Quantifying Time Series Similarity using Singular Spectrum Analysis

Zella Baig

Mansfield College,

University of Oxford

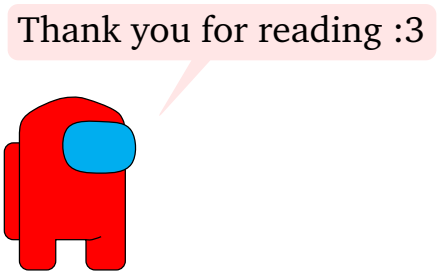
A thesis presented for the degree of

M.Sc. in Mathematical Modelling and Scientific Computing

Trinity Term, 2023

Acknowledgements

I'd like to start by thanking my supervisors Mohsin and Yuji for their continued support and help throughout this project, as well as BlackRock for the opportunity to undertake it in the first place. I'd also like to express my gratitude towards Kathryn for her help throughout the year for matters both academic and not, and especially so towards the rest of the MSc, the new friends I have made and the old ones - you have all served to make this year so enjoyable. Lastly I'd like to thank Stephen for supporting me through so much, and I'd like to finish by saying...



Contents

List of Figures

List of Tables

1 Introduction

Time series analysis arises in a wide array of contexts - from meteorology to speech recognition. A natural problem which arises in many of these use cases is in the examination and classification of *similarity*, or the question of analysing when two different time series can be quantifiably classed as being "similar" in the colloquial sense.

We are specifically, in this thesis, interested in the analysis of time series in financial contexts. We demonstrate this with an illustrative example. Consider two stocks which we expect to be similar due to perhaps operating in the same sector and having a similar userbase, or facing similar supply chain challenges. An example of this might be Lockheed Martin and Northrop Grumman, which we represent with the stock tickers LMT and NOC respectively[1]. Both these companies operate in the aerospace and defence sector, and as such we expect them to perform relatively similarly. Indeed, if we plot the returns of these over 180 trading days in Figure 1, we can see that their returns are incredibly similar over the time period plotted.

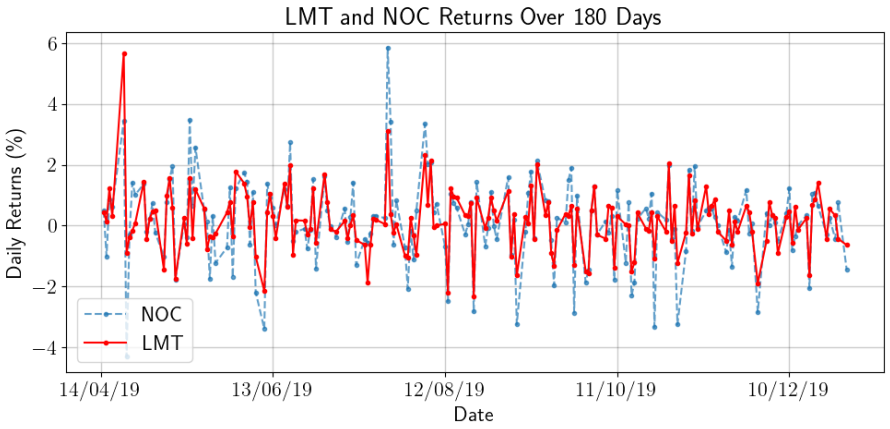


Figure 1: LMT and NOC returns over 180 trading days.

An interesting scenario now is to consider the prices for these two stocks, which we present in Figure 2. Notice how there are periods where the prices remain quite similar to one another, then diverge, and then eventually converge again. This behaviour is expected, assuming the underlying trends governing the price behaviour of these tickers is similar (which, as mentioned before, we expect it to be).

[1]Henceforth, we use the word "ticker" to denote the initialism used to represent a given company on stock exchanges. For example, the ticker V corresponds to the company Visa.

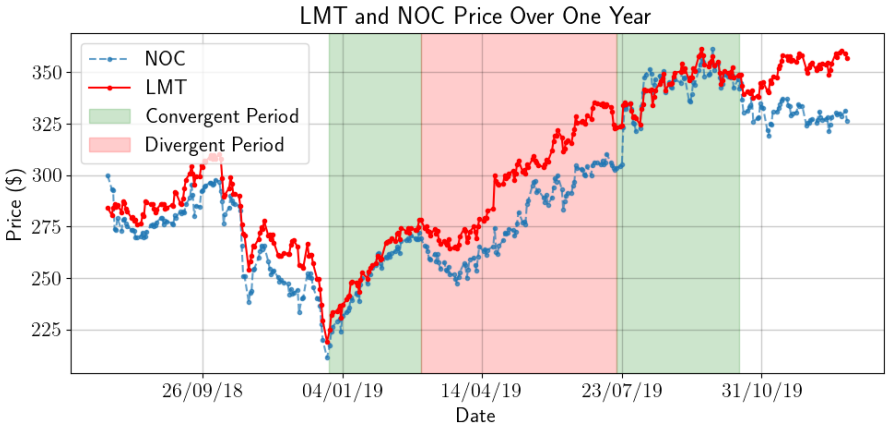


Figure 2: LMT and NOC price over one year.

If the trends governing these tickers are very similar, then the periods in which the prices diverge present an opportunity to extract value. For example, assuming we know a priori that these two tickers are similar, we wait for a divergent period when the values of these series begin to differ significantly. We then short the more expensive stock, and buy more of the cheaper stock, such that we may extract profit from their eventual reconvergence. Such trading strategies are known as "pairs trading" strategies, and can be applied to other tickers which are similar, such as Pepsi and Coca-Cola.

A subsequent issue which thus arises is of dealing with noise within data. Time series (and particularly so in financial contexts) often present themselves inclusive of a large amount of noisy movements, whereby "noise" we refer to short term, erratic fluctuations in values. Constructing a similarity measure of this noisy data is likely to lead to noisy results (giving no useful information), or even worse lead to incorrect conclusions drawn [2]. We therefore seek some method to incorporate noise reduction (or ideally, elimination) into our time series signals by extracting the useful information and discarding the useless fluctuations. This may alternatively be conceptualised as extracting the trend from a time series: the underlying signal which dominates the fluctuations in value.

To sum, the mathematical challenge in this case is twofold:

1. The overarching problem is to figure out a way to discern what time series similarity entails in a rigorous sense,

2. But this comes with the added sub-problem to tackle first in which time series

[2] See the common saying in computing: *garbage in, garbage out*.

data, particularly stock data, is often contaminated by noise and so we seek to extract the underlying trend from the time series.

To solve the second issue, we employ Singular Spectrum Analysis (SSA), a common method in time series analysis used to extract the "signal" from "noise" in a time series (Hamilton 1994). We examine this choice of technique, and outline why we employ it versus other common methods to de-noise time series data. With these "de-noised" time series, we will then be able to tackle the issue of constructing some similarity measure of the underlying signals. We provide an examination of several standard methods in time series analysis, and discuss why we prefer the Time Warp Edit Distance (TWED), discussing its performance against other metrics such as the Euclidean Distance.

In particular, our discussion presents the most recent analysis of SSA on stock data utilising recent similarity measures, and further presents a novel comprehensive discussion of the linear algebraic theory arising in these contexts (which often appears to be missing from literature). Further, we provide new insight into the behaviour of these de-noised time series, which to our knowledge has not been connected to long-term financial data sets before; we also are able to provide new examination of the relationships between the structure of the underlying signals of the time series and the singular value distributions which arise from the construction of the trajectory matrices via SSA. Lastly, many of the methods and analyses presented in this thesis are still readily applicable to other scenarios, and we hope this leads to several potential future avenues for exploration.

The layout of this thesis is as follows. We proceed first in Section 2 with a discussion of the literature within the field, and give an overview of our novel contributions. We proceed to reviewing the data set which we use for our analysis in Section 3, and then in Section 4 we discuss Singular Spectrum Analysis as well as our motivation for choosing this specific method for our use case. We move next to Section 5 where we discuss measuring similarity for time series, and then in Section 6 we discuss how to decide when two time series are *dissimilar*. We then turn to discussing how exactly we score similarity of multiple time series in Section 7, and present our results in Section 8. Lastly, we provide our conclusions in Section 9, where we finish via briefly discussing future avenues for work and questions to tackle.

2 Literature Review & Novel Contributions

Though much of the analysis on SSA has come about in the past decade, the modern methodology traces its roots via "caterpillar SSA" arising from Russian institutes in the 90s and discussed in texts such as in (V. N. Zhigljavsky and Anatoly A. 2001). In more recent years however, there has been work done on analysis of optimisation of parameters, such as in (Golyandina 2011; R. Wang et al. 2015), and novel ideas for the choice of these parameters have also been discussed such as via entropy-based approaches in (Khan and Hassani 2019). Our discussion on SSA attempts to provide further insight into the choice of one of the two parameters associated with SSA, the truncation rank r, via an analysis of the convergence to optimal results provided a set of synthetic data. Note however that this is not the primary aim of our examination of SSA and as such we leave further exploration for future work.

We also provide further insight into the analysis of the singular values associated with the matrices employed within SSA, both in terms of their empirical distributions (carrying on work as done in (Mahmoudvand and Zokaei 2011)), and particularly draw links between the work done linking SSA to Fourier mode analysis (as brought up theoretically (Guo et al. 2020; X. Zhao and Ye 2019)) to our financial context. In particular, the examination of these links helps to outline effects arising from market-level economic forces, and how they influence the distributions of the singular values associated with the matrices employed within SSA.

Links between SSA and machine learning approaches have also begun to be explored such as in (Hou, Jin, and Z. Zhao 2019; Grabocka and Schmidt-Thieme 2018), though contributions to this area have not thus far drawn much attention (though have shown promising results, for example, in forecasting).

There has also been increasing attention drawn not just to SSA in a vacuum, but in examining the ramifications of the method in specific context, such as looking at tourism trends in (Hassani, Webster, et al. 2015), image processing in (Rodríguez-Aragón and A. Zhigljavsky 2010), and specifically for financial data in (Hassani and Thomakos 2010). However, our discussion is thus far both the widest ranging contextually (such as in examination of sector-level information) but also the most in-depth in terms of answering the question of how to go about picking similar time series from a (large) set of very noisy data, constructed over a period of several years.

We also briefly delve into the examination of related techniques such as in Proper Orthogonal Decomposition (Frame and Towne 2022; Weiss 2019), and demonstrate the application of linked techniques to improve computational performance of "standard"

SSA.

Lastly, in SSA, we (with much support from Professor Yuji Nakatsukasa) provide further exploration in the links between low-rank functions, the Hankelised matrices they form, and their Chebyshev expansions - with the novel experimental results presented in Section 8.1 providing clear avenues for further exploration.

Within time series the standard method for similarity analysis appears to be Dynamic Time Warping, known as an "elastic" measure due to its ability to account for lead-lag effects (Goldstein 2015; Zha et al. 2022). However, we examine various discussions of the choice of both the similarity measures used (showcasing their benefits and drawbacks as in (Serrà and Arcos 2014; Vlachos 2017; X. Wang et al. 2010)), as well as the pre-processing required in these contexts as done in (Rakthanmanon et al. 2013; Elzinga and Studer 2019; Hassani, Yeganegi, et al. 2020); we also incorporate information specific to financial data as in (Lütkepohl and Xu 2012) showcasing the necessity of appropriate preprocessing.

Lastly, we bring up an important distinction which appears to be missing in almost all of the time series similarity analysis which we have seen: that the notions of "similarity" and "distance", while closely linked, are not precisely inverse to one another (as they are treated in the bulk of the literature). This concept has been discussed primarily sequence processing such as in text analysis as in (Elzinga 2014), but drawing on insights discussing the relationships between these two closely related concepts (as in (Chen, ma, and Zhang 2009; Emms and Franco-Penya 2013)) we are able to show that indeed the precise definition of similarity employed does matter in time series processing (and when it might not).

