SPL presentation

Minimum Variance Portfolio

Data

The data being used is provided by Kenneth French. The well known dataset '10 Industry Portfolios' contains 10 portfolios of different US-companies.

Loading the data

Read 'average value weighted returns (monthly)' from line 12 to line 1121

```
return_matrix = read.csv('10_Industry_Portfolios.CSV', header=TRUE, sep=',', skip = 11, nrows = 1109)
```

Computing the Covariance Matrix

As we saw in the section above, to minimize the overall variance, we first need to compute the covariance matrix. This can be achieved by applying the cov() function to the rows (t-th row for time t) of return_matrix.

```
nA = dim(return_matrix[,-1])[2] # number of assets
cov_return_matrix = cov(return_matrix[,-1]) # dates are deleted
cov_return_matrix
```

```
##          NoDur    Durbl    Manuf    Enrgy    HiTec    Telcm    Shops
## NoDur 20.87855 25.78169 24.03555 17.07151 24.18262 14.20811 22.88915
## Durbl 25.78169 58.57261 41.47241 28.62881 42.72165 22.05227 35.31515
## Manuf 24.03555 41.47241 38.69149 27.70223 38.67405 19.49894 30.59999
## Enrgy 17.07151 28.62881 27.70223 36.93934 26.80968 14.33150 20.70935
## HiTec 24.18262 42.72165 38.67405 26.80968 52.08515 22.47536 33.18733
## Telcm 14.20811 22.05227 19.49894 14.33150 22.47536 21.01664 18.00944
## Shops 22.88915 35.31515 30.59999 20.70935 33.18733 18.00944 33.67309
## Hlth  20.19569 26.98578 26.21504 18.93730 28.85347 15.36386 23.98317
## Utils 17.73483 25.74777 23.66667 20.35171 24.41621 15.80322 20.68618
## Other 24.47136 39.43675 36.06344 26.87643 36.87453 20.58528 30.65570
##           Hlth    Utils    Other
## NoDur 20.19569 17.73483 24.47136
## Durbl 26.98578 25.74777 39.43675
## Manuf 26.21504 23.66667 36.06344
## Enrgy 18.93730 20.35171 26.87643
## HiTec 28.85347 24.41621 36.87453
## Telcm 15.36386 15.80322 20.58528
## Shops 23.98317 20.68618 30.65570
## Hlth  30.89214 18.87372 26.22910
## Utils 18.87372 30.26306 25.25974
## Other 26.22910 25.25974 40.85289
```

All positive values in the cov_return_matrix, meaning there is at least some linear relation between all the values.

Table 6.1 Details for 10 industry portfolios

1	NoDur	Consumer NonDurables: Food, Tobacco, Textiles, Apparel, Leather, Toys
2	Durbl	Consumer Durables: Cars, TV's, Furniture, Household Appliances
3	Manuf	Manufacturing: Machinery, Trucks, Planes, Chemicals, Off Furn, Paper, Com Printing
4	Enrgy	Oil, Gas, and Coal Extraction and Products
5	HiTec	Business Equipment: Computers, Software, and Electronic Equipment
6	Telcm	Telephone and Television Transmission
7	Shops	Wholesale, Retail, and Some Services (Laundries, Repair Shops)
8	Hlth	Healthcare, Medical Equipment, and Drugs
9	Utils	Utilities
10	Other	Other: Mines, Constr, BldMt, Trans, Hotels, Bus Serv, Entertainment, Finance

Figure 1: Explanation 10 Industry Portfolios

By evaluating the restrictions and definitions of the minimization-problem, you can define a Lagrangian, which can later be transposed into a linear system of equations of form

$$A \cdot x = b.$$

Since $A = \Sigma$ and $b = 1$ are known, we can solve this system to obtain $x = w_t$.

```
b = vector(length=nA) + 1
x = solve(cov_return_matrix, b)
x = x/sum(x)
x
```

```
##       NoDur       Durbl       Manuf       Enrgy       HiTec       Telcm
##  0.75270484 -0.05959476 -0.13776060  0.21510686 -0.09869360  0.53630598
##       Shops        Hlth       Utils       Other
## -0.05085601  0.07921599  0.08130649 -0.31773519
```

Note that negative weights are interpreted as *short sales*.

Performance anaylsis

To evaluate the performance, the analysis from deMiguel will be mimiced. DeMiguel et al. used a rolling-sample approach, where over a time period $T = 120$months, returns of the assets were being calculated. In each month t, starting form $t_0 = M + 1$, the data from the previous M months was used to calculate the portfolio-weights w_t. By adding the return of the next period and dropping the earliest return, the whole dataset is processed, resulting in a series of $T - M$ monthly *out-of-sample* returns generated by each strategy.

Then, the *out-of-sample Sharpe ratio* is computed. It consists of the sample mean of out-of-sample excess return μ_k, devided by their sample standard deviation σ_k:

$$SR = \frac{\mu_k}{\sigma_k}$$

```r
nRows = dim(return_matrix)[1] # no of months
M = 120 # size of rolling-sample
t = M + 1 # adjust t_0 to size of rolling-sample
mu = c() # empty vector of excess-return
while (t <= nRows-1) {
  tStart = t - M # lower barrier for the time-interval
  tEnd = t # upper barrier for the time-interval
  return_matrix_t = return_matrix[c(tStart:tEnd),] # return matrix for time-interval
  return_matrix_future = return_matrix[c(tEnd+1),] # return matrix one month in the future
  cov_return_matrix_t = cov(return_matrix_t[,-1]) # calculate cov-matrix
  b_t = vector(length=nA) + 1 # set up system of equations
  x_t = solve(cov_return_matrix_t, b_t) # and solve it
  x_t = x_t/sum(x_t) # normalize weights
  # get out-of-sample excess return for time T
  mu = c(mu, x_t %*% as.numeric(return_matrix_future[1,-1]))
  t = t+1
}
head(mu)
```

```
## [1]  0.5398844 -1.2541664  2.0386722  3.9867465 -1.6300491 -2.5127751
```

```r
mean(mu)
```

```
## [1] 0.8339652
```