Regime Switching Models for Markets

Alesis Novik

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2011

Abstract

Search for structure in the financial markets has recently become an active field of research by scientists from different areas of expertise. The Efficient-Market Hypothesis states that the markets reflect all publically available information within their price. This also means that no profit higher than the average market returns allow can be gained using only the price information. However, reasearch has shown that this is not the case.

This thesis explores the methods of searching for structure using a single asset price information. The experiments are done with multiple profit data, generated by a basket of trading rules, selected by looking at the field of technical analysis. Methods of dimensionality reduction and market classification into regimes are used. Both linear (PCA) and non-linear (LTSA) dimensionality reduction is attempted.

The results show structure within the low-dimensional representations of the high-dimensional profit data and potential for risk analysis. The structure differets between real data and artificially generated data, showing importance of certain features of the real financial data. The market classification algorithm is shown to assign different regimes to visually different data, creating the potential for profitable exploitation.

Acknowledgements

I would like to thank my supervisor Dr. Subramanian Ramamoorthy for all the input and direction he has provided while working on this project.

I would also like to thank Sergey Lisitsyn for his help with using the non-linear dimensionality reduction algorithm.

Finally, I would like to thank my parents for supporting me in my academic pursuits.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Alesis Novik*)

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Stock markets have been a fascination of researchers across many different fields for a long time. The complex mechanics, unpredictability and apparent multi-agent environment made it a target for not only scholars of finance or economics, but also computer science and artificial intelligence. Over the years many different approaches to explore the financial markets have been attempted: from simple auction traders to robust multi-agent stock markets, from agent-based simulations to exploring the features of the real market data [2, 8].

One of the more interesting features of the stock markets is the conflict with the standard economic theory. The standard economic theory describes the market forces moving the supply/demand to an equilibrium but in real markets this isn't necessarily true. Observations are also made, that price of an asset alone isn't enough to make valid forecasts or trading decisions, but recent research has shown this not to be necessarily true[16, 29]. From this, many different features arise, such as technical trading and trend following[1]. This is also the reason for the interest in the stock markets: their unpredictability makes them a perfect target for studying prediction models, data mining techniques, multi-agent interaction and many others.

Most of the research associated with the mentioned financial market features is done in portfolio optimisation, where a number of possible assets are analysed. This thesis will focus on exploring the structure of a single asset time-series. The main motivation for this is work done by Mlnařík *et al*. [17], as they show the possible existence of different regimes in the data using experimentation. These regimes can then be used either for risk management or possible exploitation and financial gain.

1.2 Contribution

This thesis shows that low-dimensional manifolds can be found when performing dimensionality reduction on trading rule profitability data. It explores what different structures are created by using different data normalisation methods and analyses the potential meaning of the new dimensions with respect to the original price time-series.

The thesis also shows that market classification, if not profit, can be achieved by using regimes defined by sets of different trading rules. It proposes further research for exploring different methods of assembling the rule baskets.

The experiments are repeated with artificial data generated by well defined models. A method for model comparison and quality analysis using low-dimensional structures of profit data is suggested.

1.3 Objectives

The main objective of this thesis is to show that there is underlying structure to be found when looking at a single asset price time-series. The more structured objectives of this thesis are as follow:

- **Generate artificial time-series data using well defined models.** Previous research has shown that certain well defined time-series models can accurately simulate asset prices [30]. The main goal for this objective is to show that the artificially created time-series are similar to real data in their behaviour, thus better understanding the relation between the well defined models and real asset prices.

- **Create a robust basket of technical trading rules.** Theses rules are used to generate profit sequences which form a high-dimensional feature space. The said profit is the main data for the experiments in this thesis.

- **Apply dimensionality reduction techniques to explore potential low - dimensional structure.** The feature space composed of profit sequences is very high - dimensional, but because the profits are generated by applying the trading rules to the price time-series, structure of the said time-series can be explored by analysing the low-dimensional manifolds. The goal is to show, that a certain structure really exists.

- **Show the effect of market classification on market models with switching regimes.** While Mlnařík *et al.* have shown that their proposed algorithm performs well if used as an online fund allocation tool, they do not explicitly state if the market *class* changes when the time-series trends change, or if it does - how? Given the algorithms good performance it is important to understand the cause: a good static distribution over strategies or actual regime detection.

1.4 Thesis Structure

To keep the thesis well formed and build on the previous chapters, the thesis is structured in the following way:

Chapter 2 provides the background of the thesis, explaining the choice of technical analysis indicators, time-series models used to generate data and dimensionality reduction techniques.

Chapter 3 explains how the data is processed, how artificial financial data is generated and how the trading rule basket is created using the technical analysis indicators.

Chapter 4 shows the methods with which the search for structure is performed.

Chapter 5 analyses the emerging structures of the financial time-series, compares and analyses the similarities and differences between them.

Chapter 6 concludes the thesis, provides final analysis of the results and outlines the future work to be done.

Chapter 2

Background

The majority of this chapter is dedicated to describe and explain the choice of technical trading rules, the behaviour of which is used in generation of profit time-series, by describing the indicators that form them. The mentioned time-series are the main data used for exploring and identifying the structure of the market. The time-series models used in artificial financial data generation are described and non-linear dimensionality reduction methods are presented as well.

The classical market models ([24]) make many assumptions about the behaviour of the agents in the market. Strong assumptions are made about the rationality of the agents and the movement of prices around the equilibrium. Although they do say profit can be gained, it can't exceed the profit generated by the average returns of the market, created by outside forces like inflation. However, real financial markets do not behave like that. This creates a need for expanding the model to include trending and technical trading.

2.1 Technical Analysis Indicators

Previous work ([29]) has shown, that using trading rules as a method of getting insight into the financial markets is a valid research direction. The research presented by Zhoue *et al*. is based on multi-asset portfolio allocation, while this thesis focuses on the single asset trading. However they do get interesting results on discovering similarities between different markets. These results encourage to try similar methods with single asset markets.

In order to create a basket of rules, which defined in the following chapter, the technical analysis indicators must first be defined. The inspiration for this distinct set of indicators that form the base of the bag of rules comes from Mlnařík *et al.* [18], Ramamoorthy and Savani [21], Creamer and Freund [10]. These indicators are also described and explained in more detail by Achelis[1]. Clayburg [7] provides simple interpretations for the technical analysis tools, which is helpful with understanding them. With some overlap, the indicators use different parameters of the financial market to generate the signals, increasing the chance of detecting structure within the market. Using indicators that only monitor one aspect of the time-series would likely produce less interesting results.

2.1.1 Exponential Moving Average (EMA)

A lot of the described rules use EMA as a part of them. The definition and method of computing the EMA is used as described by Roberts [22] for geometric moving averages. For a time-series X, it is defined as

$$EMA_1 = X_1$$

$$EMA_t = \alpha \times X_t + (1 - \alpha) \times EMA_{t-1} \text{ for } t > 1$$

For clarity, the EMA time-series will be shortened as $EMA(X,N)$, where N is the number of time periods used for smoothing. α in the previous equation is expressed as $\alpha = \frac{2}{N+1}$.

2.1.2 Relative Strength Index (RSI)

A well known indicator, used by both Mlnařík *et al.* [18] and Creamer and Freund [10], it analyses the strength of the asset. RSI belongs to the momentum oscillator class, thus moves based on the momentum and has a defined band which can be interpreted as how overbought or oversold the asset is. RSI is defined in terms of time-series U and D

$$U_t = \begin{cases} Close_t - Close_{t-1} & \text{if } Close_t > Close_{t-1} \\ 0 & \text{otherwise} \end{cases}$$

$$D_t = \begin{cases} Close_{t-1} - Close_t & \text{if } Close_{t-1} > Close_t \\ 0 & \text{otherwise} \end{cases}$$

$$RSI(N) = 100 \times \frac{EMA(U,N)}{EMA(U,N) + EMA(D,N)}$$

where N is the RSI lookback. The high RSI value generally means that the asset has been overbought, while a low one means it has been oversold.

2.1.3 Bollinger Bands

Another common technical analysis tool is the Bollinger Bands indicator [5]. It consists of a median band, an upper band and a lower band. The median band is defined as

$$Boll^m(N) = SMA(Close, N)$$

where SMA is the Simple Moving Average. The upper and lower bands are defined respectfully as

$$Boll^u(N) = Boll^m(N) + s\sigma(N)$$

$$Boll^l(N) = Boll^m(N) - s\sigma(N)$$

where $\sigma(N)$ is the standard deviation of the time-series over the lookback period N.

2.1.4 Momentum

A very basic indicator, showing the change in the price over N periods. The definition used in this thesis is

$$Momentum_t(N) = Close_t - Close_{t-N}$$

2.1.5 Acceleration

Acceleration indicator shows the immediate change of momentum and is defined as

$$Acceleration_t(N) = Momentum_t(N) - Momentum_{t-1}(N)$$

By drawing analogies to physics, acceleration could be considered the second derivative of the price data.

2.1.6 Rate Of Change (ROC)

ROC shows the rate at which the price of an asset changes. This indicator is useful due to it's sensitivity to price change accelerations and decelerations, which might indicate a change of price movement phase. The formula for it is

$$ROC_t(N) = \frac{Close_t - Close_{t-N}}{Close_{t-N}} \times 100$$

2.1.7 Moving Average Convergence Divergence (MACD)

MACD is a momentum indicator that follows trends. It does that by comparing short window (fast moving) and long window (slow moving) EMAs. Thus the indicator itself is defined as

$$MACD_t(S,F) = EMA_t(Close,S) - EMA_t(Close,F)$$

2.1.8 Williams %R

The final technical analysis indicator used is Williams %R. It is used to determine when the asset is overbought and when it is oversold. Williams %R is considered to be a general indicator. The equation used to calculate it is

$$Will_t(N) = \frac{max(High_{t-N}\dots High_t) - Close_t}{max(High_{t-N}\dots High_t) - min(Low_{t-N}\dots Low_t)} \times (-100)$$

2.2 Time-Series Models

Modelling financial time-series with well defined models has been extensively researched and results in a lot of cases have been promising [12, 13]. This section describes the models that are used in later sections to generate the artificial financial data. The main inspiration for using these models comes from research on model fitting (parameter estimation) by Würtz *et al.* [26]. Due to time limitations, only first order models are used in this thesis, and thus only these models will be described.

2.2.1 Autoregressive Moving Average (ARMA)

ARMA is a time-series model used for value estimation (for example price). ARMA consists of two parts: the autoregresive (AR) and moving average(MA). It is usually written as $ARMA(m,n)$, where m is the order of the AR part and n is the order of the MA part. Since only $ARMA(1,1)$ is used, equations for other variations of the model are not shown. $ARMA(1,1)$ can be defined as

$$x_t = \mu + a x_{t-1} + b\varepsilon_{t-1} + \varepsilon_t$$

The $ARMA(0,0)$ model would consist only of the mean (or constant) and noise (or innovation). Thinking in financial time-series terms, AR shows how the current price directly relates to the previous price. For the model to be stationary, a has to satisfy the constraint $|a| < 1$. This can be interpreted as the strength of the relation. MA shows how the previous innovations affect the current price. In the described model, depending on the value of b, it shows the strength of trending between the current and previous time steps.