Having discussed the background for our work, we turn first towards analysing the data set that we utilise in our discussion.

3 Data Set

We have built a webscraper which parses five years worth of data (or 1258 entries) from Yahoo Finance (*Yahoo Finance – stock market live, quotes, business & finance news* 2023) from the S&P 500, a collection of the five hundred largest tickers listed within the United States. Once collecting the data over from 2015 to 2020, we then remove any tickers which did not exist within the S&P 500 for the entirety of this range due to acquisitions or market capitalisation changes. This left us with 477 tickers to work with for this five year span. Note that we then also parsed information about each of these tickers' GICS sectors (of which there are 11) and sub-sectors (of which there are 163); these GICS labels being standard independent labels applied to stock tickers pertaining to the specific sector a ticker operates in, to four levels of specificity (*GICS® - Global Industry Classification Standard* 2023).

We also note that it is discussed in, for example, (Keogh and Kasetty 2003) and (Rakthanmanon et al. 2013) that normalisation is needed when discussing time series data in order to draw meaningful comparisons; financially it is also standard practice to do so to attempt to get variables on similar scales. We see in (Hassani, Yeganegi, et al. 2020) that in time series *forecasting* Z-normalisation (subtracting the mean and dividing by the standard deviation) slightly outperforms log normalisation (another de-facto standard in financial data); the results are not conclusive by any means and furthermore there does not appear to be a set method for data normalisation when considering time series. We opt for Z-normalisation due to the ability to set variables to a similar scale regardless of initial values such that we have a set "domain" within which to examine similarity, as opposed to trying to examine values which are both very large and very small (note that log normalisation rectifies this issue somewhat, but does not fully solve it given our data set includes tickers with both extremely high and fairly low prices).

Having discussed the data which we use, we now look at the mathematical background for the work we undertake.

4 De-Noising & Extracting Trends from Data

4.1 Singular Spectrum Analysis

Before delving into the mathematical techniques used in the extraction of trends for our time series, let us first define the notion of a time series precisely.

Definition 1 (Time Series)**.** *A time series Z_T is a set of T ordered points $z_1, z_2, \dots z_T$, each associated with some given measurement.*

This definition then provides us with the groundwork upon which to then analyse our forthcoming methodologies. However, before turning towards our examination of Singular Spectrum Analysis, we introduce the tools which we need to perform this analysis.

4.1.1 Prerequisite Linear Algebra

Starting with a review of some tools from linear algebra, we first define define *Hankel matrices*, which we use later on in the SSA method.

Definition 2 (Hankel Matrices)**.** *An $m \times n$ Hankel matrix is a matrix that has the same values on the off-diagonals, i.e. of the form*

$$\begin{bmatrix} z_1 & z_2 & z_3 & \cdots & z_n \\ z_2 & z_3 & z_4 & \cdots & z_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_m & z_{m+1} & z_{m+2} & \cdots & z_{m+n-1} \end{bmatrix}$$

for entries $\{z_i\}_{i=1}^{m+n-1}$.

Next, we discuss the *singular value decomposition* of a matrix, a decomposition heavily employed in signal processing and further contexts.

Definition 3 (Singular Value Decomposition (Nakatsukasa 2023))**.** *Given any matrix $A \in \mathbb{R}^{m \times n}$ for $m \geq n$[3], we may decompose A into the product of three different matrices,*

$$A = U \Sigma V^T,$$

where $U \in \mathbb{R}^{m \times n}$ is orthonormal ($U^T U = I_n$), $V \in \mathbb{R}^{n \times n}$ is orthogonal ($V^T V = V V^T = I_n$), and $\Sigma \in \mathbb{R}^{n \times n} = diag\{\sigma_1, \dots, \sigma_n\}$, with $\sigma_i \leq \sigma_j$ if $i \leq j$.

Here, the columns of U are called the left singular vectors, the columns of V are called the right singular vectors, and the values σ_i are called the singular values.

[3] A straightforward analogue exists for the case where $m < n$ by considering A^T.

Proof. For any such matrix A, construct the Gram matrix A^TA, which has the decomposition

$$A^TA = V\Lambda V^T, \tag{1}$$

where the matrix Λ is the (diagonal) matrix of eigenvalues λ_i of A^TA. Importantly, A^TA is symmetric and so the matrix Λ is non-negative, and therefore we may rearrange (1) by

$$V^TA^TAV = \Lambda := \Sigma^2, \tag{2}$$

making $\sigma_i := \sqrt{\lambda_i}$. From (2), we may then right- and left-multiply by Σ^{-1} to arrive at

$$(\Sigma^{-1}V^TA^T)(AV\Sigma^{-1}) := I_n, \tag{3}$$

where the bracketed terms in (3) we define as the matrix U. □

A more relevant form we may write the SVD in is via

$$U\Sigma V^T := \sum_{i=1}^{n} \sigma_i(u_i v_i^T), \tag{4}$$

where u_i and v_i are the left and right singular vectors respectively. Importantly, note that the sum in (4) is a sum of n rank-1 matrices, each of the dimension $m \times n$. Given the guarantee of its existence (Nakatsukasa 2023), the SVD has been often used as a measurement of the "information" contained within a given matrix, for example in low-rank approximations. This is done by simply "cutting off" the singular values at some chosen rank r; in the form of (4) this is simply equivalent to truncating the sum at $i = r$.

While there exist many standard in-built libraries to compute the SVD within Python, we make a note that these methodologies might prove too costly for both our data set, and future consideration of larger data sets as well, owing to the $O(mn\min\{m,n\})$ cost of calculation on a matrix of size $m \times n$. We thus turn to *randomised* methods, using a self-coded implementation of an existing algorithm. Randomised methods have proven to be both incredibly popular and effective in the field of data science, with large speed increases in algorithms at the cost of relatively small errors: we believe the performance increases we attain using these methods to be worth implementing them for our purposes.

Definition 4 (Rangefinder Algorithm (Halko, Martinsson, and Tropp 2011))**.** *Suppose we wish to find the rank-r approximation ($\hat{A}$) of a matrix $A \in \mathbb{R}^{m\times n}$. To do so, we follow the procedure outlined in Pseudocode 1, as follows.*

Pseudocode 1: Rangefinder SVD

def *Rangefinder*($A \in \mathbb{R}^{m\times n}$)**:**

$G = gaussian(m,n)$; # $G \in \mathbb{R}^{m\times n}$ with $G_{i,j} \sim N(0,1)$

$B = AG$;

$Q, R = qr(B)$; # QR factorisation of B, with $Q \in \mathbb{R}^{m\times r}$

$\hat{A} = QQ^T A$; # Construct low-rank approximant of A

return $\hat{A}$;

We can see that QQ^TA is the low-rank approximant by first considering the matrix $C := Q^TA \in \mathbb{R}^{r\times n}$, and then computing the singular value decomposition $C = \bar{U}\Sigma_r V^T$, which is rank-r. We then form $U := Q\bar{U} \in \mathbb{R}^{m\times r}$, and so $QQ^TA = QC = Q\bar{U}\Sigma_r V^T = U\Sigma_r V^T$.

Now, starting with a noisy time series Z_T, we first define the *window length* L and thus the corresponding variable $K := T - L + 1$. We are now in a place to construct an outline for SSA. We then form the associated (Hankel) *trajectory matrix*

$$\mathbf{X} := \begin{bmatrix} z_1 & z_2 & z_3 & \cdots & z_K \\ z_2 & z_3 & z_4 & \cdots & z_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_L & z_{L+1} & z_{L+2} & \cdots & z_T \end{bmatrix},$$

which can be either "tall" or "wide", depending on the choice of the window length (which sets the height of the matrix) and the length of the time series T.

We pause to discuss the intuition behind Hankel matrices. Each of the columns in the Hankel matrix is simply the preceding column "rolled" back by one: discarding the oldest observation and adding the newest observation. This type of structure is commonly referred to as in literature as *time-delay embedding* (Frame and Towne 2022), with the delay in the Hankel matrix case being a single observation. In this sense, the columns can be thought to represent a set of vectors in time-space, each corresponding to a single set of observations (and thus the singular values correspond to information contained contained in the time-space spanned by the columns of the trajectory matrix, or in other words the trends which are present across the span of the time-delayed column vectors).

We then compute the SVD of the matrix $\mathbf{X}$, which we represent as a sum of rank-1 matrices:

$$\mathbf{X} = \sum_{i=1}^{L} \sigma_i (u_i v_i^T). \tag{5}$$

We now truncate the sum in (5) to some chosen rank r, with $r \leq L$. Our motivation is such:

1. We know that the initial signal Z_T may be written as the sum of a "pure" signal, which we denote by S_T, and noise, which we denote by N_T.

2. We also expect that the pure signal is of greater magnitude than the noise (i.e. that the signal dominates the noise component).

3. Therefore, in an ideal scenario we would be able to label the largest r singular values as corresponding solely to the signal and the rest of the singular values as corresponding to noise. In this sense, the "truncated" SVD that we work with is indeed an approximation to the overall signal, but can also be thought of as providing a more accurate representation of the underlying time series dynamics.

Thus, let us denote this truncated SVD by

$$\mathbb{X} := \sum_{i=1}^{r} \sigma_i (u_i v_i^T). \tag{6}$$

Note that the matrix $\mathbb{X}$ in (6) is not necessarily Hankel; this is a non-ideal situation simply due to the fact that we have already used a one-to-one mapping of a time series and a Hankel matrix in the construction of the initial trajectory matrix $\mathbf{X}$. Thus, we perform a re-Hankelisation procedure on $\mathbb{X}$ (by taking the mean of every off-diagonal, and resetting all values on that off-diagonal to the mean) to then generate a "reconstructed" trajectory matrix $\bar{\mathbf{X}}$ (with corresponding reconstructed series $\bar{Z}_T$),

$$\bar{\mathbf{X}} := \begin{bmatrix} \bar{z}_1 & \bar{z}_2 & \bar{z}_3 & \dots & \bar{z}_K \\ \bar{z}_2 & \bar{z}_3 & \bar{z}_4 & \dots & \bar{z}_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{z}_L & \bar{z}_{L+1} & \bar{z}_{L+2} & \dots & \bar{z}_T \end{bmatrix}.$$

We may also explore the connections this truncation has to image approximations, another field truncated SVDs have historically shown to be useful in. For example, consider Figure 3, where we present the initial rank-240 trajectory matrix formed from a sinusoidal signal with added Gaussian noise, as well as the rank-25 reconstruction formed via truncation and re-Hankelisation. Note in particular that with a large reduction in rank, we are able to remove most of the "noise" present in the data but retain the bulk of the underlying signal.

