Erasmus University Rotterdam

Thesis
Quantitative Finance (FEB63008-20)

Volatility clustering in Asset Allocation: a Sparsest Factor Model Approach

Author

Niklas Fastrich (444 285)

July 2nd 2021

Abstract

This paper finds that the inclusion of cluster-based selection criteria in asset allocation significantly improves the performance of returns on investment as well as the returns over realized volatility (Sharpe ratio). The process of risk minimization through diversification is a well-known strategy, however, the limited availability of high-quality assets often constrains the benefits of such a strategy and thus drawbacks in returns have to be borne. Hence, this paper asked the question to what degree clustering of those assets can increase improve the return vs risk-minimization trade-off when implemented in an existing portfolio strategy. With the help of sparsest factor models (SSFA) in a moving window framework, a stock index data set and a data set containing the constituents of the German DAX 30 index are partitioned into clusters. The silhouette method using the obtained clusters via SSFA and a complexity invariant distance (CID) measure are used to obtain the appropriate number of clusters at each investment date. Thereafter, from each cluster, the assets with the best historical Sharpe ratio are selected and placed with equal weights into a portfolio. For the stock index data, this paper finds that clear and distinct clusters are present which can be visualized and interpreted easily. For the DAX 30 data set, this is not the case. Nonetheless, for both data sets, the Sharpe ratio increased significantly as compared to a pure Sharpe ratio strategy and the market average. The main limitations of this research are with respect to the appropriateness of clustering for a given data set as the silhouette method cannot differentiate between two clusters and no clusters. In addition, SSFA has the drawback of having a constraint on the minimum sample size being equal to the number of assets included, which can become a problem for high dimensional data sets.

Contents

1 Introduction

One of the most well-known strategies in finance is the minimization of risks through diversification. On this account, many econometric models have been put forward. Amongst them are factor analysis (FA) models. They aim to explain the relationship of asset return data by summarizing their characteristics into latent variables (called factors). The formulation for factor models this paper follows given below:

$$\mathbf{X} = \mathbf{F}\mathbf{\Lambda}' + \mathbf{U}\mathbf{\Psi} + \mathbf{E}, \tag{1}$$

as defined in Adachi and Trendafilov (2018). **X** denotes a $(n * p)$ matrix of p observed variables with sample size n, $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ denote $(p * m)$ and $(p * p)$ matrices containing common factor and unique factor loadings, respectively. **F** and **U** are $(n * m)$ and $(n * p)$ matrices of common and unique factors, respectively and **E** denotes a $(n * p)$ matrix of errors following the distribution $\mathbf{E} \sim \mathbf{N}_p(\mathbf{O}_p, \mathbf{\Phi}^2)$.

The main goal of financial FA models is to gain insights into the cross-asset relationships driven by common underlying factors. Hence, the interpretation of $\mathbf{\Lambda}$ is vital. Optimal would be a relatively sparse $\mathbf{\Lambda}$ that allows for a straightforward interpretation of a specific subset of variables in **X** for each factor loading in $\mathbf{\Lambda}$. In practice, however, even if the underlying data-generating process follows a sparse $\mathbf{\Lambda}$, the noise caused by **E** will produce an estimated $\mathbf{\Lambda}$ that is non-sparse. To solve this issue, the papers of Hirose and Yamamoto (2014) and Adachi and Trendafilov (2015) proposed factor models that include restrictions or penalty terms to force $\mathbf{\Lambda}$ to obtain a certain level of sparseness. Hirose and Yamamoto do this by developing a penalized model. Their factor analysis via a non-convex penalty (FANC) can be interpreted as imposing a weak penalty on the objective function for using a less sparse $\mathbf{\Lambda}$. The model by Adachi and Trendafilov goes even further by imposing a hard restriction on the cardinality of $\mathbf{\Lambda}$. The corresponding algorithm is called sparse orthogonal factor analysis (SOFA). In a subsequent paper Adachi and Trendafilov (2018) use the SOFA model and develop it further by imposing the restriction of $\mathbf{\Lambda}$ being the sparsest as well as relaxing the assumption of no cross-correlation between common factors (orthogonality assumption). They call their revised model sparsest factor analysis (SSFA).

In the context of risk clustering of stock returns in finance, such a sparsest factor model can be fairly useful as it allows for a distinct clustering of assets into certain risk groups. Past economic shocks such as the dot-com bubble of 2000, the 2008 financial crisis and even the 2020 Covid pandemic have shown that some asset sectors tend to be hit harder than others.

As seen in Figure 1, in 2020, the majority of economic sectors in the US experienced contractions. However, it can clearly be seen that some sectors contracted harder than others and some even grew. For asset managers, one way to reduce the effect of idiosyncratic shocks is by simply increasing the number of assets included. This is, however, a strategy with diminishing returns to scale. For every additional reduction in downside risk, an asset manager must include an ever-increasing number of additional stocks. Additionally, the limited availability of robust, high-performance stocks makes it increasingly difficult to maintain a good level of returns whilst minimizing volatility. Using risk clusters to diversify asset allocations can therefore represent an effective alternative to increase overall diversification without having to inflate the number of assets. Hence, the main research question of this paper is formulated as follows:

[1] Source: https://www.bea.gov/news/2021/gross-domestic-product-third-estimate-gdp-industry-and-corporate-profits-4th-quarter-and

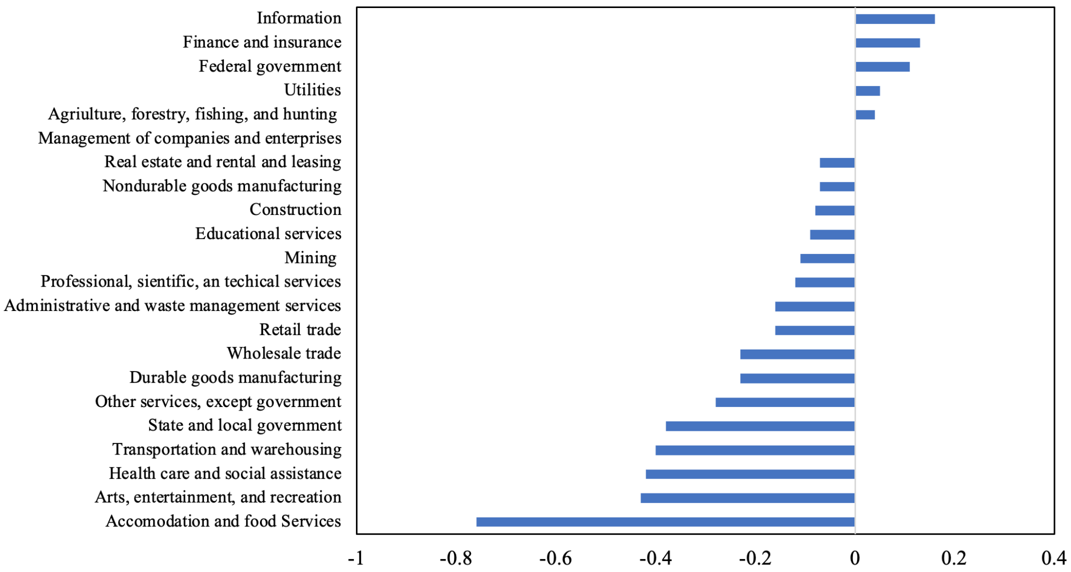


Figure 1: US 2020 contribution to GDP growth by sector, in per cent[1].

”To what extend can Sparsest Factor Models improve the return vs volatility trade-off in financial asset allocation?”

This paper answers the question above by applying the SSFA algorithm to a moving window framework for financial time series. Namely, a data set containing monthly log-returns of the world’s largest stock indices and a data set containing monthly log-returns of the constituents of the German DAX index. However, given that the SSFA algorithm was originally developed for cross-sectional data, a number of adjustments need to be addressed. In addition, Adachi and Trendafilov (2018) mention a number of shortcomings in their paper that also need to be addressed. Hence, this paper considers the following three sub-questions to answer the above research question:

1. Can SSFA recover the true clusters in (an experimental and) a real setting?

2. How can we recover the appropriate number of clusters?

Question one stems from the fact that this paper closely follows the methodology put forward by Adachi and Trendafilov (2018). With it, the feasibility of the proposed SSFA algorithm is validated by using a selection of the data-sets included in Adachi and Trendafilov. Question two deals with the first shortcoming mentioned by Adachi and Trendafilov (2018) – namely, the selection of an optimal number of factors/clusters. Given the sparsity constraint on $\mathbf{\Lambda}$, conventional model selection criteria for factor models are not appropriate, and they leave it for future research to develop appropriate model selection criteria. In the finance framework, this is highly relevant, as the appropriate number of risk clusters is often unknown. Hence, to apply the SSFA paper to asset allocation, this paper implements a revised version of the silhouette method.

Even though clustering algorithms have been around for some time, they are still an ongoing field of research, and new approaches are being put forward continuously. Initial research primarily focused on cross-sectional data, yet more recent research started to include more and more time-series data clustering algorithms. Hence, this paper’s modification of a cross-sectional clustering algorithm to time-series data adds to this trend and can be scientifically relevant to future time series clustering algorithms. The practical relevance of this clustering algorithm is laid out in the construction of a cluster-based Sharpe ratio momentum portfolio that outperforms both a market-based portfolio and a pure Sharpe Ratio portfolio. However, given clusters for a data set of asset returns, in theory, any investment strategy can easily make use of the added value the clusters provide.

All in all, SSFA can provide strong and simple visualizations of clusters when there are substantial segments present. In a more interrelated market, the visualization of clusters via SSFA might be too simplistic as it only allows for disjoint clusters. Nonetheless, even if the clusters are less segmented, SSFA still creates added value when implemented in existing portfolio strategies with respect to returns and Sharpe ratio.

This paper finds that for the stock market data included that there are indeed clusters present and that they enable the selection of the best performer from each cluster. For the DAX data, the presence of clusters is less clear. Nonetheless, the cluster-based portfolio still outperforms all benchmark portfolios. For the stock indices, the cluster portfolio yields a 0.226% increase in monthly returns compared to the market portfolio and a 0.168% monthly increase compared to the pure Sharpe ratio portfolio (2.700% and 2.016% annually, respectively). However, for this paper's data, SSFA-based cluster Sharpe ratio portfolios are not able to outperform the market in both returns and volatility. For both the stock indices and DAX constituents, the cluster-based portfolios increase the volatility compared to the market portfolio. However, an increased Sharpe ratio indicates that this is a trade-off, asset managers might be willing to do. With respect to the portfolios' downside exposure, the clustering does not seem to have a positive effect during financial crises. However, given the significant reduction in assets included, the aforementioned equivalence in performance can also be seen as a strength.

The remainder of this paper is structured as follows. First, the following section summarizes the current state of literature on factor models, sparsest factor models, appropriate cluster selection criteria and risk clusters in finance. Subsequently, the selection and preparation of the data used in the methodology are laid out. Thereafter, this paper defines the SSFA moving window structure, including the SSFA algorithm and a revised version of the Silhouette method based on a complexity invariant distance measure. Consequently, the theoretical framework lays out the three portfolio allocation strategies: an equally weighted market portfolio, a pure Sharpe ratio-based portfolio, and a cluster-based Sharpe ratio portfolio. This is followed by the empirical analysis of the data performed by firstly confirming the validity of Adachi and Trendafilov (2018)'s results, and then evaluating the moving window results for the financial data sets and discussing their implications for asset allocation diversification. Lastly, this paper answers the proposed research questions and discusses the limitations of this paper as well as areas for future research.

2 Theoretical Background

2.1 Factor Models

Ever since factor models were applied in finance they have been a popular field of study with most noteworthy contributions by authors like Rosenberg and McKibben (1973), Fama and French (1992) or Sharpe (1964). Generally speaking, factor models can be classified into three different types: macroeconomic, fundamental, and statistical factor models (Connor, 1995). Their main difference is the estimation of factors $\mathbf{F}$ as well as loadings $\mathbf{\Lambda}$ by treating them as known or unknown. Rosenberg and McKibben (1973), Fama and French (1992) and Sharpe (1964) are all part of the macroeconomic factor model class. That is, they use observed (macroeconomic) variables as factors

$\mathbf{F}$ in equation (1) and estimate the factor loadings $\mathbf{\Lambda}$. Rosenberg and McKibben for example, collect 32 factors that are partitioned into three groups: accounting-based descriptors (factors), market-based descriptors and market valuation descriptors. These factors are consequently used to predict stock returns. The main advantage of these models is a straightforward interpretation of the results. However, they require the factors to capture all pervasive risks. Although it may be a reasonable assumption that a small number of pervasive risk factors exist, in practice, their identification frequently proves inaccurate.

A solution to this is provided by the class of fundamental factor models or, more specifically, principal component factor models (Parson, 1901 and Hotelling, 1933). These models treat both the factors in $\mathbf{F}$ and their loadings $\mathbf{\Lambda}$ as unknowns. That is, they capture the main characteristics of assets by using a limited number of Principal Component (PC) with the most explanatory power. These PC's are then used as factors $\mathbf{F}$. An excellent modern reference of these models is given by Jolliffe (2002). In finance, PC models have the advantage that only asset return data is required and that there is no need for specifying factors that explicitly capture all pervasive risks as they, if significant, should be accounted for by the principal components included. One drawback of this is the interpretability of the individual factors with macroeconomic phenomena for high dimensional data.

Fundamental factor models provide the last class of factor models. These models use macroeconomic theory to identify asset attributes such as dividend yield or book-to-market ratios and uses them as factor loadings $\mathbf{\Lambda}$. This type of factor model is robust in situations where clear market forces and rules govern the behaviour of the dependent variable. However, this is barely the case in practice as financial markets are often subject to a vast number of influences. A summary of the three models with their respective inputs, estimation techniques and outputs can be found in Table 1.

Table 1: . *An overview of the empirical procedures for the three classes of factor models.*

Factor Model Type	Inputs	Estimation Technique	Outputs
Macroeconomic	Security returns and macroeconomic variables	Time-series regression	Security factor betas
Statistical	Security returns	Iterated Time-series/ Cross-sectional regression	Statistical factors and security factor betas
Fundamental	Security returns and security characteristics	Cross-sectional regression	Fundamental factors

Source: Connor (1995).

2.2 Sparse Factor Models

In order to improve the interpretability, recent research has focused on imposing restrictions on the loadings matrix $\mathbf{\Lambda}$. The idea is to impose sparsity restrictions on $\mathbf{\Lambda}$ to make it easier to interpret:

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & ... & 0 & \# & ... & \# & 0 & ... & 0 \\ 0 & ... & ... & ... & ... & 0 & \# & ... & \# \\ & & & \vdots & & & & & \\ 0 & 0 & \# & ... & \# & 0 & ... & ... & 0 \end{bmatrix}. \tag{2}$$

That is, given that only a limited number of variables in each factor are non-zero, the interpretation of each factor becomes easier. Many models have been put forward to this extend, and it is an ongoing field of research which types of restrictions on $\mathbf{\Lambda}$ are the most appropriate. However, the approaches can usually be classified into either maximum likelihood FA (MLFA) or matrix decomposition FA (MDFA).