2.2.2 Generalised Autoregressive Conditional Heteroskedasticity (GARCH)

In 1986, Bollerslev proposed a generalised Autoregressive Conditional Heteroskedasticity (ARCH) model [4]. It is now one of the most researched models for financial time-series modelling and has shown to produce positive results. The GARCH model, unlike ARMA is used to define the variance of time-series, in other words innovation. One of the important features of GARCH is that it has the property of excess kurtosis (heavy tails). $GARCH(1,1)$ can be expressed in the following form

$$\varepsilon_t = z_t \sigma_t$$

where z_t is a random variable drawn from a Gaussian distribution with zero mean and variance equal to 1.

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

The fact that variance of the model changes over time means that the series generated by it are heteroskedastic. Heteroskedasticity is observed in financial data, thus, when generating artificial financial data, in is important that the model would support it. Autoregressive part conditions the current innovation on the previous one. This creates an effect of volatility clustering, which is observed in real data [9], which means that after peaking, innovation tends to stay high. The model is generalised in the sense that the current variance of innovation depends on the previous variance.

Although in this thesis only Gaussian z_t is used, it can be drawn from other distributions.

2.2.3 Asymmetric Power Autoregressive Conditional Heteroskedasticity (APARCH)

Belonging to the family of GARCH models, APARCH shares most of the features with GARCH (volatility clustering, excess kurtosis). It was introduced in 1993 by Ding *et al.* [11]. The model $APARCH(1,1)$ can be defined as

$$\varepsilon_t = z_t \sigma_t$$

where z_t is a random variable drawn from a Gaussian distribution with zero mean and variance equal to 1.

$$\sigma_t^\delta = \omega + \alpha(|\varepsilon_{t-1}| - \gamma\varepsilon_{t-1})^\delta + \beta\sigma_{t-1}^\delta$$

APARCH mainly differs from GARCH in two ways. First of all it has a power transformation, which is used to linearise the model. It is also asymmetric with respect to the innovation dependency. This emerges from real stock returns, as they tend to have higher volatility after the prices drop. In other words, price drops destabilise the market, increasing volatility (or innovation).

2.3 Dimensionality Reduction

This section describes two methods of dimensionality reduction used in the thesis. The first one is a linear method called Principal Component Analysis (PCA) which reduces dimensionality by transforming the space to exploit the correlations between variables [3]. The second method is a non-linear dimensionality reduction technique called Local Tangent Space Alignment (LTSA), which attempts to find an embedding of points on a low-dimensional manifold.

2.3.1 Principal Component Analysis (PCA)

PCA is an simple eigenvector-based analysis tool, which, though used in dimensionality reduction, isn't strictly created only for that purpose. The idea behind PCA is that it uses eigenvalue decomposition of the covariance matrix of data to output the eigenvectors as new dimensions. In this thesis, the sample (unbiased) covariance of the data is used. It is also easy to find how much variance of the data the dimension explains by dividing the corresponding eigenvalue by the sum of all eigenvalues. Because the actual transformation of points to the new dimensions is not used in this thesis, it will not be explained. Although easy to use and interpret, PCA is too simplistic because the new dimensions it create must be orthogonal, thus greatly reducing the possible structures it can have.

2.3.2 Local Tangent Space Alignment (LTSA)

LTSA is a non-linear method of finding embeddings from high-dimensional data to a low-dimensional manifold [28]. Like most non-linear dimensionality reduction methods, it uses distance metrics and nearest neighbours to extract local information and construct alignments to a low-dimensional space. In this thesis the simplest Euclidean distance is used to represent the distances between the high-dimensional points.

The method uses PCA on the high-dimensional data neighbourhoods to get local tangent spaces. These are aligned to a low-dimensional manifold, producing low-dimensional coordinates of high-dimensional data. As with most non-linear dimensionality reduction methods, there is no trivial way of interpreting what the new dimensions mean, leaving this open for discussion.

Chapter 3

Data

This chapter is dedicated to explain what kind of data is used in the experiments and where it comes from. First of all the real financial data and the simplifications made are described. Then the method of generating artificial financial data is explained, along with the description of the generated data itself. When the real and artificial data is described, the method for data preprocessing and its results are shown. Finally, the basket of trading rules is presented and the method of generating profit data is shown.

3.1 Real Data

The real financial data used in this thesis is the ER futures price data. It consists of *ask* and *bit* orders, along with actual trades from 2004/11/01 to 2008/04/09. Each day contains the trading data between 8:15 and 15:15. Because the main goal of this thesis is to analyse what kind of structure can be found using only the price data, the *ask* and *bit* orders are ignored, and only the trades themselves are used. Furthermore, because none of the technical analysis indicators described in the previous section use trading volume as a parameter, it is ignored as well. In addition to this, the data is assumed to be a single continuous stream and end-of-day, weekend and holiday effect are ignored. This is done because processes that occur between the trading days depend on may outside factors not related to the time-series directly, and modelling these factors would be a separate research question. Thus, after initial preprocessing and cleaning, the real financial data consists of data points with trade date, time in seconds and price as fields.

3.2 Artificial Data

This section describes the method by which the artificial data is generated. First of all, the general methodology is explained. After that, the result of each model is shown.

3.2.1 Data Generation Method

The artificial data is generated using the models described in the previous chapter. First of all a snippet of the real time-series is taken. Because the time-series models assume the series to be discreet with respect to time, only a single price is used for every second of real data. Furthermore, if no trade (thus no price) is done at any second, the price for that time is linearly interpolated from the closest data points. In order to reduce the amount of data needed to be generated, only every tenth data point is then taken, resulting in a point every ten seconds. After this the returns for every remaining point in that snippet are calculated using

$$return_t = \frac{price_t}{price_{t-1}}$$

The time-series model is then fit to the modified return time-series using *nlminb* and NelderMead method. Ten days of real return data is used in training. The data itself has a downward trend, and the models pick it up. While most models like these are fit to log-returns, the small variation between data due to small time periods makes absolute returns a better option. As mentioned before, only the standard Gaussian distribution is used to generate errors. While it might not be the best fitting distribution ([14]), it has no parameters in need of fitting, making the model simpler. After the model parameters are estimated and the initial time steps for simulation initialisation are set up, the new artificial return time-series are generated. The process of fitting and generating is done by the *fGarch* package for R [25].

After the artificial returns are generated, they are applied on an initial price ($price_t = returns_t \times price_{t-1}$), creating the artificial price time-series. For all further experiments, this artificial data is treated the same way as real data.

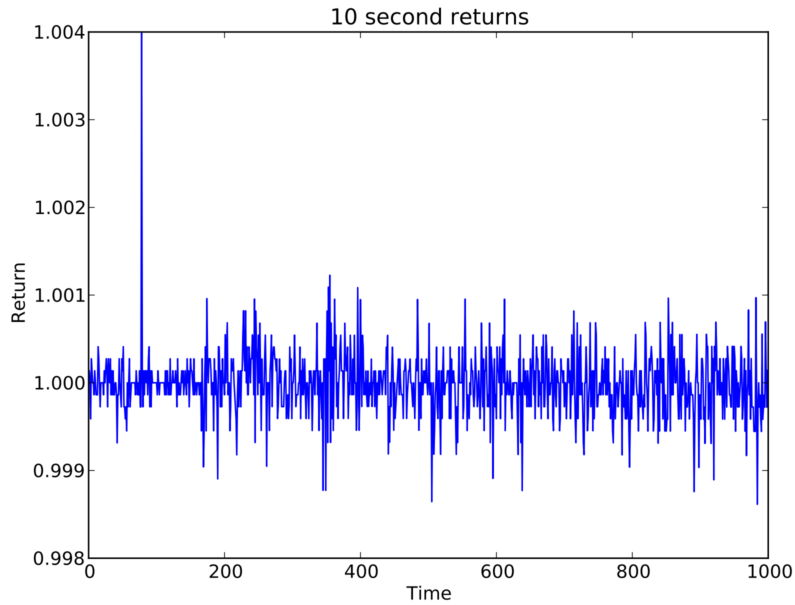

Figure 3.1: Example of real price data returns.

3.2.2 GARCH(1, 1)

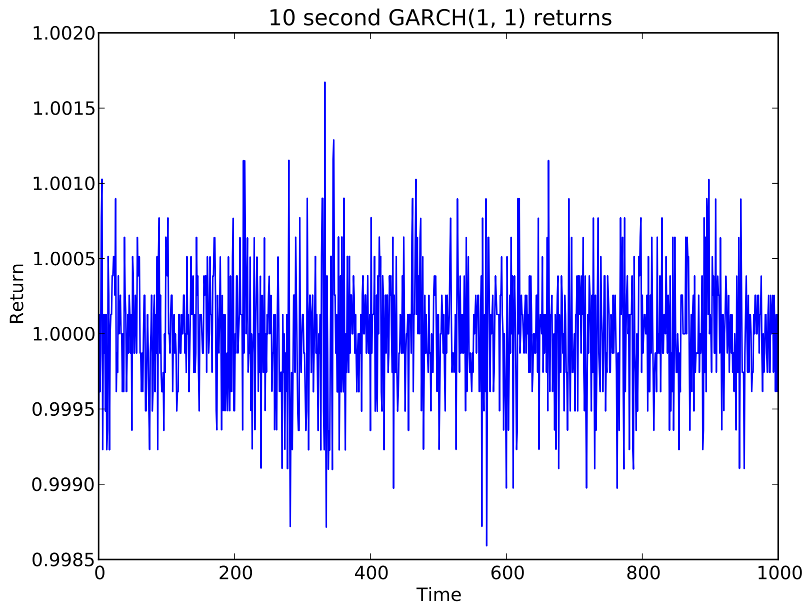

Figure 3.2: Example of $GARCH(1,1)$ generated price data returns.

$GARCH(1,1)$ generated returns exhibit the expected effect of volatility clustering and excess kurtosis of 2.519, although the sample is too small to state that the average kurtosis is as high.

3.2.3 APARCH(1, 1)

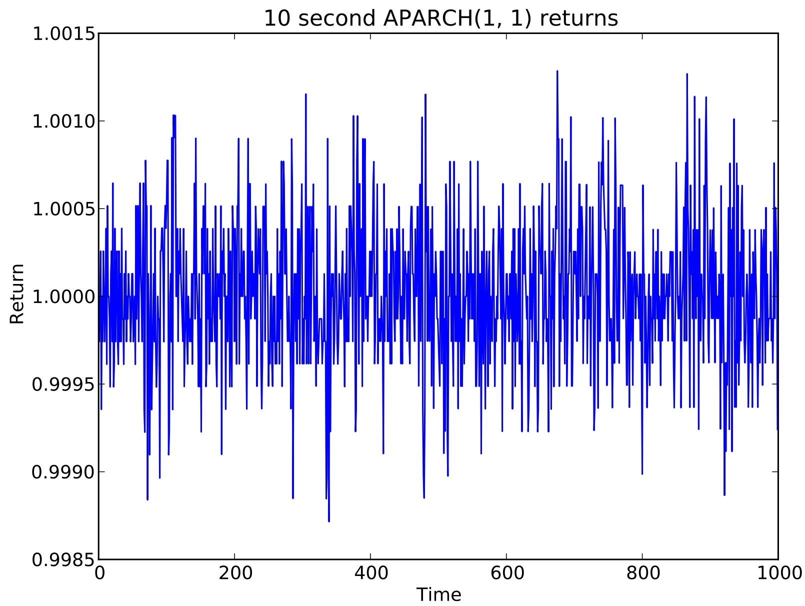

Figure 3.3: Example of $APARCH(1,1)$ generated price data returns.

