LISBON
SCHOOL OF
ECONOMICS &
MANAGEMENT
UNIVERSIDADE DE LISBOA

Master
Mathematical Finance

Master's Final Work
Dissertation

Cluster analysis of financial time series

Maria Inês Costa Correia

November 16, 2020

LISBON SCHOOL OF ECONOMICS & MANAGEMENT
UNIVERSIDADE DE LISBOA

Master
Mathematical Finance

Master's Final Work
Dissertation

Cluster analysis of financial time series

Maria Inês Costa Correia

Supervision:

Professor Marianne Clausel
Professor Onofre Simões

November 16, 2020

Abstract

This thesis applies the Signature method as a measurement of similarities between two time-series objects, using the Signature properties of order 2, and its application to Asymmetric Spectral Clustering. The method is compared with a more Traditional Clustering approach where similarities are measured using Dynamic Time Warping, developed to work with time-series data. The intention for this is to consider the traditional approach as a benchmark and compare it to the Signature method through computation times, performance, and applications. These methods are applied to a financial time series data set of Mutual Exchange Funds from Luxembourg.

According to Chevyrev and Kormilitzin (2016), the Signature transformation converts the data into multi-dimensional paths using various embedding algorithms and then proceeds to the computation of individual terms of the Signature which summarise certain information contained in the data. Levin, Lyons, and Ni (2013) introduced the possibility of its use to understand financial data and the potential this approach has for machine learning and prediction.

In our work, the approach is applied to clustering, where the goal is to build homogeneous groups of observations and discover hidden patterns in data. Performing clustering evolves at least three steps: (1) selecting a measure able to quantify the similarity between objects; (2) choosing the right method to perform clustering, according to the specific problem; (3) setting the number of desired clusters. The order of the steps depends on the algorithm chosen to cluster the data.

After the literature review, we introduce the Dynamic Time Warping method and the Signature method. We continue with the explanation of Traditional Clustering approaches, namely k-Means, and Asymmetric Clustering techniques, namely the k-Axes algorithm, developed by Atev (2011). The last chapter is dedicated to the Practical Research where the previous methods are applied to the data set. Results confirm that the Signature method has indeed potential for machine learning and prediction, as suggested by Levin, Lyons, and Ni (2013).

Key words: Signature, Dynamic Time Warping, Clustering, Machine Learning.

Resumo

Esta dissertação aplica o método da Signature como medida de similaridade entre dois objetos de séries temporais usando as propriedades de ordem 2 da Signature e aplicando-as a um método de Clustering Asimétrico. O método é comparado com uma abordagem de Clustering mais tradicional, onde a similaridade é medida usando Dynamic Time Warping, desenvolvido para trabalhar com séries temporais. O intuito é considerar a abordagem tradicional como benchmark e compará-la ao método da Signature através do tempo de computação, desempenho e algumas aplicações. Estes métodos são aplicados num conjunto de dados de séries temporais financeiras de Fundos Mútuos do Luxemburgo.

De acordo com Chevyrev e Kormilitzin (2016), a transformação da Siganture consiste em converter os dados em caminhos multidimensionais por meio de vários algoritmos de incorporação e, em seguida, procede ao cálculo dos termos individuais da Signature que resumem certas informações contidas nos dados. Levin, Lyons e Ni (2013) introduziram a possibilidade do seu uso para entender dados financeiros e o potencial que esta abordagem tem para Machine Learning e Previsão.

Esta abordagem é a aplicada a clustering, onde o objetivo é criar grupos homogéneos de observações e descobrir padrões escondidos nos dados.

A execução de clustering envolve pelo menos três etapas: (1) selecionar uma medida capaz de quantificar a similaridade entre os objetos; (2) escolher o método certo para realizar o clustering, de acordo com o problema específico; (3) definir o número de clusters desejados. A ordem das etapas depende do algoritmo escolhido para agrupar os dados.

Após a revisão da literatura, apresentamos o método Dynamic Time Warping e o método da Signature. Prossegue-se com a explicação das abordagens de Clustering Tradicional, nomeadamente k-Means, e Clustering Espectral Assimétrico, nomeadamente k-Axes, desenvolvido por Atev (2011). O último capítulo é dedicado à Investigação Prática onde os métodos anteriores são aplicados ao conjunto de dados. Os resultados confirmam que o método da Signature têm efectivamente potencial para machine learning e previsão, como sugerido por Levin, Lyons and Ni (2013).

Key words: Signature, Dynamic Time Warping, Clustering, Machine Learning.

Acknowledgements

I would like to express my special thanks to Professor Antoine Lejay for his availability to contributing to the development of this work since its beginning and his active support.

To Professor Marianne Clausel for allowing me to be part of this project and for introducing me to the subject of Machine Learning.

A very special thanks to Professor Virginie Terraza for providing the data used in the practical research, without it, the development of this work would not be possible.

To my parents for allowing me to study abroad and to extend that period so I could be part of this work.

Last but not least, my most sincere gratitude to Professor Onofre Simões for his endless motivation and great support given to me and my project throughout its several stages. Most of all I want to thank him for his patience and dedication. I could not have done it without such important guidance.

Contents

Chapter 1

Introduction

Following the studies carried out during the Masters in Mathematical Finance at ISEG, I had the opportunity to enter the double degree program in Financial Engineering at the University of Lorraine. The combination of the two programs was very significant, as it allowed me to balance an extensive and theoretical foundation with deep practical training in some of the tools commonly used in the financial sector.

From this combination, one topic seemed quite interesting to me, as the theme for my Master's Final Work: machine learning (mostly unsupervised learning) and its application to stochastic finance. Accordingly, the main purpose of this work is to progress in the knowledge of the subject and this chapter serves as an introductory note to the dissertation

The world we live in is characterized by the massive amounts of data collected every day. Very often it is necessary to find some patterns in such data, a purpose that requires analysts to use appropriate tools, to make it possible for them to extract the relevant information.

Besides the WWW (World Wide Web), other systems store daily data from businesses, medicine, science, engineering, etc. According to Han et al. (2012) this explosive growth of the available volume of data is a result of the computerization of our society and the fast development of powerful data collection and storage tools. Businesses worldwide generate gigantic data sets, for instance, sales transactions, stock trading records, product descriptions, sales promotions, company profiles and performance, and customers' feedback.

1.1 Data Mining and Machine Learning

Such an impressive growth of available data brought the need to create tools that can automatically uncover the valuable information it contains, transforming it into organized knowledge. This led to the birth of data mining. The term Data Mining became prevalent amongst the database communities in the 1990s. Data Mining owes its origin to KDD (Knowledge Discovery in Databases). Since Data Science can be viewed as an area and Data Mining is mostly a technique, it can be somehow considered a subset of Data Science. Data mining is the discovery of structures and patterns in large and complex data sets. There are two main aspects related to it: pattern detection, finding trends previously not known and making data more usable, and model building.

Model building in data mining is very similar to statistical modeling. Pattern detection seeks anomalies or small local structures in data. Indeed, one view of many large-scale data mining activities is that they primarily constitute filtering and data reduction. Although some sub-disciplines of statistics have examined special cases of this problem, the bulk of the work on pattern detection (to date) has been computational, with an emphasis on algorithms (Hand and Adams, 2015).

Most of the algorithms used in data mining are from a field known as machine learning. The confusion between these two areas is very common. Machine learning is a subset of Artificial Intelligence and its purpose is to learn how to analyze data sets and perform tasks without human intervention. Although making use of machine learning algorithms to conduct useful and proper analysis, data mining always requires human intervention. We can say data mining needs machine learning, but the opposite does not necessarily happen.

Machine learning algorithms and techniques have been applied to many different fields, namely to Finance, where the availability of large amounts of data offers the possibility to explore even more the use of the methods, namely, to increase business performance.

One of the most popular applications is in fraud detection, bringing the possibility to separate malicious from genuine credit card transactions. Once the model is constructed, when a new transaction is made, is possible to classify it into fraudulent or not fraudulent (class labels) with an accuracy that goes further beyond traditional methods. This type of strategy is called classification. It is certainly a subject of great interest for the financial institutions, allowing them to decrease their losses.

1.2 Literature Review

Regarding the techniques that are applied in pattern detection, clustering has been considered one of the most important to capture the natural structure of data. The cluster analysis aims to build homogeneous groups of observations (clusters) representing realisations of some random variable. Clustering is often used as a preliminary step for data exploration, the goal being to identify particular patterns that have some convenient interpretation. (Jacques and Preda, 2014).

Performing clustering evolves at least three steps: (1) selecting a measure able to quantify the similarity between objects; (2) choosing the right method to perform clustering, according to the specific problem; (3) setting the number of desired clusters. The order of the steps depends on the algorithm chosen to cluster the data.

Unlike classification, these types of methods analyze the data without looking into class labels. The common principle of all clustering techniques is to maximize the intraclass similarity and minimize the interclass similarity, so objects in the same cluster are more similar than others in different clusters. Once this similarity is computed we can extract rules representing each group (Han et al. 2012).

Most cluster algorithms were developed to work with similarity measures of static data that does not change over time. However, with the ability to store more and more quantities of data, algorithms have been applied to dynamic data sets, such as time-series, to gain insights about what causes data to change.

For instance, Dynamic Time Warping (DTW) is a dynamic algorithm that quantifies the similarity between two series, and it was created to work specifically with time-series data, overcoming some of the limitations of other distances considered inappropriate to work with this type of data set. This is a technique used to find an optimal alignment between two given time series across time points under certain restrictions. It is preferred to the Euclidean distance measures over all the time points because Euclidean distances may fail to produce an intuitively correct measure of similarity between two sequences that are very sensitive to small distortions in time (Maharaj et al., (2019)). However, Sardá-Espinosa (2015) highlights that even bringing some advantages, the computation time is quite expensive.

The purpose of our work falls into the subject of similarity measures. We use an approach for the computation of distances between two objects, the so-called Signature of Rough paths, and its application to cluster analysis. According to Humbly and Lyons (2010), the mathematical sense of Signature is a truthful transformation of a multidimensional time series.

The theory of rough paths was brought to the study of non-parametric statistics on streamed data and particularly to the problem of regression. Levin, Lyons, Ni et al. (2016), introduced the possibility of using this transformation to understand financial data and the great potential this method has for machine learning and prediction.

The main contribution of our work is to apply the same logic for cluster analysis, transforming multidimensional time-series and using its geometric meaning to compute the similarity between two objects. We compare our method with DTW computation and its use on a well-established clustering algorithm called k-Means.

The Signature transformation method has recently gained attention due to its connectivity with Lyon's theory of rough paths. According to Chevyrev and Kormilitzin (2016), the idea consists of converting the data into multi-dimensional paths using various embedding algorithms and then proceed to the computation of individual terms of the signature which summarise certain information contained in the data. The Signature thus transforms raw data into a set of features that are used in machine learning tasks.

