Allocation Methods for Alternative Risk Premia Strategies

Daniel Drugge
For
The First Swedish National Pension Fund

January 30, 2014

Abstract

We use regime switching and regression tree methods to evaluate performance in the risk premia strategies provided by Deutsche Bank and constructed from U.S. research data from the Fama French library. The regime switching method uses the Baum-Welch algorithm at its core and splits return data into a normal and a turbulent regime. Each regime is independently evaluated for risk and the estimates are then weighted together according to the expected value of the proceeding regime. The regression tree methods identify macro-economic states in which the risk premia perform well or poorly and use these results to allocate between risk premia strategies.

The regime switching method proves to be mostly unimpressive but has its results boosted by investing less into risky assets as the probability of an upcoming turbulent regime becomes larger. This proves to be highly effective for all time periods and for both data sources. The regression tree method proves the most effective when making the assumption that we know all macro-economic data the same month as it is valid for. Since this is an unrealistic assumption the best method seems to be to evaluate the performance of the risk premia strategy using macro-economic data from the previous quarter.

Sammanfattning

Vi använder en metod som delar upp avkastningsdata i en lågrisk-regim och en högrisk-regim, samt en metod som skapar ett binärträd vars grenar innehåller avkastningsdata givet olika makroekonomiska tillstånd, för att allokera mellan olika riskpremiestrategier. Vi bestämmer sannolikheten för att växla mellan regimer genom den så kallade Baum-Welch algoritmen och efter att ha delat upp avkastningsdata i olika regimer bestämmer vi den empiriska riskuppskattningen för lågrisk- och högrisk-regimen. Den slutgiltliga riskuppskattningen skapas genom att väga ihop de två riskuppskattningarna med sannolikheten för vardera regim. Binärträdet som används är ett så kallat regressionsträd och delar upp avkastningarna på ett sådant sätt att skillnaden i Sharpekvot maximeras mellan olika makroekonomiska tillstånd.

Regimväxlingsmetoden visar sig vara någon ineffektiv men resultaten blir bättre när mängden kapital investerat i riskpremiestrategierna minskas ju högre sannolikheten för att nästa period ska vara en högriskregim ökar. Regressionsträden fungerar väldigt väl i ett idealt scenario där vi vet relevant makroekonomisk data i förväg men ger även ganska bra resultat i realistiska scenarion.

Acknowledgements

I would like to thank Peter Emmevid at the First National Pension Fund for giving me the opportunity to write this report and for providing ongoing support and feedback. Furthermore I want to thank my supervisor at KTH, Filip Lindskog for invaluable support regarding the writing process. Lastly I thank Deutsche Bank for lending the data for me to use.

Contents

Chapter 1

Introduction

1.1 Purpose and Format of the Report

This study aims to create a method with which to allocate between the risk premia portfolios managed by Deutsche Bank. This allocation method is to be constructed in such a way that it generates stable growth, with focus on minimizing risk, and is uncomplicated. The connection between input and output of the method must be clear and easy to follow and explain.
The report is written in five main parts. The first part, Chapter 1, focuses on giving an overview of the risk premia strategies. Chapter 2 treats important theoretical concepts as well as goes into detail regarding the methods used to allocate between strategies and how to test the results. In Chapter 3 we take a look at the data available to us. Then in Chapter 4 the methods are applied to the data and results are presented. In the last part, Chapter 5, we discuss possible weaknesses in the methods and results and come to a conclusion regarding the allocation methods.

1.2 Risk Premia Strategies

Before reading this section, please note that a description of the macro-economic terms can be found in Appendix A.
Formally the risk premium is the amount by which the return of an asset exceeds the return of the risk free asset. The risk premia strategies are long short portfolios that are constructed in such a way as to be exposed to other risk factors than equity risk. These risk factors could be macro-economic factors such as GDP growth or inflation rate, or it could be all factors such as non-rational reactions to events. The treated risk premia strategies are

Value strategy in the bond market (VB) .
Value strategy in the foreign exchange market (VFX).
Value strategy in the equity market (VE).

Carry strategy in the bond market (*CB*).
Carry strategy in the foreign exchange market (*CFX*).
Carry strategy in the equity market (*CE*).
Momentum strategy in the bond market (*MB*).
Momentum strategy in the foreign exchange market (*MFX*).
Momentum strategy in the equity market (*ME*).
Size strategy in the equity market (*SE*).

Next follows a brief introduction to the methods used by Deutsche Bank to construct the portfolios as described in [1].

Value

In the case of equity the Value strategy is constructed by computing the

$$\text{Book-to-Price} = \frac{\text{Share price}}{\text{Book value}}$$

$$\text{Sales-to-Price} = \frac{\text{Share price}}{\text{Revenue per share}}$$

$$\text{12 months trailing earnings yield} = \frac{\text{Share price}}{\text{Trailing 12 months earnings per share}}$$

$$\text{12 months forward earnings yield} = \frac{\text{Share price}}{\text{Predicted 12 months earnings per share}}$$

Taking the mean of these measures the final ”value score” is computed. The Value portfolio is then constructed by going long the stocks with top 10 % value scores, and shorting the bottom 10 %.
When identifying the Value premium in foreign exchange Deutsche Bank constructs the portfolio by going long the 3 cheapest currencies and shorting the 3 most expensive in relation to the PPP of the G10 currencies.
Lastly, constructing the Value portfolio in fixed income Deutsche Bank computes the current account- and budget balance as a percentage of GDP and takes the mean of the result. Then the nominal bond yield is calculated net of the expected inflation rate. By taking the mean of both results the Value score is computed. Bonds included in the calculations are government bonds from the G10 countries and the Value strategy is going long the top 3 bonds and shorting the bottom 3 bonds.

Carry

The Carry portfolio in equity is constructed using the dividend yield by going long the 10 % with the largest dividend yield and shorting the bottom 10 %.
The Carry strategy in foreign exchange is going long the top 3 currencies of the G10 in terms of interest rate payment, going long the 3 top yielding and shorting the bottom 3.
The Carry strategy in bonds is to compute the mean of the 5-year duration bias on USD bonds, EUR bonds and JPY bonds. The USD and EUR duration bias

is found by investing in a swap with maturity 5 years paying one month LIBOR floating rate and receiving the swap rate on the USD and EUR currencies. The JPY duration bias is the same but with six month floating LIBOR rate.

Momentum

The Momentum premium in equity is found by computing the past one year cumulative returns, not including the recent 1 month, and ranking the stocks according to these scores. The Momentum premium in fixed income is formed by computing the cumulative return for the past 1 year skipping the recent 1 month and ranking the bonds according to these scores. Deutsche Bank goes long the top 3 and shorts the bottom 3 bonds. In currency Deutsche Bank ranks the G10 currencies according to their past 1 year change including the recent 1 month and goes long the top 3 performing currencies while shorting the bottom 3.

Size

The Size premium in the equity market is found by multiplying a company's stock price with the number of outstanding stocks, which is its market equity. Then the strategy is to go long the top decile and short the bottom. The Size premium is not one of the premium strategy that we have gotten from Deutsche Bank and will thus not be included in research with this data. Size premium strategy data is gotten from the Fama French library and included because the Fama French library does not have the Carry strategy.

1.2.1 Correlation Among the Risk Premia Strategies

A study into the correlation structure of the risk premia portfolios is done in [1] with the conclusion being that the correlation between portfolios is small and remains small during times of financial turbulence. Results of interest is that during periods of financial turbulence, defined by Deutsche Bank as "the 1997-1998 Asian Financial Crisis,1998 Russian Debt Default and LTCM fallout, the 2000-2001 Dot-Com Bubble Burst, 9/11 and the Credit Crisis following it, the 2008-2009 Global Financial Crisis and the more recent European Sovereign Debt Crisis", the average pairwise correlation between the risk premia strategies were 0.0 %, and 1.9 % for the rest of the time (from 1995 to 2012). The average pairwise correlation between the equity, bond and commodity markets during the same periods were 32.1 % in the turbulent period and 19.8 % for the rest. Leaning on this study correlation will be assumed to be very small and zero when constructing the methods in Chapter 2.

Chapter 2

Methods

2.1 Portfolio Optimization

The allocation methods will be built on modern portfolio theory, where we strive for a transparent and easy to understand allocation method. For this reason we form the risk parity portfolio, which is formed in the following way. Let $\mathbf{h}_t = (h_{1,t} \dots, h_{n,t})^T$ be a vector of positions in n risky assets at time t with price $\mathbf{S}_t = (S_{1,t}, \dots, S_{n,t})^T$. The value of the resulting portfolio is the sum of all positions times the value of the asset,

$$V_t = \sum_{i=1}^{n} h_{i,t} S_{i,t} = \mathbf{h}^T \mathbf{S}$$

We introduce the return vector at time t,

$$\mathbf{R}_t = \left(\frac{S_{1,t}}{S_{1,t-1}}, \dots, \frac{S_{n,t}}{S_{n,t-1}} \right)^T$$

and

$$\mathbf{w}_t = \left(h_{1,t} \frac{S_{1,t}}{S_{1,t-1}}, \dots, h_{n,t} \frac{S_{n,t}}{S_{n,t-1}} \right)^T$$

then we can express the value of the portfolio at time t as

$$V_t = \mathbf{w}_t^T \mathbf{R}_t$$

We wish to minimize the variance of this portfolio while at the same time maximizing the expected return, which leads us to solve the mean variance problem

$$\begin{aligned} maximize \quad & \mathbf{w}^T \boldsymbol{\mu} - \frac{c}{2V_0} \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \\ s.t \quad & \mathbf{w}^T \mathbf{1} \leq V_0 \end{aligned} \tag{2.1}$$

where $\boldsymbol{\mu} = E[\mathbf{R}]$, $\boldsymbol{\Sigma} = Cov(\mathbf{R})$ and c is a trade off parameter that depends on the willingness of the investor to take on risk . The solution to this problem is given by solving

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}}\left(\frac{c}{2V_0}\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} - \mathbf{w}^T\boldsymbol{\mu}\right) + \frac{\partial}{\partial \mathbf{w}}\lambda\mathbf{w}^T\mathbf{1} = 0 \\
\mathbf{w}^T\mathbf{1} \leq V_0 \\
\lambda \geq 0 \\
\lambda(\mathbf{w}^T\mathbf{1} - V_0) = 0
\end{aligned}$$

This system of equations evaluates to

$$\begin{aligned}
\mathbf{w} &= \frac{v_0}{c}\Sigma^{-1}(\boldsymbol{\mu} - \lambda\mathbf{1}) \\
\lambda &= \frac{(\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} - c)}{\mathbf{1}^T\boldsymbol{\Sigma}^{-1}\mathbf{1}}
\end{aligned} \tag{2.2}$$

with the constraint $\mathbf{w}^T\mathbf{1} = V_0$. We continue and try to find a solution that fits the requirement of a transparent and easy to follow model. In order to do this we will make good on the assumption in Section 1.2 that correlations among risk premia strategies are very small. In fact, we will assume that correlations are zero since it gives us a much easier expression of Equation (2.2). The covariance matrix becomes

$$Cov(\mathbf{R}) = \begin{pmatrix} \sigma_1^2 & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & \sigma_n^2 \end{pmatrix}$$

which causes Equation (2.2) to evaluate to

$$w_i = \frac{V_0}{c\sigma_i^2}\left(\mu_i - \frac{\sum_{j=1}^n \mu_j\sigma_j^{-2} - c}{\sum_{j=1}^n \sigma_j^{-2}}\right) \tag{2.3}$$

Going even further, we might consider that all asset returns have equal expected return, which means $\boldsymbol{\mu} = (\mu, \ldots, \mu)$. This assumption is made with the purpose of producing the easiest possible weight function, one that is currently used by the First Swedish National Pension Fund to weight among the risk premia strategies. Evaluating Equation (2.3) gives us the Inverse Volatility (IV)

$$w_i^{IV} = \frac{V_0}{c\sigma_i^2}\left(\frac{\mu\sum_{j=1}^n \sigma_j^{-2} - \mu\sum_{j=1}^n \sigma_j^{-2} + c}{\sum_{j=1}^n \sigma_j^{-2}}\right) = V_0\frac{1/\sigma_i}{\sum_{j=1}^n 1/\sigma_j} \tag{2.4}$$

that weights each asset according to contributed risk, so that all returns contributes with equal risk. Expected return is not the same for each strategy but the desire for the uncomplicated causes us to make the assumption.
Additionally, if one assumes that both the risk and expected return of all assets are the same, then one is forced to weight among the assets equally with respect

to capital invested. This assumption will serve as a reference, since it is not dependent on any previous data, and the allocation methods will be compared to it. Equal Weighting (Eq):

$$w_i^{Eq} = V_0 \frac{1}{n} \tag{2.5}$$

To illustrate the IV we consider three return series with joint probability distribution $\mathbf{r}_t \in \mathbf{N}(\mathbf{1}, \mathbf{\Sigma})$, where $\mathbf{N}(\mathbf{1}, \mathbf{\Sigma})$ is the three dimensional normal distribution with covariance matrix

$$Cov(\mathbf{R}) = \begin{pmatrix} 0.02^2 & 0 & 0 \\ 0 & 0.03^2 & 0 \\ 0 & 0 & 0.04^2 \end{pmatrix}$$

This means we want to take positions

$$\mathbf{w} = \frac{325}{3} \begin{pmatrix} 1/0.02 \\ 1/0.03 \\ 1/0.04 \end{pmatrix} \approx \begin{pmatrix} 0.46 \\ 0.31 \\ 0.23 \end{pmatrix} \tag{2.6}$$

where $\frac{325}{3}$ is the sum of the inverse volatilities.

2.2 Risk Measures

Since we are looking for a way to extract risk premium, it is only natural to use other types of risk measures than the simple standard deviation. Introducing the Value-at-Risk and Expected Shortfall.

Value-at-Risk

Literature [7] describes the Value-at-Risk at level p of a portfolio with value X at time 1 as "the smallest amount of money that if added to the position now and invested in the risk-free asset ensures that the probability of a strictly negative value at time 1 is not greater than p" [1],

$$VaR_p(X) = min\{m \ : \ P(mR_0 + X < 0) \leq p\}$$

where R_0 is the risk free return. If we let $L = -X/R_0$ be the loss at time 1 and $X = V_1 - V_0/R_0$ then

$$VaR_p(X) = min\{m : \ P(L \leq m) \geq 1 - p\}$$

where this means that the $VaR_p(X)$ is the smallest value m that covers a potential loss with probability $1-p$. Statistically the $VaR_p(X)$ is the $(1-p)$'th quantile of $\mathbf{L}$ where we let $\mathbf{L} = -\mathbf{R}^*$ where $\mathbf{R}^*$ is $\mathbf{R}$ sorted in ascending order, then the empirical Value-at-Risk is

$$\widehat{VaR}_p(X) = L_{[np]+1} \tag{2.7}$$

where $[\cdot]$ means the closest integer rounded down.
The $VaR_p(X)$ can be expanded into the Expected Shortfall as described next.

[1] [7] p.165.

V_t									
1.00	0.98	1.01	0.99	1.01	1.00	1.01	1.01	1.01	0.99
R_t									
0.00	-0.02	0.03	-0.02	0.02	-0.01	0.01	0.00	0.00	-0.01
L_t									
0.02	0.02	0.01	0.01	0.00	0.00	0.00	-0.01	-0.02	-0.03

Table 2.1: *Portfolio values V_t and portfolio returns R_t for 10 normally distributed random variables and corresponding sorted values L_t.*

Expected shortfall

The Expected Shortfall is the average $VaR_p(X)$ below level p. It is defined as

$$ES_p(X) = \frac{1}{p}\int_0^p VaR_u(X)du$$

for the continuous case. For the empirical, discrete, case we again use the definition of $\mathbf{L}$ as the possible sorted losses, and the empirical estimate becomes

$$\widehat{ES}_p(X) = \frac{1}{p}\int_0^p L_{[np]+1}du = \frac{1}{p}\left(\sum_{k=1}^{n}\frac{L_k}{n} + \left(p - \frac{[np]}{n}\right)L_{[np]+1}\right) \tag{2.8}$$

We continue with the return distributions from the illustration of IV and simulate 10 values of $\mathbf{R}_t$ representing monthly returns and take the position (2.6) at each month and get the resulting values in table 2.1. The Value-at-Risk we estimate at level 5% as $\widehat{VaR}_{0.05} = L_{[0.05*10]+1} = L_1 = 0.02$. In this case the expected shortfall is the same as the Value-at-Risk, since $[np] = 0$.

2.3 Performance Measures

Now we have the foundational method for allocating between the risk premia strategies, namely the Inverse Volatility. The allocation methods must have a way to be commonly judged to see what performs well and what does not. The desire to produce a stable method with focus on low risk leads us to treat the so called Information Ratio (IR), Sharpe Ratio (SR) and the Calmar Ratio (CR). These ratios are ways of evaluating the expected pay-off of an investment in relation to risk taken. We define them as follows:

Information Ratio (*IR*)

$$IR = \frac{CAGR}{Annual\ Volatility} \tag{2.9}$$

The term $CAGR$ stands for *Compounded Annual Growth Rate*, and it is defined as the yearly rate of return that would have produced the same total return as the investment strategy.

Compounded Annual Growth Rate (*CAGR*)

$$CAGR = \left(\frac{V_n}{V_0}\right)^{1/(t_n - t_0)} - 1 \tag{2.10}$$

where V_t is the value of the portfolio at time t and $t_n - t_0$ is the number of years from t_0 to t_n. In order to match the values of this estimate with the ones in [1] the value at time t is calculated as the product of all returns up to time t,

$$V_t = V_0 \cdot r_1 \cdot r_2 \cdots r_t$$

V_0 is the initial capital invested in the strategy. The $CAGR$ is reported throughout this report in percentages.
The annual volatility is defined as:

Annual Volatility (*Vol*)

$$Vol = \widehat{\sigma}\sqrt{12} \tag{2.11}$$

where $\widehat{\sigma}$ is the sample standard deviation of R calculated as the square root of the sample variance of the observations of R up to time t, $R_1, R_2, \ldots, R_n$, as

$$\widehat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_i - \widehat{\mu})^2}$$

where $\widehat{\mu} = \widehat{E[R]} = \frac{1}{n}\sum_{i=1}^{n} R_i$ is the sample mean. The volatility is reported throughout this report in percentages.
Next we introduce the Sharpe Ratio.

Sharpe Ratio (*SR*)

The Sharpe Ratio is defined as the estimated value of the excess return over the risk free rate. We use a zero risk free rate so the Sharpe Ratio is defined as

$$\frac{E[R]}{\sqrt{Var(R)}}$$

which extended to the yearly version becomes

$$\frac{E[R]^{12}}{\sqrt{Var(R) \cdot 12}}$$

Based on the sample $R_1, R_2, \ldots, R_t$ from R we estimate the Sharpe Ratio as

$$SR = \frac{\widehat{\mu}^{12} - 1}{\widehat{\sigma}\sqrt{12}} \tag{2.12}$$

The SR is used in order to get a sense of the expected return in comparison to the volatility, similar to the IR but less dependent on actual performance. Say there was a very big drop at time t, then the $CAGR$ would suffer greatly since the value of the portfolio would drop. So the value at the end of the period is lessened because of that drop. That is of course relevant to know but at the same time as a measure of performance we might be interested in the outcome of returns.
Lastly, the Calmar ratio is defined as

Calmar Ratio (*SR*)

$$CR = \frac{CAGR}{Maximum\ Drawdown} \tag{2.13}$$

which is a measure that relate the $CAGR$ to the so called maximum drawdown, explained below.

Maximum Drawdown (*MaxDD*)

The maximum drawdown measures the largest drawdown, or drop, in portfolio value between the start and end of simulation. Let $V(t)$ be the value of the portfolio at time t, then the maximum drawdown between time $t = 0$ and $t = T$ is defined as
Definition: Maximum Drawdown (MaxDD)

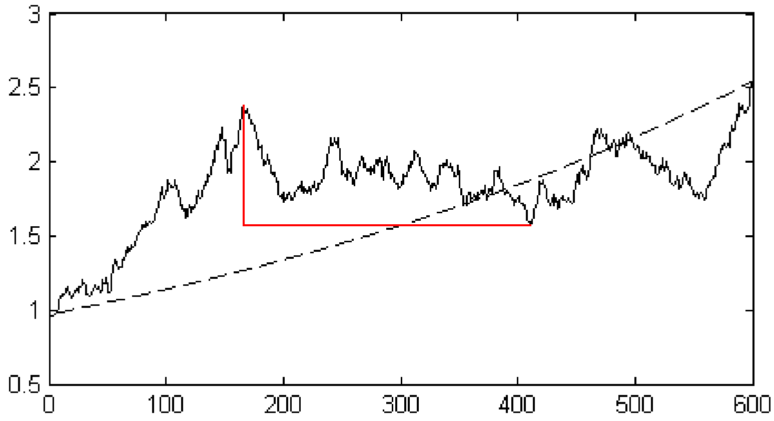
$$d(t) = \underset{\tau \in (0,T)}{max} \{ \underset{t \in (0,\tau)}{max} V(t) - V(\tau) \} \tag{2.14}$$

where

$$V(t) - V(\tau)$$

is a drawdown. This is obviously positive if $V(t) > V(\tau)$ and negative or zero else. To determine the maximum drawdown we simply traverse the whole time series and take the largest difference in portfolio value. The maximum drawdown itself is reported throughout this report in percentages, dividing the trough with the peak.

