Exam Project 1: Asset Liability Management in a pension fund

Tue Vissing Jensen, DTU Elektro* Tiago Soares, DTU Elektro†

December 4, 2014

1 Nomenclature

$t \in T = 0, \ldots, \tau$ Time steps, measured relative to current year.

$i \in E$ ETF index

$m \in M$ Months subset index of T

$T_m \subset T$ Specified interval dates

$s \in \Omega$ Scenarios index

$st \in ST$ Index over weeks included in monthly scenarios

$P_{i,t}$ Price of ETF i at time t

$r_{i,t}$ Historical weekly return for asset i at week t.

$r^m_{i,t}$ Historical monthly return for asset i at week t.

$MRS_{i,s,m}$ Monthly return for asset i in scenario s in month m.

2 Data download and clustering analysis

Price data on the ETFs is downloaded from Google Finance using the data download functionality built in to the Pandas module for Python. We use the closing price of the ETF for each day to represent the day's price. Any ETF with less than 2400 data points is excluded, leaving 93 of the initial 100 ETFs to use for further processing.

To cluster the data, it is necessary to define a distance between assets. We compute the correlation coefficient C_{ij} between the log returns of assets i and j, and define the distance between them as

$$d(i,j) = 1 - C_{ij}^2 \in [0,1]. \tag{1}$$

This distance will tend to group together assets that are more correlated and/or anti-correlated together. This is intuitively a reasonable measure, as (1) going long on an asset is the same as going short on an asset that is anti-correlated with it, and we wish to treat short and long assets the same in this analysis, and (2) the difference between a correlation of 1.00 and 0.95 is more significant than between 0.05 and 0.00.

To do hierarchical clustering, a method of combining clusters must be defined. We compare the results of using the various methods built in to the SciPy module for Python: complete, average, weighted and single. A dendrogram of the clusters found by these methods is shown on Figure 1. Both the complete, average and weighted methods lead to faily balanced trees. We choose to use the complete method, as the final clusters found by this method are the most uniform in size. We choose to use only 15 clusters for the current work, since going to 25 clusters would lead to many single-asset cluster while not breaking up the largest cluster in Figure 1 — an indication that the assets in this cluster are very similar.

We can further examine the clusters by decomposing C into its principal components PC_i, and looking at the projection of the columns of C onto the first two. This will show

*tvjens@elektro.dtu.dk
†tiasoar@elektro.dtu.dk

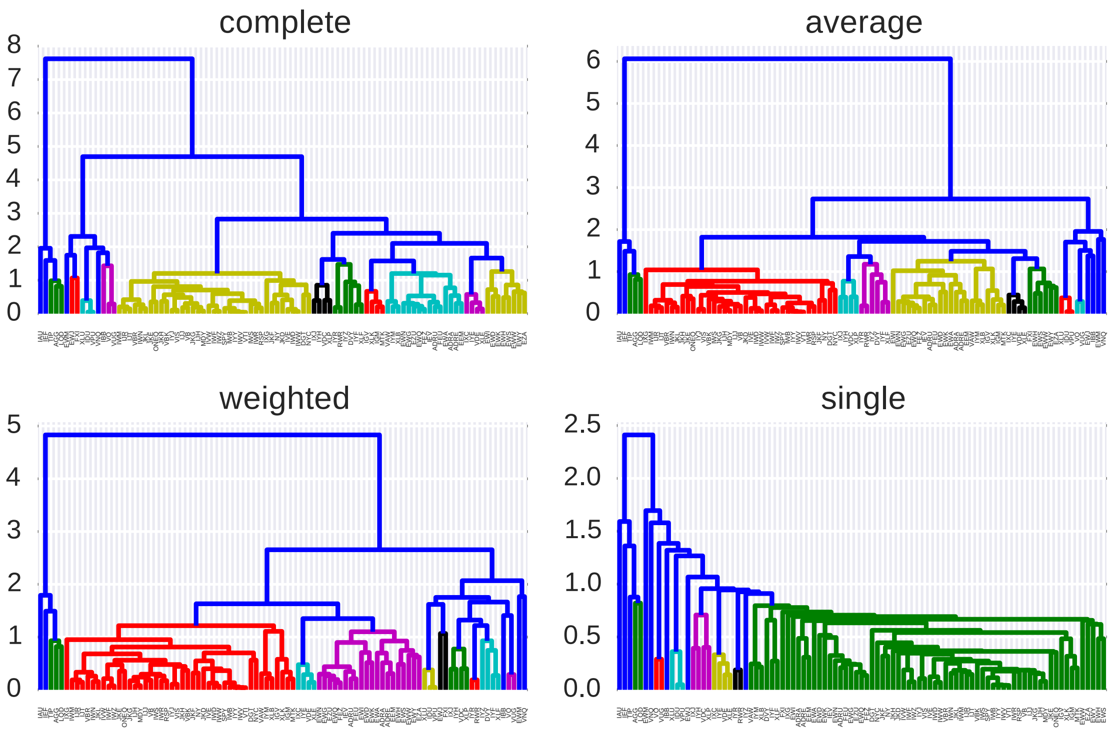

Figure 1: Dendrogram of clusters found by various methods. All non-blue leaves are trimmed by the max-return and the min-std criteria.