In fact, however, we do not need to reconstruct the entire Hankel matrix $\bar{\mathbf{X}}$, and

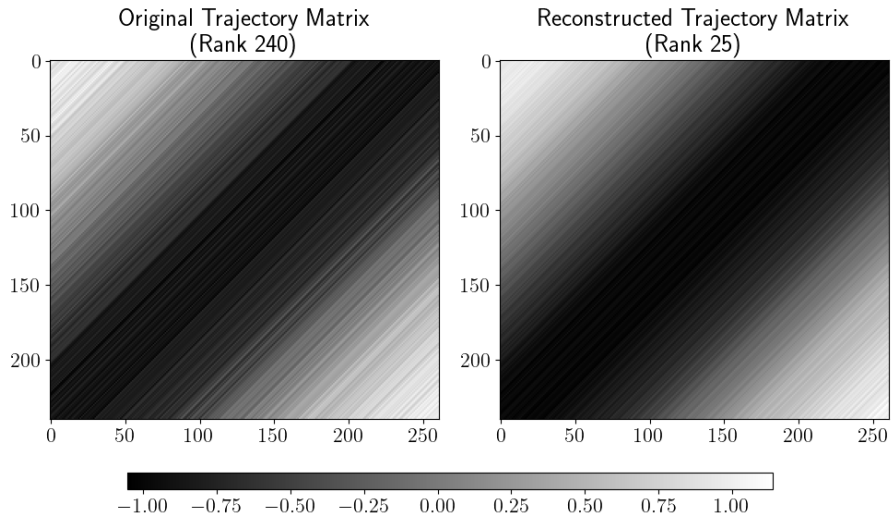


Figure 3: Trajectory matrix for AAPL price over five years (2015 to 2020), alongside its rank-25 SSA reconstruction.

instead we may compute the elements of the reconstructed series directly via

$$\bar{z}_p = \begin{cases} \dfrac{1}{p}\displaystyle\sum_{i=1}^{p} \mathbb{X}_{i,p-i+1} & 1 \le p \le L, \\ \dfrac{1}{L}\displaystyle\sum_{i=1}^{L} \mathbb{X}_{i,p-i+1} & L < p < K, \\ \dfrac{1}{T-p+1}\displaystyle\sum_{i=T-K+1}^{L} \mathbb{X}_{i,p-i+1} & K \le p \le T. \end{cases}$$

In an ideal situation, the parameters L and r would have been set such that our reconstruction $\bar{Z}_T$ has as little noise as possible; we now turn to discussing the choice of these parameters.

4.1.2 Choice of Parameters

Note that we have two parameters to set within the SSA procedure: the window length L, as well as the rank of the truncation r. We focus first on the choice of L, and then turn to choosing the latter.

The standard methodology used to discuss the choice of window length is the notion of *separability*. Intuitively, we may think of this concept as pertaining to whether, for a given composite signal $C = A + B$ with sub-signals A, B we can "separate" the two signals from one another. This relates directly to SSA in which ideally, we wish to separate the

noise from signal within our time series. To quantify the notion of separability, we first define the *weighted inner product* (V. N. Zhigljavsky and Anatoly A. 2001) for two time series Z_T and Y_T (and with fixed L) as

$$(Z_T, Y_T)_w := \sum_{i=1}^{T} \min\{i, L, T-i+1\} z_i y_i, \tag{7}$$

which has the associated norm

$$\|Z_T\|_w^2 := (Z_T, Z_T)_w .$$

Note in particular that the highlighted term in (7) (henceforth referred to as w_i^L in shorthand) is simply the amount of times that the ith time observation of a time series of length T would appear when embedded into a trajectory matrix of window length L; this weight can thus be loosely thought of as a weight in "temporal" space. With this, we then define the *weighted-correlation* (V. N. Zhigljavsky and Anatoly A. 2001), or in other words the w-correlation as

$$\rho_w(Z_T, Y_T) := \frac{(Z_T, Y_T)_w}{\|Z_T\|_w \|Y_T\|_w}, \tag{8}$$

where a ρ_w value of zero implies w-orthogonality (or complete separation), and a value of one implies that both signals Z_T, Y_T are w-parallel, or completely "mixed". We thus seek the value of L which minimises (8). In what follows, we follow a similar procedure as in (Hassani, Mahmoudvand, and Zokaei 2011) in order to show that the optimal window length to choose is $L := \frac{T+1}{2}$ for a time series of length T. However, before we proceed to constructing our main proof, we first prove several lemmas which we employ as part of the overarching proof.

Lemma 1 (Restriction of w-correlation domain (Hassani, Mahmoudvand, and Zokaei 2011))**.** *Consider a time series $Z_T = S_T + N_T$, where S_T is the signal and N_T is the noise. Define $\boldsymbol{X}_L$ and $\boldsymbol{X}_K$ as the trajectory matrices formed with window lengths L and K of the time series Z_T, the associated reconstructed series $\bar{S}_T(L), \bar{S}_T(K)$, and the associated noise components $\bar{N}_T(L), \bar{N}_T(K)$, defined by $\bar{N}_T(L) = Z_T - \bar{S}_T(L)$.*

We then have that $\bar{S}_T(L) = \bar{S}_T(K)$ and $\bar{N}_T(L) = \bar{N}_T(K)$, and so $\rho_w(\bar{S}_T(L), \bar{N}_T(L)) = \rho_w(\bar{S}_T(K), \bar{N}_T(K))$, meaning that we only consider window sizes $L \in [2, \frac{T+1}{2}]$.

Proof. Note that $\boldsymbol{X}_L = (\boldsymbol{X}_K)^T$ via the definition $K = T - L + 1$, from which the results follow directly. □

Lemma 2 (Trace of product of reconstructed matrices (Hassani, Mahmoudvand, and Zokaei 2011))**.** *With $\boldsymbol{X}$ as the $L \times K$ trajectory matrix for a time series Z_T for $K =$*

$T-L+1$ and $L \in [2, \frac{T+1}{2}]$, define $\boldsymbol{X} = \boldsymbol{S}+\boldsymbol{N} = \bar{\boldsymbol{S}}+\bar{\boldsymbol{N}}$, with $\boldsymbol{S}$ the rank-r component of $\boldsymbol{X}$ corresponding to signal, $\boldsymbol{N}$ corresponding to noise, and the barred components the corresponding reconstructions made during the SSA procedure (following directly from Lemma 1). We then have that $tr(\boldsymbol{S}\bar{\boldsymbol{N}}^T; L) > 0$, where $tr(\boldsymbol{A}; L)$ denotes the trace of an $L \times K$ matrix $\boldsymbol{A}$.

Proof. First define the operator $tr_T(\boldsymbol{X}; L) := tr(\boldsymbol{X}\boldsymbol{X}^T; L)$. Now note that

$$tr_T(\boldsymbol{S}+\boldsymbol{N}; L) = tr_T\left(\bar{\boldsymbol{S}}+\bar{\boldsymbol{N}}; L\right) \tag{9}$$

since the arguments on both sides is simply $\boldsymbol{X}$. Next, by expanding both sides of (9) explicitly and rearranging, we have

$$tr\left(\bar{\boldsymbol{S}}\bar{\boldsymbol{N}}^T; L\right) = \frac{1}{2}\left[tr\left(\boldsymbol{S}\boldsymbol{S}^T; L\right) - tr\left(\bar{\boldsymbol{S}}\bar{\boldsymbol{S}}^T; L\right) + tr\left(\boldsymbol{N}\boldsymbol{N}^T; L\right) - tr\left(\bar{\boldsymbol{N}}\bar{\boldsymbol{N}}^T; L\right)\right]. \tag{10}$$

Now, consider $tr(\boldsymbol{S}\bar{\boldsymbol{S}}^T; L)$, which we expand as

$$tr(\boldsymbol{S}\bar{\boldsymbol{S}}; L) = \sum_{i=1}^{T+1}\sum_{j=i_0}^{i_1} s_{i,i-j}\bar{s}_{i,i-j} = \sum_{i=1}^{T+1}\sum_{j=i_0}^{i_1} s_{i,i-j}\bar{s}_i = \sum_{i=1}^{T+1} w_{i-1}^L \bar{s}_i^2 = tr\left(\bar{\boldsymbol{S}}\bar{\boldsymbol{S}}^T; L\right), \tag{11}$$

again using the weight from (7) in our expression. Using the results of (11), we note that

$$tr_T\left(\boldsymbol{S}-\bar{\boldsymbol{S}}; L\right) = tr\left(\boldsymbol{S}\boldsymbol{S}^T; L\right) - tr\left(\bar{\boldsymbol{S}}\bar{\boldsymbol{S}}^T; L\right), \tag{12}$$

where the left hand side is positive, and therefore so is the right hand side. We can repeat this analysis for $\boldsymbol{N}$ and $\bar{\boldsymbol{N}}$ to thus show that the right hand side of (10) is positive, completing the proof. □

Lemma 3 ($tr_T(\bar{\boldsymbol{S}}; L)$ an increasing function (Hassani, Mahmoudvand, and Zokaei 2011))**.** *$tr_T(\bar{\boldsymbol{S}}; L)$ is an increasing function of L for $L \in [2, \frac{T+1}{2}]$, provided there exists a Hankel matrix $\boldsymbol{H} \in \mathbb{C}^{L\times K}$ such that*

$$tr_T(\boldsymbol{S}-\boldsymbol{H}; L) \leq tr_T(\bar{\boldsymbol{S}}; L) - tr_T(\bar{\boldsymbol{S}}; L-i) \tag{13}$$

for $i \in [1, L-2]$.

Proof. Recall expression (12). Note by definition of $tr_T(\boldsymbol{A}; L)$ for a Hankel matrix $\boldsymbol{A}$ that $tr_T(\boldsymbol{S}-\bar{\boldsymbol{S}}; L) \leq tr_T(\boldsymbol{S}-\boldsymbol{A}; L)$ for all Hankel $\boldsymbol{A} \in \mathbb{C}^{L\times K}$. Assuming now that we have a matrix $\boldsymbol{H}$ which satisfies (13), it follows immediately that

$$tr_T(\boldsymbol{S}; L) - tr_T(\bar{\boldsymbol{S}}; L) \leq tr_T(\boldsymbol{S}-\boldsymbol{H}; L) \leq tr_T(\bar{\boldsymbol{S}}; L) - tr_T(\bar{\boldsymbol{S}}; L-i),$$

and so we have completed the proof. □

With this work, we are now able to prove Theorem 1.

Theorem 1 (Optimal window length for SSA, adapted from (Hassani, Mahmoudvand, and Zokaei 2011))**.** *Consider a time series Z_T (with corresponding $L \times K$ trajectory matrix $\boldsymbol{X}$, for $K = T - L + 1$), comprised of the sum of the signal series S_T and noise series N_T (and which may also be written as the sum of the reconstructed series $Z_T = \bar{S}_T + \bar{N}_T$). We then may construct the corresponding signal matrix $\boldsymbol{S}$, the reconstructed signal matrix $\bar{\boldsymbol{S}}$, the noise matrix $\boldsymbol{N}$, and reconstructed noise matrix $\bar{\boldsymbol{N}}$.*

Then, the choice of window length L which minimises the w-correlation between S_T and N_T is $\frac{T+1}{2}$, provided that there exists a Hankel matrix $\boldsymbol{H}$ such that

$$tr_T(\boldsymbol{S} - \boldsymbol{H}; L) \leq tr_T(\bar{\boldsymbol{S}}; L - i) - tr_T(\bar{\boldsymbol{S}}; L - i) \tag{14}$$

for $i \in [1, L-2]$.

Proof. Consider the outline of the proof of Lemma 3. Apply the same procedure to demonstrate that $tr_T(\boldsymbol{S} - \bar{\boldsymbol{S}}; L)$ is a decreasing function of L, attaining a minimum at $L = \frac{T+1}{2}$, provided there exists a Hankel matrix $\boldsymbol{H}$ which satisfies (14). Repeat this procedure to show that $tr_T(\boldsymbol{N} - \bar{\boldsymbol{N}}; L)$ is a decreasing function of L as well. Finally, recalling Expression (10), note that the right hand side is a decreasing function of L and thus that $tr(\bar{\boldsymbol{S}}\bar{\boldsymbol{N}}^T; L)$ is a decreasing function of L; noting in particular that

$$tr(\bar{\boldsymbol{S}}\bar{\boldsymbol{N}}^T; L) = \sum_{i=1}^{L}\sum_{j=1}^{K}(\bar{\boldsymbol{S}})_{i,j}(\bar{\boldsymbol{N}})_{i,j} = \sum_{i=1}^{T} w_i^L (\bar{S}_T)_i (\bar{N}_T)_i = (\bar{S}_T, \bar{N}_T)_w,$$

thereby completing the proof □

Using this result, we henceforth set the window length to $\frac{T+1}{2}$ unless we state otherwise.

