

Extend the ideas of Kan and Zhou paper  
on Optimal Portfolio Construction under  
parameter uncertainty



George Photiou  
Lincoln College  
University of Oxford

A dissertation submitted in partial fulfilment for the degree of  
*Master of Science in Mathematical and Computational Finance*

June 25, 2010

## **Acknowledgements**

Great appreciation to my supervisor, Stefan Sluke, for his professional guidance and support throughout the duration of this thesis. I would also like to express my gratitude to Man Group Plc portfolio management team for giving me the opportunity to work on this project and for their great interest and valuable suggestions.

## **Abstract**

In this dissertation, we extend the ideas of Raymond Kan and Guofu Zhou for optimal portfolio construction under parameter uncertainty. Kan and Zhou proved analytically that under parameter uncertainty, investing in the sample tangency portfolio and the riskless is not optimal. Based on this idea we will approach the portfolio construction under parameter uncertainty in a different way. We will optimise the expected out-of-sample performance of a portfolio using a numerical approach. Using Monte Carlo simulations we will develop an algorithm that calculates the expected out-of-sample performance of any portfolio rule. We will then extend this algorithm in order to be able to input new portfolio rules and test their performance. The new portfolio rules we introduce are based on shrinkages for the mean and covariance matrix of the assets returns. These shrinkages will have some parameters that will be chosen so that we optimise the expected out-of-sample performance of the input portfolio rule. A comparison is then done between the portfolio rules we introduce and Kan and Zhou portfolio rules.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Estimation risk and Kan-Zhou ideas</b>	<b>5</b>
2.1	Main results . . . . .	5
2.2	Four portfolio rules . . . . .	7
2.2.1	Plug-in rule . . . . .	8
2.2.2	Kan-Zhou two-fund rule . . . . .	9
2.2.3	Kan-Zhou three-fund rule . . . . .	10
2.2.4	1/N rule . . . . .	11
2.3	Comparison of the rules . . . . .	12
<b>3</b>	<b>Numerical algorithm for testing portfolio performance</b>	<b>13</b>
3.1	General algorithm . . . . .	13
3.2	Monte Carlo algorithm with quadratic utility . . . . .	14
3.3	Optimise the expected out-of-sample performance under parameters .	17
<b>4</b>	<b>Shrinkage for the parameters</b>	<b>20</b>
4.1	Motivation . . . . .	20
4.2	Mean shrinkage . . . . .	21
4.3	Covariance Matrix shrinkage . . . . .	24
4.4	Mean and Covariance matrix shrinkage . . . . .	28
4.5	Correlation Smoothing . . . . .	29
<b>5</b>	<b>Numerical portfolio rules and comparisons</b>	<b>32</b>
5.1	Numerical portfolio rules . . . . .	32
5.2	Comparison of the rules . . . . .	34
<b>6</b>	<b>Conclusion</b>	<b>38</b>

# Chapter 1

## Introduction

A mean-variance investor is an investor that measures the portfolio performance from its mean return and variance. A mean-variance investor always faces the problem of how to allocate his wealth between the assets in order to have an optimal position in the market. In other words he wants to allocate his investments between the assets in such a way such that for the risk he is taking he will have the best possible expected return. Markowitz (1952) proved that the optimal investment strategy for a mean-variance investor is to invest in the riskless asset and the sample tangency portfolio. This is defined as a two fund rule as we invest in two portfolios, the riskless asset and the sample tangency portfolio. Markowitz's model is optimal if we know the actual mean and covariance matrix of the returns of the assets. However, in reality the mean and covariance matrix of the assets returns are unknown.

One approach to this problem is to treat the estimated mean and covariance matrix of the assets as if they were the true parameters and then plug them in the Markowitz's model. This rule is known as the classical plug-in rule. Several studies have shown that the plug-in rule is not optimal. Brown (1976) proved that the Bayesian decision rule outperforms the plug-in rule. Kan and Zhou derive analytically the above result and show, by calculating the expected out-of-sample performance of the two rules, that the Bayesian rule is always better regardless of what the true parameters are. By adding a scale parameter to the Markowitz optimal weight function Kan and Zhou manage to find a two fund rule that outperforms any other two fund rule. The idea is to choose this parameter in order to maximise the expected out-of-sample performance.

Later on Kan and Zhou showed that when we work under parameter uncertainty a

three fund rule can actually perform better than a two fund rule. This is because by investing in another risky portfolio an investor can diversify the estimation error. As a third portfolio to invest they chose the global minimum variance portfolio which has less estimation error as this portfolio does not include the mean parameter.

In this dissertation we are dealing with the problem of optimising the expected out-of-sample performance under parameter uncertainty with a different approach. Rather than deriving the expected out-of-sample performance of a portfolio rule analytically, we develop an algorithm that finds the expected out-of-sample performance for any portfolio rule by using Monte Carlo simulations. We extend this algorithm later on in order to input new portfolio rules which are based on shrinkage estimators for the mean and covariance matrix. These shrinkage estimators will have some parameters that will be chosen so that the expected out-of-sample performance of the input portfolio rule is optimised.

We choose the shrinkage estimators based on the idea that the less information we have for the true mean and covariance matrix of the assets returns the less trust we give to our estimates for them. Hence if we have less information about the true assets return distribution we would like to have a portfolio rule that takes into account this uncertainty and allocates the wealth by taking all the available information. On the other hand if we have enough evidence for the assets return distribution then our shrinkage estimators should give a weight allocation which will converge to the Markowitz one. This is because as we already mention if we know the exact distribution then the optimal strategy is to follow Markowitz's model. The shrinkages estimators will be based on the above idea. At the end of this project we will do a comparison of the shrinkage rules we introduce with Kan-Zhou 2 and 3 fund portfolio rules.

This project will be organised as follows. The second chapter will give a better understanding of the estimation error and the derivation of the expected out-of-sample performance of a portfolio rule. In this chapter we will recall some important results and formulas from Kan and Zhou paper. We will recall the formulas that Kan and Zhou analytically derived on their paper for the plug-in rule, the two fund rule and the three fund rule as we will need these rules through this project. In the third chapter we introduce a general algorithm that finds the expected out-of-sample performance of any portfolio rule and then we extend this in order to be able to optimise this algorithm with respect to some parameters. In the fourth chapter we

introduce various shrinkages for the mean and covariance matrix and we will show analytically the improvement of the expected out-of-sample performance we have by using these shrinkages. In chapter 5 we introduce two models by combining our findings from chapter 4 and then we make comparisons of all the rules for various situations. In chapter 6 we conclude.

# Chapter 2

## Estimation risk and Kan-Zhou ideas

### 2.1 Main results

In this chapter we are setting the scene in order to introduce and explain analytically the ideas of Kan and Zhou for optimal portfolio construction with parameter uncertainty.

A mean-variance investor, as we already mentioned, faces the problem of allocating his investments between the assets in such a way that for the risk he is taking he will have the best possible expected return. In this paper the risk refers to the variance of the investment and the return refers to the mean of the investment. We suppose that our investment universe is the riskless asset  $S_0$  and the  $N$  risky assets  $S_1, \dots, S_N$  with corresponding returns  $r_0, \dots, r_N$ . Hence the investor faces the problem of choosing the weights he will give to the corresponding assets. Let the weight vector be  $w = (w_0, w_1, \dots, w_N)$ . Notice that  $w_0 + w_1 + \dots + w_N = 1$ .

If the riskless asset return is  $r_0$  then the excess return of the other  $N$  assets to the riskless asset will be  $R_i = r_i - r_0$  for  $i = 1, \dots, N$  and we define the excess return vector as  $R = (R_1, \dots, R_N)$ . Since the excess return of the risk free asset is zero, for simplicity we choose not to include it in the vector  $R$ . Under Markowitz mean-variance analysis, in order to measure a portfolio performance we only need the mean and the variance of a portfolio. In this project we make the assumption that the assets excess returns follow a multivariate normal distribution such that  $R_i \sim N(\mu_i, \sigma_i^2)$  for  $i = 1, \dots, N$ . The overall excess return of a portfolio will then be  $R_p = w'R$ . The expected return of the asset  $i$  is  $\mu_i = E(R_i)$  and the corresponding

expected return vector  $\mu = (\mu_1, \dots, \mu_N)$ . The mean and variance of the portfolio will then be  $\mu_p = w'\mu$  and  $\sigma_p^2 = w'\Sigma w$  respectively where  $\Sigma$  is the covariance matrix of the assets. We assume that the investor's utility function is quadratic as in Kan and Zhou paper so that:

$$U(w) = \mu_p - \frac{\gamma}{2}\sigma_p^2 \quad (2.1)$$

where  $\gamma$  is the risk averse coefficient for an investor.

We can see that the investor's problem is to choose the weights  $w_1, \dots, w_N$  so that he maximises his expected utility function  $E[U(w)]$ . Markowitz solved this optimisation problem and proved that the optimal weights for an investor are:

$$w^* = \frac{1}{\gamma}\Sigma^{-1}\mu \quad (2.2)$$

which give the optimal utility function:

$$U(w^*) = \frac{1}{2\gamma}\mu'\Sigma^{-1}\mu = \frac{\theta^2}{2\gamma} \quad (2.3)$$

where  $\theta^2 = \mu'\Sigma^{-1}\mu$  is the squared Sharpe ratio. Now to make things more clear assume that we have a portfolio with two assets  $S_1, S_2$  and the riskless asset  $S_0$ . The expected return vector of the risky assets is  $\mu = \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix}$  and covariance matrix

$\Sigma = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}$ . Assume also that the investor has risk averse coefficient  $\gamma = 3$ .

We then substitute  $\mu$ ,  $\Sigma$  and  $\gamma$  in the Markowitz optimal weight formula (2.2):

$$w^* = \frac{1}{\gamma}\Sigma^{-1}\mu = \frac{1}{3} \frac{1}{0.8} \begin{pmatrix} 1 & -0.4 \\ -0.4 & 1 \end{pmatrix} \begin{pmatrix} 0.3 \\ 0.7 \end{pmatrix} = \begin{pmatrix} 0.0079 \\ 0.2302 \end{pmatrix}$$

We then normalise the above weight vector so that we can observe the proportion of wealth allocated to the assets. We get the weight vector  $w^* = (0.0332, 0.9668)$ . So the optimal strategy of the investor would be to invest 0.0332 of his wealth in asset  $S_1$  and 0.9669 in asset  $S_2$ .

The problem with Markowitz formulas is that in order to get optimal returns we need to know the exact distribution of the mean and covariance matrix of the asset returns. In reality these two parameters are unknown and so we can only find some estimates for them by using the available information we have. The standard approach is to plug-in our estimates in the Markowitz's formulas. This approach as Kan and Zhou proved analytically in their paper is not optimal. There are different

strategies that yield higher expected out-of-sample performance than the plug-in approach.