The recent applicability of rough path theory to machine learning and time series analysis has gained attention also in the financial field, see Gyurkó et al. (2014). These authors applied the Signature transformation, to financial streams to extract information, such as hidden patterns in trading strategies. Their work showed that the Signature method has the advantage of being able to represent data using a small number of features that capture its most important properties without parametrization or statistical modeling. The paper illustrates how a very small number of coefficients obtained from the Signature of financial data can be sufficient to classify data for subtle underlying features and to make useful predictions (Gyurkó et al. (2014)).

In our work, we try to explore this applicability to create a more accurate link between two objects. We use the advantages of the Signature method and its ability to extract information from the data. Then we use the geometric intuition of the first two levels (Chevyreva and Kormilitzin, et al. 2016) that comes from the computation of the Lévy Area for each pair of dimensional paths, as the foundation of our similarity measures. Next, we compute the asymmetric similarity matrix that stores information of all pairs of data. We finally use this matrix as input to our cluster algorithm.

When selecting the clustering algorithm, we considered Atev (2011) work due to his implementation of an algorithm capable of dealing with asymmetric similarity matrices, without the need for the artificial symmetrisation step. The use of asymmetric matrices brings much value to our research because it prevents loss of information when compared to standard processes that require symmetric matrices as input.

These processes and methods are applied to a time-series data set of Mutual and Trade Exchange funds of Luxembourg, with weekdays observations of Net Asset Value (NAV) from 2005 until 2019, provided by Professor Virginie Terraza from University of Luxembourg. This particular set of data has been chosen because of the importance NAV prices have for investors, as they provide the value "per-share" of each fund, making the valuation procedure easier. We intent to be able to identify clusters and patterns in different types of assets and how to price oscillations react in terms of different variables and economic factors.

1.3 Structure of the text

The structure of the text is as follows. In Chapter 2, we explain the importance of selecting an appropriate distance measure. We introduce lock-step measures and discuss the limitations that come from these long-established distances. We further introduce elastic measures, created to work specifically with time-series data, namely DTW. The final section of the chapter is dedicated to our main contribution, the Signature of rough paths, and the potential use of its geometric meaning to compute the distance, introducing a new approach to measure similarities between two objects.

Chapter 3 addresses clustering techniques. It focuses on the approach of partitional clustering, specifically k-Means clustering, and on spectral clustering techniques, a promising alternative to classical clustering approaches. Chapter 4 presents the practical case research and the results. Chapter 5 concludes.

Chapter 2

Distance measures

2.1 Finding a relevant way to measure similarities

In applications, such as clustering, there is the need to assess how identical (or different) an object is in comparison to another. For example, in marketing campaigns, it is important to know the target customer. By looking at people's characteristics and traits, becomes easier to know what type of campaign attracts people with a certain profile. Clustering algorithms allow for grouping individuals with similar characteristics and likelihood to purchase a certain product/service. Once the clusters are defined, it is possible to run tests and refine the message for each group, in order to get better returns on the marketing investment.

A cluster is a collection of data objects such that the objects within a cluster are similar to one another and dissimilar to the objects in other clusters (Han et al., 2012).

For any clustering algorithm, the starting point is to find an appropriate distance measure and compute the distance between two objects.

The choice of distance is fundamental because it will define how similarities are calculated. This is particularly important in the presence of dynamic data, such as time series. Undependable of the distance's choice, it is expected that the distance between objects within a cluster is minimized and the distance between objects in different clusters is maximized.

In the following sections, we introduce shaped-based measures, this means they compare the overall shape of the time-series based on its actual values. Shaped-based measures can be divided into two subgroups: lock-step measures and elastic measures.

We introduce as lock-step measures the well-known Minkowski distance followed by the concept of elastic measures, making specific reference to Dynamic Time Warping.

Finally, we introduce the Signature of Rough paths as a measurement of similarities between time series data. This is our main contribution and focus of the thesis.

2.2 Lock-step measures

In this section, we introduce the definition of Minkowski distance, a lock-step measure.

These measures require the two time series, that are going to be compared, to be of equal length, and compare points localized in the same moment in time. That is, they compare point i of x with point i of y, being x, y time-series of equal length.

2.2.1 Minkowski distance

The Minkowski distance is defined by

$$D_{L_p}(X,Y) = \Big(\sum_{i=1}^{n} \mid x_i - y_i \mid^p \Big)^{\frac{1}{p}} \tag{2.1}$$

This is the L_p-norm of the difference between two equal length vectors. The particular and most used case is the Euclidean distance, where p=2.

Euclidean distance

$$D_{L_2}(X,Y) = \Big(\sum_{i=1}^{n} \mid x_i - y_i \mid^2 \Big)^{\frac{1}{2}} \tag{2.2}$$

L_p norms are very intuitive, free of parameters, and take linear time, meaning time complexity increases at most linearly with the size of the time-series. On the other hand, they have some limitations, such as: high sensitivity to small distortions in the time axis (Keogh and Kasetty, 2003), noise, and outliers (Lin et. al, 2002) because fixed pairs of data are compared. For these reasons, they are called lock-step measures.

The Euclidean distance follows the following properties (Han et al.,2012):

- Non-negativity: $D(X,Y) \geq 0$: Distance is a non-negative number.
- Identity of indiscernible: $D(X,X) = 0$: The distance of an object to itself is 0.
- Symmetry: $D(X,Y) = D(Y,X)$: Distance is a symmetric function.
- Triangle inequality: $D(X,Y) \leq D(X,W) + D(W,Y)$: Going directly from object X to object Y in space is not taking a longer distance than making a detour over any other object W.

2.3 Elastic measures

Elastic distance measures are designed to work with time-series data, they create a non-linear mapping to align the series and allow comparison of one-to-many points (Abanda, Mori, Lozano, et al.(2018)), allowing for warping in time robustness, when dealing with outliers.

2.3.1 Dynamic Time Warping (DTW)

Dynamic Time Warping is a technique used to find an optimal alignment between two given time series across time points, under certain constraints. Was initially used by the data mining community to work with time-series data, in order to overcome the shortcomings of the Euclidean distance.

In DTW, the time series are warped in a nonlinear fashion to match each other. DTW was introduced to the data mining community by Berndt and Clifford (1994). Although they demonstrate the utility of the approach, they acknowledge that the algorithm's time complexity is a problem and has limitations for very large databases of time series. Many authors have used dynamic time warping and variations of it to compare or to cluster time series and it has been extensively used in data mining (Maharaj et al., 2019).

DTW is considered an elastic distance measure since the non-linearly warp between two time-series allows us to deal with time deformations and changeable speeds of data, depending on time. Figure 2.1 shows the alignment between two time series, using the algorithm. The blue line gives an example of a mapping between points of both series, the initial and final points must match but the others are warped in time in order to find the better matches.

As referred, this dynamic programming algorithm compares two series and tries to find the optimum warping path between them (under certain constraints).

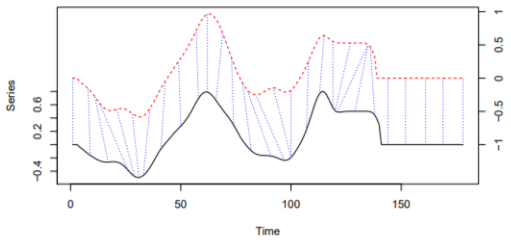


Figure 2.1: Example of a sample alignment given by the DTW algorithm between two series. Source: A. Sardá-Espinosa, *Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package* (2017, p.5)

In the next paragraphs, a brief description of the algorithm is given. Let $\mathrm{x}^{(1)}$ and $\mathrm{x}^{(2)}$ be two time-series, in order to determine the DTW, a $(n \times m)$ local cost matrix (LCM) is determined for the pair of series that we want to compare. The distance between $x_i^{(1)}$ and $x_j^{(2)}$, with $i = 1, ..., n$ and $j = 1, ..., m$ is given by the element (i, j) of the matrix.

The L_p norm between $x_i^{(1)}$ and $x_j^{(2)}$ is computed with the following equation:

$$LCM(i,j) = \Big(\sum_u \mid x_i^{(1)u} - x_j^{(2)u} \mid^p \Big)^{1/p} \tag{2.3}$$

In the next step, the algorithm iteratively finds the distance that minimizes the alignment between x$^{(1)}$ and x$^{(2)}$ passing through the LCM.
Starts at LCM(1,1) until LCM(n,m) and aggregates the cost.

At each step, the direction found by the algorithm is the one that increases less the cost, considering the imposed constraints. Defining $\Phi = \phi_1, ..., \phi_k$ as the set containing all the points that fall on the optimum path, then the final distance would be computed with following equation:

$$DTW_p(x,y) = \Big(\sum \frac{m_\phi lcm(k)^p}{M_\phi}\Big)^p, \forall k \in \phi \tag{2.4}$$

where $m\phi$ is a per-step weighting coefficient and $M\phi$ is the corresponding normalization constant (Giorgino (2009)).

The path crosses the LCM under the following constraints:

1. Boundary condition: $\phi_1 = (1,1)$ and $\phi_k = (n,m)$

2. Continuity constraint: if $\phi_p = (i,j)$ then ϕ_{p+1} is either $(i+1,j)$, $(i,j+1)$ or $(i+1,j+1)$ element of the matrix LCM., with $p = 1, ..., k-1$ and $i = 1, ..., n-1$ and $j = 1, ..., m-1$.

3. Monotonicity: monotonically increasing steps, since they represent the different points in time.

2.4 Our main contribution: *Path Signature*

Our main contribution consists in the use of the Signature, a non-parametric way for extracting features from data, to measure similarities between time-series.

The mathematical community has been giving attention to the Signature, in part for its association with Lyons' theory of rough paths (Chevyrev and Kormilitzin (2016)).

This section briefly introduces the definition of Signature and presents its properties of order two (if the reader wishes to explore the subject in more depth we advise him to read Chevyrev and Kormilitzin (2016)). The motivation for this approach is the fact that the Signature is an object related to a path and captures many important analytic and geometric properties of the path itself. For these reasons, we consider it a good candidate for measuring distances between two objects.

2.4.1 Foundations

The mathematical properties of iterated integrals of piece-wise regular multi-dimensional paths were first studied by K.T. Chen (1957). Hambly and Lyons (2010) extended these results to continuous paths of bounded variation.

Iterated integrals of continuous multi-dimensional paths naturally arise in the Taylor expansion of controlled ordinary differential equations. Moreover, several numerical approximations of the solution to stochastic differential equations (SDEs) are based on the iterated (stochastic) integrals of Brownian motion (Gyurkó et al., 2014).