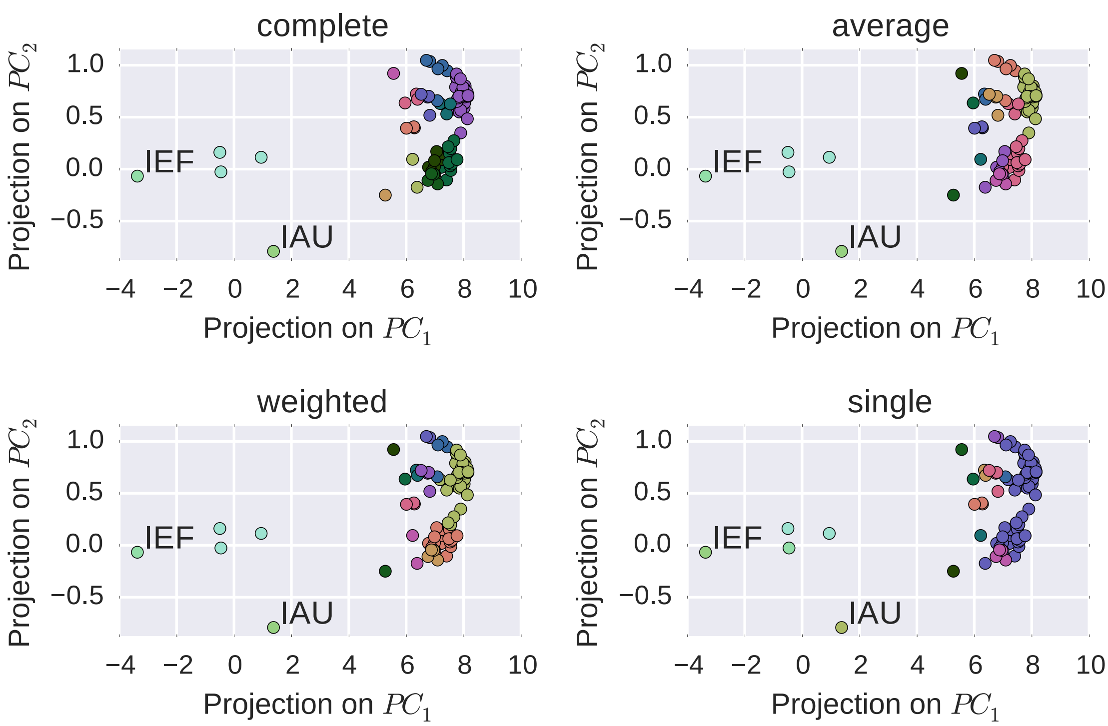

Figure 2: Projection of asset correlation onto the largest principal components. These directions account for 78.58% and 4.51% of the variance, respectively.

a slice of the space defined by $d(i,j)$, and reveal which assets tend to move in a similar manner across the time series. This slice is shown in fig 2, and clearly shows that 5 assets have radically different behaviour than the others. In particular, IAU, which tracks the gold price, and IEF, which tracks US treasury bonds, appear to move in a manner inconsistent with the overall market, and are hence clustered differently to these.

In order to prune the tree of assets, we pick from each cluster a single representative asset in two ways. The max-return criterion picks out the asset which has the highest return across the period, while the min-stdev criterion pick the asset with the smallest standard deviation, i.e. the most stable asset. The assets selected by these two criteria and their associated price histories for the entire period are shown in Figure 3.

3 Scenario generation

The period of study considered in this work is between 02/02/2005 and 12/11/2014. In order to start the scenario generation to be used on CVaR model, were established several assumptions. The historical return for each ETF over the period t is given by

$$r_{i,t} = \frac{P_{i,t+1}}{P_{i,t}} - 1, \ \ \forall i \in E, t \in T \tag{2}$$

The scenarios are randomly generated picking four dates in a specified interval and taking the historical returns for the ETF's as

$$WRS_{i,st,s,m} = r_{i,t} \text{ for } t \equiv RANDOM(T_m), \ \ \forall i \in E, st \in ST, s \in \Omega, m \in M \tag{3}$$

where $WRS_{i,st,s,m}$ is the weekly return scenarios for each month m. The first specified intervalT_1 where is picked the historical returns is between 28/01/2005 and 28/02/2008. For each m the specified interval shift itself one month until the end of the period of study (12/11/2014). The monthly return for each of scenario $MRS_{i,s,m}$ is based on accumulating the four weeks return scenario and is determined as

$$MRS_{i,s,m} = \prod_{st}(1 + WRS_{i,st,s,m}) - 1, \ \ \forall i \in E, s \in \Omega, m \in M \tag{4}$$

