Predictive Markowitzs

Problem Statement

The Markowitz portfolio algorithm only minimised risk at specified target return levels but does not have a predictive element. Therefore, portfolios constructed from Markowitz have no foresight into how returns tend evolve. The predictive Markowitz was derived to tackle this issue.

Method

By bringing in OLS into the Markowitz equation, this new Lagrangian aims to minimise the product of these two. Consequently, creating a simultaneous minimisation equation for both prediction and portfolio risk.

$$\sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T} - M_i x | R_{i+T} - M_i x >$$

$$M_i, R_i \in R^{dxd} : i \in [0,n]$$
$$s.t. < x|1 > = 1, < x|\mu_{i+T} > = u^*, t_0 = r+1,...n, r \in [0,t_0], T \in (0,n), T > r$$

Note, because the set up is to also fit the incoming returns data, better prediction could yield a larger portfolio weight even though it is going in an undesirable direction. Therefore a third market guard component may be helpful, either as a constraint or in the main objective space.

$$\sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T} - M_i x | R_{i+T} - M_i x > . Cov(M_i|x > , R_m i|d_m >)^2$$

Or

$$\sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T} - M_i x | R_{i+T} - M_i x >$$

$$M_i, R_i \in R^{dxd} : i \in [0,n], t_0 = r+1,...n, r \in [0,t_0]$$
$$s.t. < x|1 > = 1, < x|\mu_{i+T} > = u^*, Cov(< d_m|R_i, < x|R_i) = c_m$$

Note: it is ideal to be close to the optimal portfolio at time T while minimising risk from time t_0 - r to t_0. Therefore, R_i should be multiplied by the optimal Marko portfolio from time t0 to T in future. So the Lagrangian could be preferably constructed as below.

$$r^{-1} \sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T}x_T^* - M_i x | R_{i+T}x_T^* - M_i x > \text{ s.t}$$

$$|x_T^* > is\,solution\,to\ < x|M_{i+T}|x >$$

Or

$$|x_T^* > is\,solution\,to\,r^{-1} \sum_{i=t-r_1}^{t} < x|M_{i+t}|x > . < R_{i+T}x - M_{i+T}x | R_{i+T}x - M_i x >$$

As imagined, this can be set up recursively such that

$$|x_T^* > is\,solution\,to\,r^{-1} \sum_{i=t-r_1}^{t} < x|M_{i+t}|x > . < R_{i+T}x_0 - M_{i+T}x | R_{i+T}x_0 - M_i x >$$

Where

$$|x_0 > is\,solution\,to\,r^{-1} \sum_{i=t-r_1}^{t} < x|M_{i+t}|x > . < R_{i+T}x_1 - M_{i+T}x | R_{i+T}x_1 - M_i x >$$

And so on and so forth.

Solution

Starting with the first approach, set up the Lagrangian, take the derivative then solve for |x>

$$L = r^{-1} \sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T}x_T^* - M_i x | R_{i+T}x_T^* - M_i x > - \lambda(< x|1 > - 1) - \zeta(< x|\mu_{i+T} > - \mu^*)$$

Take derivative, leave the sum for now for easier readability

$$\partial_{<x|}L = M_i|x > . < R_{i+T}x_T^* - M_i x >^2 - 2 < x|M_i|x > . M_i^T | R_{i+T}x_T^* - M_i x > = \lambda|1 > + \zeta|\mu_{i+T} >$$

$$Let\ldots \tau_i = < R_{i+T}x_T^* - M_i x >^2\ and, \alpha_i = < x|M_i|x >$$

$$\partial_{<x|}L = M_i|x > . \tau - 2\alpha . M_i^T | R_{i+T}x_T^* - M_i x > = \lambda|1 > + \zeta|\mu_{i+T} >$$

$$|x > = (\tau . M_i + 2M_i^T M_i . \alpha)^{-1} . (2\alpha . M_i^T R_{i+T}|x_T^* > + \lambda|1 > + \zeta|\mu_{i+T} >)$$

$$Let\ (\tau . M_i + 2M_i^T M_i . \alpha)^{-1} = \Psi$$

$$< 1 | x > = < 1 | \Psi . (2\alpha . M_i^T R_{i+T} | x_T^* > + \lambda | 1 > + \zeta | \mu_{i+T} >) = 1$$

$$\lambda = \frac{1 - (< 1 | \Psi . 2\alpha . M_i^T R_{i+T} | x_T^* > + \zeta < 1 | \Psi | \mu_{i+T} >)}{< 1 | \Psi | 1 >}$$