As an example, we generate 600 values from the normal distribution, $R_t = \mu + \sigma N(0,1),\ t = 1, \ldots, 600$ with $\mu = 1.001$ and $\sigma = 0.02$ to represent monthly returns. Figure 2.1 shows the simulated portfolio values $V_t = V_{t-1}R_t$. The filled black line corresponds to the time series V_t, the black spaced line shows the $CAGR = 1.95\%$ and the red line shows the maximum drawdown $MaxDD = 17.34\%$. The return series R_t has standard deviation 1.95% (as opposed to a theoretical 2%) which means the annual volatility is $Vol = 0.0195 \cdot \sqrt{12} = 0.0677 = 6.77\%$. The mean return is 1.0018 (as opposed to 1.0010) which means that the Sharpe Ratio is $SR = \frac{1.0018^{12}-1}{0.0677} = 0.31$. The Information Ratio in turn is $IR = \frac{0.0195}{0.0677} = 0.29$ and the Calmar Ratio is then 0.11.



(a) *Cumulated return series.*

CAGR	*Vol*	*IR*	*SR*	*MaxDD*	*CR*
1.95	6.77	0.29	0.31	17.34	0.11

(b) *Estimates.*

Figure 2.1: *Simulated portfolio performance.*

2.4 Regime Switching

It is clear that the financial market has its periods of greater instability and other calmer periods. In this section we try to identify these periods and even more importantly, try to estimate the probability that we are entering either a calm or a volatile period. We call these periods ”regimes” and define them as follows.

Definition: Regime
A regime is defined for a vector of returns, preferably an index, as a time period in which the volatility is within some predefined interval.

This interval can be chosen arbitrarily, but the authors of [1] suggests that we should define the *normal regime* as the period of calm, occurring when the volatility is less than the 40th quantile value for a volatility index. Setting the turbulent regime at the 40th quantile might seem a bit conservative but it is really up to the investor to set this threshold so in order to keep in line with previous studies in [1] I choose the same as Deutsche Bank.
The *turbulent regime* then, is the period of financial instability occurring the rest of the time. It is then important finding an index that models general market volatility as well as possible. One option is to use one of the large stock indexes available. We use the S & P 500 volatility index (S& P 500 VIX) to estimate regimes in this report. Sometimes however, instead of trying to find an

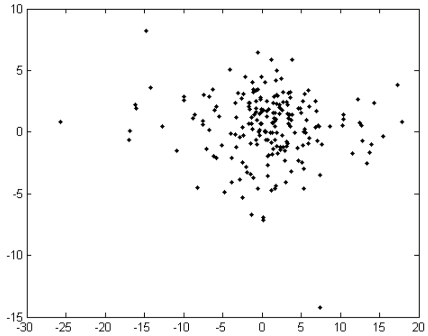


Figure 2.2: Scatter plot of return pairs from *ME* on the x-axis and *CFX* on the y-axis.

index, you could estimate the market volatility by looking at how the returns of the assets perform in relation to their typical behaviour [5] and create your own turbulence index. For example, in the case of one asset you can compute the volatility at each time point and then compare today's volatility to this data. This is the case when using the S & P 500 VIX and the regimes are separated according to quantiles of the distribution. But in the case of several assets simply computing individual volatility is not enough. For example, consider the *ME* and *CFX* returns. Figure 2.2 shows the return pairs for these strategies. Looking at the scatter plot we see that there are some clear outliers which could be caused by financial instability. But correlation is low in the tails so just because the Momentum strategy shows an unusual return doesn't mean that both strategies are in financial turbulence. In order to determine if they both perform unusual in relation to their combined behaviour we form the turbulence measure

$$d_t = (\mathbf{r}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{r}_t - \boldsymbol{\mu})$$

where $\mathbf{r}_t$ is the vector of returns at time t, $\boldsymbol{\mu}$ is the vector of mean returns for $\mathbf{r}$ based on historical data and $\boldsymbol{\Sigma}$ the covariance matrix of returns. With the assumption of zero correlation this simplifies to

$$d_t = \sum_{i=1}^{n} \left(\frac{r_{i,t} - \mu_i}{\sigma_i} \right)^2$$

This means that we measure the sum of deviations from the mean in relation to expected risk, akin to taking the square of the Sharp Ratio. This measure is called the Mahalanobis distance named after Prastanta Mahalanobis who developed it to analyse human skulls [5]. After having computed this we can set a threshold for the regimes just like before. [5] suggests using the 75th quantile as this threshold, but keeping in line with [1] we continue to use the 40th for consistency.

Now that we have managed to pick a turbulence index we would also like to be able to know in which regime we are and whether we will stay in the same or continue to another in the next period. It is suggested in [4] to use a Markov switching model to forecast the next regime. The thought is that, given that we are in any given regime, we would see characteristics that depend on the current regime. For example, if we use the S & P 500 VIX, and the volatility at time t is "high", then we might consider it an outcome of a distribution specific to the regime at time t. We do not know what the distributions are but we assume that there are two regimes 1 and 2 where the probability of observing y is $g_1(y)$ and $g_2(y)$. Then given an observation of y we evaluate both probabilities and we get an estimate of where we are. If X_t corresponds to regime 1 or 2 at time t, then X_t has the states $X_t = (1, 2)$ and X_1 corresponds to the regime at time 1. The probability of X_1 being regime i is $P(X_1 = i) = p_i$. The regime then possibly changes or stays the same but always shifts between these two, so X_t is a markov chain with transition probability matrix

$$\Upsilon = \begin{pmatrix} \upsilon_{11} & \upsilon_{12} \\ \upsilon_{21} & \upsilon_{22} \end{pmatrix}.$$

It is assumed that the probability of observing $y_t = y^*$ is dependent on the state of the Markov chain and we can form $\pi_i = P(y_t = y^* | X_t = i) = g_i(y^*)$. Let the vector $\theta = (g_1, g_2, p_1, p_2)$ be a parameter vector with g_1 and g_2 some probability functions that determine the probability of the observed outcome given regime 1 or 2 and p_1 and p_2 the initial probabilities of regime 1 and 2. Let θ be the optimal choice of parameters that maximizes the probability of observing the data $\mathbf{Y} = (y_1, y_2, \ldots, y_t)$. That is

$$\theta = \underset{\hat{\theta}}{max}\ P(\mathbf{Y}|\hat{\theta})$$

so that θ is the best choice of all possible combinations of probability functions and initial probabilities $\hat{\theta}$. Make an initial arbitrary guess for $\hat{\theta}$. Let the number of regimes be $n = 2$ and calculate the probability of observing the data at time t given that we are in the state $i = 1, 2$, $F_i(t) = P(\mathbf{Y}, X_t = i|\hat{\theta})$, called the forward probability, by the recursion

$$\begin{aligned} F_i(1) &= P(X_1 = i) \cdot P(y_1|X_1 = i) = p_i g_i(y_1) \\ F_i(t) &= P(y_t|X_t = i) \sum_{j=1}^{n} F_j(t-1) \cdot P(X_t = i|X_{t-1} = j) \\ &= f_i(y_t) \sum_{j=1}^{n} F_j(t-1)\hat{\upsilon}_{ji} \quad \forall t > 1 \end{aligned}$$

Then we calculate the probability of seeing the last $t - s, s < t$ observations given that we start in state i and end at time t, called the backward probability

$B_i(s) = P(y_{s+1}, y_{s+2}, \dots, y_t)$. It is recursively calculated as

$$\begin{aligned} B_i(t) &= \frac{1}{n} \\ B_i(s) &= \sum_{j=1}^{n} B_j(s+1) \cdot P(X_s = j | X_{s-1} = j) \cdot P(y_{s+1} | X_{s+1} = j) \\ &= \sum_{j=1}^{n} B_j(s+1) \hat{\upsilon}_{ij} g_j(y_{s+1}) \quad \forall s < t \end{aligned}$$

Now we calculate the probability that we are in regime i given the observations with the estimated parameters by

$$\gamma_i(t) = P(X_t = i | \mathbf{Y}, \hat{\upsilon}) = \frac{F_i(t) B_i(t)}{\sum_{j=1}^{n} F_j(t) B_j(t)}$$

We calculate the probability that we are in regime i at time t followed by regime j at time $t+1$ given the observations by

$$\beta_{ij}(t) = P(X_t = i, X_{t+1} = j | \mathbf{Y}, \hat{\theta}) = \frac{F_i(t) \hat{\upsilon}_{ij} B_j(t+1) g_j(y_{t+1})}{\sum_{k=1}^{n} \sum_{l=1}^{n} F_k(t) \hat{\upsilon}_{kl} B_l(t+1) g_l(y_{t+1})}$$

Then the parameters in $\hat{\theta}$ are updated as $\hat{p}_1 = \gamma_1(1)$, $\hat{p}_2 = \gamma_2(1)$,

$$\hat{\upsilon}_{ij} = \frac{\sum_{s=1} t - 1 \beta_{ij}(s)}{\sum_{s=1}^{t-1} \gamma_i(s)},$$

and

$$g_i(y) = \frac{\sum_{s=1}^{n} \gamma_i(s) I\{y = y_s\}}{\gamma_i}$$

where $I\{\cdot\}$ is the indicator function that is 1 if $y = y_s$.
The procedure is repeated until a maximum is reached. This procedure is called the Baum-Welch algorithm. The authors of the regime switching method in [4], suggests using a normal distribution to determine g_1 and g_2 with an initial guess for the mean and standard deviation. They let g_i be the relative probability based on the Gaussian probability density function

$$f_i(y_k) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(y_k - \mu_i)^2}{2\sigma_i^2}}$$

The density is calculated for every observation y_i and divided by the sum of the densities to get the relative probability of y_i

$$g_i(y_k) = \frac{f_i(y_k)}{\sum_{j=1}^{n} f_i(y_j)}$$

Then in the last step, when you have calculated the probabilities of observing $\mathbf{Y}$, you compute a new value for the expected value and standard deviation

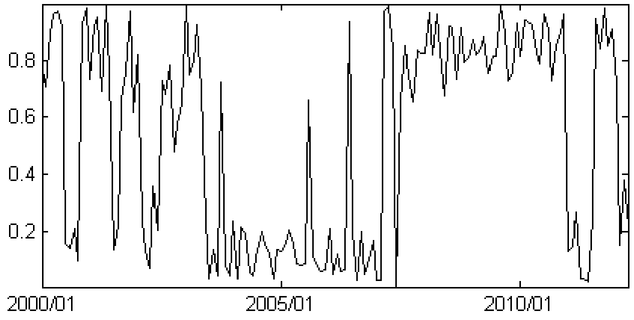


Figure 2.3: Out-of-sample forecast of a turbulent regime.

given that you are in regime i. Since $\gamma_i(t)$ are the probabilities corresponding to the observation, the expected value is

$$\mu_i = \sum_{s=1}^{t} y_s \gamma_i(s)$$

and the variance is

$$\sigma_i^2 = \sum_{s=1}^{t} \gamma_i(s)(y_s - \mu_i)^2$$

Setting a risk threshold at the 40th quantile we say that we are in the turbulent regime, regime 2, if the volatility at time t is higher than the 40th quantile of the turbulence index and normal, regime 1, otherwise. If we are in regime 1, then the probability of the same regime at time $t+1$ is υ_{11} and a regime shift happens with probability υ_{12}. So we can let p be the probability that next regime is turbulent by

$$p = \upsilon_{12} \cdot I\{Normal\} + \upsilon_{21} \cdot I\{Turbulent\} \tag{2.15}$$

Where $I\{\cdot\}$ is the indicator function that is 1 if the current regime is normal or turbulent.
An out of sample analysis using the S & P 500 VIX produces the estimated probabilities of a turbulent regime in Figure 2.3. We see that it captures history rather well. Most importantly it captures the financial crisis of 2008 very well. To further check this method we perform a simulation of how accurately the method identifies regimes. First we generate 900 values from the normal distribution with expected value 2 and standard deviation 1 and 1100 values with expected value 20 and standard deviation 10. The first state is the normal, or state zero, and the other is the turbulent, state 1. Figure 2.4 shows the different states and 2.5 shows the actual observations. Running the Baum-Welch algorithm gives us the probabilities in Figure 2.6. It is pretty clear from the

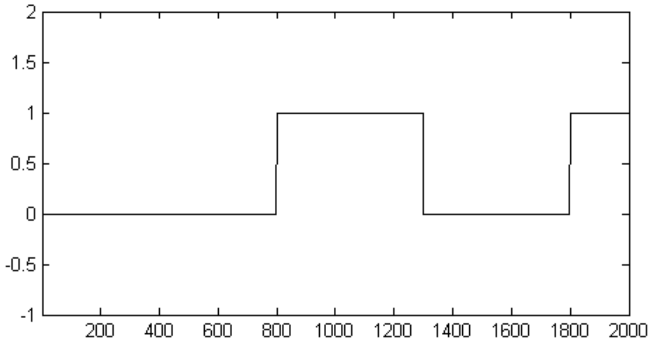


Figure 2.4: States of simulated data.

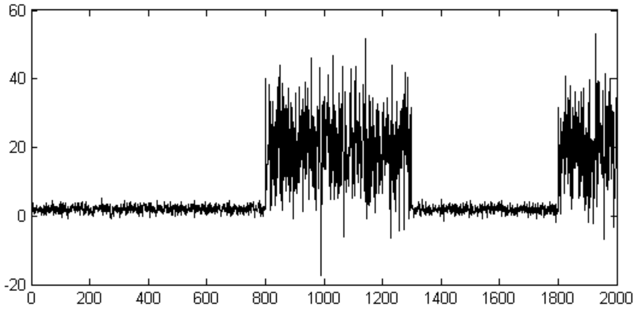


Figure 2.5: Observations of normal distribution with regimes.

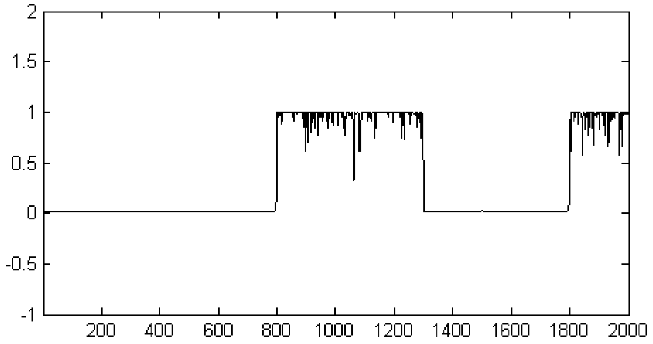


Figure 2.6: Probability of state 1.

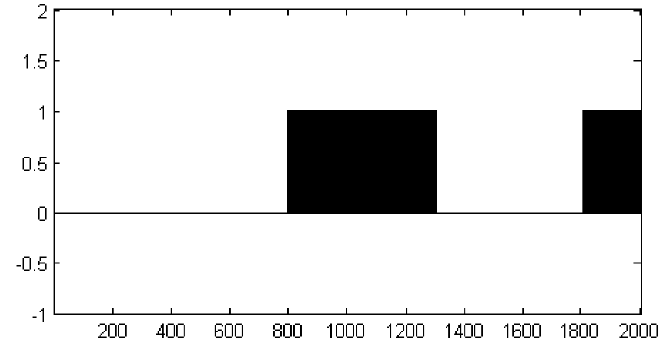


Figure 2.7: Estimated states.

figures that the probability of state 1 is captured well by the method, which is well given the very clear differences in the observations. The estimated value of state zero is 2.76 as opposed to 2 and for state 1 it is 22.07 as opposed to 20. The estimated standard deviation of state zero is 5 as opposed to 1 and for state 1 it is 20.6 as opposed to 10. So the standard deviation is captured rather well but the standard deviations are over estimated. However, we do not use the method to actually estimate risk, only to estimate which state we are in. The actual risk assessment is done on actual return data that we split manually as described previously. It is clear that the probabilities are captured well, but for the sake of analysis we first calculate the mean of the probability estimates, and then for each probability estimate we check if it is higher or lower than the mean. If it is higher, then we say that the predicted state is 1 and otherwise it is predicted to be state zero. This is not how we use the method in practice but it allows us to get an estimate of the accuracy of the method. The mean probability is 0.35, which means that any probability higher than 0.35 corresponds to state 1, and otherwise it is state zero. This is illustrated in Figure 2.7. Summing up all predicted states that are correct and dividing by 2000 gives us an estimate of the accuracy which is in this case 99.8 %. We might have just been lucky however, so we repeat the simulation with new random values 100 times. This gives us the estimated accuracy of 98.55 %. Table 2.2 shows the estimated accuracy for different values of normal distributions. μ_i, σ_i stands for expected value and standard deviation of state i. Clearly the more the distributions differ the more accurately the algorithm is able to predict the state. When state zero and state 1 is equal the method identifies state 1 correctly 53.74 % of the time, which is close to 55 % which is the odds of the state being 1 since we generated the states as such. So it does not seem to find a state that is not there. As for the precise values of μ and σ predicted by the method, they continue to be a bit off in these simulations. But we do not estimate the risk by these parameters so it serves its purpose which is to predict the state.

μ_0, σ_0	μ_1, σ_1	Accuracy
2,1	20,10	98.55
2,1	10,10	96.05
2,1	5,10	96.79
2,1	2,10	95.13
2,1	5,5	95.08
2,1	2.5,5	91.43
2,1	2.5,2.5	80.46
2,1	2,2	67.18
2,1	2,1	53.74

Table 2.2: Mean accuracy of predicted state 1 based on 100 simulations for different normal distributions.

2.4.1 Risk Parity Portfolio with Regime Shifting

We continue to expand the risk parity weights IV (2.4) by estimating risk according to the Baum-Welch estimated probabilities. Typically risk is estimated using equally weighted returns. But if we are in a very low risk period, then treating extreme outliers with as much caution as a return very close to the mean might overestimate the risk and we lose some return for the portfolio. The volatility might instead be calculated according to the data corresponding to each regime to hopefully get a more correct picture of the risk. So we form one estimate of the volatility that we say is a risk-off estimation, and one estimate that is risk-on in the following way. Consider that today is time t. Then we take h years of historical return data and assign each return vector to a turbulent regime according to if they occurred at the same time as the S & P 500 VIX (or another index) showed a volatility that exceeded the 40th quantile up to today or a normal regime if the volatility was less. Let $\mathbf{R}_t^{(on)} = (r_1^{(on)}, \dots, r_k^{(on)})$ and $\mathbf{R}^{(off)} = (r_1^{(off)}, \dots, r_m^{(off)})$ be the k and m number of returns after they are divided for the total t returns. After splitting the vector into the normal and turbulent regime we compute the risk-on estimation of the volatility by computing the sample variances $\widehat{\sigma}^{(on)}$ and $\widehat{\sigma}^{(off)}$.
Lastly we run the Baum-Welch algorithm to estimate the probability of regime i at time $t+1$ given if we are in regime 1 (volatility less than 40 % of historical values) or in regime 2 (volatility greater than 40 % of historical values. The resulting IV portfolio is computed for the risk-on and risk-off volatility.

$$w_i^{(IV-on)} = V_0 \frac{1/\widehat{\sigma}_i^{(on)}}{\sum_{j=1}^n 1/\widehat{\sigma}_j^{(on)}}$$

$$w_i^{(IV-off)} = V_0 \frac{1/\widehat{\sigma}_i^{(off)}}{\sum_{j=1}^n 1/\widehat{\sigma}_j^{(off)}}$$

Let p_t be the probability (2.15) at time t for next regime being turbulent, then the resulting portfolios are the expected value of the risk-on risk-off weights.

IV becomes the Inverse Volatility with Regime Shifting ($IV\ R/S$) and formally looks like

$$w_i^{(IV R/S)} = p_t \cdot w^{(IV-on)} + (1 - p_t) \cdot w^{(IV-off)} \tag{2.16}$$

2.5 Regression Trees

In this section we lay the foundation for exploiting macro-economic factors that might influence the performance of the risk premia strategies. Let

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ \vdots & \ddots & & \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{pmatrix}$$

be a matrix of m possibly dependent variables with n observations of each, where x_{ij} is the i'th observation of the j'th variable and

$$\mathbf{R} = (r_1, r_2, \dots, r_n)^T$$

be a vector of corresponding observations of risk premia returns. $\mathbf{X}$ corresponds to macro-economic data such as GDP or inflation or the like. We could then model $\mathbf{R}$ as a linear function of $\mathbf{X}$ by

$$\mathbf{R} = \beta_0 \mathbf{1} + \boldsymbol{\beta}^T \mathbf{X} + \boldsymbol{\epsilon}$$

where $\beta_0, \boldsymbol{\beta}$ are constant coefficients and $\boldsymbol{\epsilon}$ is the corresponding vector of residuals. This is obviously known as linear regression and it is necessary that $\mathbf{X}$ is independent across its columns in order to produce reliable predictions of $\mathbf{R}$. If $\mathbf{X}$ is not a set of independent variables, then we would have to know the dependence among them and exploit it to make $\mathbf{X}$ independent. Once that is done, we might also need to consider not regressing only on each independent value, but also among some or all interactions. If $\mathbf{X}$ is an $n \times 3$ matrix, then we would have to consider $\mathbf{R}$ in terms of 3 independent variables, plus 4 distinct interactions. It quickly becomes tedious and that is assuming we even manage to prepare the data in $\mathbf{X}$ correctly. The interaction between the macro variables is complicated and we can not expect to regress the return series on them using linear regression. Instead, we use the regression tree.