As mentioned before, the Signature transformation consists of mapping multi-dimensional paths to the sequence of their iterated integrals. The application of this transformation to time series was established by Levin, Lyons, and Ni (2016).

2.4.2 Path Integrals

A path integral is most commonly introduced against a fixed function *f*.
For a one-dimensional path $Y : [a, b] \mapsto \mathbb{R}$ and a function $f : \mathbb{R} \mapsto \mathbb{R}$, an integral path of Y against f is defined by

$$\int_a^b f(Y_t) dY_t = \int_a^b \frac{dY_t}{dt} dt \tag{2.5}$$

where $f(Y_t)$ is a real-valued path.

2.4.3 Signature of a path

Define a path $Y : [a, b] \mapsto \mathbb{R}^d$, where $Y = (Y_t^1, ..., Y_t^d)$ and each $Y^i : [a, b] \mapsto \mathbb{R}$ is a real-valued path. For any single index $i \in \{1, ..., d\}$ the Signature of the path Y can be defined as

$$S(Y)^i_{a,t} = \int_{a<s<t} dY^i_s = Y^i_t - Y^i_1 \tag{2.6}$$

This quantity corresponds to the increment of the i-th coordinate of the path at time $t \in [a, b]$.

For the double index $i, j \in \{1, ..., d\}$, the Signature is defined by the double-iterated integral

$$S(Y)^{i,j}_{a,t} = \int_{a<s<t} S(Y)^i_{a,s} dY^j_s = \int_{a<r<s<t} dY^i_r dY^j_s \tag{2.7}$$

Recursively, for the collection of indexes $i_1, ..., i_k \in \{1, ..., d\}$ the Signature is defined as

$$S(X)^{i_1,...,i_k}_{a,t} = \int_{a<t1<...<t_k<t} dX^{i_1}_{t_1} ... dX^{i_k}_{t_k} \tag{2.8}$$

Definition (Signature): The Signature of a path $Y : [a,b] \mapsto \mathbb{R}^d$ is the collection of all iterated integrals of Y, denoted by $S(Y)_{a,b}$.

$$S(Y)_{a,b} = (1, S(Y)^1_{a,b}, ..., S(Y)^d_{a,b}, S(Y)^{1,1}_{a,b}, S(Y)^{1,2}_{a,b}, ..., S(Y)^{111}_{a,b}, ...), \tag{2.9}$$

where $I = (i_1, ..., i_k)$ is a multi-index with $i_k ... i_k \in \{1, ..., d\}$ and the first element of the collection is zero by convention.

2.4.3.1 Properties of order two

According to Chevyrev and Kormilitzin (2016), the terms of first level $S(Y)^i_{a,b}$ with $i = 1, ..., d$ are the increment, $Y^i_b - Y^i_a$, of the path $Y : [a,b] \mapsto \mathbb{R}^d$.
The second level for the Signature for the same index $S(Y)^{i,i}_{a,b}$ is $(Y^i_b - Y^i_a)^2/2$
To attribute meaning to the term $S(Y)^{i,j}_{a,b}$, $i \neq j$ the Lévy Area is computed.

Definition (Lévy Area): Considering Y^i_t and Y^j_t, the Lévy Area for each pair (i,j) with $i \neq j$ of two dimensional paths is

$$A_{i,j} = \frac{1}{2}\Big(\int_{a<t_1<t_2<t} dY^i_{t_1} dY^j_{t_2} - \int_{a<t_1<t_2<t} dY^j_{t_1} dY^i_{t_2}\Big) = \frac{1}{2}(S(Y)^{i,j} - S(Y)^{j,i}) \tag{2.10}$$

This gives a signed area confined by the path and a line connecting the starting and ending point of the path, Figure 2.2. illustrates.

The areas designated by A_- and A_+ are the negative and positive areas, and ΔX^1 and ΔX^2 are the increments along each coordinate.

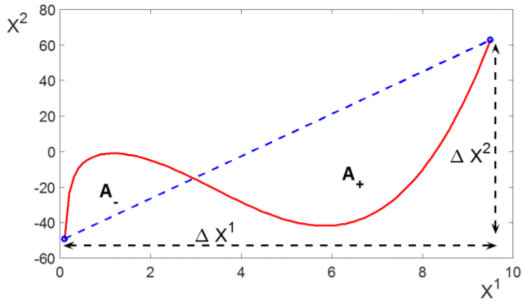


Figure 2.2: Example of a signed Lévy area of a curve. Areas above and under the chord connecting two endpoints are negative and positive respectively. Source: Chevyrev and Kormilitzin, *A Primer on the Signature Method in Machine Learning* (2016, p.11)

Chapter 3

Clustering

3.1 Classical clustering

3.1.1 Overview

The different clustering methods can be classified into four types: hierarchical clustering, partitional clustering, density-based clustering and grid-based clustering (Roelofsen, 2018).
The most common clustering methods for time series are hierarchical and partitional procedures. In this section we introduce the partitional clustering method used in our work, specifically the k-Means algorithm.

According to Liao (2005), the choice of the distance measure is more important than the choice of the clustering algorithm. Being distance measures the major focus of this thesis, we only make use of partitional method because of their simplicity and scalability.

In section 3.1.2 we look into partitional methods and discuss the general k-Means algorithm.

3.1.2 Partitional clustering

Partitional clustering is a strategy designed to create partitions. The number of k clusters is defined *a priori* and the data is assigned to one and only one of the k total clusters. The fact that the number of clusters has to be defined beforehand can be a restrictive factor. These algorithms can be seen as an optimization problem where intra-cluster distance is minimized while maximizing the inter-cluster distance.

3.1.2.1 k-Means

In this thesis, we use the most long-established partitional clustering technique, called k-Means. **Algorithm 1** presents the general definition of the method.

Algorithm 1 k-Means clustering algorithm

```
Select number of k clusters
Initialization of k cluster centers, $\mu_1, ..., \mu_k$;
while stopping is not satisfied do
    for $i = 1 : N$ do
        $c_i := \arg\min_l d(x_i, \mu_l)$
    end
    for $j = 1 : k$ do
        $\mu_j := \frac{\sum_{i=1}^{N} 1c_i=jx_i}{\sum_{i=1}^{N} c_i=j}$
    end
end
return $c_1, ..., c_N$ and $\mu_1, ..., \mu_k$
```

The algorithm starts by defining the k number of clusters. Once the number of clusters is chosen, the initialization of the k centroids starts. There are several methods for the initialization but the most common are the Forgy and Random Partitioning Techniques (Hamerly and Elkan (2002)). The first one chooses k observations from the data set and assumes those values as the initial centroids. On the other hand, the Random Partitioning method randomly allocates each observation to one of the k clusters and computes the mean of each cluster, assuming them as the initial centroids.

The next step in the algorithm is the "while" loop, that keeps running until the stopping criteria is satisfied. The most common stopping criteria are "convergence", "maximum number of iterations", "the variance did not improve by at least x and the variance did not improve by at least $x\times$ initial variance".

Two "for" loops can be found inside the "while" loop:

- The first one allocates to each observation x_i, a label c_i that specifies the cluster with the center closer in distance to the observation.

- The second "for" loop allows computing the new average (centroid) of each cluster after assigning the observations to the clusters using the first "for" loop.

  When the stopping condition is satisfied, the algorithm returns the clusters with all the observations $c_1, ..., c_N$ and the centroids $\mu_1, ..., \mu_k$.

Usually, the distance used in this algorithm is the Euclidean distance, but other distances can be applied. We make use of the elastic measure, DTW. However, there is evidence that this is not the best approach when working with time-series. The algorithm computes itself the distances, so it does not make use of the distance matrix and the triangle inequality does not hold (Niennattrakul and Ratanamahatana (2007)). For further research, we would consider a better algorithm to work with time-series data.

The most common k-Means algorithms are the Forgy/Lloyd algorithm (Lloyd (1982), Forgy

(1965)), the Hartigan-Wong algorithm (Hartigan and Wong (1979)) and the MacQueen algorithm (Macqueen (1967)). **Algorithm** 1 is the base of the Forgy/Lloyd algorithm.

The Forgy/Lloyd algorithm is appropriate to work with large data sets but because it recomputes the cluster centroids for every observation in every iteration it has slow convergence when compared to others.

The Forgy/Lloyd and the MacQueen are very similar, but the second only updates the centroids when an observation is assigned to a different cluster instead of doing it in every iteration.

3.2 Spectral clustering

3.2.1 Overview

The use of *spectral methods* for clustering has been considered a promising alternative to classical clustering approaches. There are many spectral clustering techniques and these have been applied to various fields. See some examples in U. von Luxburg (2006).

The first step of a spectral clustering method is the computation of the affinity matrix A that stores the similarities for any pairs of data, then a base reduction is performed using the k largest eigenvalues of an affinity matrix A and finally the main direction of the reduced model is pursued in order to assign one cluster among k to each object in the data set.

The main directions are specified as a basis of $\mathbb{R}^k$(or $\mathbb{C}^k$), and the cluster associated with each data is given by selecting the closest line among the k orthogonal lines specified by the basis (Lejay, 2019).

The distances between the pairs of data are stored as coefficients of the affinity matrix A. Usually, this matrix is symmetric, however, this not necessarily the case. Atev (2011) presents an algorithm capable of dealing with asymmetric matrices.

This thesis uses the **R** translation of Atev (2011) **Mathlab** code, to perform the algorithm. The following subsections present the principal mathematical foundations of the algorithm. Further details on the whole process and mathematical foundations can be found in Atev (2011).

3.2.2 Methodology

In order to perform spectral clustering, there are some relevant steps that should be followed. First step consists in building from n data $x_1, ..., x_n$ a matrix $n \times n$ A that stores the pairwise similarities between the objects to be clustered.

Afterward, the spectral decomposition step is performed in matrix A, using the spectral information contained there and transforming it into a matrix $Y_{k\times n},$, where k is the number of clusters.

Finally, we look for a unitary matrix U whose axis encode an orthonormal basis $\{u_1, ..., u_k\}$ on $\mathbb{C}^k$. The matrix minimizes the overall distance between the columns vectors of Y and the vectors emerging from 0 in one of the direction of u_i. If the j-th column vector of Y is closer to the line in the direction of u_i than any other direction $u_d \in \{u_1, ..., u_k\}$, then the data x_j is assigned to cluster i.

3.2.3 Dealing with asymmetric matrices

Affinity matrices are known for being symmetric, however, asymmetric affinities can appear in several applications. In cases like this, the usual process is to transform it into a symmetric matrix in order to perform the clustering. The majority of the common clustering methods require symmetric affinity matrix.