One way to find estimates for  $\mu_p$  and  $\Sigma_p$  will be to assume that the historical returns of the assets are normally distributed. We will denote the estimated mean and variance as  $\tilde{\mu}$  and  $\tilde{\sigma}^2$  respectively. In order to find these estimates we use the past returns of one asset for  $T$  periods  $R_1, \dots, R_T$ . Then the weight vector  $\hat{w}$  will be a function of these historical returns such that  $\hat{w} = f(R_1, \dots, R_T)$ . The overall portfolio's mean and variance will be  $\tilde{\mu}_p = \hat{w}'\mu$  and  $\tilde{\sigma}_p^2 = \hat{w}'\Sigma\hat{w}$  where  $\hat{w}$  is the estimated weight vector and  $\mu$  and  $\Sigma$  the true mean and covariance matrix of the assets. Hence by substituting  $\tilde{\mu}_p$  and  $\tilde{\sigma}_p^2$  in (2.2) the out-of-sample utility function will be:

$$\tilde{U}(\hat{w}) = \tilde{\mu}_p - \frac{\gamma}{2}\tilde{\sigma}_p^2 = \hat{w}'\mu - \frac{\gamma}{2}\hat{w}'\Sigma\hat{w} \quad (2.4)$$

There are many ways to measure the out-of-sample performance of a portfolio. One popular way is by using the Sharpe ratio. The Sharpe ratio is defined as the excess return over the risk of a portfolio such that  $\theta = \frac{\tilde{\mu}_p}{\tilde{\sigma}_p}$ . In this paper we will measure the portfolio performance in the same way as Khan and Zhou measured it. We will calculate the expected out-of-sample utility function and compare it with the optimal Markowitz's utility function (2.3). The difference between the two is defined as the expected loss function. The loss function is defined as:

$$L(w^*, \hat{w}) = U(w^*) - \tilde{U}(\hat{w}) \quad (2.5)$$

and by taking expectations in both sides we then have the expected loss function:

$$\rho(w^*, \hat{w}) = E[L(w^*, \hat{w})] = U(w^*) - E[\tilde{U}(\hat{w})] \quad (2.6)$$

## 2.2 Four portfolio rules

In this section we explain the derivation of four portfolio rules that are essential in this project as later on we will be comparing these with some new portfolio rules we will introduce. The four portfolio rules we choose to deal with are:

1. The classical plug-in rule
2. Kan-Zhou 2-fund rule
3. Kan-Zhou 3-fund rule
4. The 1/N rule

The 1/N rule is the rule that gives equal weight to all the  $N$  assets in our portfolio. We will not explain in further detail the calculations involved in each rule in order to find the expected out-of-sample performance as they are beyond the scope of the project. The analytical calculations can be found in Kan and Zhou paper. Our scope here, is to give a better understanding of the estimation error, the expected out-of-sample performance and the expected loss function of a portfolio rule.

As we have already mentioned, in this project we assume that the past returns  $R_1, \dots, R_T$  follow an i.i.d normal distribution so that the sample mean  $\hat{\mu}$  and the covariance matrix  $\hat{\Sigma}$  are given by:

$$\hat{\mu} = \frac{1}{T} \sum_{i=1}^T R_i \quad (2.7)$$

$$\hat{\Sigma} = \frac{1}{T} \sum_{i=1}^T (R_i - \hat{\mu})(R_i - \hat{\mu})' \quad (2.8)$$

and hence by doing the calculations we find that  $\hat{\mu}$  will be normally distributed with mean  $\mu$  and variance  $\frac{\Sigma}{T}$  such that:

$$\hat{\mu} \sim N\left(\mu, \frac{\Sigma}{T}\right) \quad (2.9)$$

$\hat{\Sigma}$  will then follow a Wishart distribution:

$$\hat{\Sigma} \sim W_N(T - 1, \Sigma)/T \quad (2.10)$$

where  $W_N(T - 1, \Sigma)$  denotes the Wishart distribution with T-1 degrees of freedom and covariance matrix  $\Sigma$ .

### 2.2.1 Plug-in rule

The plug-in rule is the standard approach to the problem of optimising the portfolio performance under parameter uncertainty. In this rule we treat the estimated mean  $\hat{\mu}$  and the estimated covariance matrix  $\hat{\Sigma}$ , (2.9) and (2.10) respectively, as they were the true parameters. Thus we plug them in Markowitz's formulas (2.1) and (2.2) and obtain :

$$\hat{w} = \frac{1}{\gamma} \hat{\Sigma}^{-1} \hat{\mu} \quad (2.11)$$

$$\tilde{U}(\hat{w}) = \hat{\mu}_p - \frac{\gamma}{2} \hat{\sigma}_p^2 = \hat{w}' \mu - \frac{\gamma}{2} \hat{w}' \Sigma \hat{w} \quad (2.12)$$

Then if we calculate the  $E[\tilde{U}(\hat{w})]$  we find that the resulting expected out-of-sample performance of the classic plug in rule is:

$$E[\tilde{U}(\hat{w})] = k_1 \frac{\theta^2}{2\gamma} - \frac{NT(T-2)}{2\gamma(T-N-1)(T-N-2)(T-N-4)} \quad (2.13)$$

The analytic calculations can be found in section *II.B* of Kan Zhou paper. Then substituting the above result into the formula of the expected loss function (2.6) we get:

$$\rho(w^*, \hat{w}) = (1 - k_1) \frac{\theta^2}{2\gamma} + \frac{NT(T-2)}{2\gamma(T-N-1)(T-N-2)(T-N-4)} \quad (2.14)$$

assuming  $T > N + 4$  and where  $k_1 = (\frac{T}{T-N-2}[2 - \frac{T(T-2)}{(T-N-1)(T-N-4)}])$ . The result (2.13) will be useful in the next chapter where we introduce an algorithm that finds the expected out-of-sample performance of a portfolio rule. We will run our algorithm with input the plug-in rule and compare our results with (2.13) in order to check that the developed algorithm is working properly.

## 2.2.2 Kan-Zhou two-fund rule

Kan and Zhou's idea for the 2 fund rule is to add a scale  $c$  parameter to the weight function (2.11). This parameter depends on the the number of assets  $N$  and the number of observations  $T$  and is chosen such that it optimises the expected out-of-sample utility function. The weight function will then be:

$$\hat{w} = \frac{c}{\gamma} \hat{\Sigma}^{-1} \hat{\mu} \quad (2.15)$$

and by substituting it in the utility function formula (2.1) it yields that the expected out-of-sample performance of the two fund rule is:

$$E[\tilde{U}(\frac{c}{\gamma} \hat{\Sigma}^{-1} \hat{\mu})] = \frac{c\theta^2}{\gamma} (\frac{T}{T-N-2}) - \frac{c^2}{2\gamma} (\theta^2 + \frac{N}{T}) [\frac{T^2(T-2)}{(T-N-1)(T-N-2)(T-N-4)}] \quad (2.16)$$

under the assumption that  $T > N + 4$ . By differentiating with respect to the scale  $c$  we can find the optimal  $c$  that maximises the expected out-of-sample utility function. The required scale  $c$  is given by:

$$c^* = [\frac{(T-N-1)(T-N-4)}{T(T-2)}] (\frac{\theta^2}{\theta^2 + \frac{N}{T}}) \quad (2.17)$$

Finally by substituting  $c^*$  into (2.16) we find that the optimal expected out-of-sample utility function of Kan and Zhou two fund rule is:

$$E[\tilde{U}(\hat{w}^*)] = \frac{\theta^2}{2\gamma} \left[ \frac{(T-N-1)(T-N-4)}{(T-2)(T-N-2)} \right] \left( \frac{\theta^2}{\theta^2 + \frac{N}{T}} \right) \quad (2.18)$$

We conclude that Kan and Zhou 2 fund rule is the best possible rule used when we are investing in the riskless asset and the sample tangency portfolio under parameter uncertainty. We will be using this result in the next chapter in order to check that our algorithm with a weight function shrinkage as (2.15) is working properly. The only other way to achieve a better out-of-sample performance of the 2 fund portfolio rule under parameter uncertainty is to invest in other portfolios as well because then the estimation error diversifies.

### 2.2.3 Kan-Zhou three-fund rule

The reason why a 3 fund rule performs better than the optimal 2 fund rule lies on the fact that by investing in another risky portfolio the estimation error becomes diversified. The third portfolio Kan and Zhou chose to invest in, is the global minimum variance portfolio  $\hat{\Sigma}^{-1}\mathbf{1}_N$  because it does not include the mean estimate  $\hat{\mu}$  and hence the estimation error is less. The weight function will then be given by:

$$\hat{w} = \frac{1}{\gamma} (c\hat{\Sigma}^{-1}\hat{\mu} + d\hat{\Sigma}^{-1}\mathbf{1}_N) \quad (2.19)$$

where  $c$  and  $d$  are scales chosen in order to optimise the expected out-of-sample utility function. The idea of the parameters  $c$  and  $d$  is similar to the choice of scale  $c$  of Kan and Zhou 2 fund rule. Following the same procedure as before we first find a formula for the expected out-of-sample utility function  $E[\tilde{U}(\hat{w}(c, d))]$ , then differentiate this with respect to  $c$  and  $d$  and then solve to find the optimal parameters  $c^*$  and  $d^*$ . We substitute them back in our expected out-of-sample utility function to find that the expected out-of-sample utility function of Kan and Zhou 3-fund portfolio rule is:

$$E[\tilde{U}(\hat{w}^{**})] = \frac{\theta^2}{2\gamma} \frac{(T-N-1)(T-N-4)}{(T-2)(T-N-2)} \left[ 1 - \frac{\frac{N}{T}}{\theta^2 + \left(\frac{\theta^2}{\psi^2}\right)\left(\frac{N}{T}\right)} \right] \quad (2.20)$$

under the assumption that  $T > N+4$  and where  $\psi^2 = \mu'\Sigma^{-1}\mu - \frac{(\mu'\Sigma^{-1}\mathbf{1}_N)^2}{\mathbf{1}'_N\Sigma^{-1}\mathbf{1}_N}$ . However this expected out-of-sample performance is unachievable if we are out-of-sample since  $\psi$  is unknown. We can only estimate  $\psi$ . Later on when we introduce the algorithm that is able to implement any portfolio rule, we will be able to implement Kan-Zhou 3 fund rule as well.