4 CVaR model

The CVaR model was built based on the first set of scenarios with start date 2008-02-27, obtained from scenario generaion chapter. Based on this set is determined the value and expected value of each ETF i, for each scenario s. In addition, is considered a initial budget of 1 million kr. The CVaR model is defined as

$$\sum_i x_i = \text{Budget} \tag{5}$$

$$\text{MeanReturn} \geq \mu_{\text{Target}}\text{Budget} \tag{6}$$

$$\text{VaRDev}_s \geq \text{Losses}_s - \text{VaR} \ \ \forall s \tag{7}$$

$$\text{Losses}_s = \sum_i x_i - \sum_i \text{P}_{i,s} x_i \ \ \forall s \tag{8}$$

$$\text{CVaR} = \text{VaR} + \frac{1}{1-\alpha} \sum_s \text{pr}_s \text{VaRDev}_s \tag{9}$$

$$\text{MeanReturn} = \sum_i \text{EP}_i x_i \tag{10}$$

where μ_{Target} is 0, $\text{P}_{i,s}$ is the value of each ETF i by scenario s, pr_s is the probability of each scenario s and is assumed to be linearly distributed over the scenarios, EP_i is the expected value for each ETF i, and α is assumed to be 0.5. In order to obtain the efficient frontier based on 10 optimal solutions of CVaR model is neccesary to find the minimum CVaR solution, as well as the CVaR solution related to the maximum possible return. Minimizing the model in order to CVaR variable, it returns the minimum CVaR solution of efficient frontier. Maximizing the model in order to MeanReturn variable, it returns the CVaR solution for the maximum average return. Based on this extreme points, a linear $\text{CVaR}_{\text{Target}}$ is created for the 10 runs. A constraint for limit the space solution movement to the $\text{CVaR}_{\text{Target}}$ is included in the CVaR model, and is defined as

