Combining Unsupervised and Supervised Statistical Learning Methods for

POLINA VASILJEVA
Combining Unsupervised and Supervised Statistical Learning Methods for Currency Exchange Rate Forecasting

POLINA VASILJEVA

Master’s Thesis in Mathematical Statistics (30 ECTS credits)
Master Programme in Applied and Computational Mathematics (120 credits)
Royal Institute of Technology year 2016
Supervisor at Third Swedish National Pension Fund (AP3):
Fredrik Giertz
Supervisor at KTH: Jimmy Olsson
 Examiner: Jimmy Olsson

TRITA-MAT-E 2016:48
ISRN-KTH/MAT/E--16/48-SE

Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci
Combining Unsupervised and Supervised Statistical Learning Methods for Currency Exchange Rate Forecasting

Abstract

In this thesis we revisit the challenging problem of forecasting currency exchange rate. We combine machine learning methods such as agglomerative hierarchical clustering and random forest to construct a two-step approach for predicting movements in currency exchange prices of the Swedish krona and the US dollar. We use a data set with over 200 predictors comprised of different financial and macro-economic time series and their transformations. We perform forecasting for one week ahead with different parameterizations and find a hit rate of on average 53%, with some of the parameterizations yielding hit rates as high as 60%. However, there is no clear indicator that there exists a combination of the methods and parameters that outperforms all of the tested cases. In addition, our results indicate that the two-step approach is sensitive to changes in the training set.

This work has been conducted at the Third Swedish National Pension Fund (AP3) and KTH Royal Institute of Technology.
Valutakursprediktion med hjälp av övervakade och oövervakade statistiska inlärningsmetoder

Sammanfattning

I denna uppsats analyserar vi det svårlösta problemet med att prognostisera utvecklingen för en valutakurs. Vi kombinerar maskininlärningsmetoder såsom agglomerativ hierarkisk klustering och Random Forest för att konstruera en modell i två steg med syfte att förutsäga utvecklingen av valutakursen mellan den svenska kronan och den amerikanska dollarn. Vi använder över 200 prediktorer bestående av olika finansiella och makroekonomiska tidsserier samt deras transformationer och utför prognoser för en vecka framåt med olika modellparametreringar. En träffsäkerhet på i genomsnitt 53% erhålls, med några fall där en träffsäkerhet så hög som 60% kunde observeras. Det finns emellertid ingen tydlig indikation på att det existerar en kombination av de analyserade metoderna eller parametreringarna som är optimal inom samtliga av de testade fallen. Vidare konstaterar vi att metoden är känslig för förändringar i underliggande träningsdata.

Detta arbete har utförts på Tredje AP-fonden (AP3) och Kungliga Tekniska Hogskolan (KTH).
Acknowledgements

I would like to thank Jimmy Olsson, my supervisor at KTH Royal Institute of Technology, and Fredrik Giertz, my supervisor at the Third Swedish National Pension Fund (AP3), for their valuable support throughout this thesis work.

Stockholm, August 2016

Polina Vasiljeva
Contents

1 Introduction 1
  1.1 Background ........................................ 1
  1.2 Previous Work .................................... 1
  1.3 This Thesis ....................................... 5
  1.4 Outline ........................................... 6

2 Theory 7
  2.1 Clustering ........................................ 7
  2.2 Time Series Clustering ............................. 11
  2.3 Classification with Tree Based Methods .......... 12
  2.4 Pruning and Random Forest ....................... 15
  2.5 Sequential Supervised Learning Methods .......... 16

3 Method 19
  3.1 Data Set ......................................... 19
    3.1.1 Variable Transformations ..................... 19
    3.1.2 Data Set Creation ............................ 23
  3.2 Two-Step Approach ................................ 24
  3.3 Curse of Dimensionality ........................... 25

4 Results and Analysis 26
  4.1 Clustering ........................................ 26
  4.2 Decision Tree Learning ............................ 28

5 Discussion and Conclusions 33

A Distribution of Clusters by Type and Country 35

B Random Forest Classification Accuracies 38

C Example of a Tree Model 40

D Summary of Classification Accuracies 41

Bibliography 43
Chapter 1

Introduction

1.1 Background

The currency market constitutes a large part of the global financial market, and the value of a country’s currency can be regarded as directly reflective of the country’s present state and expected future development. Hence, it is no surprise that currency exchange rate forecast is one of the principal problems in finance. Forecasting in finance is usually done by employing various regression analysis techniques. Among these techniques one finds traditional time series models of financial data, as well as more recently emerged machine learning methods.

1.2 Previous Work

Exchange rate forecasting has been a subject of study for academic researchers and financial institutions since the mid-80’s. However, up until now there is no general model that has been proven effective and accurate in forecasting exchange rates [30]. In the econometric lingo one distinguishes between two types of analysis for exchange rate forecasting: fundamentals-based analysis, also known as structural models, and non-structural analysis [30]. Furthermore, when forecasting currency exchange rates there is a distinction between building the model over short terms and long terms [30, 34]. The difference lies in the amount of historical data used to build the model, which can range between everything from 1 to 40 years [34].

Structural models merge ideas from economic modeling, which often rely on studies of supply and demand, and statistical analysis [31, 1]. Examples of structural models are behavioral and monetary models, such as the money market equilibrium model [30]. One of the foundations of the monetary models is the purchasing power parity (PPP) model. The PPP theory states that two different countries should have the same purchasing power [30]. This is equivalent to saying that the exchange rate $s$ between two countries is equal
to the domestic price level $P$ divided by the price level of the foreign country $P^*$. Hence, PPP assumes political and economic regulations to be in place in order to keep the parity. In addition, PPP model is based on the assumption on the law of one price, which states that upon adjustment for exchange rate, the same good should have the same price in two different countries. One of the problems with estimating the exchange rate by PPP is the choice of price index. Typically, one uses the consumer price index (CPI) or producer (wholesale) price index (WPI), gross domestic product (GDP), as well as import and export prices. The data available for this type of analysis includes historic values from before 1973, which makes long run predictions possible. Using data from 1973 and forward leaves fewer data points for the analyst, however it incorporates information that is more consistent with today’s policies and economic regimes [30]. In fact, the Bretton Woods system, which was in place until 1971, makes it impossible to look at historical prices before 1973, when currencies were fixed to the US dollar, which in its turn was fixed to gold, contrasting to the current free floating exchange rate regime [30]. Nonetheless, PPP can be evaluated for both short and long run terms, and it is the chosen data sample that governs the evaluation. There are several issues with using PPP for exchange rate estimation, which range from data collection methods to bias induced by differences in the level of development of the two countries assumed to exhibit parity [30].

Time series analysis is a branch of non structural models that is widely used for modeling financial data. Below we state a definition of a univariate time series model by Brockwell and Davis found in [5]:

**Definition 1.1** A time series model for the observed data $\{x_t\}$ is a specification of the joint distributions (or possibly only the means and covariances) of a sequence of random variables $\{X_t\}$ of which $\{x_t\}$ is postulated to be a realization.

The classical decomposition model suggests that a time series $X_t$ can be written as a sum of three components: the trend $m_t$ and the seasonal component $s_t$, which are considered deterministic, and the stationary residual $Y_t$ [5].

$$X_t = m_t + s_t + Y_t$$

When using time series models for prediction and forecasting, the analysis is assumed to be performed on the stationary residuals. Brockwell and Davis define stationarity of time series as follows [5]:

**Definition 1.2** $\{X_t\}$ is (weakly) stationary if

i. $\mu(X_t) = E[X_t]$ is independent of $t$
ii. The autocovariance function at lag $h$, $\text{Cov}(X_{t+h}, X_t)$, is independent of $t$ for each $h$.

These definitions are stated for the univariate case, however, they apply for a multivariate time series $\{X_t\}$ case as well. Commonly, time series data is pre-processed prior to analysis, i.e. trend and seasonal components are identified and removed. Linear trends can be eliminated by taking the natural logarithm of the series values. Differentiation, i.e. taking differences at a chosen lag $d > 0$ such that the original series $\{X_t\}$ is replaced by $\{Y_t := X_t - X_{t-d}\}$, is another example of how to deal with trend and seasonality [5]. Time series models make use of previous values of a time series to forecast future ones. After the forecasting has been performed on the residuals, the inverse of the employed transforms is applied to obtain a forecast for the original data series. A typical error measure for evaluating time series models is mean standard error [5, 34].

There exist both linear and non-linear time series models. Among the linear models, autoregressive (AR), moving average (MA), and autoregressive moving average (ARMA) type of models is common in financial modeling and forecasting [2]. Autoregressive conditional heteroscedastic (ARCH) and generalized autoregressive conditional heteroscedastic (GARCH) models are non-linear time series models frequently used for modeling financial returns [2, 5]. All of the above mentioned models assume stationarity of $\{X_t\}$.

Another model widely used for forecasting stock prices and currency exchange rates is the random walk model. Definition 1.3 states the formulation of a simple symmetric random walk or random walk without drift, i.e. the mean of the stochastic process is zero. It is contrasted to random walk model with drift, i.e. non-zero mean.