Choosing the rank of truncation r is more difficult. While there have been attempts such as in (Khan and Hassani 2019) to choose optimal parameters for r, there has yet to have been consensus on a tried-and-true method which gives a reliable extraction of the dominant underlying signals.

We turn instead to an approach more based upon the specific data set we have, rather than one more generally applicable: via an examination of the scree plot of singular values of the trajectory matrices corresponding to all the time series in our data set. Whilst we note that such an approach may be suboptimal, nevertheless it is highly adaptable to other contexts and further (as we shall see further on) provides reasonably good results in terms of de-noising our time series.

Consider the scree plot in Figure 4. Clearly, the first few singular values tend to account for the bulk of the information within these matrices, and so we set a (somewhat arbitrary) threshold of $r = 25$ as our rank of SSA truncation.

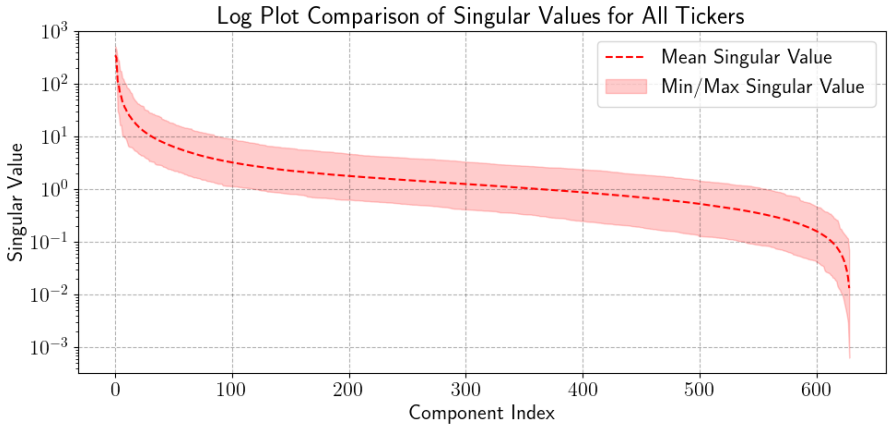


Figure 4: Scree plot of all singular values for tickers' trajectory matrices.

The overall shape of the singular value distribution also appears to be interesting in that after the first few very large singular values, there appears to be a large region wherein the singular values slowly decay, and we ultimately then arrive at a steep drop off at the very tail of the singular value distribution. We attempt to make experimentally-backed claims linking the shape of this distribution to connections to market forces in the following section. This also presents clear avenues for future work, namely in theoretically backing our conclusions up further.

4.2 Market Connections within SSA

Looking at the time series in our data set, we observe that they largely display the same overall shape since each stock is affected by the overall market. We claim that the first few large singular values correspond to these major market sentiments, which then helps to explain why these singular values are so much larger than the others (as market movements on average will dominate any day-to-day fluctuations in individual price for any ticker). We justify these thoughts as follows.

We first define by "band width" the value $\tilde{w}_i := \max\{\tilde{s}_i\} - \min\{\tilde{s}_i\}$, where $\tilde{s}_i$ is the reconstructed time series associated with the ith singular value, i.e. $\tilde{s}_i \sim \sigma_i(u_i v_i^T)$. In this sense, the band width is an indication of the magnitude of movements associated with the ith singular value.

We can now consider a plot of the band width against the index of the corresponding singular value, as presented in Figure 5.

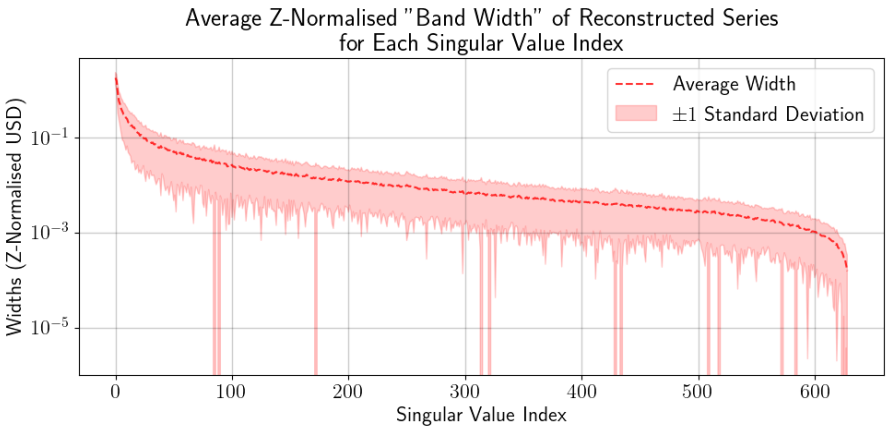
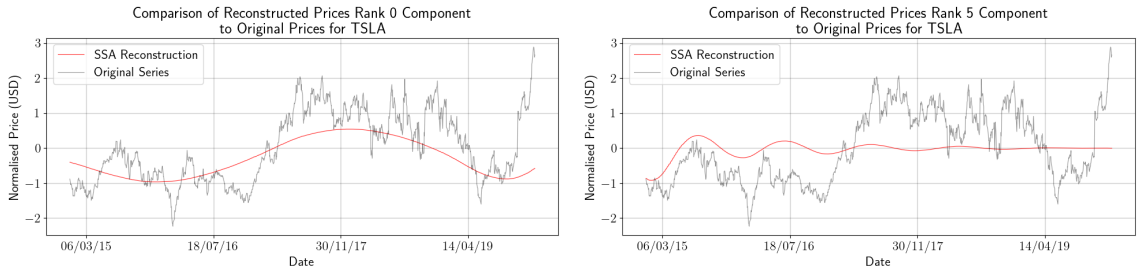


Figure 5: Comparison of "band width" of SSA reconstructions.

We see here that $\tilde{w}_i$ drops off sharply after the first few singular values for our data set; this demonstrates that the first few singular values are the ones which contribute most to the fluctuations in price for our tickers, making it likely that these first few singular values correspond to dominant trends which are seen regardless of sector or even ticker.

We can also demonstrate this more concretely with a specific example. Consider Figure 6, where we plot two different (rank zero and rank five) SSA components for TSLA.



(a) Rank 0 SSA component of TSLA reconstruction. (b) Rank 5 SSA component of TSLA reconstruction.

Figure 6: Comparison of SSA components of different ranks for TSLA.

Notice how the oscillations in Figure 6(a) are far larger than those of Figure 6(b),

in accordance with our preceding discussion[4]. We can also consider the effect of these individual rank components on the overall reconstruction further in Figure 23 given in Appendix A, which plots both these series for three increasing ranks on TSLA price.

We pause to introduce the notion of Fourier frequencies. The Fourier transform is a commonly used method to link time and frequency space by transforming between the two. Such analyses are popular particularly in signal processing, but we wish to introduce this concept to help further explain our market connections. Intuitively, we expect that market movements are in general seen at a much longer scale than any noisy movements, and so they should be associated with smaller frequencies within any signal.

To quantify the proceeding analysis, suppose we have a signal $f(t)$ with period T. Using the representation of sinusoids as complex exponentials., we can represent this signal[5] as a sum of complex exponentials

$$f(t) = \sum_{n=-\infty}^{\infty} \alpha_n \exp\left\{\frac{2\pi n t j}{T}\right\},$$

for α_n complex constants and j the imaginary unit. To determine the values α_n associated with a frequency n/T, we compute the integral

$$\alpha_n = \frac{1}{T}\int_T f(t)\exp\left\{-\frac{2\pi n t j}{T}\right\}\, dt. \tag{15}$$

Thus, by sampling discrete observations of a signal $f_0, \dots f_{N-1}$, we can then construct the *Discrete Fourier Transform* signal $F_0, \dots F_{N-1}$ by discretising (15) via

$$F_m = \sum_{n=0}^{N-1} f_n \exp\left\{-\frac{2\pi n m j}{N}\right\},$$

with the values $(F_m)^2$ representing the power spectrum of our signal (thus leading directly to the associated frequencies).

We may examine the dominant Fourier frequencies for the SSA reconstructions at a given rank in our data, which we present in Figure 7. Note here that the dominant frequency associated with $\tilde{s}_i$ is a strictly increasing function with index i. This is noteworthy particularly as this demonstrates that the first few singular values are also the ones corresponding to the slowest varying fluctuations within the SSA reconstruction. This is precisely the behaviour that we expect to see if our hypothesis is true, given that market forces vary slower than any individual ticker's values. We may also independently arrive at these conclusions by considering related work done in (X. Zhao and Ye 2019), wherein

[4]Note that we index from zero.

[5]Noting that we simply perform a sinusoidal expansion of our initial signal.

they demonstrate that higher amplitudes of frequencies correspond to larger singular values.

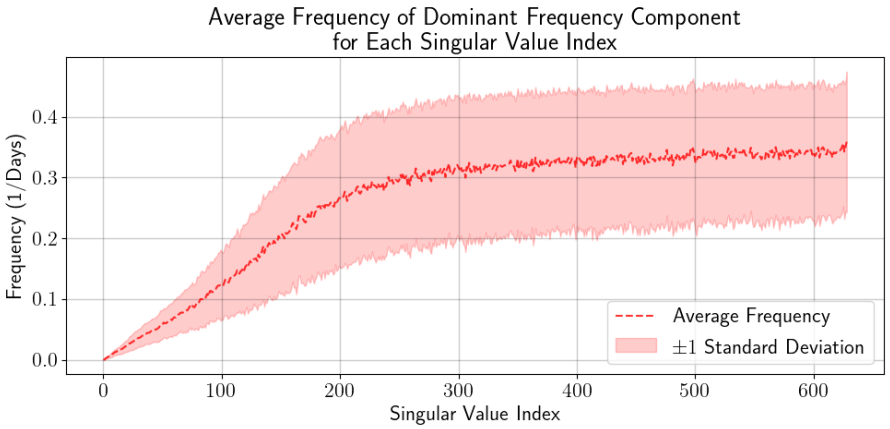
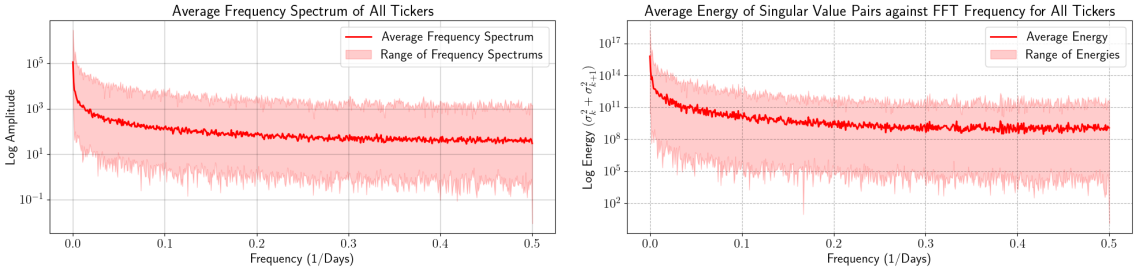


Figure 7: Comparison of DFT frequency plots for all tickers in data set.

Indeed, we may utilise this result as well as their Figure 8 (comparing energies against frequencies) and present them in our Figure 8.



(a) Frequency spectrum amplitudes for all tickers.

(b) Singular value energies against frequencies (modification of Figure 8 in (X. Zhao and Ye 2019) to our data set).

Figure 8: Comparison of frequencies and singular value energies for all tickers.