$$\text{CVaR} \leq \text{CVaR}_{\text{Target}} \tag{11}$$

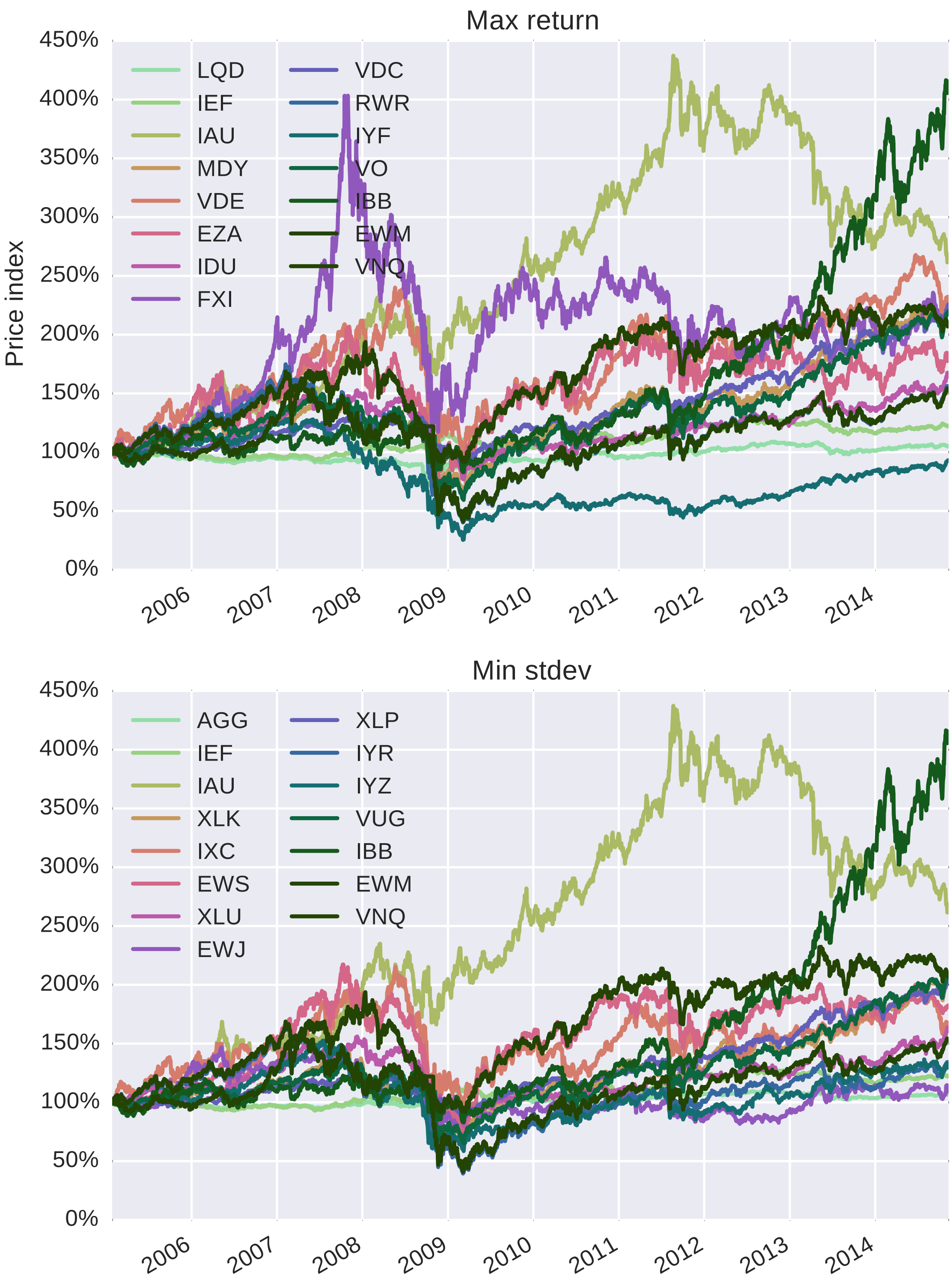

Figure 3: Price history for the assets selected by the max-return (top) and the min-stdev criteria (bottom).

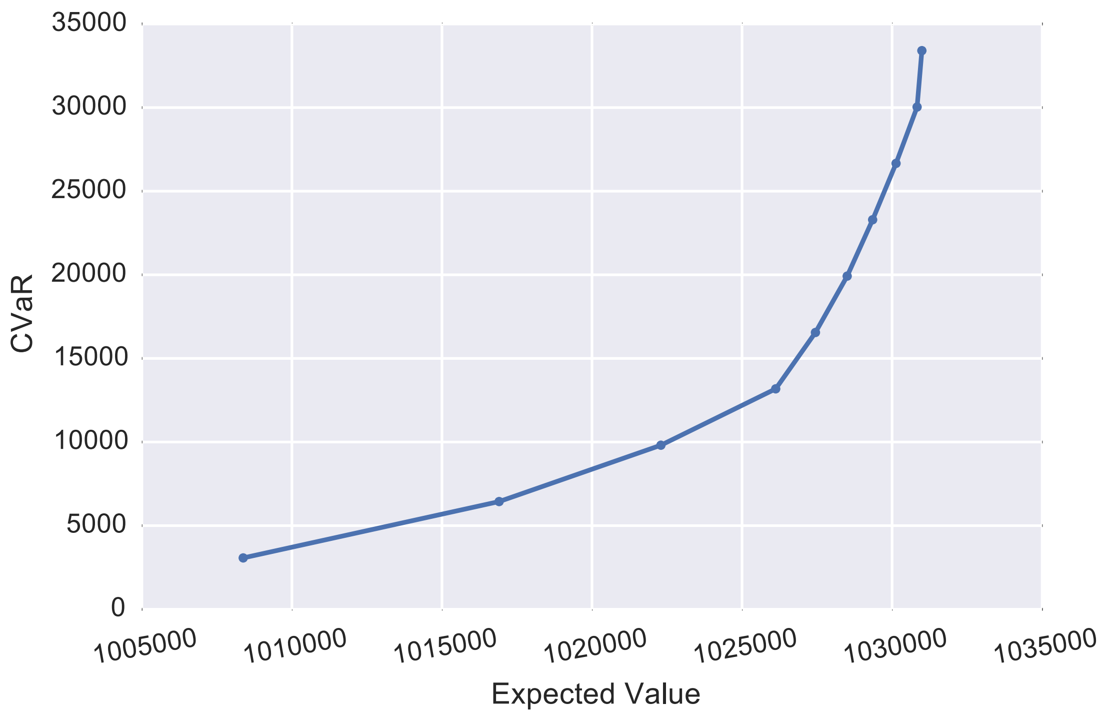

Figure 4: Optimal frontier for equidistant steps in CVaR.

For each of the 10 runs, the CVaR model is optimized in order to maximize the MeanReturn variable. The efficient frontier of the 10 optimal solutions is presented in Figure 4.

A CVaR bound of 0.0 indicates that CVaR is minimal, with 1.0 indicating that the optimization solely considers mean return.

Figure 6 compares the scenario return at varying levels of the CVaR bound.

