Ludwig-Maximilians-Universität München
Department of Statistics

Master's Thesis

Similarity Approaches for high-dimensional Financial Time Series - with an Application to Pairs Trading

---

Author:	Karem El-Oraby
Faculty supervisor:	Prof. Stefan Mittnik, Ph.D
	Chair of Financial Econometrics
External supervisor:	Dr. Andreas Fuest
	HQ Asset Management GmbH, Düsseldorf
Submission date:	March 18, 2019

Declaration of Authorship

I hereby declare that this Master's thesis titled "Similarity Approaches for high-dimensional Financial Time Series - with an Application to Pairs Trading" is my own unaided work. All direct or indirect sources used are acknowledged as references.

The thesis was not previously presented to another examination board and has not been published.

Munich, March 18, 2019 Karem El-Oraby

Abstract

This thesis introduces a range of statistical techniques in order to determine when financial time series can be considered similar. Within the framework of a statistical arbitrage trading strategy, we combine time series clustering and cointegration to identify tradable pairs of assets. This procedure mitigates the multiple testing problem which occurs when many pairs of assets are simultaneously tested for cointegration. Time series can be clustered by means of raw data-based, feature-based and model-based approaches. Moreover, most traditional clustering methods can be applied in the temporal context without any further ado. The constructed pairs trading strategy is backtested over the period from 2000 until 2018 on the MSCI Europe Index constituents. The simplest strategy, employing the Euclidean distance in combination with K-medoids clustering, yields the best overall performance with an average excess return of more than 7% per annum before transaction costs. Similar to previous studies, we find a time-varying profitability of pairs trading which, however, mostly declined after the year 2005.

Contents

List of Figures

List of Tables

List of Abbreviations

ADF	Augmented Dickey–Fuller
AGNES	Agglomerative nesting
ANOVA	Analysis of variance
ARMA	Autoregressive moving average
BCSS	Between-cluster sum of squares
CH	Calinski-Harabasz
CLARA	Clustering LARge Applications
CRSP	Center for Research in Security Prices
DI	Dunn index
DIANA	Divisive analysis
DTW	Dynamic time warping
ECM	Error correction model
GARCH	Generalized autoregressive conditional heteroscedasticity
LCM	Local cost matrix
LOCF	Last observation carried forward
MC	Monte-Carlo
MSCI	Morgan Stanley Capital International
OLS	Ordinary least squares
PAM	Partitioning Around Medoids
PT	Pairs trading
QAF	Quantile autocovariance function
S&P 500	Standard & Poor's 500 Index
SI	Silhouette index
SSD	Sum of squared deviations
SSE	Sum of squared errors
VAR	Vector autoregressive model
WCSS	Within-cluster sum of squares

Notation

$\rho_X(k)$	Autocorrelation function of X at lag k
$\gamma_X(k)$	Autocovariance function of X at lag k
n_k	Cardinality of cluster k
Δ	Change in value
$\rho_{X,Y}$	Correlation coefficient of X and Y
$\text{cov}(X, Y)$	Covariance between X and Y
$\rho_{X,Y}(k)$	Cross-correlation between X and Y at lag k
r_t	Discrete return of an asset at time t
$\|\boldsymbol{z}\|_2$	Euclidean norm of vector $\boldsymbol{z}$
$\exp(1) = e^1$	Euler's number
$\mathbb{E}[X]$	Expected value of X
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$i.i.d.$	Independent and identically distributed
$I(\cdot)$	Indicator function
$\boldsymbol{M}^{-1}$	Inverse of matrix $\boldsymbol{M}$
log	Natural logarithm to the base of e
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{P}(A)$	Probability that event A occurs
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}_{>0}$	Set of positive real numbers
$tr(\boldsymbol{M})$	Trace of matrix $\boldsymbol{M}$
$\boldsymbol{M}^\top$	Transposed of matrix $\boldsymbol{M}$

Chapter 1

Introduction

Arbitrage is one of the fundamental concepts in finance. Sharpe et al. [1990] define it as "the simultaneous purchase and sale of the same, or essentially similar, security in two different markets for advantageously different prices". Thus, arbitrage exploits violations of the law of one price, stating that in an perfectly efficient market, two assets with the same risk-return profile should be priced identically.

Statistical Arbitrage encompasses a range of short-term investment strategies that share particular features. As a quantitative and particularly computational approach, they are mainly based on statistical methods, data mining techniques and automated trading systems. In contrast to the traditional interpretation, statistical arbitrage seeks to identify mispricings arising from deviations from common stochastic trends. These kind of investment strategies are almost market neutral, meaning that the achieved returns are independent of the overall market returns.

Pairs Trading is the simplest possible form of a statistical arbitrage strategy and belongs to the family of so-called long-short investments. It represents a popular speculative investment strategy that is commonly used in financial markets by traders and hedge funds. As the name already suggests, two assets are traded according to a particular rule. One of them is always sold short, whereas the proceeds are used to purchase the other asset. Both positions are closed again at a future point in time when a predefined condition is met.

Of course, this raises many questions. How can suitable pairs of assets be identified? Do they have to possess any specific properties? When is the right time to enter and exit a trade? As the history of pairs trading reaches back the 1980s, there already exist many different approaches trying to answer these and many other questions related to this topic. One of the earliest academic papers on pairs trading is the working paper of Gatev et al. [1999], which was published in The Review of Financial Studies seven years later. In the meantime, the literature on pairs trading frameworks has grown steadily and many other academic papers and even books have been published.

In this thesis, we will focus on the identification of suitable pairs with the help of different statistical techniques, which is known to be a crucial point for the success of a pairs trading strategy. In the statistical context, pairs of assets are often identified by measuring the distance between their price paths in terms of the sum of squared deviations. We are going to introduce several more sophisticated distance measures which try to take into account

the main stylized facts of financial time series. The concept of distance or (dis)similarity measures is naturally linked to the technique of clustering. However, cointegration represents another rigorous framework for pairs trading but from the statistical point of view, it leads to difficulties in the application on high-dimensional data sets, i.e., on broad asset universes. Therefore, we are going to combine the technique of time series clustering together with various distance measures and the concept of cointegration in a meaningful way in order to identify tradable pairs.

The remainder of this thesis is structured as follows: First, Chapter 2 presents a couple of well-known stylized facts of financial time series. Chapter 3 then deals with the theory of cointegration and provides both, the general idea and two different testing procedures. As a potential response to the multiple testing problem which occurs when a great number of time series pairs is tested simultaneously for cointegration, we suggest to partition the considered asset universe first by forming groups of similar time series. For this purpose, Chapter 4 introduces the concept of time series clustering together with various representation methods and (dis)similarity measures which seem to be appropriate especially for financial time series. Furthermore, the two most popular classical clustering methods, hierarchical and partitional clustering, are addressed, followed by an overview of several criteria to validate the goodness of a clustering. Chapter 5 proceeds with a general introduction to the strategy of pairs trading and provides a brief review of existing literature. We implement the previously described theoretical concepts in the statistical programming language `R` and subsequently backtest them in an empirical application on the Morgan Stanley Capital International (MSCI) Europe Index constituents. Daily data is available for the period from January 04, 1999 to December 31, 2018. The obtained results are analyzed in great detail. Chapter 6 concludes the thesis and discusses both, the theoretical and practical findings.

Chapter 2

Stylized Facts of Financial Time Series

Modeling financial time series is a complex problem. There exist many different kinds of financial time series such as stock prices, interest rates and exchange rates, among others, which are observed at various frequencies. The majority of these time series follows specific statistical regularities. However, these so-called *stylized facts* are hard to reproduce artificially even by using advanced stochastic models. They can be observed more or less clearly depending on the frequency and on the type of financial time series. The following stylized facts are mainly linked to series of daily stock prices and have been broadly discussed in the literature during the last decades.

Non-stationarity of price series

Sample price paths of many financial assets are non-stationary with a unit root making them close to a random walk. This fact is clearly visible in the sample price path in Figure 2.0.1. It shows the historic daily closing prices of the Standard & Poor's 500 Index (S&P 500) for the period from January 2, 1990 to December 31, 2018. The price series is adjusted for all applicable stock splits and dividend distributions.

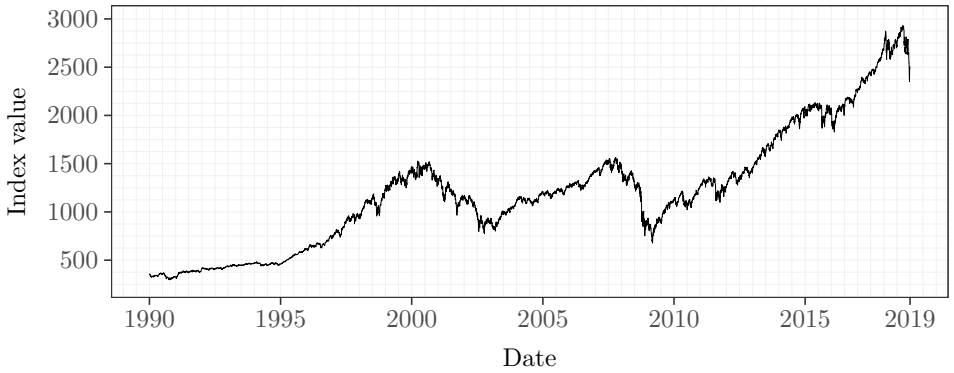

Figure 2.0.1 Historic adjusted daily closing prices of the S&P 500 for the period from January 2, 1990 to December 31, 2018.

Volatility clustering

As already noted by Mandelbrot [1963], "large changes tend to be followed by large changes –of either sign– and small changes tend to be followed by small changes", meaning that sub-periods of high volatility are followed by low-volatility periods, whereas the former

are recurrent but do not appear periodically. Hence, volatility changes over time and its degree tends to persist for some time which results in so-called volatility clusters. This property is known as autoregressive conditional heteroscedasticity and can be also observed in the sample path of the S&P 500 returns depicted in Figure 2.0.2.

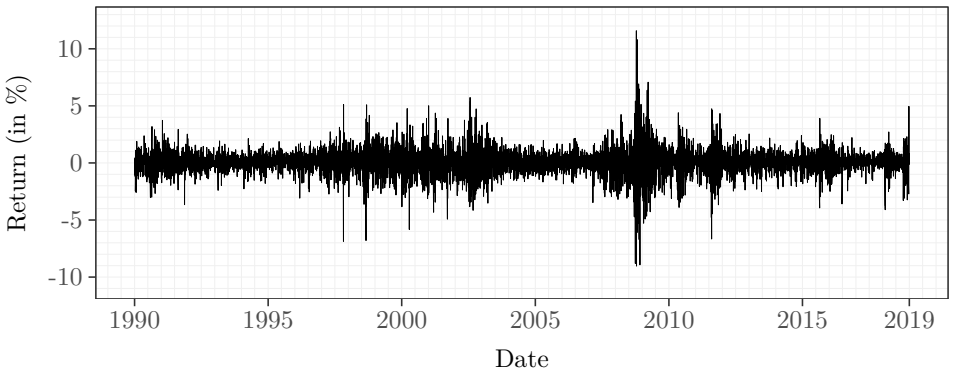

Figure 2.0.2 S&P 500 daily discrete returns (January 3, 1990 to December 31, 2018).

Absence of autocorrelation for price variations

Another widely accepted stylized fact of daily (raw) return series is a very small and mostly insignificant autocorrelation, making them close to a white noise. This fact is illustrated in Figure 2.0.3, showing the sample autocorrelation of the daily returns of the S&P 500. It is clearly visible that almost all lags considered are not significantly different from zero. As opposed to this, series at higher frequencies measured in minutes or seconds often exhibit a significant autocorrelation due to microstructure effects such as the bid-ask bounce (Satchell and Knight [2011]).

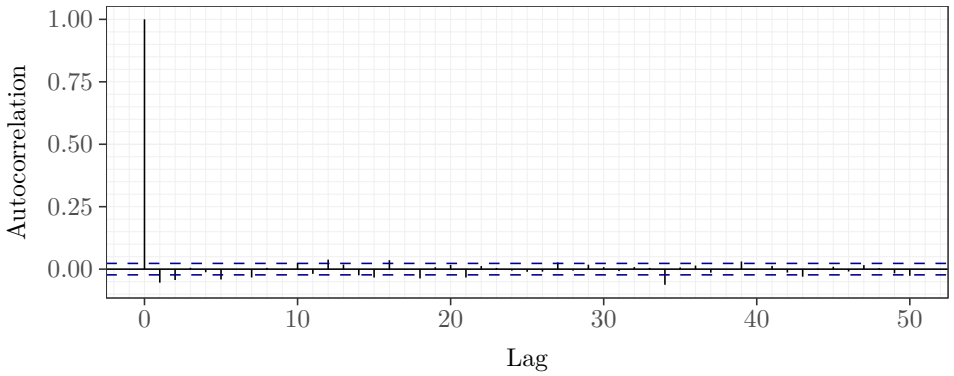

Figure 2.0.3 Sample autocorrelation of daily returns of the S&P 500 (January 3, 1990 to December 31, 2018).

Autocorrelation of absolute and squared returns

In contrast to the lack of autocorrelation in the (raw) returns, absolute and squared returns are typically strongly autocorrelated. The existing autocorrelation is always positive and decays slowly with an increasing time lag causing the appearance of the volatility clusters. The sample autocorrelation of the squared daily returns of the S&P 500 is depicted in Figure 2.0.4, showing a conspicuous persistence and a slow decay. Furthermore, the first two rows

of Table 2.0.1 contain selected values of the sample autocorrelation of absolute and squared daily returns. It follows that the autocorrelation of absolute returns tends to be larger than the autocorrelation of squared returns, implying a higher predictability of absolute returns. This feature is also known as "Taylor effect" (Thompson [2013]).

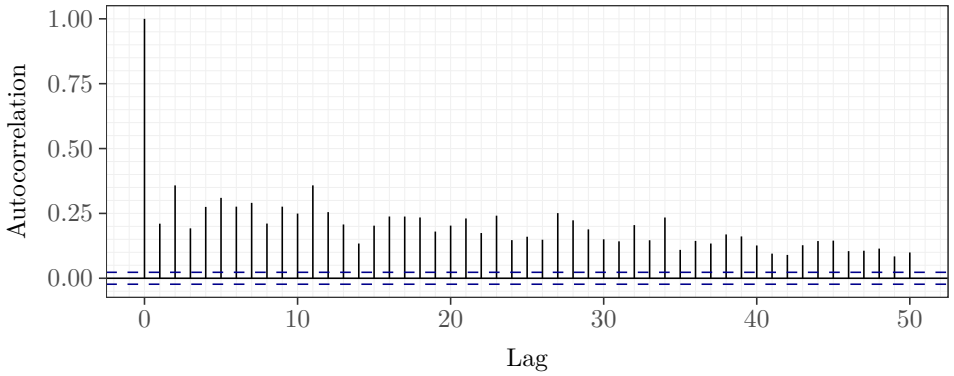

Figure 2.0.4 Sample autocorrelation of squared daily returns of the S&P 500 (January 3, 1990 to December 31, 2018).

Fat-tailed distribution

Another well-known feature of daily return series is a fat-tailed empirical distribution which is also sharply peaked at zero. This property is called leptokurtosis. The kurtosis is the fourth standardized moment of a distribution and asymptotically takes a value of 3 for *i.i.d.* Gaussian observations but is much greater for daily return series. Classical statistical tests such as the Jarque-Bera test can be performed to verify this. They typically reject the null hypothesis of normality at any reasonable level of significance. Additionally, the empirical distribution of daily returns is often slightly skewed (Francq and Zakoian [2011]). Figure 2.0.5 compares the histogram of the daily returns of the S&P 500 with a Gaussian density. The peak around zero is clearly visible but the fat-tails are more difficult to visualize. In this case, the kurtosis of the underlying empirical distribution takes a value of about 9.

Table 2.0.1 Sample (auto-) correlation of the S&P 500 returns: Taylor and Leverage effect.

h	1	2	3	4	5	10	25	50
$\hat{\rho}_{\lvert r\rvert}(h)$	0.241	0.318	0.275	0.289	0.316	0.275	0.210	0.169
$\hat{\rho}_{r^2}(h)$	0.210	0.358	0.192	0.275	0.310	0.249	0.160	0.100
$\hat{\rho}(\lvert r_{t+h}\rvert, r_t^+)$	0.045	0.099	0.103	0.111	0.128	0.107	0.089	0.095
$\hat{\rho}(\lvert r_{t+h}\rvert, -r_t^-)$	0.240	0.278	0.223	0.232	0.248	0.219	0.161	0.106

Leverage effect

The last stylized fact discussed refers to an inverse relationship between stock prices and volatility. As the price of a stock decreases, volatility tends to increase, and vice versa, but there seems to be an asymmetry in the size of the effect. A decrease in the stock price (negative return) tends to increase the volatility by a larger amount than an increase in the stock price (positive return) of the same magnitude. Thus, the sample correlation between

$r_t^+ = \max(r_t, 0)$ and $|r_{t+h}|$ is generally lower than between $-r_t^- = \max(-r_t, 0)$ and $|r_{t+h}|$ (Francq and Zakoian [2011]). The last two rows of Table 2.0.1 contain selected values of the corresponding sample correlations and confirm this fact.

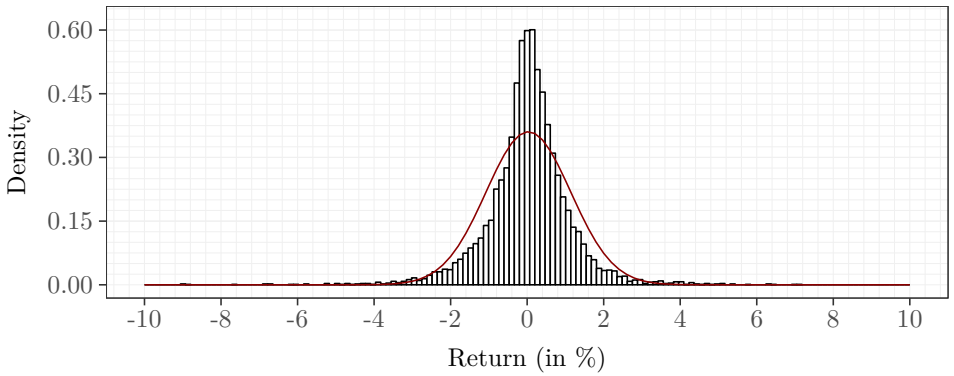

Figure 2.0.5 Histogram of the S&P 500 daily returns and density of a Gaussian distribution with mean and variance equal to the sample mean and variance of the returns (red line).

Chapter 3

Theory of Cointegration

3.1 Stationarity, Integration and Cointegration

Especially in economics and finance, many applications are concerned with dynamic modeling. There has been a lot of research and a growing interest in the area of time series analysis during the last decades. In this thesis, we also extensively deal with time series data, so let us define our main building block first:

Definition 3.1.1 – Time Series (Brockwell and Davis [2013]).
A discrete time series can be defined as a sequence of random variables:

$$\{X(\omega, t),\, \omega \in \Omega,\, t \in \mathbb{N}\} \tag{3.1.1}$$

with sample space Ω. Each observation is indexed by a date, which implies a natural temporal ordering.

In general, time series can not be observed just at equidistant points in time, but the above provided definition will be sufficient for our purpose. An observed time series $x_t = (x_1, \ldots, x_T)$ can be considered as a realization of the random vector $X_t = (X_1, \ldots, X_T)$. Henceforth, we denote the former by $\boldsymbol{x}_t$ and the latter by $\boldsymbol{X}_t$ in order to keep notation simple.

To model time series and apply standard inference procedures, time series under investigation have to be *stationary*. The majority of econometric theory relies on this assumption, which guarantees that the characteristics of a time series do not change over time:

Definition 3.1.2 – Stationarity (Brockwell and Davis [2013]).
A time series is said to be strictly stationary, if the joint distribution of $(X_{t_1}, X_{t_2}, \ldots, X_{t_n})$ corresponds to the joint distribution of $(X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_n+h})$, $\forall\, t_1, \ldots t_n, n, h$, meaning that the joint distribution does not vary over time. Weak stationarity, by contrast, requires just the following conditions:

(i) $\mathbb{E}[X_t] = \mu,\ \forall\, t,$
(ii) $\mathbb{E}[X_t^2] = \sigma^2 < \infty,\ \forall\, t,$ *and*
(iii) $\mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)] = \gamma_X(h),\ \forall\, t, h.$

In practice, strict-sense stationarity is too restrictive for many applications. Therefore, the concept of weak stationarity is usually applied. Assuming the second moment of a time

series is finite, then strict stationarity also implies weak stationarity, but not vice versa. Weak stationarity is often referred to as covariance stationarity or second-order stationarity. Henceforth, we use the term "stationarity" to refer to the weak form of stationarity.

If a time series violates the stationarity conditions, i.e., its unconditional mean or variance is not constant over time, it is said to be integrated. The order of integration, denoted by $I(d)$, represents a summary statistic for a time series and reports the minimum number d of differences required in order to transform a non-stationary time series into a stationary one.

Differencing time series seems to be a prerequisite for econometric modeling. However, in the multivariate case there may exist a linear combination of non-stationary time series, which, by contrast, is stationary without taking differences. This property is called cointegration. Engle and Granger [1987] were the first ones to formalize this concept:

Definition 3.1.3 – Cointegration (Engle and Granger [1987]).
The components of a $(k \times 1)$ *vector* $\boldsymbol{X}_t$ *are said to be cointegrated of order* (d, b)*, denoted by* $CI(d, b)$*, if*

(i) each component individually taken is $I(d)$*, and*

(ii) a vector $\boldsymbol{\beta} = (\beta_1, \dots \beta_k) \neq 0$ *exists such that the linear combination* $\boldsymbol{Z}_t \coloneqq \boldsymbol{\beta}^\top \boldsymbol{X}_t$ *is* $I(d - b)$*.*

The linear combination is called cointegration relationship and the vector $\boldsymbol{\beta}$ *is referred to as cointegrating vector.*

Cointegration means that two or more time series are linked to form a long-run equilibrium, which is represented by the linear combination $\boldsymbol{\beta}^\top \boldsymbol{X}_t$. Although the individual components may contain stochastic trends, they closely move together over time and show only short-term deviations from their equilibrium (Harris and Sollis [2003]).