This tree is best explained by describing the algorithm of forming it. First, let $\mathbf{X}$ be the matrix as before, called the predictor matrix where each column represents one predictor variable and each row one observation. We let $g(\mathbf{X}, k) = (x_{1k}, x_{2k}, \dots, x_{nk})^T$ be a function that takes the k'th column from $\mathbf{X}$. For each k, we form

$$\mathbf{C}_k = (g(\mathbf{X}, k), \mathbf{R})$$

Then we split $\mathbf{C}_k$ into two disjoint sets $\mathbf{A}_k$ and $\mathbf{B}_k$ where $\mathbf{A}_k \bigcup \mathbf{B}_k = \mathbf{C}_k$ and $\mathbf{A}_k \bigcap \mathbf{B}_k = 0$. This split is done by letting the i'th observation of $g(\mathbf{X}, k)$, that is x_{ik}, be included only in $\mathbf{A}_k$ if it is smaller than some value v_k with the corresponding risk premia return r_i joining it, thereby keeping them paired, otherwise they are included only in $\mathbf{B}_k$. The value v_k is chosen such that the difference Δ_k in Sharpe Ratio calculated for each set $\mathbf{A}_k$ and $\mathbf{B}_k$ is maximized. We then take the return vector from $\mathbf{A}_k$ and $\mathbf{B}_k$ by $g(\mathbf{A}_k, 2)$ and $g(\mathbf{B}_k, 2)$ and let S be a function $S = S(\mathbf{R}) = \widehat{\mu}/\widehat{\sigma}$ then

$$\Delta_k = \underset{v_k}{max} \left| S(g(\mathbf{A}_k, 2)) - S(g(\mathbf{B}_k, 2)) \right|$$

The difference in Sharp Ratio is weighted according to how many data points were used to calculate it according to the weight function $n_1 n_2$ where n_1 and n_2 are the amount of data points smaller than v_k and greater than v_k respectively. The resulting difference value is $\Delta_k \cdot n_1 n_2$. This is repeated for each column $k = 1, \ldots, m$ in $\mathbf{X}$. This generates m estimations of Δ and we finally pick the predictor (column) k_{max} that produced the maximum value of Δ and we split the entire matrix $\mathbf{X}$ into $\mathbf{X}^{(A)}$ and $\mathbf{X}^{(B)}$ where row i of $\mathbf{X}$ is included only in $\mathbf{X}^{(A)}$ if $x_{ik_{max}}$ is in $\mathbf{A}_{k_{max}}$, and in $\mathbf{X}^{(B)}$ otherwise. The same is done for $\mathbf{R}$ which is split into $\mathbf{R}^{(A)} = g(\mathbf{A}_{k_{max}}, 2)$ and $\mathbf{R}^{(B)} = g(\mathbf{B}_{k_{max}}, 2)$. Now we form a binary tree with node 1 containing $\mathbf{R}$. Node 1 points to node 2 and node 3 and we let node 2 contain $\mathbf{R}^{(A)}$ and node 3 contain $\mathbf{R}^{(B)}$. Then we repeat the split procedure for node 2 with $\mathbf{R}^{(A)}$ being split on the best predictor k_{max} generated from $\mathbf{X}^{(A)}$ and we do the same for node 3 with $\mathbf{R}^{(B)}$. This is repeated for the new values of $\mathbf{A}$ and $\mathbf{B}$ for each new node created. This builds a binary tree which has fewer and fewer entries in each node, until no more split can be made. A split can not be made if either $S(\mathbf{R})$ can not be calculated or if $\Delta = 0$ for all $k = 1, \ldots, m$. This can happen when the sample variance of the data in the node is zero or there are too few entries in a node. In order to be able to calculate $S(\mathbf{R})$ there has to be at least some nodes in every tree minimum. Before determining what this minimum should be we evaluate the method in an example.
We generate

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$$

where $\mathbf{x}_i = (x_{1i}, \ldots, x_{ni})^T$ for $i = 1, 2, 3$ and $n = 1000$. The x_{ij}'s are generated according to continuous uniform distributions where $\mathbf{x}_1 \sim U(-\frac{1}{100}, \frac{2}{100})$, $\mathbf{x}_2 \sim U(-\frac{0.5}{100}, \frac{0.5}{100})$ and $\mathbf{x}_3 \sim U(-\frac{1.5}{100}, \frac{3}{100})$. Then 1000 corresponding values are generated for $\mathbf{R}$ from the normal distribution with parameters that depend on the values of $\mathbf{X}$. These are chosen in the following way.
Let $\mathbf{t} = (t_1, t_2, t_3) = \frac{1}{100}(0.5, 0, 0.75)$ be a vector of threshold values and let *bool* $= (b_1, b_2, b_3)$ be a vector of boolean values where b_j can take the values either 0 or 1. Then let $b_j = 0$ mean that $x_{ij} < t_j$ and $b_j = 1$ mean that $x_{ij} \geq t_j$. Then each r_i is simulated from the normal distribution $N(\mu, \sigma)$ where μ and σ

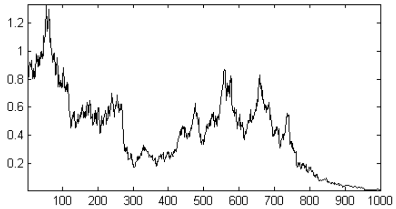


Figure 2.8: Simulated time series of returns $\mathbf{R}$.

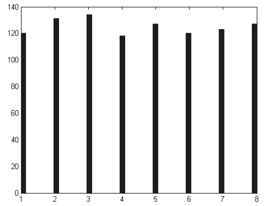


Figure 2.9: Distribution of entered states of *bool*.

are chosen according to

$$\begin{aligned}
if\ bool = (0,0,0)\ then\ \mu &= -0.04, \sigma = 0.08 \\
else\ if\ bool = (0,0,1)\ then\ \mu &= -0.03, \sigma = 0.07 \\
else\ if\ bool = (0,1,0)\ then\ \mu &= -0.02, \sigma = 0.06 \\
else\ if\ bool = (1,0,0)\ then\ \mu &= -0.01, \sigma = 0.05 \\
else\ if\ bool = (0,1,1)\ then\ \mu &= \quad 0.01, \sigma = 0.04 \\
else\ if\ bool = (1,1,0)\ then\ \mu &= \quad 0.02, \sigma = 0.03 \\
else\ if\ bool = (1,0,1)\ then\ \mu &= \quad 0.03, \sigma = 0.02 \\
else\ if\ bool = (1,1,1)\ then\ \mu &= \quad 0.04, \sigma = 0.01
\end{aligned}$$

The resulting time series $\mathbf{R}$ is illustrated in Figure 2.8 where $V_t = r_t V_{t-1}$ and $V_0 = 1$. The distribution of the different states *bool* are shown in Figure 2.9. We set the mininum amount of entries per node as 70 (arbitrarily) and generate the tree in Figure 2.10. We see that the tree splits $\mathbf{X}_1$ on 0.49% $\approx$ 0.5% , $\mathbf{X}_2$ on 0.09% and 0%, and $\mathbf{X}_3$ on 0.76%, 0.98% and 1.16%. The error is easy to estimate since we know the theoretical Sharpe Ratio for each node by $SR = \frac{\mu}{\sigma}$, so we predict the value for each possible state of *bool* and compare it to the theoretical value. This gives the mean squared error 0.07. Using the same predictor matrix $\mathbf{X}$ we generate 1000 new versions of $\mathbf{R}$, grow a new tree and evaluate the error on each. The resulting distribution of errors has a mean of 0.03. Noteworthy is that, while we know the real distributions of the data and thus the optimal split, the algorithm only sees one outcome and thus might not split it theoretically the best way. The rather small error is however satisfac-

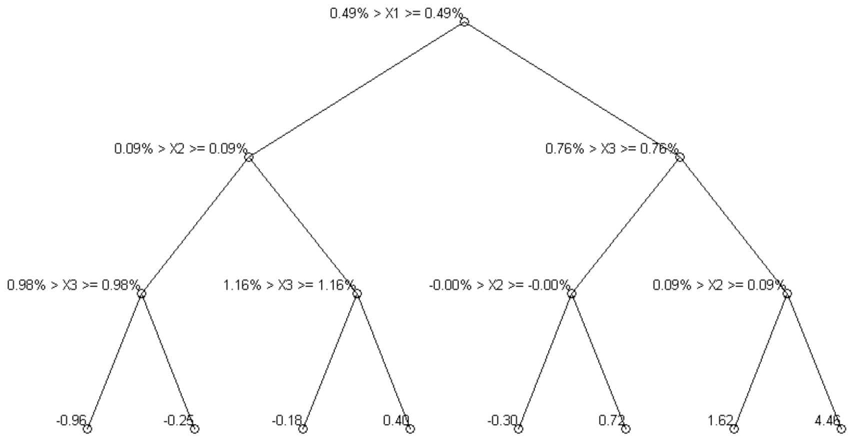


Figure 2.10: Tree grown on simulated data.

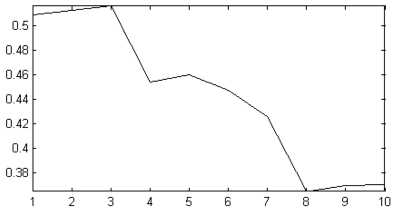


Figure 2.11: Rank correlation as function of minimum entries per node.

tory, at least for our purpose as we are not trying to predict an exact result, but rather get an "edge" in our investment methods, so it is enough if it predicts somewhat correctly.

2.5.1 Cross Validation

A problem with growing the tree like this, is that at some point we split the data too far and essentially estimate the SR on noise. This would lead to the tree predicting the data used to generate the tree really well, but it gives too much weight to factors that really are only noise. One way around this issue is by using cross validation to determine how much the data should be split. In this procedure we grow a tree only on a subset of the data and then we predict the outcome of the rest of the data and see how close we came. If we have grown a too large a tree (in terms of minimum amount of data per node; a large tree has a small minimum of data per node), and thus have over fitted it, we are trying to predict "too much" of the oscillations, noise. Then we would expect the predictions to be incorrect and we should grow a smaller tree.

Specifically we grow a tree on 9/10's of the data and predict the last 1/10. Then we grow a tree on a different subset 9/10's of data and predict the remaining. We repeat this ten times and get an estimate of the error of the tree. However, this is what we would do if we could indeed predict the remaining data. Since we are splitting the tree such that the difference in SR is maximized, we would be trying to predict the SR. But when trying the tree on one tenths of data, we will not always have enough data to estimate the SR for every node. In the case of the data provided by Deutsche Bank, one tenth of data is only 21 monthly returns. If our tree has k nodes, that leaves each node with an average $21/k$ returns per node. This quickly becomes much too little for any significant estimate to be done. Instead we work around this using rank correlation (Appendix B) to estimate the best tree. Now, we grow a tree on 9/10's of the data as before and predict the result of the remaining data. But instead of checking that the predictions are accurate by computing the SR on the remaining data, we check the rank correlation between the predicted SR values and the actual remaining data. The remaining data is actual returns, so we are not comparing the same things, but we are evaluating whether highest predicted SR corresponds to the largest positive return and the second highest SR corresponds to the second highest positive return and so on. We perform the rank correlation test on each 9/10 subset and our final rank correlation value is the mean of ten subset estimations. This will hopefully be a sufficient estimation.

Performing this cross validation technique on our sampled data and plotting the rank correlation values against minimum amount of data per node gives us Figure 2.11 with corresponding best tree in Figure 2.12. The tree is essentially the same as before but has one less node, namely it no longer splits $\mathbf{X}_3$ on 1.16%, which is well, since it means this tree is less over fitting. The error of this tree is slightly higher with 0.08 instead of 0.07, but we still know it splits the data more correctly than before, since $\mathbf{X}_3$ should not be split on 1.16%.

In conclusion, this tree splitting procedure works rather well for our sampled data and thus it seems viable to use it on the real data.

2.5.2 Bagging

A way around the problem of cross validating the trees is the bagging procedure. Bagging stands for bootstrap aggregating and does not require cross validation to avoid over fitting [2]. Here we generate m_b new samples from the data and generate a tree on each of the samples. Additionally, instead of generating a tree on all predictors, we select $[\sqrt{N}]$ of them, where N is the total number of predictors and $[\cdot]$ means the closest integer rounded down, at random and generate the tree only on them. We grow each tree as much as we can, which lets us avoid the problem of finding the right amount of minimum data points per node. After generating a forest of m_b trees the prediction is made as

$$\hat{y}_t = \frac{1}{m_b} \sum_{k=1}^{m_b} \phi_k(\mathbf{X_t}) \qquad (2.17)$$

[2] [9] p.319

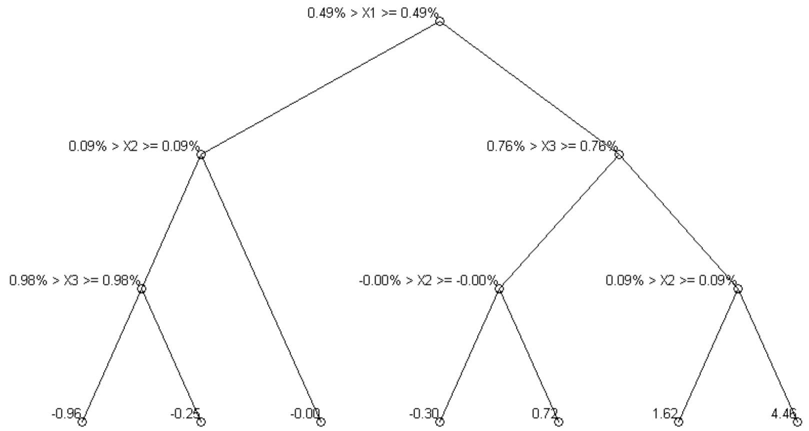


Figure 2.12: Cross validated tree.

where $\phi_k(\mathbf{X_t})$ is the prediction of the k'th tree based on the predictor values at time t. This means that the predicted value is the average of m_b predictions.

2.6 Statistical Tests

Some way of testing the results of a strategy is needed, some way of differencing significant results from pure chance. For this reason the following procedures is introduced: the Monte Carlo Bootstrap. Since we are dealing with very complex time series, which although assumed uncorrelated in the long run might exhibit more or less correlation in the short run we cannot simply hope to generate new time series to test on. This is always a weakness in financial statistics, that the results we get might be rather ad hoc in that they are created for one specific time series but might not work for others. One possibility of getting past this issue would be to simply generate new time series with the correct distributions. But as suggested, we are unlikely to actually be able to model the time series correctly. Furthermore, if we were to model the time series from an existing subset of outcomes (that we have observed so far, 208 returns per risk premia strategy in this case) we still could not be sure that we have captured the future distribution correctly. Besides, if the assumptions of this report are correct, that the returns to the risk premia portfolio are dependent on underlying market volatility and various macro-economic factors, I do not expect to be able to generate completely new time series from theoretical distributions and retain vital characteristics. Instead we might use the bootstrap technique to estimate certain parameters.

The bootstrap

Let $x_1, x_2, \dots, x_n$ be a series of identically distributed random outcomes from $X(t)$ with an unknown distribution F. As is the premise, we do not know F but we hope to disregard that since we are interested in some estimate $\hat{\theta}$ of a parameter $\theta = T(F)$. This parameter corresponds to any of the measures in Section 2.3. If we knew F we could generate new time series and compute $\hat{\theta}$ but as previously argued this is not possible. However, based on $x_1, x_2, \dots, x_n$ we can make however many copies of existing data points as we want, in whatever order we want and including how many or as few of the data points as we want. This is done by taking with replacement from $x_1, x_2, \dots, x_n$ so that any observation is picked with probability $\frac{1}{n}$ and save this in a new observation. Indeed, we pick n values from our original n x's to get an entirely new sample $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ where $x_i = x_i^{(1)}$ with probability $\frac{1}{n}$. Repeating this B times we eventually get the matrix of samples

$$\mathbf{X}_B = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots x_n^{(2)} \\ \vdots & \vdots & \vdots \\ x_1^{(B)} & x_2^{(B)} & \dots x_n^{(B)} \end{pmatrix}$$

Each row in this matrix is a copy of the original data, and thus we can calculate $\hat{\theta}^{(j)} = T(F_j)$, where F_j is the empirical cumulative distribution function for row j, for each of these rows to get

$$\hat{\theta}_B = \begin{pmatrix} \hat{\theta}^{(1)} \\ \hat{\theta}^{(2)} \\ \vdots \\ \hat{\theta}^{(B)} \end{pmatrix}$$

Once we have this sample, we have actually gotten a distribution of the parameter estimate given the original data. Then we can for example estimate $\hat{\theta} = T(F)$ as the sample mean $E[\hat{\theta}]$. Of course, there are more things to think about when it comes to the bootstrap procedure but it is not really important right now. In fact, returning to the mindset of the risk premia return, simply taking one return at a time to create a new sample might misrepresent the time series. Assuming again that the returns exhibit some dependence on the underlying volatility we want to be able to get an estimate of the return at time $t+1$ when we are at time t by using the Baum-Welch method as previously described. Splitting up the volatility might then render it impossible to predict the next outcome since only the original arrangement might render good results for our method. Again, we could maybe model a new volatility index, but corresponding returns are not possible to model accurately, at least for me. So this leaves the bootstrap method rather impotent. However, relying on the theory we expand it into the Monte Carlo Bootstrap.

Monte Carlo Bootstrap

Here we actually keep the time series intact which is following the previous argumentation a very good thing. Indeed, we run a simulation from start to finish and at each time point t we save the weight vector $\mathbf{w}_t$ that determines how much capital is allocated to what strategy. Let $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$ be the vector of saved weights, where each $\mathbf{w}_i$ is an $N \times 1$ matrix where N is the number of risk premia strategies to take a position in. Then instead of generating a new matrix of resampled return vectors, which as previously claimed might render them more or less useless to use for our purpose, we construct the matrix

$$\mathbf{W}_P = \begin{pmatrix} \mathbf{w}_1^{(1)} & \mathbf{w}_2^{(1)} \ldots & \mathbf{w}_n^{(1)} \\ \mathbf{w}_1^{(2)} & \mathbf{w}_2^{(2)} \ldots & \mathbf{w}_n^{(2)} \\ \vdots & \vdots & \vdots \\ \mathbf{w}_1^{(n!)} & \mathbf{w}_2^{(n!)} \ldots & \mathbf{w}_n^{(n!)} \end{pmatrix}$$

where $\mathbf{w}_i^{(k)} \neq \mathbf{w}_j^{(k)}$,where k is the k'th row of $\mathbf{W}_P$ and no row is equal to any other row. This means that $\mathbf{W}_P$ is the matrix of all $n!$ possible permutations of $\mathbf{W}$. Then, almost following the idea of the bootstrap method we sample a new matrix with random draws from $\mathbf{W}_P$ but without replacement to get

$$\mathbf{W}_B = \begin{pmatrix} \mathbf{w}_1^{(1^*)} & \mathbf{w}_2^{(1^*)} \ldots & \mathbf{w}_n^{(1^*)} \\ \mathbf{w}_1^{(2^*)} & \mathbf{w}_2^{(2^*)} \ldots & \mathbf{w}_n^{(2^*)} \\ \vdots & \vdots & \vdots \\ \mathbf{w}_1^{(B^*)} & \mathbf{w}_2^{(B^*)} \ldots & \mathbf{w}_n^{(B^*)} \end{pmatrix}$$

where j^* means that it's the j'th row in $\mathbf{W}_B$, where $j^* = j$ with probability $\frac{1}{n!}$. We then compute the parameter estimate

$$\theta^{(j^*)} = f(\mathbf{w}^{(j^*)}, \mathbf{R})$$

for every row in $\mathbf{W}_B$ where $f(\cdot)$ might correspond the the Sharpe or Information Ratio or maximum drawdown for example and $\mathbf{R}$ is the return vector. Furthermore, neither of the $\hat{\theta}^{(j^*)}$'s have a higher likelihood of occurring than any other, so in fact each have a $\frac{1}{B}$ of occurring. Taking the mean of the sample of $\theta^{(j^*)}$'s is rather uninteresting as the distribution of $\theta^{(j^*)}$ does not represent the performance of our strategy in of itself. However, let $G(\hat{\theta}) = \frac{1}{B}\sum_{k=1}^{B} I\{\hat{\theta}^{(k)} > \hat{\theta}\}$ where $I\{\cdot\}$ is the indicator function. $G(\hat{\theta})$ can take the values $\frac{1}{B}, \frac{2}{B}, \ldots, 1$. Since either permutation have the same probability of occurring then $G(\hat{\theta})$ follows the discrete uniform distribution. Letting $B \to \infty$ lets $G(\hat{\theta})$ converge to the continuous uniform distribution. This means that $P(G(\hat{\theta}) \leq p) = p$. So with a large enough value of B we can estimate the probability that the result gotten from our strategy was purely by chance, as other random permutations would have performed equally well or better. Of course, doing this test only lets us know if our strategy actually used the historical data in a meaningful way and not actually the value that is expected from the strategy given other time series.