For MLFA we start with the negative log likelihood:

$$l(\mathbf{\Lambda}, \mathbf{\Phi}, \mathbf{\Psi}) = \log|\mathbf{\Lambda}\mathbf{\Psi}\mathbf{\Lambda}' + \mathbf{\Phi}| + \text{tr}(\mathbf{S}(\mathbf{\Lambda}\mathbf{\Psi}\mathbf{\Lambda}' + \mathbf{\Phi})^{-1}), \tag{3}$$

as formulated in Adachi and Trendafilov (2018), where $\mathbf{S} = n^{-1}\mathbf{X}'\mathbf{X}$ denotes the sample covariance matrix. To find the solution this log likelihood has to be minimized over $\mathbf{\Lambda}$ and $\mathbf{\Psi}$ with $\mathbf{\Phi} = \mathbf{I}_m$.

Sparse MLFA models usually achieve an increase in sparsity of $\mathbf{\Lambda}$ by imposing a penalty on the log likelihood of equation (3) as follows:

$$\min_{\mathbf{\Lambda},\mathbf{\Psi},\mathbf{\Phi}} l(\mathbf{\Lambda}, \mathbf{\Psi}, \mathbf{\Phi}) + \rho P_\gamma(\mathbf{\Lambda}), \tag{4}$$

with $P_\gamma(\mathbf{\Lambda})$ penalizing lambda to be non-sparse and ρ and λ being tuning parameters defining the form of the penalty and ρ controlling the level of sparsity. Zou et al. (2006) and Zhang (2010) propose such a model by specifying the penalty function to take a lasso form ($P(\mathbf{\Lambda}) = |\mathbf{\Lambda}|$). However, as Takigawa et al. (2012) point out, lasso models are biased and tend to produce overly dense models. Hence, Hirose and Yamamoto (2014) propose a non-convex alternative for the penalty function. They find significant evidence that the non-convex lasso produced superior results with respect to true model recovery in a simulated setting.

For MDFA we minimize the least squares analog of equation (1) given as follows[2]:

$$f(\mathbf{F}, \mathbf{U}, \mathbf{\Lambda}, \mathbf{\Psi}) = ||\mathbf{X} - (\mathbf{F}\mathbf{\Lambda}' + \mathbf{U}\mathbf{\Psi})||^2, \tag{5}$$

over $\mathbf{F}, \mathbf{U}, \mathbf{\Lambda}$ and $\mathbf{\Psi}$, subject to the constraints:

$$n^{-1}\mathbf{F}'\mathbf{F} = \mathbf{I}_m, \qquad n^{-1}\mathbf{U}'\mathbf{U} = \mathbf{I}_p, \qquad n^{-1}\mathbf{F}'\mathbf{U} = \mathbf{O}_{mxp}, \tag{6}$$

where the first restriction refers to the mutual independence of the common factors in **F**, the second constraint refers to the mutual independence of the unique factors **U** and the third refers to the independence of the common factors **F** with the unique factors.

[2] $||.||^2$ refers to the squared norm.

Sparse MDFA achieves a sparse $\mathbf{\Lambda}$ by imposing additional constraints on the objective function. Adachi and Trendafilov (2015) do this by constraining the cardinality matrix of $\mathbf{\Lambda}$ in equation (5) to a pre-specified level:

$$\min_{\mathbf{F},\mathbf{\Lambda},\mathbf{U},\mathbf{\Psi}} f(\mathbf{F},\mathbf{\Lambda},\mathbf{U},\mathbf{\Psi}) \tag{7}$$

$$\text{subject to: } \operatorname{Card}(\mathbf{\Lambda}) = c \text{ and (6)} \tag{8}$$

The advantage of this model is that it allows for a distinct level of sparsity in the loadings $\mathbf{\Lambda}$.

In summary, sparse MLFA imposes a weak restriction on the sparsity by penalizing a less sparse $\mathbf{\Lambda}$ in the objective function, and MDFA imposes a strong restriction on the sparsity by adding additional constraints on $\mathbf{\Lambda}$. The advantage of MDFA is that it allows the cardinality of $\mathbf{\Lambda}$ to be fixed. Although it is heuristically possible to use MLFA to estimate a $\mathbf{\Lambda}$ with a given level of sparsity, it is not guaranteed that this $\mathbf{\Lambda}$ exists and given that it is a heuristic approach, it is much more computationally intensive.

2.3 Sparsest Factor Models

An even stricter restriction on the loadings matrix $\mathbf{\Lambda}$ is imposed by sparsest factor models. These models constrain the loadings to be the sparsest possible. That is the loadings have only one non-zero element per row:

$$\mathbf{\Lambda} = \begin{bmatrix} 0 & ... & 0 & \# & 0 & ... & 0 \\ 0 & ... & ... & ... & ... & 0 & \# \\ & & & \vdots & & & \\ 0 & 0 & \# & 0 & ... & ... & 0 \end{bmatrix}. \tag{9}$$

The advantage of this restriction is that the interpretation of the loadings is easiest as each factor loads a unique and distinct subset of the dependent variable $\mathbf{X}$. Vichi and Saporta (2009) implement a PCA that restricts $\mathbf{\Lambda}$ as specified above. Their model already makes the interpretation of $\mathbf{\Lambda}$ relatively easy. That is, it allows for clustering by partitioning all variables loaded by each factor into one cluster. Additionally, the estimated $\mathbf{\Lambda}$ gives an indication as to the relationship between variables within a cluster. However, it is only applicable to factor models without unique variances $\mathbf{\Psi}$. Given that this rarely occurs in practice, Adachi and Trendafilov (2018) propose a revised SOFA algorithm called Sparsest Factor Analysis (SSFA). It relaxes the crucial assumption of individual variances while restricting $\mathbf{\Lambda}$ to be sparsest. Hence, it can also be seen as an extension of Vichi and Saporta (2009) to a more complicated model allowing for individual variances $\mathbf{\Psi}$.

2.4 Cluster Selection Criteria

In their shortcomings Adachi and Trendafilov (2018) state that it remains for future research to determine the appropriateness and the number of clusters in a given data set. To this date, many algorithms have been proposed towards exactly that goal with varying degrees of success. In general, they can be categorized into three groups. Firstly, direct methods, which consist of optimizing a criterion, such as the average silhouette. Secondly, statistical testing methods which consist of comparing the cluster selection against a null hypothesis. An example of this would be the gap statistic. Thirdly, model selection criteria methods that determine the number of clusters based on, e.g. the BIC and AIC. However, given that the focus of this paper is on the quality of clustering and not the model fit, the latter is less appropriate.

The silhouette method, developed by Rousseeuw (1987), evaluates the cluster choice based on the tightness and separation of their silhouettes. For each data point i in a given cluster the silhouette value is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \text{ if } |\mathbf{C}_i| > 1 \tag{10}$$

$$s(i) = 0, \qquad \text{else,} \tag{11}$$

where $a(i)$ denotes the average distance between point i and all other points in the same cluster:

$$a(i) = \frac{1}{[\mathbf{C}_i| - 1} \sum_{j \in \mathbf{C}_i, i \neq j} d(i, j), \tag{12}$$

and $b(i)$ the smallest average distance of point i to all points in any other cluster:

$$b(i) = \min_{k \neq i} \frac{1}{|\mathbf{C}_i|} \sum_{j \in \mathbf{C}_k} d(i, j), \tag{13}$$

with $|\mathbf{C}_i|$ denoting the cardinality of cluster i and $d(i, j)$ being a distance measure such as the Euclidean distance. Consequently, the optimal number of clusters can be determined via $\hat{k} = \underset{k}{\text{argmax}}\, \tilde{s}(k)$, where $\tilde{s}(k)$ denotes the average silhouette of the clustering algorithm with k clusters.

The Gap-Statistic, developed by Tibshirani et al. (2001), uses changes in within-cluster dispersion obtained via a given clustering algorithm and compares it to the dispersion expected under a given null hypothesis:

$$\text{Gap}(k) = \frac{1}{B} \sum_b (\log(W_{kb}^*) - \log(W_k)), \tag{14}$$

where W_{kb}^* refers to the within cluster dispersion of bootstrapped data under the null hypothesis and W_k denotes the within cluster dispersion of the actual data. Adjusted for notation of the log return time-series data W_k is given as:

$$W_k = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{t=1}^{T} (r_{i,t} - \bar{r}_{k,t})^2 \tag{15}$$

where, C_k is the set of assets in the kth cluster and $r_{k,t}$ is the jth variable in the cluster mean of the kth cluster.

An extensive summary of the Gap statistic algorithm has been included in Appendix 6, algorithm 6. Tibshirani et al. (2001) find that the main advantage of the gap statistic over the silhouette method is that it includes the single-cluster case and thus also evaluates the appropriateness of clustering in a given data set. However, given the high likeliness of clustering for financial data sets this advantage does not provide any added value to this paper but could be used in further research to find a generalized approach to the number of cluster selection in time series data.

Most research done on classification problems and algorithms has been done on cross-sectional data. This is also the case for the silhouette method as well as the gap statistic. Given that both methods use a distance measure as input, conventional distances such as the Euclidean distance might not be sufficient for time-series data (Ding et al., 2008). Hence, more recent research on clustering algorithms for time series has shifted its focus towards revised distance measures that are able to capture the more complex nature of time series data. Amongst them recursive Dynamic Time Warping (DTW) as put forward by Müller (2007). Via the following recursive relationship, the one-to-one date relationship between two time-series is relaxed:

$$\text{DTW}(i,j) = d(i,j) + \min \left\{ \begin{array}{c} \text{DTW}(i-1,j) \\ \text{DTW}(i,j-1) \\ \text{DTW}(i-1,j-1) \end{array} \right\}, \tag{16}$$

where $d(i,j)$ usually denotes the Euclidean distance and $\text{DTW}(i,j)$ is initialized at $\text{DTW}(0,0) = 0$. For each point in a given time series, it aims to find the most appropriate point in the other time series over a specified interval. A graphical comparison of DTW against the Euclidean distance is given in Figure 2. The main advantage of DTW is that it allows accounting for lagged clusters. A common example used in literature is two persons walking at different speeds. Laying both person's movements on top of each other in a linear fashion and taking the Euclidean distances between key points of their bodies over time would yield very high distances and thus lead to the false conclusion that both persons perform inherently different movements. DTW, however, is able to 'stretch' one of the time series in such a way that the speed of one person is slowed down in a way that minimizes the distance between key body points. As a result, it would be indeed concluded that the two persons perform the same movements, although at different speeds. Given that SSFA in its current form does not look at lagged clusters, this approach might be ill-advised. In addition, it can be argued that any lagged behaviour of an asset group, let us call it group A, to another, let us call it group B, can be regarded as a distinct cluster by itself as the reaction of group A is to the change in group B and not to the underlying signal that originally affected the assets in group B.

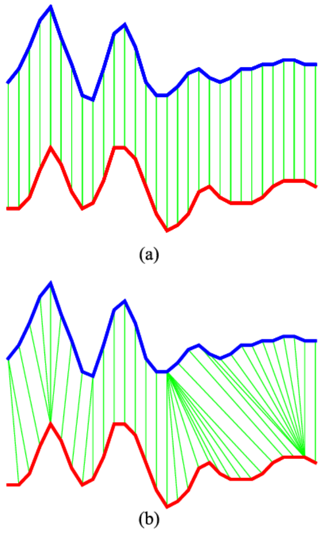


Figure 2: Graphs of (a) Euclidean distance matching and (b) DTW matching[3].

Another, more recent approach has been put forward by Batista et al. (2014) called Complexity-Invariant Distance (CID). It uses differences in the complexity of two time series and scales them accordingly:

$$\text{CID}(i,j) = d(i,j) * \frac{\max(ce(i), ce(j))}{\min(ce(i), ce(j))}, \tag{17}$$

where $d(i,j)$, again, usually denotes the Euclidean distance between two time series and $ce(i)$ denotes the complexity estimate of a log-return time series i given by: $ce(i) = \sqrt{\sum_{t=1}^{T-1} (r_{i,t} - r_{i,t+1})^2}$. That is, $ce(i)$ approximates a time series' complexity by stretching it to a line and taking its length (see Figure 3). This complexity measure is then applied as a scaling factor to the Euclidean distance measure as in equation (17) or any other distance measure. As a result CID is able to account for different levels of complexity, or in the financial context volatility. Most importantly, Batista et al. found that, in a simulated time series data setting on average, CID outperformed both the Euclidean distance as well as the the DTW distance.

[3] Source: Tan et al. (2017)

[4] Source: Batista et al. (2014)

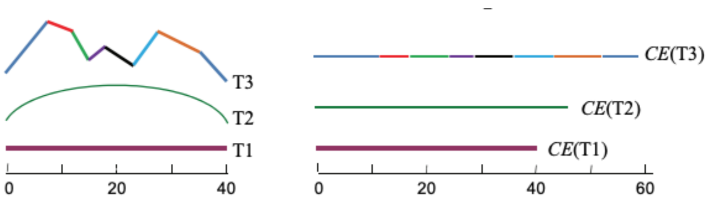


Figure 3: Graphical representation of stretching a time series to a line[4].

2.5 Risk Clusters in Finance

It is well known that there exist risk clusters in the finance world. Examples of such clusters crashing would be the dot-com bubble in 2000 which primarily affected tech companies, the 2008 financial crisis, which, although having a global effect, concentrated on the finance industry. The most recent example of a risk cluster crashing would be the 2020 Covid pandemic, in which e-commerce companies thrived whilst transportation companies have suffered heavy losses, often relying on government bailouts. However, not only the crashes tend to be sector dependent; the recovery is often also sector dependent. Looking at Figure 4 it can be seen that in the two years after the 2008 financial crisis, the speed of recovery is sector dependent. While the manufacturing sector was hit the hardest, it also took longer to recover as, e.g. compared to the financial sector, which was the original origin of the crash.

In finance, the mathematical formulation of the process of diversification through risk spreading is usually referred to as the Markowitz-minimum variance portfolio (Markowitz, 1959). It makes use of the fact that asset prices tend to move up and down together. However, as Markowitz already pointed out, this relationship is not perfect, and there are always a number of assets that violate the general trend. Looking at Figure 4 again, this suspicion is confirmed by the poor post-crisis performance of the manufacturing sector. Hence, it can be useful to cluster stocks to gain insights on which ones observe collective behaviour, thus reducing one's positions' exposure to shocks and effectively being able to pick a few good performers that do not observe collective behaviour.

Farrell (1974) is one of the first authors that has done this in the context of portfolio allocation. He makes use of cross-asset correlations to find sets of stocks that are more highly homogeneously correlated within their set as compared to stocks from other sets. Papenbrock (2011) gives a more recent application of clusters in asset allocation. He applies hierarchical clustering algorithms such as single-linkage, average-linkage, complete-linkage and Ward's method to equity and credit portfolios by assigning heavier weights to assets that are higher in the hierarchical cluster structure and vice versa less for the ones that are lower. He finds that the annual return increased significantly for both, yet the cluster portfolios did not outperform the market portfolio in terms of volatility. However, given a significantly higher Sharpe ratio, this trade-off can be seen as justified.

3 Data

There are four sets of data that this paper uses for the analysis. The first two are from the paper Adachi and Trendafilov (2018) in order to validate their findings, and the latter two are financial data sets in order to answer the main research question.