**Definition 1.3** [5] Let $\{X_t\}$ be a binary process, i.e. $P[X_t = 1] = p$ and $P[X_t = -1] = 1 - p$ with $p = \frac{1}{2}$. The random walk $\{S_t\}_{t=0,1,2,...}$ with $S_0 = 0$ is

$$S_t = \sum_{i} X_t$$

The random walk model assumes that a stock price movement is only dependent on today’s price, and the historical data past today is not relevant for determining the future. It is considered to be consistent with the efficient market hypothesis, postulated by Fama in 1970 [10]. The main idea of the hypothesis is that the market is “informationally efficient”. The hypothesis assumes that all the information that is needed for making an investment is already available. This means that every investor has the same chance of making profit. The information referred to here can for instance be historical development of prices. The random walk model for forecasting of foreign

\[^1\text{For formal definition and mathematical properties of these models, see [5].}\]
exchange rates yields 50% forecasting accuracy on average [34], once again casting light the difficulty of this type of forecasting.

One of the most influential studies of exchange rate forecast was carried out by Meese and Rogoff in 1983 [30]. They compared structural and non-structural models for one to twelve months forecasting of logarithm of foreign exchange rate [23]. Two kinds of monetary models as well as AR models for both univariate and multivariate time series were studied. The comparison was carried out on the basis of out of sample prediction accuracy. The models use variables such as bilateral money supplies, relative incomes, interest rate differentials, CPI and WPI inflation rates, as well as trade balances as predictors. In addition, the authors test the random walk model without drift and with the mean of the logarithm of monthly change in exchange rate as the drift parameter. The obtained results suggested that no tested model outperforms the random walk model. This result is one of the fundamental results in currency exchange rate forecasting that led to development of various modeling techniques to face this challenging problem.

Artificial neural networks (ANNs) are modeling methods emerged as a popular technique for exchange rate forecasting [30, 37, 33, 34, 36]. ANNs is a non-linear modeling technique developed in the middle of 20th century inspired by the way information is processed and transmitted by the human brain [3]. Tibshirani et al. defines a neural network as a two step classification or regression model viewed as a network diagram, consisting of an input layer, one or multiple hidden layers and an output layer [11]. Contrary to time series models that assume that data follows a parametric form, ANNs are non-parametric and hence do not make any explicit assumptions about the data. 2 Yu et al. provide a comprehensive summary of 45 articles on foreign exchange rate forecasting with ANNs published between 1993 and 2004 [36]. Their literature review shows that 60% report ANNs performance as better compared to other methods, and only 4% as worse.

Walczak argues that neural network modeling is sensitive to choosing short and long term data for modeling, i.e. varying the size of the training set to include less or more price history [34]. The employed currency price series is differentiated up to 10 days backwards in time. Next, variable selection is performed in order to isolate the variables with the most appropriate lags. Lastly, forecast is performed to find the price at \( t + 1 \), i.e. one day ahead. The author suggests that finding a direction of change, i.e. the sign of the difference between the predicted future currency priced and today’s price, is a better error measure due to the fact that an error, no matter how small in magnitude, will always result in capital loss. Hence, one can argue that this forecasting problem is transformed into a classification problem. Walczak finds classification accuracies of on average 55-60% and suggests that a smaller training data size, i.e. using data corresponding to more recent

\footnote{For rigorous mathematical treatment of neural networks, see [11, 3].}
years, yields better classification accuracy when using his model.

Another popular machine learning technique for financial time series forecasting, including currency exchange rate forecasting, is support vector machines (SVMs) [17, 28, 21, 19, 2]. SVMs are often contrasted to ANNs as a method that results in less overfitting [19, 17]. SVM is a regression and classification technique which originates from maximum-margin hyperplane binary classification, where the two classes are linearly separable by a hyperplane such that the distance between the points belonging to the two classes and the hyperplane is maximized. SVMs are an extension of this methodology suitable when the separation boundary between observations is non-linear. By using a kernel function, the dimensionality of the data is increased and a non-linear classifier is obtained. The most common kernel functions are polynomial up to degree $d$, radial basis and neural networks [11]. The constructed classifiers are linear in the new high dimension, but non-linear in the original dimension of the data. SVMs have two main tuning parameters: the type of kernel and the upper bound on the coefficients of the hyperplane equation. 

Kim uses SVMs to predict the sign of change of the daily Korea composite stock price index [19]. In total 13 transformations of the original price series are used as input to the model. The obtained hit rate on the test data ranges between 50.09 to 57.83%. The author also shows that SVMs perform 3.10% better than ANN.

Kamruzzaman et al. use SVMs to perform forecasting of currency exchange rate of Australian dollar against 6 different currencies one week ahead [17]. They use weekly prices and their MA($k$) transformations with $k = \{5, 10, 20, 60, 120\}$ as inputs to the model. They evaluate SVM with different kernel functions and measure the percentage of correctly predicted upward and downward trends, as well as percentage of the total amount of correctly predicted trends. The smallest obtained hit ratio is 70% and the average prediction accuracy is 80% for the investigated cases.

1.3 This Thesis

In this work, we employ methods that stem from machine learning to develop a method for prediction of upward or downward movement of the price of the US dollar (USD) in the Swedish krona (SEK). We use a data set consisting of different financial and macro-economic time series and their transforms, yielding in total 234 variables available for forecasting. The data consists of weekly values ranging from January 3, 2000 to December 31, 2015. The proposed methodology is a two-step approach. In the first step we use agglomerative hierarchical clustering, an unsupervised statistical method, to find a reduced set of variables that we assume to be sufficient for the pre-

\[3\] For a detailed treatment of SVMs, see [11, 26]
diction task. In the second step we train a random forest classifier where forecasting of the movement of the currency price series is done for one week ahead. The random forest model is learned on the data transformed according to the sliding window methodology, i.e. a technique that allows for inclusion of time dependency between observations when modeling using machine learning algorithms [14].

1.4 Outline

This thesis organized as follows. Chapter 2 provides a theoretical framework of the employed methods. In Chapter 3 we provide a detailed description of the two-step approach including the data set construction. Chapter 4 describes our findings, and finally Chapter 5 focuses on discussion and conclusions.
Chapter 2

Theory

This chapter is divided into two blocks corresponding to unsupervised and supervised statistical learning and is heavily based on [11, 26]. Consider a set of variables (also called predictors or features) $\mathbf{X} = (X_1, \ldots, X_p), X_i \in \mathbb{R}^n$, where $p$ can be very large. In a supervised learning setting, there is a response variable (also called target variable or labels) $Y \in \mathbb{R}^n$ associated with $\mathbf{X}$. If $Y$ is continuous the supervised learning problem is called regression, and if $Y$ is categorical it is called classification. Model quality is assessed by obtaining a prediction of $Y$, $\hat{Y}$, and finding the prediction error by calculating some loss function $L(Y, \hat{Y})$. Among supervised methods one finds decision tree methods for classification and regression, discussed in this chapter, as well as other regression-based modeling techniques and classification methods such as $k$ Nearest Neighbors (kNN).

In an unsupervised setup, the data set $\mathbf{X}$ is unlabeled, i.e. there is no $Y$ available. As a consequence there is no direct measure of the error of a supervised method. Rather, unsupervised methods are often referred to as exploratory as they help uncover natural groups and structures in the data. Principal Component Analysis (PCA) and hierarchical clustering are examples of unsupervised methods.

2.1 Clustering

The clustering techniques are also called data segmentation techniques since the objective is to segment or group the available data [11]. These are widely used in a large number of applications, such as consumer data segmentation, genetic applications and information retrieval [22]. They are divided into different techniques depending on the input and output of the clustering method, and whether the method is probabilistic or not. Looking at the input, there exist two kinds of clustering, namely similarity (or dissimilarity) based clustering and feature-based clustering. As for the output, one may use a clustering method that yields disjoint sets of clusters, i.e. flat clustering,
e.g. $K$-means or $K$-methodoids. The input to such methods is a predefined number of clusters $K$. Another possible output is a dendrogram produced by a hierarchical clustering method, where objects belonging to a cluster are also members of a so called parent cluster. Furthermore, one distinguishes between non-probabilistic and probabilistic clustering methods, where each observation may belong to different clusters with certain probability [26]. In this report, we are concerned with similarity-based clustering.

Similarity-based clustering reveals groups depending on how similar or dissimilar to each other pairs of observations in these groups are. The objects can be both individual observations and a collection of observations stored in a variable. Hierarchical clustering is an example of similarity based clustering method. Some clustering algorithms require dissimilarities to be distances $d$ in a mathematical sense, i.e. satisfying the following conditions for all $x, y, z$ in a set $X$:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \iff x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, y) \leq d(x, y) + d(y, z)$

We recognize the above properties as non-negativity (1), identity of indiscernibles (2), symmetry (3), and triangle inequality (4). However, most clustering algorithms, including hierarchical clustering, accept dissimilarities which often fulfill all the properties except the triangle inequality (4) [26].

Dissimilarity can be measured in different ways, and the choice is dependent on the type of data. The most commonly used dissimilarity measures are examples of $L_q$-norms, i.e. Manhattan or Euclidean norm. We cover different dissimilarity measures suitable for time series data in Section 2.2.

Hierarchical clustering procedures are either agglomerative (down-top) or divisive (top-down). Conceptually, the difference lies in the way objects in the data set are divided into clusters. Agglomerative clustering algorithms start by assigning each object into its own cluster (minimum dissimilarity) and creates clusters containing more objects by merging clusters depending on how similar they are. Divisive clustering employs the reverse procedure.