However, this process usually implies information loss. Since all information available is contained in the affinity matrix, it is important that we keep all its properties and relations with the data. For this reason, the search for solutions to work with asymmetric affinities has been the object of study from experts in the field.

Atev (2011) proceeds to the implementation of a solution that deals with asymmetric matrices. The idea is to build a Hermitian matrix $H(A)$ from the similarity matrix A $(n \times n)$, that is

$$H(A) = \frac{1}{2}(A + A^T) + i\frac{1}{2}(A - A_T) \tag{3.1}$$

The algorithm will use as input a matrix with the same information as the original similarity matrix A since the mapping $H(A)$ is a bijection:

$$A = \Re H(A) + \Im H(A) \tag{3.2}$$

The eigensystem to be solved is the following

$$H(A)\mathrm{x} = \lambda\mathrm{x}, \mathrm{x} \in \mathbb{C}^d, \lambda \in \mathbb{R} \tag{3.3}$$

and the solution is

$$V\Lambda V^H = H(A) \tag{3.4}$$

where V is a unitary matrix and

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_d \end{bmatrix} \tag{3.5}$$

is a diagonal matrix containing the eigenvalues, with $\lambda_1 \geq \ldots \lambda_d$.

3.2.4 Spectral Embedding

Once performed the decomposition of the eigenvalues, we select the k eigenvectors corresponding to the k largest eigenvalues of matrix A ($n \times n$) and transform it into a matrix Y ($k \times n$), where k is the desired number of clusters.

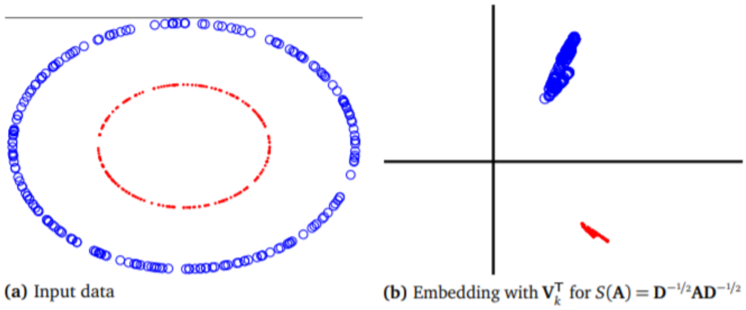
This embedding step is a dimensional reduction where a new set of coordinates is generated for the data based on the chosen eigenpairs.

Atev (2011) summarizes the most common embedding techniques, namely:

$\mathbf{Y}$
$\mathbf{V}_k^T diag(\mathbf{V}_k \mathbf{V}_k^T)$
$\mathbf{V}_k^T$
$\mathbf{V}_k^T \mathbf{D}_{out}^{-1/2}$
$\mathbf{L}_k^{-1/2} \mathbf{V}_k^T$

Table 3.1: Spectral embedding techniques. Source: Atev (2011)(table 3.2, p.23)

with $\mathbf{D}_{out} = diag(\mathbf{A1})$ where $\mathbf{1}$ is a vector of all ones of appropriate size and $\mathbf{L}_k$ is a diagonal matrix with the $\mathbf{k}$ largest eigenvalues.



(a) Input data

(b) Embedding with $\mathbf{V}_k^T$ for $S(\mathbf{A}) = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$

Figure 3.1: Illustration of the embedding. Source: Atev (2011). The coordinate axes in the spectral space are shown as thick lines. *Using Asymmetry in the Spectral Clustering of Trajectories* (2011, p.23)

To perform the spectral clustering, the spectral embedding used is the *Kernel PCA*, namely $\mathbf{Y} = \mathbf{L}_k^{-1/2}\mathbf{V}_k^T$. The scaling is defined to work with complex numbers.

$$\mathbf{Y} = \begin{bmatrix} \lambda_1^{-1/2} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_d^{-1/2} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\cdot\mathbf{1}} \dots \mathbf{V}_{\cdot\mathbf{k}} \end{bmatrix}^H \in \mathfrak{M}_{n \times k}(\mathbb{C}) \tag{3.6}$$

3.2.5 Cluster Assignment

The next step consists of assigning our objects to a cluster. When using the *Kernel PCA* transformation, $Y \in \mathbb{C}^{k \times n}$, which is a projection of the set of data into a higher dimensional feature

space. This projection ensures the linear separation, required to work with linear methods, such as PCA, of a non necessarily linear data set.
Ideally, all points of $y_1 \dots y_n$ are laying on a set of k mutually perpendicular lines through the origin.

According to S.E Atev (2011), when the embedding is produced by a perturbed ideal affinity matrix, we need to find a set of k mutually perpendicular lines such that the sum of squared distances from each point to the nearest line is minimized. The name of the algorithm is k-Axes, in analogy with the k-lines algorithm of I. Fischer and J. Poland (2005). Both algorithms share the same objective function but differ in the imposition of orthogonality constraints to the lines. The cost function is optimized under orthogonality and adds itself to optimization by the Riemannian Conjugate Gradient method proposed by Abrudan et al. (2009).

3.2.6 Algorithms' implementation

This section presents the numerical algorithms discussed above to perform spectral clustering using asymmetric matrices.
The process is characterized by the following numeric implementations: k-Axes algorithm and Conjugate Gradient with orthogonality constraints algorithm. There are some mid-steps and secondary implementations that will not be presented here, additional details are in pages 79-84 of Atev (2011).

3.2.6.1 k-Axes

According to Atev (2011), the clustering problem we are trying to solve can be written as a discrete minimization problem:

$$\begin{gathered} \text{minimize } J(Y,U,W) \\ \text{subject to } U^H U = I \\ w_{i,j} \in \{0,1\} \\ \textstyle\sum_{j=1}^{k} w_{i,j} = 1, \end{gathered}$$

where $J(Y,U,W)$ is the k-Axes cost function:

$$J(Y,U,W) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{i,j} \big(d^{\perp}(y_i, u_j)\big)^2 \tag{3.7}$$

The orthogonal distance from the point y to the line through the origin along the direction specified by the unit vector u is measured by $d^{\perp}(y_i, u_j)$. The matrix $W(n \times k)$ conceals the cluster association of the columns of Y, u_j is the j-th column of U. Since the matrix A is asymmetric, U is an Unitary matrix whose axes encoding an orthonormal basis $u_1, \dots, u_k$ on $\mathbb{R}^k$ or $\mathbb{C}^k$. This matrix is selected to minimize the overall distance between the column vectors of Y, seen as elements of

$\mathbb{R}^k$, and the rays emanating from 0 in one of the direction u_i. If the j-th column of Y is closest to the line in the direction u_i than in any other direction u_l, then the data x_j is assigned to the i-th cluster (Lejay, (2019)).

The distance $d^{\perp}(y, u)$ can be expressed as the following:

$$d^{\perp}(y, u) = \sqrt{y^H(I - uu^H)y} \tag{3.8}$$

The constrain $\sum_{j=1}^{k} w_{ij} = 1$ is satisfied by minimizing the value of J, while varying between modification steps of U and W. J is minimized under the orthogonality constrains on U, keeping W fixed. Then W is modified as follows, while keeping U fixed:

$$w_{ij} = \frac{\exp(-d^{\perp}(y_i, uj)^2/\sigma^2)}{\sum_{l=1}^{k} \exp(-d^{\perp}(y_i, u_l)^2/\sigma^2)} \tag{3.9}$$

where σ, which determines the "softness" of the softmax function, is quickly decreased after two iterations, allowing W to quickly approach a binarized cluster membership matrix (Atev, (2011)).

Algorithm 2 k-Axes spectral clustering algorithm

```
Initialization
Let $U_0 = I$ and $\sigma_0 = max_{i=1}^{n} min_{j=1}^{k} |y_{ij}|$.
Compute $W_0$ using $W_{i,j} = \frac{\exp(-d(Y_{\cdot i}, U_{\cdot j})^2)/\sigma^2)}{\sum_{l=1}^{k}(\exp(-d(Y_{\cdot i}, U_{\cdot l})^2/\sigma^2))}$. Let $J_0 = J(Y, U_0, W_0)$
while $(J_{t-1} - J_t \leq \epsilon)$ do
    for $t = 1, 2, ...$ do
        $\sigma = (J_{t-1}/n)^{t/2}$
        $U_t = argmin_{U^H U = I} J(Y, U, W_{t-1})$
    end
end
return Assign to each point to the cluster with the largest weight in $\mathbf{W_t}$
```

3.2.6.2 Conjugate Gradient with orthogonality constraints

In order to k-Axes (**Algorithm 2**) to work, we need show how to minimize $U_t = arg \min_{U^H U = I} J(Y, U, W_{t-1})$. Atev (2011) uses Abrudan et al. (2009) to perform the minimization, that requires the specification of a gradient of $J(Y, U, W)$ with respect to U.

According to Atev (2011) the procedure is:

$$
\begin{aligned}
\frac{\partial J}{\partial U^*} &= \frac{\partial}{\partial U^*}\sum_{i=1}^{n}\sum_{j=1}^{k} w_{ij} d^{\perp}(y_i, u_j)^2 \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial U^*} \sum_{j=1}^{k} w_{ij} y_i^H (I - u_j u_j^H) y_i \\
&= \sum_{i=1}^{n} \frac{\partial}{\partial U^*} \Big(y_i^H y_i - \sum_{j=1}^{k} w_{ij} y_i^H u_j u_j^H y_i \Big) \\
&= -\sum_{i=1}^{n} \frac{\partial}{\partial U^*} \Big(\sum_{j=1}^{k} w_{ij} y_i^H u_j u_j^H y_i \Big)
\end{aligned} \tag{3.10}
$$

The previous equation is evaluated term-by-term for a given row r and column c, at a fixed data point i:

$$
\begin{aligned}
\Big[\frac{\partial J}{\partial U^*}\Big]_{rc} &= -\frac{\partial}{\partial u_{rc}^*} \sum_{j=1}^{k} w_j d^{\perp}(y^H u_j u_j^H y) \\
&= -\sum_{j=1}^{k} w_j \frac{\partial}{\partial u_{rc}^*} \big((y^H u_j)(u_j^H y) \big) \\
&= -\sum_{j=1}^{k} w_j \frac{\partial}{\partial u_{rc}^*} \Big(\sum_{p=1}^{k}\sum_{q=1}^{k} y_p^* u_{pj} y_q u_{qj}^* \Big) \\
&= -w_c \frac{\partial}{\partial u_{rc}^*} \Big(\sum_{p=1}^{k} y_p^* u_{pc} y_r u_{rc}^* \Big) \\
&= -w_c \sum_{p=1}^{k} y_p^* u_{pc} y_r \\
&= -w_c y_r y^H u_c
\end{aligned} \tag{3.11}
$$