5 Portfolio revision implementation

The portfolio revision model is an extension of the CVaR model to revise and evaluate portfolios after some time has past. In this work, is assumed that four weeks are gone and the previous portfolio on section 4 could no longer be credible and acceptable. So, a revision of the portfolio on the (2008-03-28) to maximize the return under two different strategies (risk averse and risk neutral) are considered. The portfolio revision entails some costs for the transactions deviations on the portfolio. In this way, is assumed a transaction cost of 0.1% of the traded amount in kr., assuming at least 50 kr per trade.

The portfolio revision model is defines as

$$
\begin{aligned}
\text{TotalCost} &= \lambda \text{CVaR} - (1-\lambda)\text{MeanReturn} + \text{TradeCost} && (12)\\
x_i^{\text{old}} + x_i^{\text{difference}} &= x_i \quad \forall i && (13)\\
\sum_i x_i^{\text{old}} &= \sum_i x_i && (14)\\
x_i^{\text{difference}} &\leq B_i \text{M} \quad \forall i && (15)\\
-x_i^{\text{difference}} &\leq B_i \text{M} \quad \forall i && (16)\\
\text{TradeCost} &= \sum_i \text{TC}_i && (17)\\
\text{TC}_i &\geq B_i \text{penalty} \quad \forall i && (18)\\
\text{TC}_i &\geq 0.001 x_i^{\text{difference}} \quad \forall i && (19)\\
\text{TC}_i &\geq -0.001 x_i^{\text{difference}} \quad \forall i && (20)\\
& && (21)
\end{aligned}
$$

where λ is the level of risk of the strategy (1 - Risk Averse ; 0 - Risk Neutral), TradeCost is the total cost of the trades of all changes in portfolio, and x_i^{old} is the previous result of

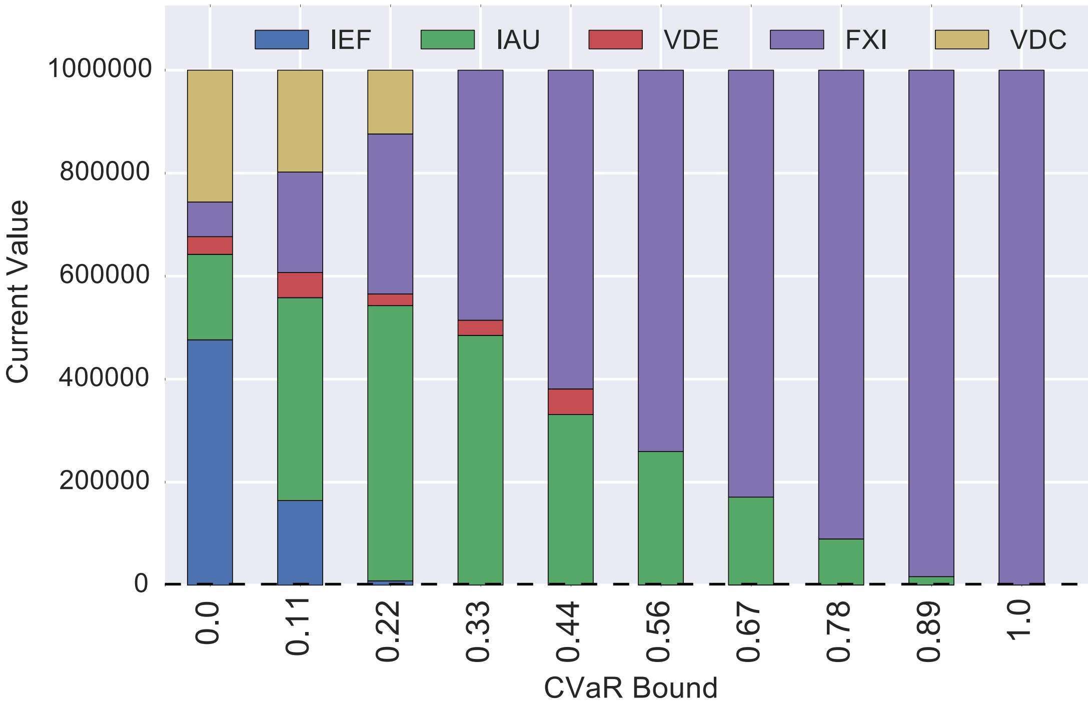

Figure 5: Portfolios at varying levels of the CVaR bound.

