UPPSALA
UNIVERSITET

UPTEC F15 039
Examensarbete 30 hp
Juni 2015

Classification of high-frequency FX market data

Master Thesis

David Lundberg

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Classification of high-frequency FX market data

David Lundberg

The goal of this master thesis was to develop a method for real-time classification of market trading data at the Foreign Exchange (FX) department at the Skandinaviska Enskilda Bank (SEB). The characteristics in the market data sets were analyzed using Principal Component Analysis (PCA). The analysis showed that the principal component subspaces for two different types of market data, normal and abnormal, for the EUR/USD instrument where significantly different.

The result from the PCA naturally led into the construction of a Single-class detector, for detecting if quote updates were normal or abnormal based on training data. The market data sets were shown to possess multicollinear characteristics, resulting in low-rank properties of the covariance matrices. To overcome this problem the solution was to transform the data using PCA, resulting in full-rank properties of the covariance matrices of the transformed data. This vital step made it possible to classify quote updates for the EUR/USD instrument.

The project resulted in a classification algorithm which is able to successfully classify if a quote update is normal or abnormal with respect to training data in real-time. The algorithm is versatile in the sense that it can be implemented on any market for any currency pair, and can easily be extended to classify the relative behaviour between several currency pairs in real-time.

Handledare: Dr. Pär Hellström
Ämnesgranskare: Dave Zachariah
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTEC F15 039

Populärvetenskaplig sammanfattning

Målet med detta examensarbete var att utveckla en metod för att klassificera marknadsdata från Skandinaviska Enskilda Bankens (SEB:s) avdelning för valutahandel. Syftet med projektet var att utveckla en klassificeringsalgoritm för att i realtid kunna klassificera marknadsdata. För att åstadkomma detta krävdes analys av karaktäristiken hos olika typer av marknadsdata. Två olika typer av marknadsdata analyserades: data som ansågs vara normal, och data som ansågs vara onormal. Analysen resulterade i att de två olika typer av marknadsdata kunde särskiljas genom deras kovariansmatriser.

För att klassificera marknadsdata som normal eller onormal konstruerades en klassificerare baserat på marknadsdatans kovariansmatriser. Analys av marknadsdata visade på att pris-variablerna besatt multikolinjära samband, vilket i sin tur ledde till att kovariansmatriserna erhöll låg rang. Lösningen till detta problem bestod av att transformera marknadsdatan till ett delrum definierat av de så kallade *principal-komponenterna* för marknadsdatan.

Projektet resulterade i en klassificerings-algoritm som i realtid kan klassificera ifall en uppdatering av marknadsdata för EUR/USD är normal eller onormal baserat på träningsdata. Algoritmen är mångsidig i den meningen att den kan implementeras för alla typer av marknadsdata och valutapar.

Contents

Acknowledgements

In full gratitude I would like to acknowledge the following individuals who encouraged, inspired, supported and assisted me in order to make this master thesis possible.

First and foremost, I would like to thank Dave Zachariah at Uppsala University for his guidance and support during this work. His guidance and teaching in the fields of signal processing, machine learning and linear algebra has been invaluable in this work. I would also like to thank him for always being available and open for questions.

I would also like to thank Dr. Pär Hellström at the Skandinaviska Enskilda Bank (SEB) for the inspiration and motivation during the work, as well as for providing guidance in the field of high-frequency trading and for providing me with market data.

Finally, I want to thank my beloved family for their motivation, encouragement and financial support during my education. Without them, none of this would have been possible.

Notational Conventions

$\mathbb{R}$	the set of real numbers
$(\cdot)^{\mathrm{T}}$	transpose of a vector or matrix
E[x]	the expected value of x
$\|x\|$	the Euclidean norm of a vector x
p	number of variables
N	number of observations
$\mathbf{x}$	p-dimensional random variable (column vector)
$\boldsymbol{\mu}$	$[p \times 1]$ mean vector
$\boldsymbol{\Sigma}$	$[p \times p]$ covariance matrix
$\mathbf{X}$	$[p \times N]$ data matrix
$\mathbf{V}$	$[p \times p]$ principal component loading matrix
$\hat{\mathbf{V}}$	$[p \times q]$ truncated principal component loading matrix
$\mathrm{var}(x)$	the variance of x
$\mathrm{cov}(x, y)$	the covariance between x and y
$p(x)$	probability density function
$\hat{G}(\mathbf{x})$	classification rule
$\delta_k(\mathbf{x})$	discriminant function
$\arg\max_x f(x)$	the value of x that maximizes $f(x)$
$\mathcal{I}$	a specific financial instrument
$\mathcal{M}$	a specific market maker

1 Introduction

Today at the Foreign Exchange (FX) department at the Skandinaviska Enskilda Bank (SEB), exchange rate data are received in real-time from different market makers at a high rate. The market data consists of buy and sell prices for different volumes, all published by the market makers. Due to the big competition in the business the margins are small and trades made in large volumes are necessary to make sufficient profits. Since the risk of a trade is proportional to the amount of the volume, the risk becomes an important factor in high frequency trading. Public announcements such as government statements and interest rates decisions from e.g. the European Central Bank [1] tend to have large impact on the market, affecting the buy and sell prices for specific exchange rates. Due to such events, along with the demand and supply of different currencies, the market data tend to be highly dynamic over time, and abnormal behaviours tend to escalate quickly ones started. Therefore classification algorithms which detects abnormal behaviour of the market data are desirable. The output of such a classification or detection algorithm could serve as an input when determining, for example, positions in a trade, or it could simply act as an alarm when the market starts behaving unnatural or odd.

The goal of the thesis is to analyze how valuable information from the high frequency exchange rate data can be extracted in order to classify the state of an exchange rate. Based on the extracted information, we aim to construct a suitable classification method for detecting normality and abnormality in the market data.

Mathematical notation
The content of this scientific report has its main focus on methods derived from the field of signal processing along with its application to relevant market data. The reader is therefore assumed to have some basic understanding in calculus, linear algebra and probability theory. The mathematical notations are consistent throughout the report, although at times departed from some of the convention used in the corresponding research literature. All vectors are denoted by bold letters and are assumed to be column vectors, that is, a vector $\mathbf{x}$ consisting of p numbers is written as

$$\mathbf{x} = [x_1,\, x_2,\, \ldots,\, x_p]^{\mathrm{T}}$$

forming a vector in $\mathbb{R}^{p\times 1}$, where $\mathbf{x}^{\mathrm{T}}$ indicates the transpose of $\mathbf{x}$. If, for example, N observations $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$ of the p-dimensional vector $\mathbf{x}$ are obtained, these observations can form a data matrix $\mathbf{X}$ in which the i^{th} column of $\mathbf{X}$

corresponds to the column vector $\mathbf{x}_i$.

2 High-frequency trading

The way trades are made have evolved drastically over the years. Many years ago, securities markets were run in an entirely manual fashion. To request a quote[1], a client would contact his/her sales representative in person or via telephone or messenger. The salesperson would then walk over or shout to the trading representative a request for prices on the securities of interest to the client. The trader would then report back the market prices obtained from other brokers. The process was slow, prone to errors and expensive. A big drawback was that the markets could move significantly between the time the market price was set on an exchange and the time the client received the quote [2]. The rapid development of computer technology, however, started opening up to electronic dealing systems where market data across multiple dealers and exchanges distributed information simultaneously to multiple market participants. This allowed parties to trade with each other at the best available prices displayed on the systems. In response to advances in computer technology the so called *algorithmic trading* started to develop, where algorithms makes transaction decisions based on mathematical models of the market. This evolution has resulted in the so called *high-frequency trading*, where a high number of trades (and lower average gain per trade) are made continuously [2].

2.1 Market quote data

A market maker is a company or an individual that quotes both a buy and a sell price in a financial instrument or commodity [3]. In the foreign exchange market, a market maker quotes buy and sell prices for currencies relative to each other, called *currency pairs*. One example is the EUR/USD instrument. For this particular currency pair, the EUR is the *base currency*, which is the currency a buyer intend to buy. The USD is called the *quote currency*, and is the price of one unit of the base currency [4]. The quote data is irregularly spaced in time, arriving randomly at very short time intervals. Each quote includes especially

- a timestamp
- a bid price
- an offer price

[1]The most recent price on which a buyer and seller agreed and at which some amount of the asset was transacted.

- an available bid volume
- an available offer volume
- an indicator of the current currency pair

The timestamp records the date and time which the quote originated. The bid price is the highest price available at the available bid volume for sale. The offer price is the lowest price entered for buying the security at the available offer volume. Figure 1 illustrates an example of how the market quote data for EUR/USD instrument from a market maker $\mathcal{M}$ can behave over time.

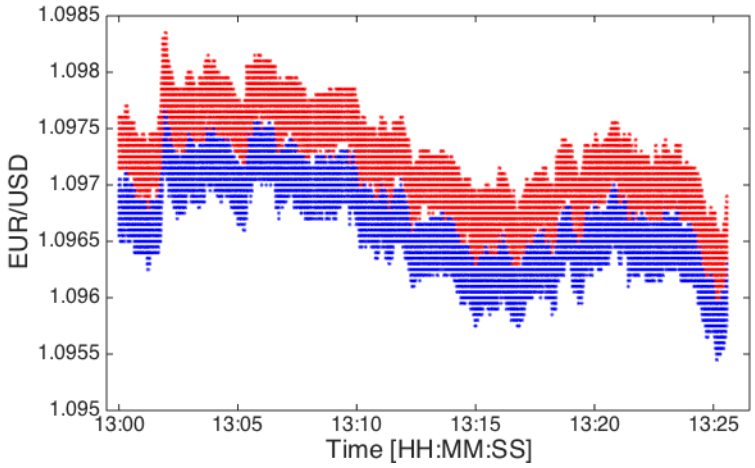

Figure 1: An example of a market quote data set for the EUR/USD instrument published by a specific market maker $\mathcal{M}$. For this particular market, each quote consists of 10 offer (or sell) prices, marked with red, and 10 bid (or buy) prices, marked with blue. Each price for a specific volume.

As can be seen in figure 1, the market maker quotes 10 bid prices (blue dots) and 10 offer prices (red dots) in each quote update. Each offer price has a corresponding bid price, forming so called bid-offer pairs. The graph is plotted with dots to illustrate the irregularly spaced quotes from the market maker. The whole sequence of quote updates will be referred to as a market quote set, or market data set. In this report, market quote data and market data will be used interchangeably.

3 Principal Component Analysis (PCA)

The central idea of principal component analysis (PCA) is to reduce the dimensionality of a multivariate data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data. This is achieved by transforming the data set to a new set of variables, known as the principal components, which are uncorrelated, and which are ordered such that the first components retain most of the variation in all of the original variables.

3.1 Definition of Principal Components

Consider a vector $\mathbf{x}$, consisting of p random variables, and that the variances and the characteristics of the covariances between the p variables are of interest. The idea is to look for a few $q \ll p$ variables that preserve most of the information given by these covariances.

The first step in finding the principal components is to regard a linear function $\boldsymbol{\alpha}_1^{\mathrm{T}}\mathbf{x}$ of the elements of $\mathbf{x}$ having maximum variance. The coefficient vector $\boldsymbol{\alpha}_1$ consists of p constants $\alpha_{11}, \alpha_{12}, ..., \alpha_{1p}$, so that

$$z_1 := \boldsymbol{\alpha}_1^{\mathrm{T}}\mathbf{x} = \alpha_{11}x_1 + \alpha_{12}x_2 + \ldots + \alpha_{1p}x_p,$$

which is a linear transformation of $\mathbf{x}$ on to the vector $\boldsymbol{\alpha}_1$, known as a principal component or *score* [5]. The next step is to consider a new linear transformation $\boldsymbol{\alpha}_2^{\mathrm{T}}\mathbf{x}$, uncorrelated with $\boldsymbol{\alpha}_1^{\mathrm{T}}\mathbf{x}$, which also obtains maximum variance. Proceeding in the same way, the p^{th} step results in obtaining $\boldsymbol{\alpha}_p^{\mathrm{T}}\mathbf{x}$, that maximizes the variance subject to being uncorrelated with the previous $p-1$ linear transformations $\boldsymbol{\alpha}_1^{\mathrm{T}}\mathbf{x}$, $\boldsymbol{\alpha}_2^{\mathrm{T}}\mathbf{x}, \ldots, \boldsymbol{\alpha}_{p-1}^{\mathrm{T}}\mathbf{x}$. Up to p principal components can be found, but it is hoped that most of the variation in $\mathbf{x}$ will be accounted for by $q \ll p$ principal components. The p coefficients vectors spans a p-dimensional linear subspace defined by the transformation matrix

$$\mathbf{V} = \left[\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_p\right] \in \mathbb{R}^{p\times p},$$

such that the k^{th} column of $\mathbf{V}$ corresponds to the k^{th} principal component coefficient- or *loading* vector $\boldsymbol{\alpha}_k$ [5]. Furthermore, the p-dimensional principal component vector, denoted by $\mathbf{z}$, is defined as

$$\mathbf{z} = \mathbf{V}^{\mathrm{T}}\mathbf{x} \in \mathbb{R}^{p\times 1}, \tag{1}$$

whose k^{th} element consists of the k^{th} principal component score, obtained by

$$z_k = \boldsymbol{\alpha}_k^{\mathrm{T}}\mathbf{x}, \;\; k = 1, \ldots, p.$$

Equation (1) is basically a linear transformation of the random vector $\mathbf{x}$ on to the subspace spanned by the principal components loading vectors, or column vectors, in $\mathbf{V}$. This means that the principal component scores can be seen as the resulting transformation of $\mathbf{x}$ on to a subspace, where the variance of the variables in $\mathbf{x}$ are maximized. How to obtain the principal components is described in the following section.

3.2 Obtaining the principal components

To obtain the principal components, the loading vectors $\boldsymbol{\alpha}_k$ need to be obtained. Consider the case where the p-dimensional random vector $\mathbf{x}$ has a known covariance matrix $\boldsymbol{\Sigma}$. The condition of the loading vector $\boldsymbol{\alpha}_k$ is that it should maximize the variance of $z_k = \boldsymbol{\alpha}_k^{\mathrm{T}}\mathbf{x}$, subject to being orthogonal to $\boldsymbol{\alpha}_{k-1}$. Thus,

$$\boldsymbol{\alpha}_k = \arg\max_{\boldsymbol{\alpha}_k}\{\mathrm{var}(\boldsymbol{\alpha}_{\mathrm{k}}^{\mathrm{T}}\mathbf{x})\} = \arg\max_{\boldsymbol{\alpha}_k}\{\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\alpha}_k\}, \tag{2}$$

where the relationship $\mathrm{var}(\boldsymbol{\alpha}_k^{\mathrm{T}}\mathbf{x}) = \boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\alpha}_k$ has been used. Without any constraints, a trivial solution to (2) would be to choose $\boldsymbol{\alpha}_k$ to be very big. Therefore, a normalization constraint of $\|\boldsymbol{\alpha}_k\|^2 = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}_k = 1$ is introduced. To maximize $\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\alpha}_k$ subjected to $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}_k = 1$, the standard approach is to use the technique of Lagrange multipliers, and therefore maximizing

$$\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\alpha}_k - \lambda_k(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha}_k - 1), \tag{3}$$

where λ_k is a Lagrange multiplier. Differentiation with respect to $\boldsymbol{\alpha}_k$ gives

$$(\boldsymbol{\Sigma} - \lambda_k\mathbf{I}_p)\boldsymbol{\alpha}_k = \mathbf{0}, \tag{4}$$

where $\mathbf{I}_p$ is the $[p \times p]$ identity matrix and $\mathbf{0}$ is the zero matrix. Thus, λ_k is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{\alpha}_k$ is the corresponding eigenvector. From (4) it can be seen that $\boldsymbol{\Sigma}\boldsymbol{\alpha}_k = \lambda_k\boldsymbol{\alpha}_k$, and therefore $\boldsymbol{\alpha}_k$ is the eigenvector corresponding to the k^{th} largest eigenvalue of $\boldsymbol{\Sigma}$. Similarly, $\mathrm{var}(\boldsymbol{\alpha}_k^{\mathrm{T}}\mathbf{x}) = \boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\alpha}_k = \boldsymbol{\alpha}_k^{\mathrm{T}}\lambda_k\boldsymbol{\alpha}_k = \lambda_k$.