Or

$$\lambda < 1 | \Psi . | 1 > + \zeta < 1 | \Psi | \mu_{i+T} >) = 1 - < 1 | \Psi . (2\alpha . M_i^T R_{i+T}) | x_T^* >$$

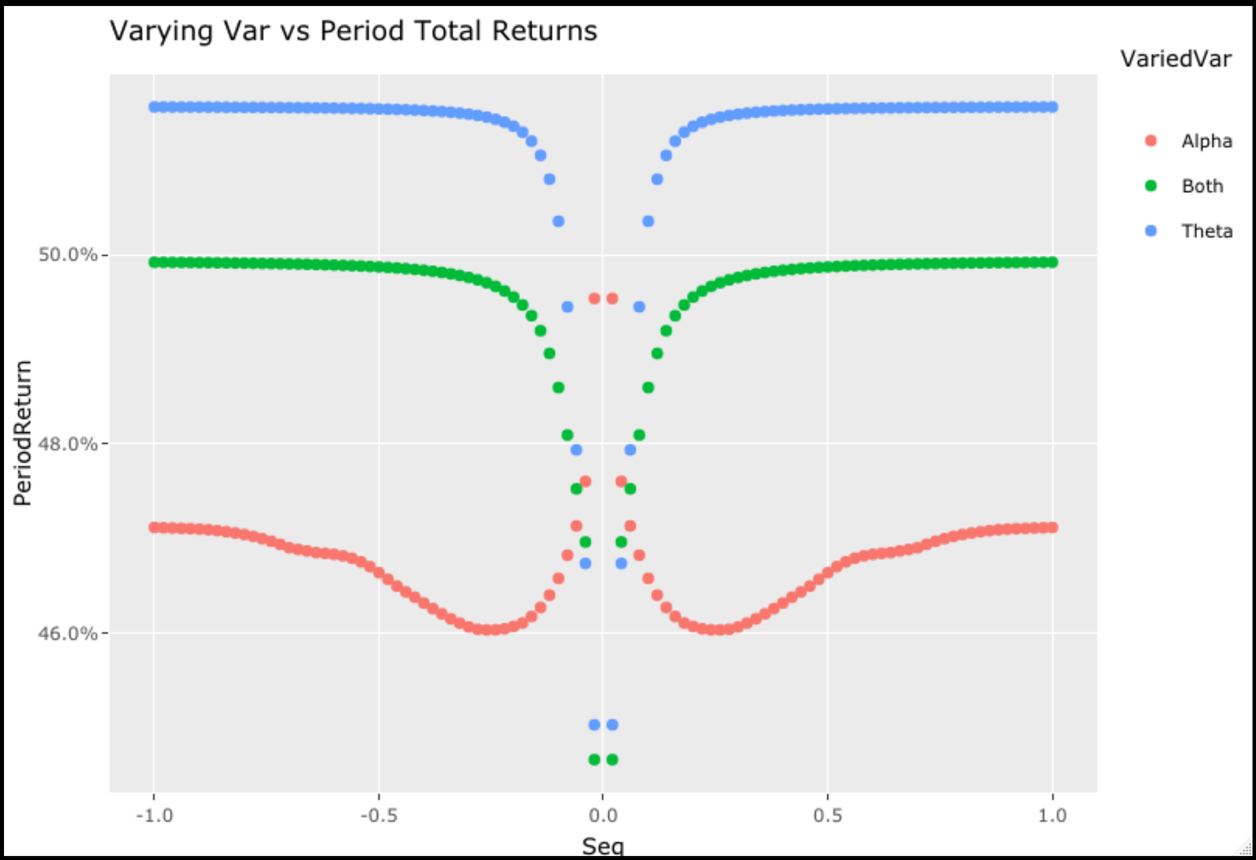
$$\lambda < \mu_{i+T} | \Psi . | 1 > + \zeta < \mu_{i+T} | \Psi | \mu_{i+T} >) = \mu^* - < \mu_{i+T} | \Psi . (2\alpha . M_i^T R_{i+T}) | x_T^* >$$