Error correction models (ECMs) are closely related to the concept of cointegration. Suppose that $X_{1,t}$ and $X_{2,t}$ are non-stationary time series and their equilibrium relation is given by $X_{1,t} = \beta X_{2,t}$. In an ECM, the changes in both series in period t depend on their deviations from the equilibrium in the previous period:

$$\begin{aligned} \Delta X_{1,t} &= \alpha_1 (X_{1,t-1} - \beta X_{2,t-1}) + u_{1,t} \\ \Delta X_{2,t} &= \alpha_2 (X_{1,t-1} - \beta X_{2,t-1}) + u_{2,t}, \end{aligned} \tag{3.1.2}$$

where $\alpha_i,\ i = 1, 2$ represent the error correction rates and $u_{i,t},\ i = 1, 2$ are error terms which are assumed to be white noise. The latter cause the deviations from the long-run equilibrium, whereas the former indicate the speed with which the time series adjust themselves to restore the equilibrium. A more general ECM may also include previous changes of $\Delta X_{i,t},\ i = 1, 2$ as explanatory variables (Lütkepohl [2007]).

Now suppose that $X_{t,1}$ and $X_{t,2}$ are both $I(1)$. Thus, the left hand side of Equation 3.1.2 as well as the white noise errors are stationary. Since a non-stationary term can not equal a stationary process,

$$\alpha_i (X_{1,t-1} - \beta X_{2,t-1}) = \Delta X_{i,t} - u_{i,t}$$

must be also stationary. Hence, if $\alpha_1 \neq 0$ or $\alpha_2 \neq 0$, then the term $X_{1,t} - \beta X_{2,t}$ is

stationary and represents a cointegration relationship (Lütkepohl [2007]). The fact that cointegration and error correction are two equivalent representations is stated by the Granger representation theorem (Engle and Granger [1987]).

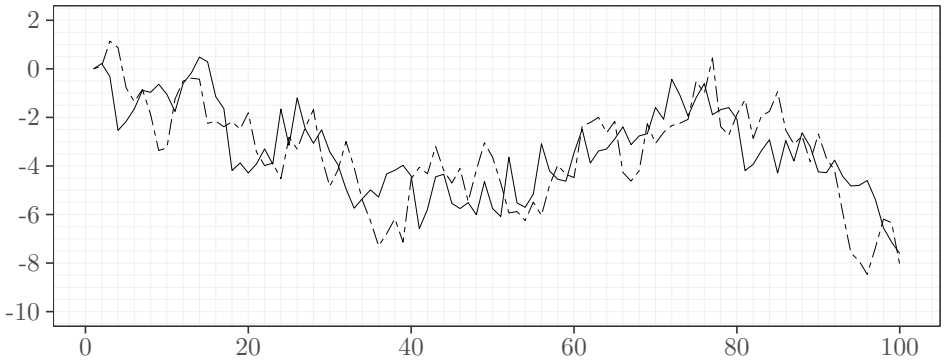

Figure 3.1.1 Simulated bivariate cointegrated time series.

The above described framework can be naturally extended to a higher dimension. In general, multiple cointegration relationships may exist, but the number of relationships is always smaller than the number of processes considered. Figure 3.1.1 shows two cointegrated time series generated from the model in Equation 3.1.2 with $\alpha_1 = 0.25$, $\alpha_2 = -0.25$, $\beta = 1$ and $u_{i,t} \sim \mathcal{N}(0,1),\ i = 1,2$. Furthermore, the corresponding equilibrium relation $X_{1,t} - X_{2,t}$ is depicted in Figure 3.1.2, which seems, indeed, to be stationary.

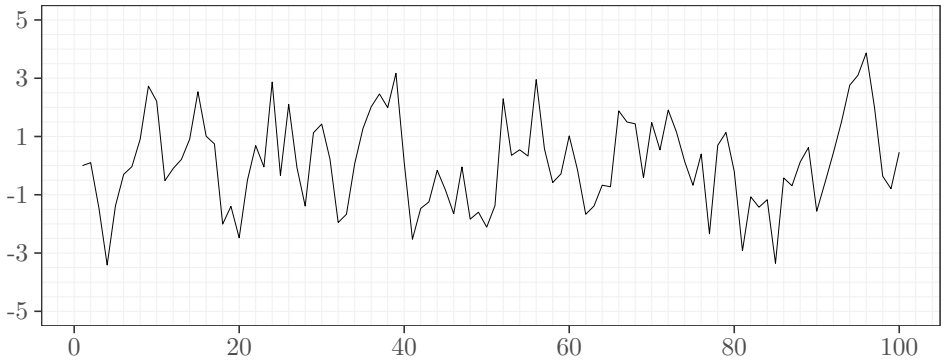

Figure 3.1.2 Equilibrium relation of the bivariate cointegrated time series.

In finance, price paths of many assets are usually assumed to be $I(1)$. Here, cointegration means that even if the individual price paths are non-stationary, there may exist portfolios of assets that are stationary. The concept of cointegration is commonly used within the context of pairs trading (PT). If price paths of two assets are cointegrated, then the corresponding price spread is usually stationary and exhibits a mean-reverting behavior.

The selection of suitable pairs is the key to success for this kind of trading strategy. The associated statistical testing for cointegration can be performed by many different tests with each of them having specific advantages and disadvantages. In the next section, we are going to discuss two residual based testing procedures which may be appropriate for our purpose.

3.2 Testing Procedures

3.2.1 The Engle-Granger Approach

Engle and Granger [1987] developed a two-step testing procedure that has become very popular in econometric modeling. It is used in order to test the null hypothesis of no cointegration between two time series.

Suppose that $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are non-stationary time series and both are integrated of order one. If necessary, both series can be prior tested to verify this. The first step of the testing procedure consists in estimating the long-run equilibrium relationship in form of an ordinary least squares (OLS) regression:

$$y_t = \alpha + \beta x_t + u_t, \tag{3.2.1}$$

where β is the cointegrating coefficient and u_t is an error term. Note that the estimation results of the OLS regression are only reliable, if both time series are cointegrated. In this case, the OLS estimator of β is said to be super-consistent, i.e., it converges to the true parameter much faster than in the standard case with $I(0)$ variables. In the absence of cointegration this technique leads to the problem of spurious regression and may provide misleading results. The estimated cointegrating regression yields the residual series $\hat{u}_t = y_t - \hat{\alpha} - \hat{\beta} x_t$.

In the second step of the testing procedure, a unit root test is applied to the residuals $\hat{u}_t$ in order to determine whether they are stationary or not. For this purpose, an augmented Dickey–Fuller (ADF) test is usually performed on the following model:

$$\Delta \hat{u}_t = \psi^* \hat{u}_{t-1} + \sum_{i=1}^{p-1} \psi_i \Delta \hat{u}_{t-i} + \varepsilon_t, \tag{3.2.2}$$

where ε_t is assumed to be white noise (Harris and Sollis [2003]).

As with univariate unit root tests, the null hypothesis of $\psi^* = 1$ is tested against the alternative of $-1 < \psi^* < 1$. In the present case, the test statistic follows a non-standard distribution under the null hypothesis. Dickey and Fuller [1979] investigated this problem and used Monte-Carlo (MC) simulations to generate the corresponding critical values for a range of sample sizes and significance levels. If the null hypothesis of non-stationary of the residuals $\hat{u}_t$ can be rejected, then they can be considered as stationary, which further implies that the investigated time series $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are cointegrated.

As the Engle-Granger approach is a single equation approach, it is inefficient when more than two time series are investigated, which have more than one cointegration relationship. In this case the Johansen test would rather be used, which can be seen as a multivariate generalization of the ADF test. Anyway, this problem is not relevant in our case, since we only need to test pairs of time series.

Moreover, the testing procedure is not invariant to the chosen normalization of the regression equation, i.e., which time series is taken to be the dependent variable. In case of

only two time series, $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, it is possible to run the following two regressions:

$$y_t = \alpha_1 + \beta_1 x_t + u_{1,t}$$
$$x_t = \alpha_2 + \beta_2 y_t + u_{2,t}$$

and test either the residuals $\hat{u}_{1,t}$ or $\hat{u}_{2,t}$ for a unit root. As the sample size goes to infinity, the theory indicates that both tests become equivalent. However, large sample properties are usually not applicable in practice due to relatively small sample sizes. In the next section, we are going to introduce a test proposed by Phillips and Ouliaris [1990], which allows to circumvent this problem (Harris and Sollis [2003]).

3.2.2 Phillips-Ouliaris Methods

Phillips and Ouliaris [1990] proposed two statistical tests for cointegration, namely the variance ratio test and the multivariate trace statistic. Both are used in order to test the null hypothesis of no cointegration, whereas the latter has the great advantage that it is invariant to the chosen normalization of the regression equation. The multivariate trace statistic is based on the residuals of a first order vector autoregressive model (VAR):

$$\boldsymbol{z}_t = \boldsymbol{\Pi} \boldsymbol{z}_{t-1} + \boldsymbol{\xi}_t, \tag{3.2.3}$$

where $\boldsymbol{z}_t = (x_t, \boldsymbol{y}_t^\top)^\top$ is an $m \times 1$ vector of time series partitioned into the scalar variate x_t and the vector $\boldsymbol{y}_t$.[1] Furthermore, $\boldsymbol{\Pi}$ is a matrix of regression coefficients and $\boldsymbol{\xi}_t$ is the residual vector. The conditional covariance matrix $\boldsymbol{\Omega}$ of $\boldsymbol{\xi}_t$ can be estimated as follows:

$$\hat{\boldsymbol{\Omega}} = T^{-1} \sum_{t=1}^{T} \hat{\boldsymbol{\xi}}_t \hat{\boldsymbol{\xi}}_t^\top + T^{-1} \sum_{s=1}^{l} \omega(s,l) \sum_{t=s+1}^{T} \left(\hat{\boldsymbol{\xi}}_t \hat{\boldsymbol{\xi}}_{t-s}^\top + \hat{\boldsymbol{\xi}}_{t-s} \hat{\boldsymbol{\xi}}_t^\top \right)$$

with total number of observations T, maximum lag l and weighting function $\omega(s,l) = 1 - s \cdot (l+1)^{-1}$. The multivariate trace statistic is then defined as:

$$\hat{P}_z = T \cdot tr\left(\hat{\boldsymbol{\Omega}} \, \boldsymbol{M}_{zz}^{-1} \right),$$

where $\boldsymbol{M}_{zz} = T^{-1} \sum_{t=1}^{T} \boldsymbol{z}_t \boldsymbol{z}_t^\top$ is the observed sample moment matrix. Note that the test statistic $\hat{P}_z$ is constructed in the form of Hotelling's T_0^2 statistic, which is commonly used in multivariate analysis.

Recall the problem of spurious regression under the null hypothesis of the test, i.e., in the absence of cointegration. Phillips and Ouliaris [1990] showed that in this case, unit root tests such as the ADF test applied to the residual series obtained from the cointegrating regression do not have the usual Dickey-Fuller distribution. The resulting limiting distributions are characterised by Wiener processes and are known as Phillips-Ouliaris distributions. Again, critical values were generated by MC simulations for a range of significance levels. The

[1] The partitioning of the vector $\boldsymbol{z}_t$ is useful in order to construct the variance ratio test. See Phillips and Ouliaris [1990], pp. 169-171 for further details.

corresponding values for classical unit root tests as well as for the variance ratio test and the multivariate trace statistic are tabulated in Phillips and Ouliaris [1990].

3.2.3 The Multiple Testing Problem

Suppose we were to pursue a simple PT strategy and could select our pairs from a broad universe of several hundred assets. We would aim to find pairs of assets, whose price paths are cointegrated, and would then trade some of them. Suitable pairs could be detected by performing a pairwise cointegration test, ideally on all possible combinations of assets.

From the statistical point of view, this procedure is certainly problematic since it leads to the well-known multiple testing problem. The reason for that is the following:

Let us assume our universe consists of 100 assets, whereas not even one cointegration relationship exists between all possible pairs of assets. We decide to use the multivariate trace statistic to detect cointegrated pairs, since this test is invariant to the chosen normalization of the regression equation.

If we initially test just 25 randomly selected pairs at a significance level of $\alpha = 5\%$, then the probability of rejecting the null hypothesis at least once just by chance, i.e., observing at least one significant cointegrated pair, amounts to:

$$\mathbb{P}(\text{at least one significant result}) = 1 - \mathbb{P}(\text{no significant results}) = 1 - (1 - 0.05)^{25} \approx 0.72.$$

However, if we want to test all possible combinations of assets, we have to perform $\frac{100!}{2! \cdot 98!} = 4950$ statistical tests, yielding about 250 pairs at a significance level of $\alpha = 5\%$. These pairs are assumed to be cointegrated and would, therefore, be eligible for PT, although we assumed that not a single cointegration relationship truly exists.

The underlying problem consists not in our chosen cointegration test but rather in the great amount of tests we have to perform. Within a cointegration-based PT strategy, even well established methods to counteract the multiple testing problem turn out to be far too conservative. The popular Bonferroni correction would, for instance, test each individual null hypothesis at $\alpha = \frac{5\%}{4950} \approx 0.001\%$ in the aforegoing case with a universe of just 100 assets. This makes it impossible to detect even truly cointegrated and potentially high profitable pairs (Harlacher [2016]).

One way to mitigate this problem could be to effectively pre-partition the considered asset universe. For this purpose, we are going to introduce the concept of time series clustering in the next chapter. Forming clusters of time series naturally leads to a reduction of possible combinations and, therefore, the number of statistical tests to be performed. Furthermore, we expect to find cointegrated pairs more easily by first testing the most similar time series of each cluster, i.e., those pairs whose dissimilarity measure takes a small value.

Chapter 4

Time Series Clustering

4.1 General Remarks

Cluster analysis or clustering is considered as one of the most important techniques of unsupervised learning. It is performed to detect groups of observations in data so that observations assigned to the same cluster are more similar to each other than to those in other clusters. In order to determine whether observations are similar, it is necessary to decide on a measure quantifying similarity. The selection of a suitable (dis)similarity measure is a crucial point as it mainly depends on the specific purpose of the clustering task and essentially affects the outcome of the process. Furthermore, a clustering method must be chosen. There exist several different techniques, but the two most popular classical methods appear to be hierarchical and partitional clustering. As the number of clusters is usually unknown, it needs to be determined on the basis of certain criteria measuring the quality of clusters according to their compactness and their degree of separation.

Cluster analysis is used in lots of different fields, including medicine, economics, finance, marketing, and genetics, among others. Its applications range from market segmentation and insurance fraud detection to object recognition and social network analysis. In the present case, we want to apply clustering to a data set of univariate financial time series, i.e., series of stock prices or returns. In contrast to static data, time series are of a dynamic nature and therefore, the behavior of the series over time must be taken into account.

Especially clustering of whole time series is regarded as a challenging issue. Often, time series data are collected over long periods of time forming huge data sets which can be far larger than available memory size. High-dimensional data sets are difficult to handle for many clustering algorithms and thus lead to a substantial decrease in speed of the clustering process. Furthermore, the calculation of the entire dissimilarity matrix can also be extremely time-consuming and computationally intensive, depending on the chosen dissimilarity measure and its time complexity (Aghabozorgi et al. [2015]).

There exist three different ways to cluster time series data: raw data-based, feature-based and model-based approaches. Moreover, plenty of dissimilarity measures are proposed in literature with each of them having specific advantages and disadvantages (Liao [2005]). All three approaches will be discussed below. The main focus lies on selected representation

methods and dissimilarity measures which seem to be appropriate especially for financial time series and which try to take into account the stylized facts described in Chapter 2.

Henceforth, the notation $d(\boldsymbol{x}_t, \boldsymbol{y}_t)$ is used to represent the distance between $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$. The function d is considered a distance metric, if it satisfies the following classical properties for all time series $\boldsymbol{x}_t \in \mathbb{R}^n$, $\boldsymbol{y}_t \in \mathbb{R}^m$ and $\boldsymbol{z}_t \in \mathbb{R}^k$:

Definition 4.1.1 – Properties of a metric (Deza and Deza [2014]).

(i) Non-negativity: $d(\boldsymbol{x}_t, \boldsymbol{y}_t) \geq 0$,

(ii) Symmetry: $d(\boldsymbol{x}_t, \boldsymbol{y}_t) = d(\boldsymbol{y}_t, \boldsymbol{x}_t)$,

(iii) Reflexitivity: $d(\boldsymbol{x}_t, \boldsymbol{y}_t) = 0 \Longleftrightarrow \boldsymbol{x}_t = \boldsymbol{y}_t$, *and the*

(iv) Triangle inequality: $d(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq d(\boldsymbol{x}_t, \boldsymbol{z}_t) + d(\boldsymbol{z}_t, \boldsymbol{y}_t)$.

Of course, the value of $d(\boldsymbol{x}_t, \boldsymbol{y}_t)$ differs depending on the dissimilarity measure chosen. Note that many dissimilarity measures do not necessarily require time series of equal length.

4.2 Representation Methods and Dissimilarity Measures

4.2.1 Raw Data-Based Approaches

In contrast to feature-based and model-based techniques, raw data-based approaches work directly with the raw time series data by comparing the overall shapes of the series. In this way, there occurs no loss of information. Often, it is necessary to operate on high-dimensional spaces which requires sufficient computing power and memory space.

4.2.1.1 Minkowski Distance

The Minkowski distance is the L_p-norm ($1 \leq p < \infty$) of the difference between two time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$ of equal length ($n = m$). It is defined as follows:

Definition 4.2.1 – Minkowski distance (Gan et al. [2007]).

$$d_{mink}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}. \tag{4.2.1}$$

It can be considered as a generalization that unifies Manhattan ($p = 1$), Euclidean ($p = 2$) and Chebyshev ($p \to \infty$) distance. The formula of the Euclidean distance is given by:

Definition 4.2.2 – Euclidean distance (Gan et al. [2007]).

$$d_{euc}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{4.2.2}$$

The Euclidean distance is perhaps the most widely used dissimilarity measure in clustering which is mainly due to its linear time complexity and competitiveness for many problems. Since the Euclidean distance is not invariant to scaling, each individual time series must be

z-normalized by subtracting its mean and dividing by its standard deviation. This operation removes a potential offset as well as differences in the amplitude, meaning that the resulting time series only differ in their natural shape (Tapinos and Mendes [2013]).

Within the temporal context, the operation carried out by the Euclidean distance consists in matching each point of one time series with the corresponding point of another series at the same time. Given two series with the same shape but slightly shifted in time, the distance is clearly different from zero. Thus, the Euclidean distance only exploits similarity in time, while similarity in shape and in change are disregarded (Aggarwal and Reddy [2018]).

4.2.1.2 Dynamic Time Warping

Dynamic time warping (DTW) is a technique to find the optimal alignment between two time series under certain restrictions. Two time series are warped non-linearly in time in order to match their shapes as well as possible. This way, also time series of unequal length can be compared.

DTW was first introduced by Vintsyuk [1968] in the field of speech recognition. Originally, it was used to match words at different speaking rates but it has also been successfully applied in many other fields dealing with time-dependent data such as handwriting recognition (Rath and Manmatha [2003]), gene expression analysis (Aach and Church [2001]) and data mining (Keogh and Pazzani [2000]).

When determining the DTW distance between a pair of time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$, first of all a local cost matrix (LCM) $\boldsymbol{C} \in \mathbb{R}^{n \times m}$ with corresponding entries $\boldsymbol{C}(i,j) \coloneqq c(x_i, y_j)$ is calculated. Euclidean or Manhattan distance is usually used as local cost measure c. Since DTW is also not invariant to scaling, z-normalized versions of time series must be always provided.

The goal consists in finding an optimal alignment between both time series having minimum overall cost. This alignment is called *warping path* $W = (w_1, w_2, \ldots, w_k)$ with $w_\ell = (p_\ell, q_\ell) \in [1:n] \times [1:m]$ for $\ell \in \{1, \ldots, k\}$ and $\max(n,m) \leq k \leq n+m-1$. The warping path traverses the entire LCM and is subject to the following three conditions:

Definition 4.2.3 – DTW path constraints (Müller [2007]).
(i) Boundaries: $w_1 = (1,1)$ *and* $w_k = (n,m)$.
(ii) Monotonicity: $p_1 \leq p_2 \leq \ldots \leq p_k$ *and* $q_1 \leq q_2 \leq \ldots \leq q_k$.
(iii) Continuity: $w_{\ell+1} - w_\ell \in \{(1,0),(0,1),(1,1)\}$ *for* $\ell \in \{1, \ldots, k-1\}$.

The boundary condition enforces that the first and last elements of both time series are aligned to each other. Thus, the resulting warping path starts and ends in two opposite corners of the LCM. The monotonicity condition states that subsequent steps of the warping path must be monotonically spaced in time. This condition is implied by the continuity condition which requires that only adjacent elements in the LCM are allowed as steps in the warping path. Hence, no element of both time series can be omitted.

The total cost of a warping path equals the sum of individual costs of all LCM elements that are traversed by the path. The optimal warping path is given by the path having minimum total cost among all possible paths. It can be calculated by an algorithm based

on dynamic programming. Given $\boldsymbol{D}(i,1) = \sum_{h=1}^{i} c(x_h, y_1)$ for $i \in \{1,\dots,n\}$ and $\boldsymbol{D}(1,j) = \sum_{h=1}^{j} c(x_1, y_h)$ for $j \in \{1,\dots,m\}$, then the accumulated cost matrix $\boldsymbol{D}$ is given by:

$$\boldsymbol{D}(i,j) = c(x_i, y_j) + \min\{\boldsymbol{D}(i-1,j-1),\, \boldsymbol{D}(i-1,j),\, \boldsymbol{D}(i,j-1)\} \tag{4.2.3}$$

for $1 < i \leq n$ and $1 < j \leq m$. By extending $\boldsymbol{D}$ with an additional column and row and formally setting $\boldsymbol{D}(i,0) \coloneqq \infty$ for $i \in \{1,\dots,n\}$, $\boldsymbol{D}(0,j) \coloneqq \infty$ for $j \in \{1,\dots,m\}$ and $\boldsymbol{D}(0,0) \coloneqq 0$, the recursion of Equation 4.2.3 holds for $1 \leq i \leq n$ and $1 \leq j \leq m$. The optimal warping path $W = (w_1, w_2, \dots, w_k)$ can be calculated in reverse order starting at $w_k = (n,m)$ and ending at $w_1 = (1,1)$ (Müller [2007]).