Like $GARCH(1,1)$, $APARCH(1,1)$ generated returns exhibit the expected effect of volatility clustering and excess kurtosis of 0.215, a lot smaller than that of $GARCH(1,1)$. The returns also exhibit the asymmetry expected of $APARCH(1,1)$. Visually comparing it to returns of real prices it can be noticed, that both of them are asymmetric.

3.2.4 ARMA(1, 1) - GARCH(1, 1)

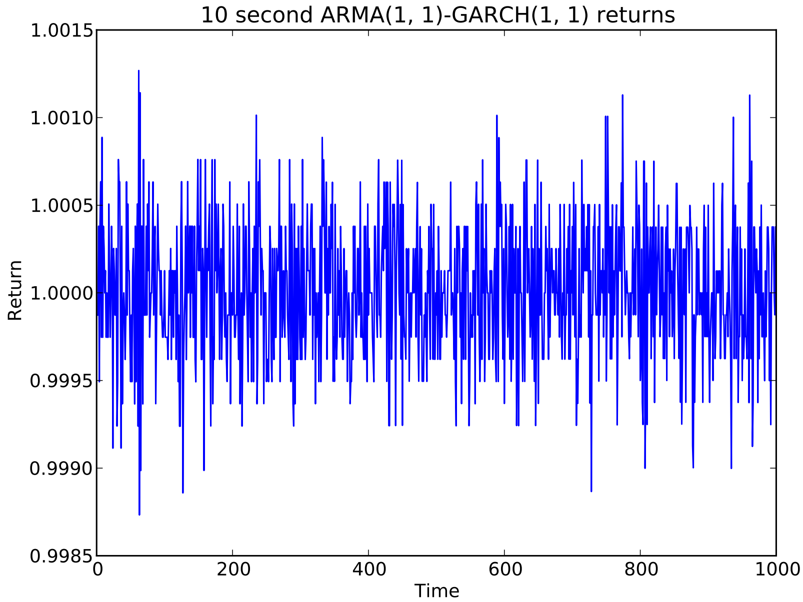

Figure 3.4: Example of $ARMA(1,1) - GARCH(1,1)$ generated price data returns.

$ARMA(1,1) - GARCH(1,1)$ generated returns show the same volatility clustering effect and excess kurtosis of 0.443. In the long run, the ARMA part of the model makes the data follow a slow downward trend, while the simple $GARCH(1,1)$ model estimates a lower mean, thus driving the reconstructed price time-series downward a lot faster.

3.2.5 ARMA(1, 1) - APARCH(1, 1)

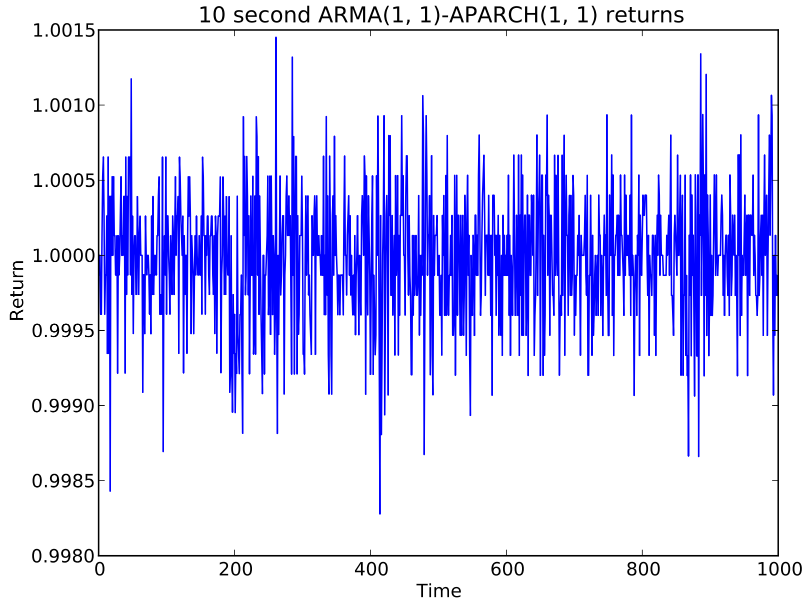

Figure 3.5: Example of $ARMA(1,1) - APARCH(1,1)$ generated price data returns.

ARMA(1, 1)-APARCH(1, 1) generated returns show the same volatility clustering effect and excess kurtosis of 0.583. Because of APARCH being asymmetric, ARMA part combined with APARCH drives the reconstructed price down faster than simple APARCH.

3.3 Data Preprocessing

Because all of the technical analysis indicators work on time ticks instead of actual price data, it needs to be preprocessed before proceeding. In the following experiments two minute time ticks are used. For every two minute period, the opening price (first trade), the closing price (last trade), highest and lowers prices are used to create a tick. As with end-of-day effects, missing time ticks (two minute periods in which no trades were made) are ignored. It should be noted, that while the plots of preprocessed price data will be simple lines, the time ticks are discreet.

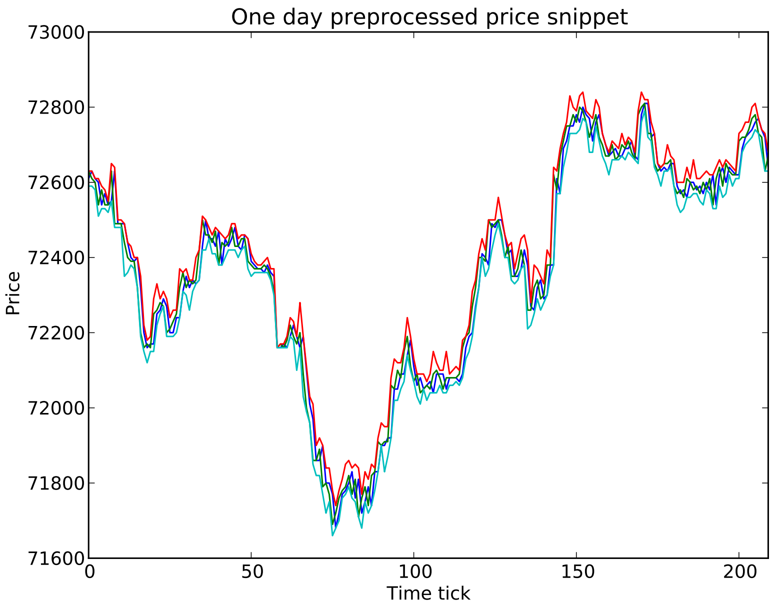

Figure 3.6: Example of preprocessed data time-series.

3.4 Basket of Trading Rules

This section uses the technical trading indicators, explained in the previous chapter, to create a basket of trading rules, which produce *buy*, *sell* or *hold* signals. It should be noted, that these rules are not random and come from research done in technical analysis (e.g. [6]).

The following rules, inspired by Mlnařík *et al.* [18] and Creamer and Freund [10], are created:

- **RSI trading rule.** RSI trading rule checks for market being overbought or oversold. When the market is overbought it is likely that the price will start decreasing in order to achieve equilibrium, thus RSI sends a sell signal. The opposite applies for buying.

$$\begin{cases} buy & \text{if } RSI_{t-1}(N) > 50 + thresh \text{ and } RSI_t(N) < 50 + thresh \\ sell & \text{if } RSI_{t-1}(N) < 50 - thresh \text{ and } RSI_t(N) > 50 - thresh \\ hold & \text{otherwise} \end{cases}$$

for $N = 200$ to 600 with 100 step and $thresh = 2$ to 6 with 1 step.

- **Inverted RSI trading rule.** Inverted RSI rule does opposite actions from RSI. Technical analysis has shown this to provide interesting results.

$$\begin{cases} buy & \text{if } RSI_{t-1}(N) < 50 - thresh \text{ and } RSI_t(N) > 50 - thresh \\ sell & \text{if } RSI_{t-1}(N) > 50 + thresh \text{ and } RSI_t(N) < 50 + thresh \\ hold & \text{otherwise} \end{cases}$$

  for $N = 200$ to 600 with 100 step and $thresh = 2$ to 6 with 1 step.

- **Sell Pullback.** Sell pullback uses a long (LL) and short (SL) lookback to generate a trading signal. These are just asset prices a certain time ago. Additionally the S parameter determines the small move percentage and B determines the long move percentage. The idea behind this rule is that if the prices have just peaked, the price is likely to decrease and if they just dipped, they are likely to increase.

$$\begin{cases} buy & \text{if } \frac{100+B}{100} \times Close_{t-LL} < Close_t < \frac{100-S}{100} \times Close_{t-SL} \\ sell & \text{if } \frac{100+S}{100} \times Close_{t-SL} < Close_t < \frac{100-B}{100} \times Close_{t-LL} \\ hold & \text{otherwise} \end{cases}$$

  for $LL = 200$ to 600 with 100 step, $SL = LL \times 0.25$ to 0.75 with 0.25 step, $B = 0.005$ to 0.01 with 0.005 step and $S = B \times 0.1$.

- **Buy Pullback.** Buy pullback is an inverted version of the Sell Pullback.

$$\begin{cases} buy & \text{if } \frac{100+S}{100} \times Close_{t-SL} < Close_t < \frac{100-B}{100} \times Close_{t-LL} \\ sell & \text{if } \frac{100+B}{100} \times Close_{t-LL} < Close_t < \frac{100-S}{100} \times Close_{t-SL} \\ hold & \text{otherwise} \end{cases}$$

  for $LL = 200$ to 600 with 100 step, $SL = LL \times 0.25$ to 0.75 with 0.25 step, $B = 0.005$ to 0.01 with 0.005 step and $S = B \times 0.1$.

- **Bollinger Band trading rule.**

$$\begin{cases} buy & \text{if } Close_{t-1} \geq Boll_t^d(N) \text{ and } Close_t < Boll_t^u(N) \\ sell & \text{if } Close_{t-1} \leq Boll_t^d(N) \text{ and } Close_t > Boll_t^u(N) \\ hold & \text{otherwise} \end{cases}$$

  for $N = 200$ to 600 with 100 step.

- **Momentum trading rule.** Momentum trading rule checks if the momentum of the asset price is increasing or decreasing significantly. If the momentum increases, that the trend follows an upward curve, and thus it is profitable to buy. The opposite applies to selling.

$$\begin{cases} buy & \text{if } Momentum_{t-1}(N) \leq EMA_t(Momentum_t(N), M) \\ & \text{and } Momentum_t(N) > EMA_t(Momentum_t(N), M) \\ sell & \text{if } Momentum_{t-1}(N) \geq EMA_t(Momentum_t(N), M) \\ & \text{and } Momentum_t(N) < EMA_t(Momentum_t(N), M) \\ hold & \text{otherwise} \end{cases}$$

  for $N = 100$ to 400 with 100 step and $M = 2$ to 3 with 1 step.