$$\begin{pmatrix} < 1 | \Psi | 1 > & < 1 | \Psi | \mu_{i+T} > \\ < \mu_{i+T} | \Psi | 1 > & < \mu_{i+T} | \Psi | \mu_{i+T} > \end{pmatrix} . \begin{pmatrix} \lambda \\ \zeta \end{pmatrix} = \begin{pmatrix} 1 - < 1 | \Psi (2\alpha . M_i^T R_{i+T}) | x_T^* > \\ \mu^* - < \mu_{i+T} | \Psi (2\alpha . M_i^T R_{i+T}) | x_T^* > \end{pmatrix}$$

$$\begin{pmatrix} \lambda \\ \zeta \end{pmatrix} = \begin{pmatrix} < 1 | \Psi | 1 > & < 1 | \Psi | \mu_{i+T} > \\ < \mu_{i+T} | \Psi | 1 > & < \mu_{i+T} | \Psi | \mu_{i+T} > \end{pmatrix}^{-1} \begin{pmatrix} 1 - < 1 | \Psi (2\alpha . M_i^T R_{i+T}) | 1 > \\ \mu^* - < \mu_{i+T} | \Psi (2\alpha . M_i^T R_{i+T}) | 1 > \end{pmatrix}$$

Varying Alpha and Theta

By varying alpha and theta, the returns from testing appear to have conflicting behaviours but with a common trait of explosive change close to zero.



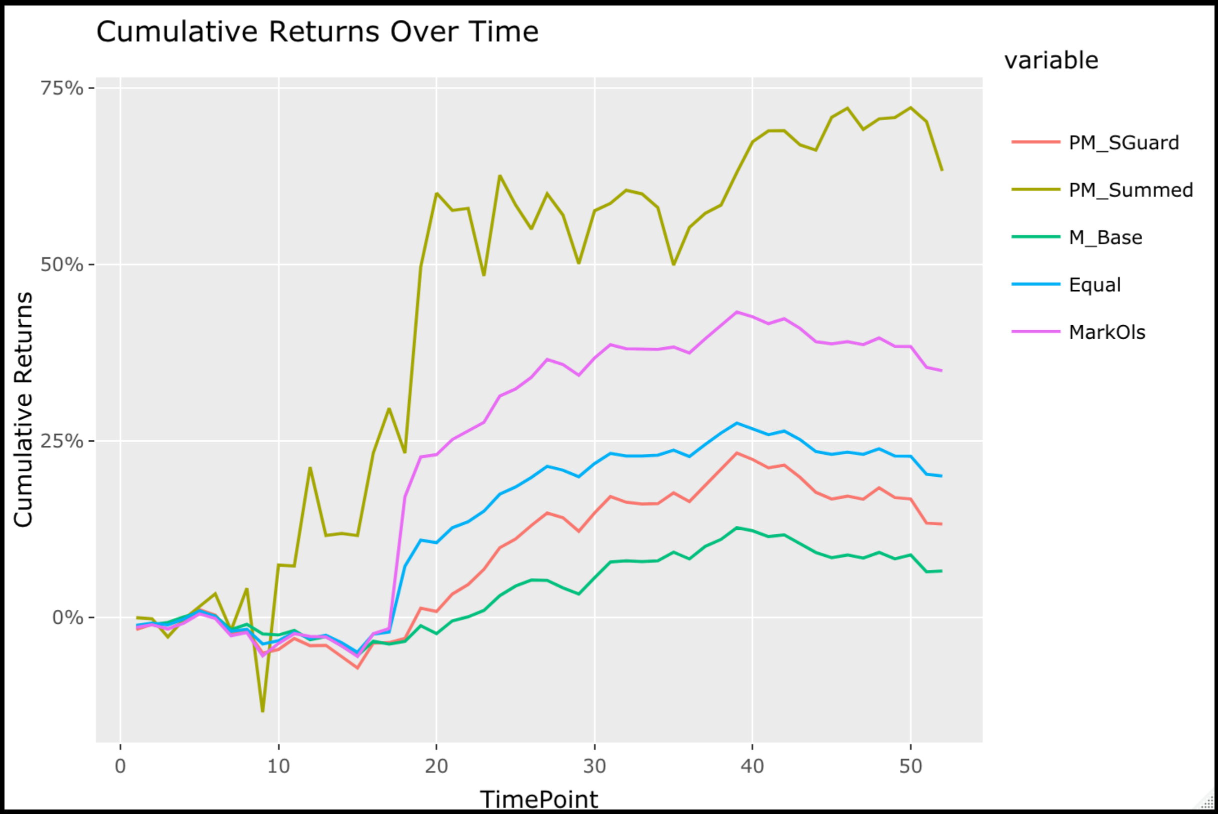
Note: *for convenience, the alpha and tau were inputed as squarable, allow us visualise mirror impact of negative options of alpha and tilda/tau.*
From graphs, looks like very high theta and very low alphas may be the best option.

