Assignment 3

xtj271 and cdv245

4/6/2021

1) Question 1

Consider the portfolio choice problem for transaction-cost adjusted certainty equivalent maximization with risk aversion parameter *gamma*:

$$w^*_{t+1} := \arg\max\{w'_{t+1}\mu - v(w_{t+1}, w^+_t, \beta) - \frac{\gamma}{2}w'_{t+1}\Sigma w_{t+1}\} \text{ s.t. } \iota' w = 1$$

where Σ and μ are (estimators of) the variance-covariance matrix of the returns and the vector of expected returns. Assume for now that transaction costs are quadratic in rebalancing and proportional to stock illiquidity such that:

$$v(w_{t+1}, \beta) := \frac{\beta}{2}(w_{t+1} - w^+_t)' B(w_{t+1} - w^+_t)$$

where $B = diag(ill_1, ..., ill_N)$ is a diagonal matrix where $ill_1, ..., ill_N)$ correspond to the Amihud measures provided to you. $\beta \in R_+$ is a cost parameter and $w^+_t := w_t \circ (1 + r_t)/\iota'(w_t \circ (1 + r_t))$ is the weight vector before rebalancing. $\circ$ denotes element-wise multiplication.

The problem can be rewritten as:

$$w^*_{t+1} := \arg\max_w \left\{ w'\mu - \frac{\beta}{2}(w - w^+_t)' B(w - w^+_t) - \frac{\gamma}{2}w'\Sigma w \right\} \text{ s.t. } \{\iota' w = 1\}$$

$$w^*_{t+1} := \arg\max_w \left\{ w'\mu - \frac{\beta}{2} \underbrace{\left[w'Bw - w'Bw^+_t - (w^+_t)'Bw + (w^+_t)'Bw^+_t\right]}_{(1\times1)} - \frac{\gamma}{2}w'\Sigma w \right\} \text{ s.t. } \{\iota' w = 1\}$$

$$w^*_{t+1} := \arg\max_w \left\{ w'\mu - \frac{\beta}{\gamma}\frac{\gamma}{2}w'Bw - \frac{2\beta}{2}w'Bw^+_t + \frac{\beta}{2}(w^+_t)'Bw^+_t - \frac{\gamma}{2}w'\Sigma w \right\}$$

$$w^*_{t+1} := \arg\max_w \left\{ w'\left[\mu - \beta Bw^+_t\right] - \frac{\gamma}{2}\left[\frac{\beta}{\gamma}w'Bw - w'\Sigma w\right] + \frac{\beta}{2}(w^+_t)'Bw^+_t \right\} \text{ s.t. } \{\iota' w = 1\}$$

and $\frac{\beta}{2}(w^+_t)'Bw^+_t$ is a (1×1) term that does not depend on w_{t+1}, such that the optimization problem with respect to w_{t+1} treats it as a scaling constant C, and disregards it:

$$w^*_{t+1} := \arg\max_w \left\{ w' \underbrace{\left[\mu - \beta Bw^+_t\right]}_{\mu^*} - \frac{\gamma}{2}w' \underbrace{\left[\frac{\beta}{\gamma}B - \Sigma\right]}_{\Sigma^*} w + C \right\} \text{ s.t. } \{\iota' w = 1\}$$

$$w^*_{t+1} := \arg\max_w \left\{ w'\mu^* - \frac{\gamma}{2}w'\Sigma^* w \right\} \text{ s.t. } \{\iota' w = 1\}$$