Algorithm 3 outlines the minimization process under orthogonality constraints:

Algorithm 3 Conjugate gradient under orthogonality constrains

```
Initialization
Let $U_0$ be an initial guess with $U_0^H U_0 = I$.
Compute $\Gamma_0 = \frac{\partial}{\partial U^*} J(U_0)$ and let $G_0 = \Gamma_0 U_0^H - U_0 \Gamma_0^H$. Set, $H_0 = G_0$
for $k = 0, 1, \dots$ do
    if tr$(G_k^H G_k)$ is small, return $U_k$ and stop.
    Let $w_{max}$ be the eigenvalue of $H_k$ of the largest magnitude. Set $T_{max} = \frac{2\pi}{2|w_{max}|}$
    Perform the line search:
        $\mu = \arg_{\mu \in [0, T_{max})} \min J(Y, \exp(-\mu H_k) U_k, W) \quad (3.12)$
    using the polynomial approximation for th first zero crossing of $\frac{\partial J}{\partial U^*}$ along $-H_k$. Perform the updates:
        $U_{k+1} = \exp -\mu H_k) U_k$
        $\Gamma_{k+1} = \frac{\partial}{\partial U^*} J(U_{k+1})$
        $G_{k+1} = \Gamma_{k+1} U_{k+1}^H - U_{k+1} \Gamma_{k+1}^H$
        $H_{k+1} = G_{k+1} + \frac{\Re tr\big((G_{k+1} - G_k)^H G_k}{tr(G_k^H G_k)} H_k$
    if $\Re tr(H_{k+1}^H G_{k+1}) \leq 0$, reset $H_{k+1} = G_{k+1}$
end
```

Chapter 4

Practical Research

In the previous chapters, we described the methods used in this work. Starting with similarity measures and moving to clustering algorithms. Our purpose is to compare two different approaches to have a better sense of the applicability of the Signature method, considering the traditional method as our benchmark.

Unlike classification, the evaluation of clustering algorithms depends much on the goal of the analysis and can change depending on the criteria.

We chose to present the two partitional methods so we could use as a form of comparison methods of the same group of clustering algorithms. However, in the testing phase of our work other methods were applied to the data set. The description of those methods will not be presented in this thesis due text size restrictions.

We start by describing the data used and how we processed it to serve as input to our methods. Then we will proceed with an illustration of the algorithms described in the previous sections.

The algorithm's illustration starts with a comparison between distance measures through computation times. Finally, we compare cluster quality also by comparing computation times, performing forecasting, and analyzing cluster composition, based on the other quantitative variables and assets type. As a preliminary approach, the application of k-Means with Euclidean distance is made and for that reason, computation times appear on Tables 4.1 and 4.2. However, our analysis is focused on the use of the DTW and Signature method to compute distances and for that reason, we will not present cluster results using any other distance (besides the computation times).

We chose to select an arbitrary variable from each cluster to present the forecast. The forecast is done using the cluster's means and then comparing it to the actual NAV prices of the variable.

The following sections describe the process of the practical research and the results obtained. In section 4.1 we present the time series data set used and the preprocessing of the same data. Section 4.2 is dedicated to the distance measures and the computation times of each distance. Finally, section 4.3 analyse cluster quality in terms of time efficiency, cluster composition, and an example of forecasting. In subsection 4.3.4 we show an example of k-Axes clustering using the Signature method in the whole sample 2005-2019 in order to highlight some interesting observations.

4.1 Data

In this section, we describe the data set used in practical research. In subsection 4.1.1 we purely describe our time-series data and its constituents. In the next subsection, 4.1.2, we discuss data preprocessing and what subset will be used to run the experiment.

4.1.1 Data description

The data used in practical research are time-series of Mutual and Trade Exchange funds of Luxembourg with weekdays observations of Net Asset Value (NAV) from the 4th of January 2005 to the 30th of January 2019. Each column of the data set corresponds to a mutual fund and the rows contain the NAV prices of each observation, day after day.

Besides the daily prices, used as input for the clustering, there are other quantitative and qualitative variables, such as total Expense Ratio, which reflects how much of a fund is used to cover the expense, total net assets (TNA), and asset type. Once the clusters are computed, we intent to compare the distinct methods of this work, specifically k-Means using DTW and k-Axes using the Signature transformation. Besides that, we'll analyze the composition of each cluster in terms of the descriptive variables, to understand if the financial properties of data were caught or not. The idea is to understand if funds with similar performance, belong to the same group.

4.1.2 Data preprocessing

Before clustering the data, we need to transform it. In our case, we use the normalization process, since the time-series have different ranges of NAV prices and we want to capture similar shapes between them when performing clustering. With normalization, all time-series are in the range of [0,1] and the process is done in the following way:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

where X is a time-series with observations from $1 \dots n$. X_{min} is the value of the observation corresponding to the minimum value of X and X_{max} the observation corresponding to the maximum value of X.

When performing initial tests with the whole data set, we concluded that using the full length of all time series could bring too much time complexity. Especially in the case of DTW, computing the distance for the whole data set seemed to be seriously inefficient and computations could take much more time than expected. For this reason, the sample is divided into four distinct partitions, the corresponding periods are from 2005-2006, 2007-2009, 2010-2011, and 2012-2019. The analysis is made for all partitions, however, we'll focus on presenting results for the most recent one.

The partition of 2012-2019 (partition 4) is divided into train and test set, the train set includes observations of the 539 funds from January 2nd of 2012 to December 30th of 2016, and the test set from January 2nd of 2017 to January 31st of 2019. We use the train set to compute the clusters and forecast, the test set is used to understand cluster quality through an example of a forecast using cluster means and comparing it with the actual time-series NAV prices.

4.2 Distance measures

This section resumes the distances measures used and the correspondent **R** functions and implementations.

We calculate the computation time for two sets, the first set corresponds to the train set with all 539 mutual funds (N=539) while the subset sample includes a random selection of 50 mutual funds (N=50) from the train set. The computations were performed several times and did not show significant variance. Since DTW is time inefficient, when compared to lock-step measures, for example, we decided to use a subset with fewer variables to compare the computation times for all distances.

- **Euclidean:** Implementation of the function *dist* from the **R** package **stats**.
- **Dyamic Time Warping (DTW):** Implementation of the function *dist* from the **R** package **proxy**.
- **Signature (2.10):** Computation of Affinity matrix using the Lévy Area.

Distance measure	Whole sample (N=539)	Subset sample (N=50)
Euclidean	3.30	1.02
DTW	$\geq$ 1200*	8.65
Signature	12.36	1.09

Table 4.1: Computation times (seconds) of each distance to compute the distance matrix. * $\geq$ 1200: Forced computation to stop after 20min. Source: Author's calculations

Table 4.1 presents the computation time of each distance, results are presented in seconds. When looking at the computation times of each distance, becomes clear the difficulties encountered by DTW when applying it to a data set with a significant number of variables and/or observations. For instance, when applied to the whole sample, the computation time exceeded 20 minutes, which was our stopping criteria.

We know time efficiency is a very important part of computation methods alongside accuracy. The Signature method outperformed Dynamic Time Warping in computation time.

The fastest distance to be computed was the Euclidean distance, as we anticipated. However, this distance has some limitations, mentioned before, due to hypersensitivity to small distortions in the time axis.

4.3 Selecting the number of clusters

This section introduces a heuristic used for selecting the optimal number of clusters. The step is required because the work of our thesis uses partitional clustering algorithms, they all require *a priori* selection. We need to give the number of clusters to the algorithm in order for it to compute the centroids and cluster membership of our objects. The heuristic consists of computing the total within the sum of squares (known as the elbow method).

4.3.1 Elbow method

The elbow method consists of plotting the explained variance (within the sum of squares) and picking the elbow curve as the number of clusters to use. The explained variance is computed in the function of the number of clusters.

The "elbow" of the curve is the cutoff point where adding an additional cluster does not improve the model significantly. Meaning that having more clusters than the ones that actually explain the majority of the information diminishes the information contained in each cluster since subdivisions of those sets will appear.

Within sum of squares can be defined in the following way:

$$\text{minimize } \sum_{i=1}^{k} W(C_i)$$

where C_i is the i-th cluster and $W(C_i)$ is the within cluster variation.

The number of clusters selected for the experiment is $k = 4$. The plot of the elbow method can be found in Appendix B, see Figure B.1. This method was applied to the **tsclust** function and we considered the same number in k-Axes when comparing both methods.

4.4 Comparing cluster quality

This section is dedicated to the comparison of clustering approaches. In subsection 4.3.1 we compare computation times of each method. In subsection 4.3.2 we present the composition of the clusters. Subsection 4.3.3 gives an example of forecasting results and finally, we discuss results obtained using the entire sample (2005-2019).

4.4.1 Cluster Computation Times

Computation times were calculated multiple times and once again did not show significant variance. The results represent an arbitrary result of our round of testing.

The functions used to compute the clusters are the following:

- **k-Means (stats):** Implementation of the function *kmeans* from the **R** package **stats**.
- **k-Means (dtwclust):** Implementation of the function *tsclust* from the **R** package **dtwclust** with type='partitional' and distance ='dtw_basic'.
- **k-Axes (Signature)**: Implementation of the function *k-Axes* from the **R** package **AsymmentricClustering**. Function originally developed by S.E Atev (2011) in **Mathlab** and converted to **R** by Lejay and myself. k-Axes uses affinity matrix A that stores the Lévy Area computations from the Signature.

Algorithm	Distance	Whole Sample (N=539)
k-Means	Euclidean	3.60
k-Means	DTW	55.69
k-Axes	Signature	7.39

Table 4.2: Computation times (seconds) of each cluster algorithm with respective distance measure. Source: Author's calculations

Table 4.2 shows the computation times of each clustering method. The asymmetric clustering method using Signature transformation outperformed the k-Means algorithm using DTW. Again, our implementation outperformed our benchmark. k-Means using Euclidean distance was the fastest method to be computed, as expected.

4.4.2 Clusters composition

In this subsection, we describe the composition of the clusters obtained. We present the means of each cluster concerning Share Class TNA and Expense Ratio and then we show the distribution of asset type in each of them.

4.4.2.1 k-Means with DTW

Cluster	Size	Share Class TNA (K EUR)	Expense Ratio (%)
1	46	76.933	2.103
2	182	222.016	1.535
3	222	219.307	1.859
4	89	221.590	1.136

Table 4.3: Means of quantitative variables of each cluster using k-Means.