As seen, the principal component loading vector $\boldsymbol{\alpha}_k$ in $\mathbf{V}$, corresponds to the k^{th} eigenvector of $\boldsymbol{\Sigma}$. Thus, the principal components in $\mathbf{z}$ are defined by an orthonormal linear transformation of $\mathbf{x}$ defined by (1) [5]. Furthermore, because $\mathbf{V}$ is orthogonal, that is, $\mathbf{V}^{\mathrm{T}} = \mathbf{V}^{-1}$, the covariance matrix can be expressed as

$$\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}} \in \mathbb{R}^{p\times p}, \tag{5}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ is the diagonal matrix whose k^{th} diagonal element is λ_k, corresponding to the k^{th} eigenvalue of $\boldsymbol{\Sigma}$. The eigenvalues in $\boldsymbol{\Lambda}$ are ordered in increasing order, such that,

$$\lambda_1 > \lambda_2 > \ldots > \lambda_p.$$

This means the first eigenvector $\boldsymbol{\alpha}_1$ retains the most of the variance in $\mathbf{x}$ and the second eigenvector $\boldsymbol{\alpha}_2$ second most and so on. The expression in (5) is the eigenvalue decomposition of $\boldsymbol{\Sigma}$. When $\boldsymbol{\Sigma}$ is positive semidefinite this coincides with the singular value decomposition (SVD) [6]. There are many methods for computing an SVD, however, this topic will not be treated in this report.

3.2.1 Unknown covariance matrix

In the previous section, the covariance matrix for the random vector $\mathbf{x}$ where assumed to be known. However, the more realistic case, where $\boldsymbol{\Sigma}$ is unknown, follows by replacing the covariance matrix with the *sample* covariance matrix.

Suppose there are $i = 1, \ldots, N$ independent observations of the p-dimensional random vector $\mathbf{x}$, forming a $[p \times N]$ matrix $\mathbf{X}$ such that

$$\mathbf{X} = \Big[\mathbf{x}_1,\, \mathbf{x}_2,\, \ldots\, , \mathbf{x}_N\Big] \in \mathbb{R}^{p\times N}.$$

Then the sample covariance matrix will be given by

$$\widehat{\boldsymbol{\Sigma}} = \begin{bmatrix} \mathrm{c}(x_1, x_1) & \mathrm{c}(x_1, x_2) & \ldots & \mathrm{c}(x_1, x_p) \\ \mathrm{c}(x_2, x_1) & \mathrm{c}(x_2, x_2) & \ldots & \mathrm{c}(x_2, x_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{c}(x_p, x_1) & \mathrm{c}(x_p, x_2) & \ldots & \mathrm{c}(x_p, x_p) \end{bmatrix} \in \mathbb{R}^{p\times p}. \tag{6}$$

where $\mathrm{c}(x_j, x_k)$ is the sample covariance between the j^{th} and k^{th} variables in $\mathbf{x}$ obtained by

$$\mathrm{c}(x_j, x_k) = \frac{1}{N-1} \sum_{i=1}^{N} (x_{ji} - \bar{x}_j)(x_{ki} - \bar{x}_k). \tag{7}$$

Here, $\bar{x}_k$ represents the sample mean for the k^{th} variable, and is given by

$$\bar{x}_k = \frac{1}{N} \sum_{i=1}^{N} x_{ki}. \tag{8}$$

Note that for $j = k$, the covariance is the variance of the variable, that is, $\mathrm{cov}(x_j, \tilde{x}_j) = \sigma_j^2$. Hence, the diagonal element (j, j) in the sample covariance matrix corresponds to the variance of the j^{th} variable. The normalization factor $(N-1)$ in (9) makes the sample covariance matrix the best unbiased estimate for the covariance matrix if the observations of $\mathbf{x}$ are from a Gaussian distribution [8].

Another way of expressing the covariance matrix is to consider the $[p \times N]$ matrix $\widetilde{\mathbf{X}}$ with $(j, k)^{th}$ element $(x_{jk} - \bar{x}_j)$. That is, each row in $\widetilde{\mathbf{X}}$ has been reduced by its corresponding sample mean $\bar{x}_j$. The sample covariance matrix of $\mathbf{x}$ is then obtained by

$$\widehat{\mathbf{\Sigma}} = \frac{1}{N-1} \widetilde{\mathbf{X}} \widetilde{\mathbf{X}}^{\mathrm{T}} \in \mathbb{R}^{p \times p}, \tag{9}$$

and the principal component scores for the N observations for the mean-centered data is finally defined as

$$\widetilde{\mathbf{Z}} = \mathbf{V}^{\mathrm{T}} \widetilde{\mathbf{X}} \in \mathbb{R}^{p \times N}, \tag{10}$$

where the columns of $\mathbf{V}$ now corresponds to the eigenvectors of $\widehat{\mathbf{\Sigma}}$, calculated in the same way as for the known covariance matrix $\mathbf{\Sigma}$. It should be noted that for all analysis in this report the covariance matrix will be unknown, and hence the sample covariance matrix will be used. However, to facilitate cross references to other literature, the covariance matrix will be denoted by $\mathbf{\Sigma}$, since it is the most common notation in literatures.

3.3 Dimensionality reduction

As seen in the previous section, the transformation $\mathbf{Z} = \mathbf{V}^{\mathrm{T}}\mathbf{X}$ transforms a data vector $\mathbf{x}$ from an original space of p variables to a new space of p variables which are uncorrelated. However, it is hoped most of the variation in $\mathbf{x}$ will be accounted for by $q \ll p$ principal components. Keeping only the first q principal components, produced by using only the first q loading vectors, gives the truncated transformation

$$\hat{\mathbf{Z}} = \hat{\mathbf{V}}^{\mathrm{T}}\mathbf{X} \in \mathbb{R}^{q \times N}, \tag{11}$$

where only the first q columns of $\mathbf{V}$ has been chosen, such that $\hat{\mathbf{V}} \in \mathbb{R}^{p \times q}$. Hence, the dimensionality of the data has been reduced from p to q by simply selecting the first q columns of $\mathbf{V}$. Of course, such dimensionality reduction implies throwing away information in the original data. However, a sensible number of principal components to include should be the number such that 70% to 90% of the variance in the original data is captured [5]. Since the eigenvalues of the covariance matrix corresponds to the variance each principal component explain, the proportion of variance the first q principal components make up for is determined by the ratio

$$\mathrm{PC}_{\mathrm{variance}} = \frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{p} \lambda_k}. \tag{12}$$

3.4 Correlation matrix

The derivation of the principal components considered in precious sections are based on the eigenvectors and eigenvalues of the covariance matrix. However, principal components can be obtained from the correlation matrix $\mathbf{R}_{\mathbf{x}}$ aswell. The correlation matrix is calculated in the similar way as the covariance matrix, with the only difference being that each variable in the data set matrix $\mathbf{X}$ is normalized by its standard deviation σ_j. The correlation matrix is defined as followed.

$$\mathbf{R}_{\mathbf{x}} = \begin{bmatrix} 1 & \rho_{12} & \dots & \rho_{1p} \\ \rho_{21} & 1 & \dots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \dots & 1 \end{bmatrix} \in \mathbb{R}^{p \times p}, \tag{13}$$

where ρ_{jk} is the correlation coefficient obtained by normalizing the covariance between the j^{th} and k^{th} variables with the product of their corresponding standard deviations, that is,

$$\rho_{jk} = \frac{\text{cov}(x_j, x_k)}{\sigma_j \sigma_k}. \tag{14}$$

The elements in the correlation matrix are standardized measures of the interdependence between the p variables in $\mathbf{x}$, and takes values between 1 and -1. The diagonal elements ρ_{jj} in $\mathbf{R}_{\mathbf{x}}$ are equal to 1 since the each variable in $\mathbf{x}$ has been normalized by its standard deviation. In general, a correlation coefficient of $\rho_{jk} = 1$ means that the j^{th} and k^{th} variables have a perfect positive correlation, whereas, if $\rho_{jk} = -1$, the variables are perfectly negative correlated. If $\rho_{jk} = 0$, the two variables have no linear relationship, and are therefore uncorrelated.

The sample correlation matrix can be obtained in a similar way as the covariance matrix, by

$$\widehat{\mathbf{R}}_{\mathbf{x}} = \frac{1}{N-1} \widetilde{\mathbf{X}} \widetilde{\mathbf{X}}^{\mathrm{T}}, \tag{15}$$

where the j^{th} row in $\widetilde{\mathbf{X}}$ now has been normalized by its corresponding sample standard deviation σ_j. A major drawback of PCA based on covariance matrices $\mathbf{\Sigma}$ is the sensitivity to the units of measurement used for each element of $\mathbf{x}$. If large differences in variances x_i and x_j are simply due to unit scaling, then those variables whose variance are largest will tend to dominate the first principal components in a misleading manner. A major argument of PCA based on correlation matrices $\mathbf{R}_{\mathbf{x}}$ is, because each variable in the data set matrix $\mathbf{X}$ is scaled by its corresponding sample standard deviation, the resulting maximum deviation becomes 1 for each variable.

4 Pattern recognition

Pattern recognition is the process of classifying data into different classes based on some key features of the data. Based on training data from different object classes, the idea is to assign new input data into one of the appropriate classes [11]. The training data sets used in such pattern recognition algorithms typically consists of a large set of N samples $\{\mathbf{x}_1, ..., \mathbf{x}_N\}$. The categories of these samples in the training sets are known in advance, typically by inspecting them individually and hand-labelling them. The parameters of the pattern recognition model to be used is then constructed based on these training data, which is referred to as the *training* phase. For most practical applications, the original input data are typically pre-processed to transform them into some new space of data set where, it is hoped, the pattern recognition problem will be easier to solve. This pre-processing stage is called the *feature extraction* and is performed on both the training data and the new input data to be classified [8]. The most general case of classification is to, given a new input vector $\mathbf{x}$, assign it to one of K discrete classes C_k, where $k = 1, ..., K$. In the most common scenario, the classes are taken to be disjoint, so that each input is assigned to one and only one class. The input space is thereby divided into so called *decision boundaries*.

In this section two suggested classification methods, called *Quadratic Discriminant Analysis (QDA)* and *Single-class detector*, are presented along with their respective properties.

4.1 Quadratic Discriminant Analysis (QDA)

Assume that $\mathbf{x}$ originates from class C_k and is described by a *multivariate Gaussian distribution* with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, that is,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

The class-conditional probability density function for $\mathbf{x}$ is then described by

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \Big\}, \tag{16}$$

where p is the dimension of the vector $\mathbf{x}$ and $|\boldsymbol{\Sigma}_k|$ is the determinant of $\boldsymbol{\Sigma}_k$. The probability that the vector $\mathbf{x}$ originates from class C_k can be described

according to Bayes' theorem [8], that is,

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_{j=1}^{K} p(\mathbf{x}|C_j)p(C_j)}, \tag{17}$$

where $p(C_j)$ are the so called *class prior probabilities*. Since the goal is to select which class the vector $\mathbf{x}$ originated from, it is quite intuitive that the class C_k to be selected should be the one having the highest class posterior probability $p(C_k|\mathbf{x})$. Hence, the *classification rule* will be as followed:

$$G(\mathbf{x}) = \arg\max_k \{p(C_k|\mathbf{x})\}, \tag{18}$$

which is known as the *maximum a posteriori* [12]. Noting that the denominator in (17) does not depend on k, (17) and (18) gives

$$G(\mathbf{x}) = \arg\max_k \{p(\mathbf{x}|C_k)p(C_k)\}. \tag{19}$$

Since the logarithm is a monotonically increasing function[2], maximizing the argument of (19) will yield the same maximum as obtained when maximizing the logarithm of it. This will simplify the subsequent mathematical analysis. Taking the logarithm of (19), the classification rule becomes

$$\hat{G}(\mathbf{x}) := \ln[G(\mathbf{x})] = \arg\max_k \ln\Big[p(\mathbf{x}|C_k)p(C_k)\Big]. \tag{20}$$

Using (16) and (20) yields

$$\hat{G}(\mathbf{x}) = \arg\max_k \{\delta_k(\mathbf{x})\}, \tag{21}$$

where $\delta_k(\mathbf{x})$ has been defined as:

$$\delta_k(\mathbf{x}) = \ln\left[\frac{1}{(2\pi)^{p/2}}\frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}\right] + \ln\Big[p(C_k)\Big] - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) \tag{22}$$

[2]A function $f(x)$ is monotonically increasing if $f(x)$ is increasing for increasing values of x [13].

See appendix 10.2.1 for complete derivation. The classification method should therefore choose the class C_k for which $\delta_k(\mathbf{x})$ is maximized. $\delta_k(\mathbf{x})$ is called a *quadratic discriminant function*, since it is a quadratic function of the input vector $\mathbf{x}$ and returns a scalar, i.e. $\delta_k(\mathbf{x}) \in \mathbb{R}$. Hence, the method is called *Quadratic Discriminant Analysis (QDA)*, and the quadratic function is applicable for any distribution of $\mathbf{x}$ with a well-defined mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ [8].

4.1.1 QDA – two classes

Consider only two possible classes C_0 and C_1 for $\mathbf{x}$. The ratio between the a posteriori probabilities of the two classes is then defined by

$$R = \frac{p(C_0|\mathbf{x})}{p(C_1|\mathbf{x})}. \tag{23}$$

If $R > 1$, then $p(C_0|\mathbf{x}) > p(C_1|\mathbf{x})$, and the classification rule defined by (18) should select C_0. If $R < 1$, then $p(C_1|\mathbf{x}) > p(C_0|\mathbf{x})$, and the classification rule should select C_1 instead. Hence, the class C_k will be chosen depending on the output of (23). Proceeding in the same way as for K classes, using (16), (17) and (23) yields

$$R = \frac{p(\mathbf{x}|C_0)p(C_0)}{p(\mathbf{x}|C_1)p(C_1)} = \frac{\frac{1}{|\boldsymbol{\Sigma}_0|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)\right\}}{\frac{1}{|\boldsymbol{\Sigma}_1|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right\}} \cdot \frac{p(C_0)}{p(C_1)}, \tag{24}$$

where the ratios $\frac{|\boldsymbol{\Sigma}_1|^{1/2}}{|\boldsymbol{\Sigma}_0|^{1/2}}$ and $\frac{p(C_0)}{p(C_1)}$ are constants. By taking the natural logarithm of (24), we obtain

$$\begin{aligned} \mathcal{T}(\mathbf{x}) = &\ln[R] = \delta_0(\mathbf{x}) - \delta_1(\mathbf{x}) = \\ &= \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) \\ &- \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) \\ &+ \frac{1}{2}\ln\left[\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right] + \ln\left[\frac{p(C_0)}{p(C_1)}\right], \end{aligned} \tag{25}$$

resulting in a quadratic discriminant function $\mathcal{T}(\mathbf{x}) \in \mathbb{R}$ (see appendix 10.2.2). The classification method should therefore predict $\mathbf{x}$ as being from class C_0 or C_1 according to

$$\mathbf{x} \in \begin{cases} C_0, \text{ if } \mathcal{T}(\mathbf{x}) > 0 \\ C_1, \text{ if } \mathcal{T}(\mathbf{x}) < 0 \end{cases}$$

4.2 Single-class detector

In the case of only having one class C_0, the main interest is to classify if the vector $\mathbf{x}$ originates from class C_0 or *not*. For C_0 we have

$$\delta_0(\mathbf{x}) = \ln\left[\frac{1}{(2\pi)^{p/2}}\frac{1}{|\boldsymbol{\Sigma}_0|^{1/2}}\right] + \ln\Big[p(C_0)\Big] - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0). \quad (26)$$

The first and second terms are constants that are independent of $\mathbf{x}$ and can therefore be ignored, while the third term $(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)$ can be seen as a measure of the distance between the vector $\mathbf{x}$ and $\boldsymbol{\mu}_0$ weighted by $\boldsymbol{\Sigma}_0^{-1}$. In fact, it can be shown that, if

$$\mathbf{x} \in \mathbb{R}^p \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0),$$

then

$$\mathcal{T}(\mathbf{x}) = (\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) \sim \chi_p^2(\mathcal{T}), \quad (27)$$

where χ_p^2 is the *chi-squared distribution* with p degrees of freedom [15]. If the reader is not familiarized with the chi-squared distribution, refer to Appendix 10.3.

Basically, the scalar $\mathcal{T}(\mathbf{x})$ describes how good the input vector $\mathbf{x}$ fits the multivariate Gaussian distribution described by $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. Since $\mathcal{T}(\mathbf{x}) \sim \chi_p^2$, the *chi-square goodness-of-fit test* can be applied to evaluate how likely it is that $\mathbf{x}$ fits the Gaussian distribution [18]. The procedure of the test includes the following steps for a new, p-dimensional, input vector $\mathbf{x}$:

1. Calculate the chi-squared test statistic, $\mathcal{T}(\mathbf{x})$.
2. Determine the degrees of freedom, p, for the distribution.

3. Select a desired significance level, α, for the test.

4. Compare $\mathcal{T}(\mathbf{x})$ to the critical value, $\mathcal{T}_{critical}$.

5. Accept or reject the *null hypothesis* that the observed distribution of $\mathbf{x}$ is *different* from the distribution described by $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. If the test statistic $\mathcal{T}(\mathbf{x}) > \mathcal{T}_{critical}$, the null hypothesis can be rejected with the confidence level $1 - \alpha$. Thus, accepting the alternative hypothesis that there *is* a difference between the distributions.