Now the optimum weights can be obtained analytically by differentiating the lagrangian:

$$
\begin{aligned}
&\arg\max_{w} L = w'\mu^* - \frac{\gamma}{2}w'\Sigma^* w - \lambda(w'\iota - 1)\\
&\frac{\partial L}{\partial w'} = \mu^* - \gamma\Sigma^* w - \lambda\iota = 0 \quad \rightarrow \quad w = \frac{1}{\gamma}(\Sigma^*)^{-1}(\mu^* - \lambda\iota)\\
&\frac{\partial L}{\partial \lambda} = w'\iota - 1 = 0 \quad \rightarrow \quad w'\iota = 1
\end{aligned}
$$

then combining equations **??** and **??** we derive λ:

$$
\begin{aligned}
1 &= \frac{1}{\gamma}\left((\Sigma^*)^{-1}(\mu^* - \lambda\iota)\right)'\iota\\
1 &= \frac{1}{\gamma}\left((\mu^*)'(\Sigma^*)^{-1}\iota - \lambda\iota'(\Sigma^*)^{-1}\iota\right)\\
\lambda &= \left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}\left((\mu^*)'(\Sigma^*)^{-1}\iota\right)
\end{aligned}
$$

and now plugging equation **??** into equation **??**:

$$
\begin{aligned}
w &= \frac{1}{\gamma}(\Sigma^*)^{-1}\left(\mu^* - \left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}\left((\mu^*)'(\Sigma^*)^{-1}\iota - \gamma\right)\iota\right)\\
w &= \frac{1}{\gamma}(\Sigma^*)^{-1}\mu^* - \frac{1}{\gamma}(\Sigma^*)^{-1}\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}\left((\mu^*)'(\Sigma^*)^{-1}\iota - \gamma\right)\iota\\
w &= \frac{1}{\gamma}(\Sigma^*)^{-1}\mu^* + \frac{\gamma}{\gamma}(\Sigma^*)^{-1}\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}\iota - \frac{1}{\gamma}(\Sigma^*)^{-1}\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}\left((\mu^*)'(\Sigma^*)^{-1}\iota\right)\iota
\end{aligned}
$$

now, using the fact that scalars are equal to their transpose, then we can rearrange (1×1) terms such that:

$$
\begin{aligned}
w &= \frac{1}{\gamma}(\Sigma^*)^{-1}\mu^* + \left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}((\Sigma^*)^{-1}\iota) - \frac{1}{\gamma}((\Sigma^*)^{-1}\iota)\underbrace{\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}}_{(1\times1)}\underbrace{\left((\mu^*)'(\Sigma^*)^{-1}\iota\right)'}_{(1\times1)}\\
w &= \left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}((\Sigma^*)^{-1}\iota) + \frac{1}{\gamma}\left[(\Sigma^*)^{-1}\mu^* - \underbrace{\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}}_{(1\times1)}((\Sigma^*)^{-1}\iota)\underbrace{\left(\iota'(\Sigma^*)^{-1}\mu^*\right)}_{(1\times1)}\right]\\
w &= \underbrace{\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}((\Sigma^*)^{-1}\iota)}_{w_{MVP}} + \frac{1}{\gamma}\left[(\Sigma^*)^{-1} - \underbrace{\left(\iota'(\Sigma^*)^{-1}\iota\right)^{-1}((\Sigma^*)^{-1}\iota)}_{w_{MVP}}\iota'(\Sigma^*)^{-1}\right]\mu^*
\end{aligned}
$$

The function "$optimal\_tc\_weight(w\_prev, mu, Sigma, beta, gamma, B)$" implements the analytical solution in Equation **??** to obtain the efficient portfolio weights. Note first, that we divide all returns by 100 because they were reported in percent but before we do so, we create a copy of the return dataframe that includes the date column (need for xts df for the GARCH). Note that compared to the results from Hautch et. al. 2019 the expression for w^*_{t+1} still has a similar structure as the the efficient portfolio weights without transaction costs but with a shifted μ^* and a shrunken Σ^* where the higher weights are incremented. Introducing the illiquidy measure B will affect both the shift in mean as well as the shrinkage of Σ^* as B replaces the identity matrix I as found in equation (5) and (6) in Hautsch et. al. 2019. The authors write that shrinkage of Σ can be achieved by replacing the identity matrix with a positive definite matrix - in this case B. Recall that B is defined as a diagonal matrix of the log of the Amihud measure for illudquidity.Now, the moments are scaled proportionally to their illiquidity.