Further analysis shows that the direction of the squeeze effect is **not** guaranteed. Meaning with varying alpha -for example- the increase of period returns as it approaches zero was some times a decrease while approaching zero.

Test vs other Allocation Algos

Comparing this method against the base Markowitz and equal weight portfolio.

Graphs shows Portfolio algorithms with inbuilt foresight outperforms base and equal weight diversification algorithms (with the exception of PM with Shock



guards).

Note:
PM summed uses past return values rather than past covariance matrix as input to the OLS portion of the Lagrangian. As one can imagine, what is used as input can be altered to fit situational needs. Likewise the Markowitz portion can be altered as well to minimise other things (like the inverse Sharpe, drawdowns, etc).

Lagrangian for PM Summed

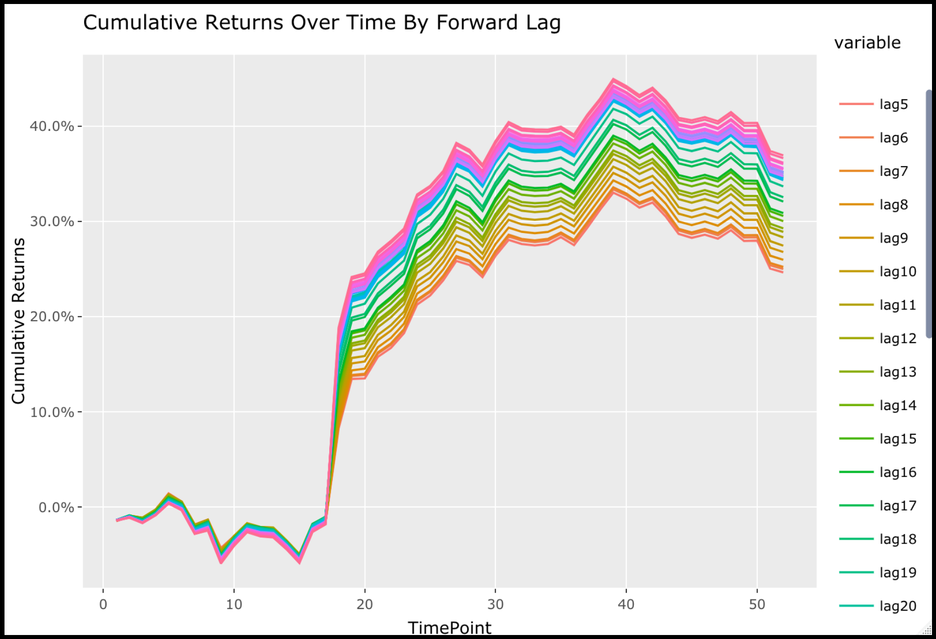
$$\sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T} - R_{i-T}x | R_{i+T} - R_{i-T}x >$$