- **Acceleration trading rule.** Acceleration trading rule checks if the price change is increasing or decreasing. It can be interpreted the same way as the momentum trading rule.

$$\begin{cases} buy & \text{if } Acceleration_{t-1}(N) + 1 \leq 0 \\ & \text{and } Acceleration_t(N) + 1 > 0 \\ sell & \text{if } Acceleration_{t-1}(N) + 1 \geq 0 \\ & \text{and } Acceleration_t(N) + 1 < 0 \\ hold & \text{otherwise} \end{cases}$$

  for $N = 100$ to 400 with 100 step.

- **ROC trading rule.** Directly related to the Momentum trading rule, the ROC trading rule checks for changes in the ROC. All three momentum related rules share similar properties when generating trading decisions.

$$\begin{cases} buy & \text{if } ROC_{t-1}(N) \leq 0 \text{ and } ROC_t(N) > 0 \\ sell & \text{if } ROC_{t-1}(N) \geq 0 \text{ and } ROC_t(N) < 0 \\ hold & \text{otherwise} \end{cases}$$

  for $N = 100$ to 400 with 100 step.

- **MACD trading rule.** MACD trading rule checks for when MACD crosses over an EMA of itself, signalling a shift in the slow and fast periods. If the the fast period average drops while the slow period remains stable, it may indicate a temporary drop and a chance to buy. The opposite applies to selling.

$$\begin{cases} buy & \text{if } MACD_{t-1}(S,F) \leq EMA_t(MACD_t(S,F),N) \\ & \text{and } MACD_{t-1}(S,F) > EMA_t(MACD_t(S,F),N) \\ sell & \text{if } MACD_{t-1}(S,F) \geq EMA_t(MACD_t(S,F),N) \\ & \text{and } MACD_{t-1}(S,F) < EMA_t(MACD_t(S,F),N) \\ hold & \text{otherwise} \end{cases}$$

for $S = 200$ to 600 with 100 step, $F = 100$ and $N = 60$.

- **Williams %R trading rule.** Like the RSI rule, Williams %R trading rule attempts to detect if the asset is overbought or oversold and generate the appropriate signals.

$$\begin{cases} buy & \text{if } WILL_{t-1}(N) \geq -50 + thresh \text{ and } WILL_t(N) < -50 - thresh \\ sell & \text{if } WILL_{t-1}(N) \leq -50 + thresh \text{ and } WILL_t(N) > -50 - thresh \\ hold & \text{otherwise} \end{cases}$$

for $N = 200$ to 600 with 100 step and $thresh = 2$ to 6 with 1 step.

These 10 general rules with all of the parameters create a basket of 161 trading rules.

It should be noted, that the parameters for the trading rules are selected by experimenting. Although technical analysis does suggest certain parameters, they only apply when the time ticks are days (thus there is sufficient change between them). For example when using RSI technical analysis suggests setting N to 14 and the threshold to 20, but with two minute ticks, the RSI indicator never leaves the $(30,70)$ interval.

3.5 Calculating Profit

The main definition of profit used in this thesis is $singleprofit$ (referred to as simply profit from here on). The function calculating the profit takes price data and signals, generated by trading rules, and outputs the profit time-series. The initial conditions of the profit are

$$holding_0 = 0$$

$$funds_0 = 0$$

which means no amount of the traded asset is held and the initial funds are zeros. The time-series itself is defined as

$$profit_t = holding_t \times Close_t + funds_t$$

$holding$ and $funds$ change over time using these rules

$$funds_t = \begin{cases} funds_{t-1} - Close_t; & \text{if } signal_{t-1} = buy \\ funds_{t-1} + Close_t; & \text{if } signal_{t-1} = sell \text{ and } holding_{t-1} > 0 \\ funds_{t-1} & \text{otherwise} \end{cases}$$

$$holding_t = \begin{cases} holding_{t-1} + 1; & \text{if } signal_{t-1} = buy \\ holding_{t-1} - 1; & \text{if } signal_{t-1} = sell \text{ and } holding_{t-1} > 0 \\ holding_{t-1} & \text{otherwise} \end{cases}$$

From the profit definition is can be seen, that a one tick slippage (trade execution lag) is applied and no short selling (borrowing assets to sell) is allowed.

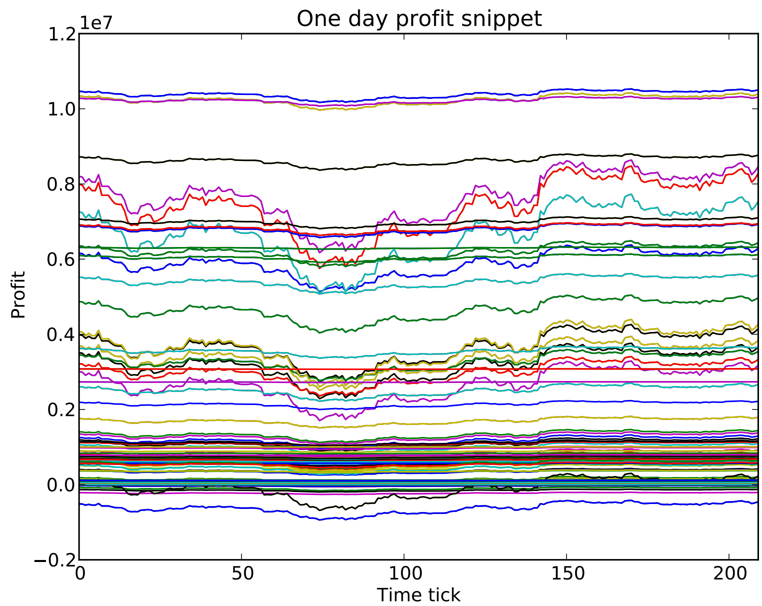

Figure 3.7: Example of trading rule basket profitability.

Another profit definition is used for the market classification algorithm. Because it uses profit as a fitness value, it is defined by how good the action is. In other words, at every time step the optimal trading decision is found and the profit is defined by how much potential profit is lost by taking a suboptimal decision.

3.6 Summary

This chapter provided all the necessary data definitions for use in the further chapters. The real data and artificial data was explained, which allows using them within the same algorithms without changing any parameters. The data preprocessing procedure was described, creating a data format to be used in profit generation. Most importantly, the basket of trading rules and profit time-series generated by it were defined. There profit time-series will be the main data for the experiments defined bellow.

Chapter 4

Structure Discovery Methods

This chapter is dedicated to describing the methods of searching for structure in the financial time-series. The two general methods used are dimensionality reduction and a market classification algorithm. First of all, the linear dimensionality reduction using PCA and it's intended effect is described. Then the non-linear dimensionality reduction using LTSA is explained, along with what potential results it might yield. Finally, the market classification algorithm is shown. All of these method use the profitability data of the basket of rules, described in the previous chapter.

4.1 Dimensionality Reduction

This section describes the dimensionality reduction techniques used when trying to find structure in the market. The techniques use basket of rules's profits as a feature-space with 161 dimension, with time ticks being observations.

4.1.1 Linear Dimensionality Reduction

For linear dimensionality reduction, PCA is used. The idea behind this method is that PCA detects the dimensions of highest variability. This method is used to analyse the rules and theirs profit, rather than try discovering structure of the market itself, though it can be argued, that through the rule analysis information about the market can be inferred. The intended result is discovering rules that have highest variability and covariation. This could potentially lead to refining the basket of rules, removing ones that do not add any additional information.

4.1.2 Non-linear Dimensionality Reduction

The main experiments of this thesis are done using non-linear dimensionality reduction. The inspiration for trying dimensionality reduction techniques to gain insight into the structure of the market came from research done on sensor space discovery [20, 19] The LTSA method is chosen for the experiments, due to interesting results it produced in other fields [27]. Two types of experiments are done using LTSA:

- **Price reconstruction.** Because the profits generated by trading rules carry within information about the real price series, it is interesting to see if reducing the dimensionality would yield a better reconstruction of the original price data. Instead of comparing the two time-series directly, a comparison is made between their returns, because they are not affected by scaling issues between the two data sets. Although it is expected, that certain rules are a better representation of the price in certain parts of the data, the reconstruction shouldn't be worse than any single rule across all of the data when averaged. Experimentation with price reconstruction is only done on real data.

- **Reduction to three dimensions.** Most of the experiments are done on reducing the dimensionality of the high-dimensional profit data to three dimensions. Three dimensions are chosen because they can be interpreted visually. Three versions of this experiment are done. First of all it is done using the original profit data. Because all of the profit time-series carry information of the price time-series, one of the dimensions should be, as expected, always highly correlated with price data. The other two, however, should not. If the points are not distributed randomly across the two out of three dimensions, it means that there are other dependencies between the data. The second experiment is done by taking the profit data and subtracting the price data from them and then doing the dimensionality reduction. Finally, the profits with price subtracted are normalised by the amount of the asset they are holding during that time step, and then dimensionality reduction is performed. Because it isn't trivial to explain what the dimensions are when using non-linear dimensionality reduction, the results are only analysed visually. The hypothesis is that if the shape of the three dimensional representation is not random and consistent over many data snippets, then there is additional, potentially exploitable structure hidden in the price data.

4.2 Market Classification

As mentionel in Chapter 2, the classical market models do not explain trending and technical trading well. This is why a definition for regimes is needed. In the case of the thesis, regimes are sets of rules with similar behaviour. The result of this is that if the rules perform well under certain market conditions, the market is in that regime.

The algorithm for market classification is described by Mlnařík *et al*. [17, 18]. In this chapter the general idea of the algorithm and the changes made are described. The main goal of this experiment is analysing if and how the algorithm assigns classes to market states. Mlnařík *et al*. have shown, that when used the algorithm for fund allocation for trading it performs satisfactory, but there was little shown about the actual classification of the market.

As mentioned in Chapter 2, this algorithm doesn't use the same profit definition the previous section used. Instead it's changed to be closer to the traditional fitness definition. Furthermore, because the regimes are supposed to be continuous periods of time, the previously defined fitness profit is added over 5 day data for a single fitness time tick.

The the market classification algorithm assigns probabilities of the current market state over 5 days being in a certain regime. As mentioned before, regimes are defined as sets of rules that behave similarly under similar market conditions. Regime distribution function assigns probabilities to rules which correspond to them being profitable under the current regime. It is defined as

$$RegimeDist(Regime_n)(rule) =$$

$$\text{probability of } rule \text{ being successful in } Regime_n$$

The market classification function assigns probabilities that they operate under a certain regime to market states. It is defined as

$$ClassifyState(MarketState_t)(Regime_n) =$$

$$\frac{1}{K} \sum_{rule \in Regime_n} \frac{RegimeDist(Regime_n)(rule) \times (fitness(MarketState_t, rule) - N_t)}{X_t - N_t}$$

where $N_t = min(fitness(MarketState_t))$ and $X_t - N_t$, with K being the normalisation constant. While the definition shows it as dependant on $RegimeDist$, it is a static distribution and does not change with $RegimeDist$ except for when explicitly stated that it is updated.