The hierarchical clustering algorithm is shown below. It takes as an input a dissimilarity matrix $D = [d_{i,j}]$, where $d_{i,j}$ is the dissimilarity of objects $i$ and $j$. The objects may for example be a collection of observations, such as time series, or individual observations.
input: \( p \times p \) dissimilarity matrix \( D \)
output: Dendrogram

Initialize clusters as singletons:

\[
\text{for } i \leftarrow 1 \text{ to } p \text{ do} \\
\quad C_i \leftarrow \{i\} \\
\text{end}
\]

Initialize set of clusters available for merging \( S \leftarrow \{i, \ldots, p\} \)

\[
\text{repeat} \\
\quad \text{Merge two clusters according to the chosen dissimilarity measure} \\
\quad (j, k) \leftarrow \text{argmin}_{j, k \in S} d_{j,k} \\
\quad \text{Create new cluster } C_l \leftarrow C_j \cup C_k \\
\quad \text{Update } S \leftarrow S \setminus \{j, k\} \\
\quad \text{if } C_l \neq \{i, \ldots, p\} \text{ then} \\
\quad \quad \text{Update } S \leftarrow S \cup \{l\} \\
\quad \text{end} \\
\quad \text{foreach } i \in S \text{ do} \\
\quad \quad \text{Update dissimilarity matrix } d(i, l) \\
\quad \text{end} \\
\text{until no more clusters are available for merging}
\]

\textbf{Algorithm 1:} Agglomerative hierarchical clustering. Adapted from [26].

There are different ways to assess the dissimilarity between groups of objects, i.e. formed clusters, and these affect the resulting clustering. The most common ways to be found in the literature are single, complete and average linkage [11, 26]. Murphy [26] describes the different linkage types by an example of two distinct groups, denoted here \( A \) and \( B \), that each contain a number of objects \( n_A \) and \( n_B \) respectively. Single linkage clustering merges the groups based on the shortest distance between the objects within \( A \) and \( B \)

\[
d_{\text{SL}}(A, B) = \min_{a \in A, b \in B} d_{a,b}.
\]

In contrast to single linkage, complete linkage defines the distance between \( A \) and \( B \) according to the maximum distance between the observations within the groups

\[
d_{\text{CL}}(A, B) = \max_{a \in A, b \in B} d_{a,b}.
\]

Average linkage is a mixture between complete and single linkage, such that the average distance between all the observations in both groups is considered.

\[
d_{\text{avg}}(A, B) = \frac{1}{n_A n_B} \sum_{a \in A} \sum_{b \in B} d_{a,b}.
\]
It is the most commonly used linkage type in hierarchical clustering [26]. It is worth to note that while complete and single linkage are invariant to transformation of the data, e.g. change of scale, average linkage is sensitive to different transformations of the data.

As mentioned previously, hierarchical clustering is an unsupervised statistical method and hence there is no universal way to assess the degree of correctness of results produced by clustering, because the method is unsupervised. There are however ways to access a measure of clustering quality, and these methods are divided into internal and external criteria of quality of clustering [22]. A high internal criterion of quality is obtained if the inter-cluster distance is maximized and the intra-cluster distance is minimized. In plain words, this implies a high degree of similarity between the objects within a cluster, and a high degree of dissimilarity between the objects in different clusters.

The external quality of cluster evaluation can be performed if the underlying distribution of objects into different groups is known. These groups are considered to be the true classes of an object, and the distribution can be taken as a benchmark against which the clustering results may then be compared. An example of such evaluation measure is purity. Let $j = \{1, \ldots, C\}$ denote the true class of an object, and $i$ the cluster that the object has been assigned to by the clustering algorithm. Then $N_{ij}$ denotes the number of objects that belong to class $j$ in cluster $i$. Murphy defines purity in the following way [26]:

$$
purity = \sum_i \frac{N_i}{N} p_i, \tag{2.1}
$$

where $N_i = \sum_{j=1}^C N_{ij}$ is the total amount of objects contained in cluster $i$ and $p_i = \max_j p_{ij}$, i.e. the majority class object in cluster $i$ with $p_{ij} = \frac{N_{ij}}{N_i}$. If clustering results are not in agreement with the true grouping, purity takes values around 0. The opposite situation yields purity close to 1. There exist other external quality evaluation methods such as rand index and mutual information [26].

Furthermore, assessing the correct amount of groups in the data is another fundamental consideration when evaluating clustering results. While some applications provide a choice to determine the desired amount of groups, e.g. consumer data segmentation [11], other applications require some method of evaluation of the different partitionings that can be obtained with clustering. Tibshirani et al. propose a method for estimating the optimal amount of clusters in the data by using a so called gap statistic [11].

A drawback of clustering methods is that they will always produce results no matter how meaningful they are. For instance, a clustering algorithm will be able to cluster a data set of random noise, which interpretability is impossible to assess due to the nature of the data.
2.2 Time Series Clustering

When working with static data, clustering is performed on each individual observation. However, time series data cannot be partitioned such that parts of the individual series end up in different clusters due to time dependency within the series. Consequently, there has been extensive development of data mining methods for time series data [20, 12, 27].

One of the challenges with financial time series data is its noisy nature and autocorrelation. Performing clustering on a financial data set where the classification into groups is known has been described in [27, 13, 18]. De Luca et al. propose a dynamic clustering algorithm for portfolio selection procedure [6]. When clustering dynamically, clustering results for data corresponding to time \( t \) are used as benchmark for clustering results corresponding to time \( t + 1 \). The authors stress the point of financial data being time dependent and that this should be taken into account when performing clustering. In addition, the authors suggest that when model is build using dynamic clustering, it should be re-estimated frequently in order to account for the possible changes. This type of models suit better for very volatile periods. The authors measure the degree of agreement between the different solutions in time using rand index.

Esling et al. propose following similarity measures for time series clustering: shape-based, edit-based, feature-based, structure-based, model-based, and compression-based [9]. In addition, Fu [12] distinguishes between whole sequence and subsequence matching. Shape-based similarity measures match time series with similar geometry. The most famous of these measures are \( L_q \) distances, such as Euclidean (\( L_2 \)) or Manhattan distance (\( L_1 \)). A prerequisite for using these measures is that the series need to be normalized as the measures are sensitive to difference in scale. Another very popular distance is based on Dynamic Time Warping (DTW), a series aligning technique originally developed for speech recognition and for series of different lengths [25]. It matches time series by finding an optimal alignment between points in time. The DTW has been used for multiple time series clustering and classification problems, where it has been found to give good performance [12, 29, 9, 35, 38]. One of the disadvantages of the method is its \( O(n^2) \) computational complexity. Different modifications of the method have been developed in order to cope with the problem [9].

An alternative to geometrical similarity representation is the abstract representation through correlation. Popular correlation-based dissimilarity measures are \( \sqrt{2(1-\rho)} \) [11, 27] where \( \rho \) is Pearson’s correlation coefficient, approximated by \( \hat{\rho} \), sample-based Pearson’s correlation coefficient. For two time series \( X = \{x_1, \ldots, x_n\} \) and \( Y = \{y_1, \ldots, y_n\} \), the sample-based Pearson correlation coefficient is defined as follows:
\[ \hat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}, \]

where \( \bar{x} \) is the sample mean \( \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \) and \( \bar{y} \) is defined in the same way.

Another dissimilarity measures involving correlation and frequently used in applications such as gene expression clustering are \( 1 - \rho \) and \( 1 - s \), where \( s \) is the Spearman’s rank correlation. While Pearson’s correlation coefficient assumes a linear relationship between variables, Spearman’s rank correlation coefficient is non-parametric. Instead, it assesses the relationship between two variables by describing it as a monotonic function [15]. The sample-based Spearman’s rank correlation coefficient \( \hat{s} \) is approximated in the same way as Pearson’s correlation coefficient, except the numerical values of the series are replaced by their ranks. This treatment of the data is assumed to give a correlation measure that is less sensitive in outliers in the data [16].

Other approaches in estimating dissimilarity for time series involve periodogram-based approaches and model-based approaches. The latter assume that time series can be modeled mathematically by ARMA or ARIMA processes. When the models are fit to the series, different model fits are clustered together [24].

2.3 Classification with Tree Based Methods

Decision trees are widely used for both classification and regression problems in different domains due to their interpretability [32]. The main idea with decision tree learning is partitioning the sample space into distinct regions \( R_m, m = 1, \ldots, M \) according to a recursive split pattern. Hastie et al. argue that binary splits yield models prone to overfitting by avoiding data fragmentation, i.e. a split leaving a shortage of data for the next split [11]. Hence, we consider here tree based methods that use binary recursive partitioning. One of the most popular tree methods is CART (Classification And Regression Tree), however, there exist other implementations, such as C4.5 and ID3 [11, 26]. In this section, we will cover the CART implementation.

To illustrate how the input space is partitioned when building a decision tree classifier with CART, we adapt an example by Tibshirani et al. [11]. Consider two variables \( X_1, X_2 \in [0, 1] \), i.e. they define a region corresponding to a unit square. Associated with these variables is a continuous response variable \( Y \). A CART implementation would first split the unit square into two separate regions. An example of such split can be \( X_1 \leq t_1 \) for some \( t_1 \in [0, 1] \). This split is called an internal node of the tree. In the next step, each of these regions is split into two new regions, e.g. the region defined by \( X_1 \leq t_1 \) is split at \( X_2 = t_2 \), and the region defined by \( X_1 > t_1 \) is split at \( X_2 = t_3, t_3 > t_2 \). Suppose that the splitting has now reached a stopping criterion. We obtain 4 terminal nodes (also called leaves), each
corresponding to regions $R_1, \ldots, R_4$. The obtained partitioning of the input space is shown in Figure 2.1. In a regression setting, the value of the response variable $Y$ in region $R_m$ is approximated by a constant $c_m = \text{ave}(Y \in R_m)$. In a classification setting it is the the majority class, i.e. the class that is represented the most in a given region.