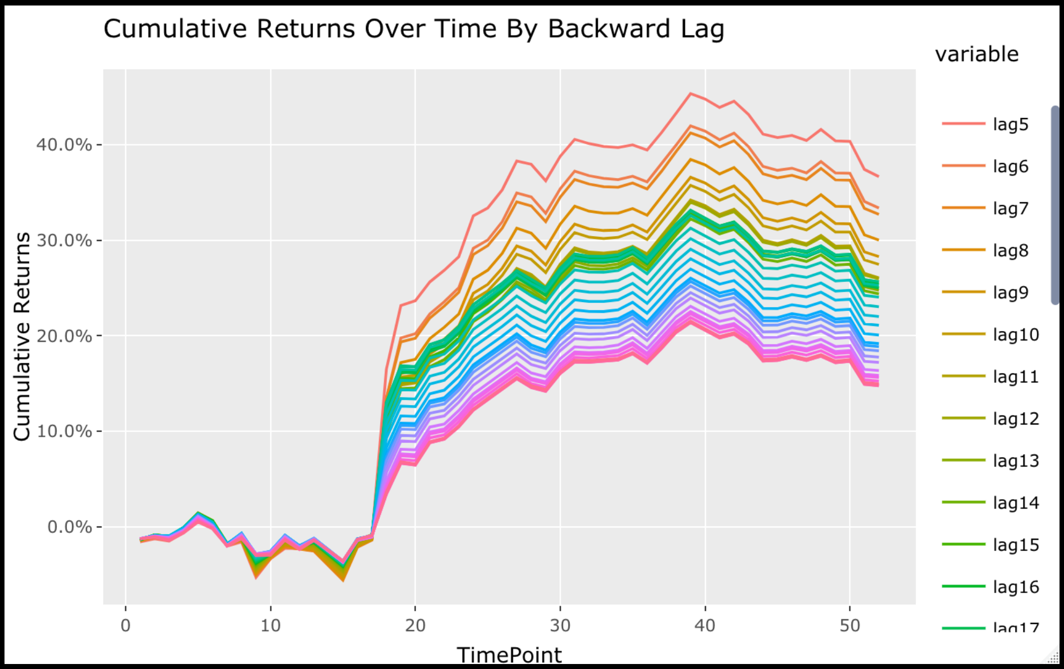
Lagrangian for PM SGuard

$$\sum_{i=t_0-r}^{t_0} < x|M_i|x > . < R_{i+T} - R_{i-T}x | R_{i+T} - R_{i-T}x > . Cov(M_i|x > , R_m i | d_m >)^2$$

Varying the forward and backward lag

Varying Forward Lag
Setting back lag to 10 and varying forward lag from 5 to 35, the graph below shows the potential effect lags have on cumulative returns from the test data set.





Higher forward lag in this case tends to yield higher cumulative returns.

Varying Backward Lag

Setting forward lag fixed to 10 and varying the backward lag from 5 to 35, the graph below shows the returns tend to increase with smaller look back lags

Takeaways from Varying Lags

This indicates -at least in this portfolio and test set- larger forward looking and smaller backward lag yields better performance. It also shows perhaps the reactive nature of Markowitz is indeed a problem, which is why larger forward looking windows tend to yield better performance in this test set.

Future Work

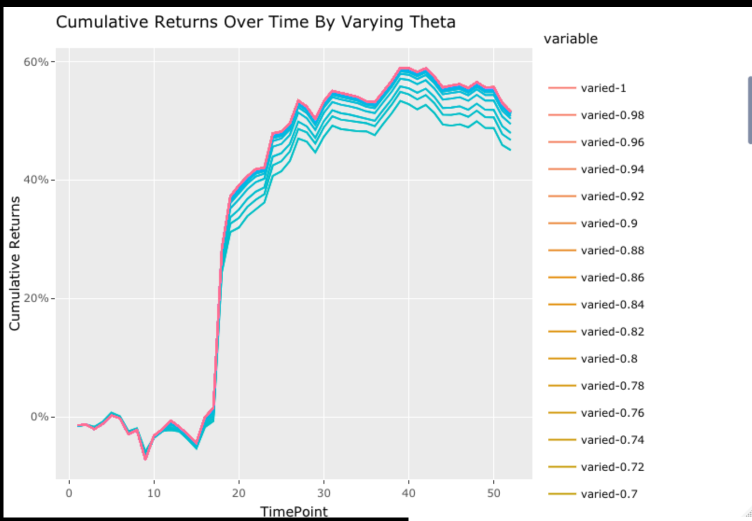
Recursive outlook either within the OLS, with the Markowitz
If the predictive Markowitz calls itself as the future optimal portfolio, then that future portfolio will need its own optimal future portfolio. And so on and so forth, now it is a matter of how much times can one do this recursive call before seeing improvement in performance (if any, at all). But this definitely worth testing.