2) Question 2

In Question 2 we are asked to illustrate the convergence towards the efficient portfolio according to Proposition 4 in Hautsch et. al. 2019. The proposition states that for $T \to \infty$ the optimal weights accounted for transaction costs will converge to a fixed point (the efficient minimum variance portfolio weights). For T approaching infinity, proposition 4 states:

$$w_{\text{inf}} = \left(I - \frac{\beta}{\gamma} A(\Sigma^*)\right)^{-1} w(\mu, \Sigma^*) = w(\mu, \Sigma)$$

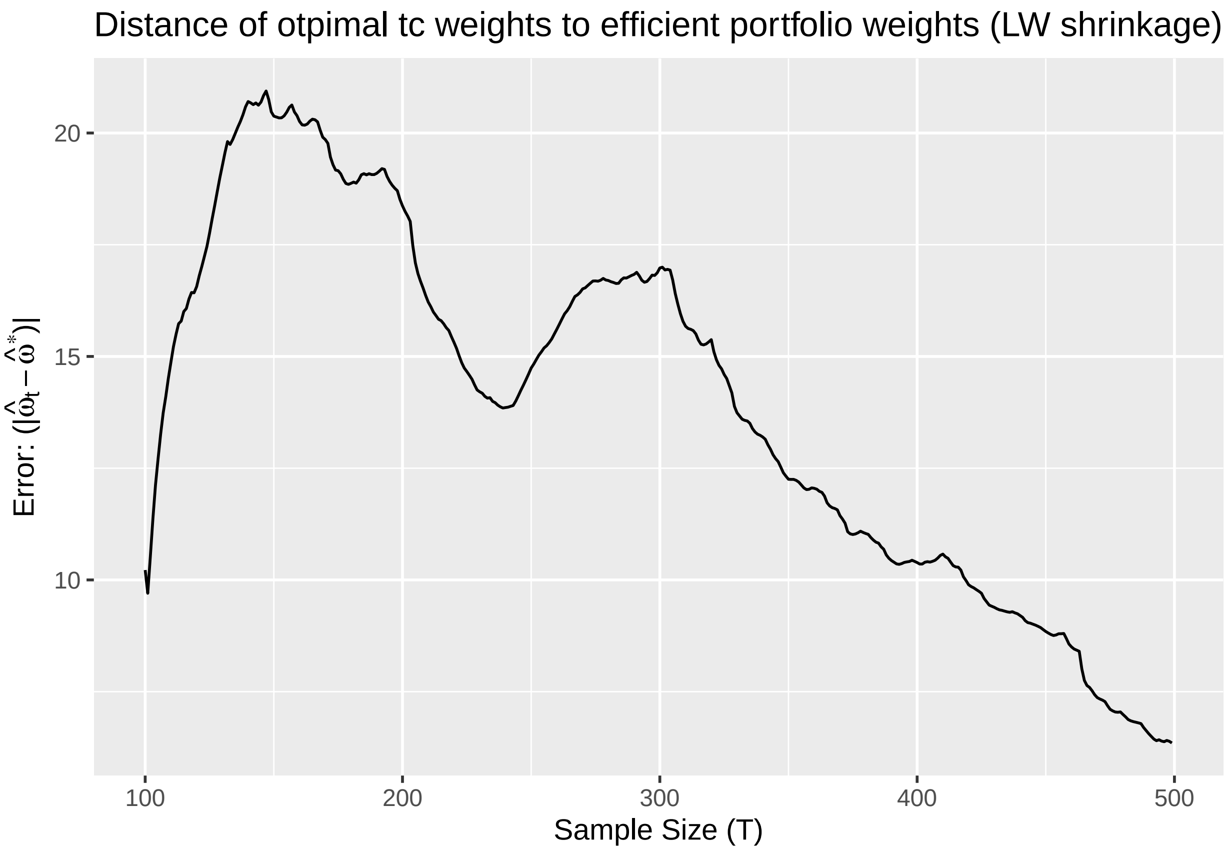
If an investor would start with the naive portfolio but would not face transaction costs, she could immediately shift to the efficient allocation by rebalancing her assets (without costs). The proposition claims that under transaction costs this convergence also happens if given enough periods where the hypothetical investor could rebalance her assets step by step.