[5] Source: https://izajolp.springeropen.com/articles/10.1186/2193-9004-1-3/figures/6

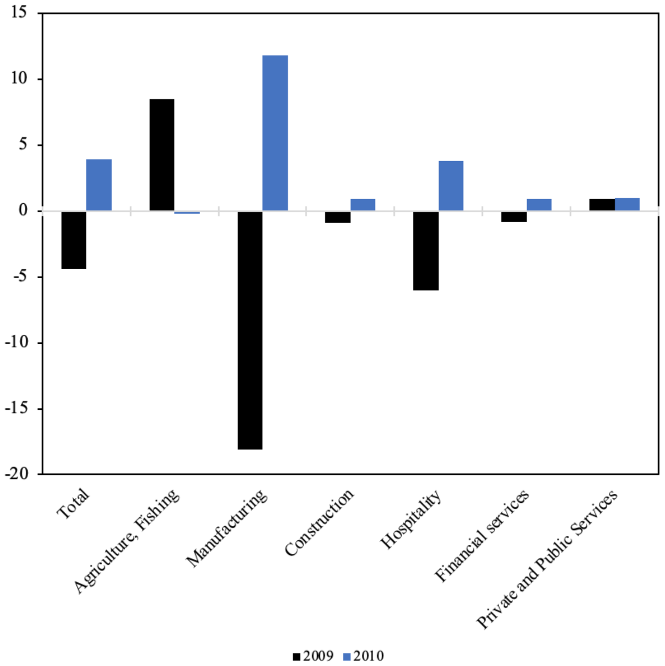


Figure 4: Post 2008 development of German industry sectors in per cent[5].

3.1 Adachi and Trendafilov (2018) Data Sets

The first data set contains the simulated data from Adachi and Trendafilov (2018). This data set is used to validate the correctness of this paper's implementation of the SSFA algorithm. The data generating variables are provided in Appendix B, Table 10 for reference. The second data set is taken from the "Big Five Personality Test Data" from http://bstat.jp/en_material/. It contains a $25 * 25$ correlation matrix obtained from 190 participants. In the test, the participants are asked to rate their personality based on 25 different given items. This data set is used to validate the findings of Adachi and Trendafilov (2018).

3.2 Financial Data Sets

As mentioned above this paper uses two financial data sets. The first one is a data set of monthly price data of 17 global stock market indices index over the sample period 01/2000 - 12/2020. The data contains the end-of-month adjusted close value of the indices and was manually retrieved from Yahoo Finance. As is common practice the prices are first transformed into log return data:

$$r_{i,t} = \log\left(\frac{P_{i,t}}{P_{i,t-1}}\right) * 100, \tag{18}$$

where $P_{i,t}$ denotes the adjusted closing price of asset i at the end of month t. The aim is to classify the indices into different risk groups based on their observed volatility and possibly find similarities of clusters within each cluster (e.g. geography). Table 2 provides a summary of the indices included.

Table 2: *Statistical summary of stock indices log-return data statistics.*

Name	Ticker	Mean	Std. Dev.	Min.	Max.	Skewness	Kurtosis	DF-Test
S&P 500	GSPC	0.395	4.417	-18.564	11.942	-0.711	4.456	-14.516**
Dow Jones	DJI	0.410	4.304	-15.153	11.187	-0.684	4.401	-15.337**
NASDAQ	IXIC	0.472	6.506	-26.009	17.559	-0.796	4.900	-14.658**
NYSE	XAX	0.402	5.374	-36.483	23.277	-1.296	12.605	-16.698**
SSE	000001S	0.282	7.449	-28.278	24.253	-0.531	5.082	-16.698**
Shenzen Composite	399001SZ	0.476	8.373	-29.619	23.979	-0.323	4.147	-13.704**
S&P/ASX	AXJO	0.298	3.974	-23.803	9.492	-1.389	8.148	-14.714**
All Ordinaries	AORD	0.313	4.030	-24.225	9.465	-1.482	8.584	-14.362**
DAX 30	GDAXI	0.278	6.140	-29.333	19.374	-0.898	5.879	-14.874**
CAC 40	FCHI	-0.055	5.212	-19.225	18.331	-0.557	4.492	-14.666**
EURONEXT 100	N100	0.030	4.934	-18.936	15.660	-0.773	4.789	-14.223**
BEL 20	BFX	0.110	5.017	-24.088	18.646	-1.108	6.936	-13.198**
Nikkei 225	N225	0.130	5.663	-27.216	14.014	-0.717	4.429	-13.935**
Hang Seng	HSI	0.199	5.979	-25.446	15.763	-0.585	4.195	-14.445**
KOSPI	KS11	0.510	6.228	-26.311	20.254	-0.462	4.657	-15.068**
S&P/TSX	GSPTSE	0.287	4.166	-19.523	10.625	-1.221	6.991	-13.503**
IBOVESPA	BVSP	0.791	7.356	-35.531	16.481	-0.780	5.354	-13.840**

Note. * $p < 0.05$, ** $p < 0.01$

From the Table, it follows that the indices adhere to the second and third stylized fact of stock return data (Cont, 2001). That is, the indices observe heavy tails given that all Kurtosis values are above three, and the indices observe gain/loss asymmetry with strictly negative Skewness values, indicating more extreme drawdown effects and more frequent moderate positive returns. Moreover, all indices do not observe significant autocorrelations (see Appendix B, Table 11). Thus they also satisfy the first stylized fact on stock returns of Cont. Finally, an augmented Dickey-Fuller test indicates that all log return series are stationary.

The second data set contains monthly price data of constituents in the German performance DAX 30 index over the sample period 04/2004 - 12/2019. The data contains the end-of-month adjusted close value of the company tickers and was manually retrieved from Yahoo Finance. Note that the actual number of companies included is only 27 as the company Metro AG does not have a sufficiently complete data series over the included sample range, MAN was excluded as the Volkswagen Group acquired it in 2012, and Linde PLC was excluded for its radical price jumps around 2004. Again, the prices were converted into log returns as laid out in equation (18). Table 3 provides a summary of the companies included.

Table 3: *Statistical summary of Dax 30 constituents log-return data.*

Name	Ticker	Mean	Std. Dev.	Min.	Max.	Skewness	Kurtosis	DF-Test
Fresenius Medical Care	FME.DE	0.750	5.895	-27.442	15.157	-0.946	3.269	-15.667**
Fresenius AG	FRE.DE	1.164	6.361	-23.186	16.219	-0.521	0.850	-12.800**
Beiersdorf	BEI.DE	0.834	5.338	-18.970	14.320	-0.289	0.534	-13.959**
Merck KGAA	MRK.DE	1.079	6.667	-17.713	20.435	0.028	0.062	-14.013**
Volkswagen	VOW3.DE	1.245	11.498	-59.611	30.565	-1.555	6.416	-12.545**
HeidelbergCement	HEI.DE	0.490	9.589	-44.388	29.013	-0.847	3.127	-12.043**
K+S	SDF.DE	0.527	11.231	-53.198	27.215	-1.156	4.120	-12.362**
Deutsche Telekom	DTE.DE	0.418	5.990	-20.781	16.399	-0.232	0.410	-14.863**
E.ON	EOAN.DE	0.157	7.570	-27.480	22.322	-0.664	1.611	-14.031**
RWE	RWE.DE	0.228	8.852	-34.692	22.058	-0.621	1.921	-13.896**
Infineon Technologies	IFX.DE	0.435	14.437	-65.601	83.988	0.415	8.769	-9.842**
SAP	SAP.DE	0.849	6.147	-31.087	21.542	-0.574	3.466	-13.677**
Munich RE	MUV2.DE	0.897	5.089	-13.785	13.082	-0.339	-0.015	-15.686**
Commerzbank	CBK.DE	-1.564	13.202	-65.842	36.524	-0.959	4.440	-11.353**
Allianz	ALV.DE	0.712	7.788	-50.647	16.514	-1.888	9.137	-14.007**
Deutsche Bank	DBK.DE	-0.848	10.566	-52.033	37.716	-0.589	3.368	-12.283**
Deutsche Post	DPW.DE	0.676	8.145	-53.572	26.636	-1.740	9.710	-14.582**
Adidas	ADS.DE	1.464	7.190	-31.438	19.101	-0.695	1.886	-13.381**
Deutsche Börse	DB1.DE	0.709	9.374	-72.181	22.105	-2.731	18.540	-12.953**
Henkel	HEN3.DE	0.875	6.002	-22.947	14.893	-0.604	1.311	-13.362**
Bayer	BAYN.DE	0.855	6.773	-21.989	15.597	-0.576	0.440	-14.226**
BMW	BMW.DE	0.629	7.874	-30.505	23.375	-0.355	1.553	-14.154**
Daimler	DAI.DE	0.495	9.183	-29.627	35.255	-0.061	1.558	-13.722**
Basf	BAS.DE	0.924	7.138	-25.819	22.604	-0.546	1.692	-12.338**
Siemens	SIE.DE	0.591	7.090	-34.727	17.271	-1.018	3.477	-13.643**
ThyssenKrupp	TKA.DE	0.008	10.245	-48.657	20.954	-0.940	2.472	-12.321**
Lufthansa	LHA.DE	0.302	8.793	-23.283	19.685	-0.322	-0.220	-13.380**

Note. * $p < 0.05$, ** $p < 0.01$

Similarly to the stock indices, the DAX stocks also observe negative Skewness. Hence the assumption of gain/loss asymmetry holds. However, the majority of stocks included do have flat tails indicating that the German market is most likely more stable than stocks usually are. Note also that the standard deviations are more spread as compared to the stock indices, indicating a less homogeneous market. Again, the autocorrelations are insignificant for the log-returns (see Appendix B, Table 12 and 13). Finally, again an augmented Dickey-Fuller test indicates that all DAX log return assets are stationary. Thus, the DAX 30 can be seen as a less volatile version of the stock market indices that does not necessarily satisfy all stylized stock return facts. Nonetheless, it might be interesting to evaluate the clusters obtained in such a market.

4 Methodology

The methodology will follow that of Adachi and Trendafilov (2018) closely. Whenever this paper deviates from it, it will state so explicitly. Given that Adachi and Trendafilov already established the SSFA's superiority over SOFA and FANC with respect to the sparsest factor model estimation, these two are excluded from the methodology in order to focus solely on the SSFA algorithm. Additionally, the methodology will address Adachi and Trendafilov's shortcoming with respect to the optimal number of clusters by including a revised version of the silhouette method for time series data.

4.1 Sparsest Factor Analysis (SSFA)

As previously mentioned, the SSFA algorithm by Adachi and Trendafilov (2018) is an extension of the SOFA algorithm. That is, it imposes a sparsest constraint on $\mathbf{\Lambda}$ in equation (1). Moreover, it relaxes the cross-factor independence assumption using a **QR** decomposition of the factors $\mathbf{F} = \mathbf{QR}$ where $\mathbf{R}$ is an upper triangular matrix with $\text{diag}(\mathbf{R}'\mathbf{R}) = \mathbf{I}_m$. For corresponding proofs validating the algorithm, I refer to the paper of Adachi and Trendafilov. Algorithm 1 summarizes all steps involved in the estimation of the SSFA $\mathbf{\Lambda}$.

Algorithm 1: Sparsest Factor Analysis (SSFA)

```
Result: $\mathbf{\Lambda}$, $\mathbf{\Psi}$ and $\mathbf{R}$;
[0]: Initialize $\mathbf{\Lambda}$, $\mathbf{\Psi}$, $\mathbf{R}$, $l = 1$ and $f_0(\mathbf{\Theta}_l) = 1$;
while $f_{k-1}(\mathbf{\Theta}_l) - f_k(\mathbf{\Theta}_l) \leq 0.1^5$ do
    [1] EVD: $\mathbf{B}'\mathbf{S}\mathbf{B} = \mathbf{L}_1\mathbf{\Delta}_1^2\mathbf{L}_1'$;
    [2] Update: $\mathbf{\Psi} = \text{diag}(\mathbf{B}'^{+}\mathbf{L}_1\mathbf{\Delta}_1\mathbf{L}_1'\mathbf{H}^p)$;
    [3] Update: $\mathbf{Y} = \mathbf{B}'^{+}\mathbf{L}_1\mathbf{\Delta}_1\mathbf{L}_1'\mathbf{H}_m$;
    [4] Update columns of R: $r_{j,1} = \frac{(\mathbf{Y}'\mathbf{\Lambda})_{j,1}}{||(\mathbf{Y}'\mathbf{\Lambda})_{j,1}||}$;
    [5] Update: $\mathbf{\Lambda}$ such that $\lambda_{i,j} = \begin{cases} y_i'r_j & iff \quad j = J(i) \\ 0 & else \end{cases}$;
    if $\mathbf{\Lambda}$ has empty column(s) then
        [6] Restart with different $\mathbf{\Lambda}$ at [0];
    end
    [7] Compute: $f(\mathbf{\Theta}_l) = 1 - \frac{\text{tr}(\mathbf{\Lambda}\mathbf{\Lambda}') + \text{tr}(\mathbf{\Psi}^2)}{\text{tr}(\mathbf{S})}$;
    if $f(\mathbf{\Theta}_{l-1}) - f(\mathbf{\Theta}_l) \leq 0.1^5$ then
        return current $\mathbf{\Lambda}$, $\mathbf{\Psi}$, $\mathbf{R}$ and $f_k(\mathbf{\Theta}_l)$;
    else
        [8] Set $k = k + 1$;
    end
end
```

In it, $\mathbf{S}$ denotes the estimated sample covariance matrix of the underlying data. $\mathbf{L}_1$ is a $([p+m] * m)$ matrix containing the first $p+m$ eigenvectors of $\mathbf{B}'\mathbf{S}\mathbf{B}$, corresponding to the largest $p+m$ eigenvalues, respectively. $\mathbf{\Delta}_1^2$ is a diagonal $(p * p)$ matrix containing the first p largest (positive) eigenvalues of $\mathbf{B}'\mathbf{S}\mathbf{B}$. $\mathbf{B}'^{+} = \mathbf{B}'(\mathbf{B}\mathbf{B}')^{-1}$ is the Moore-Penrose inverse of $\mathbf{B}$, $\mathbf{H}^p = [\mathbf{B}_{pxm}, \mathbf{I}_p]'$ and $\mathbf{H}_m = [\mathbf{I}_m, \mathbf{O}_{mxp}]'$. $\mathbf{R}$ denotes the upper triangular matrix relaxing the orthogonality constraint and $J(i) = \underset{1 \leq j \leq m}{\text{argmin}}\, g_{i,j}(y_i'r_j)$, where $g_{i,j}(\lambda_{i,j}) = \lambda_{i,j}^2 - 2(y_i'r_j)\lambda_{i,j}$ denotes the column index of a given row i that minimizes the loss function subject to the sparsity constraint. In other words, it is the column index for a given row in λ that maximises the variance explained, given the sparsest constraint. $\mathbf{\Lambda}$ is initialized as a random sparsest matrix where each column has at least three non-zero elements (if possible) and each non-zero element (i, j) is either drawn from $\lambda_{i,j} \sim \mathbf{U}(0.5, 0.98)$ or $\lambda_{i,j} \sim \mathbf{U}(-0.98, -0.5)$ with equal probability. $\mathbf{R}$ is initialized as identity matrix $\mathbf{I}_m$ with m diagonal elements and $\mathbf{\Psi}$ is initialized as $\text{diag}(\mathbf{I}_p - \mathbf{\Lambda}\mathbf{\Lambda}')^{1/2}$.

4.2 Multi Run Procedure

Given that the SSFA algorithm alternatively updates the variables, it is not guaranteed to converge to the global minimum. To avoid treating a local minimum falsely as a global minimum, Adachi and Trendafilov (2018) introduce a multi-run procedure that selects the best solution over a given number of re-runs. The main steps of their algorithm are summarized in algorithm 2.