Thus, the DTW distance between time series $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ is given by:

Definition 4.2.4 – DTW distance (Keogh and Ratanamahatana [2005]).

$$d_{dtw}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \min \sqrt{\sum_{\ell=1}^{k} \boldsymbol{C}(w_\ell)}, \tag{4.2.4}$$

where $\boldsymbol{C}(w_\ell)$ denotes the cost of the ℓ-th element of the warping path and the square root is taken to scale the total cost.

Even though the optimal path can be computed efficiently via dynamic programming, the time complexity of the algorithm is still quadratic. The comparatively high running time can make the calculation of entire distance matrices a time-consuming process. For this reason, many techniques to speed up computations and control the possible routes of the warping path are proposed in literature. These include, among others, adding local or global constraints.

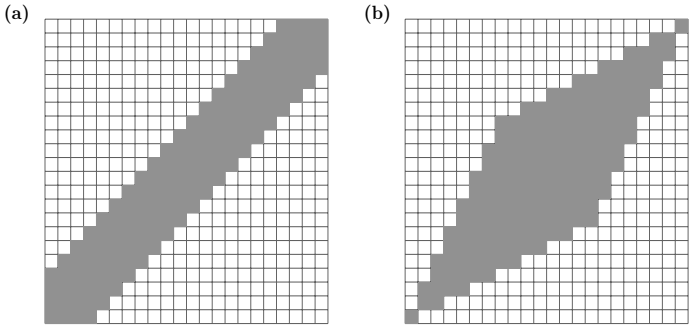

Figure 4.2.1 Global constraints: (a) Sakoe-Chiba band of width $T = 4$ and (b) Itakura parallelogram with $S = 2$.

Local constraints affect the slope of the warping path. They modify the continuity condition (iii) and allow also other steps, which can be either symmetric or asymmetric. If the above-noted condition is replaced by $w_{\ell+1} - w_\ell \in \{(2,1),(1,2),(1,1)\}$ for $\ell \in \{1,\dots,k-1\}$, this results in warping paths having a local slope with lower and upper bounds of 0.5 and 2 instead of 0 and 1. Note that in this case, some elements of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ might be omitted since not every element of $\boldsymbol{x}_t$ needs to be aligned to some element of $\boldsymbol{y}_t$, and vice versa. Of course, there exist also several other possible step size conditions which avoid omissions but still impose restrictions on the slope of the warping path (Müller [2007]).

Another modification of DTW are global constraints, also called window constraints. They restrict the area of the LCM that can be reached by the algorithm. Two well-known windows are the Sakoe-Chiba band (Sakoe and Chiba [1978]) and the Itakura parallelogram (Itakura [1975]). The former is a band of fixed horizontal and vertical width $T \in \mathbb{N}$ running along the diagonal of the LCM, whereas the latter defines a region with the shape of a parallelogram. In this region the slope of the warping path takes a value between $1/S$ and S for some fixed $S \in \mathbb{R}_{>1}$. Both global constraints are depicted in Figure 4.2.1 with $T = 4$ and $S = 2$, respectively. Note that the usage of any constraints can be problematic since the optimal warping path may traverse regions of the LCM which are not covered by the constraints. This could lead to unsatisfactory or even useless alignments.

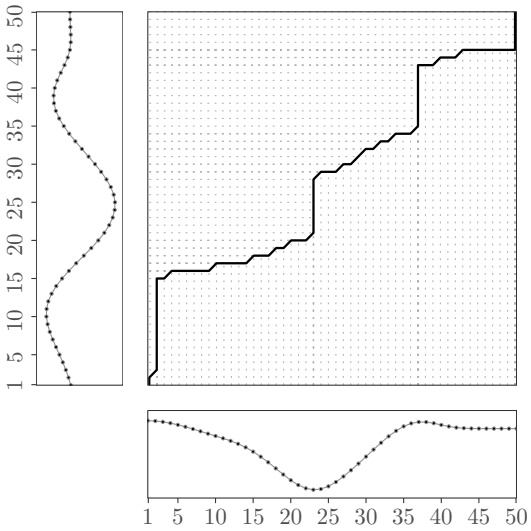

Figure 4.2.2 Optimal warping path (black line) of two aligned time series of equal length.

An example of DTW performed without any constraints is illustrated in Figure 4.2.2 and 4.2.3, respectively. The first Figure shows the optimal warping path (black line) of two aligned time series of length 50 which are depicted on the left and on the lower side of the matrix. It is clearly visible that the optimal path fulfills the three above-noted constraints. The resulting alignment between the individual points of both time series is depicted in the second Figure. As expected, several points of both time series are aligned to many other points of the other time series. This way, their slightly similar shapes are matched.

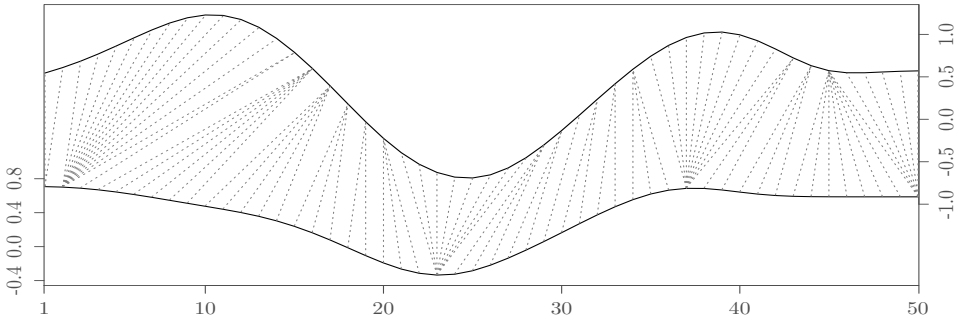

Figure 4.2.3 Matched points of two aligned time series of equal length.

4.2.1.3 Correlation

Another raw data-based technique is to consider the correlation between time series. The most common correlation coefficient, Pearson's correlation, measures the degree of linear association between two random variables X and Y. It is defined as follows:

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y},$$

where μ_X and σ_X denotes the mean and the standard deviation of X. Given two time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$ of equal length ($n = m$), its sample estimator is given by:

$$\hat{\rho}_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2}},$$

where $\bar{x} = n^{-1}\sum_{i=1}^{n} x_i$ denotes the sample mean of $\boldsymbol{x}_t$. Pearson's correlation coefficient can be computed in linear time and ranges from -1 to $+1$. The lower and upper bound is obtained if both series are perfectly (anti-)correlated, while $\rho_{x,y} = 0$ indicates no linear relationship.

A dissimilarity measure that takes into account the linear correlation between time series should assign a low distance to positively correlated time series, since they can be considered as more similar than negatively correlated ones. Thus, a correlation-based dissimilarity measure can be defined as:

Definition 4.2.5 – Pearson distance (Berthold and Höppner [2016]).

$$d_{cor}(\boldsymbol{x}_t, \boldsymbol{y}_t) = 1 - \hat{\rho}_{x,y} \tag{4.2.5}$$

such that $0 \leq d_{cor}(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq 2$.

However, Pearson's correlation measures just the contemporary correlation between time series but does not account for a potentially existing time shift. Therefore, Paparrizos and Gravano [2015] construct a dissimilarity measure based on the cross-correlation, which considers the optimal lag at which the value of the cross-correlation is maximized.

Given two z-normalized time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$ of equal length ($n = m$), Paparrizos and Gravano [2015] define the cross-correlation at lag k as follows:

$$\rho_{x,y}(k) = \frac{R_{x,y}(k)}{\sqrt{R_{x,x}(0)}\sqrt{R_{y,y}(0)}},$$

where

$$R_{x,y}(k) = \begin{cases} \sum_{i=1}^{n-k} x_{i+k}\, y_i, & k \geq 0 \\ R_{y,x}(-k), & k < 0 \end{cases}$$

for $-\ell \leq k \leq \ell$ and maximum time shift $\ell < n$. This yields a cross-correlation sequence of length $2\ell + 1$. Again, a dissimilarity measure similar to the Pearson distance can be defined:

Definition 4.2.6 – Cross-correlation distance (Paparrizos and Gravano [2015]).

$$d_{ccr}(\boldsymbol{x}_t, \boldsymbol{y}_t) = 1 - \max_k \hat{\rho}_{x,y}(k) \tag{4.2.6}$$

such that $0 \leq d_{ccr}(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq 2$.

Calculations of the dissimilarity measure seem to be very simple, but the time complexity is still quadratic. However, by choosing $\ell = n - 1$ and applying the (inverse) discrete Fourier transform together with the fast Fourier transform algorithm, computations can be carried out very efficiently in quasilinear time. See Paparrizos and Gravano [2015], p. 1860 for further details.

4.2.2 Feature-Based Approaches

Feature-based approaches aim at replacing raw time series data by a vector of extracted features of lower dimension. Similarity is then evaluated in terms of the features which usually differ depending on the particular field of application. This procedure can lead to a substantial decrease in computation time and memory allocation.

4.2.2.1 Classical Autocorrelation

Time series analysis is mostly performed in the time domain, but the frequency domain also offers an interesting alternative, which, however, will not be covered in this thesis. In the time domain, it is often useful to analyze the temporal dependence structure of a time series by considering its autocorrelation function. Autocorrelation can be seen as the correlation between two values of a single time series at different points in time, as a function of the time lag. Assuming a stationary time series $\boldsymbol{X}_t \in \mathbb{R}^n$, then the autocorrelation at lag k is defined as:

$$\rho_X(k) = \frac{\mathbb{E}[(X_{t+k} - \mu_X)(X_t - \mu_X)]}{\sigma_X^2},$$

where μ_X and σ_X^2 denotes the mean and the variance of $\boldsymbol{X}_t$. Given a time series $\boldsymbol{x}_t \in \mathbb{R}^n$, then the corresponding sample estimator is given by:

$$\hat{\rho}_x(k) = \frac{\sum_{i=1}^{n-k}(x_{i+k} - \bar{x})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Let $\hat{\boldsymbol{\rho}}_x = (\hat{\rho}_x(1), \ldots, \hat{\rho}_x(\ell))$ and $\hat{\boldsymbol{\rho}}_y = (\hat{\rho}_y(1), \ldots, \hat{\rho}_y(\ell))$ be the estimated autocorrelation functions of $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$, respectively. The maximum lag ℓ should ideally be chosen such that $\hat{\rho}_x(k) \approx 0 \approx \hat{\rho}_y(k)$ for $k > \ell$. An autocorrelation-based dissimilarity measure can be defined as the Euclidean distance between the estimated autocorrelation functions:

Definition 4.2.7 – Autocorrelation-based distance (Díaz and Vilar [2010]).

$$d_{acf}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\sum_{i=1}^{\ell} \left(\hat{\rho}_x(i) - \hat{\rho}_y(i)\right)^2}. \tag{4.2.7}$$

In this way, time series with a similar linear temporal dependence structure are assigned a low distance. However, the classical autocorrelation only measures the strength of linear

temporal dependence but is not able to recognize a more general temporal dependence structure. The assumption of linearity often does not hold in practice as financial return series are usually not autocorrelated, but absolute or squared returns exhibit a strong autocorrelation. This fact has already been discussed in Chapter 2, using the example of the historic returns of the S&P 500.

4.2.2.2 Non-Linear Dependence Structure: Copulas

Copulas are widely used in modern finance as they represent a popular tool for constructing multivariate distributions and modeling a more complex, non-linear dependence structure between random variables. For the sake of simplicity, we are going to consider only bivariate versions of copulas in the following.

A copula can be seen as a function that links multiple marginal distribution functions to their joint distribution function, i.e., it characterizes the dependence structure between its individual components. It is defined as joint multivariate distribution function, whose univariate marginal distributions are all uniform:

$$C(u, v) = \mathbb{P}(U \leq u, V \leq v),$$

where $u, v \in [0, 1]$ (Ruppert and Matteson [2015]). Given two random variables X and Y with continuous distribution functions $F_X(x)$ and $F_Y(y)$, then, by applying the probability integral transform, each of $F_X(X)$ and $F_Y(Y)$ is uniformly distributed on the interval $[0, 1]$:

$$\mathbb{P}\big(F_X(X) \leq u\big) = \mathbb{P}\big(X \leq F_X^{-1}(u)\big) = F_X\big(F_X^{-1}(u)\big) = u.$$

According to Sklar's Theorem (Sklar [1959]), a unique copula function $C(u, v)$ exists, which connects the joint distribution function $F_{X,Y}(x, y)$ to $F_X(x)$ and $F_Y(y)$ via:

$$F_{X,Y}(x, y) = C\big(F_X(x), F_Y(y)\big), \quad \text{or equivalently} \quad C(u, v) = F_{X,Y}\big(F_X^{-1}(u), F_Y^{-1}(v)\big).$$

Thus, the joint distribution function $F_{X,Y}(x, y)$ can be decomposed into the copula C and the univariate marginal distribution functions $F_X(x)$ and $F_Y(y)$. The former contains all information about the dependencies among X and Y, while the latter take into account all information about both univariate marginal distributions (Ruppert and Matteson [2015]).

Zhang and An [2018] apply the framework of copulas to capture the dynamic pattern of a time series. Therefore, the copula function of $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+h}$ is denoted by $C_x^{(h)}(u, v)$ for a fixed lag $h \in \mathbb{N}$. In practice, this function is usually unknown, but it can be estimated in a nonparametric manner. For $1 \leq i \leq (n - h)$, define:

$$U_{x,i} = \frac{n}{n+1}\hat{F}_X(x_i), \quad V_{x,i} = \frac{n}{n+1}\hat{F}_X(x_{i+h}),$$

where $\hat{F}_X(x) = n^{-1}\sum_{i=1}^{n} I(x_i \leq x)$ denotes the empirical distribution function of $\boldsymbol{x}_t \in \mathbb{R}^n$.

Then, a nonparametric estimator for the copula function $C_x^{(h)}(u,v)$ is given by:

$$\hat{C}_x^{(h)}(u,v) = \frac{1}{n-h} \sum_{i=1}^{n-h} I(U_{x,i} \leq u) I(V_{x,i} \leq v).$$

Based on the constructed estimator, a copula-based dissimilarity measure between time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$ can be defined as:

Definition 4.2.8 – Copula-based distance (Zhang and An [2018]).

$$d_{cop}^{(h)}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\int \int_{[0,1]^2} \left(\hat{C}_x^{(h)}(u,v) - \hat{C}_y^{(h)}(u,v) \right)^2 du dv}, \tag{4.2.8}$$

where $h \in \mathbb{N}$ denotes a fixed lag. If time series are dependent in a higher order, the dissimilarity measure can be extended to the following weighted version:

$$d_{cop}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sum_{h=1}^{H} w_h \, d_{cop}^{(h)}(\boldsymbol{x}_t, \boldsymbol{y}_t), \tag{4.2.9}$$

where $w_h,\ h = 1, \ldots, H$ defines a weighting scheme such as geometric weights decaying with the order h, so that $w_h = p(1-p)^{h-1},\ h = 1, \ldots, H$ and $0 < p < 1$, or simply a uniform weighting scheme with $w_h = 1, \forall\, h$.

In practice, calculations in Equation 4.2.8 can be performed by using the following proposition:

$$d_{cop}^{(h)}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{L_{x,x}^{(h)} - 2 \cdot L_{x,y}^{(h)} + L_{y,y}^{(h)}},$$

where

$$L_{x,y}^{(h)} = \frac{1}{(n-h)(m-h)} \sum_{i=1}^{n-h} \sum_{j=1}^{m-h} \left(1 - \max(U_{x,i}, U_{y,j})\right) \left(1 - \max(V_{x,i}, V_{y,j})\right)$$

and equivalent for $L_{x,x}^{(h)}$ and $L_{y,y}^{(h)}$. The proof of this proposition is provided in Appendix B.

The above described copula-based dissimilarity measure has several advantages. First, it overcomes the limitations of existing approaches as it takes into account the general serial dependence structure of time series, which also includes linear structures. Thus, it can be used for linear but also for non-linear or heteroskedastic processes. Furthermore, the estimation procedure is nonparametric and does not require the specification of any model as it relies on the empirical estimator of the copula function. The consistency of the distance estimator can also be theoretically guaranteed (Zhang and An [2018]).

4.2.2.3 Quantile Autocovariances

Based on the concept of quantilograms (Linton and Whang [2007]), Lafuente-Rego and Vilar [2016] propose to use quantile autocovariances as an additional tool to discriminate time series according to their underlying dependence structure. They allow the identification of dependence features that classical covariance-based methods are not able to capture.

Given a stationary time series $\boldsymbol{X}_t \in \mathbb{R}^n$ and an arbitrary couple of probability levels $(\tau, \tau') \in [0,1]^2$, the quantile autocovariance function (QAF) is defined as the cross-covariance of the indicator functions $I(\boldsymbol{X}_t \leq q_\tau)$ and $I(\boldsymbol{X}_{t+h} \leq q_{\tau'})$:

$$\begin{aligned}\gamma_X^{(h)}(\tau, \tau') &= \operatorname{cov}\Big\{(I(\boldsymbol{X}_t \leq q_\tau), I(\boldsymbol{X}_{t+h} \leq q_{\tau'})\Big\} \\ &= \mathbb{P}(\boldsymbol{X}_t \leq q_\tau, \boldsymbol{X}_{t+h} \leq q_{\tau'}) - \underbrace{\mathbb{P}(\boldsymbol{X}_t \leq q_\tau)\,\mathbb{P}(\boldsymbol{X}_{t+h} \leq q_{\tau'})}_{=\tau\tau'},\end{aligned}$$

where $h \in \mathbb{N}$ denotes a fixed lag and q_τ is the τ-quantile of $\boldsymbol{X}_t$. The QAF of lag h captures the dynamic pattern of a time series and accounts for serial features related to the joint distribution of $\boldsymbol{X}_t$ and $\boldsymbol{X}_{t+h}$. It examines the joint variability of the events $(\boldsymbol{X}_t \leq q_\tau)$ and $(\boldsymbol{X}_{t+h} \leq q_{\tau'})$, and determines to what extent a part of the range of variation of $\boldsymbol{X}_t$ helps to predict whether the value of the series will fall below a certain quantile at a future point in time (Vilar et al. [2018]).

Replacing the theoretical quantiles by the corresponding empirical quantiles $\hat{q}_\tau$ and $\hat{q}_{\tau'}$ of a time series $\boldsymbol{x}_t \in \mathbb{R}^n$ yields a sample estimator of the QAF:

$$\hat{\gamma}_x^{(h)}(\tau, \tau') = \frac{1}{n-h} \sum_{i=1}^{n-h} I(x_i \leq \hat{q}_\tau) I(x_{i+h} \leq \hat{q}_{\tau'}) - \tau\tau' \tag{4.2.10}$$

for a fixed value of h. By specifying the number of lags to be considered and a common range of probability levels, the time series $\boldsymbol{x}_t$ can be characterized by the vector $\Gamma_x = \Big(\hat{\gamma}_x^{(h)}(\tau_i, \tau_j)\Big)$, $h = 1, \ldots, H$, $i, j = 1, \ldots, k$. Lafuente-Rego and Vilar [2016] define a quantile autocovariance-based dissimilarity measure between time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$ as the squared Euclidean distance between their estimated quantile autocovariances Γ_x and Γ_y:

Definition 4.2.9 – Quantile autocovariance-based distance (Vilar et al. [2018]).

$$d_{qaf}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \Big\|\Gamma_x - \Gamma_y\Big\|_2^2 = \sum_{h=1}^{H} \sum_{i=1}^{k} \sum_{j=1}^{k} \Big(\hat{\gamma}_x^{(h)}(\tau_i, \tau_j) - \hat{\gamma}_y^{(h)}(\tau_i, \tau_j)\Big)^2. \tag{4.2.11}$$

In order to apply the distance measure to a set of time series, the number of lags and the probability levels to be considered have to be specified first. Vilar et al. [2018] carried out an extensive simulation study covering various different types of processes. They concluded that using a sequence of equally spaced probability levels is most meaningful whereas the sequence $\boldsymbol{\tau} = (0.1, 0.2, 0.3, \ldots, 0.9)$ produced the best results. The number of lags to be considered depends, however, on the particular field of application.

In order to illustrate the great sensitivity of the QAF to capture complex dynamic features and to gain an insight into the shapes of the true QAF, we carried out a small simulation study similar to Vilar et al. [2018] and Lafuente-Rego et al. [2018]. Three different models were specified: a Gaussian white noise process and two GARCH(1,1) processes, both with Student-t innovations causing heavy tails. Parameters of the GARCH processes were fixed to $\omega_1 = 0.1$, $\alpha_{1,1} = 0.05$, $\beta_{1,1} = 0.90$, $\nu_1 = 5$ and to $\omega_2 = 0.1$, $\alpha_{2,1} = 0.15$, $\beta_{2,1} = 0.80$, $\nu_2 = 5$. Note that the GARCH processes are uncorrelated but not independent, while the white noise process is both. For each process, we simulated 5000 series of length 1000 and averaged

the corresponding sample quantile autocovariances over all replicates. Plots of the sample quantile autocovariances $\hat{\gamma}^{(1)}(\tau, \tau')$ were generated for three fixed values of $\tau \in \{0.1, 0.5, 0.9\}$, each over the range $\tau' = 0.05 \cdot i,\ i = 1, \ldots, 19$.

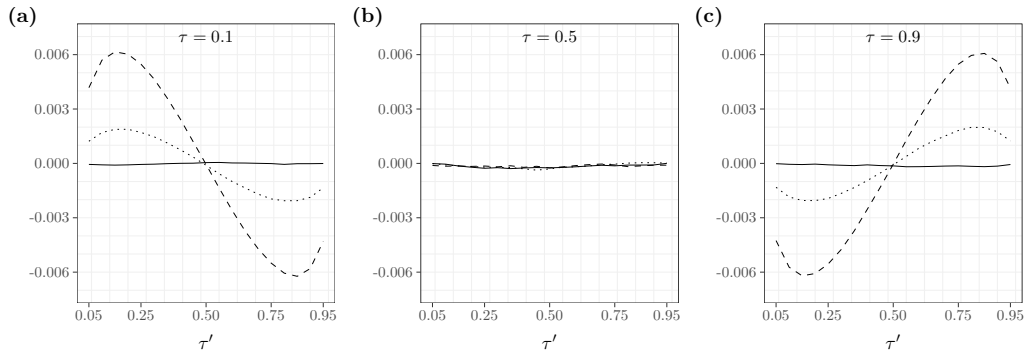

Figure 4.2.4 Sample quantile autocovariances $\hat{\gamma}^{(1)}(\tau, \tau')$ for (a) $\tau = 0.1$, (b) $\tau = 0.5$ and (c) $\tau = 0.9$ obtained from simulated realizations of a Gaussian white noise process (solid line) and two GARCH(1,1) processes with different parameters and Student-t innovations (dotted and dashed line).