The VCV matrix is estimated following Ledoit and Wolf (2003, 2004) such that:

$$\widehat{\Sigma}^{LW} := \alpha F + (1 - \alpha)\widehat{\Sigma}$$

where the highly overparametrized sample VCV ($\widehat{\Sigma}$) is shrunk towards a simpler VCV (F), to force insignificant coefficients to be zero and decrease the number of parameters to estimate towards more reasonable numbers.

As asked in the question we chose to measure convergence by calculating the sum of absolute distances between the optimal weights given transaction costs and the efficient portfolio weights. So we use a rolling window loop to estimate the optimal tc weights (where tc weights stand for weights optimal given transaction costs) for each period given an initial window length of 100. Instead of rolling the window we use an expanding window so for each new iteration it will take more previous periods into account and thus T increases for each iteration. For the first iteration the weights are initialized as the naive portfolio weights but this variable w_{prev} will be overridden at the end of each iteration with the calculated optimal tc weights. As stated in the question we ignore the price distortion effect and do not adjust the weights like we do in later exercises. Before the loop we initialize a collect matrix to store results. Within the loop we calculate Σ using the Leodoit-Wolf shrinkage and call our function from Question 1 *optimal_tc_weight()* to calculate the corresponding optimal tc weights. We then calculate and store the error (as defined as the sum of absolute differences over the two weights vectors). After the loop has run, we plot the collected results in Figure 1:



As can be seen in the figure the distance between the two weight vectors is decreasing in T. Interestingly, for T around 300 the distance increases again but then falls for even larger T. We can't really show convergence because the absolute distance is still decreasing for T = 500 which is the maximum number of observations in our sample. Possible robustness checks could be to compare the convergence with a L2 (so quadratic error) or with a shorter window size. We estimate 400 points because our initial window has length of 100 periods. What we did check and confirm is that the absolute error for the shrinkage is lower compared to using the simple sample covariance matrix, so we only report the shrinkage errors.

3) Question 3

Before we start with the backtest, we define a GARCH(1,1) model using the *rmgarch* and *parallel* packages. More specifically we use a DCC-GARCH specification which calculates the conditional correlation dynamically. We then use the *dccroll* function from the *rmgarch* package to estimate the 1-period-ahead forecast for Σ for each period in our loop (250 periods with a fixed rolling window size of 250). The resulting covariance has the shape [N,N,250] and contains the Σ_{t+1} for each out of sample period.

The GARCH(1,1) allows to specify the $(N \times 1)$ vector of log-returns at time (t) with time-varying conditional variance:

$$r_t = E(r_t|F_{t-1}) + \epsilon_t, \quad \text{where} \quad \epsilon_t = \Sigma_t^{1/2} z_t, \quad \text{and} \quad \Sigma_t = \Sigma_0 + \theta\epsilon_{t-1} + \eta\Sigma_{t-1}$$