By analyzing Tables 4.3 and 4.4 it seems to exist a relationship between clusters with higher values of Expense Ratio and quantity of equity funds, which makes sense financially since stocks have higher prices than bonds or money market. Cluster 4, which is the cluster with the lowest

Cluster	Alternatives (%)	Bonds(%)	Equity(%)	Mixed Assets(%)	Money Market(%)
1	2.2	8.7	50	39.1	0
2	1.7	40.7	24.7	32.4	0.5
3	0.4	8.6	70.7	20.3	0
4	4.5	75.3	2.2	7.9	10.1

Table 4.4: Cluster breakdown by asset type using k-Means. Source: Author's calculations

Expense ratio and one with the highest mean of Share Class TNA (k EUR) is mainly composed of bonds and aggregates most of the Money Market funds. The relationship between a low Expense Ratio and a higher quantity of mutual bond funds is also evident.

4.4.2.2 k-Axes with Signature

Cluster	Size	Share Class TNA (K EUR)	Expense Ratio (%)
1	7	524.016	1.277
2	39	218.159	1.809
3	246	179.852	1.636
4	244	228.288	1.657

Table 4.5: Clusters' means of descriptive data using k-Axes. Source: Author's calculations

Cluster	Alternatives(%)	Bonds(%)	Equity(%)	Mixed Assets(%)	Money Market(%)
1	0	57.1	14.3	28.6	0
2	0	15.4	41	41	2.6
3	1.22	32.11	43.5	21.14	2.03
4	2.46	30.74	41.8	23.4	1.6

Table 4.6: Clusters' breakdown by asset type using k-Axes. Source: Author's calculations

Tables 4.5 and 4.6 show the results obtained with the k-Axes using the Signature transformation and the same number of clusters.

Becomes evident that the properties captured by the Signature do not seem to aggregate funds with similar descriptive variables. Clusters 1 and 2 are the smallest ones and the first one, with the lowest Expense ratio is mainly composed of bonds, the second one is the one with the highest Expense ratio and mainly composed of equity and mixed assets.

Clusters 3 and 4 are quite similar, there is no significant difference between these two funds in terms of descriptive data.

4.4.3 Forecasting

In this subsection, we use an example of forecasting to compare cluster quality between the two methods previously described. By comparing the errors of both approaches we can have a better sense of what algorithm works better to fit our data.

We use the function *auto.arima* of the **R** package **forecast** to perform the forecast and then compare it to the actual normalized prices. Since forecasting is not part of our work, we will not enter into much detail about the subject, we will simply compare the results obtained.

For each cluster, we use the train set corresponding to the sample from 2012 to 2016 to forecast the last two years (2017-2019). Since we use the normalized prices, there is a possibility that the train set uses information from the last two years. However, we stick to the use of the normalized prices of the train set because it diminishes the effects caused by outliers.

The section presents an example of forecasting using cluster means to fit the model and then comparing it to the actual prices of an arbitrary time-series using the two approaches. The cluster mean used to predict the fund prices is the representative mean of the cluster where the fund belongs, in both cases. The accuracy results show the Root Mean Square Error (RMSE) that measures the standard deviation of the residuals and the Mean Absolute Error (MAE) that measures the error in absolute value, they both express the average error of the predictive model.

Algorithm	Distance	Set	RMSE	MAE
k-Means	DTW	Training	0.0050	0.0033
		Test	0.0522	0.0408
k-Axes	Signature	Training	0.0188	0.01259
		Test	0.18684	0.14527

Table 4.7: Accuracy function results from **R** package *forecast* using the prices of an arbitrary variable. Source: Author's calculations

Table 4.7 shows the accuracy results obtained for the two distinct methods. In this particular example, k-Means using DTW outperformed k-Axes with the Signature method in all measurements, as we expected. However, one should note that the results obtained with the Signature are quite reasonable for an initial approach and shows again the huge potential of the method to capture important properties of data, even when only properties of order two (Lévy Area) are considered. The predicted period was quite extensive which might have an influence on the accuracy results of both methods. The plots of the example can be seen in Appendix E.

4.4.4 k-Axes on entire data set (2005-2019)

In this subsection we present a plot obtained when clustering the entire sample.

The figure 4.1 shows clustering results when $k = 3$ and the entire sample (2005-2012) is considered. Very important to note how clusters 2 and 3 have steep decreasing and then increasing movements around the 1000-th entry, which should correspond to the crisis period of 2008. This

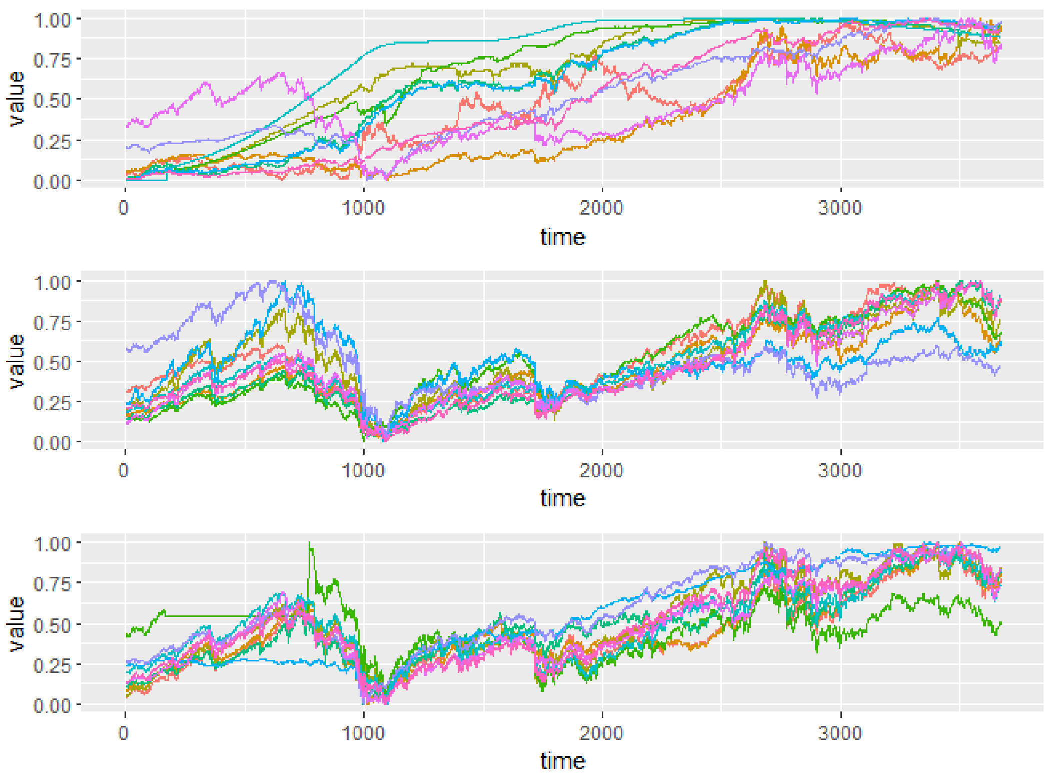


Figure 4.1: Example plot of **AsymmetricClustering** on full sample (2005-2019), when selecting 3 clusters, using Signature method. Source: Author's calculations

is a very interesting standpoint on how the Signature seems to capture prices fluctuations according to economic cycles, becomes evident how the Signature transformation seems to be a very promising alternative to measure similarities between two objects.

In cluster 1, some of the representative values of funds do not seem to capture this steep decrease around the 1000-th entry, its composition is quite even in terms of equity and bonds, no evident reason was detected to suggesting this fact.

Chapter 5

Discussion

Time-series clustering is a subject gaining more and more importance in the data science and mathematical community due to its role in discovering hidden patterns in data and uses them to make predictions. This has special importance when considering data streaming, such as stock prices.

To find patterns within the data it is important to choose an appropriate distance measure, which is the most important step of cluster analysis. The literature enumerates how lock-step measures are outperformed by elastic measures, such as Dynamic Time Warping, and, for this reason, we chose DTW, as our benchmark. However, these measurements have a huge downside when it comes to time efficiency, they are quite time consuming and this became evident when computing the similarity matrices using DTW.

The Signature method brings a whole new world of possibilities since it captures many important analytic and geometric properties of data by extracting characteristic features, without parametrization. For these reasons, the method is a good alternative candidate to measure similarities between two objects.

This work suggests a comparison between two clustering algorithms using two distinct ways of measuring similarities. Considering k-Means with DTW our benchmark, we intend to make an evaluation of the Signature method using the k-Axes algorithm.

The results of our practical research suggest that performing Asymmetric Clustering using the Signature method was a good starting point. The ability to capture the shocks of certain economic cycles when using only properties of order 2 to perform clustering tell us how powerful this method can be if we extend the incorporation of other properties and transformations of the Signature, such as the lead-lag transformation and its relationship with the variance of data.

Further research could invest in a more deep application of those properties and transformations and use them to compute the similarities between time series objects. Having an affinity matrix that condenses as much information as possible, according to the goal of the analysis, is fundamental to obtain good and significant results.

Also, we considered only partitional methods to compute clusters, we would suggest a more diverse application, such as hierarchical methods, and compare the results obtained.

Other important points one should look further are the indices for determining the optimal number of clusters. Instead of using a heuristic approach, a series of methods can be computed to provide a more accurate result, for instance, Gap Statistic or Average Silhouette.

Regarding forecasting, one suggests using the cluster means to predict but instead of using an arbitrary variable of the same group to compare results, one should compute the actual mean of the cluster to have a better sense of cluster quality.

Finally, we conclude that the choice of the algorithm and the distance measure can depend greatly on the goal of the analysis. The Signature method seems to be a promising alternative when dealing with time series and considering the difficulties encountered by elastic measures. If the end goal is to construct a model to be applied in data streams we need it to be time-efficient. However, if we intend to cluster static data then a simple k-Means might be the right and sufficient choice.

Bibliography

[1] Abanda, A., Mori, U., Lozano J.A. (2018) *A review on distance based time series classification.* University of Oxford DOI: arXiv:1806.04509.

[2] Abrudan, T., Eriksson, J., and Koivunen, V. (2009) *Conjugate gradient algorithm for optimization under unitary matrix constraint.* Signal Processing, 89(9):1704-1714.

[3] Atev, S.E (2011) *Using Asymmetry in the Spectral Clustering of Trajectories.* Dissertation submitted to the faculty of the graduate school of the University of Minnesota, pp. 12-21; 57-63; 74-95.

[4] Berndt DJ., Clifford, J. (1994) *Using Dynamic Time Warping to Find Patterns in Time Series.* In KDD Workshop volume 10, pp. 359-370, Seattle, WA.

[5] Chen, K. (1957) *Integration of paths, geometric invariants and a generalized baker-hausdorff formula.* Annals of Mathematics, pages 163–178.

[6] Chevyrev, I. and Kormilitzin, A. (2016) *A Primer on the Signature Method in Machine Learning.* DOI:arXiv:1603.03788 .