Here, we demonstrate that for all tickers the slowest varying frequencies are the ones that arise the strongest when we consider frequency space of the signals, and also that the singular value energies (i.e. sums of pairs[6] of squared singular values) are a decreasing function of frequency. This means that as our singular values drop off,

[6]One of the results discussed in (X. Zhao and Ye 2019) is that a single frequency mode corresponds to two singular values for a given Hankelised signal.

we are able to conclusively link them to faster oscillations in our SSA reconstructions; we claim that it is thus natural that these correspond to the fast random movements in price inherent to price time series. In this sense the SSA procedure in our context can be seen as an analogue of a low-pass filter, wherein we keep the low frequency "useful" signals and discard the higher frequencies. Note again, however, that this is a relatively empirical examination of this phenomenon (though we believe we have presented a strong argument for why it should be true) and so there is an avenue for future exploration in rigorously proving these results.

4.3 On the Choice of SSA

A natural question which might arise at this point is to discuss why we have chosen to proceed with SSA-based approaches, as opposed to other de-noising techniques (such as regression-based approaches, or via principal component analysis). The primary advantages are twofold.

1. First, SSA as a technique is not very dependent on assumptions on the underlying data. There is no requirement for stationary data[7], nor for the behaviour of the time series at extrapolated points. This, combined with the need for only two parameters (the window length and the rank of truncation) makes it quite appealing in terms of simplicity and efficacy to implement.

2. Secondly, primarily contrasting with PCA, is that SSA enables the examination of temporal information via the structure of the Hankel matrix $\mathbf{X}$, allowing the splitting of the signal into oscillatory and trend components; PCA on the other hand does *not* include this structure within the construction of its data matrix, and so is only able to extract the subspace with the highest variance; there is no inherent temporal information embedded into this space.

Having discussed our choice for de-noising time series, we now turn towards discussing a modification of SSA which presents interesting theoretical results and may be of use towards other contexts as well, via insight gathered from analysis of the Proper Orthogonal Decomposition.

[7] Stationary processes have constant means and variances in time (Hamilton 1994).

4.4 Proper Orthogonal Decomposition

The Proper Orthogonal Decomposition (POD) is a technique most commonly discussed in fluid flows which contains many similarities to SSA[8]. The basic outline of POD consists of writing the vector field of flows $\phi(\vec{x}, t)$ as a sum over the individual modes of flows,

$$\phi(\vec{x}, t) = \sum_{i=1}^{r} \psi_i(\vec{x})\alpha_i(t). \tag{16}$$

In the general case, we then proceed to take n snapshots at m different time values over the field with each snapshot denoted $\phi(x_i, t_j)$ for $i \in [1, n], j \in [1, m]$, and represent them in the matrix

$$\Phi := \begin{bmatrix} \phi(x_1, t_1) & \phi(x_2, t_1) & \dots & \phi(x_n, t_1) \\ \phi(x_1, t_2) & \phi(x_2, t_2) & \dots & \phi(x_n, t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(x_1, t_m) & \phi(x_2, t_m) & \dots & \phi(x_n, t_m) \end{bmatrix},$$

after which we define $\Psi := \frac{1}{m-1}\Phi\Phi^T$ as the associated correlation matrix. The eigenvectors of this matrix Ψ are then the modes ψ_i in (16); the coefficients α_i are found via taking the inner product of ϕ with the corresponding mode.

Specifically considering discrete approximations to ergodic systems, one way of constructing the matrix Φ is to take a time series of snapshots $\{\phi_i\}_{i=1}^{n+m-1}$, and input them into the matrix

$$\Phi := \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_n \\ \phi_2 & \phi_3 & \cdots & \phi_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_m & \phi_{m+1} & \cdots & \phi_{m+n-1} \end{bmatrix},$$

with the implicit assumption that the flow at $(x_2, t_1) = (x_1, t_2)$ etc. Intuitively, this may be thought of as the structure of the underlying flow "moving through" the spatial components of the field. Note now that the elements of Ψ may be written as $(\Psi)_{i,j} = \sum_{k=1}^{n} (\Phi)_{i,k}(\Phi)_{k,j}$; the parameter n thus represents the number of products summed over to construct an element of the correlation matrix.

For intuition, consider a very tall, thin matrix with $n << m$. The elements of Ψ are thus comprised of relatively small summations and thus would not be very accurate realisations of the "true" underlying correlations; similarly as we increase n the accuracy of our approximation increases. Further, note that the columns of the matrix Φ represent

[8] Note that we use the formulation discussed in (Iungo and Lombardi 2011).

more observations to be considered in the construction of the correlation matrix, and so as the time-difference along the columns tends to zero we approach much higher correlations and vice versa. One can thus imagine a scenario in which certain times are "dense" in information and other times are "sparse". Indeed, the authors of (Frame and Towne 2022) discuss discarding columns of the matrix Φ to ensure the columns are less correlated and for computational benefit, but we add on that such a technique if applied to regions of temporal-information sparsity (i.e. where certain columns in the trajectory matrix do not correspond to much novel information) we can potentially attain very even better results than the method proposed. Such procedures would no doubt be useful when dealing with longer time series that have large periods where certain trends are felt by all time series. For a visual example of POD-based SSA, consider Figure 9, where we compare it to "classical" SSA on AAPL price.

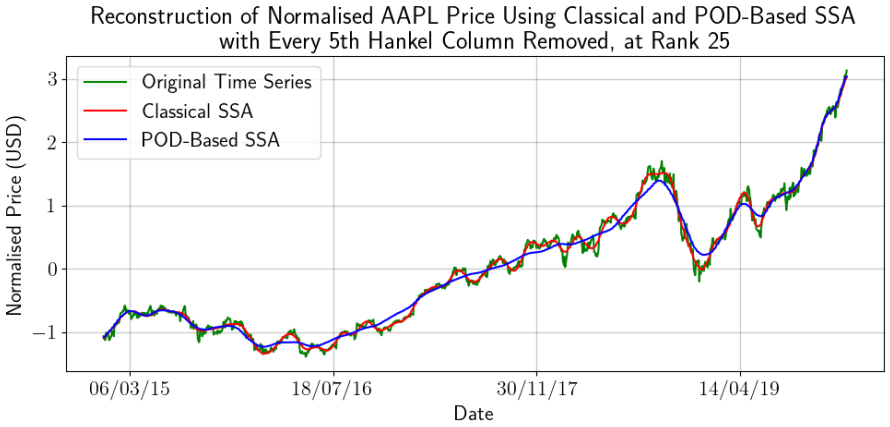


Figure 9: Comparison of POD-based and classical SSA reconstructions of a normalised-price time series for AAPL stock over five years.

Here, we have simply discarded every fifth column (as a naive demonstration of the capabilities of POD-based SSA). This implementation leads to a computational cost $\sim 64\%$ of classical SSA's due to the computational cost of the SVD on a matrix of size $n \times m$ being $O(nm \min(n, m))$, at the cost losing finer data fluctuations (which, depending on contexts, might be desirable to retain). Nevertheless, we do not examine this idea in further detail presently since there it is difficult to know a priori which columns to discard such that we maximise the novel information gained per column (thereby reducing "useless" calculation costs in the construction of the SVD). Clearly, there is an avenue for much further exploration between POD and SSA which we hope our discussion can support.

Having discussed how we extract trends from our time series, we now turn to the question for constructing some quantitative measure of *similarity* between them.

5 How Similar are Two Time Series?

The question of ascertaining when two time series are similar is one that appears deceptively simple at first glance. Intuitively, most would be able to tell which time series out of a given sample are similar, but there are a plethora of quantitative measures one may employ in this regard. In particular, the question arises of what sense of similarity is desired. Do we seek the overall shape to be the same? Are we concerned with lead-lag effects? Do we care about absolute or relative differences? These are all questions which must be discussed depending on what relationships we seek to establish between the time series.

Here we present several methods to quantify similarity, choose one, and justify our choice. The methods we discuss are split broadly into two groups: statistical measures (namely based on correlation) and distance based measures. Before we proceed, however, we first separate the notion of *distance* and *similarity*.

5.1 Distance, Similarity, and their Relationship

In all the papers we have looked at within time series analysis, authors relate a notion of high distance directly with low similarity. This intuitively makes sense if we consider two time series represented as two vectors, and the distance measure we use to be the Euclidean norm. If the vectors are the same, their distance measure is zero, corresponding to maximum similarity. Further, if the two vectors are mutually orthogonal, their norm would be relatively large - meaning a low similarity.

However, this notion of similarity does not necessarily have any mathematical backing, and it is not obvious why any classification of distance should necessarily provide the same classification based on a measure of similarity. Indeed, the authors of (Emms and Franco-Penya 2013) discuss an example where distance-based methods' results may be replicated by similarity-based methods', but *not* vice versa. Therefore, this question of effectively relating these two notions is addressed in papers such as (Elzinga and Studer 2019). The authors describe various conditions which a similarity measure of two objects $s(x,y)$ should hold,

$$s(x,y) \geq 0, \tag{17}$$

$$s(x,y) \leq \min\{s(x,x), s(y,y)\}, \tag{18}$$

$$s(x,y) = s(y,x), \tag{19}$$

$$s(x,y) + s(z,z) \geq s(x,z) + s(y,z). \tag{20}$$

Here, (17) represents the idea that similarity is strictly non-negative, (18) that the similarity of an object and a different object is less than the self-similarities of either of the objects, (18) represents that similarity is symmetric in its arguments, and the last represents an inequality akin to the triangle inequality for norms, adjusted for the notion that "closer" objects have a *higher* value for $s(x, y)$.

In particular, the authors go on to discuss various measures of similarity: for our purposes, we consider what we dub the "reference similarity measure" (RSM) defined for some distance metric $d(x, y)$ and reference object r as

$$s_{RSM}(x, y) = d(x, r) + d(y, r) - d(x, y);$$

for our analysis we simply set r to be a time series of all zero values for agnosticity to context. Furthermore, the authors of (Chen, ma, and Zhang 2009) discuss a map of the form (which we call the Gaussian similarity measure (GSM))

$$s_{GSM}(x, y) = \exp\{-d(x, y)\},$$

where again $d(x, y)$ is some distance metric. We name this similarity metric such due to the resemblance to the radial basis function $e^{-\|x-y\|^2}$ used in kernel methods for similarity measurement (as discussed in (Schoenberg 1938)).

An important point to note now, however, is that any ranking implemented by a distance metric will be replicated by s_{GSM}. To show this, consider the case where $d(x, a) \leq d(x, b) \leq d(x, c)$. Noting that the function $\exp\{-x\}$ is strictly decreasing, we thus immediately attain the ordering $s_{GSM}(x, a) \geq s_{GSM}(x, b) \geq s_{GSM}(x, c)$.

5.2 Distance Based Measures

Distance based measures, as their name implies, utilise some measure of difference of the values of two time series x, y in constructing their notion of similarity.

5.2.1 Euclidean Distance

Perhaps the simplest measure to describe is simply the Euclidean distance. For time series X_T, Y_T, first construct the vectors $\vec{x}, \vec{y} = (x_1, \ldots, x_T)^T, (y_1, \ldots, y_T)^T$, and then define the Euclidean distance d_E on them as

$$d_E(X_T, Y_T) = \|\vec{x} - \vec{y}\|_2,$$

with the subscript denoting the use of the 2-norm.

This method of distance metric has several advantages. Primarily, it is incredibly cheap: it is $O(n)$ for given vectors of length n, and moreover it is simple to implement, requiring trivial algebra.

The norm, however, does suffer from the fact that it is a "stiff" method, in that the comparisons are done element by element; this means that a smaller value for d_E arises when the series values are similar at corresponding time indices. This brings us to a feature we strongly desire in similarity measures, which is the ability to account for lead-lag effects (methods that allow for such "warping" are referred to as "elastic" methods). In order to demonstrate this with a concrete example, we first discuss two alternative similarity measures which aim to rectify this failing, and then provide an illustration of this in play.

