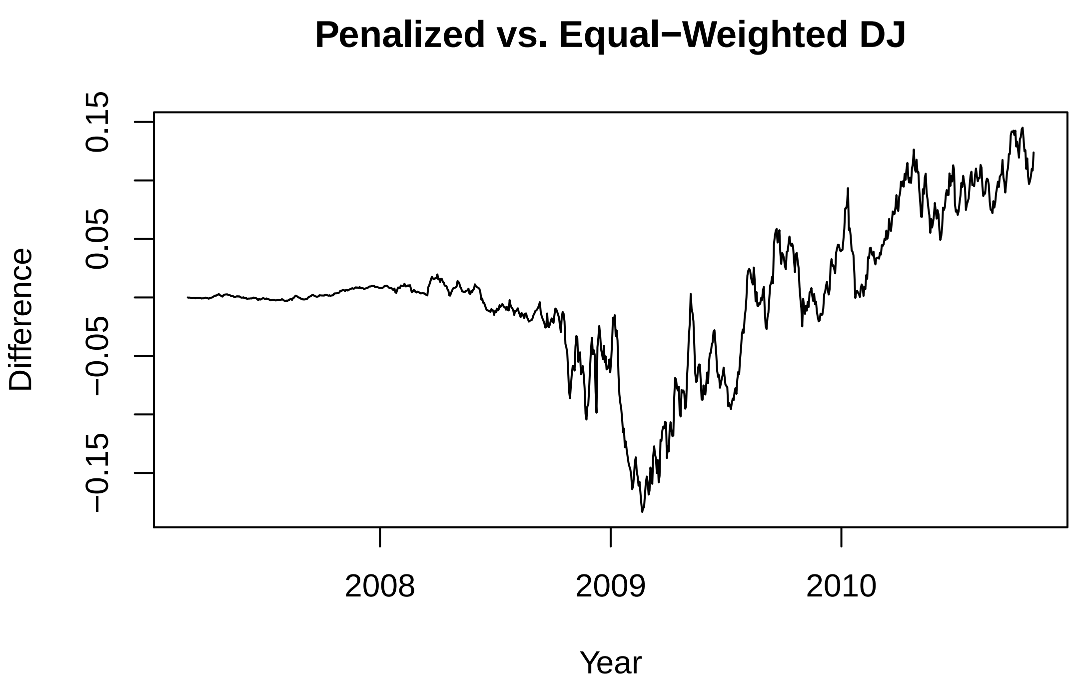
We present the following results based upon using the optimal $\lambda = 0.00022$, optimal in the sense of *ex post* out-of-sample Sharpe ratio. Choosing the correct λ *a priori* is a subject of future research. Presently, these results aim to demonstrate the nature of this portfolio management technique. Recall that the goal of this project is to temper the minimum variance portfolio away from trivial changes to the investment strategy. The plot below presents each of the different holdings over time for all 30 assets.



These visually appear to contain fairly few changes, but there are numerical differences. Over 99% of trading days have firm-weight changes by more than 0.00001. Yet, only about 6.5% of trading days have any firm-weight change by more than 0.0001. We feel this is reasonable for our numerical precision here. Naturally, we could increase precision if we decrease our tolerance levels ϵ_r, ϵ_s, but this is not necessary for our proof of concept here.

It is interesting to note the large fluctuations in holdings during late 2008, as the economy was falling and clearly the portfolio trying to compensate. Furthermore, it's also interesting how the holdings relatively stabilize before 2008 concludes (we say relatively as there are some changes moving forward, but none are drastic). This was before the trough in early 2009. However, the general holdings established at this time then carry our portfolio strategy to outperform both benchmarks in the remainder of the sample time period. This is clearly seen in the following plot of wealth over time. The plot shows the growth of investor wealth given a $1 initial investment. This pattern is also seen in the second plot, which shows the difference in wealth between our portfolio and the equal-weighted benchmark, chosen because it performs better than the minimum variance portfolio benchmark.