### 2.2.4 1/N rule

The 1/N rule is the rule that assigns to all the N assets an equal weight. The weight vector will then be  $w = (1/N, 1/N, \dots, 1/N)$ . There were some claims raised which suggested that when we are working under parameter uncertainty it is optimal to apply the 1/N rule instead of any other rule. Mark Kritzman (2010) in the paper 'In Defence of Optimization: The Fallacy of 1/N' proves the opposite of this claim. He proves that is always better to use an optimisation rule rather than give all the assets in our portfolio an equal weight. In this project we will see that even though Kritzman's result is true we observe that the less the information we have for the assets returns distribution the more our result converges to the 1/N portfolio rule. We define the weight vector as  $w = \frac{1}{N}\mathbf{1}_N$ . The expected out-of-sample performance of the 1/N rule can easily be derived as:

$$E[\tilde{U}(\hat{w})] = E[\hat{w}'\mu - \frac{\gamma}{2}\hat{w}'\Sigma\hat{w}] = \frac{1}{N}\mathbf{1}'_N\mu - \frac{\gamma}{2}\frac{1}{N}\mathbf{1}'_N\Sigma\frac{1}{N}\mathbf{1}_N \quad (2.21)$$

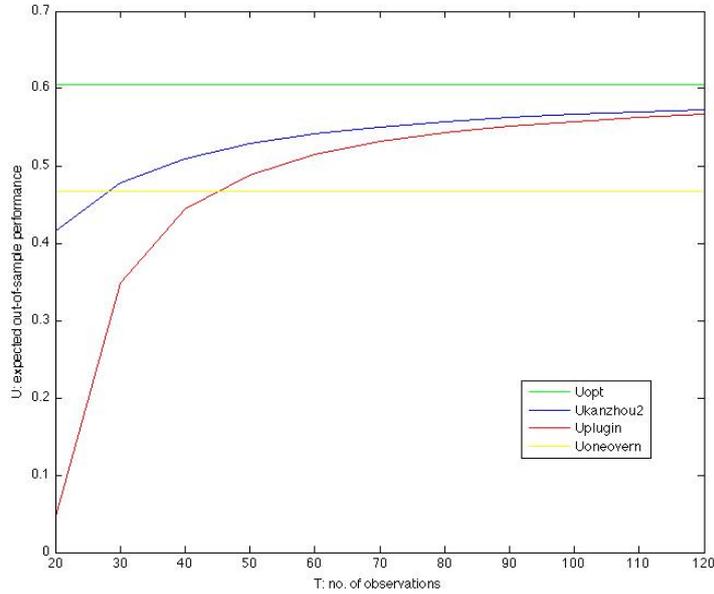
Note that the investor can choose any weight vector multiple of  $\mathbf{1}_N = (1, 1, 1, \dots, 1)$ . The investor can choose any c such that  $c\mathbf{1}_N = (c, c, c, \dots, c)$ . Then the equation for the 1/N rule will be:

$$E[\tilde{U}(\hat{w})] = E[\hat{w}'\mu - \frac{\gamma}{2}\hat{w}'\Sigma\hat{w}] = \frac{c}{N}\mathbf{1}'_N\mu - \frac{\gamma}{2}\frac{c}{N}\mathbf{1}'_N\Sigma\frac{c}{N}\mathbf{1}_N \quad (2.22)$$

## 2.3 Comparison of the rules

An easy way to summarise all the above is to draw a graph with all the rules of section 2.2. We will draw the expected out-of-sample utility of each rule with respect to the number of observations. As we already mention at this point we can't draw the curve of Kan-Zhou 3 fund rule as it is not implementable. We assume a portfolio with three assets with asset return vector  $\mu = (0.2, 1, 1.1)$  and covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 0.1 & 0.2 \\ 0.1 & 1 & 0.9 \\ 0.2 & 0.9 & 1 \end{pmatrix}.$$



In the above figure we can see that for  $T=20$  the  $1/N$  rule outperforms both Kan-Zhou 2 fund rule and plug-in rule. This is because we don't have enough information for the assets return distribution so if we split up our investment equally in the three assets performs better than following the two rules. This is not true always. It depends on the mean vector and on the covariance matrix of the assets and not only on the number of observations  $T$ . However for a very small  $T$   $1/N$  rule usually outperforms the other rules. We can also comment that Kan-Zhou 2 fund rule uniformly outperforms the plug-in rule. Furthermore as  $T$  increase both rules are converging to the optimal performance as we are more certain of the distribution of the assets returns and also they are diverging from the  $1/N$  portfolio rule.

# Chapter 3

## Numerical algorithm for testing portfolio performance

### 3.1 General algorithm

In this chapter we introduce a numerical approach in order to measure the performance of any portfolio rule. In chapter 2 we measure the performance of the portfolio rules by mathematically deriving the formulas for the expected out-of-sample performance and then comparing them to the optimal performance given by Markowitz's formula (2.3). The difference between the two is the expected loss function. We will develop an algorithm that calculates the expected out-of-sample performance given the portfolio rule as input.

First we consider a general case for the algorithm. The algorithm must find some estimates for the mean and covariance matrix of the assets returns. For example one way is to assume that the mean and covariance matrix of the asset returns are normally distributed as we did in chapter 2. Then these estimates are plugged-in a weight function of the form  $w = f(\mu, \sigma)$ . Then we plug this in the investors desired utility function. The utility function depends on the target the investor sets. We have exponential utility, power utility, quadratic utility, logarithmic utility and others. In Markowitz's framework we assume that the utility function of the investor is quadratic. We loop all the above steps and for each iteration we obtain a different utility function as  $\mu$  and  $\sigma$  are different each time. Then we find the average of all the utilities which is the expected out-of-sample performance for the input portfolio rule. The algorithm then is as follows:

1. Introduce the estimate for the mean of the assets returns.

2. Introduce the estimate for the covariance matrix of the assets returns.
3. Plug the estimates of steps 1 and 2 in the weight function.
4. Plug the weight function in the desired utility function.
5. Iterate the steps 1-4, n times and thus find a different utility in each iteration.
6. Find the average of all the utilities.
7. Output the expected out-of-sample performance.

## 3.2 Monte Carlo algorithm with quadratic utility

In this section we develop an algorithm for the Markowitz's framework. Under the Markowitz framework we make the assumptions that:

- The assets returns are normally distributed and fully characterised by their mean and variance.
- The investor's utility function is quadratic such that  $U(w) = \mu_p - \frac{\gamma}{2}\sigma_p^2$  where  $\mu_p$  and  $\sigma_p^2$  the assets returns mean and variance respectively.

In order to find estimates for the assets returns distribution we need to generate a multidimensional normal distribution with dimension equal to the number of the assets  $N$  in our portfolio. We generate the normal distribution using the true mean and covariance of the assets. From this normal distribution we then draw a number of observations  $T$ . So if for example we have information for 100 historical returns then  $T=100$ . Then we take the maximum likelihood estimators for both the mean and covariance matrix of the assets returns.

Later on we will see that we can use these maximum likelihood estimators in order to try different portfolio rules and use numerical optimisers on new parameters. For now let's continue with the simplest rule which is the plug-in rule. We plug in the maximum likelihood estimators for the mean and covariance matrix in the weight function given by (2.11) . Then we substitute the weight function in the quadratic out-of-sample utility formula (2.12). We loop again for all the previous steps as for each iteration we get different maximum likelihood estimators as we draw different observations from the multidimensional normal distribution and hence different utilities. Then we find the average of all the utilities which is the expected out-of-sample performance. So we expect that the larger the number of the simulations  $N_{sim}$  we have a more accurate estimate for the expected out-of-sample utility performance

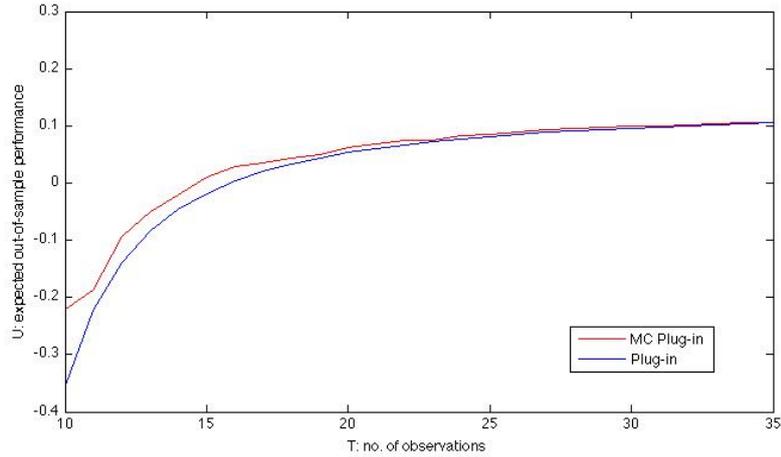
of the input portfolio rule. The algorithm for the above case will then be:

1. Define the number of observations  $T$ , the number of assets  $n$ , the risk averse coefficient  $\gamma$  and number of simulations  $Nsim$ .
2. Define the true mean vector  $\mu$  and correlation matrix  $\Sigma$  of the assets returns.
3. Generate an  $N$  dimensional normal distribution  $N(\mu, \Sigma)$  where  $T$  the number of assets.
4. Draw  $T$  realisations from the normal distribution generated in step 3.
5. Define the maximum likelihood estimator of the mean of the returns as the mean of the  $X$  realisations we draw in step 3 such that:  $\mu_{MLE} = mean(X)$
6. Define the maximum likelihood estimator covariance matrix of the returns as the covariance of the  $X$  realisations we draw in step 3 such that:  $\Sigma_{MLE} = cov(X)$
7. Substitute  $\mu_{MLE}$  and  $\Sigma_{MLE}$  in the weight function (2.11).
8. Substitute the observed weight function from step 6 in (2.12) in order to find the out-of-sample utility function.
9. We loop over steps 3-7  $Nsim$  times and for each iteration we find a different out-of-sample utility function. This is because the realisations we draw in step 3 will be different for each iteration.
10. Find the average of all utility functions.
11. Printout the expected out-of-sample utility function.

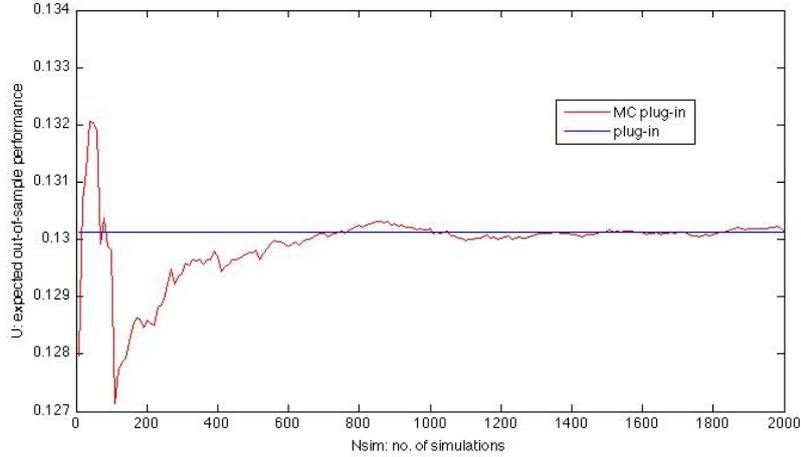
The above algorithm as we said is basically the classical plug-in rule as it plugs the estimated mean and covariance matrix in the weight function (2.11). We are able now to check if the algorithm we develop above is working properly. We can do this by running our algorithm and compare the result with the plug-in formula for the expected out-of-sample performance (2.13).