2.7 Summary

In Chapter 2 we formed the Inverse Volatility IV allocation method that allocates with equal risk contribution. The Baum-Welch algorithm was introduced to estimate a probability of an upcoming turbulent regime and this probability was used to expand IV into the Inverse Volatility with regime shifting method, $IV\ R/S$. This method allocates with equal risk contribution, but one allocation for a normal, calm, regime and one for a turbulent one and then weighted together according to the estimated probability reported by the Baum-Welch algorithm. We tested the Baum-Welch method on historical data and got very satisfying results.

Then we introduces the regression trees as a way of splitting up the return series based on an input of macro-economic data to see under what circumstances we get what Sharp Ratio. We tested the method on simulated data and saw that it captured the characteristics rather well. Then the idea of cross validation using rank correlation and the idea of bagging was introduced in order to improve the tree.

Lastly we introduced a way of checking the results with the Monte Carlo bootstrap to get a sense of the statistical significance.

Chapter 3

Data

3.1 Introduction

The main goal is to evaluate the performance on the return data from Deutsche Bank, but as support we have data from Fama French which stretches back much further. This data is collected at Dartmouth College in the United States and US Research Data can be found at http://mba.tuck.dartmouth.edu/pages/faculty /ken.french/data_library.html. We use data from September 1960 to December 2012. In order to get a feel for the data we compute the empirical mean, standard deviation, skewness and kurtosis of the data. These are computed as follows, where $\mathbf{R} = (r_1, r_2, \ldots, r_n)^T$ is the return vector with n monthly returns.

Mean ($\widehat{\mu}$)

$$\widehat{\mu} = \widehat{E[R]} = \frac{1}{n}\sum_{k=1}^{n} r_k \tag{3.1}$$

Standard Deviation $\widehat{\sigma}$

$$\widehat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(r_k - \mu)^2} \tag{3.2}$$

Let m_j stand for the empirical *j'th moment around the mean* defined for the return vector $\mathbf{R}$ as

$$m_j = \frac{1}{n}\sum_{k=1}^{n}(r_k - \mu)^j \tag{3.3}$$

then we can define the skewness and kurtosis in terms of m_j.

Skewness (S)
The skewness is a measure of how much the distribution of R leans to one side of the mean. A positive value means that it leans to the right of the mean,

	$\widehat{\mu}$ %	$\widehat{\sigma}$ %	S	K
VB	1.33	3.01	0.55	4.34
VFX	4.16	8.69	0.1	3.16
VE	8.86	15.17	0.92	7.9
CB	3.29	2.81	0.03	3.56
CFX	5.03	9.35	-1.02	6.77
CE	12.42	12.82	0.34	4.13
MB	1.33	4.23	0.65	6.69
MFX	1.94	9.28	0.34	6.60
ME	9.38	21.20	-0.52	5.43

Table 3.1: *Distribution of return data for risk premia strategies.*

and the opposite for a negative value. We use the sample skewness which is computed as

$$S = \frac{m_3}{m_2^{2/3}} \tag{3.4}$$

Kurtosis (K)
The kurtosis measures how heavy tailed a distribution is. Its sample estimate is

$$K = \frac{m_4}{m_2} \tag{3.5}$$

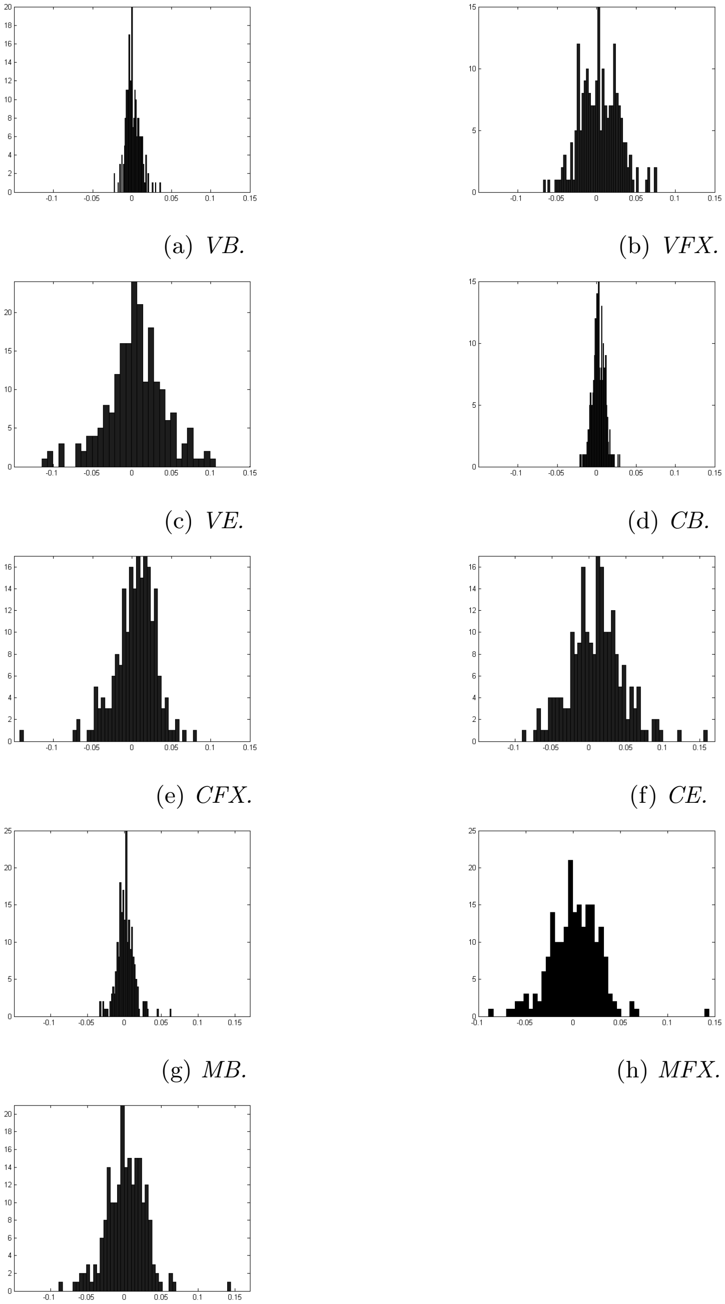
For reference, the standard normal distribution has $S = 0$ and $K = 3$.

3.2 Deutsche Bank Data

3.2.1 Introduction

Deutsche Bank has kept track of their monthly returns since 1990, and so we have access to 17 years of monthly returns. In total there are 208 monthly returns per strategy. Figure 3.1 and Table 3.1 shows the empirical distributions of the return data. The mean and standard deviation are transformed to yearly values by computing $\widehat{\mu}^{12} - 1$ and $\widehat{\sigma}\sqrt{12}$.

Bonds are the most stable investment in terms of risk while the largest risks are in equity investments. All distributions but for the *CFX* and *ME* were skewed to the positive side of their respective means and all had a kurtosis greater than 3, meaning they were heavier tailed than the normal distribution. The equity investments had the highest return on average while bonds had the lowest. Applying the measures defined in Section 2.3 to the historical data provided by Deutsche Bank gives us the values in Table 3.2. The Momentum strategy gives an average *IR* value of 0.26 and *SR* of 0.33, the Carry strategy has the average *IR* value of 0.83 and *SR* of 0.89 while the value strategy has the *IR* value 0.46 and *SR* value 0.51. The differences between the values indicates that they were subjected to a large outlying negative return. The largest difference



(a) *VB.*

(b) *VFX.*

(c) *VE.*

(d) *CB.*

(e) *CFX.*

(f) *CE.*

(g) *MB.*

(h) *MFX.*

(i) *ME.*

Figure 3.1: *Distribution of returns for individual strategies.*

	CAGR	*Vol*	*IR*	*SR*	*CR*
VB	1.31	3.01	0.44	0.45	0.18
VFX	3.80	8.70	0.44	0.48	0.24
VE	7.66	15.17	0.50	0.59	0.17
CB	3.23	2.82	1.15	1.16	0.63
CFX	4.56	9.37	0.46	0.54	0.13
CE	11.47	12.82	0.89	0.97	0.32
MB	1.28	4.22	0.30	0.33	0.17
MFX	1.52	9.29	0.16	0.21	0.06
ME	6.91	21.20	0.33	0.44	0.14

Table 3.2: *Performance of the risk premia strategies for the years 1995-2012.*

is found in *ME* and *CFX* and looking at the empirical distribution in Section 3 we see that both the distributions have heavier left tails than the others and they are the only ones with negative skew, so the results were not unexpected.

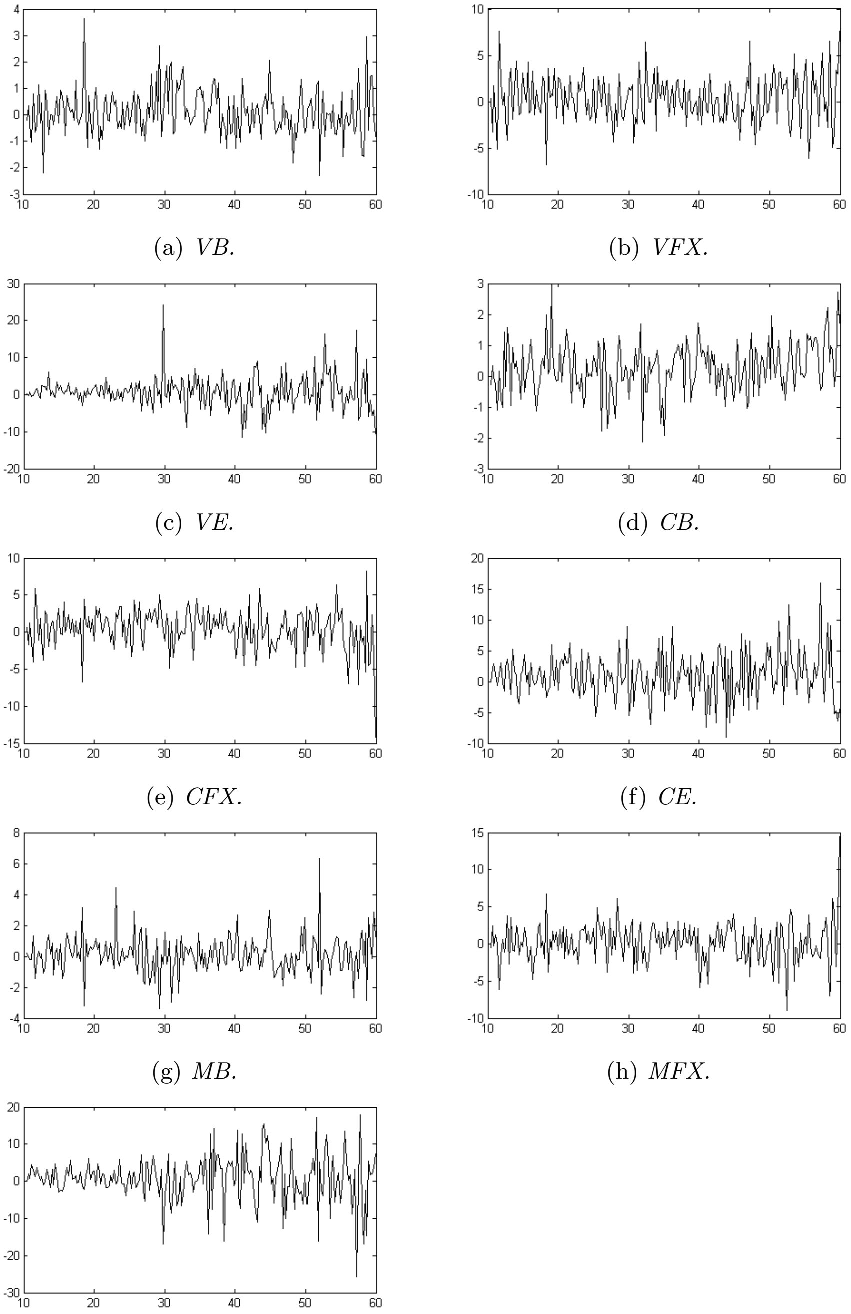
3.2.2 Historical Regimes

We are interested in seeing see how the risk premia strategies perform depending on the assumed regimes. Figure 3.2 shows the returns as function of increasing volatility of the S & P 500 VIX. These figures are formed by sorting the volatility index in ascending order and plotting the corresponding returns.

The normal regime corresponds to the data in Figure 3.2 with the x-axis values 1 to 83, and the turbulent regime to the x-axis values 84 to 208. We see clearly that the volatility of the equity strategies shows dependency on the volatility of the S & P 500. The other asset classes do not exhibit this. One is lead to think that regime shifting will be effective in equity, but may not be as effective for the other asset classes. Closer inspection into the particulars of the different regimes, into the expected return, volatility, skew and kurtosis are found in Table 3.3a and 3.3b. In order to determine whether the expected value is any different across regimes we perform the t-test for different means with unequal variance and different sample size, the Welch's t-test,

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \qquad df = \frac{(s_1^2/n_1 + s_2^2/n_2)^2}{(s_1^2/n_1)^2/(n_1-1) + (s_2^2/n_2)^2/(n_2-1)} \tag{3.6}$$

Where *df* is the degrees of freedom for the standard student's t-distribution. The p-value of the null hypothesis that the difference in the means and residual come from a normal distribution and that the difference is zero, $P(H_0)$, is included in the parenthesis in Table 3.3b. Only the *CFX* shows a statistically significant change in expected value in the turbulent regime at level 96 %. *CB* also shows some significant change, at level 92 %. All other strategies have pretty much



(a) *VB.*

(b) *VFX.*

(c) *VE.*

(d) *CB.*

(e) *CFX.*

(f) *CE.*

(g) *MB.*

(h) *MFX.*

(i) *ME.*

Figure 3.2: *Returns time series as function of underlying volatility.*

	$\widehat{\mu}$ %	$\widehat{\sigma}$%	S	K
VB	1.33	2.91	0.97	6.54
VFX	3.66	8.18	-0.08	3.69
VE	12.01	11.19	4.41	33.25
CB	1.81	2.84	0.29	3.83
CFX	11.22	7.55	-0.58	3.85
CE	11.48	9.49	0.1253	3.22
MB	1.69	4.12	19.23	5.40
MFX	4.28	7.79	0.037	3.62
ME	8.99	11.33	-1.66	11.24

(a) *Normal regime.*

	$\widehat{\mu}$ % $P(H_0)$	$\widehat{\sigma}$%	S	K
VB	1.33 ($p = 0.36$)	3.08	0.32	3.20
VFX	4.53 ($p = 0.65$)	9.04	0.18	2.88
VE	6.80 ($p = 0.43$)	17.32	0.3	3.82
CB	4.28 ($p = 0.08$)	2.77	-0.15	3.59
CFX	0.00 ($p = 0.04$)	10.29	-0.99	6.66
CE	12.95 ($p = 0.59$)	14.65	0.34	3.66
MB	0.00 ($p = 0.93$)	4.30	0.92	7.43
MFX	0.00 ($p = 0.33$)	10.18	0.47	6.92
ME	9.64 ($p = 0.78$)	25.77	-0.39	3.87

(b) *Turbulent regime.*

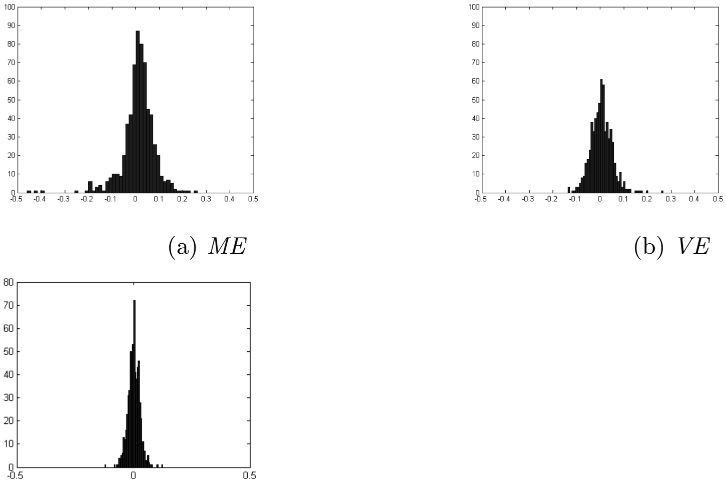
Table 3.3: Data specifications per regime.

the same expected value across regimes and differ only in volatility, which is not much different either for bonds in particular. However, the strategies invested in equity clearly exhibits volatility with a linear looking dependence on the underlying volatility index. This suggests that regime switching will be successful. As for the others they still perform somewhat worse in the turbulent regime, so at least some improvement in performance should be seen in every strategy.

3.3 Fama French Research Data

3.3.1 Introduction

In the Fama French library there is no Carry strategy, so instead we use the Size premium. Table 3.3 shows the mean, standard deviation, skewness and kurtosis of the data from Fama French. The highest but riskiest returns is gotten for the Momentum strategy, while the lowest but safest is gotten for the Size strategy. As is seen in Figure 3.3a the returns in *ME* has rather fat tails, which is reflected in the kurtosis that is considerably larger for *ME* than for the



(a) *ME*

(b) *VE*

(c) *SE*

Figure 3.3: Distribution of returns to risk premia portfolios from Fama French.

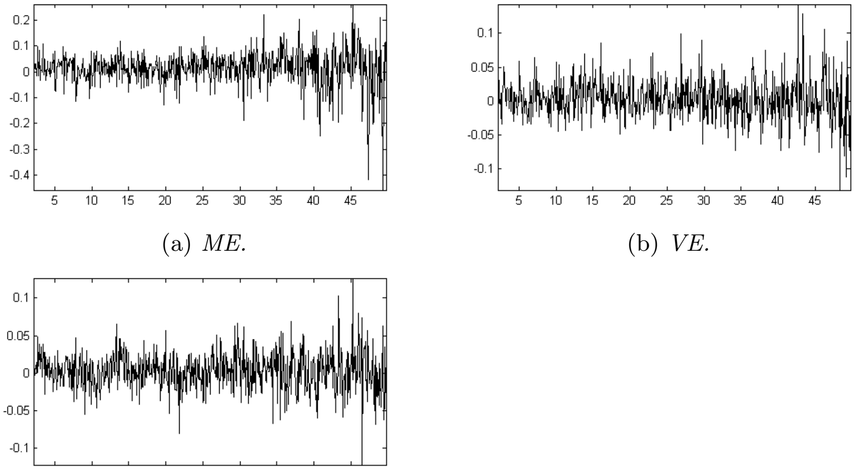
	$\widehat{\mu}\%$	$\widehat{\sigma}\%$	S	K
ME	16.77	23.80	-1.50	11.36
VE	4.16	10.36	0.17	5.14
SE	2.92	8.76	0.02	4.97

Table 3.4: Data specification for risk premia returns from Fama French.

others. Its skewness is similar to that of *ME* in the Deutsche Bank data, where it is one of the few to be skewed to the negative side.

3.3.2 Historical Regimes

The Fama French return series is plotted against underlying volatility of the S & P 500 VIX in Figure 3.4. The data in Table 3.5 shows again that the volatility of the equity strategies get larger when the underlying market volatility grows larger. No strategy shows any definitive difference in its mean, as the reported p-values from the t-test are all somewhat large. The *VE* is the closest to have a significant difference with the probability of the means being the same is about 13% while the same probability for *ME* is 81% and *SE* is 56%.



(a) *ME.*

(b) *VE.*

(c) *SE.*

Figure 3.4: Returns time series as function of underlying volatility.

	$\widehat{\mu}\%$	$\widehat{\sigma}\%$	S	K
ME	15.66	13.86	-0.40	3.68
VE	6.93	8.00	0.33	3.32
SE	3.78	6.89	0.03	3.17

(a) *Normal regime.*

	$\widehat{\mu}\%$ $P(H_0)$	$\widehat{\sigma}\%$	S	K
ME	17.46 ($p = 0.81$)	33.46	-1.43	9.02
VE	2.30 ($p = 0.13$)	11.64	0.19	4.88
SE	2.30 ($p = 0.56$)	9.80	0.05	4.76

(b) *Turbulent regime.*

Table 3.5: Data specifications per regime.

Chapter 4

Results

4.1 Introduction and Structure

The reader should make sure to read Sections 2.1, 2.3, 2.4 and 2.5 in order to be able to understand the terms used.

In this section we apply the methods introduced in Chapter 2. In order to test the results we employ the Monte Carlo Bootstrap to get an estimate of the statistical significance of the results.