5.2.2 Dynamic Time Warping

We thus move to discuss the first of our elastic measures, Dynamic Time Warping (DTW) (Keogh and Ratanamahatana 2005). Dynamic Time Warping has enjoyed much success in time series similarity analysis, and is often used as the de-facto standard against which to benchmark results as shown in (Pei, Tax, and Maaten 2016). The primary motivation for the introduction of DTW was to rectify the issues incurred via the stiffness of the Euclidean distance and other measures. We construct DTW as such.

For two time series X_T, Y_T, we first take some distance metric evaluating the difference of their values at time step t (such as $d(x_t, y_t) = |x_t - y_t|$). We then construct a $T \times T$ grid G, where we define

$$G_{i,j} = d(x_i, y_j) \; \forall (i,j) \in [(1,1), \ldots, (T,T)].$$

We then construct a *warping path*, defined as a set of elements $W := \{w_k\}_{k=1}^N = \{(i,j)_k\}_{k=1}^N$, that is to say a set of tuples of indices on the grid G. The elements of W are defined such that they map from one series to another; intuitively one may think of this as a set of "differences" to collect as we change one series' values to the other's. We then implement some constraints on these maps.

C.1 $w_1 = (1,1)$ and $w_N = (T,T)$, which serves to ensure we match the starts and ends of both series.

C.2 If $w_k = (i_0, j_0)$, then $w_{k-1} = (i_1, j_1)$, where $0 \leq i_0 - i_1 \leq 1$, and $0 \leq j_0 - j_1 \leq 1$. By forcing the tuples of indices to be non-decreasing, we ensure that we "warp" forward in time.

C.3 For a given element $w_k = (i_0, j_0)$, allow $|i_0 - j_0| \leq \omega$, for some window parameter ω. This parameter constrains the allowable points to match (so we do not match time points between series which are further than ω apart), which may serve to help pick out temporally local similarities over ones that are further in time.

With these constraints in place, note that we can also further constrain $T \leq N \leq 2(T-1)$. We may then define the DTW cost as the minimum of the sum of all elements w_i, over all maps W which fulfil the constraints C.1-C.3,

$$d_{DTW}(X_T, Y_T) = \min_W \left\{ \sum_{i=1}^{N} w_i \right\}. \tag{21}$$

However, we can practically implement the DTW cost via instead constructing a $(T+1) \times (T+1)$ grid of accumulated costs A (indexing from $0 \leq i, j \leq T$), populating it with $A_{i,j} = \infty \ \forall (i,j) \neq (0,0), A_{0,0} = 0$, and using the dynamic programming equation

$$A_{i,j} = \left\{ d(x_i, y_j) + \min \left\{ A_{i-1,j}, A_{i,j-1}, A_{i-1,j-1} \right\} \mid (i,j) \in W \right\}. \tag{22}$$

Then, we simply define $d_{DTW}(X_T, Y_T) = A_{T,T}$.

5.2.3 Time Warp Edit Distance

The last method we discuss is another elastic measure, the Time Warp Edit Distance (TWED). TWED, being a relatively recent method, has in certain contexts been shown to provide similarity classification better than that of DTW, which is why we have chosen to utilise this measure (Grabocka and Schmidt-Thieme 2018; X. Wang et al. 2010). Being another elastic measure, it can essentially be seen as a similar implementation of DTW in that it permits the "warping" of time to compute similarity between time series. The key areas in which TWED differs from DTW are namely that TWED does not impose a hard limit, but penalises warping in time depending on a parameter ν, and also that TWED can be conceptualised as acting to *edit* the time series to match one another, by either "deleting" observations or "matching" observations so the overall shape of the time series better resemble each other. This conceptualisation of an elastic difference measure follows directly from the implementation of Edit Distances in Real Sequences (Wagner and Fischer 1974).

To construct the TWED score, d_{TWED}, we first take some distance metric $d(x_i, y_j)$ as in DTW and initialise a $(T+1) \times (T+1)$ grid A similar to that as in DTW (again indexing from zero). Re-utilising the constraints C.1 and C.2 on our desired mappings W_1, we define the final cost by the element $A_{T,T}$, where A is populated via

$$A_{i,j} = \left\{ \min \left\{ A_{i-1,j-1} + \Gamma_{X,Y}, A_{i-1,j} + \Gamma_X, A_{i,j-1} + \Gamma_Y \right\} \mid i, j \in W_1 \right\}. \tag{23}$$

Here, we have

$$\Gamma_{X,Y} = d\left(x_i, y_j\right) + d\left(x_{i-1}, y_{j-1}\right) + 2\nu|i-j|, \tag{24}$$

$$\Gamma_X = d\left(x_i, x_{i-1}\right) + \nu + \lambda, \tag{25}$$

$$\Gamma_Y = d\left(y_j, y_{j-1}\right) + \nu + \lambda, \tag{26}$$

where ν is the *stiffness parameter* discussed above, and λ the *deletion penalty* penalising "deletions" in either time series, with a "deletion" being the act of removing an entry in one time series such that the overall time series better approximates the other. We can demonstrate a visual implementation of TWED via the populated grid A presented in Figure 10 for two small time series X_3, Y_3. Note in particular that our choices for the parameters ν and λ are not particularly important for this case: we simply wish to present an example of how one might construct a TWED grid and compute the desired score.

Y_3				
1	∞	18	17	16
1	∞	13	12	13
9	∞	4	13	14
	0	∞	∞	∞
		5	-3	1
			X_3	

Figure 10: Populated TWED grid for two example time series, with $\nu = \lambda = 0.5$. $A_{T,T} = 16$.

Having discussed our two elastic methods, we now provide an example demonstrating how elasticity is a desirable feature to have in our time series analysis.

Consider a signal $f(x)$, a translation of that signal $f(x-c)$, and a different signal entirely (with an additional layer of Gaussian noise added on top) $g(x)$. We *know* that the signals $f(x)$ and $f(x-c)$ are more similar than the signals $g(x)$ and $f(x)$ by construction: we may state that the initial signal lags the translated signal by c. However, we see from Figure 11 that using the Euclidean distance as our similarity measure, we note that we would observe $g(x)$ to be the more similar signal instead. However, using the two other elastic measures we have discussed we see that we correctly identify the translated signal as being more similar to the initial signal.

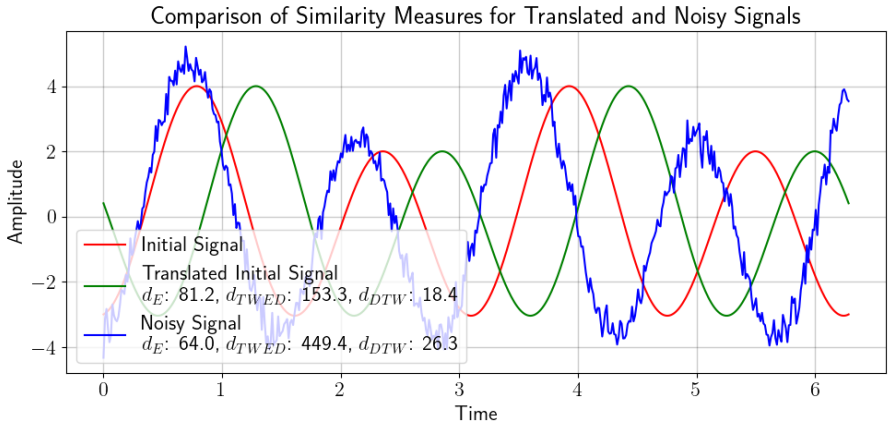


Figure 11: Comparison of similarity measures for an initial signal, the same signal translated, and a noisy signal.

Nevertheless, it must be noted that both TWED and DTW have a computational cost of $O(T^2)$, a relatively steep cost for our time series. Further, if we are to consider implementations in contexts in which we cross-examine M time series, we attain a computational cost of $O(M^2T^2)$ as opposed to the $O(M^2T)$ we would have for the simple Euclidean distance case: potentially devastating for contexts with low compute and high time sensitivity, and thus an important drawback to consider.

5.3 Correlation

Another popular method for examining similarity of shape in time series is via correlation. However, despite seeming initially desirable, note that time series which are correlated may still diverge in value (provided the movements in values remain similar); this is not the case, however, for time series which are *co-integrated*. Before we can define what we mean by co-integrated time series, we first must define integrated time series:

Definition 5 (Integrated time series)**.** *Define the operator* $(1-D)Z_t := Z_t - Z_{t-1}$*. A time series* Z *is then integrated to order* d *if* $(1-D)^d Z$ *is stationary.*

Using Definition 5, we may then define co-integration:

Definition 6 (Co-integration (Hamilton 1994))**.** *A set of time series are co-integrated if*

1. *The time series are all themselves integrated to order* d*.*
2. *There exists a linear combination of these time series which is integrated to order* $d_1 < d$*.*

Note in particular that co-integrated time series remain "close" in both value and trend, as opposed to correlation which only retains trend (Hamilton 1994). While this distinction is arbitrary, in the context of financial data we believe it does not make sense to consider just the shape when discussing similarity between time series as actual value differences may also play a part in financial strategies, such as via looking to group "large" tickers together.

In order to decide which of these methods[9] we utilise for our chosen similarity measure, we now must construct an avenue to differentiate them. This is a particularly sensitive task for our context as we seek a solution for a very specific set of data, and moreover we are working with publicly available stock series in a specified time-frame (and thus set of market conditions).

To attempt to decide upon a choice, we follow a procedure outlined as such:

1. We generate a list of 24 random tickers from various sectors (24 being $\sim 5\%$ of the number of tickers we work with).

2. For each of these twenty four tickers, we find three other tickers which are from the same GICS sub-sector; we label these three other tickers as being "similar" to our initial ticker. These choices are verified by visual inspection: if a given ticker Y is obviously very dissimilar to ticker X despite being in the same sub-sector, another ticker is chosen, potentially branching to the GICS *sector* if the sub-sector is too small.

3. Then, for each of our twenty four chosen ticker labels, we calculate the top five most similar (five being approximately 1% of the total number of tickers within our data set) other ticker labels for each of the nine different methods we have to analyse.

   Note that our choice of the value five is both somewhat arbitrary yet not particularly important: we expect our similarity measures to not perfectly replicate our results given that there is no inherent guarantee that tickers within similar sub-sectors ought to be the closest in similarity to each other, and so we allow for some "flex" with choosing to accept any appearance within the top five.

4. We then compute, for each of the twenty four chosen tickers, how many times each similarity method's top five most similar tickers has a ticker that matches one of

[9] Recall that for any given distance metric d, we also have two associated similarity metrics s_{GSM} and s_{RSM} - leading to nine different choices for the similarity measurement.

the three benchmark similar tickers we have chosen. To illustrate, consider Table 1, in which we illustrate this methodology for a chosen synthetic ticker. Suppose we find that TIC_1, TIC_2, and TIC_3 are the three chosen reference tickers. Then, after constructing the top five most similar tickers from each similarity method, we can see from Table 1 that (of the methods shown) the Euclidean distance gives the best results - matching all three of the desired tickers.

Similar Tickers	Top Five Tickers by Similarity Method				
	d_E	d_{DTW}	d_{TWED}	$s_{RSM}(d_E)$	...
TIC_1	TIC_1	TIC_2	TIC_7	TIC_{19}	...
TIC_2	TIC_{23}	TIC_{11}	TIC_{99}	TIC_{330}	...
TIC_3	TIC_2	TIC_8	TIC_3	TIC_5	...
	TIC_{366}	TIC_{223}	TIC_{23}	TIC_{20}	...
	TIC_3	TIC_3	TIC_7	TIC_8	...

Table 1: Example similarity examination for a synthetic ticker. Of the shown similarity methods, d_E attains the best results.

To generate our final choice of similarity measure, we simply choose the method which attains the most "matches" over all twenty four tickers, so (assuming the ticker utilised in Table 1 is the first) the score for the Euclidean distance would be calculated by $3 + \ldots$ and the score for DTW would be calculated by $2 + \ldots$.