In our comparison we expect to see some Monte Carlo error when the number of simulations  $Nsim$  or the number of observation  $T$  are small. This is because when  $Nsim$  is small we don't loop enough times over our algorithm so we have a good estimate of the portfolio rule. When  $T$  is small similarly means that we don't get enough observations of the normal distribution in each iteration so again we will have some error on our estimate of the expected out-of-sample performance.  $T$  and  $Nsim$  have an inverse relation. By that we mean that when the number of observations  $T$  is small we need a large number of  $Nsim$  and when  $Nsim$  is small we need a large number of  $T$  in order to have a better estimate of the performance of the portfolio rule. We named this error as Monte Carlo error. The Monte Carlo error hence will tend to zero as  $Nsim$  or  $T$  is increasing.

We can draw some graphs in order to show the relation of  $N_{sim}$  and  $T$  with the Monte Carlo error. We first draw the graph of the expected out-of-sample utility function with respect to number of observations  $T$  and then with respect to the number of simulations  $N_{sim}$ . For both figures we assume a portfolio with two assets with mean of the returns vector  $\mu = (0.5, 0.9)$  and correlation matrix  $\Sigma = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}$



In figure 1 we investigate the convergence of the plug-in rule using Monte Carlo simulations to the exact plug-in rule while keeping the number of simulations  $N_{sim}$  fixed with  $N_{sim} = 1000$ . We can see that as  $T$  increases the Monte Carlo plug-in rule converges to the Plug-in rule as expected. From  $T=5$  to  $T=20$  we observe that the Monte Carlo error is large and so we will need a very large number of simulations in order to minimise this error. For  $T=35$  the two curves almost coincide and so as  $T$  increases the Monte Carlo error tends to zero.



In figure 2 we investigate the convergence of the two curves with respect to the number of simulations  $N_{sim}$ . We keep the number of observations fixed  $T = 100$ . Again as expected the MC Plug-in rule is converging to the Plug-in rule as  $N_{sim}$  is increasing. It is important to notice that we need at least a 1000 number of simulations in order to have a relatively small Monte Carlo error when  $T = 100$ .

From these two figures we can say that by keeping the number of observations  $T$  greater or equal to 100 and the number of simulations equal or more than 2000 the Monte Carlo error is very small and hence our results will be reasonable. Furthermore from the two graphs we conclude that  $T$  and  $N_{sim}$  have to be inversely proportional in order to keep Monte Carlo error small. As the  $N_{sim}$  decreases we need a larger number of observations  $T$  and as the number of observations  $T$  decreases we need a larger number of  $N_{sim}$ .

### 3.3 Optimise the expected out-of-sample performance under parameters

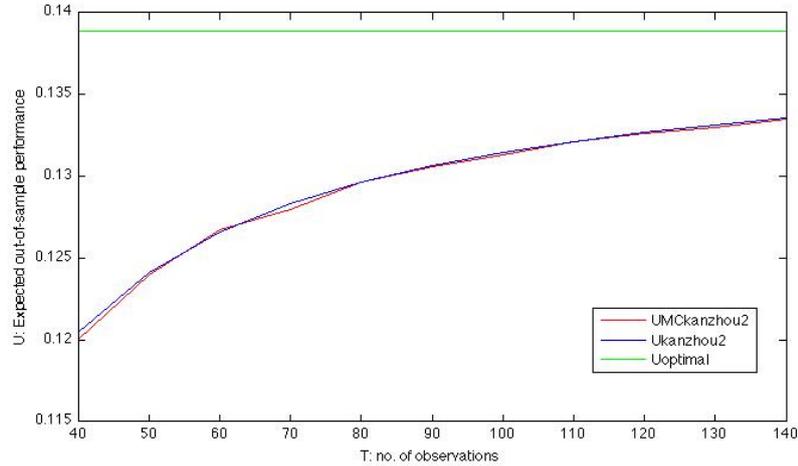
In this section we extend the Monte Carlo algorithm for the plug-in rule in order to account new portfolio rules and optimise with respect to some parameters. The idea is to add some parameters that will be chosen by numerical optimisers so that the expected out-of-sample performance of the portfolio is optimised. Examples of that are Kan-Zhou 2 fund rule and Kan-Zhou 3 fund rule. As we can see from equations (2.15) and (2.19) Kan and Zhou introduce some parameters for the weight function. Then they analytically on their paper derive the values of these parameters such

that they optimise the expected out-of-sample performance. We will do the same in this section using the algorithm introduced in the previous section 3.2. The first 6 steps of the algorithm will be the same. Then it depends on the portfolio rule we need to follow. Let's write down the algorithm that can implement Kan-Zhou 2 fund and Kan-Zhou 3 fund. The first 6 steps are the same. Then we will have:

7. Substitute  $\mu_{MLE}$  and  $\Sigma_{MLE}$  in the weight function (2.15) for Kan-Zhou 2 fund or (2.19) for Kan-Zhou 3 fund.
8. Substitute the observed weight function from step 6 in (2.12) in order to find the out-of-sample utility function.
9. We loop over steps 3-9  $Nsim$  times and for each iteration we find a different out-of-sample utility function. This is because the realisations we draw in step 3 will be different for each iteration.
10. Find the average of all utility functions.
11. Observe the expected out-of-sample utility function.
12. Run all the steps 1-11 using a numerical optimiser in order to choose the values for the parameters in the weight functions ( $c$  for Kan-Zhou 2 fund,  $c$  and  $d$  for Kan-Zhou 3 fund) introduced in step 7 in order to maximise the expected out of sample performance.
13. Printout the optimal value of the parameters and the optimal expected out-of-sample performance.

The best way to check that this algorithm works properly is to compare it against Kan-Zhou 2 fund rule. Notice that we can't compare it with Kan-Zhou 3 fund rule as it is not implementable as we already discussed. However if this algorithm works properly we will be able to implement Kan-Zhou 3 fund. We draw the graph of Kan-Zhou 2 fund rule using (2.19) and using the above algorithm.

We assume a portfolio with two assets with mean of the returns vector  $\mu = (0.5, 0.9)$ , correlation matrix  $\Sigma = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}$  and number of simulations equal to 1000.



In the above figure we can see that as no. of observations  $T$  is increasing the MC error is decreasing and the MC curve converges to the real Kan-Zhou 2 fund curve. Hence the introduced algorithm works properly. So, one application of the algorithm would be to draw Kan-Zhou 3 fund rule. We will do this later on when we will compare all the rules together. Furthermore, we can see that as  $T$  increases then both curves converge to the optimal utility function as expected.

# Chapter 4

## Shrinkage for the parameters

### 4.1 Motivation

In this chapter we will introduce some shrinkage estimators for our parameters mean and covariance matrix  $\hat{\mu}$  and  $\hat{\Sigma}$  respectively which will improve the performance of the portfolio. As we already mention Markowitz's formulas (2.2) and (2.3) are optimal if and only if we know the true distribution of the assets returns i.e. if we know the true mean and variance of each asset's return. Since in reality we don't know the exact distribution the best we can do is to find an estimate of the distribution of the assets returns. When we don't know the exact distribution as we already discussed in chapter 2 it is better not to follow Markowitz's formulas for our investment decision.

The shrinkage estimators we will introduce will depend on how much evidence we have for the assets returns distribution. For example if we are almost certain for the distribution then the model we have with the shrinkage estimators should approach to the Markowitz model (2.11). This is because we will be more certain for our parameters and  $\hat{\mu} \sim \mu$  and  $\hat{\Sigma} \sim \Sigma$ . where  $\mu$  and  $\Sigma$  the true parameters.

Also the less certain we are for the distribution of the assets returns and our model should approach the 1/N weight function  $\hat{w} \sim 1/N$ . This means for example that when T is very small we are more uncertain for the distribution of the assets returns and we prefer to allocate our wealth equally between the assets. This was examined in section 2.3 where we saw from the graph that 1/N rule outperforms the other rules for T very small. Note that this is not always the case. Hence we somehow need our weight formula to depend on how certain we are for the assets return distribution. Kan-Zhou 2 fund rule and 3 fund rule are also based on this idea since their weight formula (2.15) and (2.19) depends on the number of assets N and the

number of observations  $T$  as we discussed in chapter 2 .

We can achieve the above by introducing some parameters to our shrinkage estimators formula that will control the uncertainty of the distribution. This is a similar idea to the Kan-Zhou 2 fund and 3 fund portfolio rules as in both rules we had the parameter  $c$  for Kan-Zhou 2 fund as in formula (2.15) and  $c$  and  $d$  for Kan-Zhou 3 fund as in formula (2.19).

## 4.2 Mean shrinkage

In this section we will introduce a shrinkage for the mean of the returns of the assets. With this shrinkage we will try to control the uncertainty of the mean and hence have a better expected out-of-sample performance. An example will make things more clear.

### Example 1

Assume we have the true distribution of the assets returns.  $\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$ ,  $\mu = (1.1, 1)$  and  $\gamma = 3$ . Then the optimal Markowitz formula (2.2) will give:

$$w^* = \frac{1}{\gamma} \Sigma^{-1} \mu = \frac{1}{3} \begin{pmatrix} 5.2632 & -4.7368 \\ -4.7368 & 5.2632 \end{pmatrix} \begin{pmatrix} 1.1 \\ 1 \end{pmatrix} = (0.3509, 0.0175)$$

If we normalise that so we can see the proportion invested in each of the assets we get the optimal weight vector  $w^* = (0.9524, 0.0476)$ . This means an investor should invest 0.9524 in the first asset and 0.0474 in the other in order to keep an optimal position in the market if he knows the exact distribution of the assets returns.

Notice that in this example the difference between the mean return of the two assets is just 0.1. However the resulting difference between the proportion of wealth allocate to the first and to the second asset is much more. This is because since we know the exact distribution of the assets why we should then invest in the second one as the first gives a 0.1 higher mean return. Also this difference depend on the correlation. Since the two assets are highly correlated then they tend to move together in price. So by investing in one of the two assets or in both assets it is about the same investment since whichever you choose you will have the same increase or decrease in the price. Hence the 0.1 difference between the means is penalised even more since when the assets will move together the first asset will have higher expected return so then why investing in the second.

However now let's get back to our problem. The problem is that in reality the distribution of the assets returns is unknown so we will find some estimates for the mean and covariance matrix  $\mu_{MLE}$  and  $\Sigma_{MLE}$ . Let's assume that the values of the true parameters in the example are just estimates such that  $\mu_{MLE} = (1.1, 1)$  and  $\Sigma_{MLE} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$ . If we follow the plug-in rule as in section 2.2.1 we will get the weight vector  $\hat{w} = (0.9524, 0.0476)$ . So again it penalises the 0.1 difference of the mean a lot, but this time we are not certain about the assets returns distribution. This can be very misleading investment as the 0.1 difference might be a distribution error of our estimate and not an actual difference between the means. By taking into account the uncertainty of the distribution, a weight vector with less difference between  $\hat{w}_1$  and  $\hat{w}_2$  will have a better out-of-sample performance since it will penalise less the 0.1 difference between the means. On the other hand if we have a large number of observations and we are more certain for the distribution then the optimal investment will tend to the Markowitz weight allocation. A shrinkage for the mean that takes into consideration all the above is:

$$\hat{\mu} = \alpha\mu_{MLE} + (1 - \alpha)E(\mu_{MLE}) * \mathbf{1}_N \quad (4.1)$$

where  $\alpha$  is a parameter from 0 to 1 that will be chosen in order to optimise the expected out-of-sample performance. The second term is the expectation of the mean vector i.e the average of the means. The second term is there in order to decrease the difference between the means so that the weight allocation will not penalise the difference between the means that much.