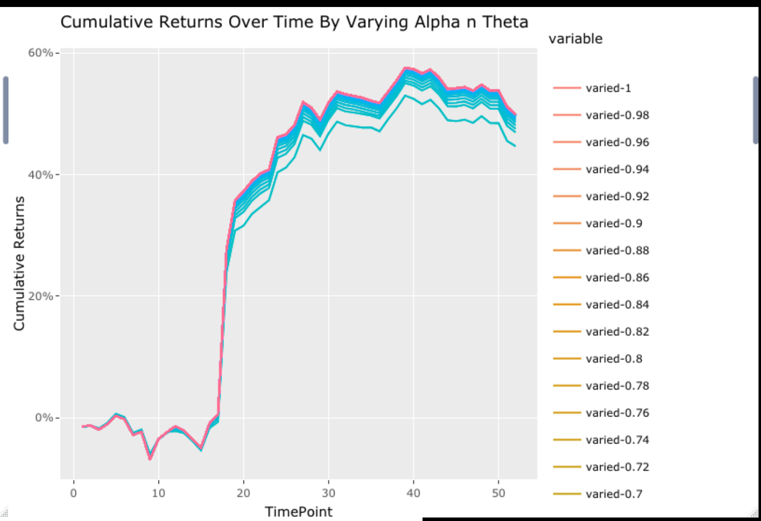
Using alternative or Increasing additional product terms
By increasing the number of product terms in the Lagrangian, it is possible to solve for more behaviours simultaneously. Now it is not clear how this will affect performance -if at all, put could be useful to explore as well.

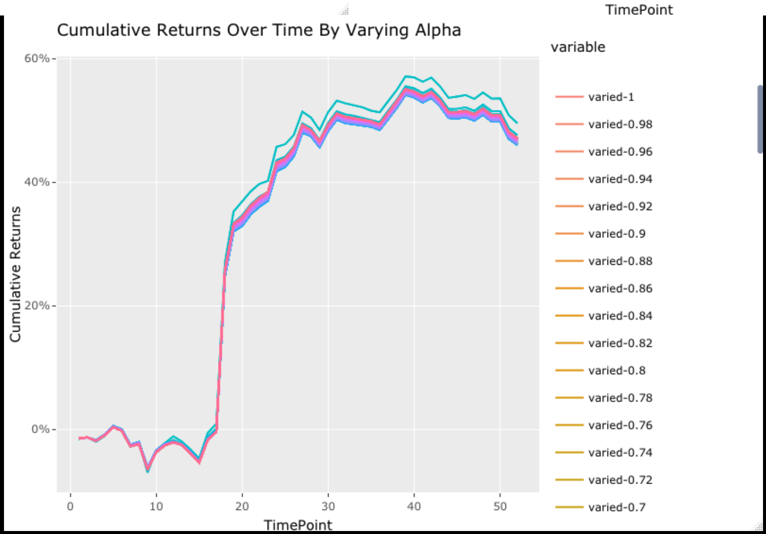
Solving for optimal inputs
Have to iterate through multiple loops to find the back lag, forward lag, alpha and or theta with best performance can be time consuming. Therefore, it is more ideal if these have some theoretical optimal number. This will require addition equation such as the second derivative of the Lagrangian and etc.

Extras and Supplements

Varying Both Alpha and Theta







List of stocks used:
This can be found in the google sheet here *https://docs.google.com/spreadsheets/d/139_mm7v9RCOvQH6A_LAcJ_zF5qJf-sh_Vih19bxfwmg/edit#gid=0*

Time period

- Start: 2022-06-16
- End: 2023-02-22
- Test: 2022-12-07 to end