Using this methodology, we can observe our results in Table 2 which show that the TWED score and the TWED-based Gaussian similarity measure provide the best results, and so due to simplicity we henceforth work solely with the TWED score.

	Score Attained by Similarity Method		
Distance Measure	$d(x,y)$	$s_{GSM}(d(x,y))$	$s_{RSM}(d(x,y))$
d_E	30	30	30
d_{DTW}	21	21	21
d_{TWED}	36	36	18

Table 2: Table of results for all similarity measure comparisons.

We next turn towards choosing a method to establish when two time series start to become dissimilar. This problem is of interest namely because we know that "similarity" of two real-world time series is not a constant, and therefore detecting when two

formerly-similar series have begun to diverge (such that we can stop tracking one, or execute a trading strategy relying on the divergence of their prices) presents an important challenge.

6 Dissimilarity Finding

We present two approaches towards answering the question of when two time series have started to diverge, both of which necessitate the usage of a "window" within which to analyse data to check if the time series being examined are converging or diverging, in the colloquial sense. Also note that we begin our analysis with synthetic data, but we present some results on our tickers in Section 8.

A point to note is that we wish to seek some "online" avenue of detecting dissimilarities which occur. Here, "online" methods refer to methods which take in new data as it arrives and output their results ideally immediately (but in practice with some delay governed by the window chosen). The alternatives to these approaches are known as "offline" methods, which look at entire time series of data and then highlight areas in which there are dissimilarities between the time series. Given the context of seeking to apply these methods to financial data, it does not make sense for us to seek offline methods - since one can fairly easily imagine a scenario in which the time lag induced by offline methods leads to disastrous financial ramifications.

To construct our synthetic data, we utilise a one-dimensional random walk of length T wherein we define our synthetic path via first constructing a set of probabilities $\{\alpha_i\}_{i=1}^{T}$, where $\alpha_i \sim U(0,1)$ for $i = 1, \dots T$. Next, generate the series of movements M_T, where

$$(M_T)_i = \begin{cases} 1 \text{ if } \alpha_i \leq 0.5, \\ -1 \text{ if } \alpha_i > 0.5, \end{cases}$$

We then define the walk W_T via

$$(W_T)_i = \sum_{j=1}^{i} (M_T)_j \,,$$

for $i = 1, \dots T$. To then generate a time series which is synthetically similar, we first generate a "noised" version of our walk N_T by adding on Gaussian noise with mean zero and variance one-hundredth of that of our walk. We then choose a subsection of $(N_T)_i, I_1 \leq i \leq I_2$ where $I_1 > 1, I_2 < T$, and re-run a random walk in that subsection, thereby generating a walk $\tilde{W}_T$ which is broadly similar but has a section in which the overall trends diverge.

Our first method of analysing for introduced dissimilarities may be thought of as using a moving TWED score, wherein we take a rolling window of our time series and compute the cross-TWED scores within this window. When the scores reach some threshold, we can state that the series have begun to show dissimilarity or similarity, and so we may act as appropriate. However, this presents us with two immediate issues.

Firstly, there is the question of what choice of window length to use as the rolling window. Though there has been work done on examining optimal window sizes for such algorithms (Imani et al. 2021), difficulties arise when utilising automatic detection methods on our data set due to the inherent noisiness of the data. Indeed, implementing the algorithm proposed in (Imani et al. 2021) results in a suggested window size of one, clearly a nonsensical result. Thus, we turn instead to a more qualitative approach.

Financially, there are several timescales that make sense for us to examine our data from. Publicly traded companies release statements quarterly so this presents one possible time-gap, but we may desire finer resolution in picking up changes so we may also seek to look for a window that is shorter (such as a month). Further, there is the issue that the actual window length will be contextually dependent on the use case of such analysis: for financial time series there might be certain windows which do not make sense (for example in looking for decade-long trends), but these might play a part in other contexts.

The next issue pertains with the actual values of the TWED scores as we slide our window. Since the value of the TWED score is dependent completely upon the two time series that we analyse, deciding when a threshold has been met such that the time series are dissimilar is something that cannot be decided a priori. We propose a method in which at first the moving TWED score is simply ran for an amount of time to generate a baseline, and then the following data is then normalised by the deviation measured. This ensures that we are able to set roughly an equivalent threshold for each examination of the moving TWED score per pair of time series, under the assumption that underlying similarities do not change *too much*. We can explore this within Figure 12.

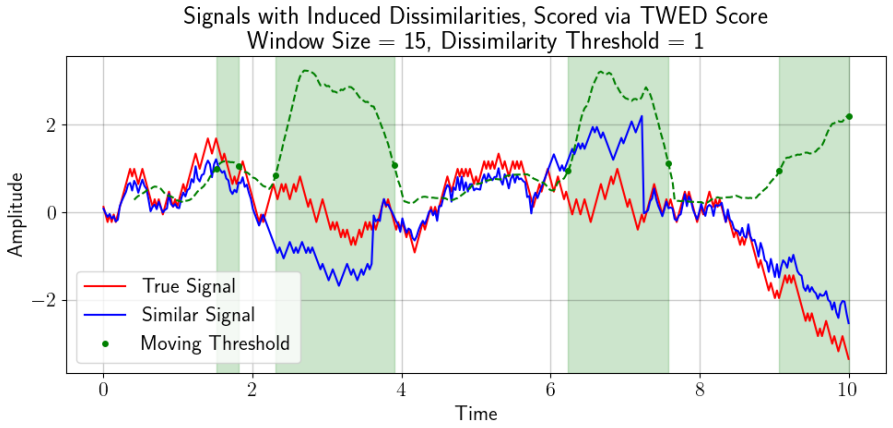


Figure 12: Example of the moving TWED score method of detecting dissimilarities.

Clearly, this method provides reasonable results (in this context using the time series until $t = 5$ to set the variance), but it is simple to see that if the trends of the time series diverge, we would need to at some point "re-calibrate" the variance normalisation. To look for alternative approaches, we examine work done in change point detection within time series. The authors of (Aminikhanghahi and Cook 2017) present a meta-analysis of various change point detection algorithms, and discuss how Relative Unconstrained Least Squares Importance Fitting (RuLSIF) (Liu et al. 2013) appears to regularly outperform other methods in detecting changes inherent to time series. We first introduce some definitions before we define RuLSIF in detail, expanding upon the analysis in (Liu et al. 2013).

To start, consider a sample from a time series $X(t) := (x(t), x(t+1), \ldots x(t+k-1))^T$ of length k. Then define by $\mathscr{X}(t) := (X(t), \ldots X(t+k-1))$ the *Hankel* sample matrix corresponding to k of these time series samples of the function x. We aim to try and model the probability distributions of the entries in the matrices $\mathscr{X}(t)$ and $\mathscr{X}(t+k)$, to try and determine if they are sufficiently differentiated to claim dissimilarity has occurred.

Next we define the *Pearson Divergence*, allowing us to measure the divergence of two probability distributions. This is helpful to us given that a "dissimilarity" is induced when the incoming time series value are not likely to be from the same distributions as the previous ones.

Definition 7 (Pearson Divergence (Liu et al. 2013))**.** *For two probability distributions P and Q (with associated probability density functions $p(X)$ and $q(X)$ respectively), define the Pearson Divergence (PE) as*

$$PE(P \,\|\, Q) := \frac{1}{2} \int q(X) \left(\frac{p(X)}{q(X)} - 1 \right)^2 dX. \tag{27}$$

The most immediate issue is that we do not know either $p(X)$ or $q(X)$, and so standard methods have (such as via KLIEP, another algorithm used in change-point detection) estimated the *density ratio*,

$$r(X) := \frac{p(X)}{q(X)}.$$

We next define the *α-Pearson Divergence*.

Definition 8 (α-Pearson Divergence (Liu et al. 2013))**.** *For two probability distributions P and Q (with associated probability density functions $p(X)$ and $q(X)$ respectively), define*

the α-Pearson Divergence (PE) as

$$PE_\alpha(P \parallel Q) := PE(P \parallel \alpha P + (1-\alpha)Q) \tag{28}$$

$$= \frac{1}{2}\int q_\alpha(X)\left(\frac{p(X)}{q_\alpha(X)} - 1\right)^2 dX, \tag{29}$$

where $q_\alpha(X) = \alpha p(X) + (1-\alpha)q(X)$ is the α-mixture density.

We then define the α-relative density ratio,

$$r_\alpha(X) := \frac{p(X)}{q_\alpha(X)},$$

as the variable we wish to estimate (rather than the individual probability density ratios themselves). Here, the parameter α represents some variable accounting for the mixing of the probability densities of P and Q: the primary reason for its inclusion is such that we may bound the ratio $r_\alpha(X)$ whereas the value $r_0(X)$ recovers the formulation as in uLSIF[10], shown to have non-ideal convergence properties when $q(X)$ is small (Yamada et al. 2013).

Next, to determine the ratio $r_\alpha(X)$, we model the α-relative density ratio via the kernel expansion

$$m(X;\theta) = \sum_{i=1}^{n} \theta_i K(X, X_i), \tag{30}$$

where X_i is the ith sample of $p(X)$, $\theta = (\theta_1, \dots \theta_n)^T \in \mathbb{R}^{n\times 1}$ are weights, and $K(X,Y)$ is the Gaussian radial basis function[11]. To generate our model m, we minimise (with respect to θ) the functional

$$J(X) := \frac{1}{2}\int q_\alpha(X)\left(r_\alpha(X) - m(X;\theta)\right)^2 dX, \tag{31}$$

$$= \frac{1}{2}\cancel{\int q_\alpha(X) r_\alpha^2(X)\, dX} \tag{32}$$

$$- \int p(X) r_\alpha(X) m(X;\theta)\, dX + \frac{\alpha}{2}\int p(X) m^2(X;\theta)\, dX + \frac{1-\alpha}{2}\int q(X) m^2(X;\theta)\, dX, \tag{33}$$

where the cancelled term is ignored as it is a constant. We can thus re-express (33) as solving the problem

$$\min_\theta \left\{ \frac{1}{2}\theta^T \hat{H}\theta - \hat{h}^T\theta + \frac{\sigma}{2}\theta^T\theta \right\}, \tag{34}$$

[10] The method upon which RuLSIF is based.