Algorithm 2: Multi Run Procedure

```
Result: $\mathbf{\Lambda}_{l^*}$, $\mathbf{\Psi}_{l^*}$ and $\mathbf{R}_{l^*}$;
[0] Set $L = 50$;
for $l = 1, ..., L$ do
    [1] Obtain: $\mathbf{\Lambda}_l$, $\mathbf{\Psi}_l$, $\mathbf{R}_l$ and $f(\mathbf{\Theta}_l)$ from SSFA;
end
[2] Obtain: $l^* = \underset{1 \leq l \leq L}{\text{argmin}}\, f(\mathbf{\Theta}_l)$;
for $l^\# = 1, ..., L$ do
    [3] Obtain: $\Delta(\mathbf{\Theta}_{l^*}, \mathbf{\Theta}_{l^\#}) = p^{-1} \sum_i |\lambda_{i,\#}^{[l]} - \lambda_{i,\#}^{[l^*]}| + p^{-1} \sum_i |\psi_{i,\#}^{[l]\,2} - \psi_{i,\#}^{[l^*]2}| + M^{-1} \sum_{j<k} |\phi_{j,k}^{[l]} - \phi_{j,k}^{[l^*]}|$;
    if ($l \neq l^*$ and $\Delta(\mathbf{\Theta}_{l^*}, \mathbf{\Theta}_{l^\#}) < 3 * 0.1^3$) or $L = 200$ then
        [4] return $\mathbf{\Lambda}_{l^*}$, $\mathbf{\Psi}_{l^*}$, $\mathbf{R}_{l^*}$ and $f(\mathbf{\Theta}_{l^*})$;
    else
        if $l^\# = 50$ then
            [5] Set $L = L + 1$;
            [6] Obtain: $\mathbf{\Lambda}_L$, $\mathbf{\Psi}_L$, $\mathbf{R}_L$ and $f(\mathbf{\Theta}_L)$ from SSFA;
            if $f(\mathbf{\Theta}_L) < f(\mathbf{\Theta}_{l^*})$ then
                [7] Set: $\mathbf{\Theta}_{l^*} = \mathbf{\Theta}_L$;
            else
        else
    end
end
```

Algorithm 3: Silhouette method.

```
Result: $\hat{k}$;
[0] Cluster data into $k = 1, ..., K$ clusters $\mathbf{C}_i$;
[1] Calculate $d(i, j)$ for $i, j \in \{1, ..., T\}$ and $i \neq j$;
for $i \in \boldsymbol{C}_i$ do
    [2] Obtain $a(i) = \frac{1}{|\mathbf{C}_i| - 1} \sum_{j \in \mathbf{C}_i, i \neq j} d(i, j)$;
    [3] Obtain $b(i) = \min_{k \neq i} \frac{1}{|\mathbf{C}_k|} \sum_{j \in \mathbf{C}_k} d(i, j)$;
    [4] Obtain $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$ if $|\mathbf{C}_i| > 1$, else $s(i) = 0$;
end
[5] Obtain $\hat{k} = \underset{k}{\text{argmin}}\, \tilde{s}(k)$;
```

In it, $\lambda_{i,\#}^{[l]}$ and $\lambda_{i,\#}^{[l^*]}$ denote the non-zero element of the ith row in $\mathbf{\Lambda}$ of the current optimal $\mathbf{\Lambda}$ and the alternative $\mathbf{\Lambda}^*$ respectively. $\psi_{i,\#}^{[l]\,2}$ and $\psi_{i,\#}^{[l^*]2}$ denote the ith diagonal element of the current optimal $\mathbf{\Psi}^2$ and the alternative

$\mathbf{\Psi}^{2*}$, respectively, and $\phi_{j,k}^{[l]}$ and $\phi_{j,k}^{[l^*]}$ denote the elements in row j and column k of the current optimal $\mathbf{\Phi}$ and the alternative $\mathbf{\Phi}^*$, respectively. Finally, $M = m(m-1)/2$.

4.3 Determining the Optimal Number of Clusters

As mentioned in section 2, Tibshirani et al. (2001) found that given the assumption that data is clustered, the gap-statistic and silhouette method perform equally well. Hence, this paper opted to use the silhouette method as it is computationally more efficient than the Gap Statistic. Given that the method was originally developed for cross-sectional data, it needs to be adjusted in order to allow for time-series data to be used. In the following, this paper lays out the implementation of the silhouette algorithm and the adjustments that are made to make it appropriate for time series data.

Algorithm 3 summarizes the main steps involved in the Silhouette method. In it, $d(i,j)$ denotes the distance between a pair of time series i and j and $|\mathbf{C}_i|$ denotes the cardinality of cluster i. $a(i)$ can be interpreted as the average distance between a given time series i and all the other time series in its cluster. $b(i)$ can be interpreted as the minimum average distance of a given time series i.

In general, the silhouette method can be used for any type of clustering algorithm and distance. Given that this paper looks at the clusters obtained via the SSFA algorithm, it seems most appropriate to make use of the clusters obtained by them. The next specification that needs to be made is the measure of distance that will be used. As mentioned in section 2, the simple Euclidean distance might not be sufficient to capture the dissimilarities of separate time series. Hence this paper makes use of the CID measure as put forward by Batista et al. (2014). Their method can be summarized to the following algorithm:

```
Algorithm 4: Complexity Invariance Distance Measure
  Result: D
  for $i \in \{1, ...N\}$ and $j \in \{1, ...N\}$ do
      [1] Obtain $CE(i) = \sqrt{\sum_{t=1}^{T-1}(r_{i,t} - r_{i,t-1})^2}$;
      [2] Obtain $CE(j) = \sqrt{\sum_{t=1}^{T-1}(r_{j,t} - r_{j,t-1})^2}$;
      [3] Obtain $\mathbf{D}(i,j) = d(i,j) * \frac{\max(CE(i),CE(j))}{\min(CE(i);CE(j))}$;
  end
```

In it $d(i,j)$ denotes the Euclidean distance between time series i and j :

$$d(i,j) = \sqrt{\sum_{t=1}^{T}(r_{i,t} - r_{j,t})^2}, \tag{19}$$

and N denotes the total number of time series included in the data. The obtained $\mathbf{D}$ can be interpreted as a matrix of Euclidean distances adjusted for different levels of complexity or, in this case, volatility. That is, if a time series observes relatively higher volatility, its distance will be discounted more heavily and vice versa for relatively less volatile time series. Nonetheless, for reference, this paper also includes the Euclidean distance to have a comparison to the CID measure. Given that SSFA also groups inversely related variables into the same group by assigning different signs to their loadings, CID would have to be adjusted to also account for this. However, it can be argued that given the pro-cyclical nature and stationarity of log returns, this aspect can be disregarded for the given data sets. The property of pro-cyclical behaviour will be estimated with the help of the well known β estimate defined as follows:

$$\beta(r_i) = \frac{Cov(r_i, r_M)}{Var(r_i)} \tag{20}$$

where r_i denotes log-returns of a given asset i and the r_M the log-returns of its market defined as an equally weighted portfolio of all assets in a given data set. Thus, a positive $\beta(r_i)$ can be interpreted as a pro-cyclical asset and vice versa a negative as a counter-cyclical. The stationarity of all assets in both data sets has already been established in section 3 with the help of the Dickey-Fuller test. Hence, positively correlated but diverging time series, which CID would also classify as distant even though they are clearly clustered, should not be present in the log return data.

4.4 Moving Window Framework

Given that financial markets are dynamic and, in the long run, risk clusters change. Simply applying SSFA over the full sample period is too simplistic. Hence, this paper uses a moving window structure with size 48, thus including the data on the last four years. Additionally, this paper assumed that the asset allocations will be reevaluated on a semi-annual basis, as is common practice for, e.g. exchange-traded funds (ETF). At any given investment date, the $\mathbf{\Lambda}$ loadings with the number of factors set from two up to the number of variables included, divided by three will be calculated[6] via SSFA as described in section 4.2. Then, as described in Section 4.3, the Silhouette method will be applied to estimate the appropriate number of clusters for that given period. Given the dynamically changing structure of and the number of clusters, a simple allocation of clusters via their column index in $\mathbf{\Lambda}$ will not be sufficient. Therefore a set of rules is needed that is able to account for new clusters forming and existing ones disappearing or partitioning into sub-clusters. Algorithm 5 defines a procedure that adheres to those requirements. In it $\mathbf{C}_t$ denotes all clusters obtained at time t, K_t the number of clusters obtained at t and T denotes the total number of dates included in the moving window structure. Essentially the algorithm assigns k clusters $a.i, b.i, ...k.i$ at the first investment date, and at each investment date: keeps the same label if the cluster does not change, increases the roman number $(i, ii, iii, ...)$ by one if the cluster only changes slightly and uses a new alphabetical letter for a new cluster. Given the limited number of assets included and investment dates for the data sets, this algorithm is manually implemented in excel.

After the labelling of dynamic clusters, this paper applies macroeconomic theory to make educated guesses about the origins of these clusters, as well as try to define some key characteristics in terms of, e.g. average returns or volatility that differentiate them from the average.

Having obtained appropriate clusters for each time period, appropriate asset allocations for the two data sets can be constructed. For both data sets, a simplified version of the Sharpe ratio defined as the following is used:

$$SP(i,t) = \frac{E[r_{i,t}]}{\sigma_{i,t}}, \tag{21}$$

where $E[r_{i,t}]$ denotes the expected return, of asset i at time t given as its average return over the current estimation window and $\sigma_{i,t}$ its standard deviation over the current estimation window. For each investment date the Sharpe

[6] This is done, as the initialization of $\mathbf{\Lambda}$ is defined to have at least three non-zero elements per column, and can therefore only have a maximum number of columns equal to the number of variables included divided by three .

Algorithm 5: Cluster Classification Labeling

```
Result: Dynamic cluster labels
[0] Initialize clusters labels $\mathbf{C}_1$ with $t = 1$ $a.i, b.i, ..., n.i$ ;
for $t \in 2, ..., T$ do
    for $c \in 1, ..., K_t$ do
        if $c$ matches to any $\boldsymbol{C}_{t-1}, \boldsymbol{C}_{t-2}, ...\boldsymbol{C}_1$ then
            [1.1] Set cluster no. $c$ label to that match ;
        end
        if $\boldsymbol{C}_{t-1}$ is partitioned into sub-clusters then
            [1.2] Largest sub-cluster at $t$ keeps letter and adds one unit to roman digit ;
            [1.3] All sub-clusters at $t$ get a new incremental letter assignment and the roman digit i;
        end
    end
end
```

ratio of all indices for the current window is calculated and then from each cluster the best performing index with the highest Sharpe ratio is selected with all equal weights. After the weights are obtained the portfolios performance over the next half year is estimated by reporting its monthly returns.

The obtained portfolio returns will be evaluated against the equally weighted market portfolio in terms of average returns, standard deviation, Sharpe ratio and VaR to estimate the portfolios downside exposure. Here, VaR is defined as the fifth percentile of the realized historical returns. In addition, a pure Sharpe ratio portfolio strategy will be included to check if possible differences in the cluster-based portfolio compared to the market portfolio are not simply caused by picking stocks with the best Sharpe ratios. This will be done by including the x best stocks based on the highest Sharpe ratio during a given window, where x denotes the average number of stocks used in the cluster-based portfolios over the entire sample period. In order to be as close as possible to the cluster-based Sharpe ratio portfolio, x is defined as the average number of clusters obtained via the SSFA clustering algorithm.

5 Results

This section firstly validates the results of Adachi and Trendafilov (2018). Consequently, it applies the aforementioned methodology to the two financial data sets of stock indices and DAX 30 log returns.

5.1 Results - Adachi and Trendafilov

When performing the SSFA algorithm on the three simulated data sets following the data generation process as laid out by Adachi and Trendafilov, I obtained the three-factor loadings $\mathbf{\Lambda}$, as well as individual variances $\mathbf{\Psi}$ and cross factor correlation matrices $\mathbf{\Phi}$ as seen in Table 4. Looking at the lambdas of the data generating process in Appendix B, Table 10 it can be clearly seen that the SSFA is able to recover the true data generating $\mathbf{\Lambda}$, $\mathbf{\Psi}$ and $\mathbf{\Phi}$. Note that the data in Table 4 has been adjusted accordingly, given that the SSFA algorithm is neither sign robust nor column robust with respect to the obtained $\mathbf{\Lambda}$ and $\mathbf{\Phi}$. This, however, is only of minor importance as both do

Table 4: . *Solutions for the three simulated data sets.*

Variable	(A) m = 3				(B) m = 4					(C) m = 5					
	$\mathbf{\Lambda}$			$\mathbf{\Psi}^2$	$\mathbf{\Lambda}$				$\mathbf{\Psi}^2$	$\mathbf{\Lambda}$					$\boldsymbol{\psi}^2$
1	0.89	.	.	0.19	0.92	.	.	.	0.19	0.91	.	.	.	.	0.19
2	-0.83	.	.	0.36	-0.82	.	.	.	0.36	-0.83	.	.	.	.	0.33
3	0.67	.	.	0.51	0.68	.	.	.	0.52	0.73	.	.	.	.	0.50
4	-0.63	.	.	0.62	-0.61	.	.	.	0.60	-0.59	.	.	.	.	0.66
5	0.49	.	.	0.75	0.52	.	.	.	0.72	0.51	.	.	.	.	0.80
6	-0.41	.	.	0.87	-0.39	.	.	.	0.80	-0.39	.	.	.	.	0.88
7	.	0.77	.	0.35	.	0.82	.	.	0.34	.	0.79	.	.	.	0.35
8	.	-0.67	.	0.50	.	-0.69	.	.	0.49	.	-0.70	.	.	.	0.50
9	.	0.59	.	0.60	.	0.59	.	.	0.66	.	0.58	.	.	.	0.65
10	.	-0.47	.	0.71	.	-0.53	.	.	0.78	.	-0.50	.	.	.	0.76
11	.	0.37	.	0.90	.	0.39	.	.	0.82	.	0.38	.	.	.	0.89
12	.	.	-0.71	0.50	.	.	0.70	.	0.57	.	.	0.69	.	.	0.49
13	.	.	0.58	0.66	.	.	-0.59	.	0.62	.	.	-0.59	.	.	0.62
14	.	.	-0.49	0.75	.	.	0.51	.	0.73	.	.	0.51	.	.	0.79
15	.	.	0.38	0.80	.	.	-0.39	.	0.87	.	.	-0.38	.	.	0.83
16					.	.	.	0.78	0.33	.	.	.	0.82	.	0.34
17					.	.	.	-0.66	0.54	.	.	.	-0.68	.	0.52
18					.	.	.	0.57	0.65	.	.	.	0.60	.	0.64
19					.	.	.	-0.53	0.73	.	.	.	-0.49	.	0.77
20					.	.	.	0.40	0.84	.	.	.	0.45	.	0.87
21										.	.	.	.	0.69	0.47
22										.	.	.	.	-0.58	0.65
23										.	.	.	.	0.46	0.76
24										.	.	.	.	-0.36	0.84
	$\mathbf{\Phi}$				$\mathbf{\Phi}$					$\mathbf{\Phi}$					
Factor 1	1.00	0.44	0.29		1.00	0.41	0.29	-0.22		1.00	0.40	0.28	-0.23	0.30	
Factor 2	0.44	1.00	-0.41		0.41	1.00	-0.40	0.33		0.40	1.00	-0.43	0.33	-0.39	
Factor 3	0.29	-0.41	1.00		0.29	-0.40	1.00	-0.28		0.28	-0.43	1.00	-0.34	0.23	
Factor 4					-0.22	0.33	-0.28	1.00		-0.23	0.33	-0.34	1.00	-0.33	
Factor 5										0.30	-0.39	0.23	-0.33	1.00	

Note. The sign of factors and their correlations were adjusted accordingly to be in line with Adachi and Trendafilov (2018).

not affect the cluster choices nor the interpretation of the intra-cluster relationships.