We evaluate the performance in terms of the measures in Section 2.3; Information Ratio *IR* (2.9), Compounded Annual Growth Rate *CAGR* (2.10), Annual Volatility *Vol* (2.11), Sharpe Ratio *SR* (2.12), Calmar Ratio *CR* (2.13), Maximum Drawdown *MaxDD* (2.14) and the Expected Shortfall *ES* at level 5%.

All graphs show the cumulated sum of log-returns plotted against the date of return and the risk free rate is assumed to be zero percent in excess of 1 ($R_0 = 1.00$).

The historical data used for estimating covariance and expected value is 5 years (60 monthly returns) for both the Deutsche Bank data and Fama French data. For reference, the graphs are also transformed in order to have the same overall annual volatility of 10 % and presented alongside the simulations in order to get a visual sense of performance when the strategies have the same risk. This is done by first calculating the weight at each time point and then calculating the volatility $\widehat{\sigma}$ of the resulting returns after multiplying each weight with its corresponding risk premia return, after which each weight is scaled by $0.1/\widehat{\sigma}$ and the resulting return of using those weights are plotted alongside the original results.

4.2 Simulations

4.2.1 Regime Switching

Deutsche Bank

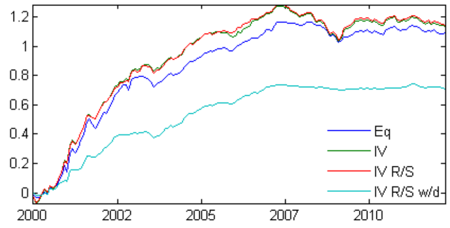
For the Deutsche Bank data we only use the S & P 500 VIX in order to estimate the turbulence probabilities because of the limited amount of data available. If we want 5 years of historical data for the covariance matrix and expected return, and create our own turbulence index also using 5 years, only 7 years is left to use for simulation. That is very few data points so we stick only to the S & P 500 VIX.

4.2.2 Equity

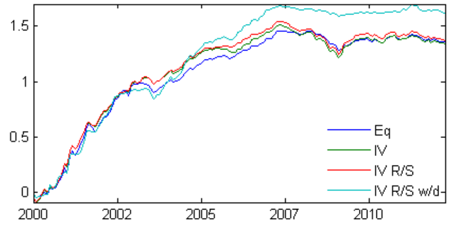
Simulating on historical equity data we get the results seen in Figure 4.1 on page 41.

All strategies perform roughly equal in the equity market which is a bit disappointing. Considering the whole period that the simulation is run (on 12 years) the standard deviation of the equity strategies are quite similar. The volatility of the strategies are rather similar so on average the *IV* allocates 37.10 % of the capital to the Value strategy, 43.35 % to the Carry strategy and 19.55 % to Momentum, reflecting the Carry strategy's lowest volatility. The regime switch method allocates on average 38.58 % to Value, 40.59 % to Carry and 20.83 % to Momentum. The *IV* strategy delivers an *IR* value of 1.14 and *SR* 1.19 while both *Eq* and *IV R/S* has the *IR* value 1.16 and *SR* 1.21. All the strategies miss the financial crisis of 2008 as we see in the figure, and since the *IV* and *IV R/S* methods had generated a higher prior return they fell the strongest to produce a *CR* of 0.44 as opposed to 0.71 produced by *Eq* due to a maximum drawdown of *IV* and *IV R/S* of about 22 % as opposed to 13 % for *Eq*. The Monte Carlo *p*-values are generally lower for the *IV R/S* which means that the results for *IV R/S* have less to do with chance than it has for the *IV*. The *CR* in particular has a very high *p*-value of over 0.8 for both *IV* and *IV R/S*. This goes hand in hand with neither strategy being able to tackle the financial crisis in any productive way.

A smart way of overcoming this is to exploit the results from Section 3.2.2 where we saw that the equity strategies exhibited no significant change in expected value across the different regimes, but showed a volatility that got higher the higher the volatility of the underlying volatility index. Since we have a way of predicting the value of the volatility index at time t when we are at time $t-1$ it seems like a good idea to invest less in an equity strategy the higher the probability is of a turbulent regime. The reported probability is higher the more the current volatility differs from past volatility which means we can use the reported probability directly to determine how much we should invest in the risky assets and how much we should save in the risk free asset. A simple straight forward method would be to invest an amount of cash according to this



(a) *Cumulated return series.*



(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium

	IR	*SR*	*CR*	
Eq	1.16	1.21	0.71	
IV	1.14	1.19	0.44	
IV R/S	1.16	1.21	0.44	
IV R/S w/d	1.36	1.39	1.42	

Monte Carlo p-values

	IR	*SR*	*CR*	
IV	0.59	0.62	0.85	
IV R/S	0.47	0.49	0.87	
IV R/S w/d	0.00	0.00	0.01	

Performance Measures

	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
Eq	9.20	7.96	3.14	13.01
IV	9.60	8.43	3.52	21.77
IV R/S	9.65	8.30	3.55	21.72
IV R/S w/d	5.84	4.30	1.40	1.42

Data

	Skew	*Kurtosis*		
Eq	1.40	7.66		
IV	1.14	6.57		
IV R/S	1.12	6.60		
IV R/S	2.16	11.63		

(c) *Data.*

Figure 4.1: Performance in the equity market.

linear function

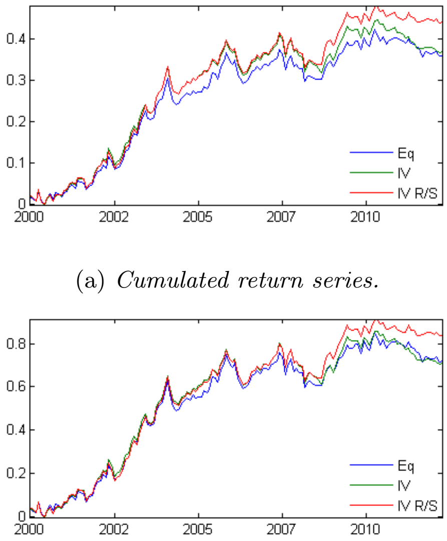
$$w_i = (1-p) \cdot w_i^{IV R/S}$$

where p is the probability of turbulence, defined in Section 2, Equation (2.15). This means that we are more invested in the strategy when we expect a calm period and less invested otherwise. We call this strategy the Inverse Volatility with regime switching and with deleverage (*IV R/S w/d*), with the word deleverage reflecting the choice to not invest all available money in risky assets. Simulating this strategy we get the results for the *IV R/S* that can be seen in Figure 4.1 along with the other results. The effect of the financial crisis is almost non existent, since the probability of turbulence was so high. In fact this strategy gives us an *IR* of 1.36, a twenty point improvement from before. The *SR* is 1.39, which means that the two differ less (0.03) than before (0.05) which reflects the lessening of the impact of the financial crisis. In fact, the maximum drawdown that was 22 % before, now is only 1.42 % which is an incredible improvement. This causes the *CR* to go from 0.44 to 1.42. The return distribution shows a larger skew and kurtosis that before. With the *ES* having dropped from 3.55 to 1.42 this implies the kurtosis reflects a fatter right tail, which is fine. The p-values from the Monte Carlo bootstrap simulation are now very small indicating greater statistical significance. The *IV R/S* now allocates on average 19.09 % to Value, 18.38 % to Carry and 9.78 % to Momentum, while saving 52.76 % of capital to do other things with. These numbers of course differ a lot during the period, being proportional to the probability of turbulence. A major drawdown to this then is the cost of rebalancing, as it is done frequently and might be large sums.

Foreign Exchange

Simulating on historical FX data we get the results seen in Figure 4.2, on page 43.

In the foreign exchange market we can not expect to do the same thing as we did in the equity market as neither strategy exhibits the same dependency in its volatility on the S & P 500 VIX. The *IV* method allocates on average 31 % of capital to Value, 43.89 % to Carry and 25.11 % to Momentum. The *IV R/S* allocates essentially the same way on average. Forming $\Delta = w_i^{IV R/S} - w_i^{IV}$ we evaluate the standard deviation of the difference in allocation to 6.74 % in Value, 9.16 % in Carry and 3.26 % in Momentum which means that *IV R/S* uses the Carry strategy in particular to exploit possible regime changes. Indeed, the *IR* is essentially the same for *Eq* and *IV* with 0.60 and 0.62, while *IV R/S* produces an *IR* value of 0.71. The *SR* is the same for *Eq* and *IV* with 0.64 where *SR* for *IV R/S* is 0.74. This reflects a drop at the financial crisis that was larger for the *IV* than for the *IV R/S* which captured it better. But all strategies dropped during the crisis, and where again both *IV* and *IV R/S* had reached higher peak values before the crisis suffered a larger drawdown with 8.88 % for *IV* and 8.26 % for *IV R/S*, while *Eq* which dropped from a lower peak having a maximum drawdown of 7.29 %. The differences here are however



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium

	IR	*SR*	*CR*	
Eq	0.62	0.64	0.40	
IV	0.60	0.64	0.34	
IV R/S	0.71	0.74	0.44	

Monte Carlo p-values

	IR	*SR*	*CR*	
IV	0.39	0.39	0.28	
IV R/S	0.07	0.07	0.09	

Performance Measures

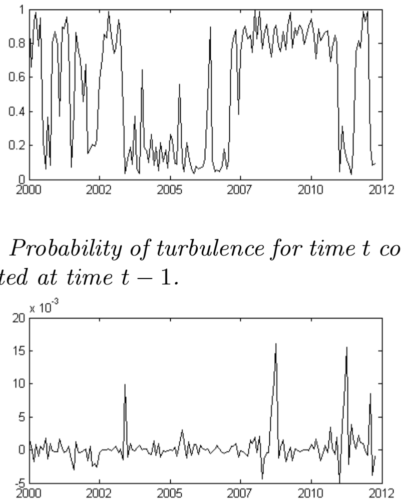
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
Eq	2.95	4.80	2.89	7.29
IV	3.03	5.02	3.19	8.88
IV R/S	3.62	5.08	3.16	8.26

Data

	Skew	*Kurtosis*		
Eq	-0.26	2.70		
IV	-0.35	3.17		
IV R/S	-0.32	3.23		

(c) *Data.*

Figure 4.2: *Performance in the FX market.*



(a) *Probability of turbulence for time t computed at time $t-1$.*

(b) *Residual returns between IV R/S and IV.*

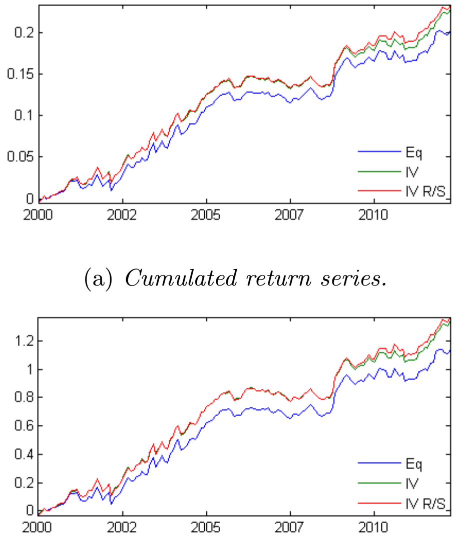
Figure 4.3

less that they were in the equity market. Its hard to say that either strategy rode out the financial crisis better than any one else by these numbers alone but according to the simulation the *IV R/S* was able to quickly identify when the risk was temporarily lower and exploit it. In fact, the Monte Carlo p-value is 0.39 for the *IV* and 0.07 for *IV R/S* which means that *IV R/S* did indeed exploit the possibility of different risk estimates in a meaningful way. In fact the *IV R/S* generally performed better than the others during switches from clearly turbulent regimes to clearly normal regimes and vice versa, that is where the probability of turbulence from period $t-1$ to t differs considerably. This can be seen comparing the graph of probabilities to the graph of residuals between the returns o *IV R/S* and *IV*, with a positive value reflecting a greater return for *IV R/S*. This comparison can be seen in Figure 4.5. With some exceptions, positive peaks in the residual is seen when regime probability quickly changes while the negative peaks are generally lower.

Bonds

Simulating on historical bond data we get the results seen in Figure 4.4, on page 45.

As in the foreign exchange market neither strategy exhibits the dependency on underlying volatility that they did in the equity market. The *IV* allocates on average 36.22 % to Value, 34.85 % to Carry and 28.93 % to Momentum, while *IV R/S* again allocates roughly the same. The standard deviation between



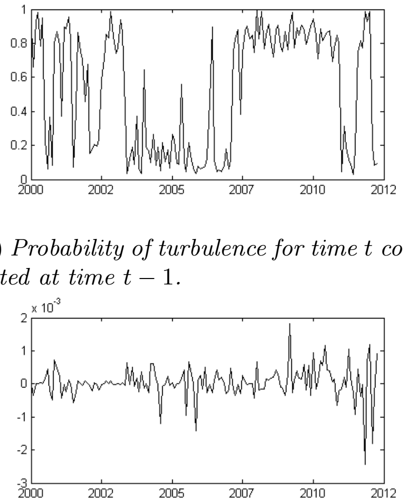
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
Eq	0.97	0.98	0.88	
IV	1.13	1.15	1.00	
IV R/S	1.16	1.17	1.02	
Monte Carlo p-values				
	IR	*SR*	*CR*	
IV	0.01	0.01	0.17	
IV R/S	0.01	0.01	0.14	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
Eq	1.65	1.70	0.96	1.87
IV	1.85	1.63	0.89	1.84
IV R/S	1.90	1.64	0.89	1.86
Data				
	Skew	*Kurtosis*		
Eq	0.02	4.00		
IV	0.02	3.98		
IV R/S	0.01	3.91		

(c) *Data.*

Figure 4.4: *Performance in the bond market.*



(a) *Probability of turbulence for time* t *computed at time* $t-1$.

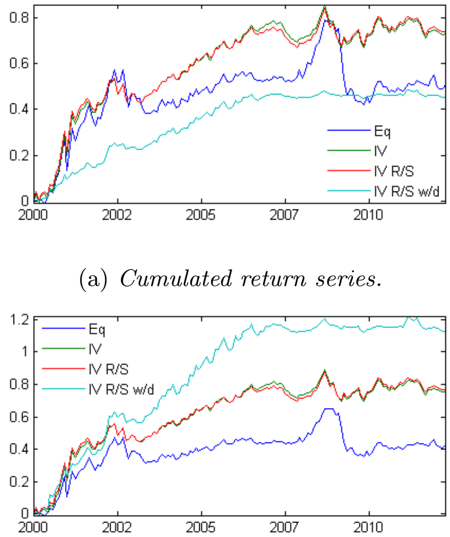
(b) *Residual returns between IV R/S and IV.*

Figure 4.5

allocations are only 3.50 % in Value, 3.16 % in Carry and 3.65 % in Momentum which implies that the regime switch did not find many opportunities to exploit the lesser or higher risk. Nevertheless, both strategies successfully allocated capital in a slightly different way than *Eq* for *IR* values of 1.13 for *IV* and 1.16 for *IV R/S*, as compared to 0.97 for *Eq*. The same *SR* were 1.15, 1.17 and 0.98 in the same order. All specifications for *IV* and *IV R/S* are essentially the same, not surprising since they allocate the capital in much the same way. Looking at Figure 4.5 we do not see the same behaviour in the residual peaks as in foreign exchange implying that the regime switch method was not able to be used in a meaningful way. Instead, *IV* was a very good method to use since the reported p-values for *IV* and *IV R/S* were essentially the same at 0.01 implying equal risk weighting worked well in this case.

4.2.3 Fama French Data - Regime Switching

To confirm or dispute the results from the Deutsche Bank data we run the same simulations on the data from Fama French. First we use the same time period in order to get comparable results and then we run them on the full time period. There are only equity strategies in the Fama French library so we expect that investing less in turbulent periods will be a good idea.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
Eq	0.34	0.41	0.13	
IV	0.63	0.68	0.35	
IV R/S	0.67	0.72	0.40	
IV R/S w/d	0.98	1.01	1.22	
Monte Carlo p-values				
	IR	*SR*	*CR*	
IV	0.05	0.08	0.03	
IV R/S	0.04	0.05	0.03	
IV R/S w/d	0.00	0.00	0.00	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
Eq	4.17	11.34	8.87	30.92
IV	6.05	9.64	5.75	17.13
IV R/S	6.43	9.62	5.46	16.22
IV R/S w/d	4.12	4.19	2.03	3.37
Data				
	Skew	*Kurtosis*		
Eq	-0.56	5.63		
IV	0.21	5.32		
IV R/S	0.27	5.08		
IV R/S w/d	1.00	5.88		

(c) *Data.*

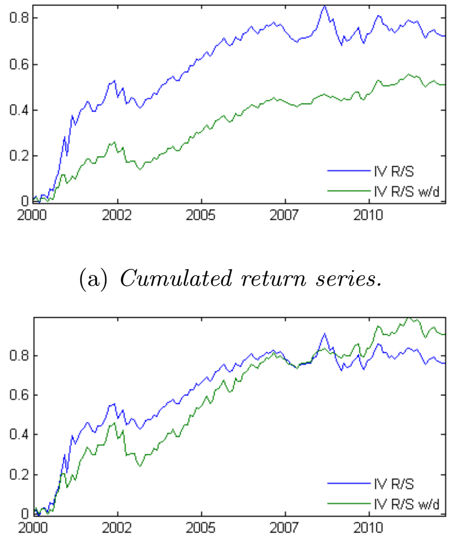
Figure 4.6: Performance in the equity market in Fama French universe.

Same Time Period 1995-2012

Simulating on historical Fama French equity data we get the results seen in Figure 4.6, on page 47.

The figure shows the performance of the *Eq* and *IV* method, and as can be seen *IV* outperforms *Eq* with quite some marginal. This is due to the much higher volatility in the Momentum strategy in comparison to Value and Size. The *IV* allocates on average only 8.06 % to the Momentum strategy while putting 58.63 % in Value and 38.31 % in Size. The standard deviation in allocation in Momentum is 3.40 % which means that *IV* keeps it low throughout the period. Meanwhile the standard deviation in both Value and Size is about 15 %. As for the *IV R/S* it allocates on average 10.11 % to Momentum, 53.71 % to Value and 36.17 % to Size. This somewhat higher allocation to Momentum seems to be strategically made since all measures are improved from *IV*. The *Eq* delivers an *IR* of only 0.34, *SR* of 0.41 and *CR* of 0.13. Being a "dumb" allocation method it performs during the financial crisis just like its underlying time series, poorly, for a maximum drawdown of almost 31 %. Meanwhile the *IV* does this time perform rather well during the crisis for a drawdown of 17 %. This causes *IV* to deliver an *IR* of almost twice that of *Eq* with 0.63. The *CR* is almost thrice improved with 0.35 thanks to the smaller maximum drawdown. The *IV R/S* delivers comparable results, clearing the financial crisis only marginally better for a maximum drawdown of 16 % with most results identical to *IV*. This is just like with the Deutsche Bank data that the two strategies perform pretty much identical. They both have roughly the same *p*-values from the Monte Carlo Bootstrap of 0.05 which is very good.

When investing less in risky assets proportionally to the probability of turbulence we do in fact get similar improvements as we did with the Deutsche Bank data. The *IV R/S w/d* allocates on average 53.09 % to pure cash, only 5.05 % to Momentum, 27.96 % to Value and 13.90 % to Size. However, the allocation to cash differs with a standard deviation of 35.45 %, to Momentum with 5.74 %, to Value with 25.19 % and 11.11 % to Size. This means that there is a lot of strategic allocation going on and it does deliver. We get an *IR* of 0.98 which is almost thrice that of *Eq* and an almost 50 % improvement from *IV* and *IV R/S*. At the same time, while hampering growth it does still deliver some return with a *CAGR* of 4.12 % down from the *CAGR* of *IV R/S* of 6.43 %. As the financial crisis showed a very large probability of turbulence the *IV R/S w/d* allocates away much to cash and rides it out very well. In fact, in January 2008 83.82 % was allocated to cash while a few years later in July 2011 only 6.61 % was allocated to cash. The risk of the strategy is thus also reduced, where *IV R/S* had an expected shortfall of 5.46 % the *IV R/S w/d* showed only 2.03 %. All in all, the results are comparable to those showed in Deutsche Bank data with the exception of *Eq* performing much worse due to the Momentum strategy's higher relative volatility.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
IV R/S	0.64	0.69	0.37	
IV R/S w/d	0.77	0.80	0.38	
Monte Carlo p-values				
	IR	*SR*	*CR*	
IV R/S	0.05	0.06	0.03	
IV R/S w/d	0.02	0.01	0.12	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
IV R/S	5.99	9.42	5.56	16.25
IV R/S w/d	4.17	5.42	3.35	11.06
Data				
	Skew	*Kurtosis*		
IV R/S	0.26	5.58		
IV R/S w/d	-0.28	5.78		

(c) *Data.*

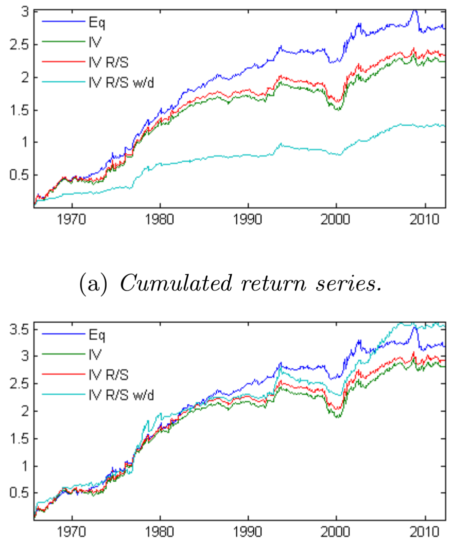
Figure 4.7: Performance in the equity market in Fama French universe using the Mahalanobis distance as volatility index.