Figure 4.2.4 shows that in each case, the graph of the white noise process (solid line) is flat due to independence. Likewise, the graphs of both GARCH-processes depicted in Figure 4.2.4 (b) are flat due to the symmetry of the GARCH model. This means if $(\boldsymbol{X}_t \leq q_{0.5})$, then $(\boldsymbol{X}_{t+1} \leq q_{0.5})$ and $(\boldsymbol{X}_{t+1} > q_{0.5})$ are events occurring with equal probability. By contrast, the graphs of both GARCH processes shown in Figure 4.2.4 (a) and (c) clearly differ from each other due to the different underlying dependence structure. In both cases, the heavy tails of the processes are clearly recognizable since small and large values in period t tend to persist until period $t + 1$ or even further (Vilar et al. [2018]).

The example shows that quantile autocovariances are highly capable to discriminate time series according to their underlying dependence structure. As stated by Vilar et al. [2018], the "QAF is well-defined even for processes with infinite moments and takes advantage from the local distributional properties inherent to the quantile methods, in particular showing a greater robustness against heavy tails, dependence in the extremes and changes in the conditional shapes (skewness, kurtosis)."

4.2.3 Model-Based Approaches

As the name already suggests, this class of approaches is based on parametric models fitted to raw time series data. First, one particular type of model must be specified which is then estimated for each time series involved in the clustering task. Similarity is evaluated in terms of features of the fitted models which mostly involve the estimated model parameters (D'Urso et al. [2016]).

Originally introduced by Engle [1982] and generalized by Bollerslev [1986], GARCH processes represent a popular class of models that are widely used in finance. Their key concept consists in modeling the conditional variance of a process. This allows to capture the main stylized facts of financial time series, as described in Chapter 2.

In general, time series can be modeled as GARCH(p, q) processes in the following way:

$$\begin{aligned} y_t &= \mu + \varepsilon_t \\ \varepsilon_t &= u_t \sqrt{h_t}, \end{aligned} \tag{4.2.12}$$

where u_t is an *i.i.d.* sequence of innovations with zero mean and unit variance. The conditional variance h_t is independent from u_t and is modeled by:

$$\mathrm{Var}(\varepsilon_t \,|\, \mathcal{J}_{t-1}) = \mathbb{E}[\varepsilon_t^2 \,|\, \mathcal{J}_{t-1}] = h_t = \omega + \sum_{i=1}^{p} \alpha_i \, \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \, h_{t-j}, \tag{4.2.13}$$

where $\mathcal{J}_{t-1}$ denotes the information set containing all information available at time $t-1$ and $\omega > 0,\ \alpha_i \geq 0,\ \beta_j \geq 0,\ i = 1, \ldots, p,\ j = 1, \ldots, q$ are coefficient constraints. Furthermore, the condition $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$ must hold, which guarantees the stationarity of the process and avoids infinite variance.

When dealing with heteroscedastic time series, analyzing and comparing the dynamics of variances is considered a fundamental aspect. A suitable dissimilarity measure for clustering of financial time series should, of course, take these dynamics into account. Depending on the exact definition of the dissimilarity measure, the clustering results in groups of time series sharing similar characteristics such as similar unconditional variances or similar dynamics, for instance (Otranto [2008]).

Otranto [2008] proposed a dissimilarity measure based on the time-varying part of the volatility. Volatility is inherently unobservable but under the GARCH model as defined in Equation 4.2.12 and 4.2.13, respectively, the squared disturbance ε_t^2 can be considered as a noisy but unbiased estimate for the conditional variance h_t. Therefore, Otranto [2008] justifies the clustering of financial time series based on the properties of their squared disturbances.

From Equation 4.2.13, the time series ε_t^2 can be represented as an ARMA(p^*, q) process:

$$\varepsilon_t^2 = \omega + \sum_{i=1}^{p^*} (\alpha_i + \beta_i) \, \varepsilon_{t-i}^2 - \sum_{j=1}^{q} \beta_j \, \eta_{t-j} + \eta_t \tag{4.2.14}$$

with $p^* = \max(p, q)$ and $\alpha_i = 0$ for $i > p$, if $p^* = q$, and $\beta_i = 0$ for $i > q$, if $p^* = p$. The term $\eta_t = \varepsilon_t^2 - h_t$ is a zero-mean error and uncorrelated with past information. Given Equation 4.2.14, the squared disturbance ε_t^2 can be represented as an AR(∞) process by recursive substitution:

$$\varepsilon_t^2 = \frac{\omega}{1 - \sum_{j=1}^{q} \beta_j} + \sum_{k=1}^{\infty} \pi_k \, \varepsilon_{t-k}^2 + \eta_t. \tag{4.2.15}$$

The corresponding AR coefficients π_k are stated in D'Urso et al. [2013]. They are given by:

$$\pi_k = \alpha_k + \beta_k + \sum_{j=1}^{\min(k,q)} \beta_j \, \pi_{k-j} \tag{4.2.16}$$

with $\pi_0 = -1,\ \alpha_k = 0$ for $k > p$, and $\beta_k = 0$ for $k > q$.

Piccolo [1990] proposed a metric for comparison of ARMA(p, q) processes. It measures the (dis)similarity of two invertible ARMA processes in terms of the Euclidean distance between the coefficients of their AR(∞) representations:

Definition 4.2.10 – AR distance for ARMA(p, q) processes (Piccolo [1990]).

$$d_{ar}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\sum_{k=1}^{\infty} \left(\pi_{x,k} - \pi_{y,k}\right)^2}, \tag{4.2.17}$$

where $\pi_{x,k}$ and $\pi_{y,k}$ indicate the coefficients at lag k of two AR(∞) processes generating time series $\boldsymbol{x}_t \in \mathbb{R}^n$ and $\boldsymbol{y}_t \in \mathbb{R}^m$, respectively.

For comparison of the time-varying part of volatilities, Otranto [2008] applies the distance measure to the AR(∞) structures with coefficients as given in Equation 4.2.16, resulting from the squared disturbances of GARCH(p, q) processes. Invertibility of the corresponding ARMA(p^*, q) processes is ensured by the constraints on the GARCH coefficients.

In practice, the GARCH$(1, 1)$ model is perhaps the most popular model adopted for financial time series. Deciding on this type of model, the dissimilarity measure stated in Equation 4.2.17 simplifies to:

Definition 4.2.11 – AR distance for GARCH(1,1) processes (D'Urso et al. [2013]).

$$d_{ar}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\frac{\alpha_x^2}{1 - \beta_x^2} - \frac{2\,\alpha_x \alpha_y}{1 - \beta_x \beta_y} + \frac{\alpha_y^2}{1 - \beta_y^2}}. \tag{4.2.18}$$

The derivation of the aforegoing expression is provided in Appendix C. If the distance between a pair of time series is zero, it does not mean their entire volatility structure is identical, but only the time-varying part of the volatility is equal (Otranto [2008]). However, it would be also desirable to incorporate additional information of the volatility structure besides the time-varying part. For this reason, Otranto [2008] derives an expression for the unconditional volatility and the minimum expected volatility at time $t + 1$, given all information up to time t. Based on the individual components, he developed a multi-level clustering algorithm. In each step, a classical Wald test is performed in order to test a particular hypothesis. The algorithm yields clusters of time series with equal unconditional volatility in the first step, equal unconditional and equal time-varying volatility in the second step, and finally a completely equal volatility structure in the third step.

The algorithm developed by Otranto [2008] represents a special procedure which is not designed to be used in combination with traditional clustering methods such as hierarchical or partitional clustering. For reasons of comparability and consistency, we are not going to consider it in our empirical application in Chapter 5.

The above described methods can naturally be applied only to the classical GARCH model but not to extensions such as the exponential GARCH model or the threshold GARCH model, for instance. However, it seems that there exists no dissimilarity measure for traditional clustering methods yet which takes into account the entire volatility structure of time series.

4.3 Clustering Methods and Algorithms

In literature, there exists a variety of different clustering methods such as hierarchical clustering, partitional clustering, density-based clustering, grid-based clustering and model-based clustering, among others. In addition, a distinction is made between the type of clustering: hard or soft. The former is also called *crisp clustering* and assigns each observation to exactly one cluster yielding non-overlapping clusters. The latter is better known as *fuzzy clustering* and allots each observation a probability or a degree of membership to each cluster (Anderberg [1973]).

Liao [2005] carried out a comprehensive survey and summarized various previous studies that address the clustering of time series in different application domains. He concluded that for time series clustering, the majority of general-purpose clustering methods can be applied without further ado but the choice of a suitable dissimilarity measure is clearly more important than the choice of the clustering method. For this reason, we are going to consider only (crisp) hierarchical and partitional clustering, as these seem to be the most commonly used methods in the literature of time series clustering.

4.3.1 Hierarchical Clustering

As the name already suggests, hierarchical clustering builds a hierarchy of clusters by successively aggregating or dividing the observations and their subsets. The resulting set of nested clusters can be organized as a tree which is referred to as a *dendrogram*. It can be cut at any given level to obtain a clustering, meaning that it is not necessary to specify the number of clusters K in advance. This feature is a main advantage over partitional clustering. Furthermore, hierarchical clustering operates on a precomputed dissimilarity matrix, meaning that any arbitrary dissimilarity measure can be used to calculate pairwise distances between the observations. This makes hierarchical clustering highly interesting for our purpose.

There exist two different types of hierarchical clustering: agglomerative and divisive. We are going to discuss both types below, but in practice, agglomerative clustering is much more popular than divisive clustering (Aggarwal and Reddy [2018]).

4.3.1.1 Agglomerative Clustering

Agglomerative clustering, also known as agglomerative nesting (AGNES), is a bottom-up approach which starts by taking all observations as individual clusters. Then, the closest pair of clusters is merged at each step. This process is carried on until only one single cluster remains, i.e., a cluster that contains all observations. The standard algorithm for agglomerative clustering can be found in Algorithm 1.

In each iteration of the algorithm, the pair of clusters with minimum distance is merged. Then, the distance between the newly formed cluster and all other clusters must be calculated. This can be done with the help of the Lance-Williams dissimilarity update formula, whereby

the distance between cluster $C_{j\cup k}$ and C_l is given by:

$$D(j \cup k, l) = \alpha_j \cdot D(j,l) + \alpha_k \cdot D(k,l) + \beta \cdot D(j,k) + \gamma \cdot |D(j,l) - D(k,l)|, \tag{4.3.1}$$

where $D(\cdot\,,\cdot)$ is the distance between two clusters and α_j, α_k, β and γ are parameters that uniquely determine the method (Aggarwal and Reddy [2018]). In the following, we are going to introduce three commonly used methods for agglomerative hierarchical clustering.

```
# Algorithm 1 – Agglomerative Clustering (Aggarwal and Reddy [2018]).
Compute the entire dissimilarity matrix between all N observations
Initialize the individual observations as clusters C_i with n_i = 1,  i = 1, ..., N
repeat
  Merge clusters as C_{j∪k} = C_j ∪ C_k and set n_{j∪k} = n_j + n_k
  Remove the rows and columns corresponding to C_j and C_k from the dissimilarity
  matrix
  Add a new row and column containing the distance between C_{j∪k} and all other
  clusters
until only one cluster remains containing all observations
return set of nested clusters
```

Single linkage

Single linkage, also known as nearest neighbours method, is one of the simplest hierarchical clustering methods. The distance between merged cluster $C_{j\cup k}$ and cluster C_l is defined as the minimum distance between their members:

$$\begin{aligned} D(C_j \cup C_k, C_l) &= 0.5 \cdot D(C_j, C_l) + 0.5 \cdot D(C_k, C_l) - 0.5 \cdot |D(C_j, C_l) - D(C_k, C_l)| \\ &= \min\{D(C_j, C_l),\, D(C_k, C_l)\} = \min_{\substack{x \in C_j \cup C_k \\ y \in C_l}} d(x, y), \end{aligned}$$

where $d(\cdot\,,\cdot)$ is the dissimilarity measure by which the dissimilarity matrix is computed. This yields the Lance-Williams dissimilarity update formula parameters $\alpha_j = 0.5$, $\alpha_k = 0.5$, $\beta = 0$ and $\gamma = -0.5$ (Gan et al. [2007]). Single linkage is able to find arbitrary shaped clusters but it is highly sensitive to outliers and noise in data (Aggarwal and Reddy [2018]).

Complete linkage

Unlike single linkage, the complete linkage method defines the distance between merged cluster $C_{j\cup k}$ and cluster C_l as the maximum distance between their members:

$$\begin{aligned} D(C_j \cup C_k, C_l) &= 0.5 \cdot D(C_j, C_l) + 0.5 \cdot D(C_k, C_l) + 0.5 \cdot |D(C_j, C_l) - D(C_k, C_l)| \\ &= \max\{D(C_j, C_l),\, D(C_k, C_l)\} = \max_{\substack{x \in C_j \cup C_k \\ y \in C_l}} d(x, y), \end{aligned}$$

which yields the Lance-Williams dissimilarity update formula parameters $\alpha_j = 0.5$, $\alpha_k = 0.5$, $\beta = 0$ and $\gamma = 0.5$ (Gan et al. [2007]). Complete linkage tends to break large clusters and generally obtains clusters of compact shape. It is less sensitive to noise and outliers compared to single linkage (Aggarwal and Reddy [2018]).

Ward's criterion

Ward's criterion differs from the previous methods in that it minimizes a specific loss of information associated with the merge at each step. This loss is usually quantified in terms of the sum of squared errors (SSE). Therefore, Ward's criterion is often referred to as *minimum variance method*.

If the squared Euclidean distance is used to compute the dissimilarity matrix, then the following formula can be used to calculate the distance between merged cluster $C_{j \cup k}$ and cluster C_l:

$$D(C_j \cup C_k, C_l) = \frac{n_j + n_l}{\sum_{jkl}} \cdot D(C_j, C_l) + \frac{n_k + n_l}{\sum_{jkl}} \cdot D(C_k, C_l) - \frac{n_l}{\sum_{jkl}} \cdot D(C_j, C_k),$$

where $\sum_{jkl} = n_j + n_k + n_l$. The corresponding Lance-Williams dissimilarity update formula parameters for α_j, α_k and β are given by the three fractions in the aforegoing equation and γ takes a value of zero. A natural advantage of Ward's criterion is the minimization of the total within-cluster variance.

In addition to the methods described above, average linkage, centroid linkage, the median method and McQuitty's method exists, among others.

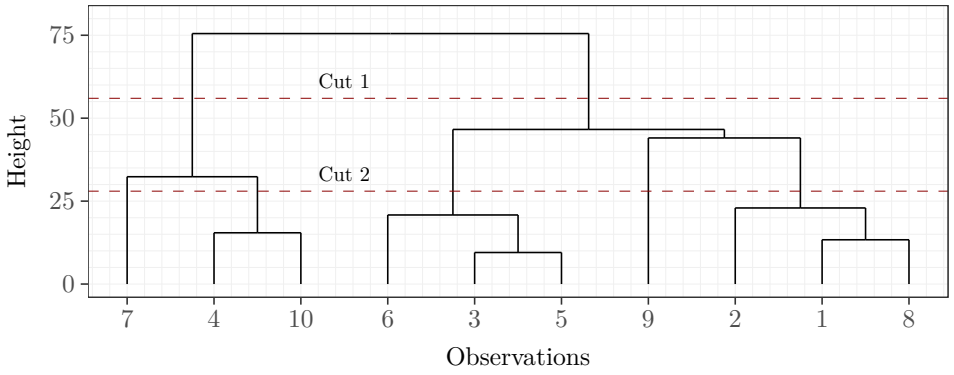

Figure 4.3.1 Exemplary dendrogram for a hierarchical clustering with single linkage and ten observations of the `R` data set *USArrests*. Two exemplary cuts are added at $K = 2$ and $K = 5$.

The resulting dendrogram for a hierarchical clustering with single linkage and ten observations is depicted in Figure 4.3.1. In addition, two exemplary cuts are added. The first cut results in two clusters –$\{1, 2, 3, 5, 6, 8, 9\}$ and $\{4, 7, 10\}$– and the second one results in five clusters: $\{1, 2, 8\}, \{3, 5, 6\}, \{4, 10\}, \{7\}$ and $\{9\}$.

4.3.1.2 Divisive Clustering

In contrast to agglomerative clustering, divisive clustering is a top-down approach which starts with one large cluster containing all observations. Then, the cluster is recursively split until N different clusters remain, each containing only one observation. Since there are $2^{N-1} - 1$ possibilities to perform a split in the first step, an exact algorithm for divisive clustering becomes quickly very time-consuming as the number of observations increases. For this reason, various heuristics such as divisive analysis (DIANA) or bisecting K-means were developed (Hennig et al. [2015]).

The process of divisive clustering can be terminated whenever a specific number of clusters is reached. This can be of advantage, if only the fundamental structure of the data is of interest. However, since divisive clustering is hardly used in practice including in the field of time series clustering, we will not further consider it.

4.3.2 Partitional Clustering

Partitional clustering is a method that divides data into K different clusters, whereas the number of clusters K has to be specified in advance. It tries to find the clustering that optimizes a previously defined clustering optimization criterion. Finding a global optimum is usually computationally infeasible since this would require trying all potential clusterings. As the number of observations increases, the number of clusterings becomes quickly prohibitively large. For this reason, partitional clustering algorithms often start with a random initialized partition and proceed by locally improving the optimization criterion. Thus, the majority of partitional clustering algorithms are greedy-like algorithms as they guarantee convergence to a local optimum, but the detection of the global optimum is, however, known to be NP-hard (Sammut and Webb [2017]).

In the following, we are going to discuss two commonly used partitional clustering algorithms: K-means and K-medoids. Both algorithms build the clusters around the means and medoids of observations, respectively. Furthermore, we are going to address a variation of K-medoids, which is constructed especially for handling large data sets.

4.3.2.1 K-Means

Currently, K-means is the most popular partitional clustering algorithm. It uses the SSE as clustering optimization criterion to be minimized. Data must be provided as a set of numerical vectors, meaning that a precomputed dissimilarity matrix based on an arbitrary dissimilarity measure can not be processed.

 # Algorithm 2 – K-Means (Han et al. [2011]).
Initialize cluster centroids $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
repeat
  for i in $1 : N$ **do**
    $c_i \coloneqq \arg\min_l \; d(\boldsymbol{x}_i, \boldsymbol{\mu}_l)$
  end for
  for j in $1 : K$ **do**
    $\boldsymbol{\mu}_j \coloneqq n_j^{-1} \sum_{i \in C_j} \boldsymbol{x}_i$
  end for
until stopping condition is fulfilled
return $c_1, \ldots, c_N$ and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$

The basic K-means algorithm starts by choosing K representative observations as initial centroids. Then, all remaining observations are assigned to their closest centroid. The assignment is based upon a specific proximity measure which is by default the (squared) Euclidean distance. Finally, the centroid of each cluster is updated by the average value of the corresponding observations. The algorithm iteratively repeats these steps until a

particular stopping condition is fulfilled (Aggarwal and Reddy [2018]). Algorithm 2 provides an outline of the basic K-means algorithm.

Different approaches for initialization of the cluster centers are available. The simplest and most widely used method consists in randomly selecting K different observations from data. K-means++ is an alternative algorithm similar to K-means which carefully selects the initial centroids using a probabilistic approach. The first centroid is again randomly selected from data. Then, each additional cluster center is selected from the remaining observations with probability proportional to the squared distance from each observation's closest centroid. This is carried on until all K centroids are initialized. The further procedure of the algorithm remains unchanged (Arthur and Vassilvitskii [2007]). See Celebi et al. [2013] for a detailed overview and comparison of various other initialization methods.

The main part of the basic K-means algorithm consists of two steps. First, each observation is assigned to the cluster with the closest centroid. Therefore, all pairwise distances between observations and cluster centroids must be calculated. This is usually done by using the (squared) Euclidean distance. Other variants using more complex proximity or dissimilarity measures such as the ones described in Chapter 4.2 would be also conceivable.

In the second step, the cluster centroids are updated by the average value of the corresponding observations. If more complex proximity measures were used in the cluster assignment step, then the calculation of the average value might be problematic in some cases. Petitjean et al. [2011] points out that the employment of DTW does not produce meaningful results since for this dissimilarity measure the triangle inequality does not hold. Calculations of the average value can instead be performed by using a technique called DTW barycenter averaging. See Petitjean et al. [2011], pp. 682-687 for further details. However, other variants of K-means that are not using the (squared) Euclidean distance are hardly used in practice and mostly not available so far.

The algorithm iteratively repeats these two steps until convergence, i.e., either the centroids no longer change or only a small percentage of observations changes their cluster membership. Often a maximum number of iteration is also specified. When the algorithm terminates, the cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$ and the final cluster assignments $c_1, \ldots, c_N$ are usually returned, as they represent the most important information (Aggarwal and Reddy [2018]).

4.3.2.2 K-Medoids

Since the framework of the K-means algorithm is very simple, it can be modified without any further ado. This makes it possible to construct different variants such as K-medoids, for instance. K-medoids is a partitional clustering algorithm which also divides the data into K different clusters. Instead of the SSE, it uses the absolute error as clustering optimization criterion to be minimized. K-medoids chooses actual observations as cluster centers which are referred to as *medoids*. They can be seen as cluster representatives as they minimize the average dissimilarity to all other observations in the same cluster.

Partitioning Around Medoids (PAM) is the most popular algorithm in order to find a good clustering using medoids, with respect to the absolute error. It is proposed by Kaufman and Rousseeuw [1987] and consists of two main steps: the build and the swap phase. In

the build phase, K observations are randomly selected as initial cluster medoids. Then, the swap phase further optimizes the clustering by swapping medoids and non-medoids until convergence. Also in this case, finding the global optimum is known to be NP-hard. When the algorithm terminates, the final cluster assignments $c_1, \ldots, c_N$ and the cluster medoids $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K$ are usually returned. An outline of PAM is given in Algorithm 3.

```
# Algorithm 3 – PAM (Kaufman and Rousseeuw [1987]).
Initialize cluster medoids m_1, ..., m_K
repeat
  Assign the remaining observations x_1, ..., x_{N-K} to their closest medoid
  for i in 1 : K do
    for j in 1 : (N − K) do
      Compute the total cost C_{i,j} of swapping m_i and x_j
      if C_{i,j} < 0 then
        Swap m_i and x_j
      end if
    end for
  end for
until the set of medoids does not longer change
return c_1, ..., c_N and m_1, ..., m_K
```

PAM has two main advantages over competing partitional clustering algorithms such as K-means. First, it generally operates on the dissimilarity matrix. If data is provided as a set of numerical vectors, then PAM computes the dissimilarity matrix before starting its build phase. This feature makes it highly interesting for our purpose since we can use any arbitrary dissimilarity measure to calculate pairwise distances between the observations and then pass the entire dissimilarity matrix to the algorithm.