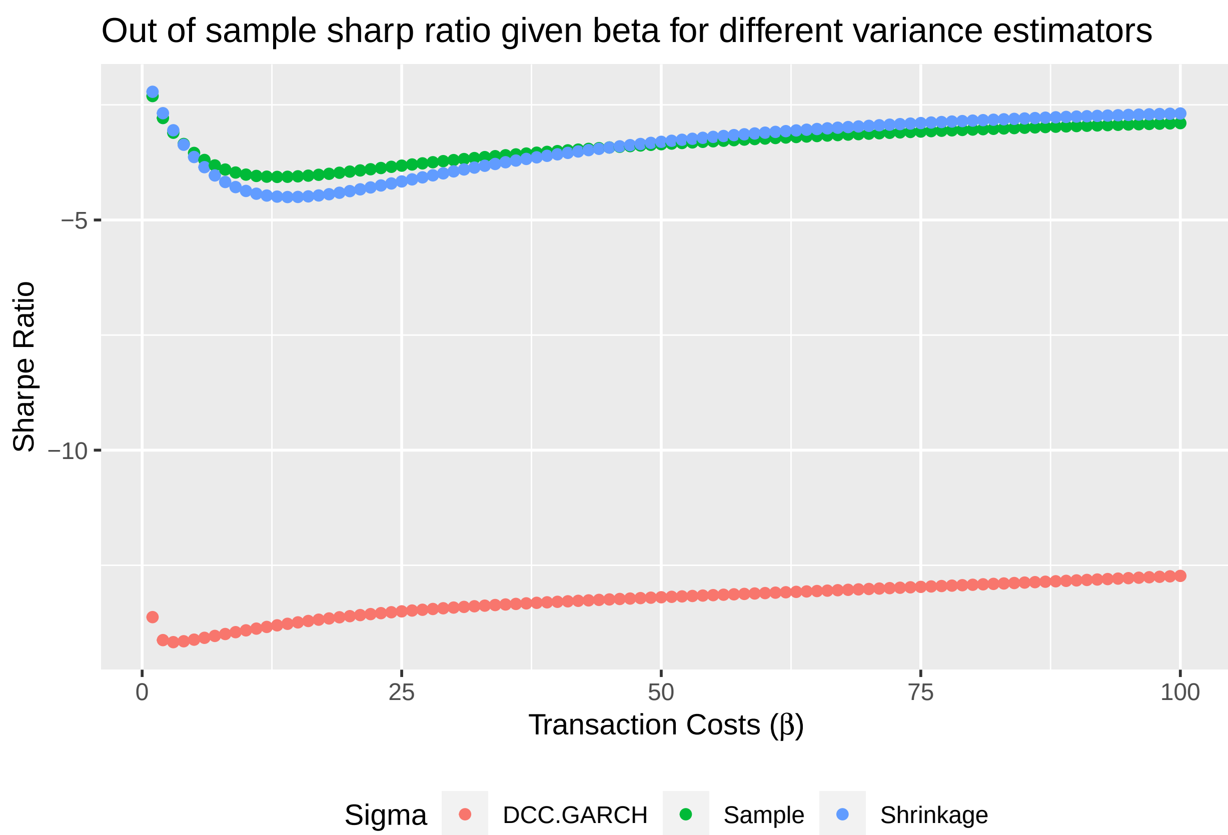
where $E(r_t|F_{t-1}) = \mu$, $z_t \sim iidN(0_N, I)$. However, in order to ensure positive definiteness of the variance covariance and allow its estimation, it can be re-expressed as a DCC-GARCH(1,1) where both conditional correlations and conditional standard deviations are time-varying:

$$t = E(r_t|F_{t-1}) + \epsilon_t, \quad \text{where} \quad \epsilon_t = \Sigma_t^{1/2} z_t, \quad \text{and} \quad \Sigma_t = D_t R_t D_t$$

where $D_t = diag(\sigma_{i,t})$ is a $(N \times N)$ diagonal matrix of individual-returns'-standard-deviations, and R_t is a $(N \times N)$ symmetric matrix with conditional correlations $\rho_{i,j,t}$ where $\rho_{i,i,t} = 1$ and $\rho_{i,j,t} = \rho_{j,i,t}$.