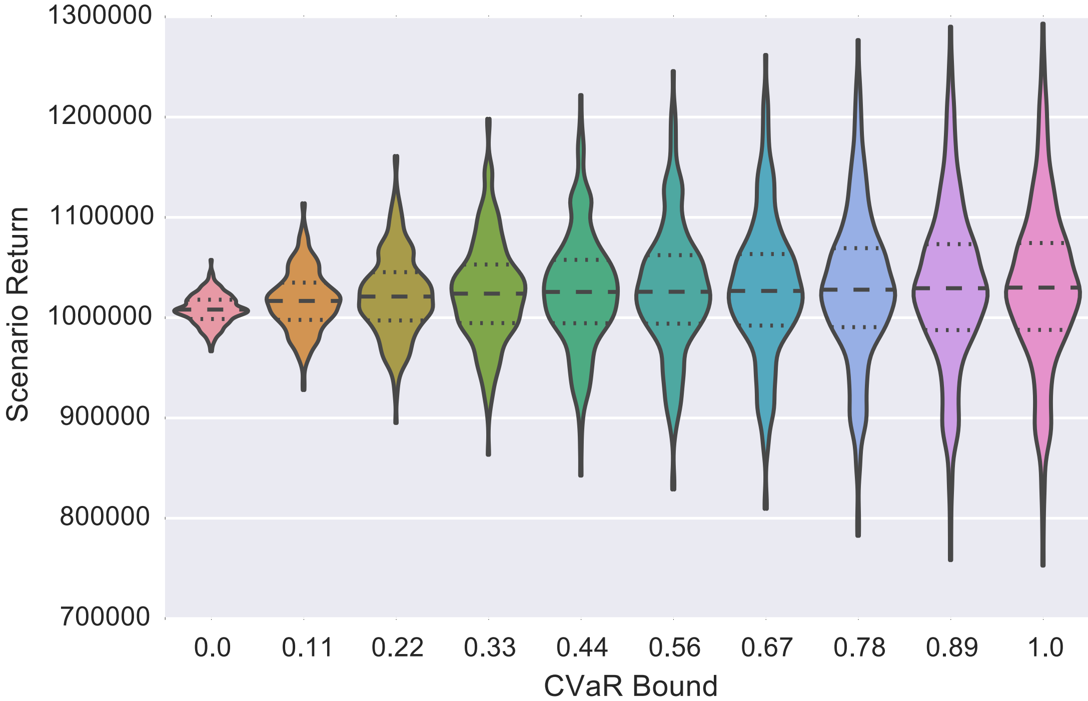

Figure 6: Scenario return for portfolios at varying levels of the CVaR bound. The distributions for each bound are mirrored on the vertical axis, with the mean (dashed) and standard deviation (dotted) shown.

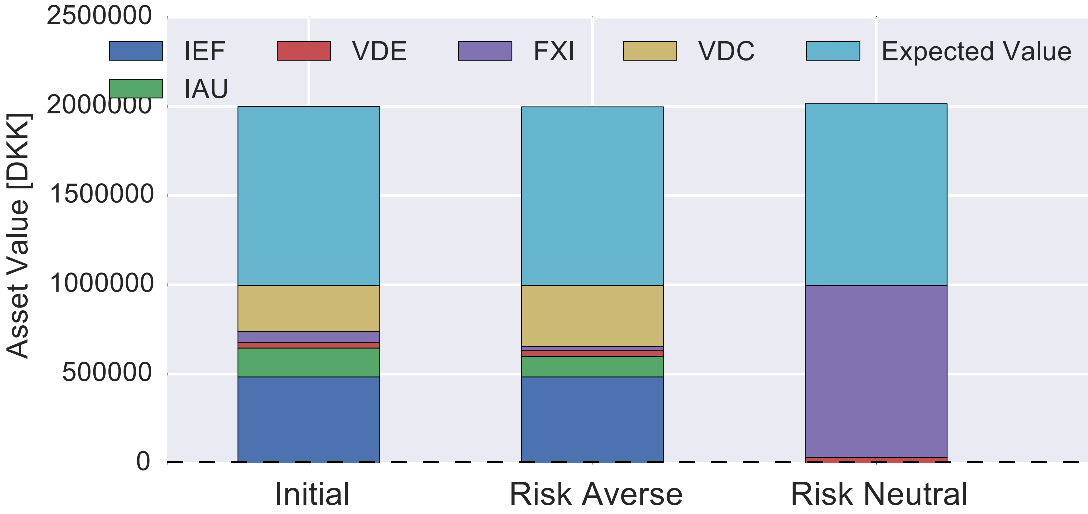

Figure 7: Initial portfolio versus those found by the two types of traders.

Table 1: Stats for portfolios found by portfolio revision model.

	Expected Profit	CVaR	Trading Cost
Type			
Initial	8423.76	3436.86	0.00
Risk Averse	6545.42	2469.89	214.37
Risk Neutral	24735.63	36958.75	1820.37

portfolio updated with the current historical monthly return. In this way, the first x_i^{old} is based on the results of section 4 considering the run related to risk averse. The difference between the x_i^{old} and the new x_i is given by $x_i^{\text{difference}}$. B_i is a binary variable to characterize if there are or not a transacation by each ETF i, and M is a high constant value to allow the selection of trade or not trade. TC_i is the tradde cost by each ETF i, and penalty corresponds to the trade cost of 50 kr. In addition to the previous constraints, the CVaR model presented in section 4 is considered as constraints in the portfolio revision model.