In addition to these, the weight of the rule is defined as a probability of selecting that rule at a certain state, or

$$weight(MarketState_t, rule) =$$

$$ClassifyState(MarketState_t)(Regime_n) \times RegimeDist(Regime_n)(rule)$$

Weighted fitness $wFitness$ is defined as the weight of the rule times it's fitness value. Finally, the successfulness of the market classification over random time step set T is defined as

$$succ(T) = avg_{t \in T} \frac{(\sum_{rule} wFitness(MarketState_t, rule)) - N_t}{X_t - N_t}$$

4.2.1 Finding Reasonable Regimes

After experimenting with different versions of regime selection, 25 regimes of 10 trading rules variant was chosen. The reason for this choice is that if regimes have too many rules within them, they become too general to represent the state of the market, but because of a large basket of rules, more regimes are needed to represent all of them. Because of that a constraint is placed stating that all the rules have to belong to at least one regime. Since the goal of a regime is do be different from the rest of the rules, the set of regimes can't be selected randomly.

The regime selection is done by first generating a candidate regime, checking if such a regime already exists and then performing a t-test [15] on it, the test statistic being sample standard deviation of fitnesses across rules in the candidate regime against those which aren't. If the candidate regime proves statistically different from the rest of the rules, it is added to the regime set. A smart candidate generation procedure is used, that guaranties all rules belonging to at least one regime. This is done by assigning a static value to how many rules must be sampled from those that don't belong to any regime yet.

4.2.2 Learning Classification

During the initialisation *RegimeDist* is set to be uniform over the rules in the regime and the $ClassifyState$ is updated with it, an update criteria M is set to 0.5. Learning of classification is an iterative procedure:

- Sample T uniformly for the training set.

- $succ(T)$ is computed. If it is higher than $thresh_a$ learning terminates with success.
- *RegimeDist* is updated for every regime proportionally to

$$avg_{t \in T}(fitness(MarketState_t)(rule) - \sum_{rule'} wFitness(MarketState_t, rule'))$$

- $succ(T)$ is computed with updated $RegimeDist$. If it's higher than M, $ClassifyState$ is updated with new $RegimeDist$, M is set to $succ(T)$.
- If $succ(T)$ is less than $thresh_b$ learning terminates with failure.

For implementation purposes, $thresh_a$ was set to 0.6 and $thresh_b$ was set to 0.4. Furthermore, if M wasn't updated in 10 iterations, the algorithm terminated prematurely.

4.2.3 Market Classification

The market is classified by simply applying the $ClassifyState$ function to the certain state. While this give a probability distribution, in some of the tests the assumption is made that the regime with the highest probability is the true regime for that state. Because the goal of this thesis is structure detection, actual trading using these regimes is not performed, though previous research suggests it would be effective.

4.3 Summary

This section described all the methods used for experimenting. Because the experimentation was started with no prior indication of what to expect, except a hypothesis, that the prices are driven by underlying causes, most of the work was not creating the experiments, but actually running them. Because of that, there wasn't too much detail put into defining them, because all of them, except form the market classification algorithm, are well known in other fields and were partially defined in Chapter 2. However it needs to be stated that the run-time of these algorithms was exceptionally long due to dealing with large amounts of data.

Chapter 5

Experiment Results

This chapter describes the actual experiments and their results. First of all, the linear dimensionality reduction results are shown and their interpretation is given. After that, multiple non-linear dimensionality reduction experiments are shown, describing the possible structures and similarities between different snippets of data and different data itself (real and artificial). Finally, the market classification algorithm is applied and its effects observed and analysed.

5.1 Dimensionality Reduction

This section describes the experiments done with different dimensionality reduction methods. First of all the experiments with PCA are shown, then the price reconstruction using LTSA is analysed and finally the three dimensional representations of data using LTSA are shown.

5.1.1 PCA

PCA was used in hopes of discovering which rules have the highest variance and how that variance correlates between them. Two experiments were done - with original profit data and with the normalised profit data. The results are displayed as a heat-map of absolute eigenvector values.

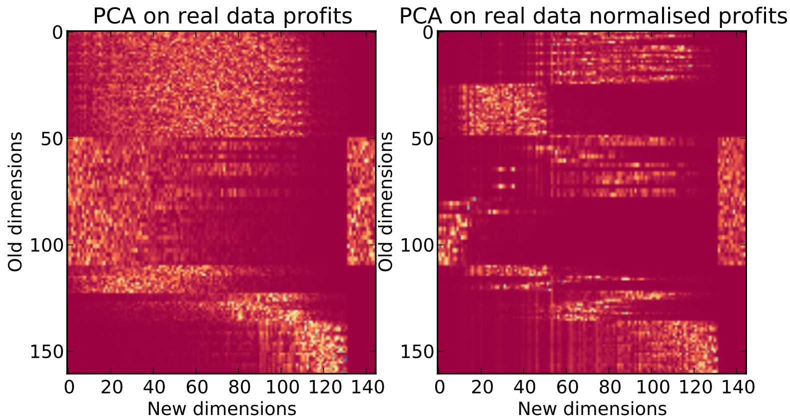

Figure 5.1: PCA results for original and normalised profits.

These results show, that rules 50 to 110 in the original data explain the largest amount of variance in the data. Furthermore, while those rule profits correlate with others, it isn't strong enough to include other rules within the first dimensions. All of these rules are Sell and Buy Pullback rules with different parameters. Because of the *buy/sell* signal generation conditions are very simple, they are true in many time ticks, thus the rules generate many *buy/sell* rules, increasing variability.

The results are even more interesting in the normalised profit case, where the dimensions seem to be assigned to rule types. To clarify, the dimensions are being assigned to profits generated by the same rules with different parameters. It seems that most of the information could be represented by a single rulefrom each category. While it's not further explored in this thesis, it would be useful to see how further experiments would act if only a single set of parameters for each rules would be used.

5.1.2 LTSA

Because implementing LTSA is outside the scope of this thesis, an external library called Shogun [23] was used. All of the methods use 5000 points of data because LTSA generates distance matrices and thus larger data sets do not fit in memory. Furthermore, for all of the experiments 300 points were used as neighbours in local space construction.

5.1.2.1 Price Reconstruction

As described in the previous chapter, price reconstruction was attempted by reducing the number of dimensions to one. The reconstruction and six best performing rule errors are reported. Two types of errors are used - the root mean squared error (RMSE) and the mean absolute percentage error (MAPE). A total of five snippets of data are experimented with and the results are reported.

	Rec	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6
RMSE 1	**3.137**	3.169	19.372	5.356	4.724	10.019	5.307
MAPE 1	**1.555**	1.692	9.643	3.083	2.782	4.006	3.258
RMSE 2	4.519	4.923	**1.958**	4.466	3.883	5.836	4.743
MAPE 2	2.676	2.882	**1.197**	2.950	2.529	3.767	2.803
RMSE 3	3.664	5.973	6.734	**2.784**	2.791	5.790	4.079
MAPE 3	2.170	3.493	3.904	1.678	**1.675**	3.565	2.479
RMSE 4	3.530	4.041	11.891	4.824	2.704	**2.126**	2.831
MAPE 4	1.648	1.867	6.260	2.527	1.447	**1.143**	1.800
RMSE 5	3.091	4.820	2.988	4.525	4.294	5.521	**2.927**
MAPE 5	1.814	2.712	1.830	2.669	2.533	3.032	**1.510**
Avg RMSE	**3.588**	4.585	8.589	4.391	3.679	5.858	3.977
Avg MAPE	**1.972**	2.529	4.567	2.581	2.193	3.103	2.370

Table 5.1: Price reconstruction accuracy results.

All the errors in the table are of e^{-5} order. The table shows the expected results that while in certain snippets a single rule can be better at reconstructing the price, because all of them carry the time-series information with them, no single rule can perform better than the reconstruction across a wide range of data. If even more data snippets would be analysed, the result would probably be even clearer.

5.1.2.2 Low-Dimensional Structure

The main goal of this section is to show that there is come kind of underlying structure in the price data. Experiments with real data are done on three separate data snippets to show the consistency of the emerging structure. For every reconstruction one point is additionally selected and it's relative profitability data is displayed.

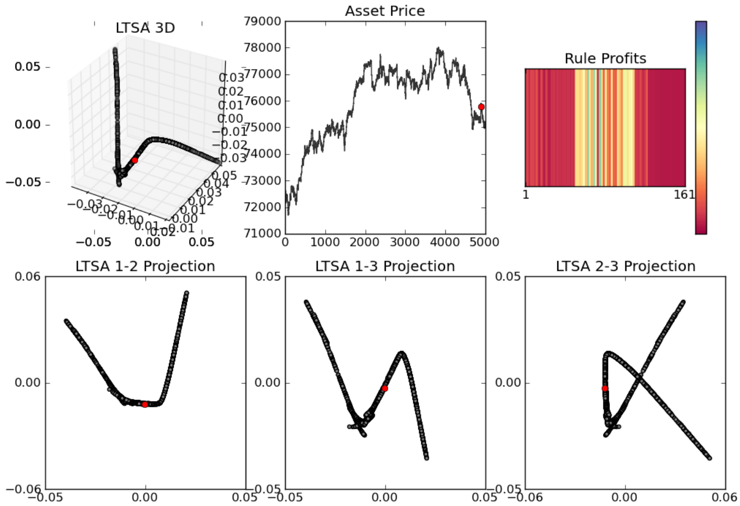

Figure 5.2: LTSA results for snippet 1.

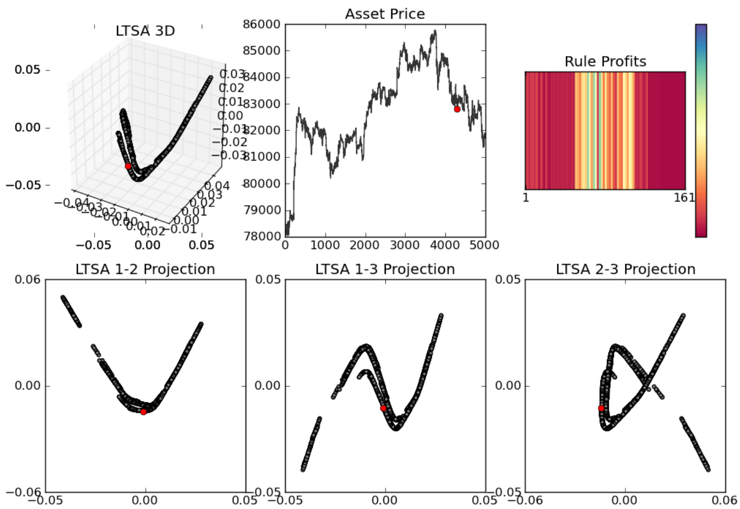

Figure 5.3: LTSA results for snippet 2.