The number of degrees of freedom, p, in the test will be the number of variables in $\mathbf{x}$, since it's the number of parameters that can vary independently of each other. The significance level, α, is the probability of rejecting the null hypothesis given that it is true [19]. For example, a significance level of 5% means that the null hypothesis will be rejected if the probability of $\mathcal{T}(\mathbf{x})$ is less then or equal to the confidence level 95%. Thus, the critical value of the test statistic is determined by α , given by

$$\mathcal{T}_{critical} = Q_p(1 - \alpha), \tag{28}$$

where $Q_p(1 - \alpha)$ is the inverse of the cumulative distribution function (cdf) of the chi-squared distribution for p degrees of freedom. See Appendix 10.3 for definition.

The combination of using the discriminant function $\mathcal{T}(\mathbf{x})$ described by (27) together with the chi-squared goodness-of-fit test could therefore serve as a classification method for determining how likely it is that the new input vector $\mathbf{x}$ originates from the class C_0. It could also be extended to the general case of K classes, by first obtaining the mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$ for each class, then calculate $\mathcal{T}_k(\mathbf{x})$ followed by performing the goodness-of-fit test for each class separately.

4.3 Conditions on the covariance matrix

The classification methods in section 4.1 are all derived from the assumption that the random variable vector $\mathbf{x}$ for a given class C_k originates from a multivariate Gaussian distribution with a probability density function described by (16). However, the probability density function for a multivariate Gaussian distribution does only exist under some conditions, especially regarding the rank and inverse of the covariance matrix $\boldsymbol{\Sigma}$. For definitions see Appendix 10.1.1 and 10.1.2.

Assume that $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$, where $\boldsymbol{\mu}_x$ is the $[p \times 1]$ mean vector and $\boldsymbol{\Sigma}_x$ is the $[p \times p]$ covariance matrix for $\mathbf{x}$. Let $\text{rank}(\boldsymbol{\Sigma}_x)$ denote the rank of $\boldsymbol{\Sigma}_x$. Then the following are true:

1. If $\text{rank}(\boldsymbol{\Sigma}_x) = p$, then $\boldsymbol{\Sigma}_x^{-1}$ exists and the density function of $\mathbf{x}$ is described by (16).

2. If $\text{rank}(\boldsymbol{\Sigma}_x) = q < p$, then $\boldsymbol{\Sigma}_x^{-1}$ does not exist and therefore no explicit determination of the density function of $\mathbf{x}$ is possible. However, the density function exists on a linear subspace:

   Let $\mathbf{V}$ be a $[p \times q]$ matrix of orthonormal column vectors belonging to the linear space spanned by the columns of $\boldsymbol{\Sigma}_x$. Consider the transformation $\mathbf{x} \in \mathbb{R}^{p\times 1} \rightarrow \mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{x} \in \mathbb{R}^{q\times 1}$. Then

$$\boldsymbol{\mu}_y = \mathrm{E}[\mathbf{y}] = \mathbf{V}^{\mathrm{T}}\boldsymbol{\mu}_x \in \mathbb{R}^{q\times 1} \tag{29}$$

   and

$$\boldsymbol{\Sigma}_y = \mathbf{V}^{\mathrm{T}}\boldsymbol{\Sigma}_x\mathbf{V} \in \mathbb{R}^{q\times q} \tag{30}$$

   so that

$$\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y). \tag{31}$$

   So the probability density function for $\mathbf{y}$ will be

$$p(\mathbf{y}|\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y) = \frac{1}{(2\pi)^{q/2}} \frac{1}{|\boldsymbol{\Sigma}_y|^{1/2}} \exp\Big\{ -\frac{1}{2}(\mathbf{y} - \boldsymbol{\mu}_y)^{\mathrm{T}}\boldsymbol{\Sigma}_y^{-1}(\mathbf{y} - \boldsymbol{\mu}_y)\Big\}. \tag{32}$$

   For proof, refer to [16].

When $\boldsymbol{\Sigma}_x$ has low rank, one or more eigenvalues of the matrix are zero. The above result shows that if $\boldsymbol{\Sigma}_x$ does not have full rank, one can transform the variables in $\mathbf{x}$ into another subspace of lower dimensionality $q < p$ by the transformation

$$\mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{x}, \tag{33}$$

yielding a new, q-dimensional, random variable $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ with a mean vector, covariance matrix and probability density function described by (29), 30 and 32 respectively. Note that the columns of the orthonormal matrix $\mathbf{V}$ belongs to the column space of $\boldsymbol{\Sigma}_x$. This is also the case of the principal components described in section 4.1. Thus, PCA can be used for discriminant analysis of low-rank data.

4.3.1 Multicollinearity

If multiple variables are highly correlated they are said to be multicollinear, meaning that there is a linear relationship among the them. Assume that there exist N samples of the p-dimensional random vector $\mathbf{x}$, forming a $[p \times N]$ matrix $\mathbf{X}$. If there is an exact linear relationship (i.e. perfect multicollinearity) among the variables in $\mathbf{X}$, the rank of $\boldsymbol{\Sigma}_x$ becomes less than p, and hence not invertible [17]. Therefore, if $\mathbf{x}$ consists of multicollinear variables, the variables may need to be transformed into a subspace in order to be able to apply the classification method described in section 4.1.

5 PCA on market data

As seen in section 2, market quote data consists of multiple quote updates, which over time can obtain high dynamical characteristics. Working with high-dimensional data sets is not always easy, since the number of parameters that can explain the behaviour of the data are many. By applying PCA on such multi-dimensional data set, one may be able to identify patterns in the data, as well as analysing the covariance of the variables in it. Since the goal with this project is to construct a robust classification method for classifying if market data is normal or abnormal, some pre-analysis of the characteristics in the market data is required.

5.1 Normal market data

In order to classify if a market data set is normal or abnormal one must find specific parameters (or measures) in the market data that are in some sense unique and which differs for the two different cases. Figure 2 shows an example of a normal market data set. The recorded data is from a market, $\mathcal{M}$, for the EUR/USD instrument, captured between $13.00 - 13.15$.

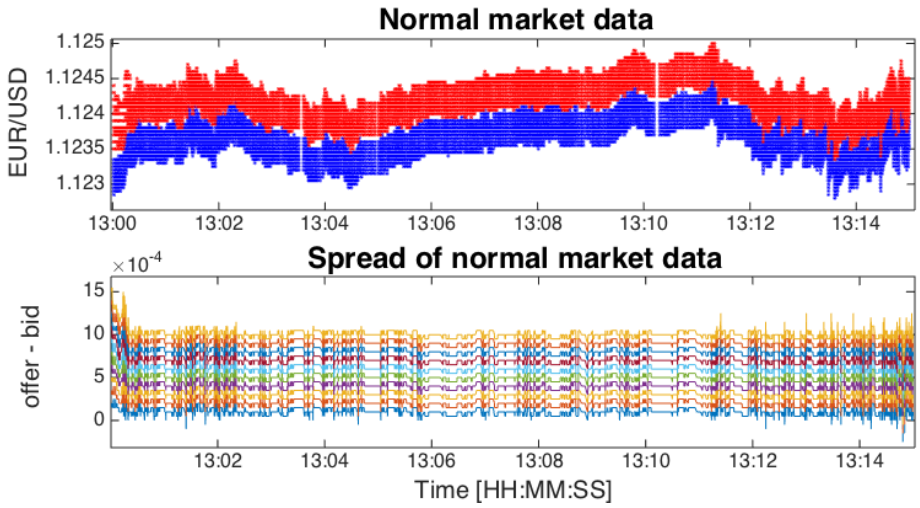

Figure 2: A typically normal market data set for the EUR/USD instrument, from market $\mathcal{M}$. The upper plot shows the bid and offer prices and the lower plot shows the spread for each volume respectively. The x-axis shows the time in both plots.

Since the market data set is quite large (spanned over 15 minutes), only a few conclusions can be made by just looking at the data. For example,

over a larger data set (more 200 than samples), all bid-offer pairs are highly correlated with each other. This can be seen since all bid-offer pairs are following each other over the long run. When the offer price increases, so does the bid price etc. The spread of a bid-offer pair, defined as the difference (offer − bid), should therefore be quite constant over time for all pairs, which is also seen in the lower plot of figure 2. However, looking at a short time interval from the same market data set one can discover other characteristics of the normal market data. This is illustrated in figure 3.

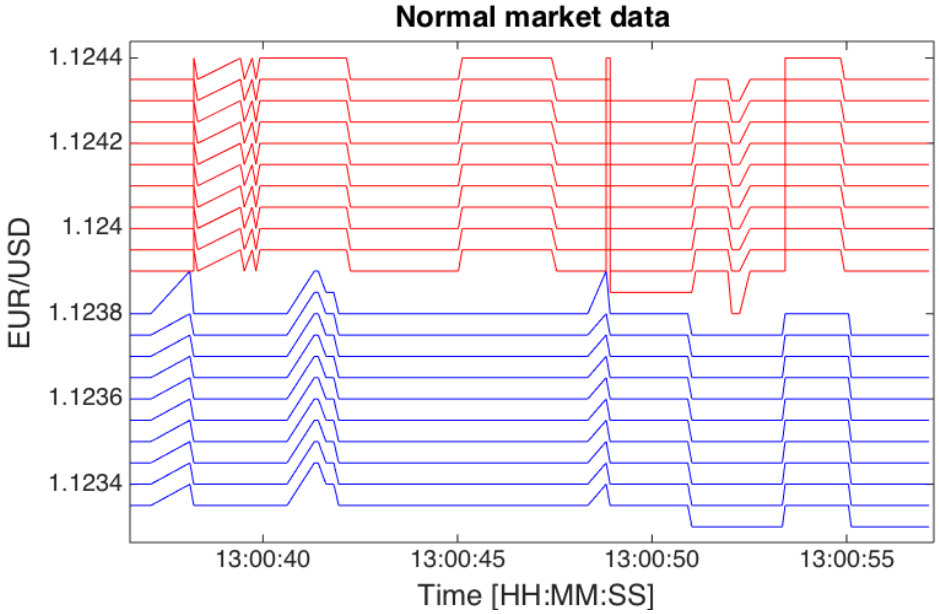

Figure 3: A 20 second snapshot from the normal market data set.

During this time interval, the corresponding bid and offer prices does not appear to be correlated. A change in the offer price does not mean a change in the corresponding bid price and vice versa. However, each offer price is highly correlated with each other and each bid bid price is highly correlated with each other. These properties of the *normal* market data set, as will be seen later, will be important for understanding the results of the PCA. Now, we perform PCA on this normal market data set. Consider the market data matrix $\mathbf{X}$, given by

$$\mathbf{X} = \big[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\big],$$

where $\mathbf{x}_i$ is the i^{th} observation of the random vector $\mathbf{x}$ consisting of all bids and offers, that is,

$$\mathbf{x}_i^{\mathrm{T}} = \left[\mathrm{bid}_{1_i}, \ldots, \mathrm{bid}_{10_i} \mid \mathrm{offer}_{1_i}, \ldots, \mathrm{offer}_{10_i}\right].$$

Each bid and offer quote are here seen as a unique variable. For the EUR/USD instrument from market $\mathcal{M}$, their are 10 bids and 10 offers per updated quote. So the dimension of the random vector $\mathbf{x}$ will be $[20 \times 1]$. Each bid has a corresponding offer such that a bid-offer pair $(\mathrm{bid}, \mathrm{offer})_j$ is quoted for a corresponding volume, and the elements in $\mathbf{x}$ are ordered such that for each quote update

$$(\mathrm{offer}_1 - \mathrm{bid}_1) < (\mathrm{offer}_2 - \mathrm{bid}_2) < \ldots < (\mathrm{offer}_p - \mathrm{bid}_p).$$

This definition of the random variable $\mathbf{x}$ will be used throughout the report unless otherwise is stated. As described in section 3, the principal component loading matrix $\mathbf{V}$ is obtained from the covariance or correlation matrix. Since all variables in $\mathbf{x}$ are measured in the same unit (EUR/USD), the covariance matrix could be used here. However, for reasons explained later in this report, the correlation matrix will be used to obtain $\mathbf{V}$. The PCA of the market data set is performed in MATLAB. For code, refer to Appendix 10.4.1.

For the market data set illustrated in figure 3, the sum of the proportion of the first two eigenvalues of the correlation matrix is approximately 0.9. This means that the first two principal component loadings, that is, the first two columns in $\mathbf{V}$, explains about 90% of the variation in $\mathbf{X}$. Plotting the first two loading vectors in $\mathbf{V}$ against each other results in a so called *biplot*. A biplot shows the magnitude and sign of each variables contribution to the first two principal component loadings. Such two-dimensional plots are particularly useful for visually detecting patterns in the data. It can be shown that the cosine of the angles between the variables contribution in the biplot indicates the correlation between the variables. Highly correlated variables point in the same direction and uncorrelated variables are at perpendicular to each other [5]. A biplot of the first two loading vectors for the data set in this example is illustrated in figure 4.

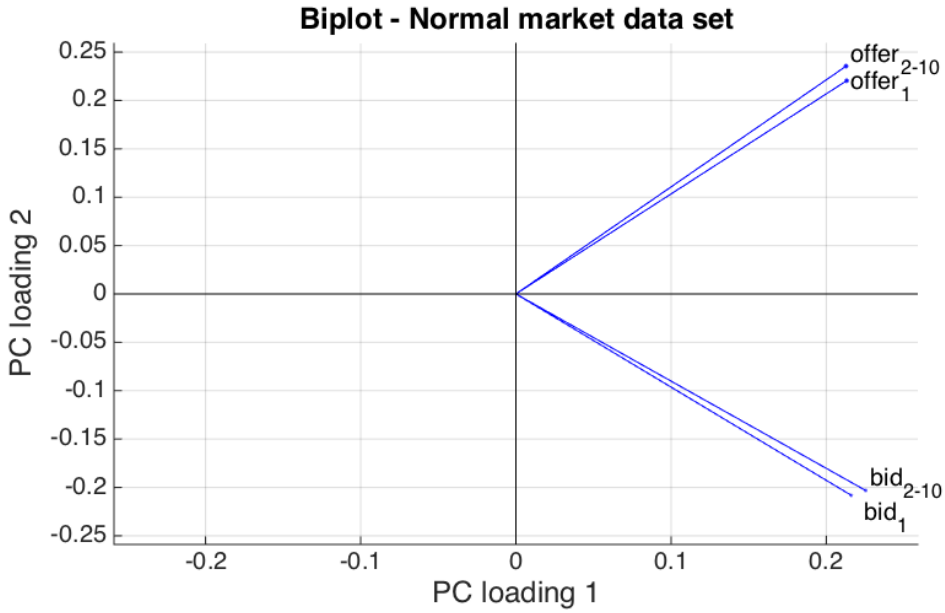

Figure 4: The biplot of the 20 second normal market data set illustrated in figure 3. The x- and y-axis represents the first and second principal component loadings respectively.

As can be seen in the biplot, the majority of the contribution of the bid variables are pointing in the same direction and the majority of the contribution of the offer variables are pointing in the same direction. This means that the bids and offers are highly correlated with each other respectively, which was the first intuition when analysing the market data set. Since 90% of variation in the data is explained by the first two loading vectors, the biplot here visualizes the the correlation (or covariance) characteristics of the original 20-dimensional market data in a 2-dimensional plot accurately.

It should be noted that the number of samples in a data set will affect the result of PCA. For example, if one were to perform PCA on the 15 minute market data set (corresponding to around 9000 samples) shown in figure 2, all the contribution of the variables in the biplot (bids *and* offers) would have appeared to be highly correlated. This is because the variation over short time intervals would be removed due to averaging when obtaining the variance of the variables for big data sets. For consistency, the number of samples per data set analyzed in this report will be around $N = 200$, corresponding to about 20 seconds of data, which can be regarded as a short time interval.

5.2 Abnormal market data

For comparison, lets look at a market data set which can be seen as abnormal. The following data set is captured between $13.25 - 13.40$ for the EUR/USD instrument from market $\mathcal{M}$, and is illustrated in figure 5. At 13.30 the *United States Department of Labor* releases the so called *Nonfarm payroll employment* (or NFP) statistics. The financial assets most affected by the NFP include US dollar, equities and gold [20]. This leads to the guess that the volatility[3] of the EUR/USD instrument should be affected around that time.