[7] Fisher, I. and Poland, J. (2005) *Amplifying the block matrix structure for spectral clustering.* Technical Report 03–05, IDSIA, Manno-Lugano, Switzerland.

[8] Forgy, E. (1965) *Cluster analysis of multivariate data: Efficiency versus interpretability of classification.* Biometrics 21(3), 768–769.

[9] Giorgino, T. *Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package.* Journal of Statistical Software, 31(7), 1–24. DOI: 10.18637/jss.v031.i07, 2009

[10] Gyurkó, L.G., Lyons, T., Kontkowski, M., Field, J. (2014) *Extracting information from the signature of a financial data stream.* University of Oxford DOI: arXiv:1307.7244v2.

[11] Hambly, B.M and Lyons, T. (2010) *Uniqueness for the signature of a path of bounded variation and the reduced path group.* Annals of Mathematics, 171(1):109–167.

[12] Han, J., Kamber, M., Pei, J. (2012) *Data Mining Concepts and Techniques.* The Morgan Kaufmann Series in Data Management Systems. Elsevier Inc. 1-23; 443-454.

[13] Hambly, B.M and Lyons, T.(2010) *Uniqueness for the signature of a path of bounded variation and the reduced path group.* Annals of Mathematics, 171(1):109–167.

[14] Hamerly, G. and C. Elkan. (2002) *Alternatives to the k-Means algorithm that find better clusterings.*In Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02, New York, pp. 600–607.

[15] Hartigan, J. A. and Wong, M. A. (1979) *A K-Means clustering algorithm.* Applied Statistics 28, 100–108.

[16] Jacques, J. and Preda, C. (2014) *Functional data clustering: a survey.*Advances in Data Analysis and Classification, Springer Verlag, 8 (3), pp.24. ff10.1007/s11634-013-0158-yff. ffhal-00771030f.

[17] Keogh, E. and S. Kasetty (2003) *On the need for time series data mining benchmarks: A survey and empirical demonstration.* Data Mining and Knowledge Discovery 7(4), 349–371.

[18] Lejay, A. (2019) *Asymmetric Spectral clustering.*

[19] Levin, D., Lyons, T. and Ni, H. (2016) *Learning from the past, predicting the statistics for the future, learning an evolving system.* University of Oxford.

[20] Liao, T.W. (2005) *Clustering of time series data - a survey.*Pattern Recognition 38(11), 1857 – 1874.

[21] Lin, J., Keogh, E., Lonardi, S. and Patel, P. (2002) *Finding Motifs in Time Series.* University of California - Riverside.

[22] Lloyd, S.P. (1982) *Least squares quantization in pcm.* IEEE Transactions on Information Theory 28, 129–137.

[23] Luxburg, U. (2006) *A tutorial on spectral clustering.* Technical Report 149, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

[24] Macqueen, J. (1967) *Some methods for classification and analysis of multivariate observations.* In In 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

[25] Maharaj, E. A., D'Urso, P. and Caiado, J. (2019) *Time Series Clustering and Classification.* Chapman & Hall/CRC, Computer Science & Data Analysis Series, pp. 21–22.

[26] Niennattrakul, V. and Ratanamahatana, C. A. (2007) *On clustering multimedia time series data using k-Means and dynamic time warping.* In Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering, MUE '07, Washington, DC, USA, pp. 733–738. IEEE Computer Society.

[27] Sardá-Espinosa (2019) *Comparing Time-Series Clustering Algorithms in R Using the dtwclust Package.* The R Journal. URL https://doi.org/10.32614/RJ-2019-023.

[28] Slonim, N., Aharoni, E. and Crammer K. (2013) *Hartigan's k-Means versus lloyd's k-Means: Is it time for a change?*. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13, pp. 1677–1684. AAAI Press.

Appendix A

Time-series decomposition

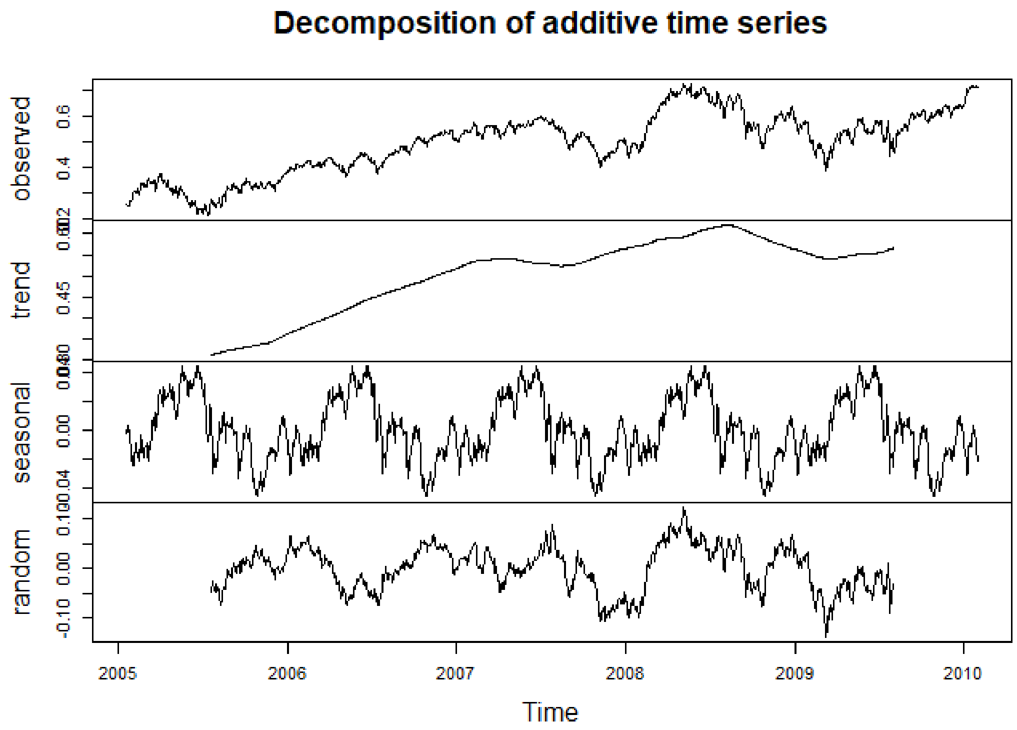


Figure A.1: Example of an arbitrary time series decomposition of the partition 2012-2016. Source: Author's calculations

Appendix B

Elbow method

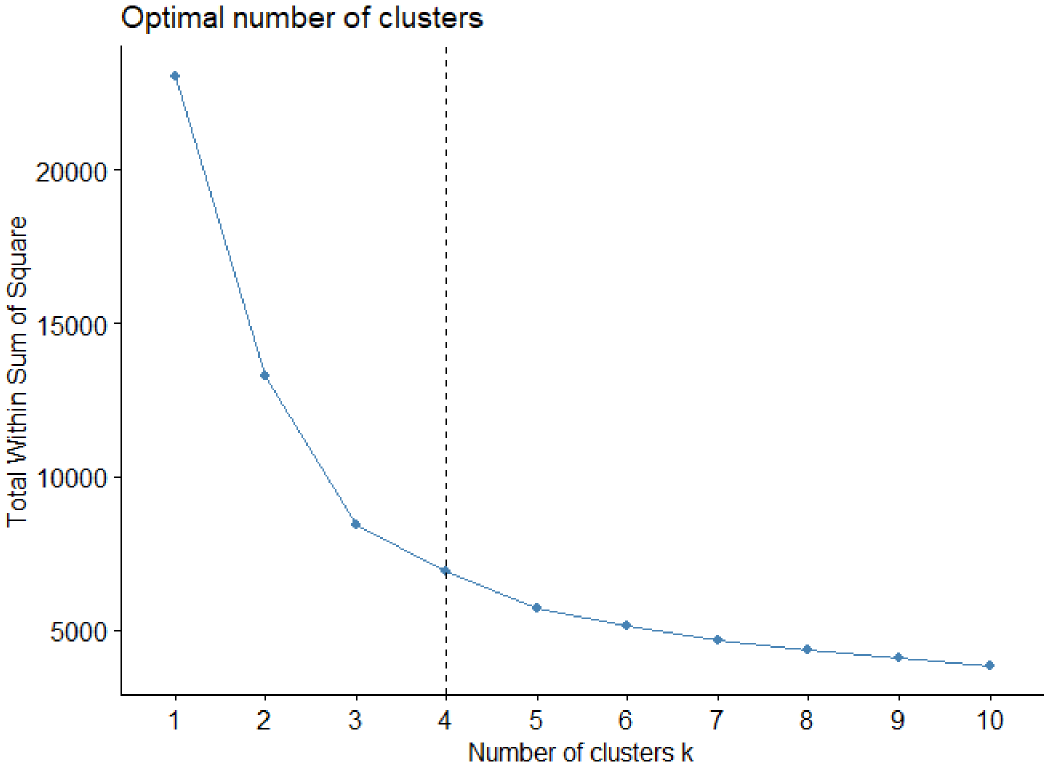


Figure B.1: Elbow method: plot of total within-cluster sum of square (WSS) for k-Means using DTW. Source: Author's calculations

Appendix C

k-Means with DTW

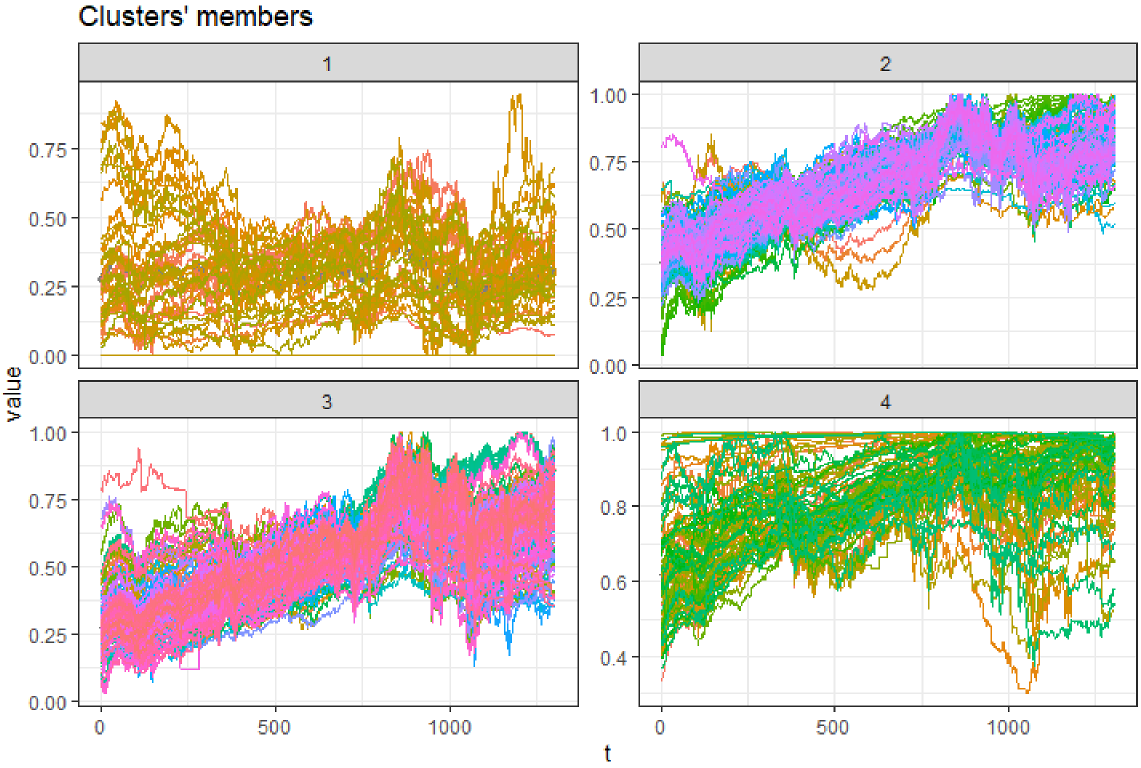


Figure C.1: Plot of the clusters using **tsclust** on sample set 2012-2016, with 4 clusters, using DTW. Source: Author's calculations