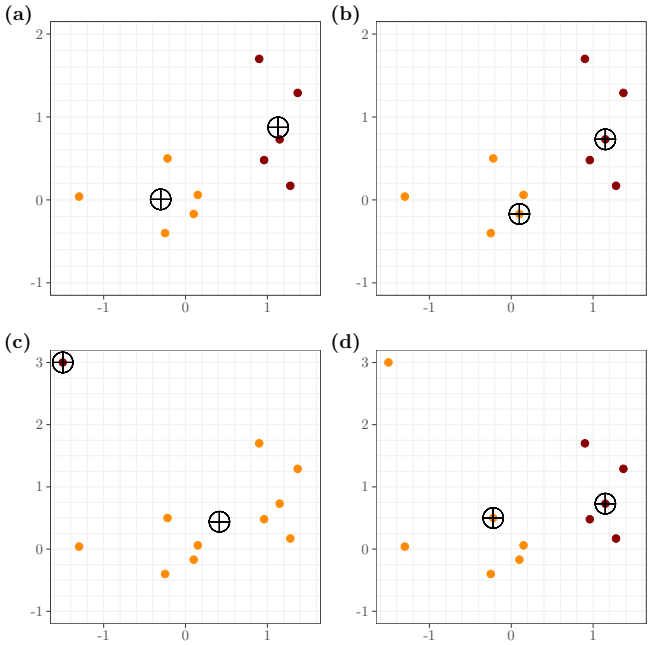

Figure 4.3.2 Influence of outliers on the clustering outcome: (a) K-means and (b) PAM clustering without any outliers, (c) K-means and (d) PAM clustering with one present outlier.

Second, it is more robust against outliers and noise in data since the medoid is naturally less sensitive than the mean (Hennig et al. [2015]). This property is visualized in Figure 4.3.2. It shows the clustering of ten artifically generated data points and $K = 2$ for different constellations. The cluster assignment is color-coded in red and yellow and the corresponding cluster centroids and medoids are marked as crosses. In part (a) and (b) no outlier exists. As expected, K-means and PAM result in the same clustering in this case. In part (c) and (d) the data contains an outlier. PAM is hardly affected by the outlier and still yields the same result together with a different medoid. K-means, by contrast, allots the outlier a separate cluster and forms a second cluster consisting of the remaining ten observations.

In general, PAM comes along with a higher time complexity compared to K-means. Applying it on a great number of observations can become very time-consuming. Therefore, a sampling-based method exists which is able to deal with large data sets (Aggarwal and Reddy [2018]).

4.3.2.3 Clustering LARge Applications

In the previous section we concluded that PAM is not able to deal efficiently with large data sets. To address this problem, Kaufman and Rousseeuw [1990] constructed a sampling-based method called Clustering LARge Applications (CLARA). This algorithm randomly draws B samples of size z from data and applies PAM to each sample. In each case, the remaining $N - z$ observations are assigned to their closest medoid which yields a clustering of the entire data set. This procedure results in B different clusterings, whereas the one with the minimal absolute error is returned. An outline of CLARA can be found in Algorithm 4.

 # Algorithm 4 – CLARA (Kaufman and Rousseeuw [1990]).
1 Set a value for z and B
2 **for** i in $1 : B$ **do**
3 Randomly draw z observations from data
4 Apply PAM to the sample
5 Assign the remaining $N - z$ observations to their closest medoid
6 Calculate the total absolute error E_i
7 **if** E_i is the lowest found so far **then**
8 Update the best clustering to the current clustering
9 **end if**
10 **end for**
11 **return** $c_1, \ldots, c_N$ and $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_K$ of the best clustering

In contrast to PAM, CLARA does not require a precomputed dissimilarity matrix of the entire data set. In each step, it calculates the dissimilarities only between observations of the sample, which is of comparatively small size. This procedure leads to a linear time complexity of the algorithm (Hesabi et al. [2015]).

CLARA shares the robustness property of PAM, while being able to handle large data sets. However, the clustering obtained by CLARA naturally depends on the sampling which yields a trade-off between efficiency and quality of the clustering. It is also not guaranteed that CLARA finds the optimal medoids during the sample process. In practice, choosing the

parameter combination $B = 5$ and $z = 40 + 2K$ turned out to produce satisfactory results (Sagvekar et al. [2013]).

4.4 Cluster Validation

As we have seen in the previous section, the number of clusters K has to be set manually for both, hierarchical and partitional clustering methods. In practice, the value of K is usually unknown and therefore, it must be either estimated or specified based on prior knowledge. In order to assess the goodness of a clustering, many different cluster validation measures are proposed in literature. However, despite the great effort spent on this problem, there still exists no cluster validation measure which can be generally considered as the best (Aggarwal and Reddy [2018]).

Cluster validation measures can be divided into two groups: external and internal measures. The former use additional information such as externally provided class labels, meaning that the number of clusters is known in advance. They evaluate to what extent the clustering, obtained by a particular clustering algorithm, matches the external structure (Hennig et al. [2015]). However, as stated by Moulavi et al. [2014], "external measures do not have practical applicability, since, according to its definition, clustering is an unsupervised task, with no ground truth solution available a priori." Internal measures, by contrast, evaluate the goodness of a clustering without the use of any external information.

Of course, our data set does not contain any external information as it only consists of stock prices together with a unique identifier. Thus, only internal measures are relevant for the evaluation of our clustering. In the following, we are going to address three prominent examples of internal validation measures which have turned out to appropriately estimate the number of clusters. For an extensive overview and comparison of external cluster validation measures, we refer to Aggarwal and Reddy [2018], Chapter 23.

Silhouette index

Just as many other cluster validation measures, the Silhouette index (SI) proposed by Kaufman and Rousseeuw [1990] is based on a compromise between intra-cluster homogeneity and inter-cluster separation.

It determines the so-called *silhouette width* for each observation $\boldsymbol{x}_i,\ i = 1, \ldots, N$, consisting of two components. First, the average dissimilarity a_i between $\boldsymbol{x}_i$ and all other observations in the same cluster is calculated:

$$a_i = \frac{1}{n_k - 1} \sum_{j \in C_k} d(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

The smaller the resulting value of a_i, the more compact is the cluster to which $\boldsymbol{x}_i$ belongs to. Next, only those clusters are considered which do not include $\boldsymbol{x}_i$. For each of them, the average dissimilarity between $\boldsymbol{x}_i$ and all observations in the corresponding cluster is calculated. The minimum of these averages yields the value of b_i:

$$b_i = \min_{k \neq l} \frac{1}{n_l} \sum_{j \in C_l} d(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

This value reflects the degree to which $\boldsymbol{x}_i$ is separated from the observations of the neighboring cluster. Given the values of a_i and b_i, the silhouette width of observation $\boldsymbol{x}_i$ is defined as:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}.$$

The denominator of the fraction normalizes the value of s_i to the range $[-1, 1]$. Positive (negative) values of s_i indicate that observation $\boldsymbol{x}_i$ has most likely been assigned to the correct (wrong) cluster. As exemplarily shown in Figure 4.4.1, the resulting set of silhouettes can be graphically illustrated in form of a bar chart, also widely known as *silhouette plot*. The individual clusters are usually coded in different colors. In this case, the clustering algorithm seems to have produced a decent clustering.

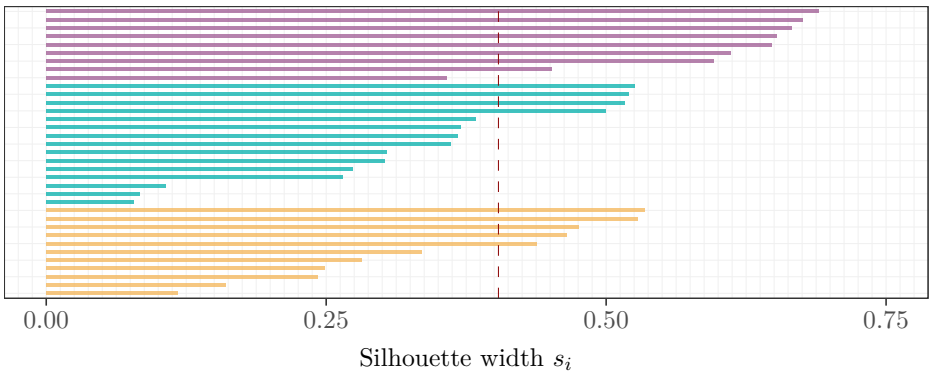

Figure 4.4.1 Silhouette plot resulting from a K-means clustering with $K = 3$ and 35 observations of the R data set *iris*. In this case, the SI takes a value of about 0.40 (red dashed line).

In order to assess the goodness of a clustering, the SI is calculated by taking the average value of all individual silhouette widths $s_i,\ i = 1, \ldots, N$:

$$SI_K = \frac{1}{N} \sum_{i=1}^{N} s_i. \tag{4.4.1}$$

As with other internal cluster validation measures, the SI is calculated for a range of different values of K. The optimal number of clusters is the value of K for which the SI is maximized (Hennig et al. [2015]).

Dunn index

Just as the SI, the Dunn index (DI) proposed by Dunn [1974] is an internal validation measure to assess a clustering based on an arbitrary dissimilarity measure. For a fixed value of K, it sets the between-cluster separation in relation to the within-cluster compactness:

$$DI_K = \frac{d_{min}}{d_{max}}. \tag{4.4.2}$$

The numerator d_{min} is defined as the minimal distance between observations of different clusters:

$$d_{min} = \min_{k \neq l} D(C_k, C_l),$$

where $D(C_k, C_l)$ is the distance between clusters C_k and C_l, measured in terms of the minimal distance between their members:

$$D(C_k, C_l) = \min_{\substack{\boldsymbol{x} \in C_k \\ \boldsymbol{y} \in C_l}} d(\boldsymbol{x}, \boldsymbol{y}).$$

The denominator d_{max} is, by contrast, defined as the largest diameter among all K clusters:

$$d_{max} = \max_{1 \leq k \leq K} DM(C_k),$$

where $DM(C_k)$ is the largest diameter of cluster C_k, i.e., the largest distance separating two distinct observations:

$$DM(C_k) = \max_{\substack{\boldsymbol{x}, \boldsymbol{y} \in C_k \\ \boldsymbol{x} \neq \boldsymbol{y}}} d(\boldsymbol{x}, \boldsymbol{y}).$$

Bringing together both components, d_{min} and d_{max}, the DI seeks to find a clustering consisting of well separated homogeneous clusters. This can be achieved by maximizing the value of DI.

Due to the required minimization and maximization operation, the index is very sensitive to changes in the cluster structure induced by outliers or noise in data, for instance. However, also other definitions for $D(C_k, C_l)$ and $DM(C_k)$ can be used in order to define a more general and robust version of the DI (Hennig et al. [2015]).

Calinski-Harabasz index

Milligan and Cooper [1985] carried out an extensive study, which compared thirty different cluster validation measures. Based on MC simulations, the performance of each measure was evaluated on a range of artificial data sets, whereas the ground truth was available for each of them.

The most successful method to correctly determine the number of clusters was the Calinski-Harabasz (CH) index originally proposed by Calinski and Harabasz [1974]. It is defined as the ratio of between-cluster and within-cluster variation and can be calculated for every value of K as follows:

$$CH_K = \frac{N - K}{K - 1} \frac{tr(\boldsymbol{BCSS}_K)}{tr(\boldsymbol{WCSS}_K)}, \tag{4.4.3}$$

where N is the total number of observations and K is the number of clusters formed. The prefactor $(N - K)/(K - 1)$ can be motivated by the consideration of the degrees of freedom as in the analysis of variance (ANOVA). The between-cluster sum of squares (BCSS) is defined as the weighted sum of differences between the cluster centroids $\bar{\boldsymbol{x}}_k,\ k = 1, \ldots, K$ and the overall centroid $\bar{\boldsymbol{x}}$ of the data set:

$$\boldsymbol{BCSS}_K = \sum_{k=1}^{K} n_k \, (\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}})(\bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}})^\top,$$

whereas the weights are given by the cardinalities n_k of clusters $C_k,\ k = 1, \ldots, K$. The within-cluster sum of squares (WCSS) is, by contrast, defined as the sum of squared deviations

from the individual observations and their corresponding cluster centroid:

$$\boldsymbol{WCSS}_K = \sum_{k=1}^{K} \sum_{i \in C_k} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_k)^\top.$$

Note that the CH index is not defined for $K = 1$. Large index values imply a good clustering, meaning that the formed clusters are well separated and compact at the same time. For a fixed value of K, maximizing the between-cluster variation $tr(\boldsymbol{BCSS}_K)$ is equivalent to minimizing the within-cluster variation $tr(\boldsymbol{WCSS}_K)$. Of course, the CH index can also be used for comparison of different clusterings with the same value of K (Hennig et al. [2015]).

Originally, the CH index is limited to using the Euclidean distance, which connects it in some way to the K-means clustering. The generalisation proposed by Hennig and Liao [2013] can, however, be used to evaluate a clustering based on an arbitrary dissimilarity measure. It redefines the within-cluster and the between-cluster variation as follows:

$$WCSS_K = \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i \in C_k} \sum_{j \in C_k} d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2,$$

and

$$BCSS_K = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} d(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 - WCSS_K.$$

Again, setting both components in relation to each other yields the CH index:

$$CH_K = \frac{N-K}{K-1} \frac{BCSS_K}{WCSS_K}. \tag{4.4.4}$$

Having discussed the theory of cointegration and the concept of time series clustering together with various dissimilarity measures and techniques related to clustering, we can finally apply these methods in an empirical application.

Chapter 5

Empirical Application: Pairs Trading

5.1 History and General Idea

The history of pairs trading (PT) reaches back to the mid 1980s. Nunzio Tartaglia, a quantitative analyst who worked at Morgan Stanley, was given the opportunity to form a group of researchers from various different fields. Their aim was the construction of quantitative arbitrage strategies based on the analysis of data using solely statistical methods. The strategies were automated to that point that all trades could be executed automatically. The developed concept was considered as somehow groundbreaking at Wall Street, since the selection of stocks was based almost only on fundamental analyses up to this point.

PT is a market neutral strategy in the sense that achieved returns are uncorrelated with overall market returns. This kind of strategy can offer potential profits regardless of whether the market goes up or down, and returns are typically generated with lower volatility.

Statistical arbitrage PT is based on the concept of relative pricing. The underlying premise is that stocks with similar characteristics should be priced nearly identical. The spread reflects the degree of the mutual mispricing which is assumed to be of a temporary nature. Therefore, it is expected that the mispricing corrects itself.

The basic methodology of PT is surprisingly simple. A pairs trade is usually performed in two subsequent stages. First, two stocks are selected, whose price paths moved together within a fixed period of time. This stage is usually referred to as formation period. Having identified a suitable pair, a trading trigger is specified. Then, both stocks are traded in a subsequent trading period. In case the price spread exceeds the trigger, the higher-priced stock is sold short and the lower-priced stock is bought in return. Both positions are unwound on convergence, i.e., at the next crossing of the price paths. Usually, not only one single pair of stocks but a portfolio of pairs is traded (Vidyamurthy [2004]).

5.2 Literature Review

First circulated as a working paper in 1999 and officially published in 2006, the academic paper of Gatev et al. [2006] is one of the earliest and by now, most cited, studies about PT. In their paper, a simple PT strategy is backtested on a large data set of the Center

for Research in Security Prices (CRSP) consisting of daily closing prices of the U.S. equity market for the period from July 1962 to December 2002. Based on a formation period of 12 months, pairs of stocks are identified that minimize the sum of squared deviations (SSD) between normalized prices. This approach is often referred to as *distance method*. In a subsequent trading period of 6 months, several self-financing portfolios of different sizes are traded, resulting in significant excess returns of up to 11% per annum before transaction costs. The profitability of the strategy can not be explained by reversal or momentum profits as in Jegadeesh [1990] or Jegadeesh and Titman [1993], respectively. Instead, it is attributed to a so far unknown risk factor.

Later, the results of Gatev et al. [2006] were independently replicated and verified by several other authors, most notably by Do and Faff [2010, 2012]. Likewise, they use the data set of the CRSP, but extend the sample period by seven years until 2009. A declining profitability of PT is documented, especially since 2002. The decrease is not only associated to an increased competition among hedge funds but it is mainly attributed to an increasing proportion of pairs that diverge but never converge again. Notably, PT performs strongly especially during phases of market turbulence, including the global financial crisis of 2007-09. However, after deduction of transaction costs, PT becomes mostly unprofitable. These results comply with Jacobs and Weber [2015], who report an immensely time-varying profitability of PT, which, however, constantly declined over the years.

Besides using the distance method, Rad et al. [2016] further extend the cointegration approach provided by Vidyamurthy [2004] and additionally construct a PT strategy involving copulas. Backtesting of these three strategies is again performed on the data set of the CRSP for the period from July 1962 to December 2014. The results are similar to the previous studies in the sense that the distance method and the cointegration approach yield the highest average excess returns of about 10% per annum before transactions costs. Again, a declining profitability together with a comparatively strong performance in highly volatile market conditions is found. Notably, the copula method performs worst, but its performance is relatively stable over time.

For further studies on PT, we refer to Elliott et al. [2005], Zeng and Lee [2014], Liu et al. [2017], and Clegg and Krauss [2018], among others.

5.3 Backtesting on the MSCI Europe Index

5.3.1 Data Set

The MSCI Europe Index serves as asset universe in order to backtest our PT strategies. It reflects the performance of large- and mid-cap equities across 15 developed markets in Europe including Austria, Belgium, Denmark, Finland, France, Germany, Ireland, Italy, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom. In September 2018 for instance, the index consisted of 443 constituents, comprising approximately 85% of the free-float adjusted market capitalization in each developed market (MSCI [2018]).

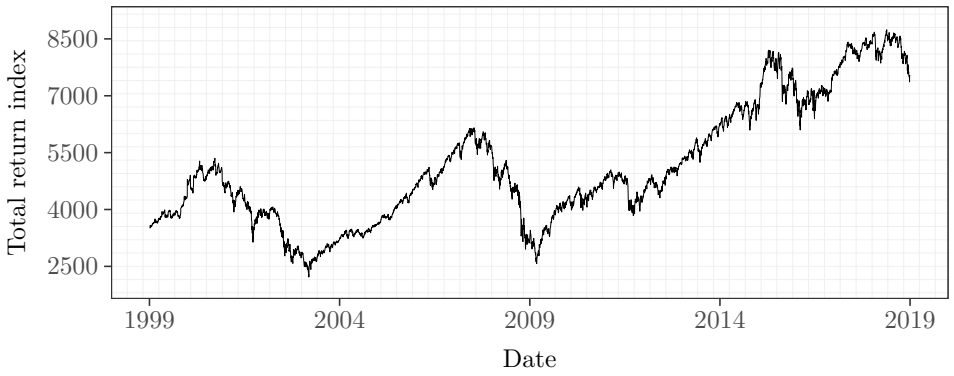

Figure 5.3.1 Total return index of the MSCI Europe Index for the period from January 04, 1999 to December 31, 2018.

Our data set consists of daily observations for the period from January 04, 1999 to December 31, 2018. This results in a sample period of 5157 days (239 months) for a total of 1227 stocks. Note that every single stock is included that has ever been an index constituent during the whole period available. On average, the MSCI Europe Index contains around 450 equities but its composition slightly changes over time.

Instead of raw price series, total return indices (in €) are used. They include all dividends distributed and account for any further corporate actions and stock splits, making them most suitable for return calculations. Figure 5.3.1 shows the total return index (in €) of the MSCI Europe Index for the past 20 years.

5.3.2 Methodology

Following Gatev et al. [2006], we choose a 12-month formation period together with a 6-month trading period and backtest our strategies by using a 1-month rolling window. This leads to six overlapping "portfolios". Each of them is linked to a trading period starting in a different month. As we use seven distance measures in combination with two clustering algorithms, this results in 14 different trading strategies.

We implement the whole framework in the statistical programming language `R` (R Core Team [2018]) and built an `R` package called *saft4pt*. See Appendix A for further details.

5.3.2.1 Pairs Formation Method

Within each formation period, we restrict our sample solely to stocks that are listed as constituents of the MSCI Europe Index. This way, illiquid stocks should be removed from data, as constituents of this index are selected based upon liquidity and size constraints.

The data set naturally contains missing observations for a number of trading days and stocks. In practice, there exist various ways of dealing with missing values. We combine two methods and proceed in the following way: Within each formation period, we remove a date, if more than 75% of the stocks contain a missing value on that day. Furthermore, we remove a single stock, if it contains more than five missing values in succession, which corresponds

to at least one week of missing data.[2] On average, two dates and 18 stocks are removed in each formation period. For the remaining missing values, we use a simple but quite popular technique called *last observation carried forward* (LOCF). This method imputes missing values by simply using the last available observation instead. Having dealt with missing values, this yields an average sample of 256 days per formation period, including 462 stocks on average.

For each stock, a cumulative total return index is constructed, taking a value of 1 € on the first day of each formation period. This can be done by either adding a value of 1 to the (discrete) returns and calculating the cumulative product or by normalizing each price series with respect to its first value:

$$p_{i,t}^{norm} = \prod_{j=1}^{t}(1 + r_{i,j}) = \frac{p_{i,t}}{p_{i,1}},$$

where $p_{i,t}$ denotes the price of stock i at time $t \geq 1$ and $r_{i,t}$ is the corresponding (discrete) return with $r_{i,1} = 0$. Normalization of prices is essential as PT is based on relative pricing (Gatev et al. [2006]). Henceforth, the normalized price or the cumulative total return index is simply referred to as "price" of a stock.

In the next step, time series are clustered. For this purpose, we use seven different distance measures (data type used indicated in parentheses), namely the Euclidean distance (prices), the DTW distance (prices), the Pearson distance (prices), the cross-correlation distance (prices), the autocorrelation-based distance (absolute returns), the copula-based distance (returns), and the quantile autocovariance-based distance (returns).