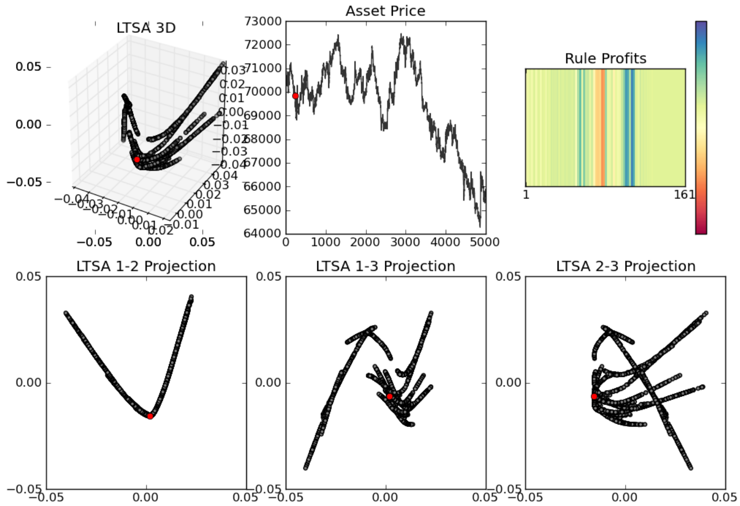

Figure 5.4: LTSA results for snippet 3.

These three figures show that there are shapes emerging in the low-dimensional space, thus the points are mapped to a certain low-dimensional manifold. Because these points are not randomly placed in space, and only the first dimension is highly correlated with the actual price data, the other two dimensions mean that there is something else driving the trading rules as well. Even more interesting is that the shape of the embedding is consistent over the three snippets of data. Although more rigorous testing would be useful, it's can be stated that some of the trading data has the same low-dimensional space, making the hypothesis of there being factors driving the profits more sound. In the last figure more noise in the embedding can be seen. It is highly likely that this noise is created by the high variability in price data in that snippet.

Further experiments are done with profits that have price data subtracted from them and the data that has been normalised. The figures show the projections and relative profitability data, making them easier to compare.

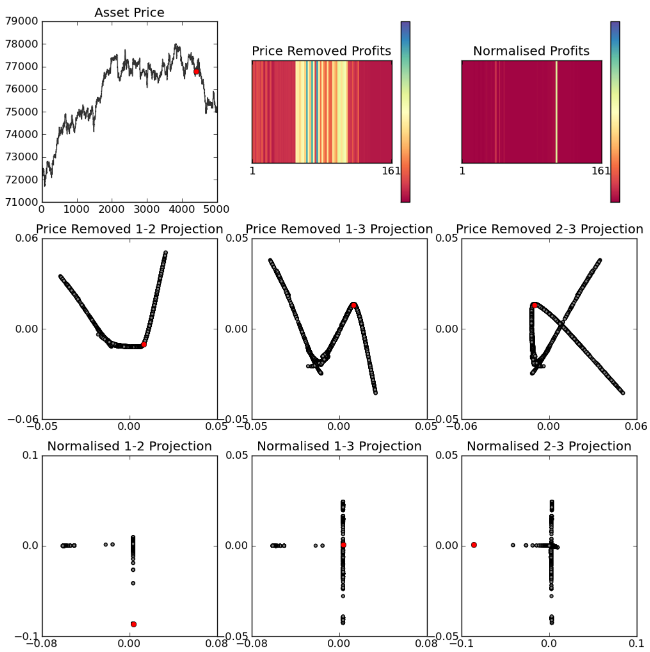

Figure 5.5: LTSA comparison for snippet 1.

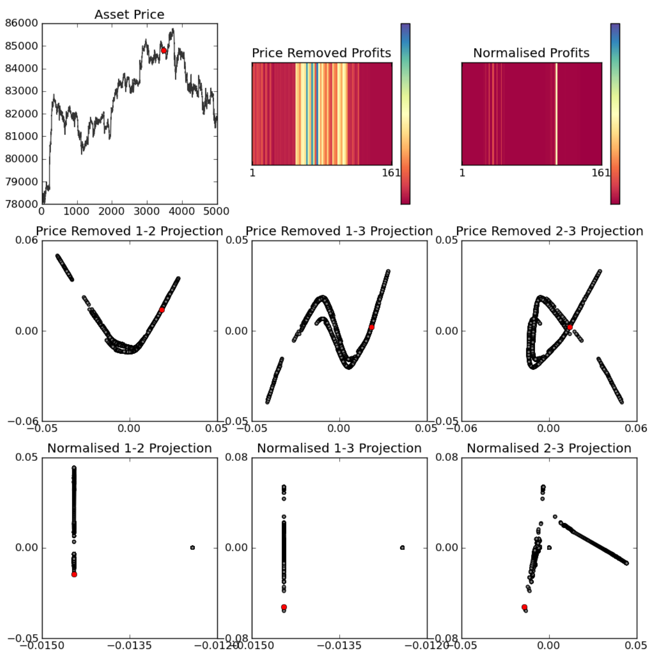

Figure 5.6: LTSA comparison for snippet 2.

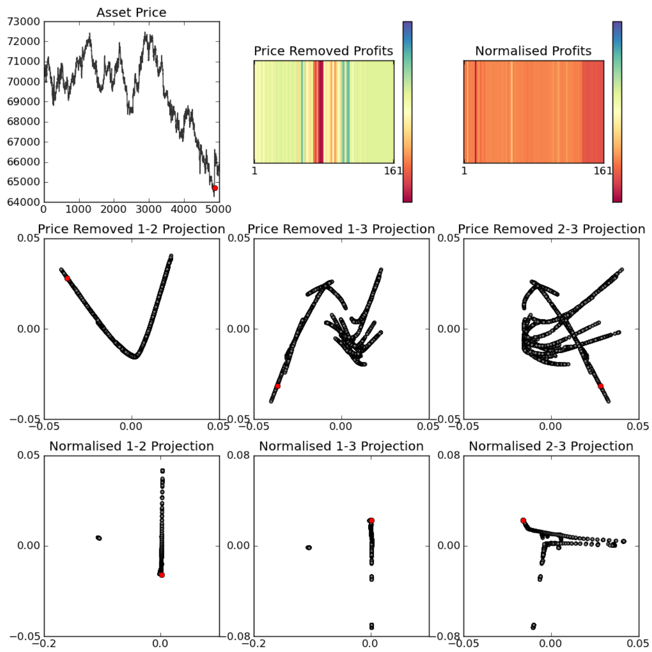

Figure 5.7: LTSA comparison for snippet 3.

The interesting result observed from these figures is that the general shape of the price removed profit embedding stays very similar to the original. It is also interesting, that the normalised profit data embedding points position themselves along the axes of the dimensions. Furthermore it seems that there is a direct relation between the noise in the non-normalised representation and the normalised representation. Another result, not directly visible in the figures, but it seems that the points with certain peeking profits are clustered next to each other and it is visible, that there are clusters of points in the normalised representation case with no points between them. These results can potentially be used in risk management, if a strong relation between the underlying dimensions and the price curve can be proved.

5.1.2.3 Generated Data

The rest of the non-linear dimensionality reduction experiments are done with artificial data. Due to the time it takes to find an embedding, only five experiments are run - one on each of the methods described before generated data and one with points from two methods combined. The experiment type is the same as the one on real data when using the price removed and normalised data.

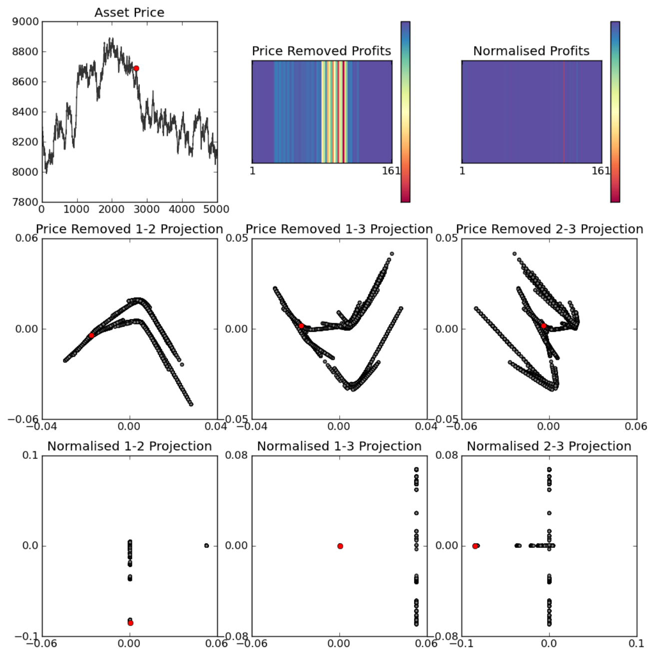

Figure 5.8: LTSA comparison for GARCH data.

This figure shows the low-dimensional representation of data generated by the $GARCH(1,1)$ model. The representation visually differs from the one created from real data. An argument can be made that either this subset of data just had different properties or that the data generated by the model does not represent real data accurately enough.

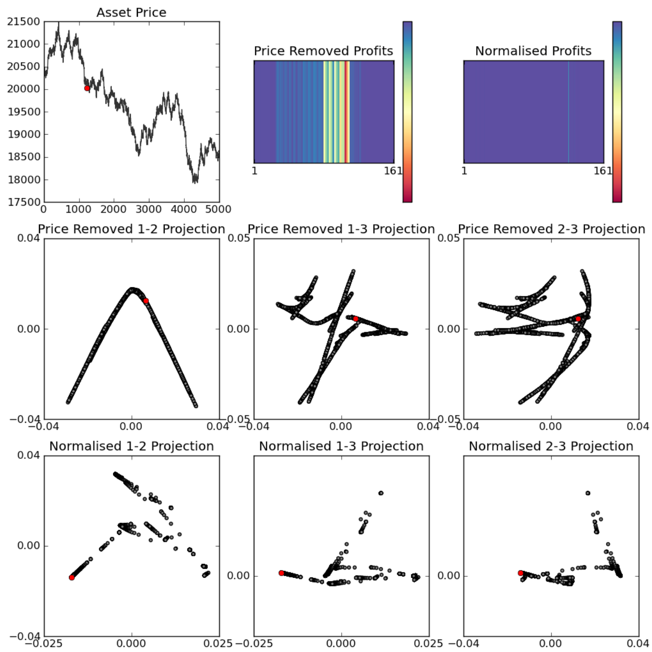

Figure 5.9: LTSA comparison for APARCH data.

This figure shows the low-dimensional representation of data generated by the $APARCH(1,1)$ model. The shape of the low-dimensional representation differs from the real data, but looks similar to the one created from $GARCH(1,1)$ generated profits. It seems that the models lacking the ARMA component do not represent the data well enough to create the same low-dimensional structure out of the high-dimensional profit data.