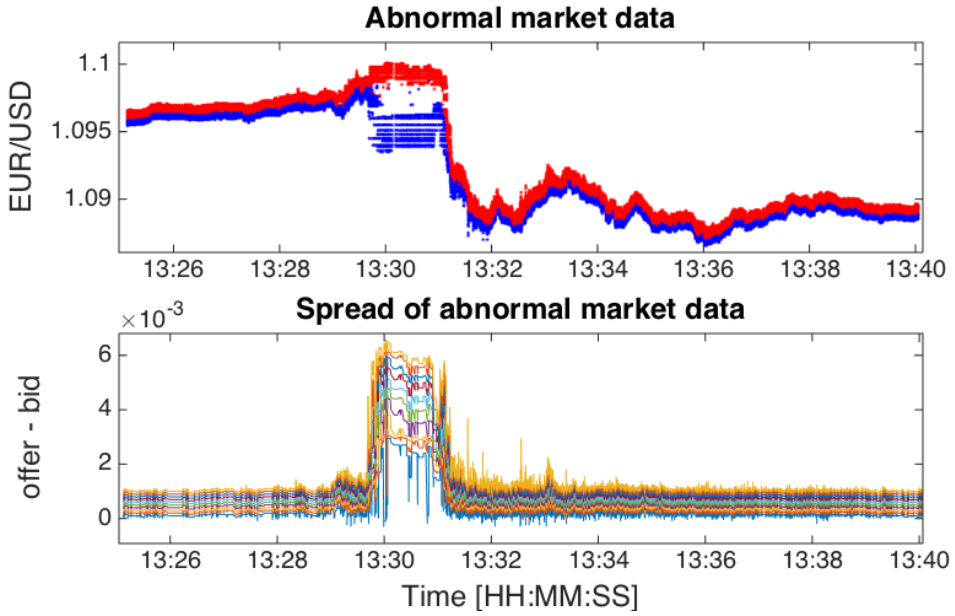

Figure 5: Illustration of an abnormal market data set for the EUR/USD from market maker $\mathcal{M}$. The NFP statistics is released at 13.30 and the effects can be seen in both the upper quote plot and in the lower spread plot.

Again, the large data set (spanned over 15 minutes) makes it more difficult to characterize the data. However, one can clearly see the large deviations of the bid and offer prices at 13.30, both in terms of absolute value and relative behaviour compared to the normal case. Figure 6 shows a 20 second snapshot of the abnormal market data, from 13:29:50 to 13:30:10, that is, 10 seconds before and after the NFP statistics release respectively. For this market data

[3]Volatility refers to the amount of uncertainty or risk about the size of changes in a security's value. A higher volatility means that the value of a security can potentially have a high variance [21].

set the bid and offer prices do not appear to have any particular structure over time. Each bid appears to be more uncorrelated with each other and each offer appears to be more uncorrelated with each other compared to the normal market data set in figure 3.

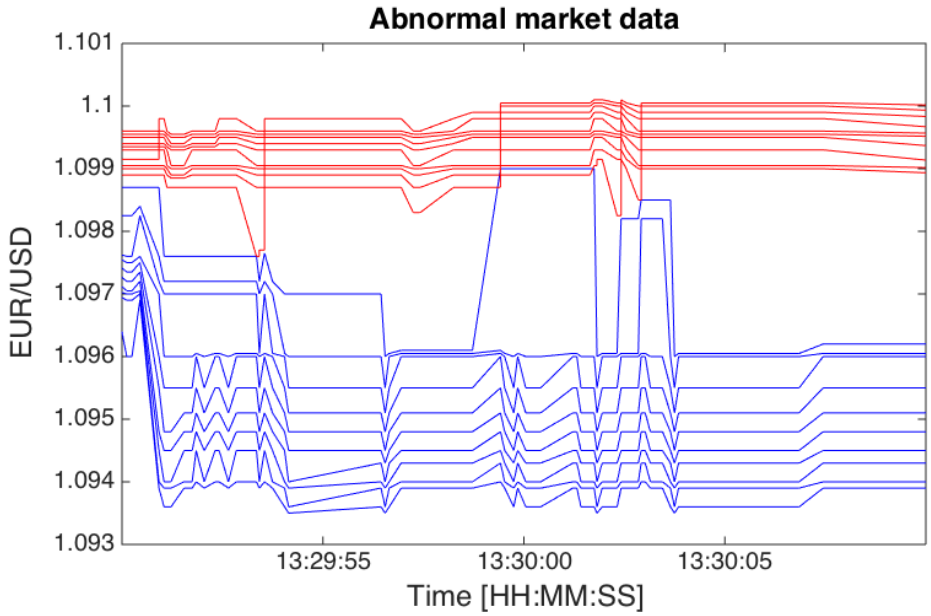

Figure 6: A 20 second snapshot from the abnormal market data set, 10 seconds before and after the NFP statistics release respectively.

Performing PCA on this market data set, in the exact same way as for the normal market data set, it turns out that three principal component loadings is needed to explain 90% of the variance in the data. This can be compared with the normal market data set which only needed two. This is because the variance in the abnormal market data set is highly spread over more variables in $\mathbf{x}$ for this particular data set, leading to a higher number of variables needed to represent the majority of the information in the data. The biplot in this case will not be as accurate as the one for the normal data set. However, the first two loading vectors account for around 80% of the data, so it will not give any misleading information. The biplot of the abnormal market data set is illustrated in figure 7.

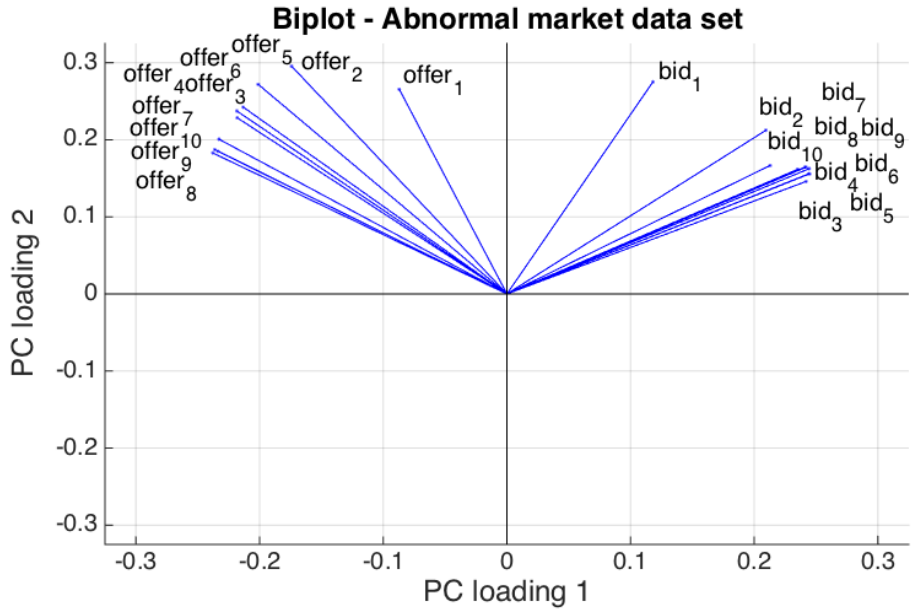

Figure 7: The biplot of the 20 second normal market data set illustrated in figure 6. The x- and y-axis represents the first and second principal component loading respectively.

By analysing the biplot in figure 7, one can see that the contribution of the bid variables are pointing in different directions. The same holds for the contribution of the offer variables. This means that the bid variables are less correlated to each other and the offer variables are less correlated with each other than compared to the normal market data set. Again, this is exactly the first intuition obtained when analysing the time-series of the data.

When comparing the normal and abnormal market data sets provided in this section, it becomes clear that the covariance characteristics between the bid and offer variables should be a suitable measure for classifying if the market data is normal or abnormal. This strengthens the idea of using the suggested classification methods described in section 4.1. Another result which strengthens this idea even more is shown in figure 8. The figure illustrates a histogram of the first bid variable (bid_1) in $\mathbf{x}$ for a randomly selected market data set for the EUR/USD instrument from market $\mathcal{M}$.

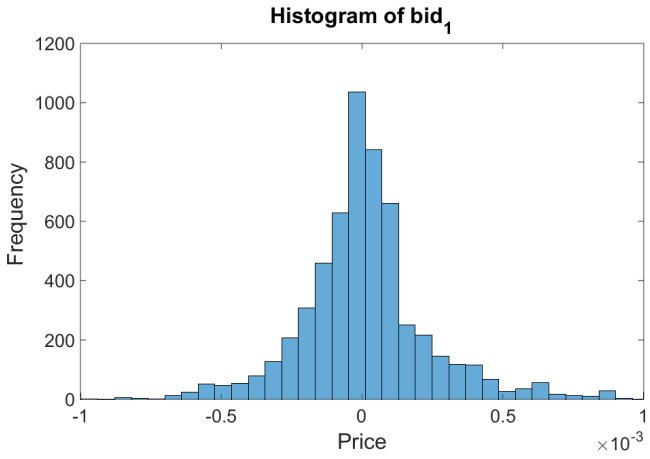

Figure 8: Illustration of the histogram for the bid_1 variable in $\mathbf{x}$ for a random market data set. The variable appears to be Gaussian distributed.

The graph illustrated in figure 8 shows that the variable appears to originate from a Gaussian distribution. It turns out that performing similar analysis on all variables in $\mathbf{x}$ yields similar results. This leads to the idea that the suggested classification methods should be well-suited for these particular data sets, since they are all derived from the assumption that $\mathbf{x}$ originates from a multivariate Gaussian distribution.

6 Comparing principal component spaces

As seen in section 3 and 4, both PCA and the suggested classification methods are all based on the uniqueness of the covariance matrix of the corresponding data. Since PCA transforms the data on to a subspace spanned by the principal component loadings, the subspaces can be used for comparing the covariance matrices for different data sets. Such a comparison could therefore clarify the suitability of the suggested classification methods. Since the goal with the project is to construct a classification method for classifying if a market data set is normal or abnormal, comparisons between the principal component spaces for normal and abnormal market data will be performed.

6.1 Training data matrix

Consider a new matrix $\mathbf{X_0}$ consisting of M number of market data sets, each of length N, such that

$$\mathbf{X_0} = \big[\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\big],$$

forming a $[p \times (N \cdot M)]$ matrix, where $\mathbf{X}_m$ denotes the m^{th} market data set matrix. $\mathbf{X_0}$ will hereafter be referred to as the *training data matrix*, where the M different market data sets will be taken from a normal market data set.

6.2 Normal principal component space

Recall from section 3, that the covariance matrix for the training data matrix $\mathbf{X_0}$ can be written as

$$\mathbf{\Sigma_0} = \mathbf{V_0}\mathbf{\Lambda_0}\mathbf{V_0}^{\mathrm{T}}, \tag{34}$$

where $\mathbf{V_0}$ is the $[p \times p]$ principal component loading matrix, where each column correspond to a principal component loading (or eigenvector) of $\mathbf{\Sigma_0}$. $\mathbf{\Lambda_0}$ is the diagonal matrix whose k^{th} diagonal element λ_k represents the eigenvalue for the k^{th} eigenvector of $\mathbf{\Sigma_0}$, that is, the k^{th} column of $\mathbf{V_0}$. Since the p principal components in $\mathbf{V_0}$ are calculated from the training data matrix $\mathbf{X_0}$, it can be seen as a basis spanning a $\mathbb{R}^{p\times p}$ *normal* principal component space.

Note that here the covariance matrix will be used to obtain the principal component loadings, since the goal is to compare the covariance matrices for the different data sets. Since all variables in $\mathbf{x}$ are measured in the same unit, this is permitted. And, since the covariance matrix is not known, the sample covariance matrix is used to obtain an estimate of $\mathbf{\Sigma_0}$.

Consider a new market data matrix $\mathbf{X}_*$ consisting of I unclassified quote updates (or observations), forming a $[p \times I]$ matrix. A single observation $\mathbf{x}_*$ (i.e. a single column) in the new market data matrix can be expressed as a linear combination of the *normal* principal components $\mathbf{V_0}$ and a corresponding *coefficient vector* $\boldsymbol{\lambda}_* = \mathbf{V}_0^{\mathrm{T}}\mathbf{x}_*$. Extended to a series of I observations, this can be expressed on matrix form as

$$\mathbf{X}_* = \mathbf{V_0}\mathbf{\Lambda}_*, \tag{35}$$

where $\mathbf{\Lambda}_*$ is a $[p \times I]$ coefficient matrix. The i^{th} column of $\mathbf{\Lambda}_*$ corresponds to the coefficient vector for the i^{th} observation projected on to the *normal* principal component space, that is,

$$\mathbf{\Lambda}_* = \begin{bmatrix} \lambda_{11}^* & \lambda_{12}^* & \dots & \lambda_{1I}^* \\ \lambda_{21}^* & \lambda_{22}^* & \dots & \lambda_{2I}^* \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{p1}^* & \lambda_{p2}^* & \dots & \lambda_{pI}^* \end{bmatrix}.$$

Using (35), the coefficient matrix is obtained by

$$\mathbf{\Lambda}_* = \mathbf{V_0}^{\mathrm{T}}\mathbf{X}_* \in \mathbb{R}^{p \times I}, \tag{36}$$

which is a linear transformation of $\mathbf{X}_*$ on to the *normal* principal component space spanned by the columns in $\mathbf{V_0}$. In a similar way, the coefficient matrix for the training data matrix $\mathbf{X_0}$ can be obtained by

$$\mathbf{\Lambda_0} = \mathbf{V_0}^{\mathrm{T}}\mathbf{X_0} \in \mathbb{R}^{p \times (I \cdot M)}. \tag{37}$$

In this way, the coefficients for all $i = 1, ..., I$ observations in the new market data set $\mathbf{X}_*$, represented in the base of a normal principal component space, can be compared with the coefficients for a normal data set $\mathbf{X_0}$ in that same space. In this way, one is able to analyze if a market data set shares the same principal component space as the trained market data in $\mathbf{X_0}$.

6.3 Choosing training data sets

The the training data matrix $\mathbf{X_0}$ should consist of M predefined *normal* market data sets. The goal is to compare the coefficients of $\mathbf{\Lambda}_*$ with that of the ones calculated from the trained data set $\mathbf{\Lambda_0}$, and analyze the behaviour of the coefficients for different types of market data. The training sets are here chosen for the EUR/USD instrument from the same market $\mathcal{M}$ as in section 5.1. A total of $M = 25$ market data sets has been chosen here, where each one of the data sets captures different characteristics of the market data. This is important in order to construct a robust classification method. The selection of training data sets is based on analysing the characteristics of the time-series as well as the resulting biplots for each data set. The PCA needed to obtain the biplots are performed on a slightly different data matrix than the standard market data matrix $\mathbf{X}$ described in 6.1. In this analysis the time variable t has been included along with the bids and offers. This makes it easier to identify trends in the data sets. The reason why the time variable t has not been included in the standard market data matrix $\mathbf{X}$ is due to the fact that the training data sets are selected from market data sets obtained at different times. Including the time variable would therefore induce fictitious characteristics of $\mathbf{X_0}$. In figure 9 and 10, two different training data sets used in $\mathbf{X_0}$ are illustrated.

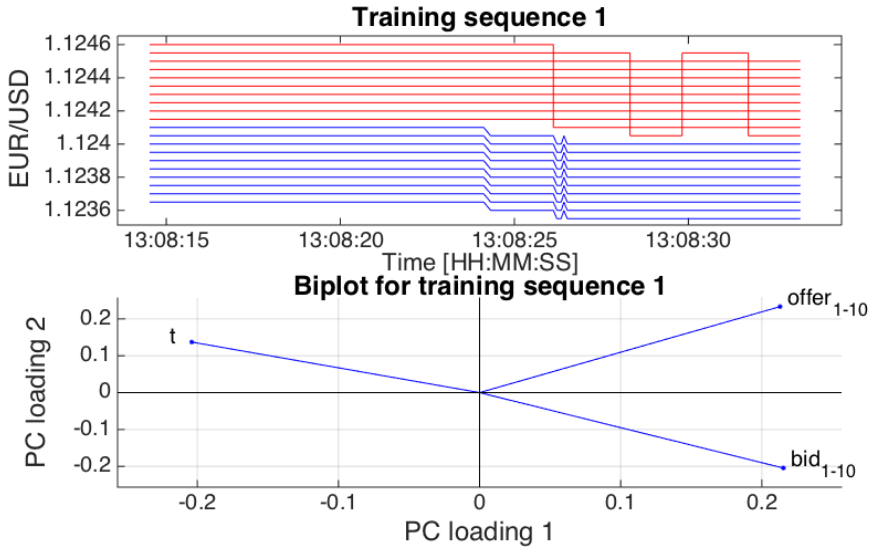

Figure 9: Illustration of the first training set included in the training data matrix $\mathbf{X_0}$. The upper plot shows the bids and offers over time and the lower plot shows the biplot obtained from the data set. By choosing data sets with different characteristics, the *normal* principal component space, spanned by $\mathbf{V_0}$, captures more characteristics of the normal market data.