Let's see how the weight allocation will be in example 1 using the mean shrinkage. We have the mean vector  $\mu_{MLE} = (1.1, 1)$ . Assume that optimal value for  $\alpha$  in this example is  $\alpha = 0.3$ . We will have then  $\hat{\mu} = 0.3(1.1, 1) + 0.7(1.05, 1.05) = (1.065, 1.035)$ . The difference of 0.1 became a difference of 0.03 and as a result the weight allocation formula will not penalise that much this difference. After doing the calculations and normalise the weight vector we will get  $\hat{w} = (0.6357, 0.3643)$ . This is exactly as we wanted as we are investing more in the first asset because of the higher expected return but the difference between  $\hat{w}_1$  and  $\hat{w}_2$  is not that much. Of course this difference will vary in each situation depending on the maximum likelihood estimators for the mean and covariance matrix, the number of assets and the number of observations T. The numerical optimiser we will introduce in the algorithm introduced in section 3.2 is responsible to take all the above into account

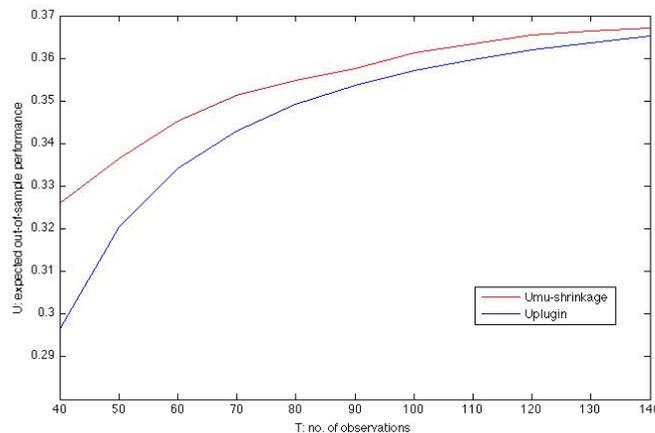
and find the value for  $\alpha$  that will give the highest possible expected out-of-sample performance.

We need to plugin the shrinkage estimator for  $\alpha$  in the algorithm in section 3.2 and then optimise the result of the algorithm with respect to  $\alpha$ . The algorithm will then then be the same up to step 6 and then as follows:

7. Define  $\hat{\mu} = \alpha\mu_{MLE} + (1 - \alpha)E(\mu_{MLE})$
8. Substitute  $\hat{\mu}$  and  $\Sigma_{MLE}$  in the weight function (2.11).
9. Substitute the observed weight function from step 6 in (2.12) in order to find the out-of-sample utility function.
10. We loop over steps 3-7  $Nsim$  times and for each iteration we find a different out-of-sample utility function. This is because the realisations we draw in step 3 will be different for each iteration.
11. Find the average of all utility functions.
12. Find the expected out-of-sample utility function.
13. Run all steps 1-12 using a numerical optimiser in order to choose the value of the parameter  $\alpha$  that optimises the expected out-of-sample performance.
14. Printout the optimised expected out-of-sample performance and the optimal value for parameter  $\alpha$ .

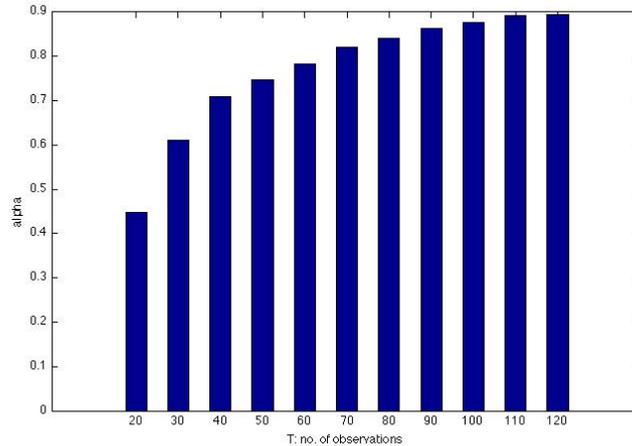
We can now show the improvement of the performance of the portfolio rule using  $\alpha$  by drawing a graph and compare it with the plug-in rule.

Let  $Nsim = 1000$ ,  $\Sigma = \begin{pmatrix} 1 & 0.2 & 0.6 \\ 0.2 & 1 & 0.8 \\ 0.6 & 0.8 & 1 \end{pmatrix}$ ,  $\mu = (0.9, 0.5, 0.3)$



From the above graph we can see clearly the impact of the  $\alpha$  shrinkage for the

mean. For  $T$  small since we don't have enough evidence for the distribution of the asset returns we expect that  $\alpha$  will be closer to 0 and the expected out-of-sample performance to be much better than the one of the plug-in rule. As  $T$  is increasing we can see that the plug-in curve is approaching the  $\alpha$  shrinkage curve. This is because as we have a larger number of  $T$  observations the  $\hat{\mu}$  shrinkage estimator will approach  $\mu_{MLE}$  as  $\alpha$  will tend to 1. It will be very interesting to see how  $\alpha$  reacts with respect to the  $T$  observations.



In the above histogram we can see the values of the parameter  $\alpha$  with respect to the number of observations  $T$ . The value of  $\alpha$  is increasing as  $T$  is increasing. This is because as  $T$  increases we are more certain for the distribution of the asset returns and  $\alpha$  approaches 1 so that the weight formula approaches the Markowitz's weight formula (2.2). For  $T$  small  $\alpha$  is smaller as we don't want to penalise the difference between the means a lot because we don't know their exact distribution. For  $T$  small enough  $\alpha$  will be zero and the portfolio will be the minimum variance portfolio. The optimal for the case when  $T$  is that small would be to converge to the  $1/N$  rule. In order to achieve the  $1/N$  portfolio rule for  $T$  small we will need to combine the shrinkage for the mean with a shrinkage for the covariance matrix as well. We will investigate this combination in section 4.4 after introducing the covariance matrix shrinkage in section 4.3.

### 4.3 Covariance Matrix shrinkage

In this section we introduce a shrinkage for the covariance matrix of the assets returns. With similar logic as for the mean shrinkage we will try with the covariance matrix shrinkage to control the uncertainty of the distribution depending on the information we have.

Let's consider again the example 1 from the previous section which is a very good example in order to show the impact of the covariance matrix shrinkage estimator. Recall that in example 1 we had  $\mu_{MLE} = (1.1, 1)$ ,  $\Sigma_{MLE} = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$  and  $\gamma = 3$ . If we apply the plug-in rule we get the weight vector  $\hat{w} = (0.9524, 0.0476)$ . Which as we discussed it can be very misleading if we don't have enough information for the assets return distribution. The 0.1 difference between the means is penalised more than it should and our asset allocation is not safe. The 0.1 difference as we already mention it can be an error from the distribution estimation.

In this section instead of reducing the difference between the mean returns of the assets we will reduce the correlation between the assets. The idea is that if the correlation between two or more assets is less than the estimated one, we allocate the wealth between the assets more uniformly. This of course will depend on the mean returns of the assets, the number of assets, the number of T observations and the estimated covariance matrix. For example if we are more certain of the distribution then the covariance shrinkage should converge to the maximum likelihood estimator for the covariance matrix. A shrinkage then for the covariance matrix will then be:

$$\hat{\Sigma} = \beta \Sigma_{MLE} + (1 - \beta) \text{diag}(\Sigma_{MLE}) \quad (4.2)$$

where  $\beta$  is a parameter from 0 to 1 that we wish we choose in order to optimise the expected out-of-sample performance. We choose the second term to be the diagonal of the covariance matrix so that we keep it fixed. The diagonal remains the same as in the first term we multiply the diagonal of  $\Sigma_{MLE}$  with  $\beta$  and then we add again the diagonal of  $\Sigma_{MLE}$  multiplied by  $1 - \beta$  so that  $\beta \text{diag}(\Sigma_{MLE}) + (1 - \beta) \text{diag}(\Sigma_{MLE}) = \text{diag}(\Sigma_{MLE})$ . At the same time we reduce the off-diagonal terms of  $\Sigma_{MLE}$ .

Let's again go back to example 1 to check the effect we will have by reducing the covariance matrix of the assets returns. Recall from above that using the plug-in rule we have  $\hat{w} = (0.9524, 0.0476)$ . Now let's say we run our algorithm and it outputs that for the given information the best estimate for  $\beta$  is 0.7. Then we will have:

$$\hat{\Sigma} = \beta \Sigma_{MLE} + (1 - \beta) \text{diag}(\Sigma_{MLE}) = 0.7 \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} + 0.3 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0.63 \\ 0.63 & 1 \end{pmatrix}$$

We plug this in the weight function formula (2.11) together with the maximum likelihood estimator for the mean  $\mu_{MLE}$  so that:

$$\hat{w} = \frac{1}{\gamma} \Sigma^{-1} \mu = \frac{1}{3} \begin{pmatrix} 1.6581 & -1.0446 \\ -1.0446 & 1.6581 \end{pmatrix} \begin{pmatrix} 1.1 \\ 1 \end{pmatrix} = (0.2598, 0.1697)$$

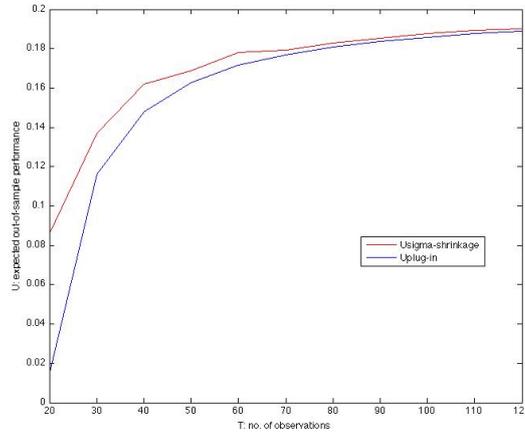
Then we normalise the weight function in order to see the proportion of wealth that will be invested in each of the asset. The weight vector will be  $\hat{w} = (0.6049, 0.3951)$ . The result is as we wanted in order to decrease the difference between  $\hat{w}_1$  and  $\hat{w}_2$ . This again will depend on the available information for the assets returns. If we are more certain for the distribution  $\beta$  will converge to 1,  $\hat{\Sigma}$  to  $\Sigma$  optimal, and  $\hat{w}$  to  $w^*$ . The numerical optimiser is responsible to assign the value to  $\beta$  that optimises the expected out-of-sample performance given all the available information.