For a general case, consider a data set that consists of observations $(X_i, Y_i), i = 1, \ldots, N$ with $X_i = \{X_{ij}\}_{i=1,\ldots,N,j=1,\ldots,p}$ and a response $Y_i$. Equation 2.3 shows an expression for splitting criterion formulated by Murphy [26], i.e. which feature and what value of this feature that yield the best split. It assumes that $X_i$ are continuous and $j^*$ and $t^*$ are the best attribute to split on and the best value of this attribute, respectively. $T_j$ represents the ordered set of unique values of $X_j$.

$$(j^*, t^*) = \arg \min_{j \in \{1,\ldots,p\}} \min_{t \in T_j} \text{cost}(\{X_i, Y_i : X_{ij} \leq t\}) + \text{cost}(\{X_i, Y_i : X_{ij} > t\}).$$
The splitting criterion involves a cost function that should be minimized, which is dependent on whether the model assumes regression or classification. It gives information whether the split should occur or not. Denote $D$ as the available data in a region $R_m$. An approach to evaluating the cost of splitting on a certain attribute in the regression setting is given by

$$\text{cost}(D) = \sum_{i \in D} (Y_i - \bar{Y}),$$

where $\bar{Y}$ is the sample mean of the responses corresponding to the region $R_m$.

Classification problems require a different cost function. First we define the probability of a class $c = 1, \ldots, C$ being represented in the region $R_m$ by

$$\hat{p}_{mc} = \frac{1}{|D|} \sum_{i \in D} 1(Y_i = c).$$

Let $\hat{p}_{mc(m)}$ be the probability of the majority class in $D$, i.e. the probability of $\hat{Y}_c = \arg\max_c \hat{p}_{mc}$. The following cost functions are used for inferring classification trees:

$$\frac{1}{D} \sum_{i \in D} 1(Y_i \neq \hat{Y}_c) = 1 - \hat{p}_{mc(m)},$$

known as misclassification rate, i.e. proportion of the observation with incorrect class assignments.

$$\sum_c \hat{p}_{mc}(1 - \hat{p}_{mc}),$$

(2.2)

known as Gini index. Let the event $C$ be the outcome of a discrete random variable $C$, governing the class membership of a random observation in the region $R_m$. It is easily concluded that Gini index corresponds to the expected error rate.

$$-\sum_c \hat{p}_{mc} \log \hat{p}_{mc},$$

known as entropy or deviance.

The splitting continues until a splitting criterion is reached, which can be a certain pre-specified amount of terminal nodes, or a tree size. Methods for determining an optimal tree sized are described in Section 2.4. Finally, the classification tree algorithm proposed by Murphy [26] that summarizes these concepts is shown below.
**input**: Set of training examples $\mathcal{D}$

**output**: Binary tree

function `fitTree` (node, $\mathcal{D}$, depth)

node.prediction = $\hat{Y}_c$

$(j^*, t^*, \mathcal{D}_L, \mathcal{D}_R) = \text{split}(\mathcal{D})$

if `notWorthSplitting`(depth, cost, $\mathcal{D}_L, \mathcal{D}_R`) then

    return node

else

    node.left = `fitTree`(node, $\mathcal{D}_L$, depth+1)

    node.right = `fitTree`(node, $\mathcal{D}_R$, depth+1)

    return node

end

**Algorithm 2**: Classification tree algorithm. Adapted from [26].

### 2.4 Pruning and Random Forest

Tree methods are usually very sensitive to new inputs, thus having a high variance. The parameter that controls the model complexity is the tree size, as a very large tree might overfit the data and a small one will yield a poor representation of the data. Cost complexity pruning is a technique that has been developed to find the optimal tree size. The idea behind pruning is to grow a large tree $T_0$ and then reduce its size to $T$ by computing the complexity criterion in each split [11]. The complexity criterion is a sum of the cost function and a tree size constraint. The tree that minimizes the complexity criterion is considered to be the optimal in terms of model complexity. However, when the data is exceptionally noisy and non-stationary, pruning techniques do not provide sufficient variance reduction. Random forest is a variance reduction technique developed by Leo Breiman and is an extension of his earlier work on bagging i.e. bootstrap aggregation method for reducing variance [4].

The idea with bagging is to use bootstrap to create $B$ artificial replicates of the training set $(X, Y)$ by sampling from the original training set with replacement. A non-pruned tree model is then fit on each of these artificial training sets, such that an ensemble of trees $\{T_b\}_{b=1}^B$ is produced. The bagging estimate is then acquired by taking the mean of the predicted values over the bootstrapped samples in the regression case, or finding out the majority class in the classification case. Each tree is unbiased but highly variable, and averaging over $B$ trees reduces variance. However, this methodology produces highly correlated trees, since if there exists a very strong predictor in the training data set, it will be the first attribute to split on in each of the tree models [11].

Random forest reduces the variance of the estimator by decorrelating the trees. This is done by at each resampling picking $m < p$ predictors to build a tree model on, where $p$ is the total number of predictors. Typically, $m$ is
chosen such that \( m = \lfloor \sqrt{p} \rfloor \). However, the choice of \( m \) is case specific and the proposed number serves as a guideline. Note that choosing \( m = p \) yields bagging. The number \( B \) of trees to grow is the tuning parameter of both the bagging and random forest algorithms. It is claimed that an increased \( B \) does not lead to overfitting the data. Breiman provides a proof of this claim, key results of which we briefly state below.

Let \( f_{b,\Theta_b} \) be the output from the classification tree \( T_{b,\Theta_b} \) grown using the set \( \Theta_b \) of randomly chosen predictors. Breiman defines the marginalization function of a random forest as follows:

\[
mg(X, Y) = \frac{1}{B} \sum_{b=1}^{B} 1_{f_{b,\Theta_b}=y} - \max_{k \neq y} \frac{1}{B} \sum_{b=1}^{B} 1_{f_{b,\Theta_b}=k}.
\]

The interpretation of the marginalization function follows from the definition: it is the difference between the averages of correctly predicted classes and incorrectly predicted classes. Hence, the larger the margin the better classification accuracy. In addition, Breiman defines the generalization error of a random forest as:

\[
PE = P(mg(X, Y) < 0),
\]

and shows that it converges as \( B \to \infty \). This result implies that an increase in the number of trees grown does not cause the model to overfit the data. In addition, Breiman shows that the generalization error is bounded in the following way

\[
PE \leq \frac{\rho}{s^2}(1 - s^2).
\]

In this expression, \( s > 0 \) is the measure of the strength of each of the produced classifiers, and \( \rho \) is the correlation between two trees in the forest. Breiman argues that a small value of \( \frac{\rho}{s^2} \) is preferred when constructing a tree model using random forest. It is necessary to note that although an increase in the number of trees \( B \) does not cause the model to overfit the data, the upper limit of the generalization error may lead to overfitting due to too complex trees grown at each resampling [11].

Reducing variance through pruning and random forest approaches comes at a cost of model interpretability. As the final random forest consists of an ensemble of different tree models, it is not possible to obtain one tree model that describes the data. In addition, training a random forest model requires more computational power [11].

### 2.5 Sequential Supervised Learning Methods

Most supervised learning methods are designed under assumption of independence between the observations. However, a large portion of available
data comes in sequences. Time series is a type sequential data since they exhibit autocorrelation between points in time.

In [8], Dietterich provides a comprehensive review of the sequential classification problems. In such problems, both each observation as well as the associated class label are sequences. An example is a sentence as the associated sequence representing whether a word in the sentence is a verb, noun etc. Classifiers that are designed for sequential data account for the correlation structure that is present within the sequence [8]. Among frequently used methods for modeling sequential data one finds Markov models, hidden Markov models, recurrent neural networks, as well as sliding window methods [3, 8]. Below we describe the latter ones.

Sliding window methods allow division of the sequence of observations \( \{x_{i,1}, \ldots, x_{i,j}\} \) by creating subsequences of a window length \( 0 < m < j \). The learning process is done by stepping through the sequence by these subsequences. The classifier is trained on each subsequence, which results in a sequence of labels. In a recurrent sliding window approach, the label prediction made on the previous subsequence is taken into account for finding the label in the next one. Hence, not only the correlations within the sequence, but also the possible correlations with the label sequence are considered when training the classifier.

Time series prediction requires a somewhat different methodology [8]. However, the main principles of training a classifier using sliding window apply in the case of time series data. The main difference for the time series data lies in the fact that if current time is assumed to be some time \( t + 1 \), and observations up to time \( t \) are known to us, the problem encompasses predicting \( y_{t+1} \) given \( y_1, \ldots, y_t \) and, in the presence of additional \( p \) predictor variables \( X_{t+1 \times p} \).