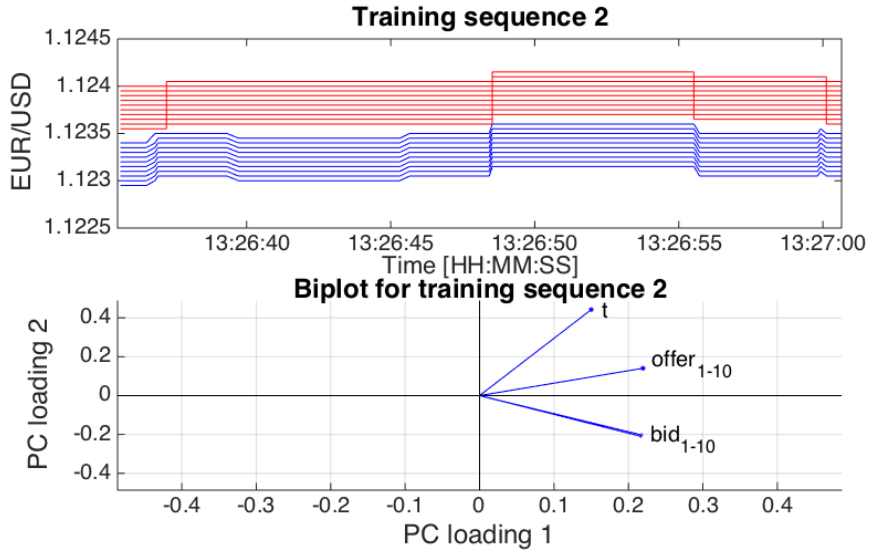

Figure 10: Illustration of the second training set included in the training data matrix $\mathbf{X_0}$.

As can be seen in figure 9 and 10, each training sequence have different

characteristics, especially in terms of their principal component loadings. A common feature for both sequences is that the 10 bids and the 10 offers are highly correlated with each other respectively. This can be seen in the biplots, where all the bid contributions and all the offer contributions on the loading vectors are close to each other respectively (which is how the biplots should be interpreted). The contribution of the time variable t, however, tend to be quite different in the biplot for the different data sets. In figure 9 the contribution of the time variable is antiparallel to the contributions of the bids, indicating a decreasing trend whereas in figure 10 the time variable appears to be quite uncorrelated (near orthogonal) to the contribution of the bid variables. In figure 11, an additional training set used in $\mathbf{X_0}$ is shown.

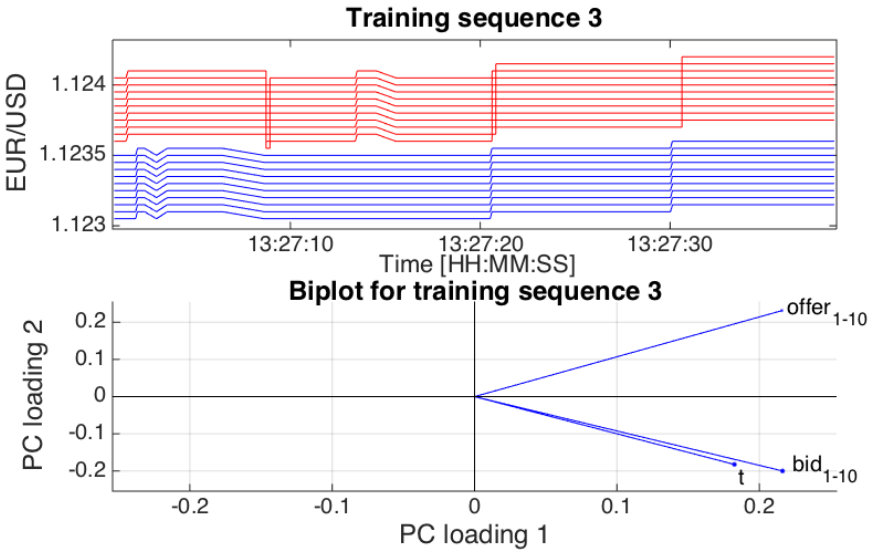

Figure 11: Illustration of the third training set included in the training data matrix $\mathbf{X_0}$.

As can be seen in figure 11, the contribution of the time variable t is parallel to the contribution of the bid variables, indicating an increasing trend in the bid variables. Again, all the bids and all the offers are highly correlated with each other respectively, which can be seen both plots. The fourth sequence, illustrated in figure 12, has been included in $\mathbf{X_0}$ since it captures a sequence where all the bid and offer variables are highly correlated to each other compared to the other sequences.

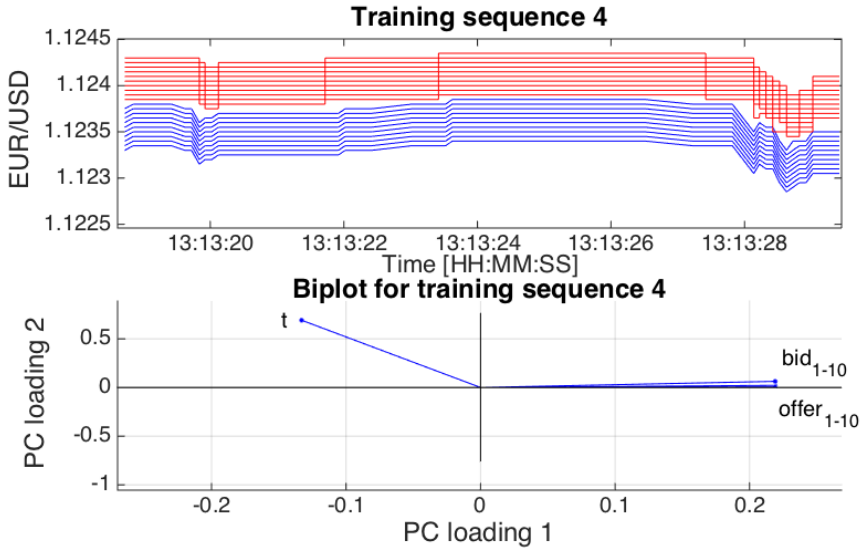

Figure 12: Illustration of the fourth training set included in the training data matrix $\mathbf{X_0}$.

Analysing figure 9 to 12, it becomes clear that the selected market data sets has quite similar characteristics in terms of the correlations between the bids and offers. One should therefore expect that the coefficients matrix $\mathbf{\Lambda_0}$ will have different characteristics compared to $\mathbf{\Lambda}_*$ obtained from abnormal market data.

6.4 Comparing coefficients

In order to compare the coefficient matrices $\mathbf{\Lambda}_*$ and $\mathbf{\Lambda_0}$, the sample mean of the absolute values is calculated for each coefficient, that is, for each row in both matrices respectively. For $\mathbf{\Lambda}_*$, the sample mean of the absolute values for the k^{th} coefficient is calculated by

$$\bar{\lambda}_k^* = \frac{1}{I}\sum_{i=1}^{I} |\lambda_{k,i}^*|, \tag{38}$$

where I is the number of samples in $\mathbf{\Lambda}_*$. For $\mathbf{\Lambda_0}$, the sample mean of the absolute value for the k^{th} coefficient is calculated by

$$\bar{\lambda}_k^0 = \frac{1}{(N \cdot M)} \sum_{i=1}^{(N \cdot M)} |\lambda_{k,i}^0| \tag{39}$$

where $(N \cdot M)$ is the total number of samples in $\mathbf{\Lambda_0}$. Figure 13, 14 and 15 illustrates a comparison between the coefficients $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$ for three different market data sets $\mathbf{X}_*^1$, $\mathbf{X}_*^2$ and $\mathbf{X}_*^3$. All three data sets are for the EUR/USD instrument from market $\mathcal{M}$.

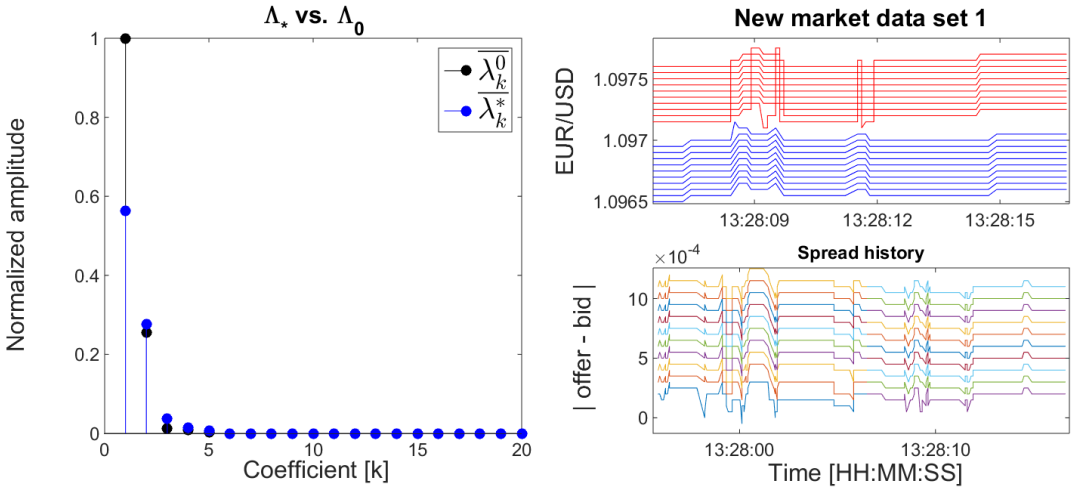

Figure 13: Comparison between $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$ for the first market data set $\mathbf{X}_*^1$, shown in the left-hand plot. The top-right plot shows the quote plot for the new market data set and the bottom-right plot shows the corresponding spread plot.

In figure 13, the plots to the right shows the quote plot for the new market data set $\mathbf{X}_*^1$ (top) and a history of spread levels (bottom). The plot to the left shows the values of $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$, for each coefficient k. As can be seen in this figure, the market data appears to be quite normal with a quite constant spread varying between 0 and $11{\cdot}10^{-4}$. The coefficients shown in the left-hand plot appears to have similar behaviour, that is, having the largest value for the first component $k = 1$, which decreases in a somewhat exponential way for increasing values of k. In figure 14, a new market data set $\mathbf{X}_*^2$, captured shortly after the market data set in figure 13, has been analyzed.

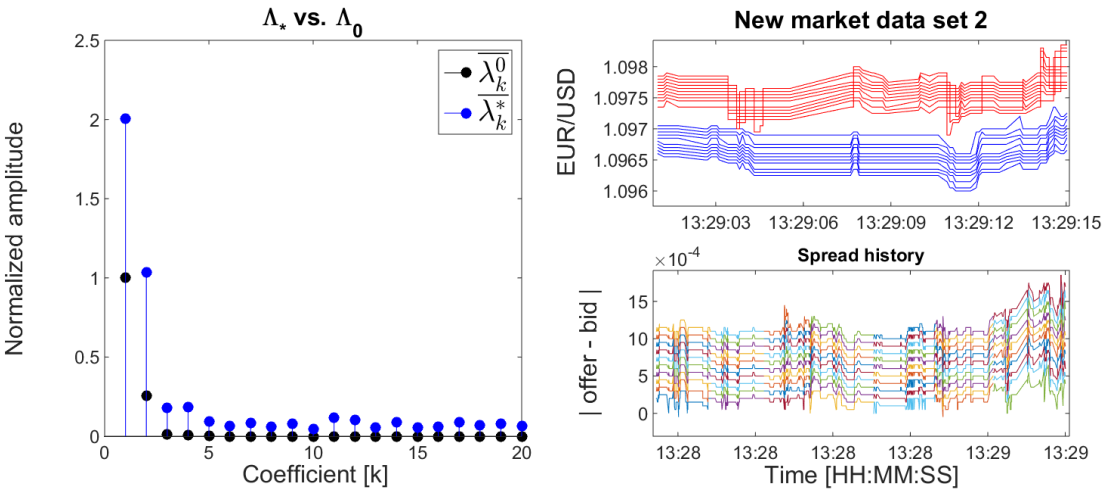

Figure 14: Comparison between $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$ for the second market data set $\mathbf{X}_*^2$.

For the second market data set the bids and offers appears to be more and more uncorrelated with each other respectively over time. This can be seen as the spread starts increasing, varying around 0 and $17 \cdot 10^{-4}$. The coefficients $\bar{\lambda}_k^*$ still has some of that exponentially decreasing characteristics, but the levels of the components for $k \geqslant 5$ are higher compared to that of $\bar{\lambda}_k^0$.

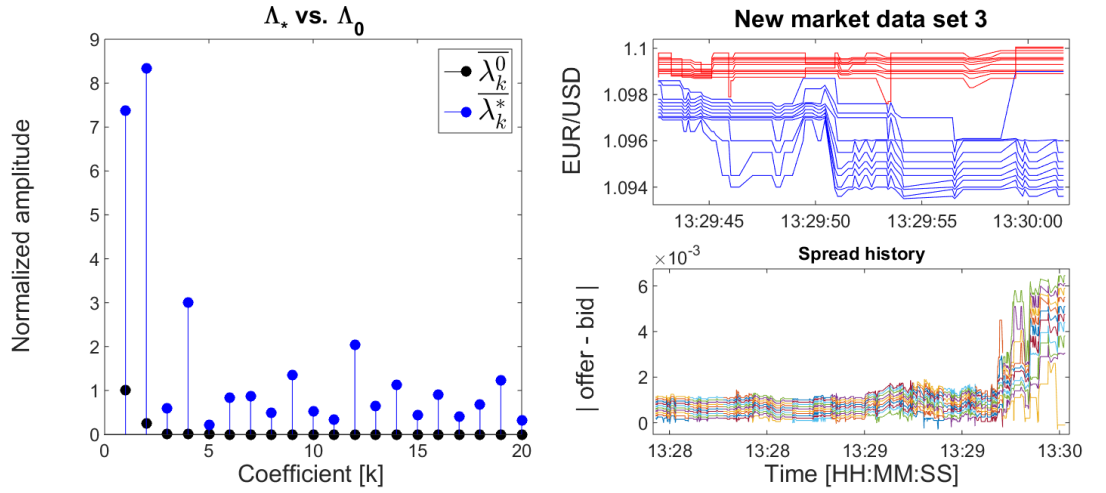

Figure 15: Comparison between $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$ for the third market data set $\mathbf{X}_*^3$.

The third market data set, illustrated in figure 15, is captured shortly after the market data set shown in figure 14. The market data is captured between 13:29:42 and 13:30:01, so the NFP statistics described in section 5.2 has been

announced during this sequence. The bids and offers for this market data set appears to be even less correlated with each other respectively than in previous sets. Again, this can also be shown in the history of the spread levels, where the levels have went from varying around 0 and $11 \cdot 10^{-4}$ to varying around 0 and $60 \cdot 10^{-4}$. For this data set the coefficients $\bar{\lambda}_k^*$ have lost the decaying behaviour and now obtain large values for the majority of the k coefficients.

6.5 Conclusion

The comparison of the coefficients of $\bar{\lambda}_k^*$ and $\bar{\lambda}_k^0$ in 6.4 shows that there exist a quite unique behaviour for the coefficients for a normal market data set compared to an abnormal market data set. Basically, what the comparison says is how much of the energy in $\mathbf{X_0}$ and $\mathbf{X}_*$ that are present in each principal component loading obtained from $\mathbf{X_0}$. For $\bar{\lambda}_k^0$, the first three coefficients ($k = 1, 2, 3$) are dominating, indicating that it should be sufficient to explain the majority of the data in $\mathbf{X_0}$ using only three principal components. The third comparison, illustrated in figure 15, shows that the energy of $\mathbf{X}_*^3$ is distributed over the majority of the principal component loadings obtained from $\mathbf{X_0}$, indicating that the principal component space obtained from the two different matrices are not equal. Since the principal component loadings obtained from the normal and abnormal market data sets respectively are not equal, this means that their covariance matrices, $\mathbf{\Sigma_0}$ and $\mathbf{\Sigma}_*$, are not equal. This strengthens the idea of using the classification methods derived in section 4 for classifying if a market data set is normal or abnormal, since they all assumes that a market data matrix $\mathbf{X}$, originating from a class C_k, can be described by a unique covariance matrix $\mathbf{\Sigma}_k$.

7 Classifying market data

In this section, our goal is to construct a classification method for detecting when a specific market data set is normal or not. In section 6, the principal component spaces for a normal and abnormal market data set for the EUR/USD instrument from market $\mathcal{M}$ were shown to be significantly different. Since the principal component space for a market data set is determined by its covariance (or correlation) matrix $\mathbf{\Sigma}_k$, this suggests that the methods for classifying market data described in section 4.1 should be well suited.

For this particular problem, there are only two classes considered, normal and abnormal. The QDA for two classes would be a proper classification method for this case, but so would the Single-class detector. Since the market data is either regarded as being normal *or* abnormal, only one of these classes is needed to be determined. If the market data set is not classified as normal, it must imply that it is abnormal. The advantage of using the Single-class detector is that it only requires to find the covariance matrix for one of the classes, compared to QDA which requires the covariance matrix for both. Therefore, the Single-class detector will be the preferred method in for this problem. And, since it it easier to characterize data as being normal than being abnormal, the Single-class detector will be used with respect to the covariance matrix for normal market data sets.