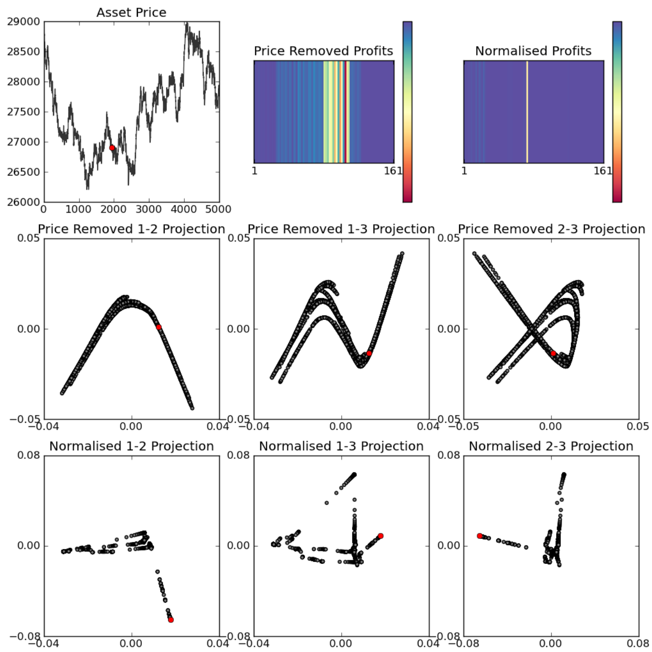

Figure 5.10: LTSA comparison for ARMA-GARCH data.

This figure shows the low-dimensional representation of data generated by the $ARMA(1,1) - GARCH(1,1)$ model. The low-dimensional embedding of the data is visually very similar to the one observed in real data. This reinforces the idea that the ARMA component is important in creating artificial data capable of simulating the effects of the real asset price time-series.

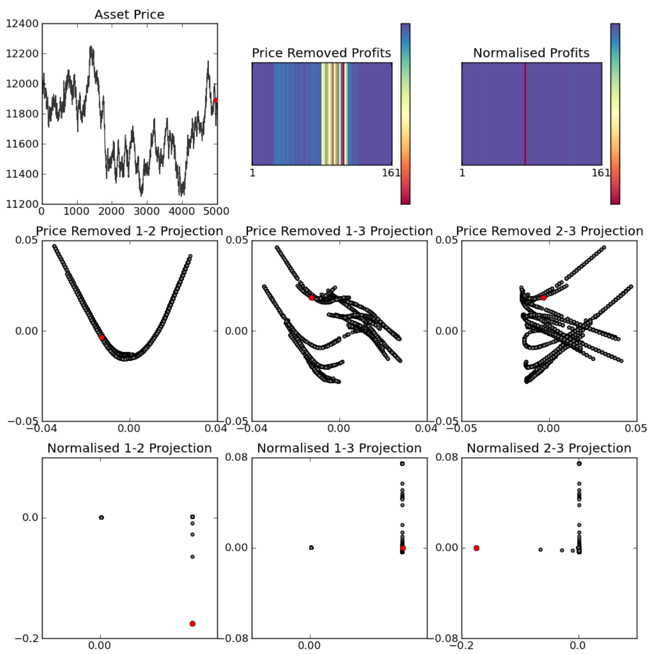

Figure 5.11: LTSA comparison for ARMA-APARCH data.

This figure shows the low-dimensional representation of data generated by the $ARMA(1,1) - APARCH(1,1)$ model. It confirms the previously stated hypothesis that ARMA is important to recreating the effect of real asset price time-series, because the shape of the low-dimensional is, while very noisy, similar to the one seen in real data.

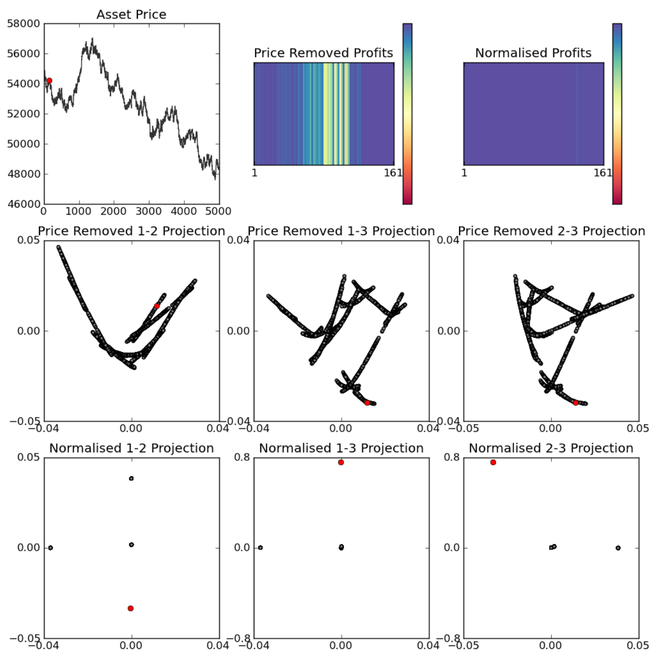

Figure 5.12: LTSA comparison for switching data.

This figure shows the low-dimensional representation of of which the first half is generated by $ARMA(1,1) - GARCH(1,1)$ and the second half is generated by the $GARCH(1,1)$ model. The shape of its representation is unlike any seen before, which is understandable, but also suggests that real data has consistent features, and does not simply turn certain elements like autoregressiveness or history.

5.1.2.4 Summary

Potentially these findings can still be disproved with extensive testing of different parameter setting or data snippets of real and generated data, but that is beyond the scope of this thesis. So while without proof, the hypothesis of there being underlying structure in the price data visible through representing the rule profits in low-dimensional space and that it might be related to autoregressiveness of returns.

These results are interesting because they give a lot of room for further experimentation. Clustering the normalised data could yield information for risk management, because it appears that small and remote clusters contain points from profit vectors, where only a few of the values are in the extremes.

5.2 Market Classification

Because previous research has shown that the algorithm is quite effective when using it for fund allocation, this section focuses on showing if different regimes in different looking data are actually selected. Because regimes are computed for five day periods, price data displaying an upward or downward trend over the five days is selected and the probability distribution over possible regimes is reported. What rules the regimes contain is not reported, because the algorithm never explicitly states them and it is not as important, because they are proven to be statistically different from the rest.

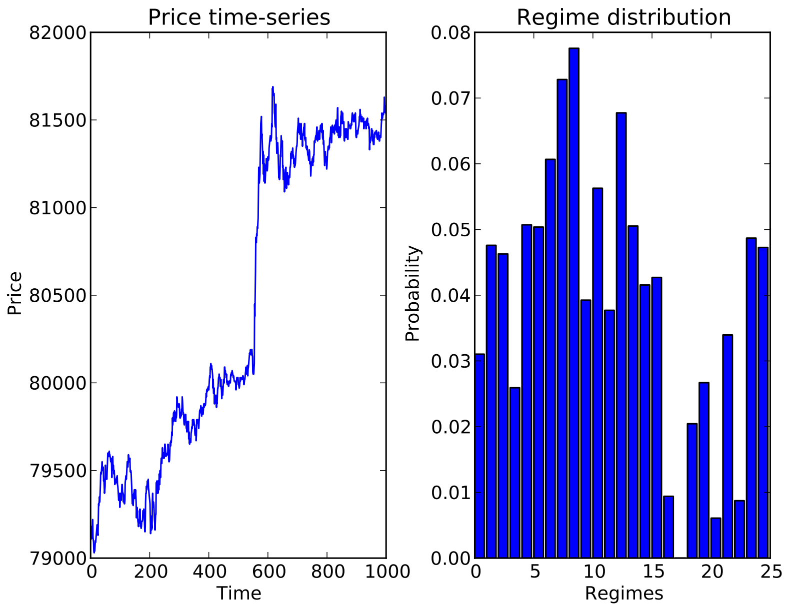

Figure 5.13: Regime probability distribution on increasing snippet 1.

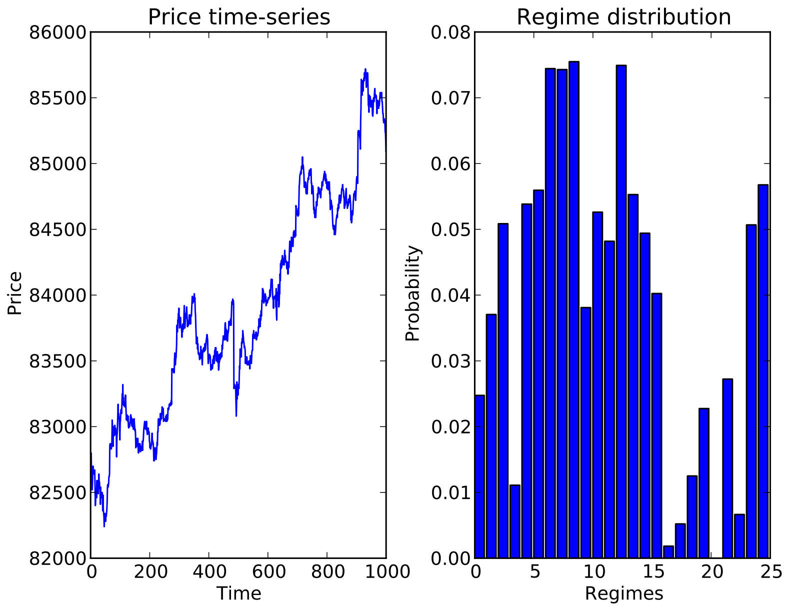

Figure 5.14: Regime probability distribution on increasing snippet 2.

Both of the figures above display the same dominating regime for upward trending price time-series. The general distribution over the regimes also looks similar.

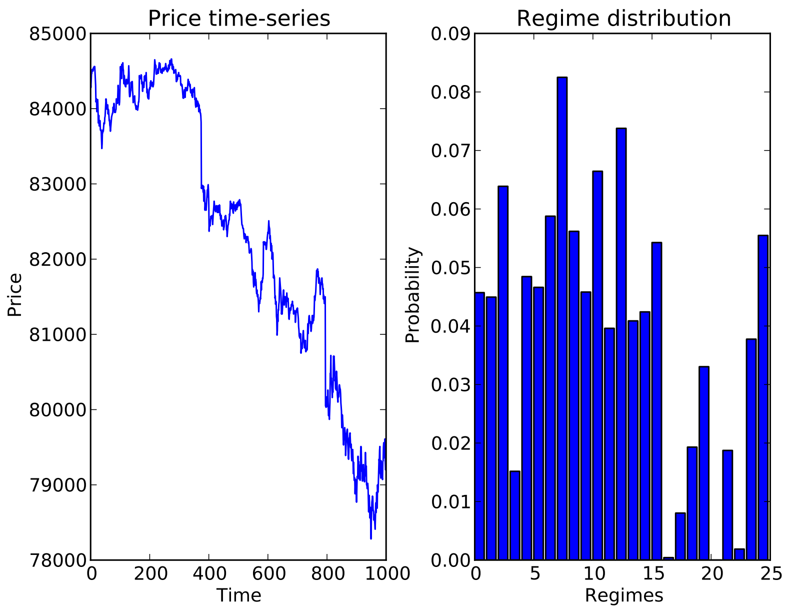

Figure 5.15: Regime probability distribution on decreasing snippet 1.