Application of sliding window methodology requires a change of the shape of the training and test data sets. Below we provide a simple illustration of the methodology. In this example, we assume three time series \( X_1, X_2, X_3 \) to be predictors and \( Y \) the response variable. The predictor time series have observations on the time interval \( t \in [1, 6] \) time units. The original data has the following form:

\[
\begin{array}{cccc}
X_{t,1} & X_{t,2} & X_{t,3} & Y_t \\
x_{1,1} & x_{1,2} & x_{1,3} & y_2 \\
x_{2,1} & x_{2,2} & x_{2,3} & y_3 \\
\vdots & \vdots & \vdots & \vdots \\
x_{5,1} & x_{5,2} & x_{5,3} & y_6 \\
x_{6,1} & x_{6,2} & x_{6,3} & \end{array}
\]

Below is the representation of the example data set using the sliding window approach. For this example, we set the size of the sliding window to 2, which yields a new data set where observations are now windows \( w_i, i \in \)
\{1, n - \text{window size}\} instead of successive points in time. The number of features $\xi$ grows with the window size.

<table>
<thead>
<tr>
<th>$\xi_1$</th>
<th>$\xi_2$</th>
<th>$\xi_3$</th>
<th>$\xi_4$</th>
<th>$\xi_5$</th>
<th>$\xi_6$</th>
<th>$W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{1,1}$</td>
<td>$x_{1,2}$</td>
<td>$x_{1,3}$</td>
<td>$x_{2,1}$</td>
<td>$x_{2,2}$</td>
<td>$x_{2,3}$</td>
<td>$w_1 = y_3$</td>
</tr>
<tr>
<td>$x_{2,1}$</td>
<td>$x_{2,2}$</td>
<td>$x_{2,3}$</td>
<td>$x_{3,1}$</td>
<td>$x_{3,2}$</td>
<td>$x_{3,3}$</td>
<td>$w_2 = y_4$</td>
</tr>
<tr>
<td>$x_{3,1}$</td>
<td>$x_{3,2}$</td>
<td>$x_{3,3}$</td>
<td>$x_{4,1}$</td>
<td>$x_{4,2}$</td>
<td>$x_{4,3}$</td>
<td>$w_3 = y_5$</td>
</tr>
<tr>
<td>$x_{4,1}$</td>
<td>$x_{4,2}$</td>
<td>$x_{4,3}$</td>
<td>$x_{5,1}$</td>
<td>$x_{5,2}$</td>
<td>$x_{5,3}$</td>
<td>$w_4 = y_6$</td>
</tr>
</tbody>
</table>
Chapter 3

Method

In this chapter we describe how the employed data set was created, including the different transformations, and give a detailed explanation to the two-step approach used in the investigation.

3.1 Data Set

3.1.1 Variable Transformations

The general idea of the problem setup is the following. Given a number of financial and macro-economic variables that are believed to influence the cross-currency market of USA and Sweden one wishes to incorporate the possible predictive power of these variables. In addition different types of transforms of these variables are considered. The lists of macro-economic time series are displayed in Table 3.1. Table 3.2 shows the employed financial time series. Below we provide an overview of the different time series, which we refer to as variables, together with their transforms.

Table 3.1 provides description of the macroeconomic variables together with the recording frequency and transformations performed on these variables. The variables are recorded monthly, except Swedish money supplies that are recorded on a weekly basis, and RGDP for both countries is reported quarterly.
Table 3.1: List of macroeconomic variables and their abbreviations and transformations.

<table>
<thead>
<tr>
<th>Description</th>
<th>Abbreviation</th>
<th>Transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer Price Index</td>
<td>CPI</td>
<td>YOY</td>
</tr>
<tr>
<td>Import Prices</td>
<td>ImpPrc</td>
<td>YOY</td>
</tr>
<tr>
<td>Export Prices</td>
<td>ExpPrc</td>
<td>YOY</td>
</tr>
<tr>
<td>Money Supply M1</td>
<td>M1</td>
<td>YOY diff</td>
</tr>
<tr>
<td>Money Supply M2</td>
<td>M2</td>
<td>YOY diff</td>
</tr>
<tr>
<td>Trade Balance</td>
<td>TrBal</td>
<td>YOY diff Normed</td>
</tr>
<tr>
<td>Monthly Earnings</td>
<td>MntEarn</td>
<td>YOY</td>
</tr>
<tr>
<td>Retail Sales</td>
<td>RtlSls</td>
<td>YOY</td>
</tr>
<tr>
<td>Consumer Confidence</td>
<td>CnsConf</td>
<td></td>
</tr>
<tr>
<td>Currency Reserves</td>
<td>FXRes</td>
<td>YOY</td>
</tr>
<tr>
<td>Real Gross Domestic Product</td>
<td>RGDP</td>
<td>YOY</td>
</tr>
</tbody>
</table>

The year-over-year (YoY) transformation is used to evaluate percentage change in events on a yearly basis. Let $x_i$ be the value of a financial time series at time $i$, and define $k$ to be the time interval corresponding to one year back in time starting from time $i$. Then, YoY is defined as:

$$
\text{YoY} = \frac{x_i - x_{i-k}}{x_{i-k}}. \tag{3.1}
$$

In addition, YoY is useful for understanding and interpreting changes in macroeconomic variables. A single value of such time series may not provide sufficient information itself, but when put into a context of yearly change reveals insights into the ongoing economic processes. It is assumed that this transformation also handles the seasonal patterns present in the data.

The next transformation to be considered is YoY difference, denoted YoY diff, which is defined as the difference between a value of a time series at time $i$ and its equivalent in the previous year. Employing the same notation as above, we can write a mathematical definition of YoY diff as

$$
\text{YoY diff} = x_i - x_{i-k}. \tag{3.2}
$$

Finally, normalized YoY difference, YoY diff Normed, is computed in the following way, where $\hat{\sigma}$ is the sample-based estimate of the standard deviation computed as follows:

$$
\hat{\sigma} = \left(\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2\right)^{1/2},
$$

$$
\text{YoY diff Normed} = \frac{x_i - x_{i-k}}{\hat{\sigma}_{x_{i-k},...,y_i}}. \tag{3.3}
$$
It is important to realize that the transformations 3.1-3.2 can be computed for any specified time period. Month-over-month (MoM) or quarter-over-quarter (QoQ) are other frequent transformations of financial data.

Table 3.2 shows financial variables employed in the analysis together with a brief description of the performed transforms. The measurements are recorded daily. We will now go through the list of variables one by one and explain the choice of transforms in more detail.

<table>
<thead>
<tr>
<th>Description</th>
<th>Abbreviation</th>
<th>Transforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Total Return</td>
<td>EQ_TRGD</td>
<td>Return, Volatility of return</td>
</tr>
<tr>
<td>Price</td>
<td>EQ_PX</td>
<td></td>
</tr>
<tr>
<td>Earnings Price Ratio</td>
<td>EQ_EY</td>
<td>YoY, QoQ on EQ_E</td>
</tr>
<tr>
<td>Dividend Price Ratio</td>
<td>EQ_DY</td>
<td>YoY, QoQ on EQ_D</td>
</tr>
<tr>
<td>Price-to-Book Ratio</td>
<td>EQ_PB</td>
<td>YoY, QoQ on EQ_B</td>
</tr>
<tr>
<td>Price-to-Sales Ratio</td>
<td>EQ_PS</td>
<td>YoY, QoQ on EQ_S</td>
</tr>
<tr>
<td>Price/EBITDA</td>
<td>EQ_PtoEB</td>
<td></td>
</tr>
<tr>
<td>N-Year Currency Swap Rate</td>
<td>SWAP_NY</td>
<td>Fit Diebold-Li model to swap curve</td>
</tr>
<tr>
<td>Currency exchange rate</td>
<td>USDSEK_Price</td>
<td>Return, Volatility of return</td>
</tr>
</tbody>
</table>

The Swedish equity data belongs to OMX 30 Index, and the USA equivalent is S&P 500 Index. A common way of looking on historical prices of equity indices is to consider its returns \( R_t(\tau) \) over a certain time period \( \tau \), as well as the standard deviation of past returns, i.e. volatility [2].

**Definition 3.1** [2] Simple return \( R_t(\tau) \) on a price \( P_t \) at time \( t \) for a time period of \( \tau \), where \( \tau \) can be specified on any time scale, is computed by

\[
R_t(\tau) = \frac{P_t}{P_{t-\tau}} - 1. \tag{3.4}
\]

There exist other types of returns, however we limit us to the ones computed according to Definition 3.1.

In the case of EQ_TRGD, the returns are computed on a weekly, monthly, quarterly, 6 months and yearly basis. In addition, volatility of returns over a time period \( \tau \) is computed as a sample-based standard deviation of returns.

The variables EQ_EY and EQ_DY represent ratios of earnings to price and dividend to price, respectively. Thus, given the price variable EQ_PX, one can infer earnings EQ_E and dividends EQ_D by taking the product of the ratio variable and the price variable. Similarly, the price-to-book EQ_PB and price-to-sales EQ_PS variables can be used to infer book EQ_B and sales EQ_S variables by dividing the original variables by price variable and taking the inverse.
For the currency swap rates variables, \( N \) represents the number of years, \( N = \{1, 2, 5, 10, 20, 30\} \). At each time point \( t \), the values of these variables can be used to represent the yield curve, i.e. the evolution of interest rates with time. The curve is of importance due to its ability to reflect a country’s economic status. Diebold and Li propose a model for forecasting the yield curve [7]. The model assumes that the curve can be decomposed into a model with three factors, each one with different loadings, that evolve in time. The factors \( \beta_{i,t}, i = \{1, 2, 3\} \) are considered latent and represent the level, slope and curvature of the yield curve, respectively. The factor \( \beta_1 \) is regarded as a long-term factor, \( \beta_2 \) a short-term factor and \( \beta_3 \) medium-term factor. We fit the yield curves at each time point using Diebold-Li model to extract the latent factors. Once again, we assess the change in the values of these new variables by taking differences between values in time, i.e. computing \( y_t - y_{t-\tau} \) for successive time points \( t \) and some time period \( \tau \). We construct new variables by taking weekly, monthly, quarterly, 6-months and yearly differences. Furthermore, we compute volatility of these differences, and store them in additional new variables.