7.1 Input vector

Consider a $[p \times 1]$-dimensional random input vector, $\mathbf{x}$, consisting of p bids and offers for one published quote update for instrument $\mathcal{I}$ from market $\mathcal{M}$ (see section 5 for definition of $\mathbf{x}$). The goal is to classify if $\mathbf{x}$ is normal or abnormal for that particular market and instrument. Let class C_0 represent a normal state for $(\mathcal{I}, \mathcal{M})$. Hence, the main interest is to classify if the vector $\mathbf{x}$ originates from class C_0 or *not*.

7.2 Training phase

The classification methods derived in section 4.1 assumes that for each class C_k, a unique mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\mathbf{\Sigma}_k$ can be obtained from the training data matrix $\mathbf{X}_k$. In this example, the classification method will be trained on normal market data sets for the EUR/USD instrument

from market $\mathcal{M}$. The training data matrix $\mathbf{X_0}$ will consist of M hand picked market data sets which has been categorized as normal for the EUR/USD instrument from market $\mathcal{M}$, just as described in section 6.3. Thus, the mean vector and covariance matrix for the trained data can be obtained by

$$\boldsymbol{\mu_0} = \mathrm{E}\big[\mathbf{X_0}\big] \in \mathbb{R}^{p\times 1} \tag{40}$$

and

$$\boldsymbol{\Sigma_0} = \frac{1}{N-1}\mathbf{X_0}\mathbf{X_0}^{\mathrm{T}} \in \mathbb{R}^{p\times p} \tag{41}$$

respectively. It is important to note that p depends on the number of quoted bids and offers. In this example, market $\mathcal{M}$ quotes 10 bids and 10 offers for the EUR/USD instrument. Hence, $p = 20$.

7.3 Feature extraction

7.3.1 Step 1

The analysis of the market data in the previous sections have only been with respect to the correlations and the behaviour of the bid and offer prices relative to each other, and not with respect to their absolute rates. This means that in order for the classification algorithm to be robust, the absolute levels of the trained data matrix $\mathbf{X_0}$ is needed to be removed. Consider the linear transformation

$$\mathbf{Y_0} = \mathbf{L}\mathbf{X_0} \in \mathbb{R}^{p-1\times N}, \tag{42}$$

where $\mathbf{L}$ is a $[p-1\times p]$ anti-diagonal transformation matrix such that

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & \dots & 0 & -1 \\ 0 & 0 & \dots & -1 & 0 \\ \vdots & \vdots & \iddots & \vdots & \vdots \\ 0 & -1 & \dots & 0 & 0 \\ -1 & 0 & \dots & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(p-1)\times p}. \tag{43}$$

In this way, $\mathbf{Y_0}$ obtains the difference between each variable in $\mathbf{X_0}$. That is, the i^{th} column (or observation) of $\mathbf{Y_0}$ is

$$\mathbf{y}_i = \begin{bmatrix} \text{offer}_{10} - \text{offer}_9 \\ \text{offer}_9 - \text{offer}_8 \\ \vdots \\ \text{offer}_2 - \text{offer}_1 \\ \text{offer}_1 - \text{bid}_1 \\ \text{bid}_1 - \text{bid}_2 \\ \vdots \\ \text{bid}_8 - \text{bid}_9 \\ \text{bid}_9 - \text{bid}_{10} \end{bmatrix} \in \mathbb{R}^{(p-1)\times 1}, \tag{44}$$

where the dimension of the new training data matrix has been reduced by 1. In this way, the absolute rates of the bids and offers are removed, and the classification method can be applied with respect to the new random vector $\mathbf{y}$ defined as

$$\mathbf{y} = \mathbf{L}\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{0}}^{\boldsymbol{y}}, \boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}),$$

where $\boldsymbol{\mu}_{\mathbf{0}}^{\boldsymbol{y}}$ and $\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}$ are the new mean vector and covariance matrix respectively obtained from $\mathbf{Y_0}$. Note that since $\mathbf{y}$ is a linear transformation of $\mathbf{x}$, it will be Gaussian distributed with a new mean vector and covariance matrix.

7.3.2 Step 2

As discussed in section 4.3, the QDA methods can only be regarded if $\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}$ has full rank, that is, if $\text{rank}(\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}) = p - 1$. In the above numerical example, it turns out that

- $\text{rank}\big(\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}\big) = 16 < p - 1$

The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{0}}^{\mathbf{y}}$ has not full rank since $p-1 = 20-1 = 19$. This is probably due to the fact that the variables in $\mathbf{Y_0}$ (bids and offers) are highly correlated for the normal market data sets, resulting in a low rank covariance matrix. However, as shown in section 4.3, the covariance matrix of $\mathbf{Y_0}$ will have full rank on a subspace. By performing the linear transformation

$$\mathbf{Z_0} = \mathbf{V}^{\mathrm{T}}\mathbf{Y_0}, \tag{45}$$

and choosing an appropriate transformation matrix $\mathbf{V} \in \mathbb{R}^{p-1\times q}$, the covariance matrix of $\mathbf{Z_0}$ will obtain full rank. As described in section 4.3, the

column vectors of $\mathbf{V}$ must be orthonormal and must belong to the linear space spanned by the columns of $\mathbf{\Sigma_0^y}$. The principal component loadings obtained from $\mathbf{Y_0}$ fulfills these conditions of $\mathbf{V}$, and will therefore be the selected transformation matrix. Thus,

$$\hat{\mathbf{Z}}_\mathbf{0} = \hat{\mathbf{V}}_\mathbf{0}^\mathrm{T}\mathbf{Y_0} \in \mathbb{R}^{q \times N}, \tag{46}$$

where $\hat{\mathbf{V}}_\mathbf{0}$ consists of the first q principal component loadings obtained from $\mathbf{Y_0}$. This transformation can be seen as transforming the data in $\mathbf{Y_0}$ to its corresponding principal component space spanned by the q most significant principal component loadings. In this way, by choosing a proper value of $q < p - 1$, the covariance matrix of $\hat{\mathbf{Z}}_\mathbf{0}$ will obtain full rank, and the classification methods can be applied with respect to the new random vector defined by

$$\hat{\mathbf{z}} = \hat{\mathbf{V}}_\mathbf{0}^\mathrm{T}\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{0}^{\hat{\mathbf{z}}}, \mathbf{\Sigma}_\mathbf{0}^{\hat{\mathbf{z}}}),$$

where $\boldsymbol{\mu}_\mathbf{0}^{\hat{\mathbf{z}}}$ and $\mathbf{\Sigma}_\mathbf{0}^{\hat{\mathbf{z}}}$ are the new mean vector and covariance matrix respectively obtained from $\hat{\mathbf{Z}}_\mathbf{0}$. The question now is whether the principal component loadings should be determined from the covariance or correlation matrix of $\mathbf{Y_0}$. As discussed in 3.4, the covariance matrix should be suitable in this particular example, since all the market data are obtained for the same instrument $\mathcal{I}$ from market $\mathcal{M}$. Hence, the units of all the p variables in the original vector $\mathbf{x}$ are the same. However, since the goal is to derive a robust classification algorithm, it should be possible to train the classification method on market data where $\mathbf{x}$ may consist of market data not only for instrument $\mathcal{I}$, but, for example, for both instrument $\mathcal{I}$ and $\mathcal{J}$ simultaneously. In such case, the units of the different variables in $\mathbf{x}$ will not be the same, and the obtained principal components will then dominate for the variables whose variances are largest. Therefore, the correlation matrix $\mathbf{R_0}$ will be used to obtain the principal component loading matrix $\mathbf{V_0}$.

Table 1 illustrates a summary of the rank, the determinant and the condition numbers of $\mathbf{\Sigma}_\mathbf{0}^{\hat{\mathbf{z}}}$ for different number of principal components q, along with the amount of variance in the data explained.

Table 1: Figures of merit for $\mathbf{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}}$.

q	PC variance	$\text{rank}\left(\mathbf{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}}\right)$	$\det\left(\mathbf{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}}\right)$	$\text{cond}\left(\mathbf{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}}\right)$
13	95%	13	$4.28 \cdot 10^{-147}$	$1.17 \cdot 10^{7}$
10	85%	10	$1.22 \cdot 10^{-111}$	$1.39 \cdot 10^{6}$
8	75%	8	$7.84 \cdot 10^{-88}$	$3.29 \cdot 10^{3}$
6	65%	6	$2.10 \cdot 10^{-65}$	$2.73 \cdot 10^{3}$
4	50%	4	$2.50 \cdot 10^{-44}$	98.30

As can be seen in table 1, the higher the number of principal component loadings included (q), the higher the amount of variance in the original data captured in the transformation. For $q = 13$, as much as 95% of the variance in the data is captured. The condition number has decreased by a factor of 10^{16} compared to that of $\mathbf{\Sigma}_{\mathbf{0}}^{\mathbf{y}}$. The fewer the number of principal component loadings selected, the smaller the condition number tends to get. For $q = 8$, as much as 75% of the variance in the data is captured. The condition number, $3.29 \cdot 10^3$, can be regarded as quite large compared to the condition number obtained for $q = 4$. However, when performing PCA, a sensible number of principal components to include is the number such that 70% to 90% of the variance is captured [5]. In this case, choosing $q = 4$ implies throwing away 50% of the information in the original data. Therefore, even though it would be preferable to choose $q = 4$ due to the small condition number, the number of principal component loadings will here be chosen to be $q = 8$. As a rule-of-thumb, the covariance matrix will be regarded as ill conditioned if $\log\left[\text{cond}\left(\mathbf{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}}\right)\right]$ is larger than the precision of the bid and offer values [22]. For the EUR/USD instrument, the precision of the data is 6 digits for market $\mathcal{M}$, yielding a maximum condition number of approximately $1 \cdot 10^6$. Hence, the covariance matrix for $q = 8$ will be regarded as being well conditioned, and therefore, the covariance matrix for $\hat{\mathbf{z}}$ will be numerically stable.

When analysing the resulting determinant for the different values of q, it becomes clear that the determinant becomes extremely small for all value of q. Although the determinant is close to zero, the covariance matrices for $q \leqslant 8$ are not treated as being ill conditioned. Therefore, the matrix is not close to being singular, and the classification methods will therefore be applicable on the transformed data. In fact, the determinant of a matrix can be arbitrarily close to zero without conveying information about singularity [10].

As seen in this section, two necessary steps in the feature extraction stage are necessary in order to make the Single-class detector method robust for the market data sets. The first step involves removing the dependence of the absolute rates of the bids and offers, yielding $\mathbf{y} \in \mathbb{R}^{p-1\times 1}$. The second step involves performing PCA on the training data matrix $\mathbf{Y_0}$ in order to handle the multicollinearity problem, yielding $\hat{\mathbf{z}} \in \mathbb{R}^{q\times 1}$.

7.4 Classification

The goal of the feature extraction step is to remove the effect of the absolute rates of the bids and offers, and to remove the effects from the multicollinear characteristics of the data. Once this step has been performed, the classification step of the classification method described in section 4.1 can be applied. Here, the Single-class detector will be used for classification, and the discriminant function will be

$$\mathcal{T}(\hat{\mathbf{z}}) = (\hat{\mathbf{z}} - \boldsymbol{\mu}_{\mathbf{0}}^{\hat{\mathbf{z}}})^{\mathrm{T}}(\boldsymbol{\Sigma}_{\mathbf{0}}^{\hat{\mathbf{z}}})^{-1}(\hat{\mathbf{z}} - \boldsymbol{\mu}_{\mathbf{0}}^{\hat{\mathbf{z}}}) \lesseqgtr \mathcal{T}_{critical}. \tag{47}$$

The threshold, $\mathcal{T}_{critical}$, is determined by the inverse cdf for the chi-squared distribution for a significance level $\alpha = 0.05$ and $q = 8$, since it is the resulting dimension of $\mathbf{y}$, yielding

$$\mathcal{T}_{critical} = Q_8(1 - 0.05) = 15.91.$$

The decision rule for the algorithm will then be as described in algorithm 1.

Algorithm 1 Decision rule

```
procedure DecisionRule
    if T(ẑ) ⩽ T_critical then
        return 0
    else
        return 1
    end if
end procedure
```

The decision part of the algorithm thereby returns 0 if the input vector $\mathbf{x}$ is

normal and 1 if it is abnormal. For a pseudocode of the complete general classification algorithm, refer to section 8.2.

7.5 Test run

To ensure that the classification method provided in this section works, it needs to be tested on *new* market data. To ensure robustness, the new test market data must not have been included in the training data matrix $\mathbf{X_0}$. The classification method will here be simulated in MATLAB on new input data for the EUR/USD instrument from market $\mathcal{M}$. The MATLAB script is constructed such that it performs all the pre-processing of the data as described in the previous section. For this particular test, the following user-defined parameters has been set:

- $\text{PC}_{\text{variance}} = 0.75$
- $\alpha = 0.05$

Figure 16 shows the output of the decision rule of the classification method for a new market data set $\mathbf{X}_*$. The market data set consists of 25 quote updates (or samples) of $\mathbf{x}$, corresponding to around 2.5 seconds, where each individual quote has been classified.

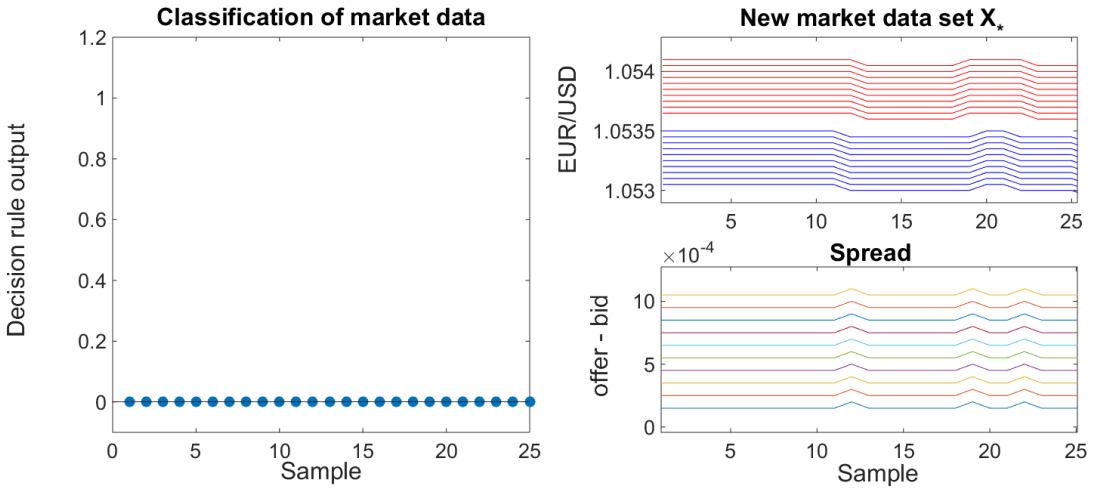

Figure 16: Classification of 25 quote updates for a new market data set $\mathbf{X}_*$. The top-right plot shows the quote plot and the bottom-right plot shows the corresponding spread plot. The left plot shows the output of the decision rule for each individual quote update $\mathbf{x}_i$. For this market data set all updates are classified as normal.

Analysing the quote data in figure 16, one can see that all bids and all offers are highly correlated with each other respectively. And, a change in an offer quote is followed by a change in the corresponding bid quote, with a slight delay. These characteristics of the data can also be seen in the spread plot, where the distance between all spread levels are equal for all bid-offer pairs. The classification method classifies all quotes as being normal, or, in other words, sharing the same distribution with that of $\mathbf{X_0}$. This is seen in the left-hand plot. The result should not be surprising, since the training data sets selected in section 6.3 were chosen due to obtaining the above explained characteristics.

In figure 17, the classification of the quote updates for a new market data set with different characteristics is considered. Again, the data set consists of 25 quotes, but for this data set the bids and offers are not as correlated as for the previous market data set.