Wealth by Strategy
Wealth
0.4
0.6
0.8
1.0
1.2
λ= 0.00022
Equal−Weighted
Unpenalized (λ=0)
2008
2009
2010
Year


Penalized vs. Equal−Weighted DJ
Difference
−0.15
−0.05
0.05
0.15
2008
2009
2010
Year

7. Conclusion

We have presented a methodology for adjusting the minimum-variance portfolio, namely restricting its ability to make small changes to the investment strategy. We show the merits of this strategy by using high-frequency data, from which we construct daily realized covariance matrices. These matrices are using as the covariance estimates in optimization to find the target portfolio weights, which is rebalanced daily according to the target. We find that this investment strategy is advantageous, increasing wealth by more than 10% over an equal-weighted Dow Jones benchmark during the Great Recession.

References

Barndorff-Nielsen, O. E., Hansen, P. R., Lunde, A. and Shephard, N. (2011). Multivariate realised kernels: consistent positive semi-definite estimators of the covariation of equity prices with noise and non-synchronous trading. *Journal of Econometrics* **162** 149–169. 5

Boyd, N. S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2010). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* **3** 1-122. 2, 4

DeMiguel, V., Garlappi, L., Nogales, F. J. and Uppal, R. (2009). A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science* **55** 798–812. 1

Jagannathan, R. and Ma, T. (2003). Risk reduction in large portfolios: Why imposing the wrong constraints helps. *The Journal of Finance* **58** 1651–1684. 1

Merton, R. C. (1980). On estimating the expected return on the market: An exploratory investigation. *Journal of financial economics* **8** 323–361. 1

Tansey, W., Koyejo, O., Poldrack, R. A. and Scott, J. G. (2016). False discovery rate smoothing. *arXiv.org*. 4

Windle, J. and Carvalho, C. M. (2012). Forecasting high-dimensional, time-varying covariance matrices for portfolio selection. *The University of Texas at Austin*. 5

Windle, J., Carvalho, C. M. et al. (2014). A tractable state-space model for symmetric positive-definite matrices. *Bayesian Analysis* **9** 759–792. 2, 5

8. Appendix

8.1. Derivation of Minimums

Here we show how to simplify the ADMM components, line by line. We simplify equation 7 into equation 10 by finding the gradient.

$$
\begin{aligned}
\frac{\partial}{\partial w} L_\rho(w, z_k, y_k) &= \Sigma_t \boldsymbol{w} + \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k + \rho \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} \mathbf{0}' \\ -I_n \end{bmatrix} \boldsymbol{z}_k - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= \Sigma_t \boldsymbol{w} + \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k + \left(\rho \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix} \boldsymbol{w} + \rho \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \begin{bmatrix} \mathbf{0}' \\ -I_n \end{bmatrix} \boldsymbol{z}_k - \rho \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= \Sigma_t \boldsymbol{w} + \begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k + \rho(I_n + J_n)\boldsymbol{w} + \rho(-I_n)\boldsymbol{z}_k - \rho(\mathbf{1} + \tilde{\boldsymbol{w}}) \\
&= \left(\Sigma_t + \rho(I_n + J_n)\right) \boldsymbol{w} + \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k - \rho(\boldsymbol{z}_k + \tilde{\boldsymbol{w}} + \mathbf{1}) \right) \\
\Rightarrow 0 &\overset{set}{=} \left(\Sigma_t + \rho(I_n + J_n)\right) \hat{\boldsymbol{w}} + \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k - \rho(\boldsymbol{z}_k + \tilde{\boldsymbol{w}} + \mathbf{1}) \right) \\
\Rightarrow \hat{\boldsymbol{w}} &= -\left(\Sigma_t + \rho(I_n + J_n)\right)^{-1} \left(\begin{bmatrix} \mathbf{1}' \\ I_n \end{bmatrix}' \boldsymbol{y}_k - \rho(\boldsymbol{z}_k + \tilde{\boldsymbol{w}} + \mathbf{1}) \right)
\end{aligned}
$$

Finally, we simplify equation 8 into 11 by expanding, completing the square, and then employing the soft thresholding function $S_\lambda(x) = sign(x) * (|x| - \lambda)_+$.