The implementation of the data taken from the ”Big Five Personality Test” data yielded the results as displayed in Table 5. Analogously to Adachi and Trendafilov (2018) participants of the test can be partitioned into five main groups, as indicated by the columns in Table 5. Note that some of the obtained values differ slightly from the ones obtained by Adachi and Trendafilov (2018). This is most likely due to the fact that the algorithm is of iterative nature that only produces an approximate best fit. However, reducing the convergence bound combined with increasing the total number of times the multi-run procedure is performed should yield increasingly equivalent results.

5.2 Results - Financial Data Sets

This section lays out the results of the financial data sets. Given the computational intensity of the algorithm, the maximum number of iterations for the multi-run procedure was reduced from 200 in Adachi and Trendafilov (2018) to 150 as initial results over the first few windows showed that all minimal values were obtained within the first 100 iterations even if no double optimum was reached within 200 iterations.

Starting with the stock indices data set, firstly, the number of clusters obtained via the Euclidean distance and the CID measure are compared. Then the most frequent cluster affiliation for each index are reported, and changes in cluster affiliations over time are discussed. Finally, the results section lays out how the obtained results can benefit an asset allocation manager. Consequently, the same is done for the DAX 30 return data.

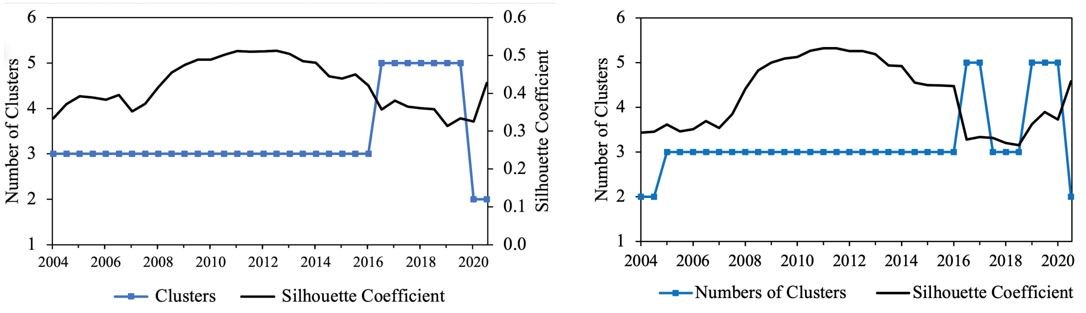
Table 5: . *Solutions for the big-five data set.*

Variable	$\boldsymbol{\Lambda}$					$\boldsymbol{\Psi}^2$
worry	0.70	.	.	.	.	0.40
sensitive	0.61	.	.	.	.	0.53
pessimistic	0.77	.	.	.	.	0.34
unrest	0.42	.	.	.	.	0.72
careful	0.67	.	.	.	.	0.45
sociable	.	0.85	.	.	.	0.26
talkative	.	0.75	.	.	.	0.38
voluntary	.	0.75	.	.	.	0.39
cheerful	.	0.84	.	.	.	0.27
showy	.	0.63	.	.	.	0.55
creative	.	.	0.70	.	.	0.41
adventurous	.	.	0.78	.	.	0.36
progressive	.	.	0.68	.	.	0.50
flexible	.	.	0.54	.	.	0.65
imaginative	.	.	0.41	.	.	0.75
mild	.	.	.	0.51	.	0.68
tenderhearted	.	.	.	0.59	.	0.60
altruistic	.	.	.	0.70	.	0.48
cooperative	.	.	.	0.68	.	0.50
sympathetic	.	.	.	0.79	.	0.34
deliberate	.	.	.	.	0.61	0.59
reliable	.	.	.	.	0.60	0.52
diligent	.	.	.	.	0.77	0.38
systematic	.	.	.	.	0.64	0.55
methodical	.	.	.	.	0.77	0.35
			$\boldsymbol{\Phi}$			
Factor 1	1.00	-0.29	-0.43	0.15	0.25	
Factor 2	-0.29	1.00	0.41	0.25	0.12	
Factor 3	-0.43	0.41	1.00	0.09	-0.17	
Factor 4	0.16	0.25	0.09	1.00	0.37	
Factor 5	0.25	0.12	-0.17	0.37	1.00	

Note. The sign of factors and their correlations were adjusted accordingly to be in line with Adachi and Trendafilov (2018).

Stock Indices Data

Figure 5a and 5b show the appropriate number of clusters determined by the silhouette method using the clusters obtained vie SSFA and the Euclidean distance as well as the CID, respectively. As mentioned by Batista et al. (2014), the CID should converge to the Euclidean distance if all time series have the same complexity. From the figures, it can be seen that, although they are similar, this is not the case. The general form of the silhouette coefficient over time is roughly similar, and for the majority of the time, both distance measures yield the same number of clusters. However, it seems that the Euclidean distance is less sensitive to changes in the underlying clusters and assumes a relatively higher silhouette coefficient for lower silhouette coefficients obtained via the CID. Following the analogy of the results of Batista et al. (2014), this paper, therefore, assumes that for the stock indices data, the CID represents a more appropriate distance measure than the Euclidean distance and disregards results



(a) Euclidean distance

(b) CID

Figure 5: Number of clusters and silhouette coefficient over time for the stock indices data based on (a) the Euclidean distance and (b) the CID measure.

obtained via the Euclidean distance for the remainder of the stock indices data analysis.

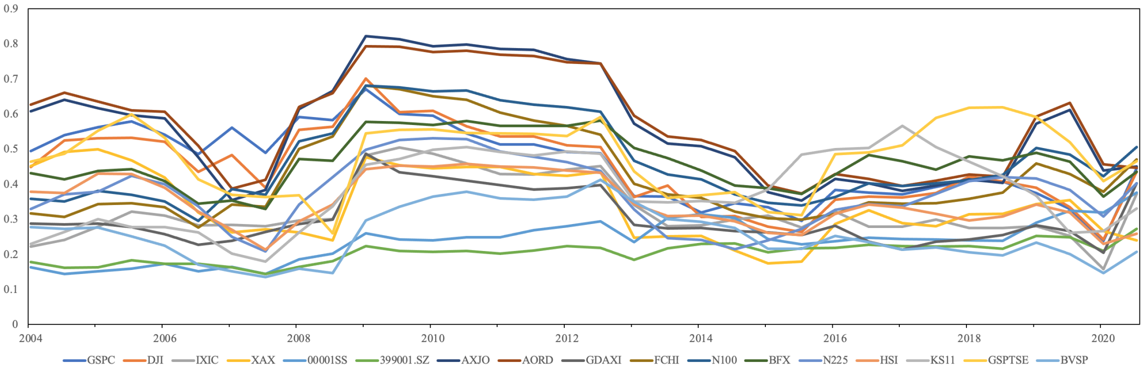


Figure 6: Betas over time for the stock indices log returns.

Inspecting the shortcoming of the current formulation of the distance matrix with respect to inverse cross-asset relationships Figure 6 displays the beta coefficient as laid out in section 4. It can be seen that, for all assets and all periods, the beta coefficient is sufficiently above zero, and thus the shortcoming can be disregarded as all assets are pro-cyclical in nature.

Next, this paper has a closer look at the clusters obtained via algorithm 2 as laid out in section 4. Figure 7 shows a flow diagram of the dynamic cluster affiliations over the full sample period. The obtained clusters show that there exist two main market segments as denoted by the tickers in cluster a.i and cluster b.i with only one mutual interaction in 01/2020 by BSVP. While the tickers of cluster a.i predominantly stay in the same cluster, the tickers in b.i are more fragmented over time, thus indicating a less robust segment. Note, however, that for the majority of the sample period (01/2005 - 01/2016), there are three stable clusters denoted by a.i, b.ii and c.i. Merely the period from 2019 onward observed instability, primarily in b-labelled and c-labelled clusters.

Looking at the practical implications of such a figure, there are numerous areas within the field of finance where it can add value. First of all, the dynamic risk cluster affiliation of Figure 7 gives portfolio managers an implication of the risk relationships of the different markets; that is, for example, a crash of the German markets will most likely affect the North American sectors more heavily than other geographic markets. Hence, looking at it from a diversification point of view, between two equivalent assets with one on the US market and one on the Hong Kong

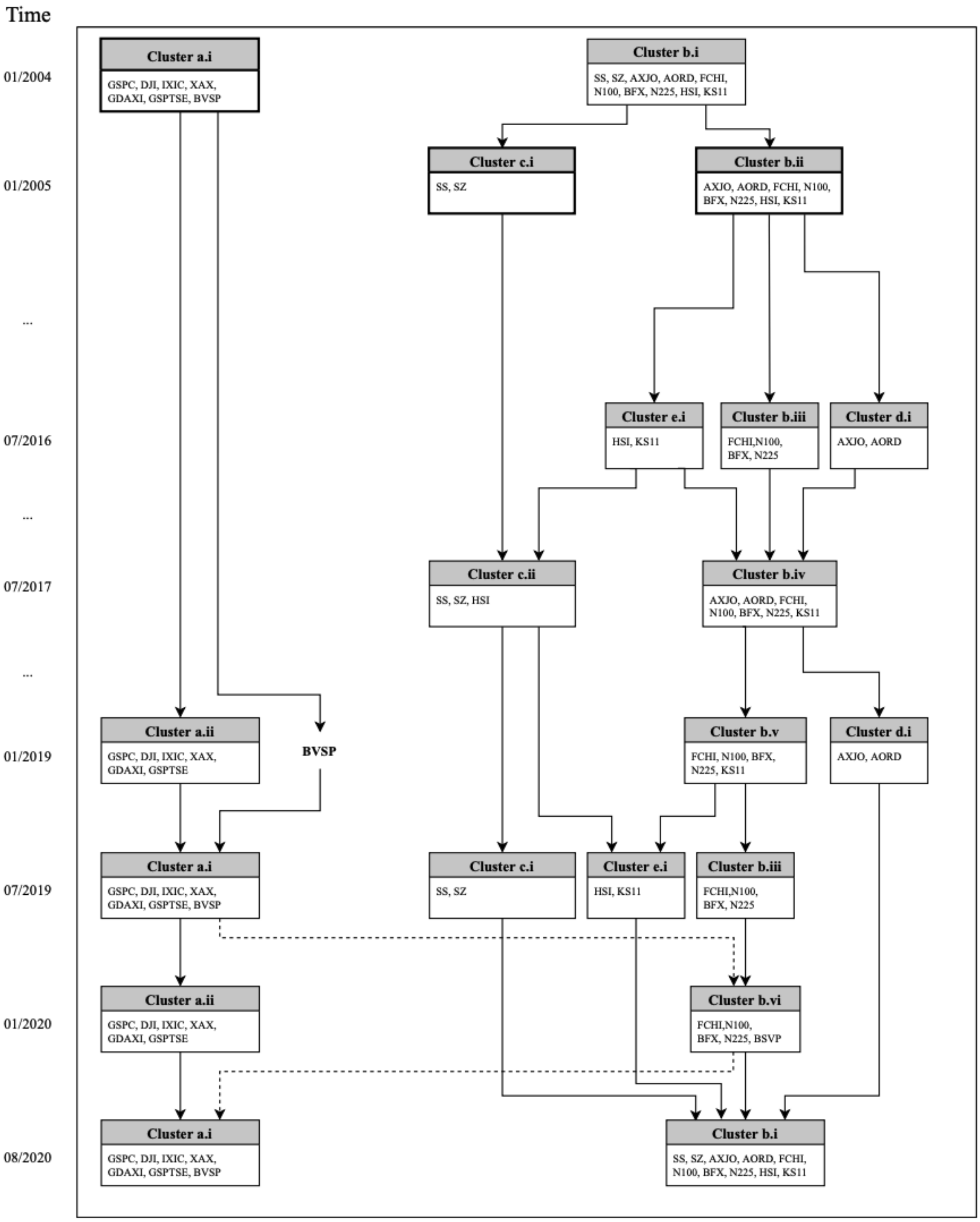

Time
01/2004
Cluster a.i
GSPC, DJI, IXIC, XAX, GDAXI, GSPTSE, BVSP
Cluster b.i
SS, SZ, AXJO, AORD, FCHI, N100, BFX, N225, HSI, KS11
01/2005
Cluster c.i
SS, SZ
Cluster b.ii
AXJO, AORD, FCHI, N100, BFX, N225, HSI, KS11
...
07/2016
Cluster e.i
HSI, KS11
Cluster b.iii
FCHI,N100, BFX, N225
Cluster d.i
AXJO, AORD
...
07/2017
Cluster c.ii
SS, SZ, HSI
Cluster b.iv
AXJO, AORD, FCHI, N100, BFX, N225, KS11
...
01/2019
Cluster a.ii
GSPC, DJI, IXIC, XAX, GDAXI, GSPTSE
BVSP
Cluster b.v
FCHI, N100, BFX, N225, KS11
Cluster d.i
AXJO, AORD
07/2019
Cluster a.i
GSPC, DJI, IXIC, XAX, GDAXI, GSPTSE, BVSP
Cluster c.i
SS, SZ
Cluster e.i
HSI, KS11
Cluster b.iii
FCHI,N100, BFX, N225
01/2020
Cluster a.ii
GSPC, DJI, IXIC, XAX, GDAXI, GSPTSE
Cluster b.vi
FCHI,N100, BFX, N225, BSVP
08/2020
Cluster a.i
GSPC, DJI, IXIC, XAX, GDAXI, GSPTSE, BVSP
Cluster b.i
SS, SZ, AXJO, AORD, FCHI, N100, BFX, N225, HSI, KS11

Figure 7: Dynamic cluster affiliation of stock indices over the full sample period.

Market, adding the latter to a portfolio of German stocks will achieve a higher level of diversification compared to the former. However, the potential benefits are not limited to the aspect of diversification. For instance, take the scenario of abnormally high growth forecasts for the North American stock market and more uncertain forecasts for the European market, here the figure would indicate contrary to forecasts that abnormally high growth of the German markets will also be likely. Given the uncertainty about European markets in this scenario, a portfolio manager could take long positions in the German markets and reap abnormally high returns that have not yet been realized.

However, the potential areas of applications for Figure 7 are not limited to the field of finance. Kwon and Shin (1999) investigated causality relationships between stock market index data and macroeconomic variables. They found that stock market index data is significantly co-integrated with macroeconomic variables such as production index, exchange rate, trade balance, and money supply. Thus such a figure can also serve as a potential risk indicator to policymakers. Namely, it indicated which other economies or stock market indices pose a threat from a recession risk standpoint to the domestic economy. In addition, the estimated $\mathbf{\Lambda}$ can be used to estimate the degree of such an exposure. To that extent, SSFA could also be directly applied to macroeconomic variables such as GDP growth to get a more direct measure. All in all, such a figure can give policymakers the ability to estimate the domestic exposure to adverse events happening in foreign economies. Even though this application is less relevant to the research question, it shows that the moving window SSFA can be applied in areas beyond finance and could be an interesting starting point for future papers.