Similarly to the mean shrinkage case we need to change the algorithm in section 3.2. The first 6 steps of the algorithm will be the same. Then we will have:

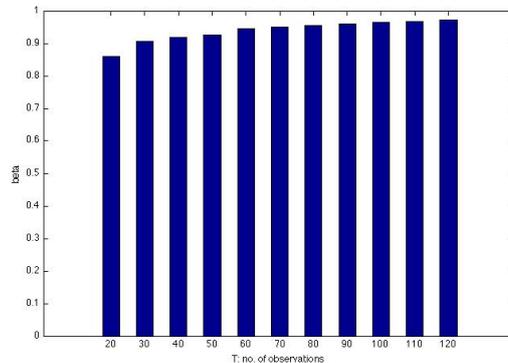
7. Define  $\hat{\Sigma} = \beta \Sigma_{MLE} + (1 - \beta) \text{diag}(\Sigma_{MLE})$
8. Substitute  $\mu_{MLE}$  and  $\hat{\Sigma}$  in the weight function (2.11).
9. Substitute the observed weight function from step 6 in (2.12) in order to find the out-of-sample utility function.
10. We loop over steps 3-7 *Nsim* times and for each iteration we find a different out-of-sample utility function. This is because the realisations we draw in step 3 will be different for each iteration.
11. Find the average of all utility functions.
12. Find the expected out-of-sample utility function.
13. Run all steps 1-12 using a numerical optimiser in order to choose the value of the parameter  $\beta$  that optimises the expected out-of-sample performance.
14. Printout the optimised expected out-of-sample performance and the optimal value for parameter  $\beta$ .

It will be interesting now to see the improvement in the expected out-of-sample performance when we use the covariance matrix shrinkage. Furthermore is important to see the values that the optimiser assigns to  $\beta$  with respect to the number of observations  $T$ . We will first draw the graph of the expected out-of-sample utility performance of the plug-in rule and the rule using the covariance matrix shrinkage in order to compare them. Then we will draw a histogram to see the values assigned to  $\beta$  for some values of  $T$ .

For both graphs Nsim=1000,  $\Sigma = \begin{pmatrix} 1 & 0.1 & 0.2 \\ 0.1 & 1 & 0.9 \\ 0.2 & 0.9 & 1 \end{pmatrix}$ ,  $\mu = (0.2, 1, 1.1)$ .



In the above figure we can see that using the sigma shrinkage estimator we increase the expected out-of-sample performance. For T small the difference between the curves is larger as the covariance matrix shrinkage controls the uncertainty when we don't have enough information for the asset returns distribution. As T increases the plug-in curve approaches the the sigma shrinkage curve as sigma approaches 1 for large T.



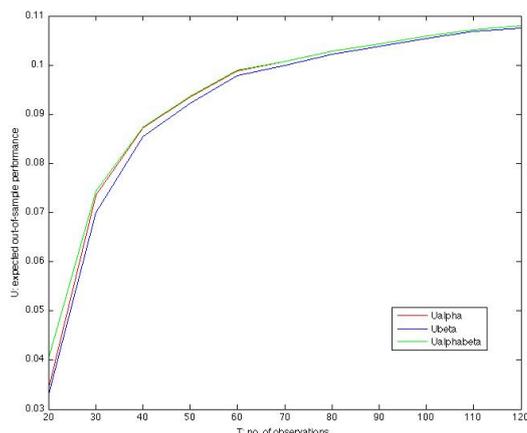
In the above figure we can see that beta is increasing with the number of observations T. This is because as T increases we are again more certain about the distribution of the asset returns so sigma converges to one so that our model converges to the Markowitz's optimal weight formula (2.2).

## 4.4 Mean and Covariance matrix shrinkage

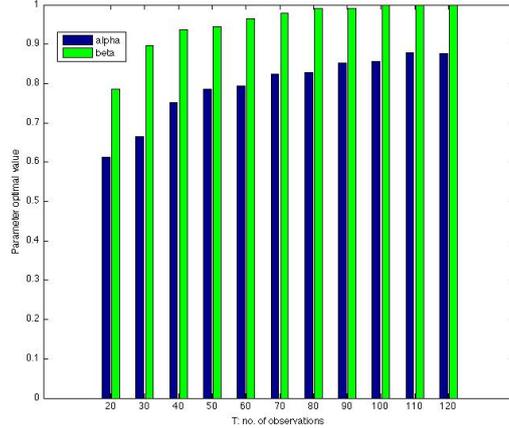
In this section we combine the shrinkage for the mean and covariance matrix introduced in sections 4.2 and 4.3 respectively. By combining them we expect to have even better expected out-of-sample performance as we take the benefits of both shrinkage estimators. The mean shrinkage will bring the means closer to each other and the covariance matrix shrinkage will reduce the correlation between the assets returns when it is required.

In order to see the improvement of using both shrinkages we can draw the expected out-of-sample performance of the mean shrinkage, covariance matrix shrinkage and both shrinkage in the same plot against the number of observations  $T$ . For both

graphs  $N_{sim}=1000$ ,  $\Sigma = \begin{pmatrix} 1 & 0.3 & 0.5 \\ 0.3 & 1 & 0.2 \\ 0.5 & 0.2 & 1 \end{pmatrix}$ ,  $\mu = (0.7, 0.6, 0.2)$



From the above figure we can see that using both shrinkages uniformly outperforms using just one of the two. We can also see on this graph that the curve with the mean shrinkage performs better than the one of covariance matrix shrinkage. This is not always true as it depends on the number of assets, their mean returns and their correlation matrix.



In the above bar chart we can see the values that are being assigned to the parameters  $\alpha$  and  $\beta$  in order to optimise the expected out-of-sample performance of the mean covariance shrinkage. As expected the optimal values for the parameters  $\alpha$  and  $\beta$  are increasing with  $T$ . This is because as  $T$  increases we are more certain for the distribution of the assets returns and our model converges to Markowitz's optimal model where  $\alpha$  and  $\beta$  are equal to 1.

## 4.5 Correlation Smoothing

In this section we will introduce another efficient shrinkage for the mean. This shrinkage will give a very good weight allocation especially in cases where the mean returns of the assets are relatively close and there is a big difference between the correlations of the assets returns. In order to achieve this we need a shrinkage that will consider the high correlated assets as one asset together. If two assets are highly correlated then is better to consider them as one asset as they will tend to move together. Let's give an example to make things more clear:

### Example 2

Consider  $\Sigma_{MLE} = \begin{pmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ ,  $\mu_{MLE} = (1.1, 1, 1)$

If we substitute them in the plug-in formula for the weights (2.11) we will get a weight vector  $\hat{w} = (0.5, 0.025, 0.475)$ . This is not a very good weight allocation between the three assets as it penalises the 0.1 difference between the assets too much. If we follow the plug-in strategy we almost invest nothing in the second asset which has almost the same return as the first asset. This would be optimal allocation if

we knew that the 0.1 difference between the assets returns is true and not just an estimate.

A good idea would be to shrink the means according to correlations. The way to achieve that is to normalise each row of  $\Sigma_{MLE}$  and then multiply it with  $\mu_{MLE}$ . By performing this we keep the mean value of the return of the uncorrelated assets fixed and we bring together the values of the mean returns of the correlated assets. Then we expect that the weight allocation will not penalise the second asset that much. Let's derive the result make some conclusions.

$$\hat{\mu} = norm(\Sigma_{MLE})\mu_{MLE} = \begin{pmatrix} 0.5263 & 0.4737 & 0 \\ 0.4737 & 0.5263 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1.1 \\ 1 \\ 1 \end{pmatrix} = (1.0526, 1.0474, 1)$$

Then we substitute  $\hat{\mu}$  and  $\hat{\Sigma}$  to the weight function where  $\hat{\Sigma} = \Sigma_{MLE}$ .

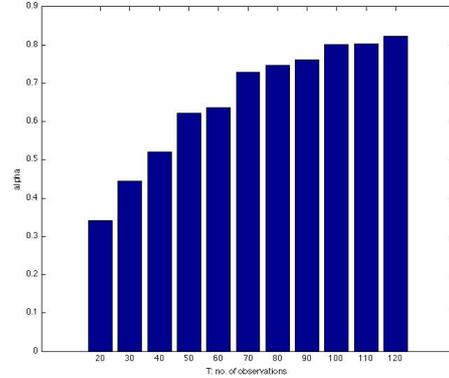
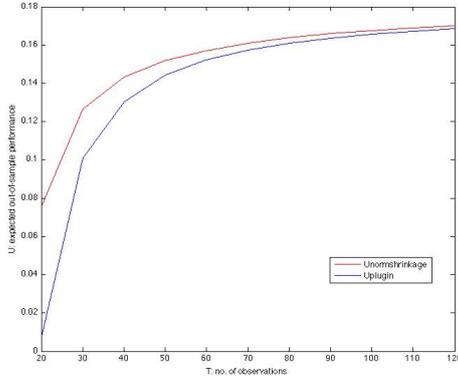
$$\hat{w} = \frac{1}{\gamma} \hat{\Sigma}^{-1} \hat{m} u = (0.5789, 0.5263, 1)$$

We normalise the observed weight vector in order to see the proportion of the weight allocation in each asset. We will then have  $\hat{w} = (0.275, 0.25, 0.475)$ . We can see here that by using this estimator for  $\hat{\mu}$  we get a very good weight allocation when we are out-of-sample. It separates the three asset portfolio in two portfolios. The portfolio with the first two assets and another portfolio with the third asset. Then it gives almost equal wealth to both portfolios as their mean returns is almost the same. Furthermore in the first portfolio it does not penalise the 0.1 difference between asset 1 and asset 2. A good shrinkage then for the mean would be:

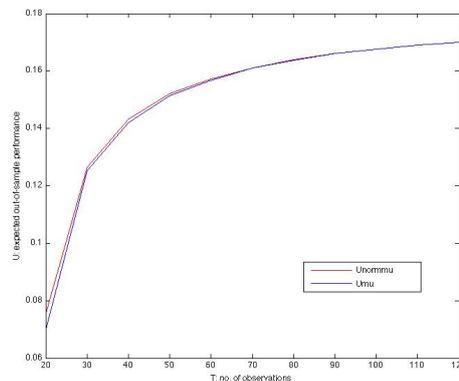
$$\hat{\mu} = \alpha \mu_{MLE} + (1 - \alpha) norm(\Sigma_{MLE}) \mu_{MLE} \quad (4.3)$$

Let's now check the improvement of this norm shrinkage estimator by comparing the expected out-of-sample performance with the one of plug-in's rule.