Using Mahalanobis Distance

Simulating on historical Fama French equity data with the Mahalanobis distance we get the results seen in Figure 4.7, on page 49.
Due to having more data we check if results can be improved by generating our own volatility index on historical data. We use 5 years additional data to create our volatility index. It is very interesting to see that, although not as prominent, we still get a clear improvement from switching to cash. The *IR* and *SR* is improved for 0.64 and 0.69 to 0.77 and 0.80 respectively with an improvement in statistical significance. The fact that results are comparable to those using the S &P 500 VIX suggests that we might find the same improvement in bonds and FX by generating our own index for them. Unfortunately we do not have enough data to try this since the Fama French library only contains equity data.

4.2.4 Fama French Data - Regime Switching, 1960-2012

Simulating on historical Fama French equity data using all data available we get the results seen in Figure 4.1, on page 41.
We check if we get similar results when expanding the simulations to include data prior to 1995, going back to 1960. Regime switching still performs roughly the same as *IV* but responds well, again, to investing less in risky assets during turbulent times. The *IR* value for the period is reported as about 0.6 for both *IV* and *IV R/S* but investing less as function of turbulence probability raises this to 0.80 while reporting improved statistical significance. This time however neither method handles the financial crisis of 2008 perfectly as some drawdown is seen for both. Overall results are indeed comparable to the more recent time period suggesting further that regime switching is a good idea.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
Eq	0.71	0.76	0.20	
IV	0.63	0.67	0.14	
IV R/S	0.65	0.69	0.15	
IV R/S w/d	0.80	0.82	0.17	
Monte Carlo p-values				
	IR	*SR*	*CR*	
IV	0.10	0.10	0.34	
IV R/S	0.05	0.05	0.02	
IV R/S w/d	0.00	0.00	0.00	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
Eq	6.09	8.53	5.84	30.92
IV	4.89	7.82	4.64	35.30
IV R/S	5.13	7.88	4.71	33.83
IV R/S w/d	2.72	3.39	1.82	15.81
Data				
	Skew	*Kurtosis*		
Eq	-0.68	7.92		
IV	0.09	6.26		
IV R/S	0.02	6.47		
IV R/S w/d	0.54	9.36		

(c) *Data.*

Figure 4.8: Performance in the equity market in Fama French universe using all data.

4.2.5 Regression Trees

Using the tree methods in Section 2.5 we predict the Sharpe Ratio for month t in three ways; assuming we know the macro-economic state for time t at time t, assuming we only know the macro-economic states up to time $t-1$ and assuming we only know the macro-economic states up to time time $t-3$. This is due to the fact that sometimes macro-economic data is reported with delay. For example GDP might be reported for March in April or June, so we run the simulations assuming the GDP for March is reported in March, April and June. This means that with "no delay", the macro-economic state of time t corresponds to the risk premia return at time t, with "one month delay" the macro-economic state at time $t-1$ is taken as surrogate for the time t value, and thus corresponds to the risk premia return at time t. For "three months delay" data from one quarter ago is used instead of data from time t and corresponds to the risk premia return at time t. We then weigh across assets proportionally to the predicted Sharp Ratio with nothing invested in an asset if a negative Sharp Ratio is predicted. So we allocate as follows,

$$w_i = \gamma_i / \sum_{j=1}^{n} \gamma_j$$

where γ_i is the predicted Sharp value for one asset, with n total assets.

We use data constructed in the same way as in [2]. The construction of this data is described in Appendix A. We use the S & P 500 VIX, the US money supply (m1), ISM manufacturing and non-manufacturing index, US GDP growth, government-, household- and private debt. All of this data is transformed into monthly returns and the Z-score of the S & P 500 VIX and M1 money supply is also used. Again, specific construction can be seen in either [2] or, briefer, in Appendix A.

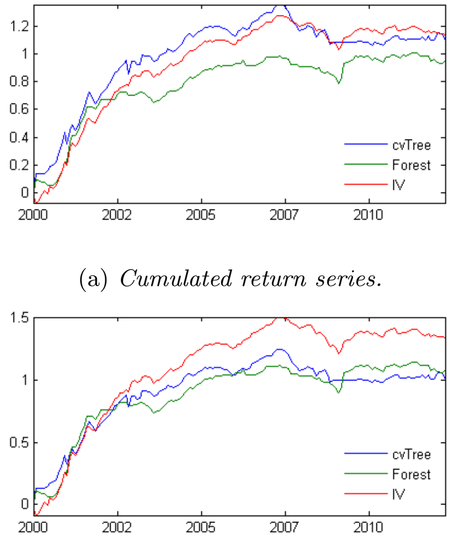
In the simulations to come, the cross validation method is written as *cvTree* and the bagging method is written as *Forest*.

Deutsche Bank Data

Equity

Simulating on historical equity data we get the results seen in Figures 4.9, 4.10 and 4.11 on pages 54, 55 and 56.

In Figure 4.9 simulation is done with no delay, meaning that we split the returns on macro-economic data of the same month as the returns. The results do not beat the IV strategy for either the cross validation trees or the bagging method. The only noteworthy thing is that the statistical significance in the results is somewhat higher, so the methods used underlying factors in a slightly more meaningful way. However, results are still worse than the IV strategy. Results are actually somewhat improved for the bagging method with three months delay, while the cross validation method produces only disappointing results. The results are over all poor in comparison to the IV strategy which is most likely due to the small number of data points used in the simulation. Simulations are done on a minimum of 60 and maximum of 148 data points, which produce rather rough estimates of the Sharpe Ratio.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium

	IR	*SR*	*CR*
cvTree	0.85	0.91	0.36
Forest	0.91	0.96	0.44
IV	1.14	1.19	0.44

Monte Carlo p-values

	IR	*SR*	*CR*
cvTree	0.35	0.33	0.48
Forest	0.27	0.30	0.47
IV	0.59	0.62	0.85

Performance Measures

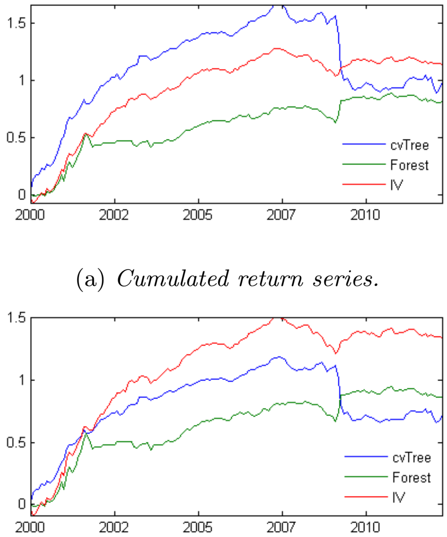
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	9.22	10.88	6.04	25.47
Forest	7.98	8.74	3.15	17.97
IV	9.60	8.43	3.52	21.77

Data

	Skew	*Kurtosis*
cvTree	0.46	5.40
Forest	2.29	11.82
IV	1.14	6.57

(c) *Data.*

Figure 4.9: Performance in the equity market using regression trees with no delay.



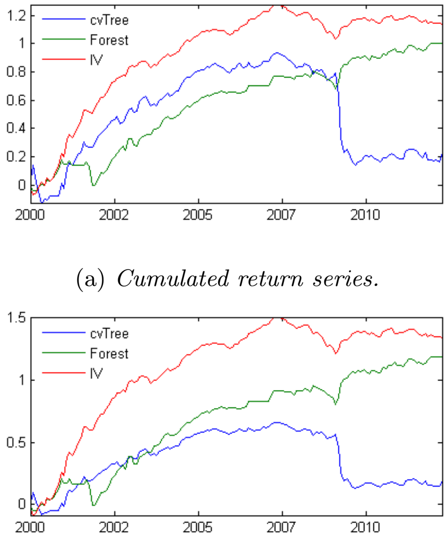
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.59	0.67	0.15	
Forest	0.72	0.78	0.49	
IV	1.14	1.19	0.44	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.67	0.65	0.86	
Forest	0.81	0.82	0.53	
IV	0.59	0.62	0.85	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	8.28	14.12	10.51	53.70
Forest	6.73	9.30	4.72	13.83
IV	9.60	8.43	3.52	21.77
Data				
	Skew	*Kurtosis*		
cvTree	-1.96	16.10		
Forest	1.48	9.20		
IV	1.14	6.57		

(c) *Data.*

Figure 4.10: Performance in the equity market using regression trees with one month delay.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

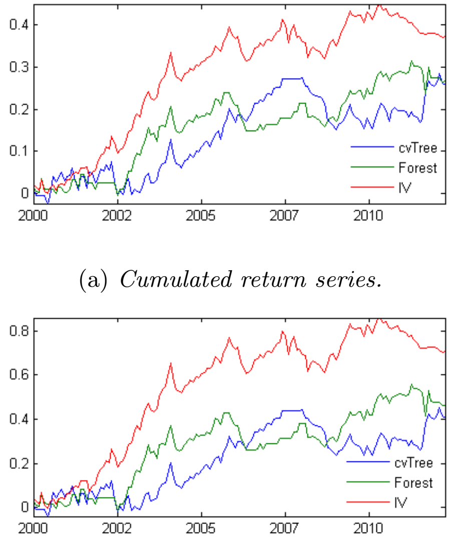
Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.13	0.21	0.03	
Forest	1.00	1.06	0.52	
IV	1.14	1.19	0.44	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.98	0.98	0.98	
Forest	0.21	0.23	0.38	
IV	0.59	0.62	0.85	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	1.90	14.40	11.28	54.81
Forest	8.40	8.36	4.58	16.22
IV	9.60	8.43	3.52	21.77
Data				
	Skew	*Kurtosis*		
cvTree	-1.63	14.19		
Forest	0.43	9.76		
IV	1.14	6.57		

(c) *Data.*

Figure 4.11: Performance in the equity market using regression trees with three months delay.

Foreign Exchange

Simulating on historical equity data we get the results seen in Figures 4.12, 4.13 and 4.14 on pages 58, 59 and 60.
The simulations have the same problems as with the equity data as there are very few data points. Almost all measures are poor in comparison to the *IV* strategy, while the bagging method outperforms the cross validation method. Just like in equity the cross validation method steadily performs worse with higher delay, as can be seen in all measures. The bagging method perform the best with one month delay this time around, but judging from the Monte Carlo bootstrap values this is most likely just a coincidence. Again, more data points would probably produce more stable results.



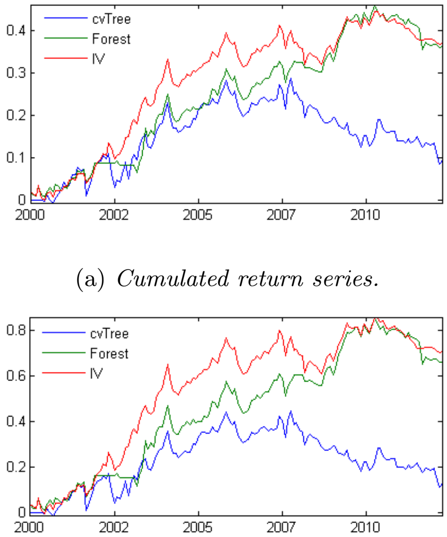
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.35	0.38	0.18	
Forest	0.40	0.43	0.25	
IV	0.60	0.64	0.34	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.57	0.56	0.47	
Forest	0.58	0.58	0.38	
IV	0.39	0.39	0.28	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	2.08	5.97	3.46	11.44
Forest	2.17	5.40	3.18	8.58
IV	3.03	5.02	3.19	8.88
Data				
	Skew	*Kurtosis*		
cvTree	0.15	3.92		
Forest	0.22	4.93		
IV	-0.35	3.23		

(c) *Data.*

Figure 4.12: Performance in the FX market using regression trees with no delay.



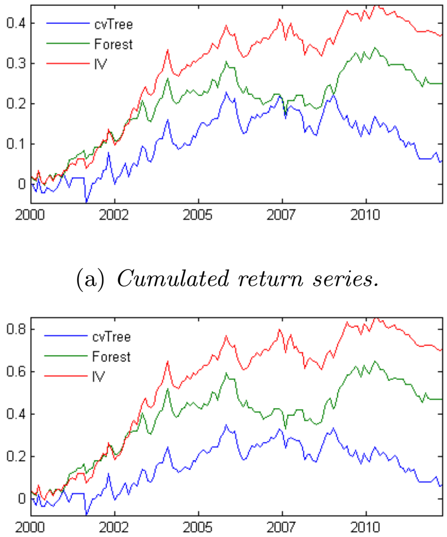
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.12	0.16	0.04	
Forest	0.57	0.58	0.29	
IV	0.60	0.64	0.34	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.90	0.86	0.86	
Forest	0.37	0.40	0.46	
IV	0.39	0.39	0.28	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	0.77	6.19	4.19	18.21
Forest	2.95	5.21	3.17	10.06
IV	3.03	5.02	3.19	8.88
Data				
	Skew	*Kurtosis*		
cvTree	-0.39	3.86		
Forest	0.07	5.04		
IV	-0.35	3.23		

(c) *Data.*

Figure 4.13: Performance in the FX market using regression trees with one month delay.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

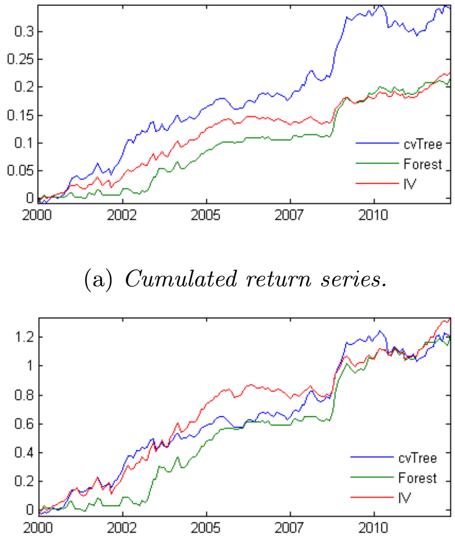
Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.07	0.10	0.03	
Forest	0.41	0.44	0.17	
IV	0.60	0.64	0.34	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.92	0.93	0.93	
Forest	0.53	0.53	0.64	
IV	0.39	0.39	0.28	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	0.46	6.31	4.19	16.01
Forest	2.04	4.99	2.99	12.15
IV	3.03	5.02	3.19	8.88
Data				
	Skew	*Kurtosis*		
cvTree	-0.71	4.03		
Forest	0.01	4.89		
IV	-0.35	3.23		

(c) *Data.*

Figure 4.14: Performance in the FX market using regression trees with three months delay.

Bonds

Simulating on historical equity data we get the results seen in Figures 4.15, 4.16 and 4.17 on pages 62, 63 and 64.
For the first time results show a high statistical significance in the results, with the bagging method producing Monte Carlo bootstrap values of below 0.1 and in particular outperforms the IV method for the Calmar ratio due to having a smaller maximum drawdown. These results are only seen in Figure 4.15 however, with no delay which is an ideal situation. Extending to one and three months delay we again see very statistically insignificant results. All in all, results from the tree simulations using the Deutsche Bank data has failed to show an improvement over the IV either in risk premium or in statistical significance.



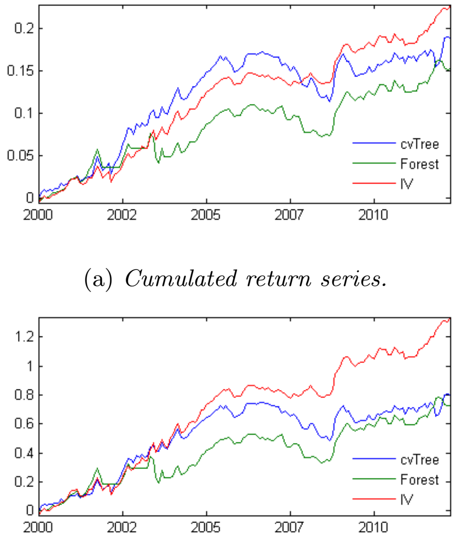
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	1.03	1.05	0.51	
Forest	1.03	1.05	1.34	
IV	1.13	1.15	1.00	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.15	0.13	0.49	
Forest	0.07	0.07	0.02	
IV	0.01	0.01	0.17	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	2.79	2.72	1.41	5.50
Forest	1.77	1.71	0.77	1.33
IV	1.85	1.63	0.89	1.84
Data				
	Skew	*Kurtosis*		
cvTree	-0.02	4.03		
Forest	0.05	4.01		
IV	0.02	3.98		

(c) *Data.*

Figure 4.15: Performance in the bond market using regression trees with no delay.



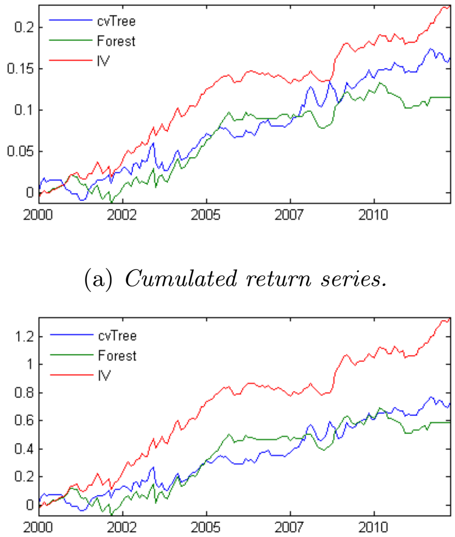
(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.69	0.70	0.27	
Forest	0.63	0.65	0.33	
IV	1.13	1.15	1.00	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.68	0.69	0.87	
Forest	0.88	0.86	0.91	
IV	0.01	0.01	0.17	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	1.53	2.22	1.13	5.63
Forest	1.24	1.95	1.21	3.70
IV	1.85	1.63	0.89	1.84
Data				
	Skew	*Kurtosis*		
cvTree	-1.12	5.23		
Forest	-0.57	5.87		
IV	0.02	3.98		

(c) *Data.*

Figure 4.16: Performance in the bond market using regression trees with one month delay.



(a) *Cumulated return series.*

(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium

	IR	*SR*	*CR*
cvTree	0.63	0.65	0.40
Forest	0.52	0.53	0.29
IV	1.13	1.15	1.00

Monte Carlo p-values

	IR	*SR*	*CR*
cvTree	0.83	0.85	0.70
Forest	0.64	0.66	0.61
IV	0.01	0.01	0.17

Performance Measures

	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	1.35	2.13	1.25	3.34
Forest	0.93	1.81	1.19	3.30
IV	1.85	1.63	0.89	1.84

Data

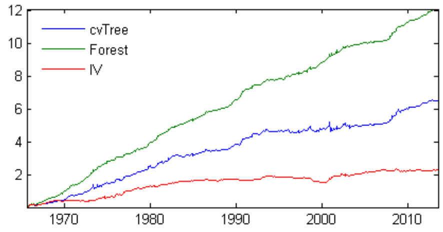
	Skew	*Kurtosis*
cvTree	-0.08	4.69
Forest	-0.61	5.67
IV	0.02	3.98

(c) *Data.*

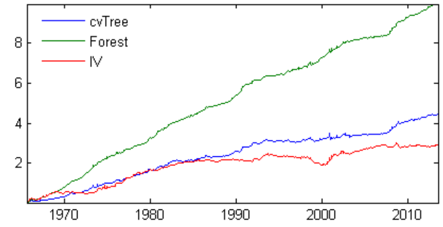
Figure 4.17: Performance in the bond market using regression trees with three months delay.

Fama French Data

We now have much more data and see more satisfying results when predicting Sharpe values. Results of simulations with Fama French data are seen in Figures 4.18, 4.19 and 4.20 on pages 66, 67 and 68.
The cross validation method steadily performs worse when predictions are made with greater delay. Assuming we know the current macro-economic state, the cross validation method gives us an *IR* of 0.95, as compared to 0.80 given by the *IV R/S w/d* method for Fama French data. However, the bagging method gives us an incredible 2.34 *IR* value which is of course a huge improvement from any other method previously. The $CAGR$ is up to about 28 % with a volatility of 12 %. Of course, the data would not have been known at the current date so the results are an ideal unrealistic case. Realistically we must look to the delayed simulations. When assuming data is only known up to time $t-1$ we simulate performance to get the results in Figure 4.19. We see results more in line with previous simulations with the bagging method giving us an *IR* value of 0.72. This is slightly worse than the regime switching counterpart with all other measures also being slightly worse. Finally, simulating with with the assumption that we only know the macro-economic values from one quarter back we see that the cross validation method gives an even worse result that previously reported but that the bagging method gives an improves result again. Since the Deutsche Bank data also exhibited an improvement when going from one month delay to three month delay this might not only be pure coincidence.