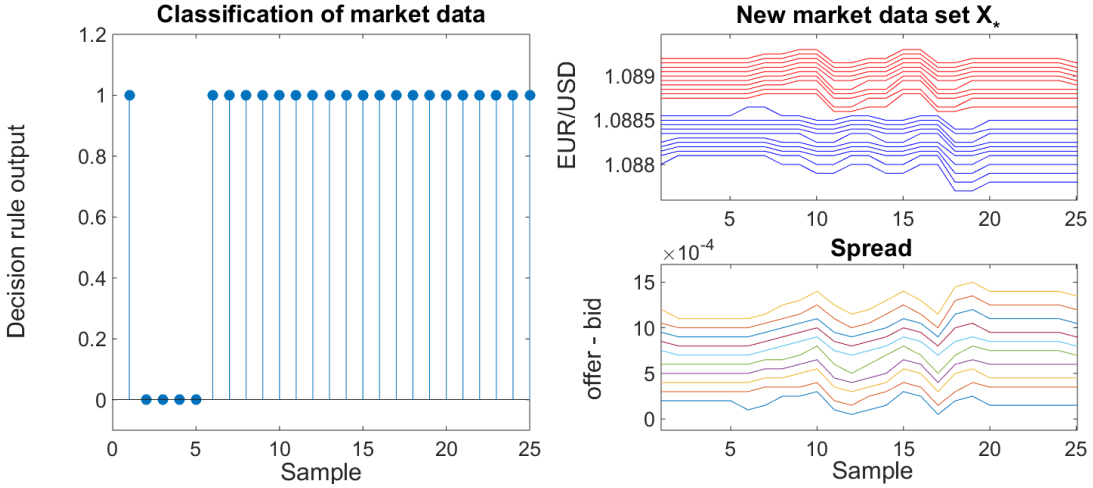

Figure 17: Classification of 25 quote updates for a new market data set $\mathbf{X}_*$. Quote update $2-5$ are the only ones being classified as normal, since the spread levels are equidistant for these particular samples.

As seen in figure 17, all but the 2^{nd} to 5^{th} quote updates are being classified as abnormal with respect to $\mathbf{X_0}$. For all of these abnormal samples, the spread levels are *not* equidistant, which again is abnormal compared to the trained data sets. A sample quote where the spread levels are not equidistant simply means, for this particular training data matrix, that not all of the bids and offers are highly correlated with each other respectively, or, that the spread between any of the bid-offer pairs are different from that of $\mathbf{X_0}$.

These relationships can easily be seen in the top-right quote plot and the bottom-right spread plot for the particular samples.

The classification of an additional market data set is illustrated in figure 18. For this particular data set, all quote updates are regarded as abnormal with respect to $\mathbf{X_0}$.

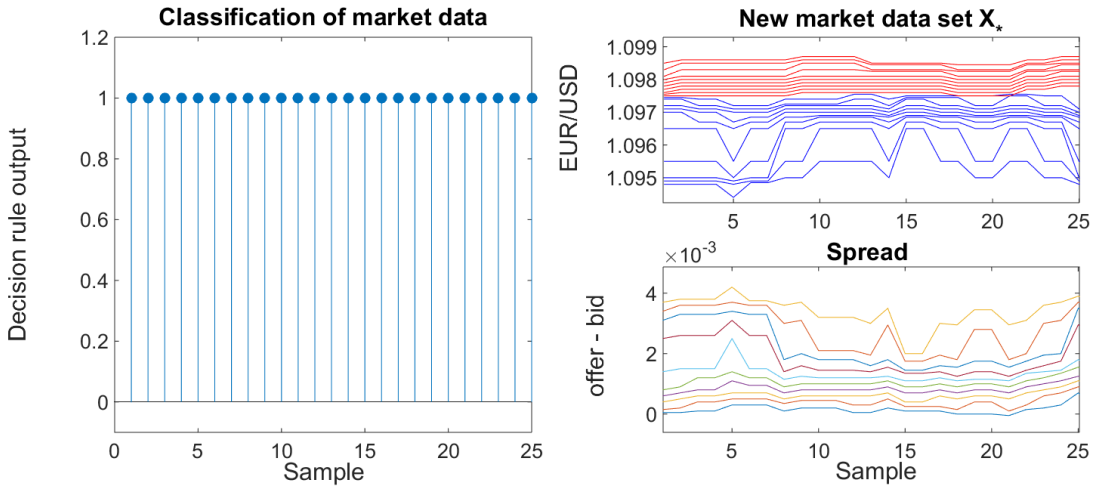

Figure 18: Classification of 25 quote updates for a new market data set $\mathbf{X}_*$. For this data set, all quote updates are classified as being abnormal. All the bids and offers are either not highly correlated, or the spread levels for the bid-offer pairs are not equidistant.

When analysing the offers for the quote data in figure 18, it becomes clear that for the majority of the quote updates, the offers are not highly correlated with each other, and the bids are not highly correlated with each other. And when they are correlated, the spread of the bid-offer pairs are not equidistant. This applies for all samples, leading to the decision rule output being 1 for all of them.

7.6 Conclusion

As seen in figure 16 to 18, the classification method works as desired. That is, the classification method successfully discriminates if a new quote update $\mathbf{x}$ is sharing the distribution of $\mathbf{X_0}$ or not. Therefore, it should be stressed out that depending on what market data sets are included in the training data matrix, the decision rule outputs 0 or 1 depending on the characteristics of those particular data sets. In this report, the training data matrix has

been constructed by hand-picking market data sets which appeared to be normal compared to abnormal ones. These normal data sets were shown to have highly correlated characteristics between the bids and offers respectively. Furthermore, the spread levels for the bid-offer pairs were equidistant for those particular data sets. Therefore, the classification method provided here will detect whenever a quote update does not fulfill these properties. In general, however, a training data matrix can be constructed using any particular set of market data, depending on what is to be classified. An attractive feature is that the classification steps of the classification method will be the same – making the method very versatile. Therefore, a suggestion of a general classification algorithm, where the user can set how and when training data is to be collected, is provided in the next section.

8 The classification algorithm

As seen in chapter 7, the training phase, feature extraction step and the classification step are all vital in order to be able to implement the Single-class detector method described in 4.2. The classification algorithm will therefore be constructed such that it subsequently performs the above explained steps.

8.1 Block diagram

The algorithm is divided into two main sequences; a *training phase*, and a *real-time classification phase*. A block diagram of the main steps in the algorithm is shown in figure 19. In the training phase, market training data sets are first obtained, forming the training data matrix $\mathbf{X_0}$. How and when training data sets should be collected can be configured by a predefined *training scheme* parameter. The absolute rate dependencies of the data is then removed, followed by obtaining the principal component loading matrix $\mathbf{V_0}$ for the transformed training data $\mathbf{Y_0}$. In the PCA stage, the input parameter $\mathrm{PC}_{\mathrm{tolerance}}$ must be set by the user and must be chosen such that the conditions of the covariance matrix $\mathbf{\Sigma_0^{\hat{z}}}$ are met (see section 7.3.2). The final step in the training phase is then to calculate the sample mean vector and sample covariance matrix for the transformed data $\hat{\mathbf{Z}}_\mathbf{0}$.

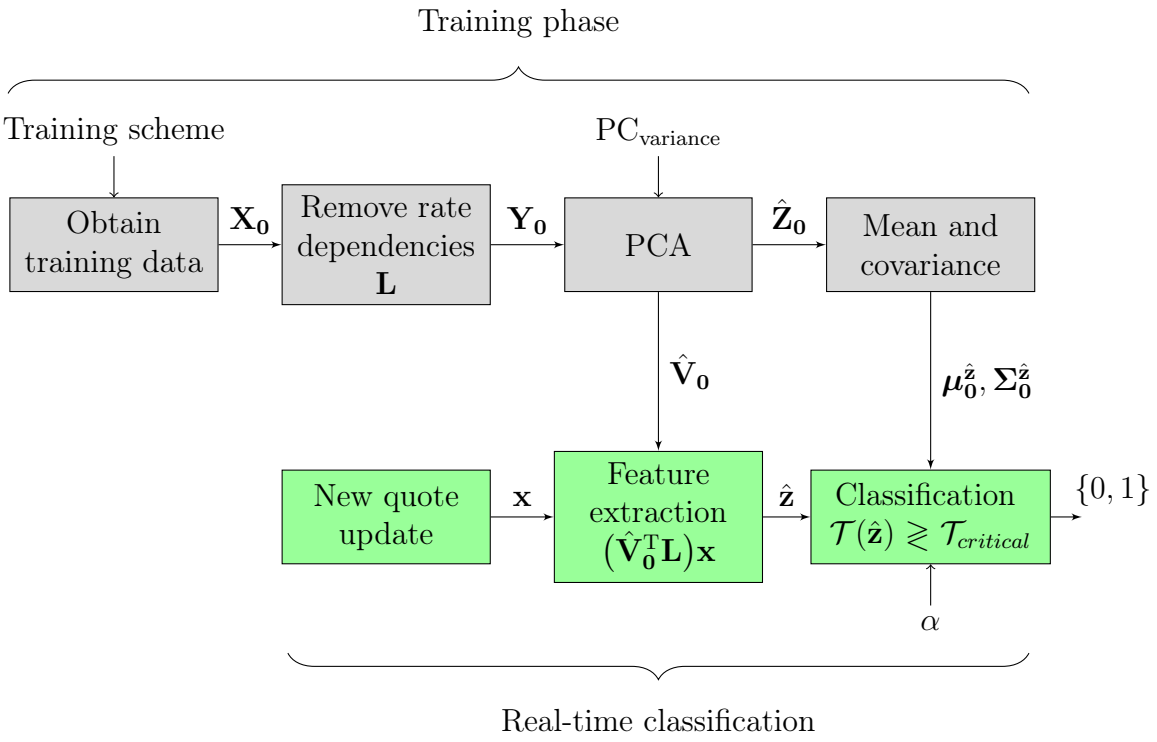

Figure 19: A block diagram of the main steps in the classification algorithm. The system is divided into a training phase (grey blocks) and a real-time classification stage (green blocks). The training phase trains the classification method based on selected market data sets and is performed offline. The real-time classification step classifies a new quote update $\mathbf{x}$ online and returns 0 if the the quote is regarded as normal and 1 if it is regarded as abnormal with respect to the training data. Training scheme, $\mathrm{PC}_{\mathrm{variance}}$ and α are input parameters set by the user.

The real-time classification phase can be seen as an infinite loop, which, for a new quote update $\mathbf{x}$ classifies if it is normal or abnormal with respect to the training data matrix, that is, if it shares the distribution with that of $\mathbf{X_0}$ or not. The classification phase consists of first performing feature extraction of $\mathbf{x}$ by using the offline calculated combined transformation matrix $\hat{\mathbf{V}}_\mathbf{0}^{\mathrm{T}}\mathbf{L}$. The second, and final step, of the classification phase is to classify the new data matrix $\hat{\mathbf{z}}$ using the Single-class detector for a given significance level α. The system then outputs 0, if $\mathbf{x}$ is regarded as normal, and 1 if it is regarded as abnormal.

8.2 Pseudo code

The block diagram illustrated in figure 19 shows an overview of the main steps in the classification algorithm. In this section pseudo code for the training phase and the real-time classification phase are provided. The algorithm has been designed such that it is independent of how the training data sets are obtained, and can therefore be obtained in any desired way. Therefore, one can easily construct separate algorithms for controlling the collection of the training data sets automatically. The idea is that the user selects a training scheme which is linked to a unique timer, for triggering the training phase, and a unique way of constructing the training data matrix $\mathbf{X_0}$. How the training data matrix should be constructed for a particular training scheme is then predefined in the training phase. In section 8.3, examples of suitable training schemes are discussed. The triggering of the timer is controlled by an external process running in the background and its corresponding pseudo code is illustrated in algorithm 2.

Algorithm 2 Training trigger algorithm

```
1: procedure TrainingTrigger(training scheme)
2:     Set training flag = 0
3:     Start training timer
4:     Set expiration of timer for training scheme
5:
6:     %Wait for timer to expire:
7:     while training timer expired = false do
8:         nothing
9:     end while
10:
11:    Set training flag = 1
12:    return training flag
13: end procedure
```

The pseudo code for the training phase is illustrated in algorithm 3.

Algorithm 3 Training phase algorithm

1: **procedure** TRAININGPHASE(training scheme, $\mathrm{PC}_{\mathrm{variance}}$)
2:
3: **procedure** INITIALIZATION
4: `%Construct L:`
5: p = number bids and offers
6: $\mathrm{I_p} = [p - 1 \times p]$ anti-diagonal identity matrix
7: $\mathrm{L} = (-1) \cdot \mathrm{I_p}$
8: **return** L
9: **end procedure**
10:
11: **procedure** OBTAINTRAININGDATA(training scheme)
12: Collect training data
13: Construct X_0
14: **return** X_0
15: **end procedure**
16:
17: **procedure** REMOVERATEREPENDENCIES(L, X_0)
18: $\mathrm{Y}_0 = \mathrm{LX}_0$
19: **return** Y_0
20: **end procedure**
21:
22: **procedure** PCA($\mathrm{Y}_0, \mathrm{PC}_{\mathrm{variance}}$)
23: Obtain sample correlation matrix of $\mathrm{Y}_0 \rightarrow \mathrm{R}_0^{\mathrm{y}}$
24: Compute SVD of $\mathrm{R}_0^{\mathrm{y}} \rightarrow (\mathrm{V}_0, \lambda_0)$
25: Obtain diagonal elements of $\lambda_0 \rightarrow \lambda_{\mathrm{diag}}$
26:
27: `%Total variance of all PCs:`
28: $\lambda_{\mathrm{tot}} = 0$
29: **for** $i = 1$ to size of λ_{diag}, **do**

30: $\lambda_{\text{tot}} = \lambda_{\text{tot}} + \lambda_{\text{diag}}(i)$
31: **end for**
32:
33: `%Proportion of variance for each eigenvalue:`
34: prop = $[p - 1 \times 1]$ zero vector
35: **for** $j = 1$ to size of λ_{diag}, **do**
36: $\text{prop}(i) = \lambda_{diag}(i)/\lambda_{\text{tot}}$
37: **end for**
38:
39: `%Choose` q `such that` $\text{PC}_{\text{variance}}$ `is captured:`
40: cumsum = 0
41: **for** $k = 1$ to size of prop, **do**
42: $\text{cumsum} = \text{cumsum} + \text{prop}(k)$
43: **if** cumsum $\geqslant \text{PC}_{\text{variance}}$ **then**
44: $q = k$
45: **return**
46: **end if**
47: **end for**
48:
49: `%Project` Y_0 `on the` q `first PC loadings:`
50: Select first q columns of $\text{V}_0 \rightarrow \hat{\text{V}}_0$
51: $\hat{\text{Z}}_0 = \hat{\text{V}}_0^{\text{T}}\text{Y}_0$
52:
53: **return** $\hat{\text{V}}_0$ and $\hat{\text{Z}}_0$
54:
55: **end procedure**
56:
57: **procedure** MeanAndCovariance(Z_0)
58: $\mu_0^{\hat{z}} = \text{mean}(\hat{\text{Z}}_0)$
59: $\Sigma_0^{\hat{z}} = \text{cov}(\hat{\text{Z}}_0)$
60: **return** $\mu_0^{\hat{z}}$ and $\Sigma_0^{\hat{z}}$

61: **end procedure**

62:

63: `%Execute real-time classification phase:`

64: Execute RealTimeClassification($\hat{\mathrm{V}}_0$, $\mu_0^{\hat{\mathrm{z}}}$, $\Sigma_0^{\hat{\mathrm{z}}}, \alpha$)

65:

66: **end procedure**

Algorithm 4 Real-time classification algorithm

procedure RealTimeClassification($\hat{\mathrm{V}}_0$, $\mu_0^{\hat{z}}$, $\Sigma_0^{\hat{z}}$, α)

  procedure Initialization
    Calculate $\hat{\mathrm{V}}_0^{\mathrm{T}}\mathrm{L}$
    Obtain $\mathcal{T}_{critical}(1-\alpha)$
    return $\hat{\mathrm{V}}_0^{\mathrm{T}}\mathrm{L}$, $\mathcal{T}_{critical}$
  end procedure

  `%Infinite while loop:`
  procedure InfiniteLoop
    while true **do**

      `%Check training flag:`
      if training flag = 1 **then**
        Execute TrainingPhase(training scheme, $\mathrm{PC}_{\mathrm{variance}}$)
      end if

      `%Wait for new quote data x to arrive:`
      procedure NewQuoteUpdate
        while New quote is false **do**
          nothing
        end while
        return x
      end procedure

      procedure FeatureExtraction($\hat{\mathrm{V}}_0^{\mathrm{T}}\mathrm{L}$, x)
        $\hat{\mathrm{z}} = (\hat{\mathrm{V}}_0^{\mathrm{T}}\mathrm{L})\mathrm{x}$
        return $\hat{\mathrm{z}}$
      end procedure

```
30:
31:             procedure CLASSIFICATION(T_critical, ẑ)
32:                 Calculate T(ẑ)
33:
34:                 procedure DECISIONRULE(T(ẑ), T_critical)
35:                     if T(ẑ) ⩽ T_critical then
36:                         return 0
37:                     else
38:                         return 1
39:                     end if
40:                 end procedure
41:
42:                 return output from DECISIONRULE
43:
44:             end procedure
45:         end while
46:     end procedure
47: end procedure
```

8.3 Training schemes

As described in the previous section, the idea with the training scheme is to control whenever training data is to be collected. The training data matrix $\mathbf{X_0}$ used in the previous sections where obtained by hand-picking market data sets which where regarded as being normal for the EUR/USD instrument from market $\mathcal{M}$. However, consider the case where the definition of normal market data changes with time. For example, it may be the case that the definition of normal market data is different from one week to another depending on the economy in the world. In such case, the classification method must be trained every week, in order for it to be robust. As an example, if market data where assumed to be normal during 10:00:00 - 10:15:00 every Monday, the procedure of obtaining training data defined in algorithm 2 would then be as shown in algorithm 5.