The aim of the present section is to minimize the TotalCost of the full portfolio revision model for each type of strategy (risk averse and risk neutral). The results concerning the expected value, CVaR and assets portfolio are presented in Table1 considering a comparison between the different strategies and the initial solution.

Results...

6 Back-test results

In the $1/N$ strategy, on the first trading day we invest an equal amount into each asset, and then leave the assets be for the entire trading period.

Since we are investing into 10 assets at a cost of 0.1% of the trading volume, it costs us 1000 DKK to make the initial trade. The remaining 999.000 DKK are left to grow during the entire period.

This strategy yields an annualized return of 5.7% for the max-mean assets, and of 2.4% for the min-stdev.

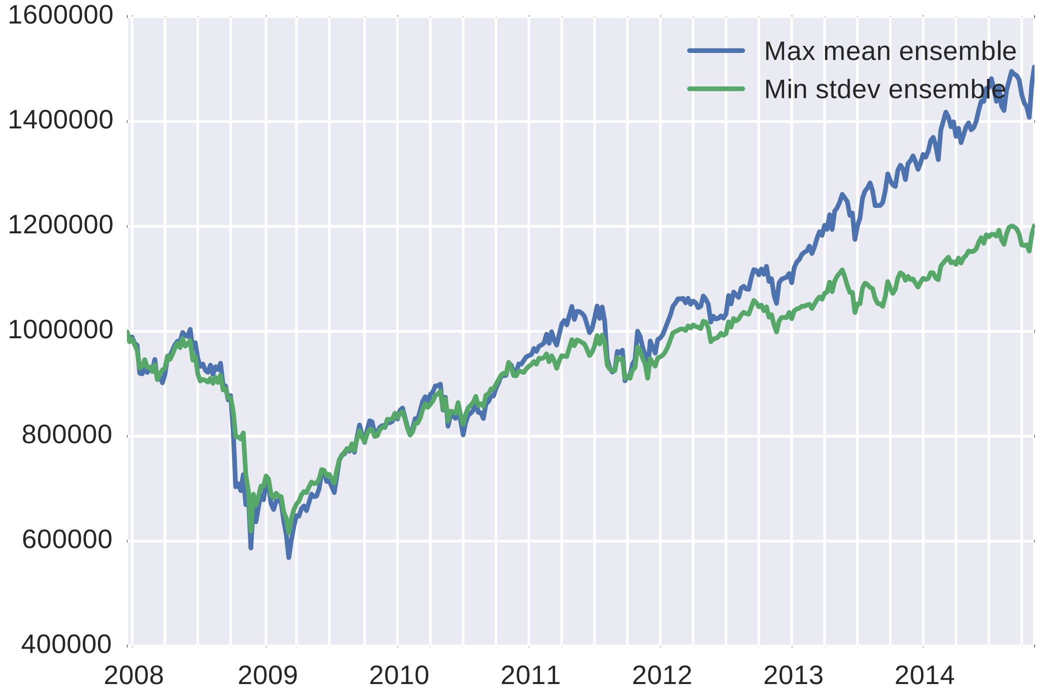

Figure 8: DELETE ME IN FINAL: Returns for the 1/N strategy for the ensembles indicated.

7 Scenario generation via moment matching

For the moment matching scenarios, we are looking to match the mean, variance, skewness and kurtosis with historical data. Calculating these for each month m as

$$\mu_{m,i} = \frac{1}{|T_m|} \sum_{t \in T_m} r_{i,t}^m \tag{22}$$

$$\beta_{m,i} = \frac{1}{|T_m|} \sum_{t \in T_m} \left(r_{i,t}^m - \mu_{m,i}\right)^2 \tag{23}$$

$$\gamma_{m,i} = \frac{1}{|T_m|} \sum_{t \in T_m} \left(r_{i,t}^m - \mu_{m,i}\right)^3 \tag{24}$$

$$\eta_{m,i} = \frac{1}{|T_m|} \sum_{t \in T_m} \left(r_{i,t}^m - \mu_{m,i}\right)^4, \tag{25}$$

we solve the following optimization problem to find the scenario sets $\{\xi_{i,s}\}_m$:

$$\min \sum_i \left(\left(\tilde{\mu}_i - \mu_{m,i}\right)^2 + \left(\frac{\tilde{\beta}_i}{\beta_{m,i}} - 1\right)^2 + \left(\frac{\tilde{\gamma}_i - \gamma_{m,i}}{\beta_{m,i}^{3/2}}\right)^2 + \left(\frac{\tilde{\eta}_i - \eta_{m,i}}{\beta_{m,i}^2}\right)^2 \right) \tag{26}$$