(a) *Cumulated return series.*

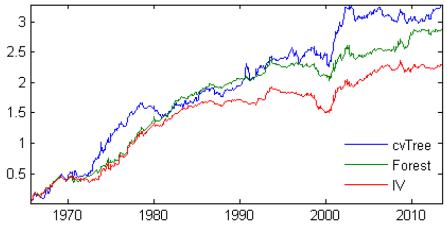


(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

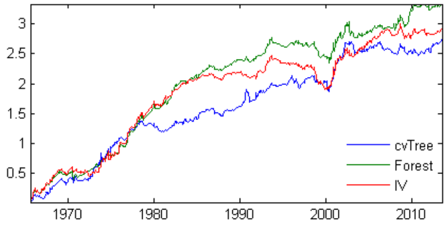
Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.95	1.05	0.34	
Forest	2.34	2.42	1.77	
IV	0.63	0.67	0.14	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.00	0.00	0.01	
Forest	0.05	0.00	0.00	
IV	0.10	0.10	0.34	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	14.50	15.21	9.70	42.31
Forest	28.54	12.21	7.70	16.14
IV	4.89	7.82	4.64	35.30
Data				
	Skew	*Kurtosis*		
cvTree	-1.72	22.15		
Forest	0.12	7.24		
IV	0.09	6.26		

(c) *Data.*

Figure 4.18: Performance in the equity market in Fama French universe using regression trees with no delay.



(a) *Cumulated return series.*

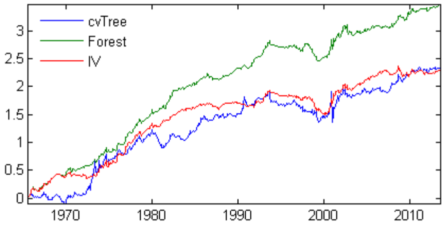


(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

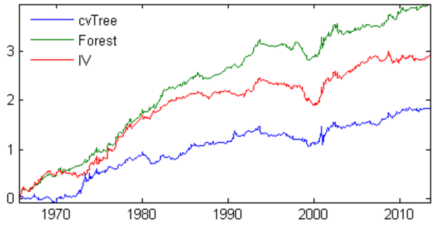
Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.58	0.64	0.25	
Forest	0.72	0.77	0.20	
IV	0.63	0.67	0.14	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.11	0.12	0.05	
Forest	0.24	0.23	0.31	
IV	0.10	0.10	0.34	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	7.08	12.28	7.76	28.45
Forest	6.16	8.54	5.62	30.10
IV	4.89	7.82	4.64	35.30
Data				
	Skew	*Kurtosis*		
cvTree	-0.12	6.97		
Forest	-0.22	6.65		
IV	0.09	6.26		

(c) *Data.*

Figure 4.19: Performance in the equity market in Fama French universe using regression trees with one month delay.



(a) *Cumulated return series.*



(b) *Cumulated return series scaled so all strategies have 10 % annual volatility.*

Risk Premium				
	IR	*SR*	*CR*	
cvTree	0.37	0.45	0.12	
Forest	0.86	0.91	0.25	
IV	0.63	0.67	0.14	
Monte Carlo p-values				
	IR	*SR*	*CR*	
cvTree	0.56	0.53	0.42	
Forest	0.01	0.00	0.11	
IV	0.10	0.10	0.34	
Performance Measures				
	CAGR	*Vol*	$ES_{0.05}(X)$	*MaxDD*
cvTree	4.99	13.34	8.63	42.31
Forest	7.45	8.66	5.38	30.33
IV	4.89	7.82	4.64	35.30
Data				
	Skew	*Kurtosis*		
cvTree	-2.27	30.87		
Forest	-0.32	9.19		
IV	0.09	6.26		

(c) *Data.*

Figure 4.20: Performance in the equity market in Fama French universe using regression trees with three months delay.

	Equity	*FX*	*Bond*
IV	15.43	24.38	24.80
IV R/S	69.28	82.83	81.71
IV R/S w/d	197.64	-	-
cvTree (no delay)	52.02	53.18	64.17
cvTree (1m delay)	51.71	50.28	46.92
cvTree (3m delay)	52.41	64.40	51.55
Forest (no delay)	68.24	62.11	69.77
Forest (1m delay)	45.02	37.76	38.43
Forest (3m delay)	61.38	59.28	55.89

Table 4.1: Yearly rebalancing volume for Deutsche Bank data.

4.2.6 A Simple Cost Analysis

In order to get a sense of rebalancing cost we perform the following analysis. At each time point t the weight $\mathbf{w}_t$ is saved which contain the allocation for the different risk premia strategies. The rebalancing cost for period t is proportional to the changes in allocation. For example, if 40 % is allocated to the Value premium and 60 % to the Momentum strategy at time $t-1$, and then changed to 35 % in Value and 65 % to Momentum at time t, then total rebalancing is proportional to $|40-35|+|60-65| = 5+5 = 10$ %. However, the cost of money in the bank is zero, so a position change from 60 % in cash to 65 % to cash only cost proportional to the assets that are sold off, the change in cash position does not cost anything. Using this method, the total rebalancing volume per year is presented in percentages in Table 4.1 for the Deutsche Bank simulations and in Table 4.2 for the Fama French simulations. As expected the *IV R/S w/d* rebalances more since it moves capital away from the risky assets in unstable times while *IV* and *IV R/S* allocates depending on the risk assessment of the current month which probably does not differ that much from the month before. With some smart rebalancing methods the cost might not have to be so great however, but that particular method will not be treated in this report.

The rebalancing cost of the tree methods are comparable to the other methods, with the winner of the tree methods, the bagging with no delay having slightly smaller rebalancing volume than the *IV R/S w/d*.

	1995-2012	1960-2012
IV	32.36	27.39
IV R/S	81.71	114.26
IV R/S w/d	191.48	259.87
cvTree (no delay)	-	244.33
cvTree (1m delay)	-	344.85
cvTree (3m delay)	-	325.04
Forest (no delay)	-	243.57
Forest (1m delay)	-	89.85
Forest (3m delay)	-	110.88

Table 4.2: Yearly rebalancing volume for Fama French data.

Chapter 5

Conclusions

5.1 Regime Switching

As we saw in Chapter 4 the *IV* and *IV R/S* performed roughly the same in all asset classes. This was a very disappointing result that suggests that the underlying risk factors were not found. However we did see in equity a way to leverage returns in calm financial periods and keeping risk low in turbulent ones. We did this by predicting the turbulence of next month and adding capital into the investment if we saw a low probability of turbulence and taking it away if not. This proved successful not only for the data provided by Deutsche Bank, but also for the Fama French data for the same time period and the full period since 1960. This method exclusively provided the best results in the Monte Carlo bootstrap simulation which suggests that the performances of the equity strategies really are closely connected to underlying volatility regimes. The strategies simply were already quite stable so the difference between turbulent regime and normal regime is not enough to really set the *IV* and *IV R/S* apart. The *IV R/S w/d* boosted the difference and thus got the regime switching method to really work. But not only did we see this result for the S & P 500 volatility index but when we created our own turbulence index we saw similar results. Investing as a function of the turbulence index also boosted the difference between *IV* and *IV R/S* in order to get better results. This result is equally as important as the first since it carries one important suggestion.

Since the FX and bond classes did not exhibit the same dependency on the S & P 500 volatility index, we can not repeat the same strategy and expect good results. This means that we had to rely on the *IV R/S* method which again proved to give similar results as the *IV*. But the performance in equity using the Mahalanobis turbulence index was independent from the S & P 500 volatility index and yet produced similar results. This could be taken as suggesting that similar result can be seen in FX and bonds. Unfortunately the Fama French library contains only data in equity and the Mahalanobis index could not be formed for FX and bonds. Accessing larger amounts of data for these strategies could hopefully help showing that the *IV R/S w/d* performs equally well in these as it did in equity.

Lastly one should not forget that I have set the risk free return to be zero which is not the case when depositing money at a bank. So actual returns will be slightly higher.

Weaknesses of this method is greater rebalancing cost, as probabilities differ from time to time. But this may be solved by modifying the investing function so that instead of investing all money at zero turbulence probability and no money at probability one, you create another more effective method. The result from this report shows that the probabilities can be exploited and further explorations into the subject may find an optimal way of doing it.

5.2 Regression Trees

The regression trees calculated from the Deutsche Bank data showed results that were highly unstable, both by common sense and by the Monte Carlo simulations. But the Sharp Ratio should be calculated on enough data, and 60 data points for generating the entire tree is simply not enough to get any kinds of stable results. The method was redeemed when using the Fama French data, which provided more data points and suddenly results were much more stable. The way the trees are formed is rather unorthodox and given the amount of data available it is hard to estimate the level of statistical significance in their predictions. This lead to the problem of determining how many data points should be in each end node of the trees. The variant of cross validation used in this report seems not to produce the best results, so one might conclude that it is not to be used. However, bagging the trees does not require a minimum to be set and thus does not have the same problem.

Essentially the trees find what macro-economic state gave the best Sharp Ratio for one specific time period and we simply assume that the same will hold for the next period so the predictions need not be exact as long as proportions are kept intact. In other words, it is enough to know that one strategy provides roughly twice the sharp value that another given a certain input of macro-economic data and it is not interesting what the values actually are. This seemed to work rather well in the Fama French library where the highest *IR* value of all was reached. That results however assumed that we had access to data the same time period as for when it was valid. In reality this is not the case. Instead it would be more realistic to use last months value that was received this month, or even the value of three months ago. Much data is reported quarterly, so not every month has an updated value. This might be the reason why the bagging method, which should be seen as the only working method, produced results that was worst for the one month delay.

In the end the results suggest that bagging the trees could very well work where bagging the trees with one quarter delay produces results similar to but somewhat lower than the *IV R/S w/d*.

Weaknesses of this method is that bagging does not produce intuitive results.

You do not get one tree that illustrates dependencies on underlying macroeconomic data, but a multitude of different trees that each was produced differently. So in terms of easiness of understanding it may not be the best method, especially since the regime switching method worked very well.

Appendix A

Macro-Economic Terms and Data

Share price
The price of one share of some financial asset, for example a company stock [1].

Book value
The total value of a company's assets not including intangible assets such as patents [2].

Dividend Yield
The total annual dividend that is paid per share [3].

12 months earnings
A company's net profit for the last 12 months [4].

12 months predicted earnings
What a company is expected to earn in the next 12 months [5].

Current account
A macro-economic term signifying the difference between a nation's savings and investment. Savings is a term referring to aggregated income that is not consumed and investment refers to the aggregated amount of income that is not consumed, but spent on durable goods and services such as factories or education [6].

[1] http://en.wikipedia.org/wiki/Share_price

[2] http://www.investopedia.com/terms/b/bookvalue.asp

[3] http://www.investopedia.com/terms/d/dividendyield.asp

[4] http://www.investopedia.com/terms/t/trailingpe.asp

[5] http://www.investopedia.com/terms/t/trailingpe.asp

[6] http://www.investopedia.com/terms/c/currentaccount.asp

Budget balance
The difference between a nation's spending and income [7].

Nominal yield on a bond
The interest rate that the issuer of a bond has promised to pay [8].

Duration Bias
The duration measures the sensitivity of the price of an instrument to a change in interest rates. The bias is calculated as the performance when going long a high duration instrument and shorting a low duration instrument [9].

Purchasing power parity (PPP)
If P_1 is the price of some good in country 1, and P_2 is the cost of the same good in country 2, then the PPP is calculated as P_1/P_2 [10].

G10 Currencies
USD, GBP, CAD, EUR (France), EUR (Germany), EUR (Italy), EUR (Netherlands), EUR (Belgium), JPY, SEK, CHF.

G10 Countries
USA, Canada, Japan,Australia, New Zealand, Germany, France, Italy, UK, Switzerland, Norway, Denmark and Sweden

LIBOR: London Interbank Offered Rate.

GDP
The yearly Gross Domestic Product (GDP) is the aggregated monetary value of all goods and services produced by a nation during one year.

M1 Money Supply
The aggregated amount of all assets that can quickly be converted to currency, such as cash and checking accounts [11].

ISM Manufacturing - ISM Non Manufacturing The ISM Manufacturing Index measures employment, production inventories, new orders and supplier deliveries based on surveys from more than 300 manufacturing firms [12]. The ISM Non Manufacturing does the same but for more than 400 non manufacturing firms.
Household/Corporate Debt
The aggregated amount that is owed to financial institutes by a countries citizens (household debt) or by the private sector (corporate debt).

[7] http://www.investopedia.com/terms/b/balanced-budget.asp

[8] http://www.investopedia.com/terms/n/nominalyield.asp

[9] https://index.db.com/htmlPages/DB_Duration_Bias_Guide_20050812.pdf

[10] http://www.investopedia.com/terms/p/ppp.asp

[11] http://www.investopedia.com/terms/m/m1.asp

[12] http://www.investopedia.com/terms/i/ism-mfg.asp

Federal Debt
The federal debt is in one part the amount that is owed domestic lenders and in one part debt that is owed foreign lenders. Lenders could be buyers of government bonds.

A.1 Macro-Economic Data

Thanks to Deutsche Bank, we have gotten access to the data that they use in [2] and use the tree method to find periods where the strategies work the best. The variables that are included in the regressions are constructed as follows.

Standard % Poor 500 volatility (VIX)
(Figure A.1a) This is the same volatility index as i used in Section 2.4 and is calculated as the monthly standard deviation of the S & P 500 index.

Z transformation of the S & P volatility (VolZ)
(Figure A.1b)The Z transformation measures rapid changes in volatility by computing how much the current value differs from its n month moving average, in relation to the 3 month standard deviation. Specifically, let $V(t)$ be the monthly volatility of S & P 500 at time t,

$$MA[V(t)] = \frac{1}{n}\sum_{i=1}^{n} V(t-i+1)$$

be the n month moving average and

$$std(V(t)) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(V(t-i+1) - MA[V(t)]}$$

be the n month standard deviation. As us used in [2] $n = 3$.

M1 growth rate (M1G)

(Figure A.1c) This is is year on year growth rate of the M1 money supply. If $m_1(t)$ is the M1 money supply at month t, then the year on year growth rate is calculated as

$$\frac{d}{dt}m_1(t) = \frac{m_1(t)}{m_1(t-12+1)} - 1$$

in the standard way.

Z transformation of M1 growth rate (M1Z)
(Figure A.1d) The Z transformation is done in the same way as above.

ISM index (ISM)
(Figure A.1e) We take the average of the ISM Manufacturing and ISM Non Manufacturing index to compute the ISM index.

GDP quarterly growth rate (GDP)
(Figure A.1f) Just like it sounds this is simply the quarterly growth in GDP.

Government debt and household/corporate credit(LGG)
(Figure A.1g) The ratio between government debt and household plus corporate credit demand. $F(t)$ represents federal debt, $H(t)$ household credit demand and $C(t)$ corporate credit demand. Then

$$LGG(t) = \frac{F(t)}{H(t) + C(T).}$$

The year on year growth rate $\Delta_{LGG}(t)$ is calculated by taking

$$\Delta_{LGG}(t) = \frac{LGG(t)}{LGG(t-12+1)} - 1$$

.

Household and Corporate debt(LBB)
(Figure A.1h) The ratio between household and corporate debt.

$$LBB(t) = \frac{H(t)}{C(t)}$$

and year on year growth rate

$$\Delta_{LGG}(t) = \frac{LBB(t)}{LBB(t-12+1)} - 1.$$

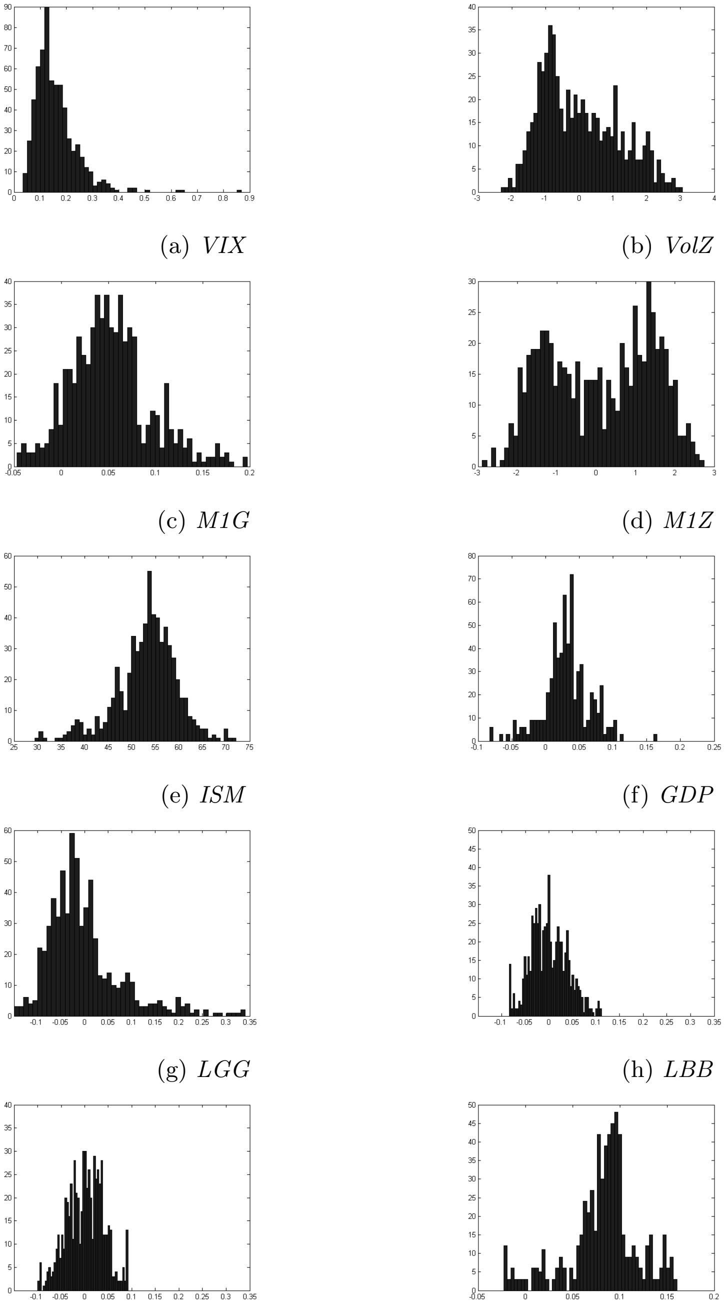
Corporate and household debt(LHH)

(Figure A.1i) The inverse of LBB.

Growth Rate of Household and Private Sector Credit Demand (CG)

(Figure A.1j) Let $D(t)$ signify the household and private sector credit demand at month t as $D(t) = HHD(t) + PSD(t)$, where HHD stands for household debt and PSD stands for private sector debt, then the yearly growth rate is

$$\frac{d}{dt}D(t) = \frac{D(t)}{D(t-12+1)} - 1$$



(a) *VIX*

(b) *VolZ*

(c) *M1G*

(d) *M1Z*

(e) *ISM*

(f) *GDP*

(g) *LGG*

(h) *LBB*

(i) *LHH*

(j) *CG*

Figure A.1: *Distribution of macro-economic variables.*

Appendix B

Mathematics

Rank Correlation

We use Spearman's ρ to estimate rank correlation. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ be two vectors of some observations sorted in ascending order. Then we let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ and $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$ be the corresponding ranks. The smallest rank is rank 1 and is given to the smallest values in $\mathbf{x}$ and $\mathbf{y}$. If there are more than one smallest value, say m duplicate values, then the rank is $\frac{1}{m}\sum_{i=1}^{m} i$. The next rank is then $1 + m$ and so on. Spearman's rank correlation is then defined as

$$\rho = \frac{\sum_{i=1}^{n}(X_i - \widehat{\mu}_X)(Y_i - \widehat{\mu}_Y)}{\sqrt{\sum_{i=1}^{n}(X_i - \widehat{\mu}_X)^2(Y_i - \widehat{\mu}_Y)^2}} \tag{B.1}$$

where $\widehat{\mu}_X = \frac{1}{n}\sum_{i=1}^{n} X_i$ is the sample mean of the vector $\mathbf{X}$.

Bibliography

[1] Mesomeris, S., Wang, Y., Salvini, M., Avettand-Fenoel, J-R., *A New Asset Allocation Paradigm.* Deutsche Bank AG, London, 2012.

[2] Mesomeris, S., Beceren, M., Salvini, M., Alvarez, M., Natividade, C., *A Macroeconomic Switchboard for Risk Factor Allocation.* Deutsche Bank AG, London, 2013.

[3] Chow, G., Jacquier, E., Kritzman, M., Lowry, K., *Optimal Portfolios in Good Times and Bad.* Financial Analysts Journal, Charlottesville, VA, 1999.

[4] Kritzman, M., Page, S., Turkington, D., *Regime Shifts: Implications for Dynamic Strategies.* Financial Analysts Journal, Charlottesville, VA, 2012.

[5] Kritzman, M., Li, Y., *Skulls, Financial Turbulence, and Risk Management.* Financial Analysts Journal, Charlottesville, VA, 2010. American Statistical Association, Alexandria, VA, 1994.