Algorithm 5 Training scheme - weekly

```
1: procedure ObtainTrainingData(weekly)
2:     if day = Monday then
3:         for time = 10:00:00 - 10:15:00, do
4:             Collect quote data x_i → X_0
5:         end for
6:     end if
7:     return X_0
8: end procedure
```

In this way the training data matrix $\mathbf{X_0}$ would been updated every Monday at 10:15:00, and the classification algorithm would then execute the real-time classification phase, classifying new quote updates, until 10:00:00 next Monday. The process would then be repeated.

9 Discussion

The goal of the project was to investigate how valuable information from the market data could be extracted in order to construct a classification method and algorithm for detecting deviations in the market data. The project resulted in a successful construction of a classification method, which is able to classify if a new quote update $\mathbf{x}$, for the EUR/USD instrument and market $\mathcal{M}$, is normal or abnormal with respect to the training data in $\mathbf{X_0}$. The project also resulted in the development of a classification algorithm, which enables the classification method to be implemented on a system for real-time classification of quote data along with automated collection of training data.

To make the project plausible, the work has been focused on the EUR/USD instrument for a unique market $\mathcal{M}$. By using PCA, the market data has been shown to obtain unique characteristics for two particular states of the market data, normal and abnormal. The bid and offer prices (or variables) in the market data regarded as being normal has been shown to possess high correlation characteristics, whereas the opposite has been shown to be true for the market data regarded as abnormal. The comparison of the principal component subspaces for normal and abnormal market sets shows that the two different states originate from different covariance structures. This leads

to the assumption that data originating from the two different classes should be able to be discriminated using QDA. In this project, the Single-class detector has been used rather than QDA. Classifying if a quote update $\mathbf{x}$ is normal or abnormal can be performed using QDA with two classes, C_0 and C_1. However, this would not only require to collect normal training data, but also to collect abnormal training data, forming an additional training data matrix $\mathbf{X_1}$. Since there are only two classes regarded, normal and abnormal, the Single-class detector will be more robust. Why? Since the market data is either normal or abnormal, classifying only one of them is needed. And, since the characteristics of the normal market data is more well-defined than the characteristics of the abnormal market data, the classification method will be more robust if it is trained on the normal case. Therefore, the Single-class detector trained on normal market data has been considered in this report. However, if market data is to be categorized into more than two classes, QDA would be the suggested method.

To be able to implement any of the suggested classification methods in this report, a vital pre-processing step of the data must is required. Due to the multicollinear characteristics in the market data, the covariance matrices for the market data will always obtain low-rank properties. It has been shown that by transforming the data on to its principal component space, spanned by the dominating loading vectors, the covariance matrices for the new transformed data will obtain full rank. This is an important result and is a vital step in order for the QDA and Single-class detector to work properly. Since the classification method is trained on the principal components of the trained market data $\mathbf{X_0}$, it can be seen as classifying the principal components of the data. The suggested classification algorithm can therefore be seen as storing the fundamental principal components of the normal market data, and for a new quote update $\mathbf{x}$, compares how much energy of $\mathbf{x}$ that is in the trained principal component space. If it is not enough, then it is not regarded as normal.

9.1 Future developments

The classification method developed in this project has only been with respect to the EUR/USD instrument, from market $\mathcal{M}$. The classification method is therefore only able to indicate when that particular instrument, for that particular market, is behaving unnatural compared to the trained data. Now, there is a good reason for believing that there is a strong connection between the following currency pairs from a market $\mathcal{M}$:

- EUR/USD
- EUR/SEK
- USD/SEK

A change in the data for any of the above mentioned currency pairs should affect the others ones and vice versa. Therefore, a suggestion of a future project is to investigate how these currency pairs are correlated to each other and if there exist any normal behaviour in the joined data sets. If this would be the case, then the classification method derived in this report could be implemented with respect to the joint random vector $\mathbf{x}_{\text{joint}}$ defined by

$$\mathbf{x}_{joint}^{\text{T}} = \left[\ \mathbf{x}_{\text{EUR/USD}}^{\text{T}} \ \mathbf{x}_{\text{EUR/SEK}}^{\text{T}} \ \mathbf{x}_{\text{USD/SEK}}^{\text{T}} \ \right],$$

such that

$$\mathbf{x}_{\text{EUR/USD}}^{\text{T}} = \big[\text{bid}_1, \ldots, \text{bid}_{10} \mid \text{offer}_1, \ldots, \text{offer}_{10}\big].$$

is the quote update for the EUR/USD instrument. In this way, the classification method can be used for detecting if a joint quote update is normal or abnormal, that is, if the market data for the EUR/USD, EUR/SEK and USD/SEK is behaving normal or not. Such a pattern recognition algorithm could be used as an input parameter in deciding positions in a specific trade.

Another suggestion for further development of the classification algorithm is to extend it to K classes. It may be the case that the market data can be categorized into more than two classes. For example, one could consider forming three classes, C_1, C_2 and C_3, where the first class indicates increasing trends in the market data, the second indicates decreasing trends and the third one indicates random dispersion of the data. A suitable classification method would then be the QDA for three classes, where the training phase then would require to form three training data matrices $\mathbf{X_1}$, $\mathbf{X_2}$ and $\mathbf{X_3}$ for each class, yielding three covariance matrices $\mathbf{\Sigma_1}$, $\mathbf{\Sigma_2}$ and $\mathbf{\Sigma_3}$. Again, since the characteristics of the market data does have multicollinear characteristics, transforming the trained data sets into their principal component spaces respectively is required. Hence, the extended classification method can be seen as storing the principal component loadings for C_1, C_2 and C_3 in a filter bank, where new quote updates $\mathbf{x}$ can be classified as either having an increasing trend, a random trend, or simply having no structure at all.

10 Appendix

10.1 Theory of Matrices

10.1.1 Rank of a Matrix

A matrix may be considered as a set of column vectors written in a particular order. The rank of a matrix is defined as the number of independent columns it contains. A $[p \times p]$ matrix $\mathbf{\Sigma}$ is said to have full rank if the number of independent columns in $\mathbf{\Sigma}$ is p, that is, $\text{rank}(\mathbf{\Sigma}) = p$ [6].

10.1.2 Inverse of a Matrix

A square matrix $\mathbf{\Sigma}$ of order $[p \times p]$ is said to be non-singular if its rank is p. In such case there exists a unique matrix $\mathbf{\Sigma}^{-1}$, known as the inverse of $\mathbf{\Sigma}$ such that

$$\mathbf{\Sigma}\mathbf{\Sigma}^{-1} = \mathbf{\Sigma}^{-1}\mathbf{\Sigma} = \mathbf{I} \ ,$$

where $\mathbf{I}$ is a $[p \times p]$ identity matrix [6].

10.2 Derivations

10.2.1 Derivation of optimal classifier for QDA

$$\delta_k(\mathbf{x}) = \ln \left[\tfrac{1}{(2\pi)^{p/2}} \tfrac{1}{|\mathbf{\Sigma}_k|^{1/2}} \exp \Big\{ -\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k) \Big\} p(C_k) \right]$$

$$= \ln \left[\tfrac{1}{(2\pi)^{p/2}} \tfrac{1}{|\mathbf{\Sigma}_k|^{1/2}} \right] + \ln \Big[p(C_k) \Big] - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)$$

10.2.2 Derivation of QDA for two classes

$$\ln(R) = \ln \left[\frac{\frac{1}{|\mathbf{\Sigma}_0|^{1/2}} \exp \left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}} \mathbf{\Sigma}_0^{-1} (\mathbf{x}-\boldsymbol{\mu}_0) \right\}}{\frac{1}{|\mathbf{\Sigma}_1|^{1/2}} \exp \left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}} \mathbf{\Sigma}_1^{-1} (\mathbf{x}-\boldsymbol{\mu}_1) \right\}} \cdot \frac{p(C_0)}{p(C_1)} \right]$$

$$= \ln\left[\frac{1}{|\Sigma_0|^{1/2}}\right] - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)$$
$$- \ln\left[\frac{1}{|\Sigma_1|^{1/2}}\right] + \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \ln\left[\frac{p(C_0)}{p(C_1)}\right]$$

$$= \tfrac{1}{2}\ln\left[\frac{|\Sigma_1|}{|\Sigma_0|}\right] + \ln\left[\frac{p(C_0)}{p(C_1)}\right] - \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^{\mathrm{T}}\boldsymbol{\Sigma}_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)$$
$$+ \tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)$$

10.3 The Chi-squared distribution

A random variable formed as

$$\mathcal{T}(\mathbf{x}) = \sum_{i=1}^{p} x_i^2, \tag{48}$$

where x_i, for $i = 1, ..., p$, are independent Gaussian distributed random variables with zero mean and unit variance, is chi-squared distributed with p degrees of freedom, with probability density function (pdf)

$$f_{\mathcal{T}}(t) = Pr(\mathcal{T} = t) = \frac{1}{2^{p/2}\Gamma(p/2)} t^{p/2-1} e^{-t/2}, \tag{49}$$

where Γ denotes the Gamma function [15]. Its cumulative distribution function (cdf) for p degrees of freedom is

$$F_{\mathcal{T}}(t) = Pr(\mathcal{T} \leq t) = \frac{1}{\Gamma(p/2)} \gamma\left(\frac{p}{2}, \frac{t}{2}\right), \tag{50}$$

where γ denotes the incomplete Gamma function. The inverse of the chi-squared cdf for p degrees of freedom given a probability y is then

$$Q_{\mathcal{T}}(y) = \Gamma\left(\frac{p}{2}, \frac{1}{2y}\right) \Big/ \Gamma\left(\frac{p}{2}\right), \tag{51}$$

yielding the maximum value of $\mathcal{T}$ for which $Pr(\mathcal{T} \leq \mathcal{T}_{max}) \leq y$. Figure 20 and 21 shows the pdf and the cdf for the chi-squared distribution for 8 degrees of freedom respectively. The inverse function $Q_p(1-\alpha)$ for a significance level of $\alpha = 0.05$ (5%) is indicated in both plots.

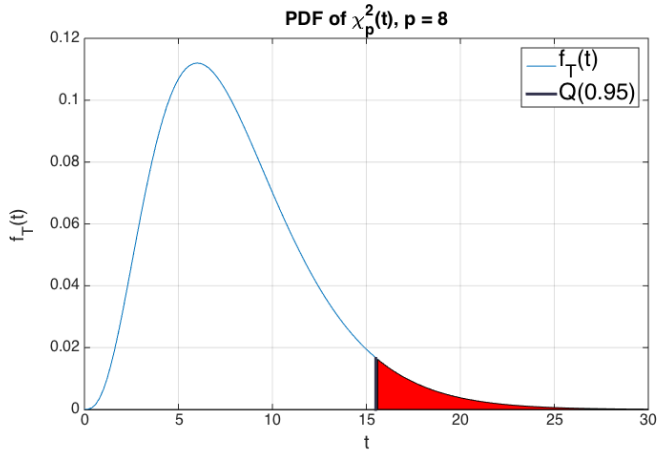

Figure 20: PDF of the χ^2_p-distribution for $p = 8$ degrees of freedom. The red area under the curve is the rejection region, which, for a significance level of $\alpha = 0.05$, corresponds to 5% of the total area.

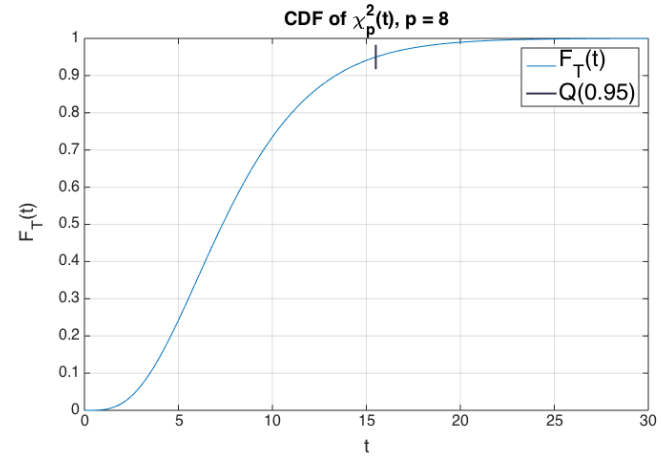

Figure 21: The CDF of the χ^2_p-distribution for $p = 8$ degrees of freedom.

10.4 MATLAB code

10.4.1 PCA in MATLAB

The sample correlation matrix is obtained in MATLAB using the `cov` and `zscore` functions, which is equivalent to using (13). Next step is to compute the SVD on the sample correlation matrix to obtain $\mathbf{V}$ and $\mathbf{\Lambda}$. Ones the SVD has been computed, the amount of variance that each eigenvalue in $\mathbf{\Lambda}$ make

up for can be computed. Finally the principal components can be obtained by projecting $\mathbf{X}$ on to $\mathbf{V}$. How to perform all these steps in MATLAB are shown in algorithm 6.

Algorithm 6 PCA in MATLAB

```
1: procedure PCA
2:     %Sample correlation matrix:
3:     R = cov(zscore(X'));
4:
5:     %SVD on R:
6:     [V, Lambda, V_T] = svd(R);
7:
8:     %Variance proportion:
9:     eigvalues = diag(Lambda);
10:    prop_var = eigvalues./sum(eigvalues);
11:
12:    %Principal components:
13:    Z = (V')*X;
14: end procedure
```

References

[1] European Central Bank, *Statistics*, https://www.ecb.europa.eu/stats/html/index.en.html

[2] I. Aldridge, 2010, *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*, 1th ed, Wiley, New Jersey, US.

[3] Investopedia, *Market Maker*, http://www.investopedia.com/terms/m/marketmaker.asp

[4] Investopedia, *Exchange Rate*, http://www.investopedia.com/terms/e/exchangerate.asp

[5] I.T. Jolliffe, 2002, *Principal Component Analysis, Second Edition*, Springer, New York, p. 1-6, 92, 113.

[6] C. Radhakrishna Rao, 1973, *Linear Statistical Inference and Its Applications, Second Edition*, 2nd ed., Wiley, New York, p. 16, 42.

[7] Mathworks, 2015.05.27, *Documentation: svd* http://se.mathworks.com/help/matlab/ref/svd.html

[8] C.M. Bishop, 2006, *Pattern Recognition and Machine Learning*, 1th ed., Springer, Cambridge, p. 1-3, 25, 93-94, 198-199.

[9] Wikipedia, *Principal Component Regression*, http://en.wikipedia.org/wiki/Principal_component_regression#Shrinkage_effect_of_PCR

[10] Mathworks, *Documentation:det*, http://se.mathworks.com/help/matlab/ref/det.html

[11] Mathworks, *Pattern Recognition*, http://se.mathworks.com/discovery/pattern-recognition.html

[12] Wikipedia, *Maximum a posteriori estimation*, http://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation

[13] Wolfram Mathworld, *Monotonic Function*, http://mathworld.wolfram.com/MonotonicFunction.html

[14] Wikipedia, *Quantile Function*, http://en.wikipedia.org/wiki/Quantile_function

[15] A. Jakobsson, 2013, *An Introduction to Time Series Modeling*, Studentlitteratur AB, Lund, p. 30, 294.

[16] C. Radhakrishna Rao, 1973, Linear Statistical Inference and Its Applications, Second Edition, *Density Function of the Multivariate Normal Distribution*, 2nd ed., Wiley, New York, p. 527-528.

[17] Wikipedia, 2015.05.15, *Multicollinearity*, (May, 2015) http://en.wikipedia.org/wiki/Multicollinearity

[18] NIST/SEMATECH e-Handbook of Statistical Methods, 2015.05.18, *Chi-Square Goodness-of-Fit Test*, (April, 2012) http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm

[19] NIST/SEMATECH e-Handbook of Statistical Methods, 2015.05.18, *Critical Values of the Chi-square Distribution*, (April, 2012) http://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm

[20] Wikipedia, 2015.04.01, *Nonfarm payrolls*, (July, 2014) http://en.wikipedia.org/wiki/Nonfarm_payrolls

[21] Investopedia, 2015.05.19, *Volatility: Definition of 'Volatility'* http://www.investopedia.com/terms/v/volatility.asp

[22] Wolfram MathWorld, 2015.05.22, *Condition Number*, (May 2015) http://mathworld.wolfram.com/ConditionNumber.html