Table 6 reports the percentages of cluster dependencies corresponding to Figure 7. The bold figures mark the most frequent clusters. It can be seen that all indices primarily belong to one of three clusters. Namely, a.i, b.ii and c.i. Looking more closely at these clusters, it can be seen that they bear some similarities with location-based clusters. Cluster a.i contains all North American Indices plus the German DAX index. Given that the US is the largest export partner to Germany[7] and that all North American indices are located in the same free trade union (NAFTA), this seems reasonable. Given the cluster's stability over the entire sample period, it is most likely the most distinct. The cluster c.i can be seen as the Hong Kong cluster. This cluster also makes sense, given that the Chinese stock market is highly regulated by the Chinese authorities, and thus is not able to move as freely as, e.g. its US counterparts. Given that the cluster only interacts with one other index with the exception of the first and last window, it can also be seen as a relatively strong and distinct one. The cluster b.ii which, contains indices from Europe, Asia and Oceania, is less clear. Given the unstable behaviour from 07/2016 onward, this cluster can be seen as less strong, and the sparsest constraint on $\mathbf{\Lambda}$ might be too strict. However, given that the three clusters are the dominant ones over the majority of the sample period, they can serve as a good approximation of clusters for the given indices.

[7] Source: https://www.destatis.de/EN/Themes/Economy/Foreign-Trade/Tables/order-rank-germany-trading-partners.html

Table 6: *Average cluster affiliation of stock indices return data in per cent.*

	Cluster										
Index Ticker	a.i	a.ii	b.i	b.ii	b.iii	b.iv	b.v	b.vi	c.i	c.ii	d.i
GSPC	**94.11**	5.88	.	.	.	.	.	.	.	.	.
DJI	**94.11**	5.88	.	.	.	.	.	.	.	.	.
IXIC	**94.11**	5.88	.	.	.	.	.	.	.	.	.
XAX	**94.11**	5.88	.	.	.	.	.	.	.	.	.
000001S	.	.	8.82	.	.	.	.	.	**79.41**	11.76	.
399001SZ	.	.	8.82	.	.	.	.	.	**79.41**	11.43	.
AXJO	.	.	8.82	**67.65**	.	8.82	.	.	.	.	14.71
AORD	.	.	8.82	**67.65**	.	8.82	.	.	.	.	14.71
GDAXI	**94.11**	5.88	.	.	.	.	.	.	.	.	.
FCHI	.	.	8.82	**67.65**	8.82	8.82	2.94	2.94	.	.	.
N100	.	.	8.82	**67.65**	8.82	8.82	2.94	2.94	.	.	.
BFX	.	.	8.82	**67.65**	8.82	8.82	2.94	2.94	.	.	.
N225	.	.	8.82	**67.65**	8.82	8.82	2.94	2.94	.	.	.
HSI	.	.	8.82	**67.65**	.	.		.	.	11.43	.
KS11	.	.	8.82	**67.65**	.	8.82	2.94	2.94	.	.	.
GSPTSE	**94.11**	5.88	.	.	.	.		.	.	.	.
BVSP*	**94.11**	.	.	.	.	.		2.94	.	.	.

Note. * BSVP does not sum up to 100% as it is in no cluster in 01/2019.

Figure 8 plots the realized returns of the equally weighted portfolio and the portfolio based on the best Sharpe ratio, and Table 7 reports the summary statistics, respectively. It can be seen that the cluster returns outperform the market returns. However, this increase in returns still comes at the cost of increased volatility. Especially during periods of financial stress like, e.g. the 2008 financial crisis, it can be seen that cluster diversification does not outperform the market. However, given that the average increase in return is 48.49% compared to a volatility increase of 30.01%, this might be a reasonable trade-off in the long run. In fact, the Sharpe ratio increases from 0.140 to 0.160 when using the cluster-based portfolio as compared to the market portfolio. In addition, the cluster portfolio has, on average, only 3.21 indices as compared to the market portfolio with 17 indices, thus reducing, e.g. the transaction costs of maintaining that portfolio. Looking at the pure Sharpe ratio-based portfolio, it can be seen that it actually performs worse than both the market and cluster-based portfolios in terms of Sharpe ratio. Although it has a higher average return than the market portfolio, its volatility increases more extremely. Thus, it can be concluded that when picking portfolio components based on their Sharpe ratio, picking them from a diverse set of clusters actually outperforms both of the aforementioned alternatives in returns and their Sharpe ratio. All in all, the increased returns of the cluster-based portfolio still come at the cost of higher volatility. However, that increase is relatively smaller than the increase in returns; hence, it might be a reasonable trade-off in the long term.

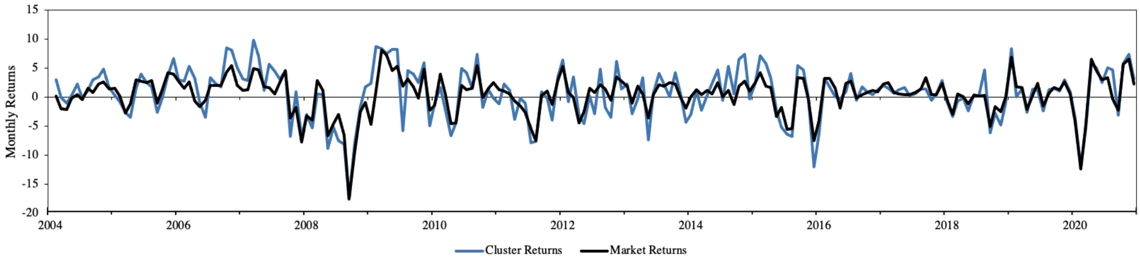
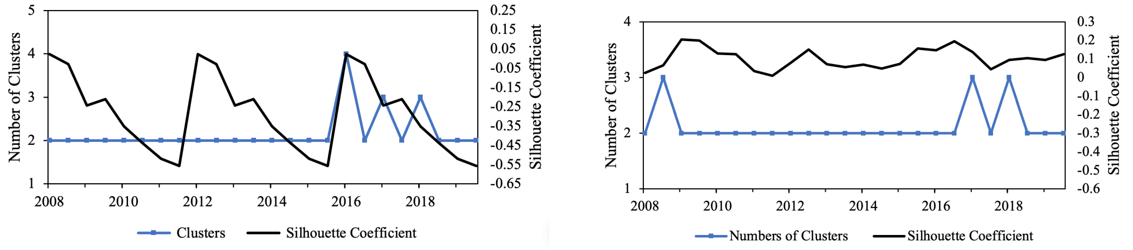


Figure 8: Stock indices cluster portfolio and market portfolio returns over time.

Table 7: *Stock indices performance summary of different portfolio strategies.*

	Cluster PF	Market PF	Pure Sharpe Ratio
Mean	0.689	0.463	0.521
Standard Deviation	4.316	3.317	4.167
Min.	-17.529	-17.451	-17.812
Max.	9.885	8.140	13.946
Sharpe Ratio	0.160	0.140	0.125
VaR(0.05)	-8.672	-6.737	-8.384

Dax30 Data



(a) Euclidean distance

(b) CID

Figure 9: Number of clusters and silhouette coefficient over time for the DAX 30 data based on (a) the Euclidean distance and (b) the CID measure.

Figure 9a and 9b show the appropriate number of clusters for the DAX 30 log return data determined by the silhouette method, again using the clusters obtained via SSFA and the Euclidean distance as well as the CID, respectively. Different from the stock indices, the justification for clustering is less evident here. The Silhouette coefficient is a lot closer to zero, and the majority of numbers of clusters obtained are at the lower bound two, which could indicate a true number of clusters equal to 1. Again we observe that the average silhouette coefficient of CID is higher than the average silhouette coefficient obtained via the Euclidean distance, which indicates that CID is likely the more appropriate choice. In addition, the obtained CID silhouette coefficients are much more stable than the obtained Euclidean silhouette coefficients, which seems more realistic as each window only updates 25 per cent of the included data, and thus the fundamental relationships between variables and the level of clustering should not change too much. Hence, this paper will again use the number of clusters obtained via the CID-based silhouette method for the remainder of this analysis.

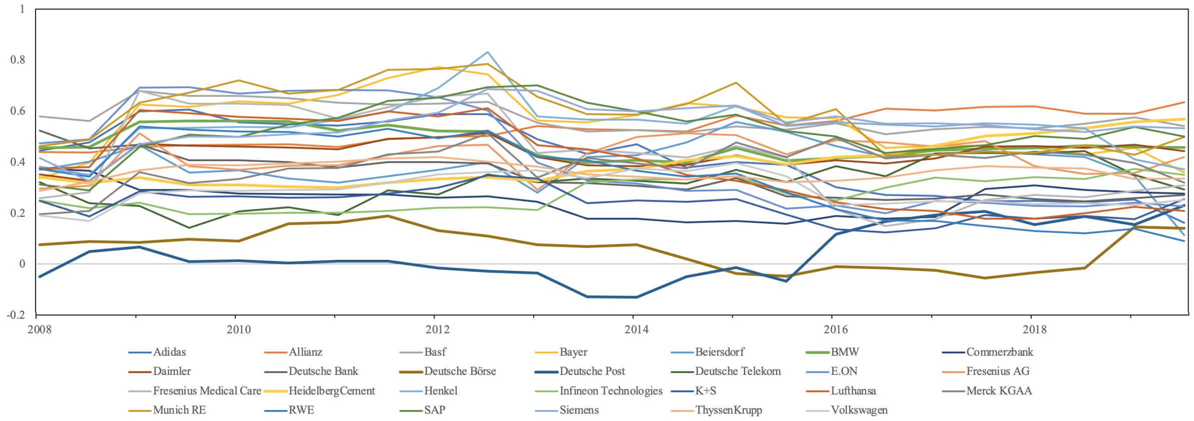


Figure 10: Betas over time for the DAX 30 constituents log returns.

With respect to the shortcoming of inversely related assets, Figure 10 displays the beta coefficient as defined in section 4 for all DAX constituents included over all investment dates. Although the beta value for *'Deutsche Post'* and *'Deutsche Börse'* can be observed to be below zero for some sample periods, they are still far away from having a sufficient negative level that would prompt the conclusion of being counter-cyclical. Again, all other assets have positive beta coefficients over all investment dates, and thus inverse relationships should not be a problem for the DAX data set.

The labelling of the clusters also indicated weak distinct, and strongly interrelated clusters. Different from the stock indices data, there is no continued period of stable clusters. Often only single stocks change the cluster affiliations in what seem to be more or less random manners. Looking simply at the cluster labels obtained via the rank of the factor the stock was found in (Table 8, one can see that only a minority (eight) of stocks manage to have an average cluster affiliation above two thirds for a given factor. This indicates that the stocks of the DAX 30 index have less distinct clusters which are interrelated. As a result, a visualization as given in Figure 7 proved itself to be too simplistic and very ambiguous and was thus excluded from this paper. However, even though the DAX 30 clusters were less stable than the stock indices clusters, a cluster-based investment strategy could still provide some added value. Figure 11 plots the realized returns of the equally weighted and the portfolio based on the best Sharpe ratio, based on the DAX log return data. The initial assessment is that the clusters actually perform worse during crises than the market-based portfolio. However, when looking at Table 9, which reports their summary statistics as well as the summary statistics of the pure Sharpe Ratio strategy, one can see that the cluster-based strategy actually outperforms the market. Despite uncertainty about the existence of clusters, the cluster-based portfolio outperforms the market as well as the pure Sharpe ratio portfolio in terms of returns. This increase in returns again comes at the price of higher volatility. However, given that the Sharpe ratio increases from 0.079 to 0.124 as compared to the market portfolio, this trade-off can be seen as justified. In addition, the cluster portfolio, again, outperforms the pure Sharpe ratio based portfolio in terms of Sharpe ratio and returns. Hence, it can be concluded that even in this case, using clusters to diversify Sharpe ratio performance-based stock picks adds value to the portfolio.

Table 8: *Average cluster affiliation of DAX 30 return data.*

Sector	Cluster 1	Cluster 2	Cluster 3
Fresenius Medical Care	**100.00**	.	.
Fresenius AG	**62.50**	37.50	.
Beiersdorf	**70.83**	29.17	.
Merck KGAA	**66.67**	29.17	4.17
Volkswagen	29.17	**66.67**	4.17
HeidelbergCement	**50.00**	45.83	4.17
K+S	25.00	**66.67**	8.33
Deutsche Telekom	**58.33**	37.50	4.17
E.ON	**70.83**	25.00	4.17
RWE	**75.00**	20.83	4.17
Infineon Technologies	25.00	**66.67**	8.33
SAP	**62.50**	33.33	4.17
Munich RE	**58.33**	41.67	.
Commerzbank	29.17	**66.67**	4.17
Allianz	33.33	**58.33**	8.33
Deutsche Bank	25.00	**70.83**	4.17
Deutsche Post	25.00	**70.83**	4.17
Adidas	**66.67**	33.33	.
Deutsche Börse	25.00	**75.00**	.
Henkel	**70.83**	29.17	.
Bayer	**58.33**	41.67	.
BMW	41.67	**54.17**	4.17
Daimler	37.50	**58.33**	4.17
Basf	37.50	**58.33**	4.17
Siemens	**50.00**	45.83	4.17
ThyssenKrupp	25.00	**66.67**	8.33
Lufthansa	37.50	**58.33**	4.17

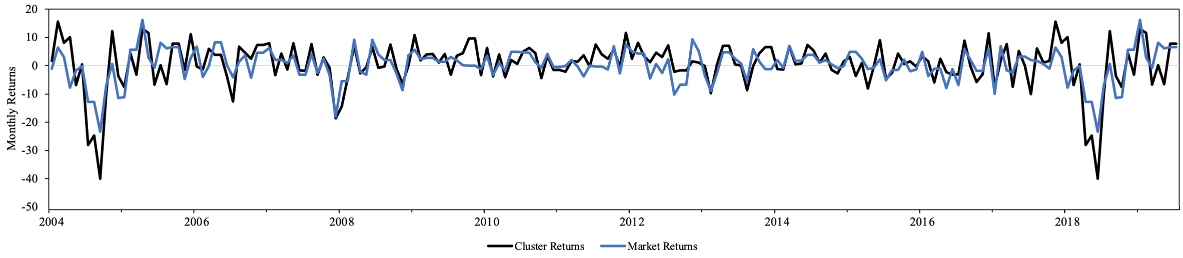


Figure 11: DAX 30 cluster portfolio and market portfolio returns over time.

Table 9: *DAX 30 performance summary for different portfolio strategies.*

	Cluster PF	Market PF	Pure Sharpe Ratio
Mean	0.883	0.416	0.640
Standard Deviation	7.108	5.242	7.085
Min.	-39.953	-23.388	-30.124
Max.	15.653	16.248	15.653
Sharpe Ratio	0.124	0.079	0.090
VaR(0.05)	-10.069	-9.915	-10.077

Note. The pure Sharpe ratio PF uses two assets

6 Conclusion

As laid out by Adachi and Trendafilov (2018), the SSFA algorithm is an effective tool in clustering cross-sectional data sets. SSFA applied on the 'big-five' personality dataset as well as the simulated data set consistently estimates the correct sparsest $\mathbf{\Lambda}$. Thus, cross-sectional data can effectively cluster both positively and inversely related variables and explain their relationships relatively easily. Hence the answer to the first sub-research question of this paper is that, yes, SSFA can recover true clusters in an experimental and real setting. Additionally, this paper found that this is also highly likely for the stock indices time series data, given the stability of the stock indices clusters over the majority of the moving window structure. However, it is beyond the scope of this paper to formally prove and validate this with a simulated data set and is thus left for future research.