For the figures below we have  $\Sigma_{MLE} = \begin{pmatrix} 1 & 0.9 & 0.3 \\ 0.9 & 1 & 0.5 \\ 0.3 & 0.5 & 1 \end{pmatrix}$ ,  $\mu_{MLE} = (0.9, 0.7, 0.5)$



In the first figure we can see the improvement using the norm shrinkage for the mean by comparing its expected out-of-sample performance with the plug-in rule. From both graphs we can see that for small T the shrinkage is more active since  $\alpha$  is smaller and gives a much better expected out-of-sample performance than the plug-in rule. For T large  $\alpha$  converges to 1 and both curves approach each other and converge to the optimal Markowitz portfolio rule as we are more certain for the assets distribution.



In the above figure we compare the two shrinkages for the mean we introduced in section 4.2 and section 4.5. We compare (4.1) with (4.3). From the graph we can see that the last shrinkage (4.3) performs slightly better than (4.1). This result again will vary between each situation. Both curves depend on the number of assets, their estimated distribution and the number of observations so for different situations we will have different rank for their expected out-of-sample performance.

# Chapter 5

## Numerical portfolio rules and comparisons

### 5.1 Numerical portfolio rules

In this section we will introduce two portfolio rules that are based on the shrinkage estimators we introduce in chapter 4. Using the shrinkage estimators from chapter 4 we can develop portfolio rules that uniformly outperform Kan-Zhou 2 fund and Kan-Zhou 3 fund. We first need to notice some facts for these two rules.

For Kan-Zhou 2 fund rule we define the weight function as  $\hat{w} = \frac{c}{\gamma} \hat{\Sigma}^{-1} \hat{\mu}$  with a scale parameter  $c$ . Then we find the scale  $c$  in order to optimise the expected out-of-sample performance. As we did in section 3.3 we proved that using the algorithm introduced we can implement Kan-Zhou 2 fund rule. This is the same as using the shrinkage  $\hat{w} = \frac{c}{\gamma} \hat{\Sigma}^{-1}$  for the weight function in the algorithm. Another way is to use the shrinkage for the mean  $\hat{\mu} = c \mu_{MLE}$ . Because then when we substitute  $\hat{\mu}$  to the weight function  $\hat{w} = \frac{1}{\gamma} \hat{\Sigma}^{-1} \hat{\mu}$  you get  $\hat{w} = \frac{c}{\gamma} \hat{\Sigma}^{-1} \hat{\mu}$ .

Kan-Zhou 3 fund, as we already mentioned, is not implementable but since we proved that the algorithm introduced in section 3.3 works properly then we can use it in order to implement Kan-Zhou 3 fund rule. We expect some Monte Carlo error on this implementation but as we showed in section 2.2 this error is very small when we have a large number of observations  $T$  or large number of simulations  $N_{sim}$ .

We can implement Kan-Zhou 3 fund in two ways. The first way is to define the shrinkage  $\hat{w} = \frac{1}{\gamma} (c \hat{\Sigma}^{-1} \hat{\mu} + d \hat{\Sigma}^{-1} \mathbf{1}_N)$  for the weight function and optimise for parameters  $c$  and  $d$  using the algorithm in section 2.2 The other way is to use the shrinkage

for the mean  $\hat{\mu} = c\mu_{MLE} + d\mathbf{1}_N$  and then by substituting it to the weight function  $\hat{w} = \frac{1}{\gamma}\hat{\Sigma}^{-1}\hat{\mu}$ . So Kan-Zhou 3 fund rule is just a shrinkage for the mean. We can uniformly then outperform Kan-Zhou 3 fund rule if for example we use the mean shrinkage of Kan-Zhou 3 fund combined with a shrinkage for the covariance matrix as well.

We would now like to compare all the results of chapter 4 together with Kan-Zhou 3 fund. From chapter 4 we introduced 4 different rules using shrinkages for the mean and covariance matrix that improve the expected out-of-sample performance. The mean and covariance matrix shrinkage introduced in section 4.4 outperforms the mean shrinkage and the covariance matrix shrinkage introduced in sections 4.2 and 4.3 respectively since it is a combination of both. So one of the models we deal with in this chapter is the mean and covariance matrix shrinkage of section 4.4. The other model we would like to deal with is the model introduced in section 4.5 with the norm shrinkage for the mean.

We now add another shrinkage to these two rules in order to proceed to our comparison. The two rules that we proved both allocate the wealth in a very good way so that they maximise the expected out-of-sample performance. We need to notice here that although both rules have a very good weight allocation our target is to maximise the expected out-of-sample performance. So we might have a good weight allocation but we don't have a parameter to control how much wealth we should invest in the risky assets. For example, an investor can achieve a very good relative weight allocation but will not invest the optimal wealth in the risky assets in order to maximise the portfolio performance. A shrinkage that will control that will be:

$$\hat{w} = \frac{c}{\gamma}\hat{\Sigma}^{-1}\hat{\mu} \quad (5.1)$$

with a parameter  $c$  chosen so that we invest the optimal proportion of wealth in the risky assets. The three models that we would like to compare in this chapter are:

### Model 1

$$\hat{\mu} = \alpha\mu_{MLE} + (1 - \alpha)E(\mu_{MLE}) \quad (5.2)$$

$$\hat{\Sigma} = \beta\Sigma_{MLE} + (1 - \beta)diag(\Sigma_{MLE}) \quad (5.3)$$

$$\hat{w} = \frac{c}{\gamma}(\hat{\Sigma}^{-1}\hat{\mu}) \quad (5.4)$$

## Model 2

$$\hat{\mu} = \alpha\mu_{MLE} + (1 - \alpha)norm(\Sigma_{MLE})\mu_{MLE} \quad (5.5)$$

$$\hat{\Sigma} = \Sigma_{MLE} \quad (5.6)$$

$$\hat{w} = \frac{c}{\gamma}(\hat{\Sigma}^{-1}\hat{\mu}) \quad (5.7)$$

## Model 3 (Kan-Zhou 3 fund)

$$\hat{\mu} = \mu_{MLE} \quad (5.8)$$

$$\hat{\Sigma} = \Sigma_{MLE} \quad (5.9)$$

$$\hat{w} = \frac{1}{\gamma}(c\hat{\Sigma}^{-1}\hat{\mu} + d\hat{\Sigma}^{-1}\mathbf{1}_N) \quad (5.10)$$

In the next section we will compare the three models in order to find out which works best under different situations.

## 5.2 Comparison of the rules

In this section we will compare the three models introduced in the previous section. We will test these rules under several situations in order to make conclusions. We will also include the value of the parameters in the tables below in order to see the values the parameters are being assigned under different values of  $\mu$  and  $\Sigma$ . First we will keep the mean of the assets returns fixed and test the rules for the cases when the assets are highly correlated, random correlated and low correlated. Then we will keep the covariance matrix fixed and change the mean of the returns.

Let  $\mu = (0.3, 0.5, 0.7)$  and  $Nsim = 1000$ .

For  $T = 20$ :

$\Sigma = (\rho_{12}, \rho_{13}, \rho_{23})$	$\alpha$	$\beta$	$c$	$Umodel1$	$Umodel2$	$Ukanzhou3$
(0.1, 0.5, 0.9)	1	0.5897	0.3399	0.0526	-	0.1578
	1	-	0.5429	-	0.1578	0.1578
(0.5, 0.5, 0.5)	1	0.4647	0.5016	0.0598	-	0.0582
	0.6308	-	0.5308	-	0.0548	0.0582
(0.3, 0.5, 0.7)	1	0	0.3655	0.0564	-	0.0518
	0.7557	-	0.5037	-	0.0491	0.0518

For  $T = 80$ :

$\Sigma = (\rho_{12}, \rho_{13}, \rho_{23})$	$\alpha$	$\beta$	$c$	$Umodel1$	$Umodel2$	$Ukanzhou3$
(0.1, 0.5, 0.9)	1	0.9996	0.8968	0.2113	-	0.2112
	1	-	0.8708	-	0.2112	0.2112
(0.5, 0.5, 0.5)	1	0.8546	0.8386	0.0798	-	0.0797
	0.8248	-	0.8740	-	0.0792	0.0797
(0.3, 0.5, 0.7)	1	0.9187	0.8397	0.073	-	0.0729
	0.9095	-	0.8511	-	0.0725	0.0729

For  $T = 150$ :

$\Sigma = (\rho_{12}, \rho_{13}, \rho_{23})$	$\alpha$	$\beta$	$c$	$Umodel1$	$Umodel2$	$Ukanzhou3$
(0.1, 0.5, 0.9)	1	0.9997	0.9397	0.2196	-	0.2195
	1	-	0.9205	-	0.2195	0.2195
(0.5, 0.5, 0.5)	1	0.9148	0.8993	0.0838	-	0.0838
	0.8952	-	0.9233	-	0.0836	0.0838
(0.3, 0.5, 0.7)	1	0.9505	0.9035	0.0771	-	0.077
	0.9394	-	0.9124	-	0.0769	0.077

From the above tables we can see that all the rules have a very close expected out-of-sample performance. It is important to notice the behaviour of the scale parameter  $c$ . The parameter  $c$  is very small when  $T$  is small and it is increasing with  $T$ . This is not surprising as the less observations we have, the less uncertain we are about the assets return distribution and the less we should invest in the risky assets. Since parameter  $c$  is responsible for controlling the proportion invested in the risky assets we have the observed result. Furthermore the other parameters  $\alpha$  and  $\beta$  are increasing with  $T$ .

Now we choose a random covariance matrix  $\Sigma = \begin{pmatrix} 1 & 0.2 & 0.6 \\ 0.2 & 1 & 0.8 \\ 0.6 & 0.8 & 1 \end{pmatrix}$ , and  $Nsim = 2000$ . We would like to test this for various  $T$  as it is very important to see how the parameters react when we have different level of uncertainty.