We apply DTW without imposing any constraints in order to obtain the optimal alignment between time series. For the cross-correlation distance, choosing a maximum time shift of ± 5 lags appears to be adequate. For the three "feature-based distance measures", the number of lags to be considered is set to 50. This number seems to be sufficient in order to capture all essential information. As suggested by Vilar et al. [2018], we use the sequence of probability levels $\boldsymbol{\tau} = (0.1, 0.2, 0.3, \ldots, 0.9)$ for the quantile autocovariance-based distance. For reasons of comparability, we do not consider any model-based distances within our framework. Obviously, a formation period of 12 months does not contain enough observations to consistently estimate the parameters of a GARCH model.

For each formation period, we calculate the entire dissimilarity matrix using the seven above-mentioned distance measures. As some of them are quite sensitive to outliers, we decide to remove a fixed fraction of observations from the dissimilarity matrix prior to clustering. Removing 5% of the observations with the comparatively highest distance[3] turns out to produce satisfactory results. This method is referred to as *trimmed approach* and is also applied by D'Urso et al. [2017] and Lafuente-Rego et al. [2018] in the context of time series clustering.

[2] Choosing a value greater than five results on average in only slightly more than 18 stocks removed in each formation period. Thus, the majority of stocks containing five subsequent missing values also contains much more. This justifies to exclude them from the further analysis.

[3] We determine the average distance of one observation to the remaining observations by simply calculating the average value of all entries in the corresponding row/column of the dissimilarity matrix.

Based on the dissimilarity matrices, clustering is performed with two different algorithms, namely agglomerative clustering with complete linkage and partitional clustering in the form of K-medoids. As we use seven distance measures in combination with two clustering algorithms, this results in 14 different clusterings within each formation period. For each of them, the appropriate number of clusters is determined by means of the CH index which is calculated over the range $K = 2, \ldots, 25$.

The last step prior to trading consists in the identification of suitable pairs. In each formation period, our data set is divided into several clusters, for a given strategy. Within each cluster, we choose three pairs of stocks in the following way: First, we sort the corresponding pairs of stocks by ascending distance. Then, we pick that pair of stocks having the smallest distance[4] and perform a Phillip-Ouliaris test for cointegration by computing the multivariate trace statistic. If both price series are cointegrated on a significance level of 5%, they are nominated as a tradable pair. We proceed this way and test pairs of stocks by ascending distance until we classified three pairs as tradable. Note that we do not allow stocks to be part of more than one pair in order to avoid concentration in single stocks. This way, we obtain three distinct pairs of stocks per cluster whereas each of them is significantly cointegrated on a level of 5%.

The above described procedure combines aspects of the distance method and the cointegration approach for the identification of suitable pairs. We adapt the distance method used by Gatev et al. [2006] and introduce several more sophisticated distance measures. Each of them takes into account different information of price or return series which are related to the main stylized facts of financial time series. We investigate if any other information besides the SSD can improve the identification of suitable pairs.

Distance measures or (dis)similarity naturally lead to the technique of clustering, which can be applied in a meaningful way at this point. The separation of the asset universe into clusters of time series sharing similar characteristics substantially reduces the number of possible pairs and therefore, the number of cointegration tests to be performed. We expect most of the pairs with a comparatively small distance to be cointegrated. Proceeding this way, the otherwise existing multiple testing problem can be mitigated in a natural way.

Due to the econometrically sound foundation defining equilibrium relationships between time series, cointegration represents a rigorous framework for PT, which should certainly be applied. Gatev et al. [2006] identify tradable pairs solely by means of their SSD. However, as stated by Krauss [2017], "omitting cointegration testing is contradictory to the requirements of a rational investor. Spurious relationships based on an assumption of return parity are not mean-reverting. The potential lack of an equilibrium relationship leads to higher divergence risks and thus to potential losses."

In practice, stocks are often grouped by industry or sector. The underlying idea is that if two stocks were of companies in the same industry, then they also have similarities in their business operations and thus, their stocks should move together to some extent. However,

[4] We skip all pairs with a distance smaller than $\omega = \sqrt[4]{\varepsilon} \approx 0.00012$, where ε denotes the machine epsilon. Using this value of ω turned out to produce satisfactory results for all distance measures used. Without skipping these pairs, we would select too many pairs moving too close together and thus, being unprofitable.

this is certainly not always the case. Moreover, this approach can not simply be adapted to different asset universes. By contrast, the technique of time series clustering does not make any assumptions about the data structure and therefore, it can be applied to any arbitrary sample without any further ado.

5.3.2.2 Trading Strategy

Having identified a set of tradable pairs within each formation period, we need to construct a trading rule specifying when to open or close a position. Vidyamurthy [2004] provides a basic framework applying cointegration to PT. Rad et al. [2016] further extend the concept into an executable PT strategy, which serves as a basis for our strategy.

Just as in the formation period, a cumulative total return index is constructed for each stock, taking a value of 1 € on the first day of each trading period. This way, there occurs no look-ahead bias since prices and returns are directly observable at any given point in time.

In order to construct a suitable trading rule, we have to focus on the price spread s_t. Since each nominated pair is cointegrated, the spread series can be defined as scaled difference of the individual price series:

$$s_t = p_{2,t} - \beta\, p_{1,t}, \tag{5.3.1}$$

where $p_{i,t}$ denotes the price of stock $i \in \{1, 2\}$ at time t and β is the cointegrating coefficient of a given pair. By definition, the spread series s_t is stationary and exhibits a mean-reverting behavior. This fact is crucial for a PT strategy that uses deviations from the long-run equilibrium as indicator to open long-short positions (Rad et al. [2016]).

Based on the prices of the formation period, we calculate the spread's mean μ_f and standard deviation σ_f for each pair. These parameters serve as trade's open and close triggers. In the subsequent trading period, we consider the normalized cointegration spread $\tilde{s}_t$ for each pair:

$$\tilde{s}_t = \frac{s_t - \mu_f}{\sigma_f}. \tag{5.3.2}$$

Following practice, we open a long-short position if the normalized spread diverges beyond ± 2. By construction, we buy 1 € worth of stock 1 and sell short $1/\beta$ € worth of stock 2, if the normalized spread exceeds the value $+2$. Equivalently, we buy 1 € worth of stock 2 and sell short β € worth of stock 1, if the normalized spread drops below the value -2. We close both positions *i*) once the normalized spread returns to zero, or *ii*) in case one stock is not listed as a constituent of the MSCI Europe Index anymore, or *iii*) at the latest by the end of each trading period. In the first case, a closed pair is monitored for the rest of the trading period for another potential trade. Note that opening or closing a position on the same day of divergence or convergence may bias the achieved returns upwards due to the bid-ask bounce (Jegadeesh [1990], Jegadeesh and Titman [1995], Gatev et al. [2006]). Therefore, we open or close a position only on the following day of divergence or convergence.

To illustrate the methodology of PT and our trading rule, Figure 5.3.2 shows the cumulative return indices of two companies, Rexel S.A. and Dufry AG, for the trading period from August 2016 to January 2017. In addition, the normalized spread series is depicted beneath.

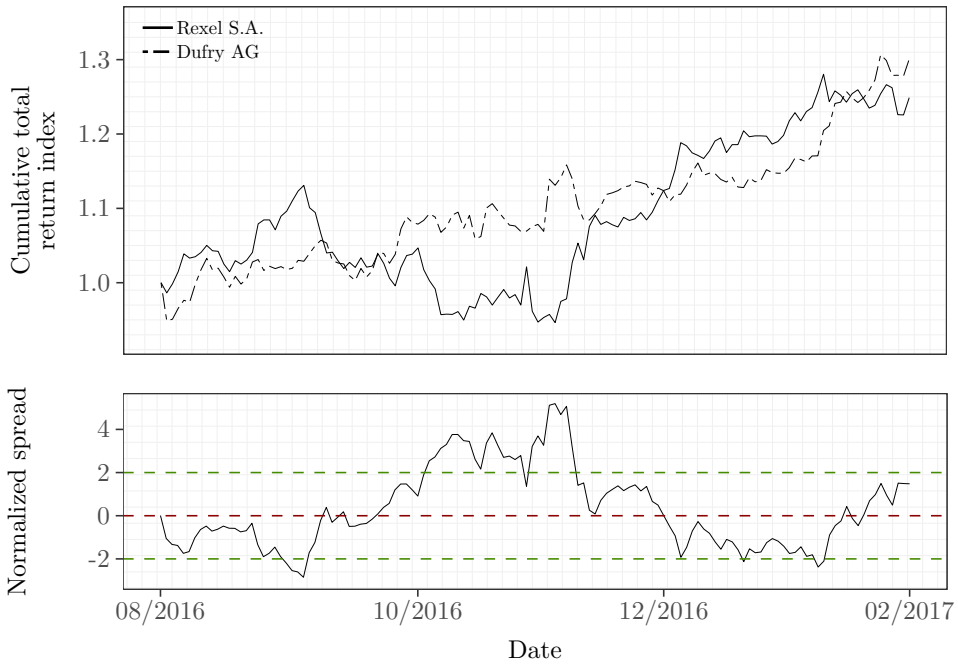

Figure 5.3.2 Exemplary pair of stocks traded in the period from August 2016 to January 2017: Cumulative return indices (top) and normalized spread (bottom).

5.3.2.3 Calculation of Returns

We record the performance of all 14 strategies and evaluate it by means of different performance measures, including returns. Gatev et al. [2006], Do and Faff [2010] and Rad et al. [2016] calculate two types of returns, namely the return on committed capital and the return on employed capital. The former scales the portfolio payoffs by the number of pairs in the portfolio, whereas the latter divides the payoffs just by the number of pairs that are actually traded during a trading period. In our case, time series are clustered and the appropriate number of clusters is determined by means of the CH index in each formation period. Therefore, the number of clusters and thus, the number of traded pairs, can naturally change over time. For this reason, we only consider the return on committed capital for calculation of returns in order to achieve a consistent result.

As we backtest our strategies by using a 1-month rolling window, this leads to six overlapping portfolios each month. Calculating their equally weighted average return yields the monthly excess return of a strategy. Note that the performance outcomes are slightly underestimated as we do not take into account any potential interest earned on capital while it is not involved in a trade (Rad et al. [2016]).

For reasons of simplicity, we do not consider any transaction costs in our analysis. If constant commissions are assumed, the approximate transaction costs can still be determined by means of the average number of round trips per pair (four trades), the average proportion of stocks that are unlisted from the MSCI Europe Index during a trade (two trades), and the average percentage of pairs that did not converge at the end of a trading period (two trades). The corresponding information will be provided for each strategy.

5.3.3 Empirical Results

Table 5.3.1 reports the characteristics of the monthly excess return distribution for each strategy for the period from the beginning of the year 2000 until the end of 2018. Henceforth, the first eight strategies that employ a raw data-based distance measure are referred to as "raw data-based strategies". Equivalently, the remaining six strategies that employ a feature-based distance measure are referred to as "feature-based strategies".

Table 5.3.1 Pairs Trading Strategies' Monthly Excess Return Characteristics.

Strategy	Mean	t-statistic	Std Dev	Skewness	Kurtosis	Min	Max	% Obs < 0
EUC/AC/CMPL	0.0051	6.8308***	0.0133	1.2753	4.4376	-0.0186	0.0708	33.48
EUC/PC/KMED	0.0062	7.3083***	0.0128	2.6284	14.1511	-0.0218	0.0935	28.19
DTW/AC/CMPL	0.0045	6.6078***	0.0104	1.1437	4.0239	-0.0220	0.0610	32.16
DTW/PC/KMED	0.0043	6.5375***	0.0098	0.6547	1.5278	-0.0204	0.0458	35.68
COR/AC/CMPL	0.0043	5.2553***	0.0124	1.2085	4.9941	-0.0308	0.0694	34.80
COR/PC/KMED	0.0048	5.2982***	0.0138	0.3494	2.4041	-0.0347	0.0664	32.16
CCR/AC/CMPL	0.0044	4.8077***	0.0138	0.7659	2.9769	-0.0362	0.0694	33.92
CCR/PC/KMED	0.0047	5.0973***	0.0138	0.3705	1.8340	-0.0329	0.0645	33.04
ACF/AC/CMPL	0.0023	2.6446***	0.0131	1.4578	5.6410	-0.0299	0.0736	44.93
ACF/PC/KMED	0.0023	2.3538**	0.0147	2.2101	12.0579	-0.0421	0.1028	45.37
COP/AC/CMPL	0.0027	4.7098***	0.0086	0.2005	1.4965	-0.0292	0.0317	37.89
COP/PC/KMED	0.0037	5.6114***	0.0098	0.6791	3.1193	-0.0337	0.0530	36.56
QAF/AC/CMPL	0.0027	3.8009***	0.0107	0.8425	3.5418	-0.0307	0.0591	39.65
QAF/PC/KMED	0.0019	2.7916***	0.0084	0.4715	1.4619	-0.0322	0.0356	44.49

EUC: Euclidean distance, DTW: Dynamic time warping, COR: Pearson distance, CCR: Cross-correlation distance, ACF: Autocorrelation-based distance, COP: Copula-based distance, QAF: Quantile autocovariance-based distance, AC: Agglomerative clustering, PC: Partitional clustering, CMPL: Complete linkage, KMED: K-medoids, Std Dev: Standard deviation, Min: Minimum, Max: Maximum, Obs: Observations, Significance levels: *** 1%, ** 5%.

Each of the 14 strategies shows a positive statistically significant average monthly excess return over the full sample period of 227 months. Some of the achieved returns are relatively high in a statistical and economical sense, but we have to keep in mind that transaction costs are not considered. The highest average excess return of 0.62% per month (t-statistic of 7.3083) is earned by the strategy that employs the Euclidean distance in combination with K-medoids. By contrast, the strategy that employs the quantile autocovariance-based distance in combination with K-medoids performs worst and yields an average monthly excess return of just 0.19% (t-statistic of 2.7916).

Moreover, all strategies show a relatively low standard deviation, with the highest of 0.0147 belonging to the strategy that employs the autocorrelation-based distance in combination with K-medoids, and the lowest of 0.0084 belonging to the strategy that also achieved the lowest return. All return distributions are positively skewed, indicating that more positive than negative returns are achieved. The last column of Table 5.3.1 confirms this, showing that around one third of the returns are negative for the raw data-based strategies. Feature-based

strategies yield a slightly higher proportion of negative returns with the maximum of 45.37% belonging to the strategy that employs the autocorrelation-based distance in combination with K-medoids.

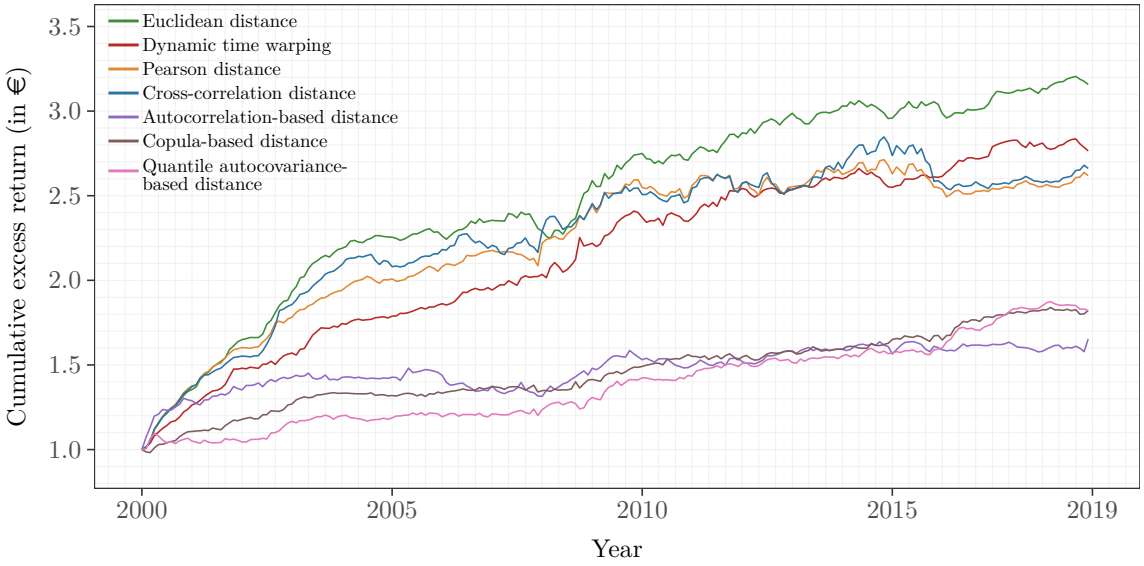

Figure 5.3.3 Cumulative excess return (Agglomerative clustering/Complete linkage).

In each case, the maximum monthly excess return is of (much) larger magnitude than the minimum monthly excess return. The strategy that employs the Euclidean distance in combination with K-medoids earns a maximum monthly excess return of more than 9%. Due to this, it also shows a comparatively high kurtosis, taking a value of larger than 10.

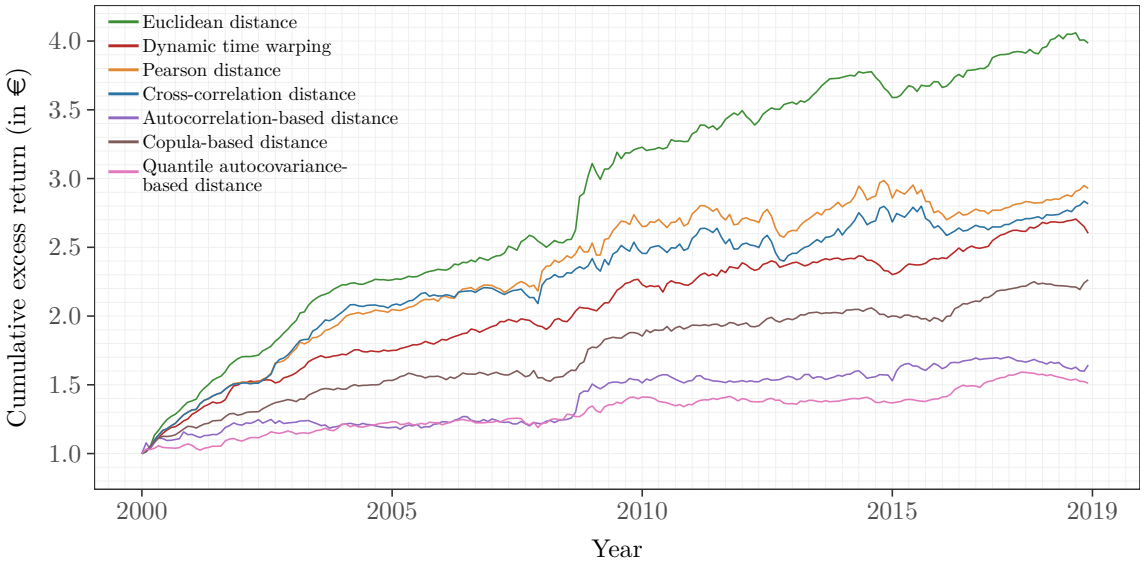

Figure 5.3.4 Cumulative excess return (Partitional clustering/K-medoids).

Notably, both strategies that employ the simplest distance measure –the Euclidean distance– achieve the highest average monthly excess returns, followed by the remaining raw data-based strategies. Those strategies generally seem to outperform the feature-based

strategies, regardless of the clustering algorithm. This can be also seen in Figure 5.3.3 and 5.3.4, respectively. Both figures compare the cumulative excess return (in €) of the different strategies, each of them for a fixed clustering algorithm.[5] More precisely, they show the evolution of wealth over a period of 19 years upon an investment of 1 €. Obviously, the choice of the distance measure is much more important than the choice of the clustering algorithm, which seems to have a comparatively small impact on the performance of the strategies. The only relevant difference exists for the two strategies that employ the Euclidean and the copula-based distance. Here, K-medoids clustering clearly leads to better results.

Especially in the beginning of the backtesting period, raw data-based strategies perform considerably well. Most of them also show a good performance during the global financial crisis of 2007-09. However, the individual strategies suffer several drawdowns. The performance of feature-based strategies in combination with agglomerative clustering is surprisingly stable over time. They generate comparatively small returns which, however, remain almost constant throughout the whole backtesting period. In combination with K-medoids, the copula-based strategy performs almost as well as raw data-based strategies, while the autocorrelation-based and the quantile autocovariance-based strategies yield a rather poor performance. Notably, feature-based strategies hardly suffer any large drawdowns.

Table 5.3.2 Pairs Trading Strategies' Performance Measures.

Strategy	General measures			LPM measures		Drawdown measures	
	VaR (95%)	ES (95%)	Sharpe ratio	Omega	Sortino ratio	MDD	Calmar ratio
EUC/AC/CMPL	-0.0098	-0.0131	0.4513	3.7059	1.2394	0.0649	0.9659
EUC/PC/KMED	-0.0095	-0.0130	0.4844	5.2023	1.6709	0.0498	1.5231
DTW/AC/CMPL	-0.0100	-0.0132	0.4327	3.5359	1.1094	0.0419	1.3174
DTW/PC/KMED	-0.0101	-0.0137	0.4388	3.2177	1.0196	0.0559	0.9273
COR/AC/CMPL	-0.0154	-0.0190	0.3468	2.7846	0.7818	0.0809	0.6454
COR/PC/KMED	-0.0185	-0.0256	0.3478	2.6267	0.6714	0.0956	0.6115
CCR/AC/CMPL	-0.0163	-0.0226	0.3188	2.4669	0.6518	0.1097	0.4842
CCR/PC/KMED	-0.0182	-0.0242	0.3406	2.5037	0.6528	0.0910	0.6184
ACF/AC/CMPL	-0.0139	-0.0199	0.1756	1.6623	0.3490	0.1127	0.2389
ACF/PC/KMED	-0.0161	-0.0216	0.1565	1.6129	0.3201	0.0613	0.4347
COP/AC/CMPL	-0.0095	-0.0150	0.3140	2.3811	0.5988	0.0292	1.1034
COP/PC/KMED	-0.0099	-0.0142	0.3776	2.8492	0.8088	0.0489	0.9022
QAF/AC/CMPL	-0.0122	-0.0171	0.2523	2.0147	0.4905	0.0548	0.5872
QAF/PC/KMED	-0.0121	-0.0184	0.1881	1.6493	0.3198	0.0543	0.4067

EUC: Euclidean distance, DTW: Dynamic time warping, COR: Pearson distance, CCR: Cross-correlation distance, ACF: Autocorrelation-based distance, COP: Copula-based distance, QAF: Quantile autocovariance-based distance, AC: Agglomerative clustering, PC: Partitional clustering, CMPL: Complete linkage, KMED: K-medoids, LPM: Lower partial moments, VaR: Value at Risk, ES: Expected shortfall, MDD: Maximum drawdown.