[6] Masters, T., *Monte-Carlo Evaluation of Trading Systems* Timothy Masters, 2006.

[7] Hult, H., Lindskog, F., Hammarlid, O., Rehn, C., *Risk and Portfolio Analysis.* Springer Science + Business Media, New York, 2012.

[8] Englund, G, *Datorintensiva metoder i matematisk statistik* KTH, unknown date. (http://www.math.kth.se/matstat/gru/5b1555/datortot.pdf)

[9] Clifton D. Sutton, *Classification and Regression Trees, Bagging, and Boosting* Elsevier B.V, Handbook of Statistics, Vol. 24, ISSN: 0169-7161

Appendix C

Matlab Code

Regime Switching

I use the code written by the authors in [4] for the Baum-Welch algorithm.

```
    function [A,mu,sigma,p,smoothed] = fit_hmm(y)
T=length(y);
% Simple initial guesses for parameters - can be changed
mu=[mean(y),mean(y)]+randn(1,2)*std(y);
% mu=[4,4];
sigma=[std(y),std(y)];
A=[.8,.2;.2,.8];
p=.5;
iteration=2;
likelihood(1)=-999;
change_likelihood(1)=Inf;
tolerance=0.000001;
stop=false;
% while and(change_likelihood(iteration-1) > tolerance,stop==false)
for i=1:1
    for t=1:T % 0. probability of observing data, based on gaussian PDF
        B(t,1)=exp(-.5*((y(t)-mu(1))/sigma(1)).^2)/(sqrt(2*pi)*sigma(1));

        B(t,2)=exp(-.5*((y(t)-mu(2))/sigma(2)).^2)/(sqrt(2*pi)*sigma(2));


    end
    forward(1,:)=p.*B(1,:);
    scale(1,:)=sum(forward(1,:));
    forward(1,:)=forward(1,:)/sum(forward(1,:));
    for t=2:T % 1. probability of regimes given past data
        forward(t,:)=(forward(t-1,:)*A).*B(t,:);
        scale(t,:)=sum(forward(t,:));
        forward(t,:)=forward(t,:)/sum(forward(t,:));
    end
        backward(T,:)=B(T,:);
        backward(T,:)=backward(T,:)/sum(backward(T,:));
    for t=T-1:-1:1 % 2. probability of regime given future data
        backward(t,:)=(A*backward(t+1,:)')'.*B(t+1,:);
        backward(t,:)=backward(t,:)/sum(backward(t,:));
    end
    for t=1:T % 3-4. probability of regimes given all data
        smoothed(t,:)=forward(t,:).*backward(t,:);

        smoothed(t,:)=smoothed(t,:)/sum(smoothed(t,:));
    end
    for t=1:T-1 % 5. probability of each transition having occurred
        xi(:,:,t)=(A.*(forward(t,:)'*(backward(t+1,:).*B(t+1,:))));
        xi(:,:,t)=xi(:,:,t)/sum(sum(xi(:,:,t)));
    end
    p=smoothed(1,:);
    exp_num_transitions=sum(xi,3);
    A(1,:)=exp_num_transitions(1,:)/sum(sum(xi(1,:,:),2),3);
    A(2,:)=exp_num_transitions(2,:)/sum(sum(xi(2,:,:),2),3);
    mu(1)=(smoothed(:,1)'*y)'/sum(smoothed(:,1));
    mu(2)=(smoothed(:,2)'*y)'/sum(smoothed(:,2));

    sigma(1)=sqrt(sum(smoothed(:,1).*(y-mu(1)).^2)/sum(smoothed(:,1)));
    sigma(2)=sqrt(sum(smoothed(:,2).*(y-mu(2)).^2)/sum(smoothed(:,2)));
    likelihood(iteration+1)=sum(sum(log(scale)));
    change_likelihood(iteration)=abs(likelihood(iteration+1)-...
     likelihood(iteration));
    iteration=iteration+1;
end
end
```

The Baum-Welch algorithm is used to estimate turbulence probabilities and splitting the data in the following method.

```
function [sigmaTurb,sigmaNorm,p]=mswitch(volIndex,returnMatrix)
% Inputs: Volatility Index (volindex), and a matrix of return data
% (returnMatrix).
% Outputs: Correlation matrix corresponding to the turbulence regime
% (sigmaTurb), correlation matrix corresponding to the normal regime
% (sigmaNorm) and probability of turbulence, p.


v=volIndex;
y=returnMatrix;
%Find 40th quantile.
```

```
x=quantile(v,.4);

% Perform the Baum-Welch algorithm on the volatility index.
[A,mu,~,~,~]=fit_hmm(v);

% Determine the indices of the transition matrix A. The highest expected
% turbulence corresponds to turbulent regime.
index=[find(mu==max(mu)), find(mu==min(mu))];

% Determine if we are now in turbulent or normal regime.
curTurbState=v(end);

yNorm=y(v<x,:);
yTurb=y(v>=x,:);

if curTurbState>=x
    pTurb=A(index(1),index(1));
    pNorm=A(index(1),index(2));
else
    pTurb=A(index(2),index(1));
    pNorm=A(index(2),index(2));
end
% Outputs
sigmaTurb=cov(yTurb);
sigmaNorm=cov(yNorm);
p=[pTurb pNorm];
```

The tree class uses a modified version of a matlab method called classregtree, which I have named fitMyTree. My tree class is called myTree.

```
classdef myTree
%Inputs: X is a predictor matrix, y is corresponding returns, rnd
% is a boolean value that is 1 for random tree and 0 else.
% Minleaf is the number of data points per node minimum.
    properties
        endnodes
        names
        Tree
        Ret
        Predictors
        isTree
        ypred
    end
    methods
        function Obj=myTree(X,y,minleaf,rnd,varargin)
```

```
tree=fitMyTree(X,y,rnd,'minleaf',minleaf);
Obj.Tree=tree;
Obj.Ret=y;
Obj.names=varargin;
Obj.Predictors=X;
r=y;
t=tree;
indx=find(tree.var==0);

for i=1:length(indx)
    path={};
    j=indx(i);
    firstInst=1;
    sh=mean(r(t.assignednode{j}))/std(r(t.assignednode{j}));
    counter=1;
    prevnode=j;
    while ~t.parent(j)==0
        if firstInst
            path{counter,1}=sprintf('%.2f',sh);
            firstInst=0;
            counter=counter+1;
            j=t.parent(j);
        else
            hl=find(t.children(j,:)==prevnode);
            prevnode=j;
            if hl==1
                dir='<';
            elseif hl==2
                dir='>=';
            end
            path{counter,1}=sprintf('x%i: %s %.3f',t.var(j),...
            dir,t.cut{j});
            counter=counter+1;
            j=tree.parent(j);
        end
    end
    hl=find(tree.children(1,:)==prevnode);
    if hl==1
        dir='<';
    elseif hl==2
        dir='>=';
    end
    path{counter,1}=sprintf('x%i: %s %.2f',t.var(1),dir,t.cut{1});
    path=flipud(path);
    path{end+1}=indx(i);
    Obj.endnodes{i}=path;
```

```
        end
        ypred=zeros(numel(Obj.endnodes),2);

        for i=1:numel(Obj.endnodes)
            node=Obj.endnodes{i}{end};
            sharpe=Obj.endnodes{i}{end-1};
            ypred(i,1)=node;
            ypred(i,2)=str2num(sharpe);
        end
        Obj.ypred=ypred;
    end
end
methods
    function viewMyTree(Obj,mode)
        t=Obj.Tree;
        r=Obj.Ret;
        if nargin==1
            mode=’plot’;
        end
        if strcmp(mode,’text’)
        for i=1:length(t.parent)
            if ~t.var(i)==0
                p1=sprintf(’If x%i < %.2f’,round(t.var(i)),t.cut{i});
                p2=sprintf(’ Then node %i or ’,round(t.children(i,1)));
                p3=sprintf(’ If x%i >= %.2f’,round(t.var(i)),t.cut{i});
                p4=sprintf(’ Then node %i’,round(t.children(i,2)));
                disp(strcat(p1,p2,p3,p4))
            else
                sharpe=mean(r(t.assignednode{i}))/...
                std(r(t.assignednode{i}));
                p5=sprintf(’Node %i has sharpe value %.2f’,...
                round(i),sharpe);
                p6=sprintf(’ Based on %i value points’,...
                t.nodesize(i));
                disp(strcat(p5,p6))
            end
        end
        elseif strcmp(mode,’plot’)
        for i=1:length(t.var)
            nodes(i)=t.parent(i);
        end
        figure(’position’,[200 200 1200 800])
        treeplot(nodes)
        count = size(nodes,2);
        [x,y] = treelayout(nodes);
        x = x’;
```

```
            y = y';
            for i=1:count
                if ~t.var(i)==0
                    varname=Obj.names{1}(t.var(i));
                    name1{i}=sprintf('%.2f > %s >= %.2f',t.cut{i},...
                    varname{1},t.cut{i});
                else
                    name1{i}=sprintf('%.2f',...
                    mean(r(t.assignednode{i}))/std(r(t.assignednode{i})));
                end
            end
            text(x(:,1), y(:,1), name1,...
             'VerticalAlignment','bottom','HorizontalAlignment','right')
            axis off
            end
        end
    end
    methods
    function val=predictMyTree(Obj,X)
        i=1;
        y=Obj.Ret;
        t=Obj.Tree;
        while ~t.var(i)==0
            if X(t.var(i))<t.cut{i}
                i=t.children(i,1);
            else
                i=t.children(i,2);
            end
        end
        indx=t.assignednode{i};
        val=mean(y(indx))/std(y(indx));
    end
    end
end
```

The following method is a modified version of classregtree.

```
classdef fitMyTree
    properties
          node = zeros(0,1);
        parent = zeros(0,1);
         class = zeros(0,1);
           var = zeros(0,1);
           cut = cell(0,1);
      children = zeros(0,2);
      nodeprob = zeros(0,1);
       nodeerr = zeros(0,1);
      nodesize = zeros(0,1);
      assignednode={};
         npred = 0;
    end
    methods
        function a = fitMyTree(x,y,rnd,varargin)
               a = fitTheTree(a,x,y,rnd,varargin{:});
        end
    end
end
function Tree=fitTheTree(Tree,X,Y,rnd,varargin)
C = Y(:);
W=ones(size(X,1),1);
nvars=size(X,2);
N = size(X,1);
Wtot=size(X,1);
splitMin=varargin{2};
M = 2*ceil(N/splitMin)-1; % number of tree nodes for space reservation
nodenumber = zeros(M,1);
parent = zeros(M,1);
yfitnode = zeros(M,1);
cutvar = zeros(M,1);
cutpoint = cell(M,1);
children = zeros(M,2);
nodeprob = zeros(M,1);
resuberr = zeros(M,1);
nodesize = zeros(M,1);
nvarsplit = zeros(1,nvars);
nodenumber(1) = 1;
assignednode = cell(M,1);% list of instances assigned to this node
assignednode{1} = 1:N;
nextunusednode = 2;

% Keep processing nodes until done
tnode = 1;
while(tnode < nextunusednode)
   % Record information about this node
   noderows = assignednode{tnode};
   critsave=[];
   Nt = length(noderows);
   Cnode = C(noderows,:);
   Wnode = W(noderows);
   Wt = sum(Wnode);
   nodeprob(tnode) = Wt/Wtot;
   nodesize(tnode)   = Nt;
   cutvar(tnode)     = 0;
```

```
cutpoint{tnode}   = 0;
children(tnode,:) = 0;

% Split this node
if Nt>=2*splitMin
    nusevars=size(X,2);
    Xnode = X(noderows,:);
   bestvar = 0;
   bestcut = 0;

   % Reduce the number of predictor vars as specified by nvarstosample
   varmap = 1:nvars;

    %% Random Forest
    rf=ones(nusevars,1);
    if rnd==1
        m=ceil(sqrt(nusevars));
        rn=randperm(nusevars);
        rn=rn(1:nusevars-m);
        rf(rn)=0;
    end

   % Find the best of all possible splits
   for ivar=1:nusevars
      % Index of variable to split on
      if ~rf(ivar)==0
      jvar = varmap(ivar);

      % Get rid of missing values and sort this variable
      idxnan = isnan(Xnode(:,jvar));
      idxnotnan = find(~idxnan);
      if isempty(idxnotnan)
          continue;
      end
      [x,idxsort] = sort(Xnode(idxnotnan,jvar));
      idx = idxnotnan(idxsort);

      c = Cnode(idx,:);
      [critval,cutval,splitIt]=bestSplit(x,c,splitMin);
      % Change best split if this one is best so far
     if splitIt==1
         critsave(ivar)=critval;
     else
         critsave(ivar)=0;
     end
```

```
         if and(critval>=max(critsave),splitIt)
            bestvar = jvar;
            bestcut = cutval;
         end
         end
      end

      % Split this node using the best rule found
      if bestvar˜=0
         nvarsplit(bestvar) = nvarsplit(bestvar)+1;
         x = Xnode(:,bestvar);

         % Send observations left or right
        cutvar(tnode) = bestvar;
        leftside = x<bestcut;
        rightside = x>=bestcut;


         % Store split position, children, parent, and node number
         cutpoint{tnode} = bestcut;
         children(tnode,:) = nextunusednode + (0:1);
         nodenumber(nextunusednode+(0:1)) = nextunusednode+(0:1)’;
         parent(nextunusednode+(0:1)) = tnode;

         % Assign observations for the next node
         assignednode{nextunusednode} = noderows(leftside);
         assignednode{nextunusednode+1} = noderows(rightside);

         % Update next node index
         nextunusednode = nextunusednode+2;
      end
   else
   end
   tnode = tnode + 1;
end
topnode        = nextunusednode - 1;
Tree.node      = nodenumber(1:topnode);
Tree.parent    = parent(1:topnode);
Tree.class     = yfitnode(1:topnode);
Tree.var       = cutvar(1:topnode);
Tree.cut       = cutpoint(1:topnode);
Tree.children  = children(1:topnode,:);
Tree.nodeprob  = nodeprob(1:topnode);
Tree.nodeerr   = resuberr(1:topnode);
Tree.nodesize  = nodesize(1:topnode);
Tree.npred     = nvars;
Tree.assignednode=assignednode;
end
```

The following method finds the optimal Sharp value to split on.

```
function [critval,cutval,splitIt]=bestSplit(X,y,splitMin)
%% bestSplit
% Outputs: critval = maximum absolute weighted difference in sharp ratio.
%           cutval = corresponding predictor value that generated critval.
%          splitIT = Boolean that determines if split can be made.
%          Typically if a split can't be made such that the resulting split
%          has both sides longer than the minimum splitMin.
% Inputs:  X = sorted vector of predictors.
%          y = corresponding return vector.
%          splitmin = minimum required size of the splitted vectors.

ER=[];
un=unique(X);
% for i=minN:N-minN
if numel(un)<2 %All predictors are the same, no split is possible.

    critval=NaN;
    cutval=NaN;
    splitIt=0;
elseif numel(un)==2
% There are only two distinct predictor values, a
% split has to be made on the larger one. If the result
% vector lengths are not both greater than splitMin,
% then splitIt=0;

    indexLeft=find(X==un(1));
    indexRight=find(X==un(2));
    w1=length(indexLeft);
    w2=length(indexRight);

    if and(length(unique(y(indexLeft)))>1,length(unique(y(indexRight)))>1)

     ER(1,1)=mean(y(indexLeft))/std(y(indexLeft));
     ER(1,2)=mean(y(indexRight))/std(y(indexRight));
   % Outputs
     critval=(w1*w2)*(2/length(X))^2*(ER(1)-ER(2));
     cutval=un(2);

    else
         % If there are no more than one unique return value, then std(y)=0;
        cutval=-Inf;
    end
    if ~cutval==-Inf % If a split may be possible

    if or(length(find(X<cutval))<splitMin,length(find(X>=cutval))<splitMin)
        % If the resulting split vectors are smaller than splitMin

   % Outputs
    splitIt=0;
    critval=NaN;
    cutval=NaN;

    else
     splitIt=1;
    end
    else
    % Outputs
     critval=NaN;
     cutval=NaN;
     splitIt=0;

    end
else % There are more than two unique predictor values.

for i=1:length(un)
    indexLeft=find(X<un(i));
    indexRight=find(X>=un(i));
    if and(length(indexLeft)>3,length(indexRight)>3)
    % If there is only one value for the split value, then std(y)=0.
        sharpLeft=mean(y(indexLeft))/std(y(indexLeft));
        sharpRight=mean(y(indexRight))/std(y(indexRight));
        if or(sharpLeft==Inf,sharpRight==Inf)
        % If for any split value the resulting return vector
        % contains only one unique value then std(y)=0. This can't be used.
            sharpLeft=0;
            sharpRight=0;
        end
        ERDiff(i,1)=abs(sharpLeft-sharpRight);
        W(i,1)=(length(indexLeft)*length(indexRight))*(2/length(X))^2;
        if ERDiff(i,1)==Inf
            ERDiff(i,1)=0;
        end
    else
        ERDiff(i,1)=0;
        W(i,1)=0;
    end
end
```

```
if sum(ERDiff)==0 % If all calculations failed, then no split can be made.
    % outputs
    critval=NaN;
    cutval=NaN;
    splitIt=0;
    %------
else
ERDiff=ERDiff.*W; % Weighted absolute difference in sharpe ratio.
opticut=find(ERDiff==max(ERDiff));
% outputs
critval=ERDiff(opticut);
cutval=un(opticut);
if or(length(X(find(X<cutval)))<splitMin,length(X(find(X>=cutval)))<splitMin)
   % if the resulting split is smaller in size than splitMin.

    splitIt=0;

else
    splitIt=1;
end
%-------
end
end
```

The following method crossvalidates the trees.

```
function [corval, minleaf,bestleaves,besttree]=...
crossvalMyTree(tree,kfold,minMinleaf,maxMinleaf)
% Inputs: tree is a myTree class object, kfold is how many division should
% be made to test the data on, minMinleaf is the least amount of data per
% node that is desired and maxMinLeaf is the maximum amount of data per
% node that is tested.
% Outputs: corval is a vector off rank correlations corresponding to each
% test, minleaf is the amount of data per node that is tested, bestleaves
% is the amount of data per node that was found to have the maximum
% rank correlation and besttree is the myTree object with the maximum
% rank correlation.
X=tree.Predictors;
y=tree.Ret;
names=tree.names{:};
N=length(y);
v=1:N;
minleaf=linspace(minMinleaf,maxMinleaf,10);
for leaves=1:numel(minleaf)
curLeaf=minleaf(leaves);
v=v(randperm(N));
```

```
indxCell = {};
vectLength = numel(v);
splitsize = 1/kfold*vectLength;
for i = 1:kfold
    idxs = [floor(round((i-1)*splitsize)):floor(round((i)*splitsize))-1]+1;
    indxCell{end + 1} = v(idxs)';
end
indxCell=indxCell';
corcf=[];
for i=1:numel(indxCell)
    real=[];
    sub=[];
    nonindx=indxCell;
    nonindx(i)=[];
    indx=cat(1,nonindx{:});
    newTree=myTree(X(indx,:),y(indx),curLeaf,0);
    nonindx=indxCell{i};
    for j=1:numel(nonindx)
        ysub=predictMyTree(newTree,X(nonindx(j),:));
        sub(j,1)=ysub;
        real(j,1)=y(nonindx(j));
    end
    corcf(i)=corr(real,sub,'type','spearman');
end
corval(leaves)=mean(corcf);
end
bestleaves=minleaf(find(corval==max(corval)));
besttree=myTree(X,y,bestleaves,0,names);
```

The following method is the bagging method.

```
function forest=growRandomForest(X,y,leaves,trees,names)
% Inputs: X is the matrix of predictors, y is the vector of corresponding
% returns, leaves is the amount of data per node that the trees are grown
% to and names is the names of the predictors.
% Outputs: forest is a collection of myTree objects.
nboot=trees;
N=length(y);
randomForest=1;
b=randomForest;
x=1:N;
bootindx=zeros(N,1);
for j=1:N
    bootindx(j)=x(unidrnd(N));
end
bootX=X(bootindx,:);
booty=y(bootindx,1);
```

```
randomTree=myTree(bootX,booty,leaves,b,names);
forest=myForest(randomTree);
for i=1:nboot-1
for j=1:N
bootindx(j)=x(unidrnd(N));
end
bootX=X(bootindx,:);
booty=y(bootindx,1);
randomTree=myTree(bootX,booty,leaves,b,names);
plantMyTree(forest,randomTree)
end
```

```
classdef myForest
    properties
        plantedTrees={};
    end
    methods
        function Obj=myForest(myTree)
            Obj.plantedTrees{1}=myTree;
        end
        function plantMyTree(Obj,myTree)
            forestSize=numel(Obj.plantedTrees);
            Obj.plantedTrees{forestSize+1}=myTree;
        end
        function p=predictMyForest(Obj,X)
            forestSize=numel(Obj.plantedTrees);
            val=zeros(forestSize,1);
            for i=1:forestSize
                val(i)=predictMyTree(Obj.plantedTrees{i},X);
            end
            p=mean(val);
        end
    end
end
```