In addition, we include the transforms of the currency exchange rate series into the data set by calculating returns and volatility of returns as in the case of EQ_TRGD variable. This way we incorporate the traditional time series forecasting approach when the past values of the series are included to forecast the future values.

Lastly, we compute inter-country differences between variables, e.g. \( \text{USD\_EQ\_EY} - \text{SEK\_EQ\_EY} \), and include them in the data set. To illustrate this point, consider a simulation study of three standard normal random variables \( Y_i \sim N(0, 1), i = \{1, 2, 3\} \) with \( Y_1 \) and \( Y_2 \) being 90\% correlated (Pearson correlation coefficient), and \( Y_3 \) being uncorrelated with the other two, i.e. \( \hat{\rho}(Y_3, Y_j) = 0, j = \{1, 2\} \). We introduce a new variable \( Y_4 := Y_1 - Y_2 \). Figure 3.1 shows the hierarchical clustering results with average linkage and \( 1 - \rho \) as dissimilarity measure. One can clearly see that the difference of two highly correlated variables, captured by \( Y_4 \), forms its own cluster, as does \( Y_3 \). This difference can thus not be considered redundant, but perhaps would have been discarded due to the high degree of association between \( Y_1 \) and \( Y_2 \). In financial context the inter-country differences may add valuable information to the considered set of predictors.
3.1.2 Data Set Creation

The obtained number of predictor variables (or inputs to the model) after the transformations have been carried out is 234. As macro-economic series are typically recorded on a monthly basis, they are shorter than financial time series. Thus, after the transformations have been applied, the time series need to be aligned such that they match the dates of the currency price series, which we use as date benchmarking series. We provide a toy example to illustrate this point. Given a succession of dates of the benchmark series $D_1 < D_2 < D_3 < \cdots < D_n$ for a large integer $n$, if the values of a macro-economic series $M$ correspond to the following dates from the benchmark series $D_2 < D_{20} < D_{40}$, the aligned macro-series $M_a$ becomes a collection of values such that $M_a = \{NA, M(D_2), \ldots, M(D_{20}), M(D_{20}), \ldots, M(D_{40}), \ldots, M(D_{40})\}, |M_a| = n$. In addition, macro-economic series have two types of dates associated with them: a report date and a release date. Release date is the date that a value of such series corresponds to, and report date is when this value was recorded into the series. We use release dates for our analysis, and in the case when there is only report date available we approximate the release date by adding the mean value of the differences of available release dates to the report date. In practice, macro-economic series are transformed into step functions. The new data set now consists of daily values ranging from January 3, 2000 to
December 31, 2015, with 4173 values in total. As we wish to perform forecast one week ahead, we subset the data such that it only contains weekly values, which leaves us with a data set with time series consisting of 835 values.

3.2 Two-Step Approach

We carry out the analysis in a two-step fashion: variable selection and classification learning. Denote the data set $X$, i.e. the predictor variables of the model. Create a categorical variable $Y = (y_1, \ldots, y_t)$ representing the signs (denoted by "+1" and "-1") of the weekly differences on the currency price series up to date $t$. We wish to predict the sign of $y_{t+1}$. In addition, we use the original price series as well as the weekly differences and the signs of the weekly differences as predictors, a methodology that is in line with the theory of time series analysis briefly touched upon in Section 1.2 and 2.5. Thus, the forecasting problem is transformed into a classification problem, where the goal is to predict whether the returns on the currency rate will be positive or negative.

The first step is finding a subset of variables that we assume to be sufficient for modeling the currency price differences. This is done by applying a hierarchical clustering method with average linkage and two correlation based dissimilarity measures: $1 - \rho$ and $1 - s$, where $\rho$ is the linear Pearson’s correlation coefficient, and $s$ represents Spearman’s rank correlation coefficient. As mentioned in Section 2.1, clustering produces distinct groups of data, and the variable that is most correlated with the currency rate price series (Spearman correlation) is extracted from each group. The number of groups considered in the analysis is $\{5, 10, 15\}$.

The second step is to perform classification by training a random forest on the data set created with the sliding window approach as described in Section 2.5. The data set is split into a training and test set at random such that 80% of the data belongs to the training set and the rest to the test set. The size of the final data set $n_{\text{new}}$ depends on the size of the sliding window $sw$ according to the relation $n_{\text{sw}} := n - sw$. We test a sliding window of size 13, and no sliding window. The size $sw = 13$ corresponds to a quarter of a year in weeks. The random forest model is estimated for different number of bootstrap samples ranging from 500 to 5000. The error evaluation measure used is Gini index (Equation 2.2).

The overall evaluation measure of the two-step approach is the classification accuracy, also referred to as hit rate or classification rate, on the test set of size $n_{\text{test}}$:

$$\text{Classification accuracy} = 1 - \text{Classification error}$$

where classification error is defined as proportion of observations that have been assigned to the wrong class, i.e.
Classification error = \frac{1}{n_{\text{test}}} \sum_{i}^{n_{\text{test}}} 1(Y_i \neq \hat{Y}_i)

3.3 Curse of Dimensionality

It is a well known fact that the size of the data set has an effect on the performance of both supervised and unsupervised statistical learning methods. For a training set of a constant size, i.e. number of observations is fixed to \(N\), a large number \(p\) of predictors leads to an increase in the volume of the space spanned by the data. This induces sparsity in the data set, i.e. the amount of available data points is not enough to produce statistically significant results due to the large amount of predictors [11]. This is usually referred to as curse of dimensionality in literature. Clustering high dimensional data is also subject to this phenomenon, as it can be shown that the distance between observations approaches zero as the number of dimensions increases [11]. Data sets with dimensions such that \(p\) is close to \(N\) or \(p > N\) are typically described as subjects to curse of dimensionality.
Chapter 4

Results and Analysis

The obtained results are divided in two groups: clustering and decision tree learning. In Section 4.1 we present the clustering analysis results from an exploratory view point. In Section 4.2 we present the results obtained from implementing the two-step approach for forecasting the sign of the change in currency exchange price on a weekly horizon. In both parts, clustering partitions of \{5, 10, 15\} clusters are considered and the employed dissimilarity measures are \(1 - \rho\) and \(1 - s\).

4.1 Clustering

We perform hierarchical clustering on the predictor data set in order to uncover the underlying groups in the data and find a subset of predictors that will be used in the decision tree learning part of the analysis. Although we use the whole data set for the two-step approach, in this section we are concerned with exploratory analysis of the data set. We perform dynamic clustering as proposed by De Luca et al., described in Section 2.2, to get an insight in how clusters change with time. The employed cluster evaluation measure is purity (Equation 2.1) Figure 4.1 shows charts of evaluated purities for clustering on \{1, 2\} year basis, e.g. using the clustering results on the data corresponding to time period 2000-2001 as the benchmark for comparison of clustering results on the time period 2001-2002 for 1 year difference, and 2000-2002 with 2002-2004 for comparison over 2 year difference.
Figure 4.1: Cluster purity for clustering with one and two years difference. The charts to the left correspond to 5 clusters, middle charts to 10 clusters, and rightmost charts to 15 clusters. The black lines represent $1 - \rho$ dissimilarity measure, and the red $1 - s$. The x-axis represents time periods (years formatted 20XX = XX) that are compared to one time period back in time.

Inspection of the charts suggests that clusters change over time. The employed dissimilarity measures seem to produce similar clusterings for partitionings of 5 and 15 clusters. However, looking at 10 cluster purity measures, one sees that the two dissimilarities produce different clusters between 2006 and 2010. Table 4.1 shows the average value of purity for each considered case. The values suggest that the agreement in groupings present in the data between two consecutive periods of one and two years is in on average 50% when evaluated using purity.
Table 4.1: Average purity values for clustering in time with 1 and 2 year time differences and 5, 10, 15 clusters.

<table>
<thead>
<tr>
<th>Dissimilarity</th>
<th>Number of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>1 – ρ</td>
<td>0.55</td>
</tr>
<tr>
<td>1 – s</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Two years

<table>
<thead>
<tr>
<th>Dissimilarity</th>
<th>Number of clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td>1 – ρ</td>
<td>0.52</td>
</tr>
<tr>
<td>1 – s</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Next we investigate the proportions of country specific and type specific variables in three cluster partitions \{5,10,15\} when clustering is performed on the whole data set. Tables A.1-A.3 show the proportions of the macroeconomic and financial variables in the clusters, and Tables A.4-A.6 show the proportions of USD and SEK variables in the clusters, In addition, Tables A.4-A.6 show the proportion of the differences between the country specific variables. As expected, clusters are not formed around type and country to which variables belong. Rather, they are represented across all the clusters. In addition, it can be seen from the tables that larger proportion of the variables are found when the data is partitioned into 4-5 clusters, and the additional clusters contain only a small amount of the data set.