Appendix D

k-Axes with Signature

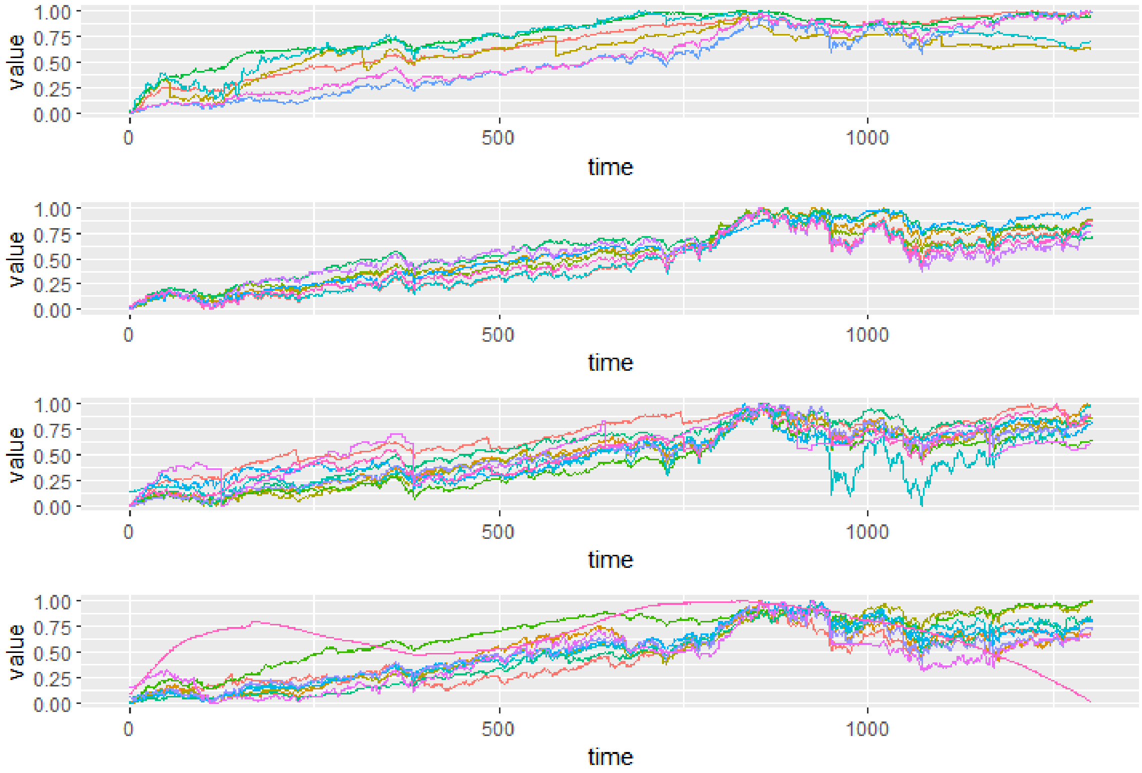


Figure D.1: Example plot of **AsymmetricClustering** on sample set 2012-2016 with 4 clusters, using Signature method and selecting a random sample to plot. Source: Author's calculations

Appendix E

Forecasting example

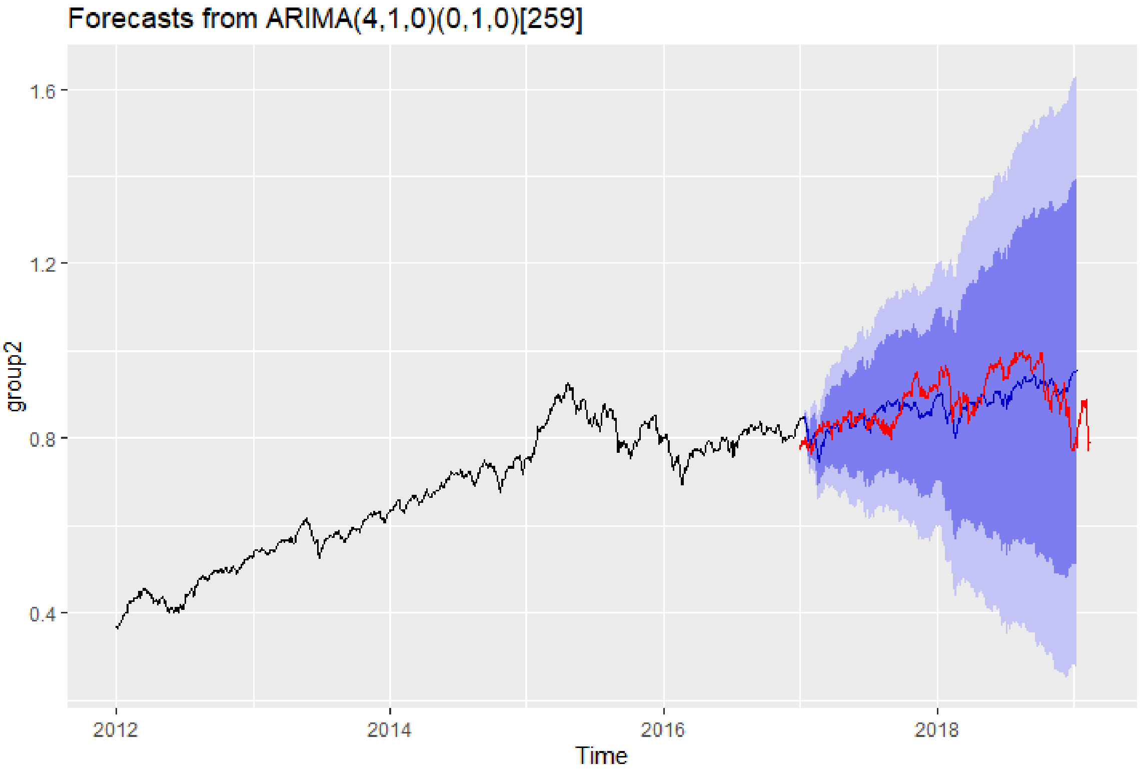


Figure E.1: Using ARIMA to forecast aprox. 2 years. The cluster means are used to fit the model (2012-2016) and normalized data of the fund LP60037109 is used as test set. The means were obtained using k-Means with dtw distance. Source: Author's calculations

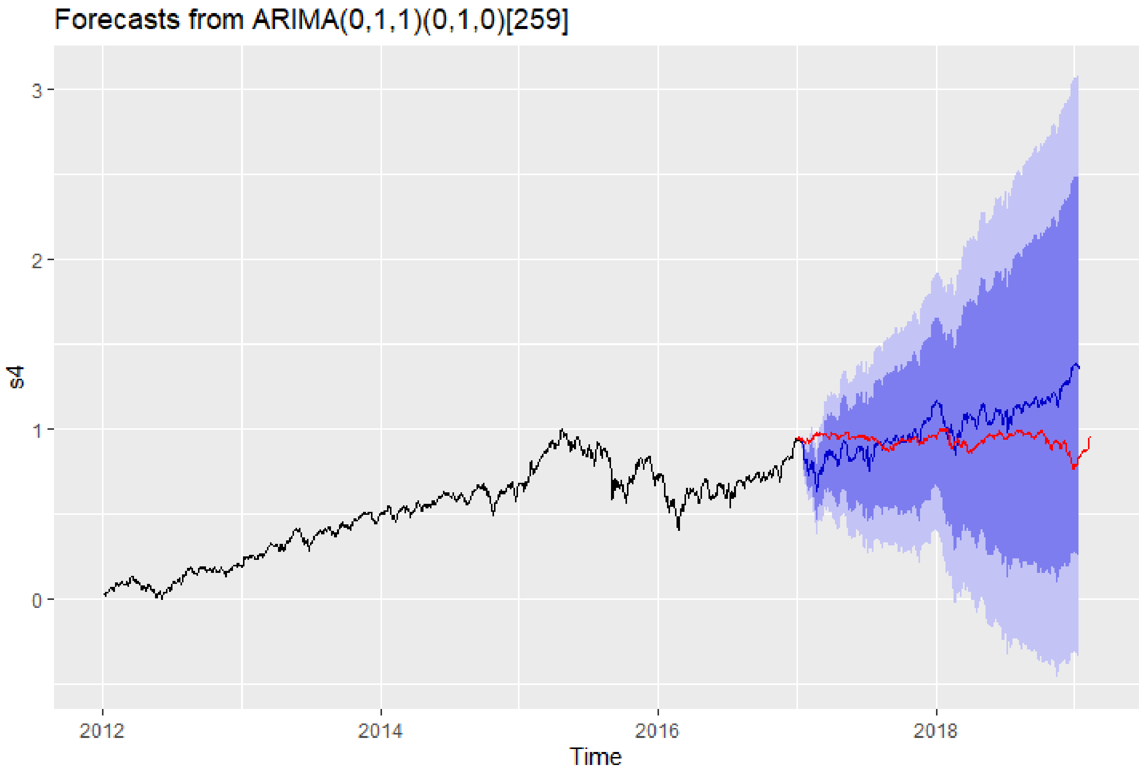


Figure E.2: Using ARIMA to forecast aprox. 2 years. The value of LP60037109 obtained trough k-Axes using Signature from 2012-2016 is used to fit the model. The next 2 years normalized prices of the fund are used as test set. Source: Author's calculations