[5] In Appendix D, seven additional figures are provided, where each of them compares both clustering algorithms for a fixed distance measure.

Besides analyzing the performance of the strategies by means of their excess returns, we further consider their risk profiles using several risk-adjusted measures. These include the Sharpe ratio as the classical risk-adjusted measure, which is defined as the excess return relative to the standard deviation of an investment. Similar to Rad et al. [2016], we report various downside measures, which can be divided into two main groups: lower partial moment measures and drawdown measures.

The Omega and the Sortino ratio belong to the first group and consider only negative deviations of returns from a minimum acceptable return, which is set to 0%. The Omega can be seen as the ratio of returns above this threshold to returns below this threshold. The Sortino ratio is defined as the average excess return relative to the downside deviation, which corresponds to the standard deviation of negative returns.

The maximum drawdown and the Calmar ratio belong to the second group and measure the loss incurred over a certain period of time. The maximum drawdown is the largest peak-to-trough decline and therefore, it states the maximum possible loss of a strategy. The Calmar ratio is defined as the average excess return relative to the maximum drawdown. For a detailed explanation of these measures, we refer to Eling and Schuhmacher [2007].

Table 5.3.2 contains the above described risk-adjusted measures for each strategy. For the sake of completeness, the Value at Risk and the expected shortfall are also reported. As expected, the strategy that employs the Euclidean distance in combination with K-medoids yields the best overall performance except for the maximum drawdown measure. Generally, raw data-based strategies seem to perform better than feature-based strategies, but they mainly come along with (much) larger drawdowns. The risk-adjusted performance of the copula-based strategies are surprisingly good. Both strategies perform almost as well as the four strategies that employ the Pearson distance and the cross-correlation distance. Notably, the autocorrelation-based strategies yield the worst overall performance, whereas the one in combination with agglomerative clustering suffers the largest drawdown of more than 11%. Furthermore, they both show the lowest Sharpe ratios of all 14 strategies.

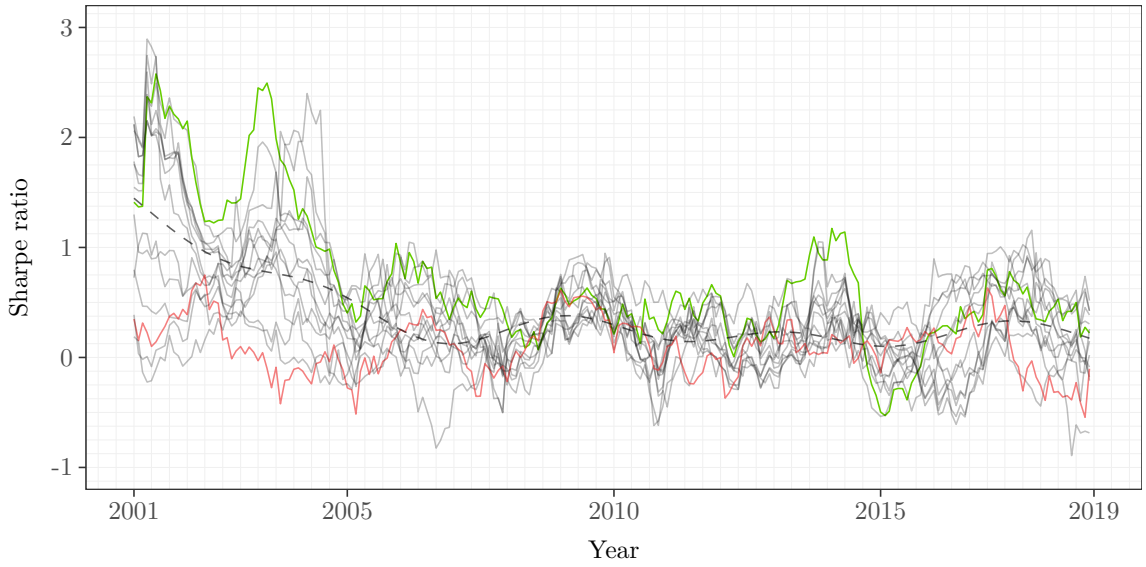

Figure 5.3.5 1-year rolling sample Sharpe ratio of each PT strategy.

Figure 5.3.5 shows the risk-adjusted performance of all strategies in form of a 1-year rolling sample Sharpe ratio. The strategy with the highest (lowest) total Sharpe ratio of 0.4844 (0.1565) is color coded in green (red). Moreover, a dashed line in form of a thin plate regression spline is added to illustrate the overall trend development. The highest risk-adjusted performance is achieved by most of the strategies in the first years of the backtesting period. However, the performance considerably declines in subsequent years. Throughout the whole backtesting period, the risk-adjusted performance clearly fluctuates.

Table 5.3.3 Pairs Trading Strategies' Traded Stocks & Pairs and Trading Frequency.

Strategy	Stocks		Pairs				
	# Distinct	% Unlisted	# Total	# Distinct	Round trips	Days open	% Converged
EUC/AC/CMPL	499	4.67	1582	684	1.20/1.93	21.01/11	73.97
EUC/PC/KMED	379	4.76	1342	485	1.29/2.02	20.45/11	76.66
DTW/AC/CMPL	641	5.60	2087	1159	1.05/1.72	23.27/14	71.30
DTW/PC/KMED	569	4.71	1962	1031	1.07/1.74	22.97/14	71.49
COR/AC/CMPL	589	7.78	1601	678	1.04/1.57	24.11/15	70.11
COR/PC/KMED	486	6.87	1399	487	1.11/1.59	23.30/14	72.06
CCR/AC/CMPL	564	7.83	1572	645	1.01/1.49	24.76/15	69.43
CCR/PC/KMED	490	6.53	1408	493	1.10/1.59	23.79/15	71.84
ACF/AC/CMPL	622	5.85	1493	787	0.81/1.53	27.77/18	61.78
ACF/PC/KMED	616	7.84	1442	792	0.80/1.52	28.44/18	61.76
COP/AC/CMPL	525	5.36	1469	625	0.86/1.63	25.76/16	63.20
COP/PC/KMED	502	3.84	1490	652	0.90/1.62	25.46/15	64.78
QAF/AC/CMPL	526	4.17	1426	627	0.83/1.53	27.50/17	62.52
QAF/PC/KMED	538	3.44	1530	753	0.80/1.47	27.06/17	60.85

EUC: Euclidean distance, DTW: Dynamic time warping, COR: Pearson distance, CCR: Cross-correlation distance, ACF: Autocorrelation-based distance, COP: Copula-based distance, QAF: Quantile autocovariance-based distance, AC: Agglomerative clustering, PC: Partitional clustering, CMPL: Complete linkage, KMED: K-medoids.
"% Unlisted" reports the percentage of trades where at least one stock was unlisted from the Index during the trade. "Round trips" reports the mean and standard deviation of complete round trips per pair. "Days open" reports the mean and median number of days that a trade remained open. "% Converged" reports the average percentage of trades that converged.

Having analyzed the performance of the strategies, we further investigate the properties of the individual trades executed. For this purpose, Table 5.3.3 reports several summary statistics on the amount of traded pairs and the trading frequency of each strategy.

Each strategy trades on average around 1500 pairs (3000 stocks) throughout the entire backtesting period except for the strategies that employ the DTW distance. However, only around 45% (17%) of the pairs (stocks) are distinct in each case, implying that the underlying clustering and cointegration mechanism tends to select only a limited set of stocks and pairs that are traded by the strategies.

Surprisingly, the average number of complete round trips[6] per pair is relatively low. Raw

[6] In case a stock is unlisted from the MSCI Europe Index during a trade or a trade is automatically closed at the end of the trading period due to non-convergence, the trade is not classified as a round trip.

data-based strategies generate on average slightly more than one round trip per pair in each trading period, whereas feature-based strategies generate on average slightly less than one round trip. The comparatively high standard deviations indicate that for each strategy, there exist several pairs generating many more round trips than the average pair does. A round trip takes on average around one month, whereas the duration is slightly longer for feature-based strategies than for raw data-based strategies. However, a much smaller value of the median indicates that on the one hand, many trades converge much faster, but on the other hand, some trades take considerably longer to converge. The strategy with the best performance (Euclidean distance combined with K-medoids) shows the highest number of round trips together with the shortest trade duration and the highest proportion of converged trades. It can be clearly seen that a decreasing number of round trips comes along with a higher average trade duration and a lower proportion of converged trades.

Chapter 6

Summary and Discussion

In this thesis, we focused on a range of statistical techniques in order to determine when financial time series can be considered similar. Based on these methods, we developed a statistical arbitrage strategy, known as *Pairs Trading*, and backtested it over a period of 19 years on a broad asset universe, consisting of the MSCI Europe Index constituents.

After a short review of the main stylized facts of financial time series, we discussed the property of cointegration, together with two testing procedures. Moreover, we highlighted the multiple testing problem which occurs when many pairs of assets are simultaneously tested for cointegration. This is generally considered a crucial step to identify suitable pairs for a PT strategy.

Besides testing for cointegration, pairs of assets are often identified by measuring the distance between their price paths in terms of the SSD. This approach is known as distance method and is used by Gatev et al. [2006], among others. We adapted this approach and introduced various other distance measures which try to take into account the main stylized facts of financial time series. The concept of distance or (dis)similarity measures is naturally linked to the technique of clustering. Most traditional clustering methods can be applied in the temporal context without any further ado. We addressed several hierarchical and partitional clustering algorithms together with three popular external measures to validate the goodness of a clustering. The division of time series into homogeneous clusters naturally mitigates the otherwise existing multiple testing problem. The number of cointegration tests to be performed can substantially be reduced due to a smaller number of possible pairs within each cluster. In addition, we expect the majority of pairs with a comparatively small distance to be cointegrated.

In the empirical application, we constructed a PT strategy and backtested it over the period from January 2000 to December 2018. Our data set consisted of daily total return indices of the MSCI Europe Index constituents. The combination of seven distance measures and two clustering algorithms yielded 14 different strategies, which could be divided into two subgroups: raw data-based and feature-based strategies. The former showed a better overall performance measured in terms of the average monthly excess return and several other risk-adjusted performance measures. Feature-based strategies earned a comparatively low excess return, but in combination with agglomerative clustering, their performance was

surprisingly stable over time. Raw data-based strategies showed a comparatively higher proportion of converged trades and generated on average a higher amount of round trips per pair. The strategy that employed the simplest distance measure –the Euclidean distance– earned in combination with K-medoids the highest average monthly excess return of over 7% per annum, before transaction costs. Similar to previous findings, the performance of most of our strategies considerably declined after the year 2005 and clearly fluctuated over time.

We had to take many decisions while developing the PT strategy, which were surely not always the optimal ones. Most likely, the performance of the strategies can be further improved by tuning some critical parameters such as the input parameters of the (feature-based) distance measures, the significance level of the cointegration test, the number of selected pairs per cluster, and of course, the value of the trigger when to open a trade. In order to limit potentially high losses, the implementation of a stop-loss rule would also be advantageous since around one third of the trades diverged, but never converged again.

Our analysis was solely based on the information about the prices of the individual stocks. However, the whole framework could be slightly modified by incorporating fundamental information such as the market capitalization or the creditworthiness of a company but also the sector or the industry to which it belongs to. By performing a principal component analysis, the dimensionality of the price or return data could be further reduced, yielding a set of common component loadings for each asset. Together with the fundamental data, some of these features could be processed by a suitable clustering algorithm which again yields a partition of the asset universe.

Furthermore, the term “pairs” trading could be extended in the sense that not only pairs consisting of two assets but rather a “portfolio” of pairs is simultaneously traded. This would require a number of assets, whose linear combination is stationary and exhibits a mean-reverting behavior. In this case, an effective pre-partition of the asset universe would be essential, as otherwise the number of possible combinations would be far too large.

References

Aach, J. and G.M. Church (2001). *Aligning Gene Expression Time Series with Time Warping Algorithms*. In: Bioinformatics 17(6), pp. 495–508.

Aggarwal, C.C. and C.K. Reddy (2018). *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press.

Aghabozorgi, S. et al. (2015). *Time Series Clustering – A Decade Review*. In: Information Systems 53, pp. 16–38.

Anderberg, M.R. (1973). *Cluster Analysis for Applications*. Probability and Mathematical Statistics. Academic Press.

Arthur, D. and S. Vassilvitskii (2007). *K-Means++: The Advantages of careful Seeding*. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, pp. 1027–1035.

Berthold, M.R. and F. Höppner (2016). *On Clustering Time Series using Euclidean Distance and Pearson Correlation*. In: arXiv fpreprint arXiv:1601.02213.

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

Brockwell, P.J. and R.A. Davis (2013). *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York.

Calinski, T. and J. Harabasz (1974). *A dendrite Method for Cluster Analysis*. In: Communications in Statistics – Theory and Methods 3(1), pp. 1–27.

Celebi, M.E. et al. (2013). *A Comparative Study of efficient Initialization Methods for the K-Means Clustering Algorithm*. In: Expert Systems with Applications 40(1), pp. 200–210.

Clegg, M. and C. Krauss (2018). *Pairs Trading with partial Cointegration*. In: Quantitative Finance 18(1), pp. 121–138.

Deza, M.M. and E. Deza (2014). *Encyclopedia of Distances*. Springer Berlin Heidelberg.

Díaz, S.P. and J.A. Vilar (2010). *Comparing several parametric and nonparametric Approaches to Time Series Clustering: A Simulation Study*. In: Journal of Classification 27(3), pp. 333–362.

Dickey, D.A. and W.A. Fuller (1979). *Distribution of the Estimators for autoregressive Time Series with a Unit Root.* In: Journal of the American Statistical Association 74(366a), pp. 427–431.

Do, B. and R. Faff (2010). *Does simple Pairs Trading still work?* In: Financial Analysts Journal 66(4), pp. 83–95.

Do, B. and R. Faff (2012). *Are Pairs Trading Profits robust to Trading Costs?* In: Journal of Financial Research 35(2), pp. 261–287.

Dunn, J.C. (1974). *Well-separated Clusters and optimal fuzzy Partitions.* In: Journal of Cybernetics 4(1), pp. 95–104.

D'Urso, P. et al. (2016). *GARCH-based robust Clustering of Time Series.* In: Fuzzy Sets and Systems 305, pp. 1–28.

D'Urso, P. et al. (2017). *Autoregressive Metric-based trimmed fuzzy Clustering with an Application to PM10 Time Series.* In: Chemometrics and Intelligent Laboratory Systems 161, pp. 15–26.

D'Urso, P. et al. (2013). *Clustering of financial Time Series.* In: Physica A: Statistical Mechanics and its Applications 392(9), pp. 2114–2129.

Eling, M. and F. Schuhmacher (2007). *Does the Choice of Performance Measure influence the Evaluation of Hedge Funds?* In: Journal of Banking & Finance 31(9), pp. 2632–2647.

Elliott, R.J. et al. (2005). *Pairs Trading.* In: Quantitative Finance 5(3), pp. 271–276.

Engle, R.F. (1982). *Autoregressive conditional heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.* In: Econometrica: Journal of the Econometric Society, pp. 987–1007.

Engle, R.F. and C.W.J. Granger (1987). *Co-Integration and Error Correction: Representation, Estimation, and Testing.* In: Econometrica: Journal of the Econometric Society, pp. 251–276.

Francq, C. and J.M. Zakoian (2011). *GARCH Models: Structure, Statistical Inference and financial Applications.* Wiley.

Gan, G. et al. (2007). *Data Clustering: Theory, Algorithms, and Applications.* ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics.

Gatev, E. et al. (2006). *Pairs Trading: Performance of a relative-value Arbitrage Rule.* In: The Review of Financial Studies 19(3), pp. 797–827.

Gatev, E.G. et al. (1999). *Pairs Trading: Performance of a relative-value Arbitrage Rule.* Working Paper 7032. National Bureau of Economic Research.

Han, J. et al. (2011). *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science.

Harlacher, M. (2016). *Cointegration based algorithmic Pairs Trading*. Dissertation. University of St. Gallen. URL: https://www1.unisg.ch/www/edis.nsf/vEDISByTitleDE/E508BC43F8A44A9EC1257D2600052841.

Harris, R. and R. Sollis (2003). *Applied Time Series Modelling and Forecasting*. Wiley.

Hennig, C. and T.F. Liao (2013). *How to find an appropriate Clustering for mixed-type Variables with Application to socio-economic Stratification*. In: Journal of the Royal Statistical Society: Series C (Applied Statistics) 62(3), pp. 309–369.

Hennig, C. et al. (2015). *Handbook of Cluster Analysis*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press.

Hesabi, Z.R. et al. (2015). *Data Summarization Techniques for Big Data – A Survey*. In: Handbook on Data Centers. Springer, pp. 1109–1152.

Itakura, F. (1975). *Minimum Prediction Residual Principle applied to Speech Recognition*. In: IEEE Transactions on Acoustics, Speech, and Signal Processing 23(1), pp. 67–72.

Jacobs, H. and M. Weber (2015). *On the Determinants of Pairs Trading Profitability*. In: Journal of Financial Markets 23, pp. 75–97.

Jegadeesh, N. (1990). *Evidence of predictable Behavior of Security Returns*. In: The Journal of Finance 45(3), pp. 881–898.

Jegadeesh, N. and S. Titman (1993). *Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency*. In: The Journal of Finance 48(1), pp. 65–91.

Jegadeesh, N. and S. Titman (1995). *Overreaction, delayed Reaction, and contrarian Profits*. In: The Review of Financial Studies 8(4), pp. 973–993.

Kaufman, L. and P.J. Rousseeuw (1987). *Clustering by means of Medoids*. In: Data Analysis based on the L1-Norm and Related Methods, pp. 405–416.

Kaufman, L. and P.J. Rousseeuw (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley New York.

Keogh, E.J. and M.J. Pazzani (2000). *Scaling up Dynamic Time Warping for Data Mining Applications*. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 285–289.

Keogh, E.J. and C.A. Ratanamahatana (2005). *Exact Indexing of Dynamic Time Warping*. In: Knowledge and Information Systems 7(3), pp. 358–386.

Krauss, C. (2017). *Statistical Arbitrage Pairs Trading Strategies: Review and Outlook*. In: Journal of Economic Surveys 31(2), pp. 513–545.

Lafuente-Rego, B. and J.A. Vilar (2016). *Clustering of Time Series using Quantile Autocovariances*. In: Advances in Data Analysis and Classification 10(3), pp. 391–415.

Lafuente-Rego, B. et al. (2018). *Robust fuzzy Clustering based on Quantile Autocovariances*. In: Statistical Papers, pp. 1–56.

Liao, T.W. (2005). *Clustering of Time Series Data – A Survey*. In: Pattern Recognition 38(11), pp. 1857–1874.

Linton, O. and Y. Whang (2007). *The Quantilogram: With an Application to evaluating directional Predictability*. In: Journal of Econometrics 141(1), pp. 250–282.

Liu, B. et al. (2017). *Intraday Pairs Trading Strategies on high frequency Data: The Case of Oil Companies*. In: Quantitative Finance 17(1), pp. 87–100.

Lütkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis*. Springer Berlin Heidelberg.

Mandelbrot, B.B. (1963). *The Variation of certain speculative Prices*. In: Journal of Business 36(4), pp. 394–419.

Milligan, G.W. and M.C. Cooper (1985). *An Examination of Procedures for Determining the Number of Clusters in a Data Set*. In: Psychometrika 50(2), pp. 159–179.

Moulavi, D. et al. (2014). *Density-based Clustering Validation*. In: Proceedings of the 2014 SIAM International Conference on Data Mining. SIAM, pp. 839–847.

MSCI (2018). *MSCI Europe Index*. URL: `https://www.msci.com/europe` (visited on March 3, 2019).

Müller, M. (2007). *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg.

Otranto, E. (2008). *Clustering heteroskedastic Time Series by Model-based Procedures*. In: Computational Statistics & Data Analysis 52(10), pp. 4685–4698.

Paparrizos, J. and L. Gravano (2015). *K-shape: Efficient and accurate Clustering of Time Series*. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM, pp. 1855–1870.

Petitjean, F. et al. (2011). *A global Averaging Method for Dynamic Time Warping, with Applications to Clustering*. In: Pattern Recognition 44(3), pp. 678–693.

Phillips, P.C.B. and S. Ouliaris (1990). *Asymptotic Properties of Residual based Tests for Cointegration*. In: Econometrica: Journal of the Econometric Society, pp. 165–193.

Piccolo, D. (1990). *A Distance Measure for classifying ARIMA Models*. In: Journal of Time Series Analysis 11(2), pp. 153–164.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. URL: `https://www.R-project.org/`.

Rad, H. et al. (2016). *The Profitability of Pairs Trading Strategies: Distance, Cointegration and Copula methods*. In: Quantitative Finance 16(10), pp. 1541–1558.

Rath, T.M. and R. Manmatha (2003). *Word Image Matching using Dynamic Time Warping*. In: Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2. IEEE, pp. 512–527.

Ruppert, D. and D.S. Matteson (2015). *Statistics and Data Analysis for Financial Engineering: With R Examples*. Springer Texts in Statistics. Springer New York.

Sagvekar, V. et al. (2013). *Performance Assessment of CLARANS: A Method for Clustering Objects for Spatial Data Mining*. In: Global Journal of Engineering, Design and Technology 6(2), pp. 1–8.

Sakoe, H. and S. Chiba (1978). *Dynamic Programming Algorithm Optimization for spoken Word Recognition*. In: IEEE Transactions on Acoustics, Speech, and Signal Processing 26(1), pp. 43–49.

Sammut, C. and G.I. Webb (2017). *Encyclopedia of Machine Learning and Data Mining*. Encyclopedia of Machine Learning and Data Mining. Springer US.

Satchell, S. and J. Knight (2011). *Forecasting Volatility in the Financial Markets*. Quantitative Finance. Elsevier Science.

Sharpe, W.F. et al. (1990). *Investments, William F. Sharpe, Gordon J. Alexander, Fourth Edition: Instructor's Manual*. Prentice Hall.

Sklar, M. (1959). *Fonctions de repartition an dimensions et leurs marges*. In: Publ. Inst. Statist. Univ. Paris 8, pp. 229–231.

Tapinos, A. and P. Mendes (2013). *A Method for Comparing multivariate Time Series with different Dimensions*. In: PLOS ONE 8(2), pp. 1–11.