To answer the question we define a function called *sharp_ratio()* that takes our returns data, the transaction cost parameter β, the risk aversion parameter γ and the illiquidity measure B as inputs. Note that β is set to 1 and γ is set to 4 as defaults. Our function recomputes the optimal weights each day (taking the last 250 observations into account for estimating Σ and μ). This is again done by passing the estimated moments, the fixed β and γ and the vector of previous weights to our function from Question 1 *optimal_tc_weight()*. These weights are then used to calculate estimated portfolio returns, the rebalancing turnover relative to the previous weights and the net returns accounting for rebalancing costs. Note that this process is repeated 3 times for each iteration because we compare 3 different estimation strategies for the covariance matrix. Specification 1 will calculate Σ as the simple covariance matrix from the previous 250 observations, specification two will compute the Ledoit-Wolf shrinkage covariance matrix and specification 3 will use the estimated DCC-GARCH forecast as Σ (this is done by accessing the NxN matrix stored in the [N,N,250] shaped rcov object that we got from dccroll at index [N,N,i]. All results are stored in a previously initialized list. This list of lists is then used to calculate the out-of-sample sharp ratio for the given value of β by the provided formula of dividing oos mean by oos standard deviation.

Finally, we loop over our *sharp_ratio* function to call our function with different values of β (ranging from 1:100) and collecting the resulting sharp-ratios for each of the 3 strategies. The plot in figure 2 illustrates the relationship between sharp ratio and transaction costs:



It can be seen that estimating Σ with either a DCC-GARCH or with the LW-Shrinkage produces higher sharp ratios compared to just using the sample covaraince matrix. The estimated sharp ratios seem to increase with increasing transaction costs but are negative for all values of $\beta > 1$.

4) Q4

For Question 4 we again conduct a backtesting strategy with 250 periods and calculate net returns and turnover for each iteration. In contrast to question 3 we do not compare different estimator for Σ but 3 different portfolios. The first portfolio is as before the one optimal under transaction costs. Note that transaction costs are now assumed to be an L1 penalization and not quadratic as in question 1. This means that there is no longer a closed form solution for the optimal tc weights so we do not use the already introduced *optimal_tc_weight()* function. Instead we call the *constrOptim.nl* optimizer form the *alabama* package. The relevant inputs for the optimizer are an objective function (we chose to implement equation 13 from Hautsch et. al. 2019) as well as a constraint (denoted as *heq* in the code) which in our case is that $\iota' w = 1$. The second portfolio will rebalance to the naive portfolio ($1/N$) in each period and the third portfolio will calculate the minimum variance portfolio under a no short-selling restriction. Recall that we implemented the no short-selling restriction with a numerical optimizer, in our case the *solve.QP* routine from the *quadprog* package. All 3 portfolios start with the naive portfolio weights as initial previous weights. In each iteration the Ledoit-Wolf shrinkage operator is used to estimate the covariance matrix. We also account, in contrast to question 2, for the distortion effect of prices on weights. This means that our optimal weights are used to calculate net returns but before they are stored as the next periods previous returns, we adjust them by the realized returns in that period. Note that this procedure is only done for portfolio 1 and 3 because portfolio 2 will by construction always rebalances to $1/N$ anyways.

As can be seen in the table, the estimated results are exactly the same for the naive portfolio and our L1 transaction cost weights. This indicates that our solver did not start properly. A test on the weights which the alabama optimizer spit out seems to indicate that the condition that the sum over all weights (*sum(sol$solution)* $= 1$) holds. If valid, our results would suggest that under L1 penalization due to transaction costs, a naive portfolio strategy would be optimal and would outperform the no short-selling strategy with about double sized mean net returns and sharp ratios.

Table 1: Portfolio Performance Overview

strategy	Mean	SD	Sharpe	Turnover
MV (TC)	0.3472	0.4236	0.8194	0.0000
Naive	0.3472	0.4236	0.8194	0.0000
MV (no short-selling)	0.1296	0.2939	0.4408	0.1076

Table 2: Weight vector for last iteration of constrOptim.nl

w
0.0000
0.0000
0.0183
0.0000
0.0703
0.0000
0.0036
0.0000
0.0000
0.0000
0.0000
0.0326
0.0000
0.0000
0.0000
0.0000
0.0000
0.5207
0.0194
0.0907

w
0.0000
0.0000
0.0123
0.0000
0.0000
0.0000
0.0000
0.0000
0.2321
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

```
## [1] "sum of weights: sum(sol$solution)"
```

```
## [1] 1
```