S.t.

$$\tilde{\mu}_i = \frac{1}{|\Omega|} \sum_{s \in \Omega} \xi_{i,s} \quad \forall i \tag{27}$$

$$\tilde{\beta}_i = \frac{1}{|\Omega|} \sum_{s \in \Omega} \left(\xi_{i,s} - \tilde{\mu}_i\right)^2 \quad \forall i \tag{28}$$

$$\tilde{\gamma}_i = \frac{1}{|\Omega|} \sum_{s \in \Omega} \left(\xi_{i,s} - \tilde{\mu}_i\right)^3 \quad \forall i \tag{29}$$

$$\tilde{\eta}_i = \frac{1}{|\Omega|} \sum_{s \in \Omega} \left(\xi_{i,s} - \tilde{\mu}_i\right)^4 \quad \forall i \tag{30}$$

The normalization in (26) by powers of $\beta_{m,i}$ ensures that the objective is scale-invariant, and that each term has an equal weight.

A Data tables

DOT DOT

B Gams code

B.1 Bootstrapped rates

```
* Generate scenarios using bootstrap method
* Author: Tue Vissing Jensen and Tiago Soares
* DTU fall 2014 for course "Optimization in Finance."

$eolcom //

SETS
scenariotimes 'Index over weeks included in monthly scenarios' /t1*t4/
scenario 'Index of scenario' /s1*s250/;

SET ETF /
$include "..\data\etfs_max_mean.csv"
/;

set BaseDate /
$include "..\data\dates.csv"
/;

ALIAS (ETF, i);
ALIAS (BaseDate, t);
ALIAS (scenariotimes, st);
ALIAS (scenario, s);

// The new .csv file is read into the table prices
table prices(t,i)
$ondelim
$include ..\data\etfs_max_mean_prices.csv
$offdelim
;

PARAMETER HistoricalWeeklyReturn(i,t), HistoricalMonthlyReturn(i,t) ;
* Determining weekly return
HistoricalWeeklyReturn(i,t) = prices(t+1,i)/prices(t,i) - 1;
* Determining monthly return (used in the portfolio revision model)
HistoricalMonthlyReturn(i,t)$(ord(t) > 3) = (1+HistoricalWeeklyReturn(i,t))
    * (1+HistoricalWeeklyReturn(i,t-1)) * (1+HistoricalWeeklyReturn(i,t-2)) *
    (1+HistoricalWeeklyReturn(i,t-3)) - 1;

set tmonth(t) 'trading dates' ;
*Selecting the dates that will correspond to the number of months for the
    scenario set - 87
tmonth(t)$( (ord(t)>=161) and ( mod(ord(t)-1,4) eq 0 ) ) =1;

PARAMETER WeeklyScenarios(i,st,s,t)          'Week scenarios return considering
     interval 1 (28/01/2005 to 28/02/2008)'
          MonthlyScenarios(i,s,t)            'Month scenarios return
              considering interval 1'
          ScenarioReport(*,*,*)              'Summary of the generated
              scenarios'
          temp(t)                            'Number of months for generating
              scenarios - since 2005 until end of 2014'
          temp_2(t)                          'Number of months for exponential'
;

scalars    BeginNum        'Number of the first period'
           EndNum          'Number of the last period';

*Generating scenarios for period between 2005-1-28 and 2008-2-28
*BeginNum - first period after 2005-1-28;
BeginNum=1;
*EndNum - last period before 2008-2-28;
EndNum=161;

loop(tmonth,
loop(s,
     loop(st,
*random uniform function
temp(tmonth)=uniformint(BeginNum,EndNum);
```

```
*exponetial function for select more current data
temp_2(tmonth)=(1/2)*exp( (1/2)*(EndNum-BeginNum));

*Getting week scenarios
WeeklyScenarios(i,st,s,tmonth)=sum(t$(ord(t)=temp(tmonth)),
    HistoricalWeeklyReturn(i,t));
);
);
*Getting monthly scenarios
MonthlyScenarios(i,s,tmonth)= prod(st, (1+WeeklyScenarios(i,st,s,tmonth)))-1;
*selecting new period
BeginNum=BeginNum+4;
EndNum=EndNum+4;
);

display WeeklyScenarios, MonthlyScenarios, temp_2;

* Extracting MonthlyScenarios data to gdx file (to be used on CVaR model)
EXECUTE_UNLOAD 'Scenario_generation.gdx', MonthlyScenarios;
* Extracting historicalMonthlyreturn to gdx file (to be used on portfolio
    revision model)
Execute_unload 'Historical_month_return.gdx', HistoricalMonthlyReturn;
```