4.2 Decision Tree Learning

The two-step model has been implemented for different parameters. In the cases when sliding window methodology was implemented, the size of the sliding window is chosen to \(sw = 13\). The clustering has been performed using \(1 – \rho\) and \(1 – s\) as dissimilarity measure. The cases considered are the following:

- Partitioning the data into 5 clusters and training the model on 5 predictors using both no sliding window and sliding window
- Partitioning the data into 10 clusters and training the model on 10 predictors using both no sliding window and sliding window
- Partitioning the data into 15 clusters and training the model on 15 predictors using both no sliding window and sliding window
• Training a model on the whole data set containing the 231 transformations described in Chapter 3.1.2 and the 3 additional predictor series mentioned in Chapter 3.2 with no sliding window.

The latter test is used as a benchmark to evaluate the performance of the sliding window treatment of the data set and the clustering procedure. For each of the cases, we train a non-pruned tree classifier and a random forest classifier on ten different training sets. The benchmark for evaluation of model performance is the 50% prediction rate of the random walk model. In addition, the 50 to 60% hit rate of models trained with SVMs and ANNs described in 1.2 is regarded as guidelines for the performance of machine learning methods for currency exchange rate forecasting on a daily and weekly horizon.

Figure 4.2 shows boxplots of classification accuracy obtained from training a decision tree classifier on 10 different randomly sampled training sets. The obtained classification accuracy is surprisingly high with the majority of values being above 50%. However, the classification accuracies show a high degree of variability between different training sets as well different cases of study that are considered. It is however clear that the data sets obtained by clustering with $1 - s$ as dissimilarity measure tend to yield higher classification accuracies than those obtained with $1 - \rho$. The tree model with 5 predictors, $1 - s$ dissimilarity measure and sliding window yields the distribution of errors across different training sets that has the most variability with maximum value above 60%. The rightmost boxplot, representing the tree model trained on the full data set without sliding window, suggests a classification accuracy above 50%. Table D.1 summarizes the statistics presented in the boxplot 4.2.
Figure 4.2: Boxplots of the obtained classification accuracies obtained for 10 different training sets using a non-pruned tree classifier.

Next, the random forest approach is implemented in order to test whether there will be a reduce of variance and hence better classification rate. Figures 4.3 and B.1-B.2 show the evolution of the classification rate with increased number of bootstrap samples starting with 200 samples, i.e. the classifier is trained for different number of bootstrap samples $B$ and the classification accuracy is recorded for each random forest model. The figures suggests that a stable classification accuracy can be obtained already at low values of bootstrap samples, e.g. 200. The oscillations in the charts are a natural consequence of bootstrap which serves as basis for random forest technique. In addition, Figure 4.3 random forest classifier shows high variability with training set when data set is partitioned into 5 clusters and sliding window is implemented. However, for 10 and 15 clusters sliding window methodology yields slightly lower variation on the classification accuracy as shown in Figures B.1-B.2.
Figure 4.3: Random forest classification accuracy for a model trained on 5 predictor variables. The left charts represent $1 - \rho$ dissimilarity measure, and the right $1 - s$.

Figure 4.4 shows the boxplots of the classification accuracies averaged over the bootstrap samples for the 10 different training sets considered. It is seen from the charts that the classification accuracy is variable across the different training sets, with classification accuracies above 50%. Similar trend is obtained as for regular tree model, i.e. $1 - s$ dissimilarity measure yields a data set that produces models with higher accuracy than its $1 - \rho$ counterpart. The statistics presented in the boxplot are summarized in Table D.2. Table 4.2 summarizes the overall performance of non-pruned tree and random forest classifiers for the two tested dissimilarity measures.
As mentioned in Chapter 2.4, random forest comes at a cost of interpretability. For illustrative purposes, we have constructed a tree model on the whole data set without sliding window. We used pruning and found that the optimal tree size is 10 terminal nodes. The illustration of the tree model is provided in Appendix C. The hit rate of this particular model is 54.49%.
Chapter 5

Discussion and Conclusions

The obtained results are sensitive to different replications of the training set. As mentioned in Section 2.3, tree based methods have high variance, which makes them sensitive to new inputs. Similar results were obtained by Walczak when training an ANN model on different subsets of financial time series ranging from 1 year back in time to 16 years [34]. Our results indicate that the choice of training data has impact on the obtained model. However, since the choice of the training data is random, this phenomenon cannot be motivated by long or short term ranges of data.

The obtained classification rates are on average between 52 and 54 per cent, and the values range from 41 to 62%. The obtained results are comparable to the performance of machine learning methods such as SVMs and ANNs on this type of problem. The instability in the results can be induced by multiple factors. We have seen that financial data is highly noisy and hard to build robust models on. In time series methodology, one removes trend and seasonality patterns in data prior to modeling. This work focuses on non-parametric methods which do not make any explicit assumptions about the data. Hence, the possible seasonal and trend patterns have not been modeled explicitly, but rather assumed to be handled by the data transformations described in Chapter 3. In their key work on currency exchange rate forecasting, Meese and Rogoff perform forecasting on seasonally unadjusted data, which they motivate by not using information that in reality is unavailable at the time of forecast [23].

The chosen size of the sliding window does not show any clear indication on having an impact of the model performance. Due to the decision of performing the analysis on the subset of the data based on weeks, there has been a significant reduction in the number of available measurements in time, i.e. data points. This has led to a limitation in the possible sliding window size employed in the analysis to 13 weeks. This window size corresponds to a quarter of a year, however, in financial context, both 6 months and yearly periods are of interest. An increased window size leads to inclusion of
larger portion of time dependence into the data used for building the random forest model, but also an increase in the dimension of the data set in terms of the number of considered predictors and a reduction of the number of available data points, i.e. sparsity. An increased time dependency possibly yields more robust models that do not change with changes of the training set, however, on this particular data set we consider it impossible due to the curse of dimensionality briefly touched upon it Chapter 2.

In conclusion, the dissimilarity measure $1 - s$ seems to consistently yield better model performance than $1 - \rho$ on all data sets considered in the analysis. Although the random forest approach leads to a slight increase in model performance, it might not be sufficiently large to compensate for the high computational cost of the method. The obtained results indicate however that there is no parameterization in terms of number of clusters and type of dissimilarity measure that can be regarded as the best performing model. Despite the instability of the obtained results, it is necessary to point out that this analysis yields a prediction accuracy of 50% and higher for most of the considered training sets, which is more than the simple random walk model.
Appendix A

Distribution of Clusters by Type and Country

Table A.1: Proportion of macro-economic and financial variables in each of the 5 clusters.

<table>
<thead>
<tr>
<th></th>
<th>$1 - \rho$</th>
<th>$1 - s$</th>
<th>$1 - \rho$</th>
<th>$1 - s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.06</td>
<td>0.22</td>
<td>0.21</td>
<td>0.30</td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>0.32</td>
<td>0.27</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>0.24</td>
<td>0.22</td>
<td>0.39</td>
<td>0.18</td>
</tr>
<tr>
<td>4</td>
<td>0.55</td>
<td>0.18</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>5</td>
<td>0.06</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: Proportion of macro-economic and financial variables in each of the 10 clusters.

<table>
<thead>
<tr>
<th></th>
<th>$1 - \rho$</th>
<th>$1 - s$</th>
<th>$1 - \rho$</th>
<th>$1 - s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.21</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>0.32</td>
<td>0.27</td>
<td>0.32</td>
</tr>
<tr>
<td>3</td>
<td>0.21</td>
<td>0.11</td>
<td>0.18</td>
<td>0.08</td>
</tr>
<tr>
<td>4</td>
<td>0.55</td>
<td>0.13</td>
<td>0.30</td>
<td>0.12</td>
</tr>
<tr>
<td>5</td>
<td>0.03</td>
<td>0.11</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>6</td>
<td>0.05</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.01</td>
<td>0.09</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>8</td>
<td>0.01</td>
<td>0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.06</td>
<td>0.01</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>10</td>
<td>0.05</td>
<td>0.01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table A.3: Proportion of macro-economic and financial variables in each of the 15 clusters.

<table>
<thead>
<tr>
<th></th>
<th>$1-\rho$</th>
<th></th>
<th>$1-s$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Macro</td>
<td>Fin</td>
<td>Macro</td>
</tr>
<tr>
<td>1</td>
<td>0.17</td>
<td></td>
<td>0.12</td>
</tr>
<tr>
<td>2</td>
<td>0.03 0.17</td>
<td></td>
<td>0.09 0.19</td>
</tr>
<tr>
<td>3</td>
<td>0.18 0.10</td>
<td></td>
<td>0.18 0.08</td>
</tr>
<tr>
<td>4</td>
<td>0.55 0.13</td>
<td></td>
<td>0.30 0.12</td>
</tr>
<tr>
<td>5</td>
<td>0.12 0.14</td>
<td></td>
<td>0.15 0.13</td>
</tr>
<tr>
<td>6</td>
<td>0.03 0.11</td>
<td></td>
<td>0.12 0.13</td>
</tr>
<tr>
<td>7</td>
<td>0.01</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>8</td>
<td>0.05</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>9</td>
<td>0.01</td>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td></td>
<td>0.03</td>
</tr>
<tr>
<td>11</td>
<td>0.03</td>
<td></td>
<td>0.03</td>
</tr>
<tr>
<td>12</td>
<td>0.06 0.01</td>
<td></td>
<td>0.09 0.02</td>
</tr>
<tr>
<td>13</td>
<td>0.05</td>
<td></td>
<td>0.05</td>
</tr>
<tr>
<td>14</td>
<td>0.01</td>
<td></td>
<td>0.03 0.01</td>
</tr>
<tr>
<td>15</td>
<td>0.03 0.01</td>
<td></td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table A.4: Proportion of USD, SEK and difference variables together with currency price series in each of the 5 clusters.