For  $T = 20$ :

$\mu$	$\alpha$	$\beta$	$c$	$U_{model1}$	$U_{model2}$	$U_{kanzhou3}$
(1, 0.5, 0.1)	1	0.9823	0.7312	0.5535	-	0.5407
	1	-	0.5897	-	0.5402	0.5407
(0.5, 0.5, 0.5)	0.1577	0.9799	0.5577	0.518	-	0.599
	0.1737	-	0.5105	-	0.0499	0.599
(0.7, 0.5, 0.3)	1	0.9905	0.6186	0.1814	-	0.1824
	1	-	0.5572	-	0.1796	0.1824

For  $T = 80$ :

$\mu$	$\alpha$	$\beta$	$c$	$U_{model1}$	$U_{model2}$	$U_{kanzhou3}$
(1, 0.5, 0.1)	1	0.9893	0.9782	0.6751	-	0.6733
	0.9812	-	0.8999	-	0.6733	0.6733
(0.5, 0.5, 0.5)	0.0973	0.9922	0.8796	0.0704	-	0.0726
	0	-	0.859	-	0.0698	0.0726
(0.7, 0.5, 0.3)	1	0.9947	0.9122	0.2334	-	0.2334
	0.985	-	0.8808	-	0.2332	0.2334

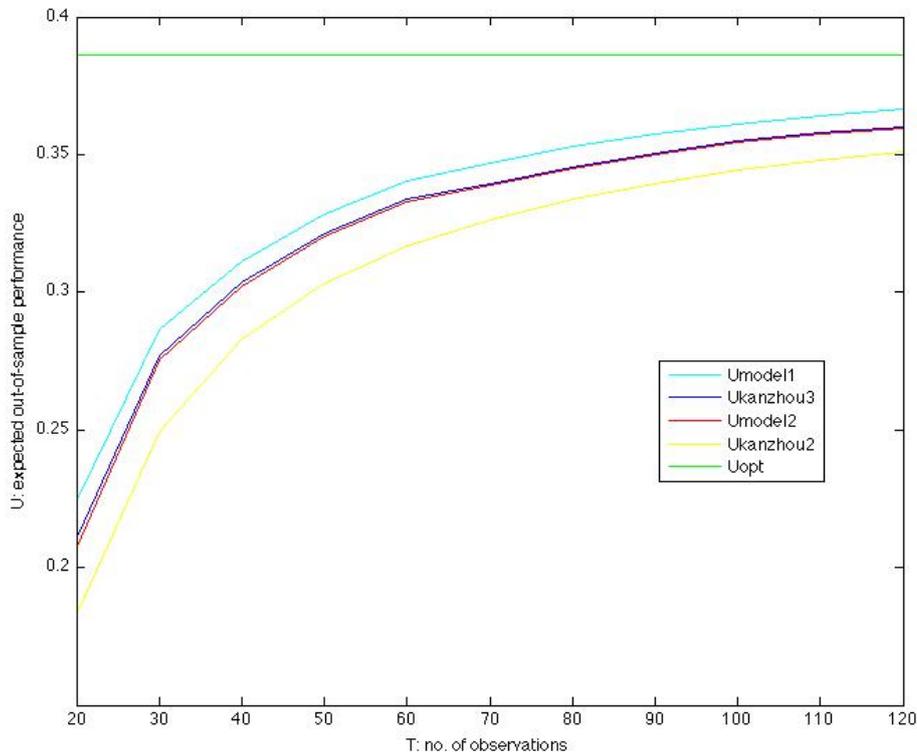
For  $T = 150$ :

$\mu$	$\alpha$	$\beta$	$c$	$U_{model1}$	$U_{model2}$	$U_{kanzhou3}$
(1, 0.5, 0.1)	1	0.9932	0.9979	0.695	-	0.6944
	0.9737	-	0.9578	-	0.6944	0.6944
(0.5, 0.5, 0.5)	0.1067	0.9968	0.9343	0.0733	-	0.0746
	0	-	0.926	-	0.0731	0.0746
(0.7, 0.5, 0.3)	1	0.9968	0.9572	0.2423	-	0.2424
	0.9823	-	0.9426	-	0.2423	0.2424

For the above tables we can make the same comments for parameter  $c$  as the previous case when  $\mu$  was fixed. The portfolios again have a very close expected out-of-sample performance. We need to notice here the value of the parameter  $\alpha$  when  $\mu$  is the same for all the assets (i.e. for this example  $\mu = (0.5, 0.5, 0.5)$ ). Parameter  $\alpha$  takes very small values. This is because the expectation of the  $\mu_{MLE}$  is closer to

the true  $\mu$  than the  $\mu_{MLE}$ . For example if we have the vectors  $\mu = (0.5, 0.5, 0.5)$  and  $\mu_{MLE} = (0.45, 0.52, 0.55)$  then the shrinkage for the mean (4.1) will give  $\hat{\mu} = \alpha(0.45, 0.52, 0.55) + (1 - \alpha)(0.5067, 0.5067, 0.5067)$ . Hence for  $\alpha$  small we give more weight to the second term which is closer to the real  $\mu$ .

Now let's assume a larger portfolio with 7 assets. We will draw all the rules we discussed in this project. Note that for Kan-Zhou 2 fund we will use the formula (2.18). We will also draw the optimal curve which is given by Markowitz formula (2.3).



From the figure we can see that all the curves as expected they converge to the optimal curve as  $T$  is increasing. For this particular situation we can see that Model's 1 curve uniformly outperforms the others for all values of  $T$ . Kan-Zhou 3 curve is very close to the Model's 2 curve along the way and Kan-Zhou 2 has the less expected out-of-sample performance for all values of  $T$ . We need to comment that if we consider a different covariance matrix and different mean the rank list might not be the same. The only fact we are certain about is that in all situations model 1, model 2 and Kan-Zhou 3 fund rules uniformly outperform Kan-Zhou 2 fund since they include it in their shrinkages. Model 1 is the same as Kan-Zhou 2 fund when  $\alpha$  and  $\beta$  equal 1. Model 2 when  $\alpha$  equals 1 and Kan-Zhou 3 fund when  $d$  equals 0.

# Chapter 6

## Conclusion

In this dissertation we dealt with optimal portfolio choice under parameter uncertainty. Several studies on this issue proved that when we don't have the real distribution of the parameters i.e. the mean and the covariance matrix of the assets returns then it is not optimal to follow Markowitz's formulas. It is better to follow different portfolio strategies.

We introduce another way of calculating the expected out-of-sample performance of a portfolio rule by developing an algorithm. This makes the investigation of the portfolio rules much easier as we can calculate the performance of a portfolio rule without needing to derive analytically the expected out-of-sample performance with calculations.

We use the algorithm we develop and introduce some shrinkages for the mean and covariance matrix. These shrinkages include some parameters. Then using numerical optimisers we choose the parameters in order to optimise the expected out-of-sample performance of our input portfolio rule. We analytically explained the motivation for our shrinkages and we showed the improvement in the portfolio performance.

In the last part of the dissertation we explain how we can consider Kan-Zhou 2 fund and 3 fund portfolio rules as shrinkage estimators for the mean or for the weight function. This is an important result as one can use Kan-Zhou weight function formulas and extend them by combining shrinkages with the covariance matrix in order to achieve an even better performance. By comparing in the last chapter all the portfolio rules together we show that each rule works better in different situations.

In this project as well as the Kan and Zhou paper we can have a negative weight for an asset. An extension of this project would be to put a constrain to the weight function in order to prevent it to go negative i.e. to short sell an asset. This can be done using the introduced algorithm. This investigation will be very useful because when we are dealing for example with capital allocation we can't go short the capital.

Since we can easily measure the performance of any portfolio rule using the algorithm then we can try various combinations of shrinkages for different situations, bigger portfolios and find portfolio rules that might perform even better.

# Appendix

In this section we include the MATLAB code for calculating the expected out-of-sample performance of model 1 in section 5.1. The MATLAB code for all the rest portfolio rules is similar with the only difference in the shrinkages.

```
1 function U=model1(x)
2
3 T=100; %initialise T no. of observations
4 mu=[1.1 1 0.7]; %initialise mean vector of asset returns
5 sigma=[1 0.9 0.3;0.9 1 0.5; 0.3 0.5 1] %initialise covariance
6 %matrix for asset returns
7 nsim=1000; %initialise number of simulations
8 gamma=3; %initialise risk averse coefficient
9 N=length(mu); %define N equal to the number of assets
10 R=chol(sigma); %define R as the cholesky factorisation of sigma
11 U=0; %initialise utility equal to 0
12
13 for j=1:nsim %loop nsim times over the following algorithm
14 randn('seed',j); %set a seed in order to get different numbers
15 %in each iteration
16 z=repmat(mu,T,1)+randn(T,N)*R; %draw T observations from
17 %N dimensional normal distribution N(mu, sigma)
18 sigmaMLE=cov(z); %find the MLE cov. matrix
19 muMLE=mean(z); %find the MLE mean
20 diagSigma=diag(diag(sigmaMLE)); %observe the diagonal of
21 %sigmaMLE to use it for the shrinkage
22 avgmMLE = mean(muMLE); %observe the mean of returns
23 %to use it for the shrinkage
24 muHat=x(1)*muMLE+(1-x(1))*avgmMLE; %shrinkage (5.1)
25 sigmaHat=x(2)*sigmaMLE+(1-x(2))*diagSigma; %shrinkage (5.2)
26 w = x(3)*(1/gamma) * inv(sigmaHat) * muHat'; %shrinkage (5.3)
27 u = -(w'*mu' - (gamma/2)*w'*sigma*w); %find out-of-sample utility
28 %function (we put the minus because the function we use later
29 %to optimise the result finds the minimum so that
30 %the min of -ve is the max of +ve)
31 U=U+u; %sum the new utility to the sum of the previous
32 end
33
34 Uopt=U/nsim; %find the average of all the utilities and hence the
35 %expected out-of-sample performance
36 end
```

The function below is the one that uses the numerical optimisers in order to optimise the function model 1 above with respect to the input parameters.

```
1 function U=Uperformance()
2
3 %the command below optimises the function model 1 which has 3
4 %parameters. The function fminsearchbnd
5 %is responsible to find the optimal value for the
6 % parameters and also the optimal expected
7 %out-of-sample performance. The first term in the function
8 %fminsearchbnd denotes the function we wish to optimise.
9 %The first bracket defines from which number to start
10 %searching for the optimal solution.
11 %The second bracket denotes the lower bound for the 3
12 % parameters and the third the upper bound
13
14 [x,fval]=fminsearchbnd(@modell,[.5 .5 1],[0 0 0],[1 1 5]);
15 x %printout the optimal value for the parameters
16 modell=abs(fval) %printout the optimal expected
17 %out-of-sample performance
18 %it is in absolute as the fminsearchbnd will find
19 %the minimum value for modell which is negative
20 %but we want the same value to be +ve
21
22 %similarly for model 2
23 [x,fval]=fminsearchbnd(@model2,[.5 1],[0 0],[1 5]);
24 x
25 Utest=abs(fval)
26
27 %similarly for Kan-Zhou 3 fund rule
28 [x,fval]=fminsearchbnd(@KanZhou3,[1 1],[0 0],[5 5]);
29 x
30 UKZ=abs(fval)
31
32 end
```

# Bibliography

- [1] Raymond Kan and Guofu Zhou, Optimal Portfolio Choice with Parameter Uncertainty, *Journal of Financial and Quantitative Analysis*, 621-656, September 2007
- [2] Mark Kritzman, CFA, Sebastian Page, CFA, and David Turkington, CFA, In Defence of Optimization: The Fallacy of 1/N, *Financial Analysts Journal*, 2010
- [3] DeMiguel, V., L. Garlappi and R. Uppal, Optimal versus Naive Diversification: How inefficient is the 1/N Portfolio Strategy, *The Review of Financial Studies*, 22.5, 1915-1953, 2009
- [4] Vijay K. Chopra and William T. Ziemba, The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice, *Journal of Portfolio Management*, 1993
- [5] Michael Monoyios, Asset Pricing and Portfolio Theory, Lecture notes, *Mathematical Institute, University of Oxford*, November 2009