$$
\begin{aligned}
\boldsymbol{z}_{k+1} &= arg \min_z \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \left|\left| \begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right|\right|_2^2 \\
&= arg \min_z \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \left(\begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right)' \left(\begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} + \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right) \\
&= arg \min_z \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \rho \left(\begin{bmatrix} \mathbf{1}' \\ I \end{bmatrix} \boldsymbol{w}_{k+1} - \begin{bmatrix} 1 \\ \tilde{\boldsymbol{w}} \end{bmatrix} \right)' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \left(\begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} \right)' \left(\begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} \right) \\
&= arg \min_z \quad \lambda||\boldsymbol{z}||_1 + \boldsymbol{y}_k' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \rho \begin{bmatrix} \sum_i w_{k+1,i} - 1 \\ \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix}' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \boldsymbol{z}' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix}' \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} \\
&= arg \min_z \quad \lambda||\boldsymbol{z}||_1 + \left(\boldsymbol{y}_k' + \rho \begin{bmatrix} \sum_i w_{k+1,i} - 1 \\ \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix}' \right) \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{\rho}{2} \boldsymbol{z}' I \boldsymbol{z} \\
&= arg \min_z \quad \frac{\lambda}{\rho}||\boldsymbol{z}||_1 + \left(\frac{1}{\rho}\boldsymbol{y}_k' + \begin{bmatrix} \sum_i w_{k+1,i} - 1 \\ \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix}' \right) \begin{bmatrix} \mathbf{0}' \\ -I \end{bmatrix} \boldsymbol{z} + \frac{1}{2} \boldsymbol{z}' \boldsymbol{z} \\
&= arg \min_z \quad \frac{\lambda}{\rho}||\boldsymbol{z}||_1 + \begin{bmatrix} \frac{1}{\rho} y_{k,1} + \sum_i w_{k+1,i} - 1 \\ \frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \end{bmatrix}' \begin{bmatrix} 0 \\ -\boldsymbol{z} \end{bmatrix} + \frac{1}{2} \boldsymbol{z}' \boldsymbol{z} \\
&= arg \min_z \quad \frac{\lambda}{\rho}||\boldsymbol{z}||_1 - \left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right)' \boldsymbol{z} + \frac{1}{2} \boldsymbol{z}' \boldsymbol{z} \\
&= arg \min_z \quad \frac{\lambda}{\rho}||\boldsymbol{z}||_1 + \frac{1}{2} \left|\left| \boldsymbol{z} - \left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right) \right|\right|_2^2 \\
&= S_{\lambda/\rho} \left(\frac{1}{\rho} \boldsymbol{y}_{k,(-1)} + \boldsymbol{w}_{k+1} - \tilde{\boldsymbol{w}} \right)
\end{aligned}
$$

8.2. The Minimum Variance Portfolio, Without Penalties

Consider

$$
\begin{aligned}
\min_w \quad & \frac{1}{2} w^T \Sigma w \\
s.t. \quad & w^T \mathbf{1} = 1.
\end{aligned}
$$

This has the Lagrangian and first derivatives

$$
\begin{aligned}
L(w, \lambda) &= \frac{1}{2} w^T \Sigma w - \lambda (w^T \mathbf{1} - 1) \\
\frac{\partial L}{\partial w} &= \Sigma w - \lambda \mathbf{1} \\
\frac{\partial L}{\partial \lambda} &= -w^T \mathbf{1} + 1
\end{aligned}
$$

which yields the following first order needs, where the later is the obvious.

$$\begin{aligned}\Sigma w &= \lambda \mathbf{1} \\ \mathbf{1}^T w &= 1\end{aligned}$$

First, invert Σ.

$$w = \lambda \Sigma^{-1} \mathbf{1}$$

and then plug in this value for w in the second condition.

$$\begin{aligned}\mathbf{1}^T(\lambda \Sigma^{-1}\mathbf{1}) \quad &= 1 \\ \Rightarrow \mathbf{1}^T \Sigma^{-1} \mathbf{1} \quad &= \lambda^{-1}\end{aligned}$$

Hence, our solution is

$$w = \frac{1}{(\mathbf{1}^T \Sigma^{-1} \mathbf{1})} \Sigma^{-1} \mathbf{1}.$$