<table>
<thead>
<tr>
<th></th>
<th>$1-\rho$</th>
<th></th>
<th>$1-s$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>USD</td>
<td>SEK</td>
<td>USDSEK</td>
</tr>
<tr>
<td>1</td>
<td>0.31 0.35</td>
<td>0.24 0.29</td>
<td>0.08 0.03</td>
</tr>
<tr>
<td>2</td>
<td>0.32 0.33</td>
<td>0.28 0.32</td>
<td>0.29 0.09</td>
</tr>
<tr>
<td>3</td>
<td>0.11 0.22</td>
<td>0.17 0.26</td>
<td>0.37 0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.25 0.10</td>
<td>0.32 0.12</td>
<td>0.15 0.01</td>
</tr>
<tr>
<td>5</td>
<td>0.01 0.11</td>
<td>0.11 0.11</td>
<td>0.01 0.11</td>
</tr>
</tbody>
</table>
Table A.5: Proportion of USD, SEK and difference variables together with currency price series in each of the 10 clusters.

<table>
<thead>
<tr>
<th></th>
<th>1 − ρ</th>
<th></th>
<th>1 − s</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD</td>
<td>SEK</td>
<td>USD</td>
<td>USD</td>
</tr>
<tr>
<td>0.29</td>
<td>0.22</td>
<td>0.06</td>
<td>0.29</td>
</tr>
<tr>
<td>0.32</td>
<td>0.28</td>
<td>0.29</td>
<td>0.33</td>
</tr>
<tr>
<td>0.11</td>
<td>0.08</td>
<td>0.17</td>
<td>0.06</td>
</tr>
<tr>
<td>0.24</td>
<td>0.25</td>
<td>0.09</td>
<td>0.21</td>
</tr>
<tr>
<td>0.08</td>
<td>0.20</td>
<td>0.10</td>
<td>0.12</td>
</tr>
<tr>
<td>0.07</td>
<td>0.06</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.01</td>
<td>0.06</td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>0.11</td>
<td></td>
<td>0.11</td>
<td></td>
</tr>
</tbody>
</table>

Table A.6: Proportion of USD, SEK and difference variables together with currency price series in each of the 15 clusters.

<table>
<thead>
<tr>
<th></th>
<th>1 − ρ</th>
<th></th>
<th>1 − s</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD</td>
<td>SEK</td>
<td>USD</td>
<td>USD</td>
</tr>
<tr>
<td>0.21</td>
<td>0.21</td>
<td>0.05</td>
<td>0.14</td>
</tr>
<tr>
<td>0.28</td>
<td>0.12</td>
<td>0.07</td>
<td>0.29</td>
</tr>
<tr>
<td>0.11</td>
<td>0.08</td>
<td>0.14</td>
<td>0.06</td>
</tr>
<tr>
<td>0.24</td>
<td>0.25</td>
<td>0.09</td>
<td>0.21</td>
</tr>
<tr>
<td>0.04</td>
<td>0.15</td>
<td>0.21</td>
<td>0.03</td>
</tr>
<tr>
<td>0.08</td>
<td>0.20</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>0.07</td>
<td>0.06</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>0.07</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>0.11</td>
<td></td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>0.03</td>
<td></td>
<td>0.03</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Appendix B

Random Forest Classification Accuracies

![Graphs showing random forest classification accuracy](image)

Figure B.1: Random forest classification accuracy for a model trained on 10 predictor variables. The left charts represent $1 - \rho$ dissimilarity measure, and the right $1 - s$. 
Figure B.2: Random forest classification accuracy for a model trained on 15 predictor variables. The left charts represent $1 - \rho$ dissimilarity measure, and the right $1 - s$. 

39
Appendix C

Example of a Tree Model

Figure C.1: Graphical representation of a pruned tree model with optimal size of 10 leaf nodes. This model’s hit rate is 54.49%.
Appendix D

Summary of Classification Accuracies

Table D.1: Minimum value, first quartile, second quartile (median), third quartile, maximum value, mean and standard deviation of the obtained classification accuracies with non-pruned tree classifier across 10 different training sets.

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>max</th>
<th>mean</th>
<th>$\bar{\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 - \rho$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>- 0.51</td>
<td>0.53</td>
<td>0.54</td>
<td>0.56</td>
<td>0.57</td>
<td>0.54</td>
<td>0.02</td>
</tr>
<tr>
<td>13</td>
<td>0.44</td>
<td>0.48</td>
<td>0.51</td>
<td>0.52</td>
<td>0.53</td>
<td>0.50</td>
<td>0.03</td>
</tr>
<tr>
<td>10</td>
<td>- 0.44</td>
<td>0.50</td>
<td>0.52</td>
<td>0.54</td>
<td>0.56</td>
<td>0.52</td>
<td>0.03</td>
</tr>
<tr>
<td>13</td>
<td>0.48</td>
<td>0.50</td>
<td>0.51</td>
<td>0.55</td>
<td>0.56</td>
<td>0.52</td>
<td>0.03</td>
</tr>
<tr>
<td>15</td>
<td>- 0.47</td>
<td>0.49</td>
<td>0.51</td>
<td>0.53</td>
<td>0.59</td>
<td>0.52</td>
<td>0.05</td>
</tr>
<tr>
<td>13</td>
<td>0.49</td>
<td>0.50</td>
<td>0.51</td>
<td>0.52</td>
<td>0.55</td>
<td>0.52</td>
<td>0.02</td>
</tr>
<tr>
<td>$1 - s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>- 0.49</td>
<td>0.52</td>
<td>0.55</td>
<td>0.56</td>
<td>0.56</td>
<td>0.54</td>
<td>0.02</td>
</tr>
<tr>
<td>13</td>
<td>0.41</td>
<td>0.48</td>
<td>0.52</td>
<td>0.53</td>
<td>0.61</td>
<td>0.51</td>
<td>0.06</td>
</tr>
<tr>
<td>10</td>
<td>- 0.50</td>
<td>0.51</td>
<td>0.52</td>
<td>0.54</td>
<td>0.54</td>
<td>0.52</td>
<td>0.02</td>
</tr>
<tr>
<td>13</td>
<td>0.48</td>
<td>0.49</td>
<td>0.52</td>
<td>0.55</td>
<td>0.58</td>
<td>0.52</td>
<td>0.04</td>
</tr>
<tr>
<td>15</td>
<td>- 0.49</td>
<td>0.51</td>
<td>0.53</td>
<td>0.56</td>
<td>0.60</td>
<td>0.53</td>
<td>0.04</td>
</tr>
<tr>
<td>all</td>
<td>0.50</td>
<td>0.51</td>
<td>0.53</td>
<td>0.54</td>
<td>0.57</td>
<td>0.53</td>
<td>0.02</td>
</tr>
</tbody>
</table>
Table D.2: Minimum value, first quartile, second quartile (median), third quartile, maximum value, mean and standard deviation of the obtained classification accuracies with random forest classifier across 10 different training sets.

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>max</th>
<th>mean</th>
<th>$\hat{\sigma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 - \rho$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>0.46</td>
<td>0.47</td>
<td>0.49</td>
<td>0.49</td>
<td>0.51</td>
<td>0.48</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>0.45</td>
<td>0.51</td>
<td>0.55</td>
<td>0.56</td>
<td>0.62</td>
<td>0.54</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>0.48</td>
<td>0.51</td>
<td>0.54</td>
<td>0.56</td>
<td>0.58</td>
<td>0.53</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.52</td>
<td>0.52</td>
<td>0.54</td>
<td>0.55</td>
<td>0.57</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.46</td>
<td>0.48</td>
<td>0.50</td>
<td>0.53</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>0.50</td>
<td>0.51</td>
<td>0.54</td>
<td>0.57</td>
<td>0.54</td>
</tr>
<tr>
<td>$1 - s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>0.50</td>
<td>0.53</td>
<td>0.54</td>
<td>0.56</td>
<td>0.58</td>
<td>0.54</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>0.45</td>
<td>0.51</td>
<td>0.56</td>
<td>0.57</td>
<td>0.62</td>
<td>0.55</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>0.51</td>
<td>0.53</td>
<td>0.55</td>
<td>0.59</td>
<td>0.60</td>
<td>0.55</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>0.53</td>
<td>0.53</td>
<td>0.55</td>
<td>0.56</td>
<td>0.57</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.49</td>
<td>0.51</td>
<td>0.53</td>
<td>0.56</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>0.51</td>
<td>0.53</td>
<td>0.55</td>
<td>0.56</td>
<td>0.59</td>
</tr>
<tr>
<td>all</td>
<td></td>
<td>0.48</td>
<td>0.52</td>
<td>0.52</td>
<td>0.57</td>
<td>0.61</td>
<td>0.54</td>
</tr>
</tbody>
</table>
Bibliography