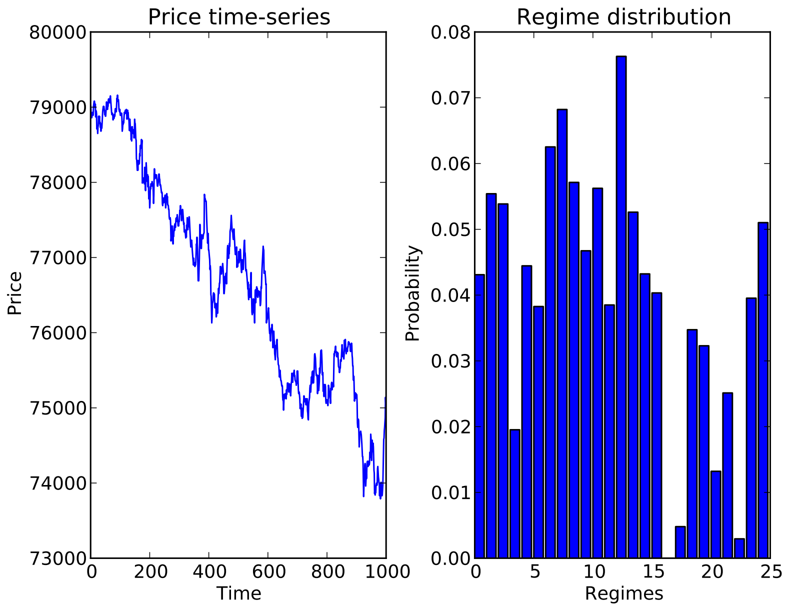

Figure 5.16: Regime probability distribution on decreasing snippet 2.

In both of the downward trending price displaying figures the dominating regimes are different, though the general distributions look similar.

The regime that worked well on the upward trending data is significantly less probable to be selected in these cases. Furthermore there are regimes, that have very low probability of being selected across all of the data. This suggests that not all of the regimes are useful. This is to be expected, because if a regime performs worse that all of the other data, it is still statistically significantly different from the performance of other regimes.

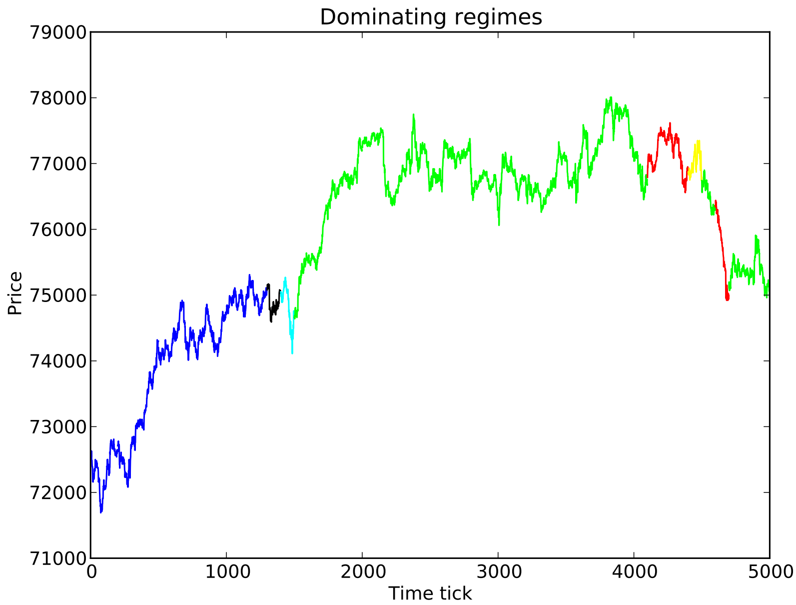

Figure 5.17: Dominating regimes over 5000 time ticks.

The figure above marks different regimes with different colours. These results reinforce the hypothesis that the market can be classified by using a set of regimes, because different regimes are obviouslly dominating in different market conditions. Although the regimes are said to be calculated for a 5 day time span, in this figure the market classification is updated every 100 time ticks.

Because the candidate regimes are selected randomly, it is difficult to predict which of them will be performing well under different conditions and if a wide range of these conditions will be covered. A potential solution would be not to select the regimes randomly, but rather select a few representative subsets of data and classify the rules into regime depending on how well they perform under those conditions.

5.2.1 Summary

These experiments show that the market classification algorithm works as intended and assigns regimes to market states depending on their effectiveness. It is also obvious that the regimes are assigned according to the trend the price data is following.

Chapter 6

Conclusions

This chapter present the final conclusions of this thesis, emphasises the important results produced along with a possible way to use them and finally suggests future work.

6.1 Summary

The goal of this thesis was to show that there was structure in the single asset market beyond the simple price data. In order to prove this hypothesis, numerous experiments using dimensionality reduction and market classification using trading rule profits were done. Real data from the ER futures market was used and additional artificial data using well defined time-series models ARMA, GARCH and APARCH was generated.

The results of the dimensionality reduction experiments with PCA and LTSA have shown that the profit data has underlying structure beyond the price data and the market classification algorithm has shown that it is possible to create regimes that accurately classify the market dependant on its trends.

All of these results support the original hypothesis and provide room for exploring the financial markets in this direction further.

6.2 Main Results

Because the main goal of this thesis was to create new hypothesis about the data by showing the existance of structure within the profit time-series, the results are given along with their possible usefulness in other areas of finance.

This thesis has shown that using methods like dimensionality reduction can reveal low-dimensional structure of profit time-series, generated by a carefully selected basket of rules. The low-dimensional representations show similarity between real data and artificially generated data which suggest which features in the models might be important to real price time-series. If these features, for example the autoregressiveness of returns, can be exploited, more accurate predictions of future prices can be made.

Normalised profit data analysis using dimensionality reduction shows a potential way of using the representation to determine potential extreme values in profits. These can be used for risk management purposes. Because risk management involves analysing potential loss, finding extreme values of profits generated by different rules can be useful.

Although not explored in this thesis, the fact that price data can be accurately reconstructed using the profit information can yield interesting results as well. If filtering techniques are applied, it can be possible to do one step predictions with some accuracy (for example using the Kalman filter).

Because this thesis has shown, that the market classification algorithm does classify the market into regimes, if trading would be associated with regimes instead of the strategies that compose them, it would be possible to use this algorithm within decision making in different ways, and not only just as a method of assigning probabilities to rules in the basket. Using the regimes new trading rules can be formes, encompassing more data than a simple moving average of data or its deviation.

6.3 Future Work

The is a lot of potential for future work to be done with the results shown in this thesis. First of all, because most of the analysis done was based on visual perception, it is might not be very accurate, and thus better methods of analysing the results should be suggested and implemented. As mentioned above, clustering of profitability data based on the location in the low-dimensional representation could yield interesting results both in terms of profitability and risk analysis, due to extreme values of profits potentially being outliers.

The market classification algorithm can potentially be improved by a careful selection of regimes, instead of random assignment. The ideas for new regimes can be drawn from results received by applying PCA to the profit data. These results have shown similar behaviour between certain rules, and assigning similar behaving rules to regimes is one of the conditions for the market classification algorithm being successful. It should also be analysed in depth if there are relations between the low-dimensional structure and the regime assignment.

Additionally, rules used in the regimes can be created by using genetic algorithms with a set of binary indicators of the market state as parameters. Similar work was done by Arthur *et al*. [2] for agent trading simulations.

Bibliography

[1] Achelis, S. B. (1995). *Technical Analysis from A to Z.* Irwin.

[2] Arthur, W. B., Holland, J. H., LeBaron, B., Palmer, R., and Taylor, P. (1996). Asset Pricing Under Endogenous Expectation in an Artificial Stock Market. Technical report, Santa Fe Institute.

[3] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing edition.

[4] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307 – 327.

[5] Bollinger, J. A. (2001). *Bollinger on Bollinger Bands*. McGraw-Hill.

[6] Brock, W., Lakonishok, J., and LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5):1731–64.

[7] Clayburg, J. F. (2001). *Four Steps to Trading Success: Using Everyday Indicators to Achieve Extraordinary Profits*. John Wiley & Sons, Inc.

[8] Cliff, D. and Bruten, J. (1997). Zero is Not Enough: On The Lower Limit of Agent Intelligence for Continuous Double Auction Markets.

[9] Cont, R. (2005). Volatility clustering in financial markets: Empirical facts and agent-based models.

[10] Creamer, G. and Freund, Y. (2010). Automated trading with boosting and expert weighting. *Quantitative Finance*, 10(4):401–420.

[11] Ding, Z., Granger, C. W., and Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, 1(1):83 – 106.

[12] Engle, R. (2001). Garch 101: The use of arch/garch models in applied econometrics. *Journal of Economic Perspectives*, 15(4):157–168.

[13] Feng, Y., Beran, J., and Yu, K. (2007). Modelling financial time series with semifar-garch model. CoFE Discussion Paper 07-14, Center of Finance and Econometrics, University of Konstanz.

[14] Ferenstein, E. and Gasowski, M. (2004). Modelling stock returns with ar-garch processes. *SORT 2004 Vol 28 Nm 1 JanuaryJune*, 28(October 2003):55–68.

[15] Good, P. I. (2004). *Permutation, Parametric, and Bootstrap Tests of Hypotheses (Springer Series in Statistics)*. Springer, 3rd edition.

[16] Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *The Journal of Economic Perspectives*, 17(1):pp. 59–82.

[17] Mlnařík, H., Ramamoorthy, S., and Savani, R. (2009). Multi-Strategy Trading Utilizing Market Regimes.

[18] Mlnařík, H., Ramamoorthy, S., and Savani, R. (2010). Multi-strategy Asset Allocation Utilizing Market Regimes.

[19] Modayil, J. (2010). Discovering Sensor Space: Constructing Spatial Embeddings That Explain Sensor Correlations. In *International Conference on Development and Learning*.

[20] Pierce, D. and Kuipers, B. J. (1997). Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92(1-2):169 – 227.

[21] Ramamoorthy, S. and Savani, R. (2008). A Basket of Trading Strategies.

[22] Roberts, S. W. (1959). Control Chart Tests Based on Geometric Moving Averages. *Technometrics*, 1(3):pp. 239–250.

[23] Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., Bona, F. D., Binder, A., Gehl, C., and Franc, V. (2010). The shogun machine learning toolbox. *Journal of Machine Learning Research*, 11:1799–1802.

[24] Stapleton, R. C. and Subrahmanyam, M. G. (1983). The market model and capital asset pricing theory: A note. *Journal of Finance*, 38(5):1637–42.

[25] Wuertz, D. and Rmetrics Core Team (2009). Package 'fgarch'. `cran.r-project.org/web/packages/fGarch/fGarch.pdf`.

[26] Würtz D., C. Y. and L., L. (2009). Parameter estimation of arma models with garch/aparch errors: An r and splus software implementation. *Journal of Statistical Software (forthcoming)*.

[27] Zhang, T., Yang, J., Zhao, D., and Ge, X. (2007). Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7-9):1547 – 1553. Advances in Computational Intelligence and Learning - 14th European Symposium on Artificial Neural Networks 2006, 14th European Symposium on Artificial Neural Networks 2006.

[28] Zhang, Z.-y. and Zha, H.-y. (2004). Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *Journal of Shanghai University (English Edition)*, 8:406–424. 10.1007/s11741-004-0051-1.

[29] Zhou, W.-X., Mu, G.-H., Chen, W., and Sornette, D. (2011). Strategies used as spectroscopy of financial markets reveal new stylized facts. *Arxiv preprint arXiv11043616*.

[30] Zivot, E. and Wang, J. (2005). *Modelling Financial Time Series with S-PLUS , Second Edition*, volume 49. Springer Verlag, New York etc.