For the stock indices data, the silhouette method with the Euclidean distance already supplies surprisingly stable results. However, following the results of Batista et al. (2014) the CID is a more appropriate measure for time series distances. The silhouette method applied to the DAX data yielded less clear results. Although the use of CID improved the stability of the number of obtained clusters, the actual number of obtained clusters was almost always at the lower bound of two. This indicated a much lower or possibly heavily interrelated level of clustering. Hence, the answer to the second sub research question is that given the assumption of clusters being present, the silhouette method with the clusters obtained via SSFA and the CID matrix as a distance measure can be used to obtain the appropriate number of clusters. However, the DAX index has shown that if the assumption of clusters is relaxed, the silhouette method cannot differentiate between two and no clusters being present. Hence, here a modified Gap Statistic that can deal with time-series data could be the more reasonable choice. This is, however, left up to future research.

Having answered these two sub-questions, this paper can provide an answer to the main research question as proposed in section 1:

"To what extend can Sparsest Factor Models improve the return vs volatility trade-off of financial asset allocation?"

The results have shown that given apparent clustering, SSFA provides a clear visual and interpretable representation of relevant clusters over time. Such a visual representation of risk clusters over time can already by itself be an aiding tool for asset allocation managers when it comes to evaluating potential new assets to be included in a given portfolio. However, given less clear clusters, this is not the case as elements tend to jump frequently between clusters. With respect to their statistical usefulness to portfolio managers, they can help them select a small high-performance subset of assets less dependent on each other as when simply picking the high-performance assets without clustering. For both data sets, the inclusion of clustering in a Sharpe-Ratio based asset allocation strategy yields increases in realized returns. Even though they still come at the cost of higher volatility, an increased Sharpe ratio for both data sets indicates the volatility/return trade-off to be worth it. Consequently, managers can increase the number of stocks included from each cluster, thus being able to effortlessly trade-off returns with volatility.

7 Discussion and Further Research

As already pointed out, there are a few limitations to the findings of this research. The first and most crucial one is the appropriateness of clustering for a given set of time series. As Adachi and Trendafilov (2018) already pointed out, the assumption of clustering must hold for the data in order to obtain good results with the SSFA algorithm. Although the silhouette method already gives some indication on the number of clusters, it is still not able to differentiate between two clusters and no clusters at all as it is not defined for just a single/no cluster. Here I would suggest the paper of Ribeiro and Rios (2021) that already implemented a version of the Gap-Statistic for time series. For its successful application most likely, the definition of distribution of the returns under the null hypothesis will be crucial for the level of success and this could be a thought-provoking starting point for future research.

A second limitation is that the obtained clusters do not account for possible lagged behaviour clusters. However, given that monthly data is used, this should not occur too often, and in addition, a lagged relationship can also be seen as two separate clusters as the lagged assets do not react to an initial signal but react to the assets changing due to that signal.

A third limitation is the use of constituents of a single stock market index (DAX) which is known to be closely interrelated. The use of global indices yielded more distinct clusters, and hence it might be interesting to combine constituents of different global indices and not just the DAX and consequently apply the same methodology. The hope would be to obtain a number of clusters that is bigger than two, thus obtaining a data set of individual stocks and not indices that observe clusters.

However, increasing the number of variables also forces an increase in the window length. That is, as the SSFA algorithm has a lower bound on the sample size that is equal to the number of variables included. Increasing the window length could dilute the actual clusters currently present with past clusters. One way to tackle this is to increase the frequency of the data to, e.g. daily or to use a dynamic covariance structure that assigns heavier weights to more recent observations. With respect to the dynamic covariance matrix, this paper refers to the research of Chen and Leng (2016) whose dynamic covariance implementation for time series showed promising results. However, given that the primary purpose of this paper is to demonstrate the feasibility of the SSFA in a time series framework, this is beyond the scope and also left for future research.

References

Adachi, K., & Trendafilov, N. T. (2015). Sparse orthogonal factor analysis. In M. Carpita, E. Brentari, & E. M. Qannari (Eds.), *Advances in latent variables: Methods, models and applications* (pp. 227–239). Springer International Publishing. https://doi.org/10.1007/10104_2014_2

Adachi, K., & Trendafilov, N. T. (2018). Sparsest factor analysis for clustering variables: A matrix decomposition approach. *Advances in Data Analysis and Classification*, *12*, 559–585. https://doi.org/10.1007/s11634-017-0284-z

Batista, G., Keogh, E., Tataw, O., & de Souza, V. A. (2014). Cid: An efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, *28*, 634–669. https://doi.org/10.1007/s10618-013-0312-3

Chen, Z., & Leng, C. (2016). Dynamic covariance models. *Journal of the American Statistical Association*, *111*(515), 1196–1207. https://doi.org/10.1080/01621459.2015.1077712

Connor, G. (1995). The three types of factor models: A comparison of their explanatory power. *Financial Analysts Journal*, *51*(3), 42–46. https://doi.org/10.2469/faj.v51.n3.1904

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, *1*(2), 223–236. https://doi.org/10.1080/713665670

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., & Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, *1*(2), 1542–1552. https://doi.org/10.14778/1454159.1454226

Fama, E. F., & French, K. R. (1992). The cross-section of expected stock returns. *Journal of Finance*, *47*(2), 427–465. https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

Farrell, J. L. (1974). Analyzing covariation of returns to determine homogeneous stock groupings. *The Journal of Business*, *47*(2), 186–207. https://doi.org/10.2307/2353379

Hirose, K., & Yamamoto, M. (2014). Sparse estimation via nonconvex penalized likelihood in factor analysis model. *Statistics and Computing*, *25*, 863–875. https://doi.org/10.1007/s11222-014-9458-0

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. https://doi.org/10.1037/h0071325

Jolliffe, I. T. (2002). Principal component analysis. *Springer series in statistics*. Springer-Verlag New York. https://doi.org/https://www.springer.com/gp/book/9780387954424

Kwon, C. S., & Shin, T. S. (1999). Cointegration and causality between macroeconomic variables and stock market returns. *Global Finance Journal*, *10*(1), 71–81. https://doi.org/10.1016/S1044-0283(99)00006-X

Markowitz, H. M. (1959). Portfolio selection: Efficient diversification of investments. Yale University Press.

Müller, M. (2007). Dynamic time warping. *Information Retrieval for Music and Motion*, *2*, 69–84. https://doi.org/10.1007/978-3-540-74048-3_4

Papenbrock, J. (2011). *Asset clusters and asset networks in financial risk management and portfolio optimization* (Doctoral dissertation). https://doi.org/10.5445/IR/1000025469

Parson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, *2*(6), 559–572. https://doi.org/10.1080/14786440109462720

Ribeiro, R. G., & Rios, R. (2021). Temporal gap statistic: A new internal index to validate time series clustering. *Chaos, Solitons Fractals*, *142*, 110–326. https://doi.org/10.1016/j.chaos.2020.110326

Rosenberg, B., & McKibben, W. (1973). The prediction of systematic and specific risk in common stocks. *The Journal of Financial and Quantitative Analysis*, *8*(2), 317–333. https://doi.org/10.2307/2330027

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, *19*(3), 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Takigawa, M., Yamada, T., Yoshida, Y., Ishikawa, K., Aoyama, Y., Yamamoto, T., Inoue, N., Tatematsu, Y., Nanasato, M., Kato, K., Tsuboi, N., & Hirayama, H. (2012). The incidence and clinical significance of non-isolation of the pulmonary vein carina after encircling ipsilateral pulmonary veins isolation for paroxysmal atrial fibrillation: A pitfall of the double-lasso technique. *EP Europace*, *15*(1), 33–40. https://doi.org/10.1093/europace/eus243

Tan, C. W., Webb, G., Petitjean, F., & PaulReichl. (2017). Tamping effectiveness prediction using supervised machine learning techniques. https://doi.org/10.1061/9780784481257.101

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(2), 411–423. https://doi.org/10.1111/1467-9868.00293

Vichi, M., & Saporta, G. (2009). Clustering and disjoint principal component analysis. *Computational Statistics Data Analysis*, *53*(8), 3194–3208. https://doi.org/10.1016/j.csda.2008.05.028

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, *38*(2), 894–942. https://doi.org/10.1214/09-AOS729

Zou, H., Hastie, T., & Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, *15*(2), 265–286. https://doi.org/10.1198/106186006X113430

A Algorithms

Algorithm 6: Gap statistic

```
Result: $\hat{k}$
[1] Cluster data into $W_k$ for $k = 1, ..., K$;
[2] Generate: $B$ reference datasets using $Unif(*)$ and cluster each;
[3] Obtain: $W^*_{k,b}$ for $b = 1, 2, ..., B$ and $k = 1, ..., K$;
[4] Compute: $GAP(k) = \frac{1}{B}\sum_b \log(W^*_{k,b}) - \log(W_k)$;
[5.1] Set: $\bar{l} = \frac{1}{B}\sum_b \log(W^*_{k,b})$ ;
[5.2] Set: $sd_k = \left[\frac{1}{B}\sum_b \left\{\log(W^*_{k,b} - \bar{l}\right\}\right]$;
[6] Define: $s_k = sd_k\sqrt{1 + \frac{1}{B}}$;
[7] Obtain: $\hat{k} = \underset{1 \leq k \leq K}{\text{argmin}}\, GAP(k)$ s.t. $GAP(k) \geq GAP(k+1) - s_{k+1}$;
```

B Tables

Table 10: *Variables for the three DGP's of the simulated datasets.*

Variable	(A) m = 3				(B) m = 4					(C) m = 5					
	$\mathbf{\Lambda}$			$\mathbf{\Psi}^2$	$\mathbf{\Lambda}$				$\mathbf{\Psi}^2$	$\mathbf{\Lambda}$					$\mathbf{\Psi}^2$
Variable 1	0.9	.	.	0.19	0.9	.	.	.	0.19	-0.9	.	.	.	.	0.19
Variable 2	-0.8	.	.	0.36	-0.8	.	.	.	0.36	0.8	.	.	.	.	0.36
Variable 3	0.7	.	.	0.51	0.7	.	.	.	0.51	-0.7	.	.	.	.	0.51
Variable 4	-0.6	.	.	0.64	-0.6	.	.	.	0.64	0.6	.	.	.	.	0.64
Variable 5	0.5	.	.	0.75	0.5	.	.	.	0.75	-0.5	.	.	.	.	0.75
Variable 6	-0.4	.	.	0.84	-0.4	.	.	.	0.84	0.4	.	.	.	.	0.84
Variable 7	.	0.8	.	0.36	.	0.8	.	.	0.36	.	0.8	.	.	.	0.36
Variable 8	.	-0.7	.	0.51	.	-0.7	.	.	0.51	.	-0.7	.	.	.	0.51
Variable 9	.	0.6	.	0.64	.	0.6	.	.	0.64	.	0.6	.	.	.	0.64
Variable 10	.	-0.5	.	0.75	.	-0.5	.	.	0.75	.	-0.5	.	.	.	0.75
Variable 11	.	0.4	.	0.84	.	0.4	.	.	0.84	.	0.4	.	.	.	0.84
Variable 12	.	.	-0.7	0.51	.	.	0.7	.	0.51	.	.	0.7	.	.	0.51
Variable 13	.	.	0.6	0.64	.	.	-0.6	.	0.64	.	.	-0.6	.	.	0.64
Variable 14	.	.	-0.5	0.75	.	.	0.5	.	0.75	.	.	0.5	.	.	0.75
Variable 15	.	.	0.4	0.84	.	.	-0.4	.	0.84	.	.	-0.4	.	.	0.84
Variable 16					.	.	.	0.8	0.36	.	.	.	-0.8	.	0.36
Variable 17					.	.	.	-0.7	0.51	.	.	.	0.7	.	0.51
Variable 18					.	.	.	0.6	0.64	.	.	.	-0.6	.	0.64
Variable 19					.	.	.	-0.5	0.75	.	.	.	0.5	.	0.75
Variable 20					.	.	.	0.4	0.84	.	.	.	-0.4	.	0.84
Variable 21										.	.	.	.	0.7	0.51
Variable 22										.	.	.	.	-0.6	0.64
Variable 23										.	.	.	.	0.5	0.75
Variable 24										.	.	.	.	-0.4	0.84
	$\mathbf{\Phi}$				$\mathbf{\Phi}$					$\mathbf{\Phi}$					
Factor 1	1.0	0.4	0.3		1.0	0.4	0.3	-0.2		1.0	0.4	0.3	-0.2	0.3	
Factor 2	0.4	1.0	-0.4		0.4	1.0	-0.4	0.3		0.4	1.0	-0.4	0.3	-0.4	
Factor 3	0.3	-0.4	1.0		0.3	-0.4	1.0	-0.3		0.3	-0.4	1.0	-0.3	0.2	
Factor 4					-0.2	0.3	-0.3	1.0		-0.2	0.3	-0.3	1.0	-0.3	
Factor 5										0.3	-0.4	0.2	-0.3	1.0	

Note. The dot indicates zeros.

Table 11: Summary of VAR(1) estimates over Jul. 1985 - Dec. 2004.