Thompson, S. (2013). *The stylised Facts of Stock Price Movements*. In: The New Zealand Review of Economics and Finance 1, pp. 50–77.

Vidyamurthy, G. (2004). *Pairs Trading: Quantitative Methods and Analysis*. Wiley Finance. Wiley.

Vilar, J.A. et al. (2018). *Quantile Autocovariances: A powerful Tool for hard and soft Partitional Clustering of Time Series*. In: Fuzzy Sets and Systems 340, pp. 38–72.

Vintsyuk, T.K. (1968). *Speech Discrimination by dynamic Programming*. In: Cybernetics 4(1), pp. 52–57.

Zeng, Z. and C. Lee (2014). *Pairs Trading: Optimal Thresholds and Profitability*. In: Quantitative Finance 14(11), pp. 1881–1893.

Zhang, B. and B. An (2018). *Clustering Time Series based on Dependence Structure*. In: PLOS ONE 13(11), pp. 1–22.

A Digital Appendix

The accompanying CD contains the following files and folders:

- `thesis_el-oraby.pdf`: PDF version of the thesis.
- `latex_figures.R`: R script to create the plots of the thesis.
- `backtests.RData`: R data file containing the backtest results of Chapter 5.
- `saft4pt`: R package folder containing the individual package components and the following main functions (a documentation is included in each R script):
  - `check_dist_mat.R`
  - `distance_matrices.R`
  - `distance_measures.R`
  - `execute_trades.R`
  - `extract_backtest.R`
  - `get_clustering.R`
  - `get_pairs.R`
  - `pairs_trade_backtest.R`
  - `performance_measures.R`
  - `process_data.R`
  - `remove_outliers.R`
  - `return_characteristics.R`
  - `stock_pair_summary.R`
  - `subtract_costs.R`
  - `transform_returns.R`
  - `est_acf.cpp`
  - `est_cop.cpp`
  - `est_qaf.cpp`
- `saft4pt_1.0.tar.gz`: R package source file, ready to install and to gain wealth.

B Copula-Based Distance Proof

By Equation 4.2.8, we have:

$$d_{cop}^{(h)}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{\int\int_{[0,1]^2} \left(\hat{C}_x^{(h)}(u,v)^2 - 2 \cdot \hat{C}_x^{(h)}(u,v)\, \hat{C}_y^{(h)}(u,v) + \hat{C}_y^{(h)}(u,v)^2 \right) dudv}.$$

By the definition of $\hat{C}_x^{(h)}(u,v)$, we can see that:

$$\begin{aligned}
&\int\int_{[0,1]^2} \hat{C}_x^{(h)}(u,v)^2\, dudv \\
&= \frac{1}{(n-h)^2} \int_0^1 \int_0^1 \sum_{i=1}^{n-h} \sum_{j=1}^{n-h} I(U_{x,i} \le u) I(V_{x,i} \le v) I(U_{x,j} \le u) I(V_{x,j} \le v)\, dudv \\
&= \frac{1}{(n-h)^2} \sum_{i=1}^{n-h} \sum_{j=1}^{n-h} \int_0^1 I(U_{x,i} \le u) I(U_{x,j} \le u)\, du \int_0^1 I(V_{x,i} \le v) I(V_{x,j} \le v)\, dv \\
&= \frac{1}{(n-h)^2} \sum_{i=1}^{n-h} \sum_{j=1}^{n-h} \left(1 - \max(U_{x,i}, U_{x,j})\right) \left(1 - \max(V_{x,i}, V_{x,j})\right) \\
&= L_{x,x}^{(h)}.
\end{aligned}$$

Similarly, we can verify that $\int\int_{[0,1]^2} \hat{C}_x^{(h)}(u,v)\, \hat{C}_y^{(h)}(u,v)\, dudv = L_{x,y}^{(h)}$. From this follows:

$$d_{cop}^{(h)}(\boldsymbol{x}_t, \boldsymbol{y}_t) = \sqrt{L_{x,x}^{(h)} - 2 \cdot L_{x,y}^{(h)} + L_{y,y}^{(h)}}$$

which simplifies calculations of the copula-based dissimilarity measure (Zhang and An [2018]).

C AR Metric for GARCH(1,1) Processes

Given two time series $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, they can be independently modeled as GARCH$(1,1)$ processes as follows:

$$\begin{aligned} x_t &= \mu_x + \varepsilon_{x,t} \qquad \varepsilon_{x,t} = u_{x,t}\sqrt{h_{x,t}}, \\ y_t &= \mu_y + \varepsilon_{y,t} \qquad \varepsilon_{y,t} = u_{y,t}\sqrt{h_{y,t}}, \end{aligned}$$

where the conditional variances $h_{x,t}$ and $h_{y,t}$ are independent from $u_{x,t}$ and $u_{y,t}$, respectively. They are modeled by:

$$\begin{aligned} \mathrm{Var}(\varepsilon_{x,t} \,|\, \mathcal{J}_{x,t-1}) = \mathbb{E}[\varepsilon_{x,t}^2 \,|\, \mathcal{J}_{x,t-1}] = h_{x,t} = \omega_x + \alpha_x\, \varepsilon_{x,t-1}^2 + \beta_x\, h_{x,t-1} \\ \mathrm{Var}(\varepsilon_{y,t} \,|\, \mathcal{J}_{y,t-1}) = \mathbb{E}[\varepsilon_{y,t}^2 \,|\, \mathcal{J}_{y,t-1}] = h_{y,t} = \omega_y + \alpha_y\, \varepsilon_{y,t-1}^2 + \beta_y\, h_{y,t-1} \end{aligned}$$

Adding the terms $\varepsilon_{i,t}^2$ and $\beta_i\, \varepsilon_{i,t-1}^2$ to $h_{i,t}$ for $i \in \{x, y\}$ and subtracting them again yields the ARMA$(1,1)$ representations of the squared disturbances:

$$\begin{aligned} \varepsilon_{x,t}^2 &= \omega_x + (\alpha_x + \beta_x)\, \varepsilon_{x,t-1}^2 - \beta_x(\varepsilon_{x,t-1}^2 - h_{x,t-1}) + (\varepsilon_{x,t}^2 - h_{x,t}) \\ \varepsilon_{y,t}^2 &= \omega_y + (\alpha_y + \beta_y)\, \varepsilon_{y,t-1}^2 - \beta_y(\varepsilon_{y,t-1}^2 - h_{y,t-1}) + (\varepsilon_{y,t}^2 - h_{y,t}). \end{aligned}$$

By recursive substitution, the squared disturbances can be represented as AR(∞) models:

$$\begin{aligned} \varepsilon_{x,t}^2 &= \frac{\omega_x}{1-\beta_x} + \sum_{k=1}^{\infty} \alpha_x\, \beta_x^{k-1}\, \varepsilon_{x,t-k}^2 + (\varepsilon_{x,t}^2 - h_{x,t}) \\ \varepsilon_{y,t}^2 &= \frac{\omega_y}{1-\beta_y} + \sum_{k=1}^{\infty} \alpha_y\, \beta_y^{k-1}\, \varepsilon_{y,t-k}^2 + (\varepsilon_{y,t}^2 - h_{y,t}), \end{aligned}$$

yielding the corresponding AR coefficients $\pi_{i,k} = \alpha_i\, \beta_i^{k-1},\ i \in \{x, y\}$. Plugging them into Equation 4.2.17 yields the AR distance for GARCH(1,1) processes:

$$\begin{aligned} d_{ar}(\boldsymbol{x}_t, \boldsymbol{y}_t) &= \sqrt{\sum_{k=1}^{\infty} \left(\pi_{x,k} - \pi_{y,k}\right)^2} = \sqrt{\sum_{k=1}^{\infty} \left(\alpha_x\, \beta_x^{k-1} - \alpha_y\, \beta_y^{k-1}\right)} = \sqrt{\sum_{k=0}^{\infty} \left(\alpha_x\, \beta_x^{k} - \alpha_y\, \beta_y^{k}\right)} \\ &= \sqrt{\alpha_x^2 \sum_{k=0}^{\infty} \beta_x^{2k} - 2\,\alpha_x\alpha_y \sum_{k=0}^{\infty} \left(\beta_x\beta_y\right)^k + \alpha_y^2 \sum_{k=0}^{\infty} \beta_y^{2k}} = \sqrt{\frac{\alpha_x^2}{1-\beta_x^2} - \frac{2\,\alpha_x\alpha_y}{1-\beta_x\beta_y} + \frac{\alpha_y^2}{1-\beta_y^2}}. \end{aligned}$$

D Cumulative Excess Returns of the PT Strategies

Evolution of wealth upon an investment of 1 € in the each strategy – comparison of both clustering algorithms for a fixed distance measure:

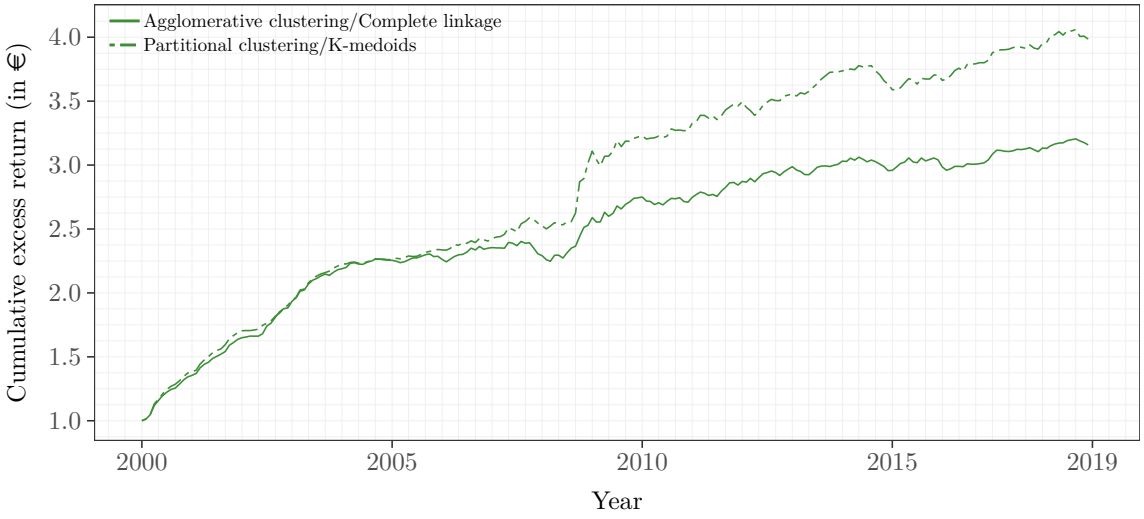

Figure D.0.1 Cumulative excess return (Euclidean distance).

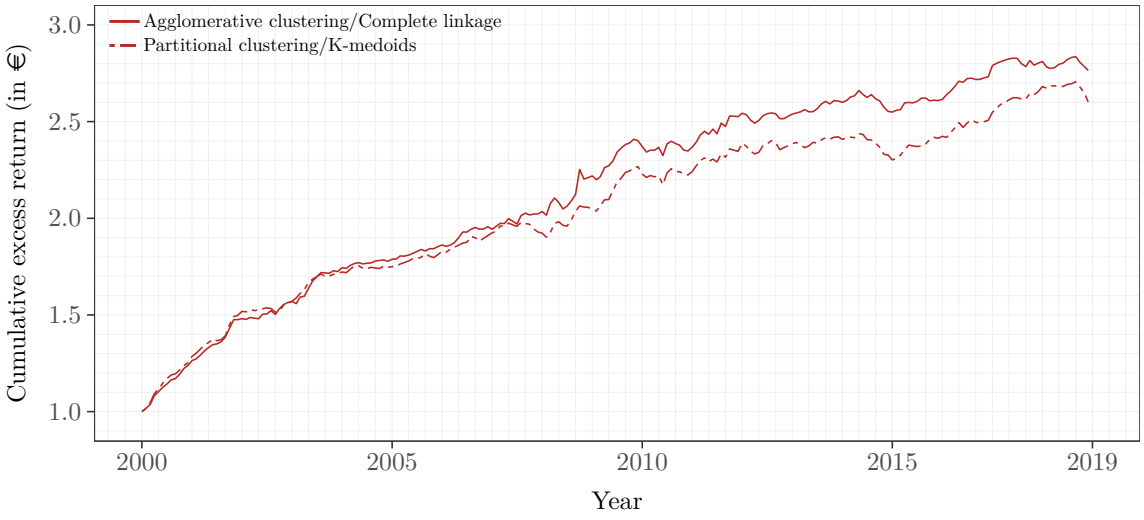

Figure D.0.2 Cumulative excess return (DTW distance).

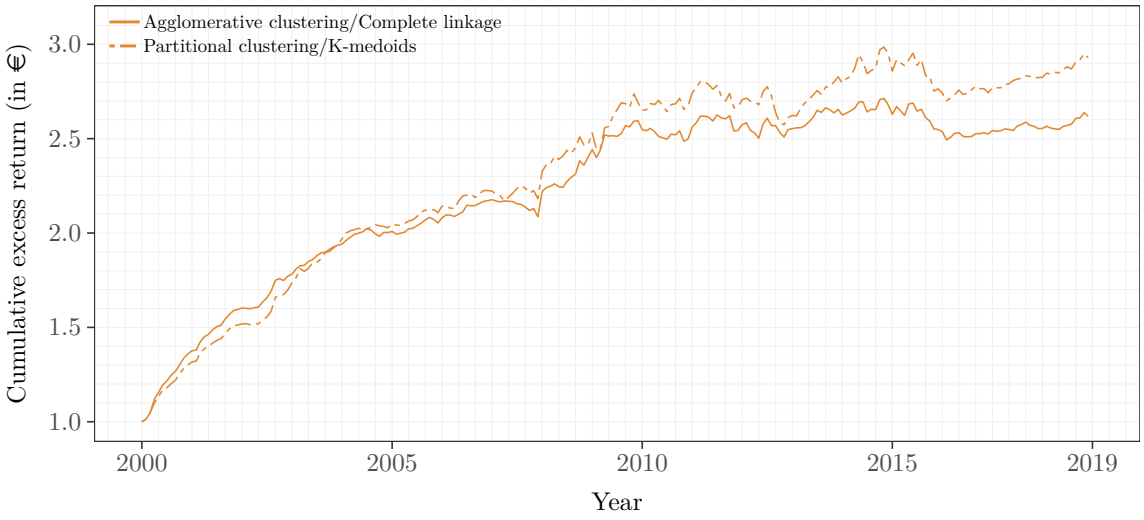

Figure D.0.3 Cumulative excess return (Pearson distance).

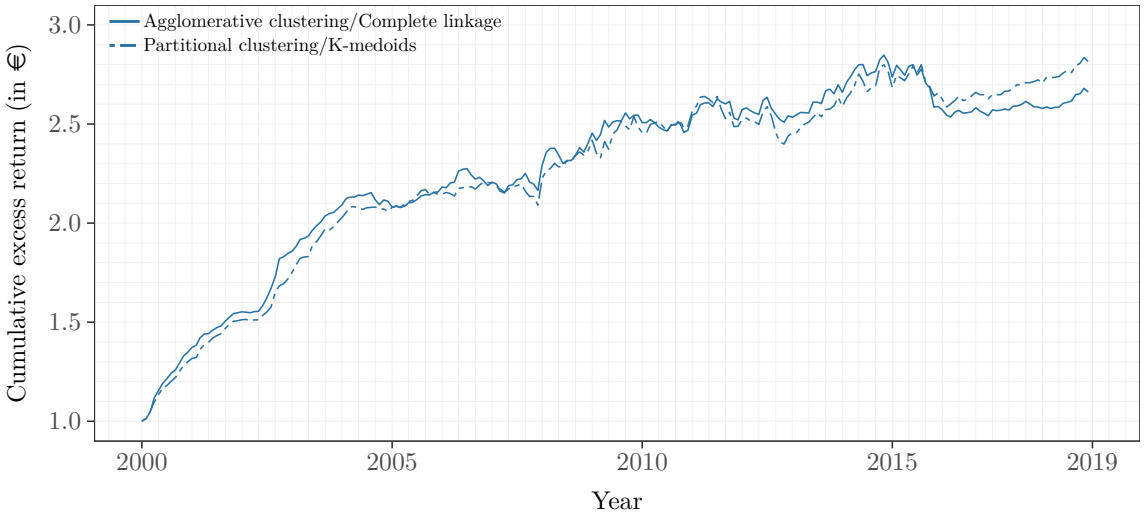

Figure D.0.4 Cumulative excess return (Cross-correlation distance).

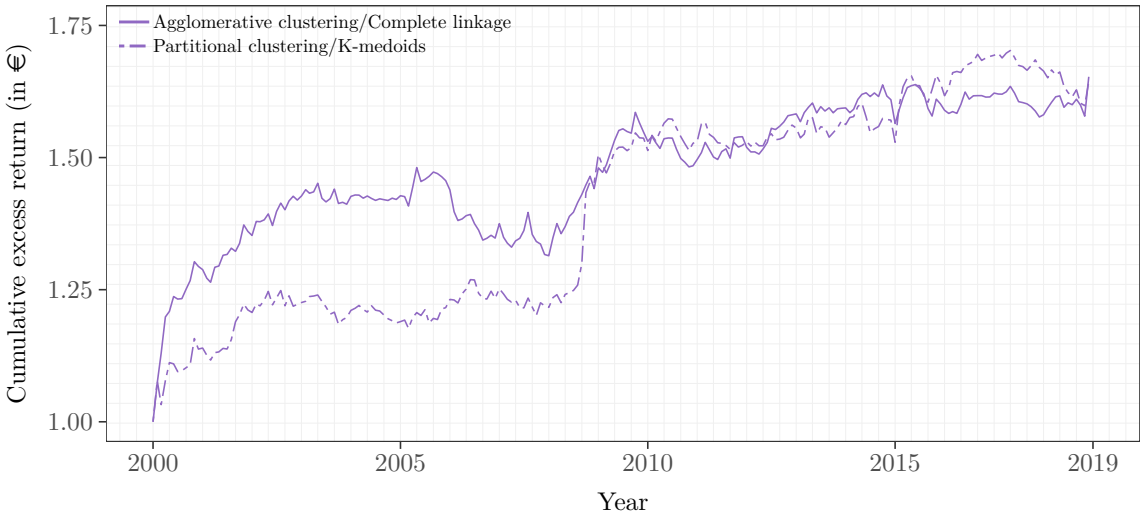

Figure D.0.5 Cumulative excess return (Autocorrelation-based distance).

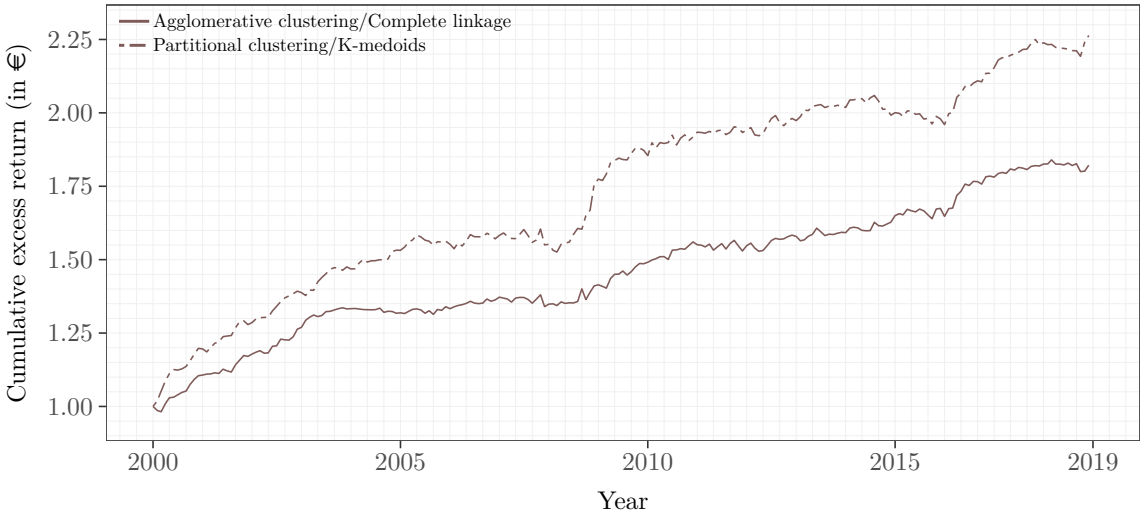

Figure D.0.6 Cumulative excess return (Copula-based distance).

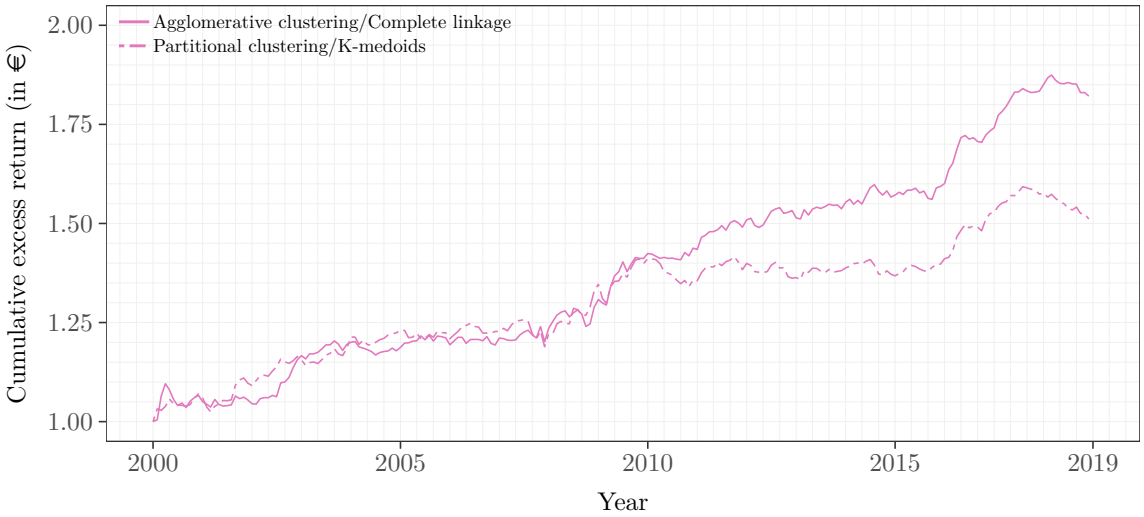

Figure D.0.7 Cumulative excess return (Quantile autocovariance-based distance).