Lag	GSPC	DJI	IXIC	XAX	000001S	399001SZ	AXJO	AORD	GDAXI	FCHI	N100	BFX	N225	HSI	KS11	GSPTSE	BVSP
1	0.075	0.024	0.083	-0.053	0.107	0.137	0.065	0.088	0.064	0.073	0.103	0.177	0.123	0.086	0.040	0.157	0.119
2	-0.070	-0.109	-0.083	0.025	0.130	0.140	-0.012	-0.010	-0.062	-0.061	-0.040	-0.019	0.034	0.018	0.001	0.035	-0.074
3	0.073	0.041	0.010	-0.020	0.018	0.038	0.065	0.071	0.055	0.058	0.058	0.024	0.073	-0.010	0.100	0.035	-0.021
4	0.060	0.074	0.026	0.049	0.193	0.145	0.006	0.003	0.000	0.040	0.045	0.142	-0.008	0.008	-0.022	0.019	0.006
5	0.030	0.004	0.061	-0.042	0.084	0.107	0.004	-0.011	0.029	0.019	0.052	0.112	0.008	0.029	-0.035	-0.034	-0.032
6	-0.074	-0.077	0.052	0.033	-0.040	-0.057	0.010	0.000	0.000	0.014	0.004	0.008	-0.066	0.033	-0.095	-0.073	0.039
7	0.073	0.075	0.103	0.014	0.097	0.079	0.084	0.076	0.003	0.025	0.020	0.014	0.055	-0.084	0.034	-0.014	0.004
8	0.039	0.031	-0.051	-0.026	-0.035	0.003	-0.097	-0.099	0.074	0.014	0.038	0.011	-0.006	-0.127	-0.064	-0.112	-0.091
9	-0.087	-0.079	-0.058	-0.088	-0.035	-0.083	-0.041	-0.052	-0.020	-0.020	-0.014	-0.020	0.016	-0.007	-0.021	-0.075	-0.135
10	0.005	-0.056	0.136	0.064	-0.018	-0.032	-0.023	-0.024	-0.080	-0.024	-0.041	-0.029	0.048	0.068	0.028	0.039	0.071
11	0.002	0.011	0.050	-0.002	-0.004	0.022	-0.057	-0.054	-0.046	-0.064	-0.064	-0.065	0.008	0.019	0.003	0.056	0.112
12	0.021	0.060	-0.025	-0.077	-0.092	-0.093	0.084	0.067	0.091	0.089	0.073	0.030	0.000	-0.061	-0.025	-0.054	0.052
13	-0.036	-0.080	-0.134	0.004	-0.212	-0.219	-0.143	-0.140	-0.077	-0.052	-0.064	-0.067	0.033	-0.146	-0.151	-0.079	-0.093
14	-0.051	-0.063	-0.035	0.009	-0.193	-0.199	-0.083	-0.080	0.003	-0.014	-0.017	-0.023	0.009	-0.101	-0.107	-0.082	-0.045
15	0.093	0.099	0.039	0.079	-0.006	0.020	0.026	0.035	0.006	-0.004	0.000	-0.006	0.041	0.062	0.024	0.072	-0.027
16	0.057	0.030	0.114	-0.002	-0.062	-0.009	0.080	0.079	0.036	0.047	0.050	0.076	0.001	-0.045	-0.098	0.013	-0.007
17	-0.027	-0.048	0.047	-0.002	-0.057	-0.054	-0.007	0.003	0.016	0.038	0.032	0.037	-0.049	0.015	-0.013	0.000	-0.071
18	-0.047	-0.065	-0.019	-0.082	-0.121	-0.062	0.033	0.024	-0.010	-0.004	-0.022	-0.053	-0.010	-0.012	-0.007	0.010	0.015
19	0.119	0.111	0.066	0.064	0.070	0.065	0.016	0.021	0.062	0.072	0.060	-0.074	0.065	0.099	0.041	0.060	0.028
20	-0.083	-0.090	-0.104	-0.044	-0.017	-0.027	-0.123	-0.121	-0.078	-0.067	-0.064	-0.045	0.039	-0.021	-0.011	-0.030	-0.008
21	-0.025	-0.065	0.066	0.028	-0.034	-0.033	-0.038	-0.047	-0.080	-0.015	-0.030	-0.018	-0.006	0.043	0.055	0.018	0.048
22	-0.044	-0.056	0.048	-0.062	-0.076	-0.069	-0.048	-0.049	0.055	0.010	0.010	-0.009	0.010	0.037	0.078	-0.023	0.076
23	0.009	0.013	-0.009	-0.035	-0.006	0.014	-0.028	-0.035	-0.014	-0.040	-0.054	-0.058	-0.001	-0.091	-0.035	-0.034	0.014
24	0.032	0.067	-0.025	0.012	-0.092	-0.059	0.008	-0.001	0.034	-0.017	-0.019	-0.039	0.002	-0.041	0.118	-0.062	-0.021
25	-0.098	-0.089	-0.041	0.068	-0.005	-0.029	-0.048	-0.053	0.012	-0.056	-0.060	-0.092	-0.030	-0.002	-0.024	-0.048	-0.046
26	-0.097	-0.084	-0.043	-0.021	0.034	0.047	-0.033	-0.029	-0.091	-0.058	-0.053	-0.070	-0.022	-0.120	-0.049	-0.058	-0.046
27	-0.039	-0.032	-0.011	-0.060	0.127	0.127	-0.040	-0.040	0.019	0.025	0.014	-0.072	0.011	0.001	0.045	-0.092	0.022
28	-0.038	-0.018	-0.017	-0.036	-0.117	-0.128	0.036	0.033	-0.076	-0.061	-0.057	-0.033	-0.022	-0.064	-0.048	-0.069	0.129
29	-0.040	-0.053	-0.050	-0.052	-0.060	-0.016	-0.037	-0.022	-0.118	-0.095	-0.076	-0.039	0.039	-0.011	0.049	-0.012	0.026
30	0.008	0.005	-0.003	-0.032	-0.050	-0.043	0.012	0.009	0.079	0.000	-0.007	-0.026	0.033	-0.014	-0.058	-0.028	-0.047

Note. * $p < 0.1$, ** $p < 0.05$.

Table 12: Autocorrelations of Dax 30 constituents' log returns up to lag 30 (1)

Lag	FME.DE	FRE.DE	BEI.DE	MRK.DE	VOW3.DE	HEI.DE	SDF.DE	DTE.DE	EOAN.DE	RWE.DE	IFX.DE	SAP.DE	MUV2.DE	CBK.DE	ALV.DE	DBK.DE	DPW.DE
1	-0.151	0.038	-0.043	-0.045	0.078	0.126	0.100	-0.084	-0.023	-0.014	0.319	-0.017	-0.168	0.177	-0.024	0.103	-0.062
2	0.028	-0.048	0.094	-0.005	-0.101	0.034	0.011	-0.053	-0.115	-0.025	0.065	-0.065	-0.026	-0.045	-0.266	-0.106	-0.123
3	-0.042	0.093	-0.166	-0.084	-0.026	0.150	0.000	-0.024	0.109	0.049	0.067	-0.125	-0.022	0.018	0.103	0.045	0.022
4	-0.106	0.066	-0.012	-0.058	0.037	0.027	-0.024	0.066	-0.013	-0.075	-0.020	0.021	-0.009	0.141	0.255	0.028	0.229
5	0.045	0.024	-0.074	-0.033	0.012	0.075	-0.004	-0.137	-0.035	0.052	-0.163	-0.198	0.048	0.066	-0.050	-0.003	-0.020
6	-0.018	0.034	0.058	0.011	-0.034	-0.117	-0.036	-0.097	0.005	0.067	-0.281	-0.167	-0.146	-0.184	-0.142	-0.180	-0.029
7	0.151	0.032	0.011	0.062	-0.081	-0.041	-0.182	-0.040	-0.040	-0.110	-0.102	-0.026	0.064	-0.078	-0.067	-0.033	-0.043
8	0.017	0.152	0.105	0.027	-0.029	-0.031	0.094	0.138	0.105	0.147	0.038	0.171	0.047	-0.019	0.117	0.027	0.085
9	0.085	0.025	0.032	0.087	0.127	-0.106	0.042	0.020	0.029	0.025	0.051	0.116	0.104	-0.022	0.142	0.044	-0.042
10	-0.029	-0.017	-0.055	-0.024	-0.178	-0.116	0.003	-0.062	-0.192	-0.112	0.057	0.015	-0.068	-0.035	-0.128	0.026	-0.075
11	-0.147	-0.040	0.011	-0.005	-0.035	0.017	0.091	-0.009	0.020	0.011	0.156	-0.015	-0.022	-0.009	-0.036	-0.052	-0.077
12	-0.013	-0.077	0.055	-0.109	0.129	0.044	-0.071	0.119	0.119	0.028	0.110	0.110	0.098	0.140	0.125	0.047	0.096
13	-0.044	0.025	-0.043	0.039	0.002	0.071	-0.004	-0.086	-0.051	0.047	-0.086	-0.038	0.018	0.066	-0.062	-0.064	0.018
14	0.093	0.080	0.014	0.035	0.005	0.000	0.078	0.065	0.010	0.077	-0.089	-0.088	-0.057	-0.010	-0.086	-0.069	-0.110
15	0.080	-0.052	-0.109	0.051	0.000	0.093	0.038	0.000	0.044	-0.077	-0.013	-0.026	0.005	-0.008	0.131	0.038	0.030
16	0.018	0.073	-0.072	0.108	-0.085	0.000	-0.059	0.070	0.032	0.037	-0.021	-0.002	-0.063	-0.062	-0.040	-0.069	-0.017
17	-0.044	0.020	-0.040	-0.029	-0.040	-0.065	-0.058	-0.102	-0.033	0.003	-0.131	0.008	-0.023	-0.041	-0.148	-0.143	0.023
18	-0.063	-0.015	0.013	-0.105	0.086	0.073	-0.018	0.008	-0.037	-0.134	-0.036	-0.046	-0.081	-0.031	-0.011	-0.118	-0.022
19	0.002	-0.042	-0.064	0.096	-0.005	0.008	0.184	-0.002	0.048	0.037	0.023	0.006	0.030	0.035	0.059	0.118	0.102
20	-0.068	-0.037	0.027	0.021	-0.077	0.029	0.038	-0.062	0.097	-0.019	-0.025	0.003	-0.003	-0.127	-0.047	0.015	-0.070
21	0.069	-0.011	-0.079	-0.057	-0.015	-0.009	-0.070	-0.061	-0.120	-0.077	-0.097	0.043	0.010	-0.070	-0.038	-0.049	-0.071
22	0.033	0.042	0.049	0.042	-0.013	-0.046	0.008	0.128	-0.002	0.108	-0.006	-0.021	0.033	-0.003	0.006	0.022	-0.005
23	-0.012	-0.035	-0.074	0.023	-0.032	-0.069	-0.049	-0.026	-0.027	-0.028	-0.094	0.095	-0.050	0.031	0.042	0.073	0.033
24	-0.101	-0.136	0.035	0.100	-0.120	-0.130	-0.054	0.091	0.034	0.046	-0.095	-0.006	0.055	-0.121	-0.092	-0.034	-0.132
25	0.030	-0.019	0.003	0.068	0.028	-0.034	-0.009	-0.056	-0.082	-0.017	-0.042	0.057	-0.065	-0.108	-0.013	-0.068	0.012
26	0.058	-0.010	0.001	-0.044	-0.052	-0.065	0.005	-0.077	-0.098	-0.013	-0.052	-0.095	-0.042	0.014	-0.053	-0.027	-0.083
27	-0.107	-0.086	-0.101	-0.004	-0.020	-0.015	0.098	0.081	0.063	0.006	-0.044	-0.055	-0.009	0.027	0.009	0.012	0.025
28	0.050	0.015	-0.082	0.008	-0.058	-0.025	0.038	-0.049	0.034	0.045	-0.062	-0.133	-0.012	-0.028	-0.092	-0.120	-0.071
29	-0.101	-0.011	0.022	-0.025	0.067	-0.018	0.021	-0.092	-0.006	0.061	-0.065	0.021	0.023	-0.019	0.019	0.008	0.047
30	-0.034	-0.013	0.066	-0.065	0.057	0.067	0.072	0.096	0.083	-0.003	0.046	0.097	0.048	0.129	0.099	0.063	-0.013

Note. * $p < 0.1$, ** $p < 0.05$.

Table 13: Autocorrelations of DAX 30 constituents' log returns up to lag 30 (2)

Lag	ADS.DE	DB1.DE	HEN3.DE	BAYN.DE	BMW.DE	DAI.DE	BAS.DE	SIE.DE	TKA.DE	LHA.DE
1	-0.015	0.052	0.003	-0.045	-0.038	-0.003	0.090	-0.002	0.106	0.025
2	-0.081	0.043	0.039	0.028	-0.142	-0.065	-0.047	-0.071	-0.074	0.106
3	0.052	-0.154	-0.065	-0.060	0.138	0.059	0.015	-0.006	-0.017	0.045
4	0.029	-0.046	0.125	0.175	0.026	0.068	0.113	0.058	0.095	0.105
5	0.038	-0.081	-0.017	0.047	-0.032	0.057	-0.059	0.046	-0.002	-0.075
6	-0.020	-0.061	0.069	-0.053	-0.094	-0.078	-0.170	-0.194	-0.017	-0.115
7	0.078	0.142	-0.005	0.028	0.012	0.027	0.063	0.090	-0.105	0.030
8	0.114	-0.008	0.137	0.077	0.012	-0.007	0.039	0.115	0.070	0.059
9	-0.023	0.025	-0.052	0.127	0.153	0.061	-0.013	0.024	0.058	0.003
10	-0.129	0.020	-0.002	-0.188	-0.022	-0.113	-0.142	-0.063	-0.078	-0.065
11	-0.032	0.015	-0.059	-0.083	-0.021	-0.100	-0.029	-0.119	-0.037	0.016
12	0.047	0.153	-0.003	-0.064	-0.002	0.035	0.031	0.016	-0.021	0.011
13	0.013	-0.029	-0.021	0.079	-0.014	-0.003	-0.048	-0.017	-0.115	-0.030
14	-0.092	0.007	0.026	0.036	0.003	0.025	-0.053	-0.076	0.082	-0.031
15	0.069	-0.109	-0.020	-0.043	-0.027	-0.105	0.075	0.043	0.114	-0.022
16	0.020	0.029	0.088	0.000	0.046	-0.051	-0.010	-0.082	0.032	0.059
17	-0.087	-0.036	-0.043	0.073	-0.082	-0.072	-0.018	-0.072	-0.107	-0.176
18	-0.104	0.050	-0.098	-0.088	0.001	-0.024	-0.054	-0.073	-0.052	-0.107
19	-0.068	0.135	-0.102	0.006	0.073	0.059	0.058	-0.044	0.072	-0.105
20	-0.060	0.087	0.017	-0.112	-0.108	-0.142	-0.158	-0.003	-0.043	-0.062
21	-0.022	-0.092	-0.130	-0.085	-0.024	0.013	-0.072	-0.098	-0.080	-0.142
22	-0.015	-0.130	0.008	0.003	-0.017	0.064	0.014	-0.006	-0.045	-0.023
23	0.058	-0.061	-0.033	0.028	-0.068	-0.160	0.021	0.063	-0.063	0.077
24	-0.082	0.033	-0.004	-0.066	-0.034	-0.080	-0.041	-0.100	-0.112	-0.046
25	-0.005	-0.096	-0.146	0.045	-0.031	0.016	-0.096	-0.178	-0.007	-0.038
26	-0.045	0.033	-0.035	-0.042	-0.021	0.036	-0.023	-0.063	-0.078	-0.075
27	-0.051	-0.025	-0.058	0.034	0.001	-0.045	-0.050	0.057	-0.067	-0.070
28	0.132	0.021	0.091	0.031	-0.026	-0.075	-0.072	-0.094	-0.078	-0.103
29	-0.011	-0.055	-0.110	-0.013	-0.104	-0.090	-0.060	0.021	0.056	-0.080
30	0.028	-0.023	0.102	0.116	0.021	0.070	0.182	0.058	-0.001	-0.062

Note. * $p < 0.1$, ** $p < 0.05$.

C Code

This appendix explains the included code scripts and their content. The code itself and the data sets can be found in the complimentary zip file. Note, the scripts themselves also contain a detailed description of their contents.

- SSFA_Utils.R: This script contains all relevant functions needed for the SSFA multi-run procedure. That is, a function initializing $\mathbf{\Lambda}$, a function running a single SSFA procedure and a function running the single SSFA procedure multiple times.
- SOFA_Utils.R: This script contains all relevant functions needed for the SOFA multi-run procedure. That is, a function initializing $\mathbf{\Lambda}$, a function running a single SOFA procedure and a function running the single SOFA procedure multiple times. Note, this procedure was not actually used in this paper. It is, however, included for completeness as it was used in the paper of Adachi and Trendafilov (2018).
- Run_Simulation.R: This script estimates the parameters of the simulated data set of Adachi and Trendafilov (2018).
- Get_Sim_Parameters.R: This script defines the data generating matrices of the simulated data of Adachi and Trendafilov (2018) and is sourced in *'Run_Simulation.R'*
- Run_Big_Five.R: This script estimates the parameters for the big five personality test used in Adachi and Trendafilov (2018).
- DIST_Utils.R: This script contains the functions that estimate the different distance matrices. That is, the distance matrix obtained via the Euclidean distance, dynamic time warping (included for completeness) and the complexity invariant distance measure.
- Run_Stock_Indices.R: This script runs the moving window SSFA structure for the stock indices data.
- Run_DAX30.R: This script runs the moving window SSFA structure for the DAX30 constituents data.
- Run_Pure_SR_Indices.R: This script runs the pure Sharpe ratio based portfolio strategy for the stock indices data set.
- Run_Pure_SR_Dax.R: This script runs the pure Sharpe ratio based portfolio strategy for the DAX30 data set.