[11] Correspondingly, $\tilde{X}_i$ is the ith sample of $q(X)$.

where the parameter σ is a smoothing term associated with regularisation, $\hat{h}_j := n^{-1}\sum_{i=1}^{n} K(X_i, X_j)$, and

$$\hat{H}_{i,j} := \frac{\alpha}{n}\sum_{p=1}^{n} K(X_p, X_i)K(X_p, X_j) + \frac{1-\alpha}{n}\sum_{p=1}^{n} K(\tilde{X}_p, X_i)K(\tilde{X}_p, X_j). \tag{35}$$

Thus, the solution to the minimisation problem (34) is (after differentiating with respect to θ)

$$\theta := \left(\hat{H} + \sigma I_n\right)^{-1}\bar{h}, \tag{36}$$

where I_n is the identity; this value of θ is inserted into (30) to provide our model for the α-relative density ratio. Next, consider expression (29). We expand by writing

$$PE_\alpha(P \,\|\, Q) = \frac{1}{2}\int q_\alpha(X)\left(\frac{p(X)}{q_\alpha(X)}\right)^2 dX - \int \cancel{q_\alpha(X)}\frac{p(X)}{\cancel{q_\alpha(X)}}\, dX + \frac{1}{2}\int q_\alpha(X)\, dX, \tag{37}$$

$$= \frac{1}{2}\int q_\alpha(X)\left(\frac{p(X)}{q_\alpha(X)}\right)^2 dX - \frac{1}{2}, \tag{38}$$

$$= -\frac{1}{2}\int q_\alpha(X)\left(\frac{p(X)}{q_\alpha(X)}\right)^2 dX + \int p(X)\left(\frac{p(X)}{q_\alpha(X)}\right) dX - \frac{1}{2}. \tag{39}$$

Next, substituting $q_\alpha(X) = \alpha p(X) + (1-\alpha)q(X)$ into the coloured term within the first integral in (39), discretising, and substituting in our model for $r_\alpha(X)$, we arrive at a calculable expression for the α-Pearson Divergence as

$$\hat{PE}_\alpha(P \,\|\, Q) = -\frac{\alpha}{2n}\sum_{i=1}^{n} m^2\left(X_i;\theta\right) - \frac{1-\alpha}{2n}\sum_{i=1}^{n} m^2\left(\tilde{X}_i;\theta\right) + \frac{1}{n}\sum_{i=1}^{n} m(X_i;\theta) - \frac{1}{2}. \tag{40}$$

To implement a measure for calculating the change points, we utilise the symmetric expression[12]

$$PE_\alpha := PE_\alpha(P_t \,\|\, P_{t+n}) + PE_\alpha(P_{t+n} \,\|\, P_t), \tag{41}$$

where P_t denotes the probability distribution associated with $\mathcal{X}(t)$, approximating (41) using the expressions in (40). We attain our values for θ via a neural network using the Adam optimiser minimising the loss function (34). We use this neural-network based approach as opposed to the kernel-based approach discussed primarily due to past studies (as in (Hushchyn and Ustyuzhanin 2021)) demonstrating better overall performance; note that this is not a novel result and is a known limitation of kernel-based methodologies (Ghorbani et al. 2021).

[12]Note that the expression (29) is not symmetric in P and Q.

We demonstrate RuLSIF via an implementation on our data, plotting the similarity alongside the moving TWED score method. We present our results in Figure 13, where we utilise RuLSIF on the moving TWED score (plotting our dissimilarity (41)); we find the relative peaks of the dissimilarity by looking for maxima within windows of values (defined as $w = 30$), with $n = 2w$ and $\alpha = 0.2$.

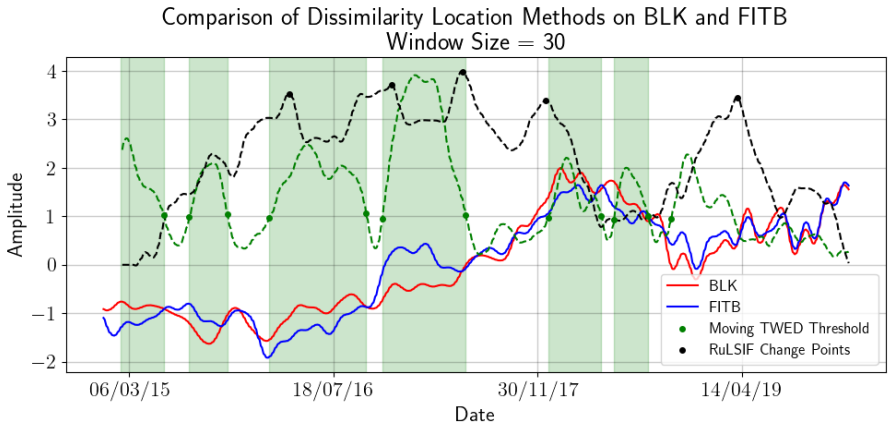


Figure 13: Comparison of moving TWED score with RuLSIF on BLK and FITB tickers, with Z-normalised prices.

One clear disadvantage with RuLSIF is that we do not detect when we enter or exit a certain regime, but merely when we change regimes. This means that there is a greater induced complexity in seeking dissimilarities as we must keep track of the changes within regimes, as these may simply be going from "very similar" to "similar" (in a qualitative sense). Further, we do not get rid of the issue of needing to decide upon a window size for our analysis (and the effects of poor window size can be seen in the spurious "similarities" detected), and also that utilising this method is necessarily more computationally complex than the moving TWED score.

However, we do get rid of the issue of needing to decide upon thresholds for when dissimilarity occurs as these should be picked up inherently within the algorithm. We believe this benefit is great enough that we suggest to use RuLSIF for our dissimilarity detection, as deciding when and how the TWED score ought to be re-normalised is not a trivial task, and is certainly one that is more difficult than simply checking to see what "regime" one is in when the RuLSIF algorithm identifies a change point.

Having decided upon a method for noticing changes in similarity for our time series, we now discuss some specifics in the actual similarity analysis for our data set. We also provide more insight into the incorporation of sector-level data within our analysis.

7 Similarity Scoring

For the bulk of our data generation, we utilise `series_scorer` (Baig 2023), a Python package developed by the author for the smoothing and cross-scoring of multiple time series[13]. It is worth noting at this point that the data generation was not a trivial task: using parallelised code, generation of a cross-score[14] matrix of size 477×477 using TWED takes approximately 12 hours given the length of the time series we are working with.

Once we have extracted trends via SSA and then cross-scored, we then turn towards incorporating sector-level information regarding our data set, which we hope provides further useful insight. To proceed, let us first introduce some terminology which will be useful when discussing cross-sector similarities.

We first discuss *intra-sector* scoring. To generate scores of this type, consider a given sector s_1. We generate all the ticker labels which correspond to this sector, and then compute the TWED score between every ticker in this sector and every ticker corresponding to a sector s_2 (with s_2 potentially the same sector as s_1). Lastly, we take the means of these scores - which represents the intra-sector TWED score between sectors s_1 and s_2.

Next, we discuss *inter-sector* scoring. Here, we calculate the mean normalised price movements of all the tickers within a sector s_1, and compute the TWED score between this averaged time series with the averaged time series of another sector s_2. Note how that while s_2 may be the same sector as s_1, this would lead to a cross-score of zero. Nevertheless, this score we calculate is thus the inter-sector TWED score.

To then decide which of these methods to use, we utilise a similar approach as in our discussion of choice of similarity measure. We generate 24 random tickers, and get the sectors that they belong to. Then, using both methods, we compute the closest sector to a given ticker: we choose the method that better demonstrates that the closest sectors are the ones that these randomly chosen tickers belong to.

Using this analysis, we find that the intra-sector method outperforms the inter-sector method (the former predicting 11 sectors correctly versus the latter's 7 correct predictions), and so we make the decision to use this in further discussion; we now turn towards analysing our results and discussing the insights we have found. Note that the intra-sector performing somewhat better may be due to the fact that intra-sector

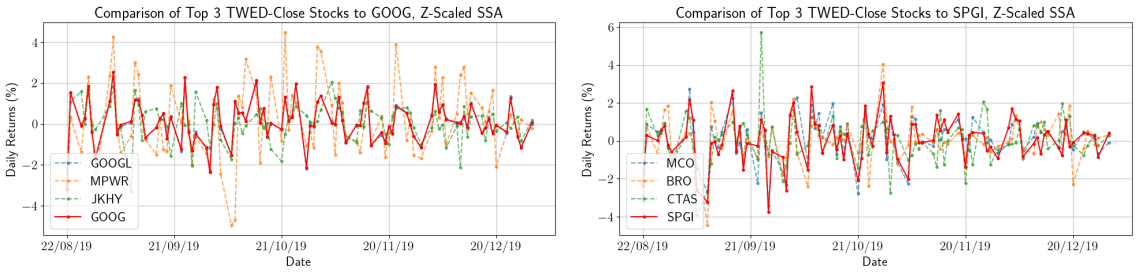
[13] Note that this software has already been submitted for credit as part of the Python in Scientific Computing special topic as part of the MSc MMSC 2022-2023.

[14] Defining "cross-scoring" as computing the TWED score between any given time series.

scoring is an indication of the closeness, on average, to the tickers *within* a given sector, whereas inter-sector scoring is the closeness to the *average* movements of the sector. Intuitively, the former may provide better results since there is better incorporation of ticker-ticker effects between specific companies, rather than them being "washed out" in the comparison between just two time series.

8 Results

We can begin our discussion of results by examining Figure 14, where we present the top three closest tickers to GOOG in Figure 14(a) and the top three most similar tickers to SPGI in Figure 14(b).



(a) Top three most similar tickers to GOOG.

(b) Top three most similar tickers to SPGI.

Figure 14: Comparison of top three most similar tickers for GOOG and SPGI, presented as returns over 90 days.

Considering first Figure 14(a), we see that we pick up on GOOGL (the Class A Alphabet stock) when we desire similarity with the Class C ticker GOOG, a good sign that we are picking up on obvious similarities. The other two tickers we see are Monolithic Power Systems, an industrial computing hardware provider, and Jack Henry & Associates, a FinTech company. Importantly, while these might be relatively surprising choices, that these are still clearly very visually similar, wherein we see their returns following very similar patterns of the three months plotted.

Next, we turn to considering Figure 14(b). Here, the most similar tickers to S&P Global are comprised of Moody's Corporation (also providing financial information), Brown & Brown (an insurance firm), and Cintas, which provides products such as apparel and services to other businesses. Within this selection of results we once again have perhaps a surprising outcome (CTAS) but also one that we clearly expect (MCO): this is a desirable end-result in so much as our analysis appears to be providing potentially non-intuitive insight into ticker similarities, beyond "standard" approaches. Were it the case that our analysis simply provided similar tickers to one another that we may otherwise expect (such as Pepsi and Coca-Cola), one could then indeed ask why we would bother to use this complicated analysis in production environments: we consider the generation of these novel pairs a success in terms of providing us avenues to explore further whether indeed that the tickers generated are "similar" in a meaningful sense (that is to say, is it indeed possible to extract financial utility from our analysis).

We may also turn towards examining similarity across sectors by examining the means of the cross-sector scores which we present in Figure 15.

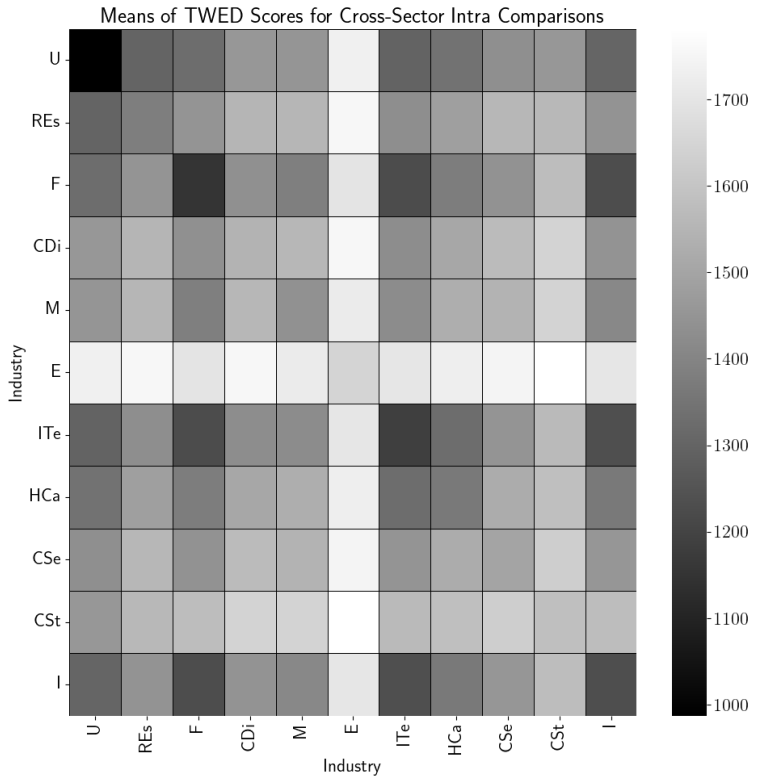


Figure 15: Cross-sector mean TWED scores using the intra-sector method.

Several conclusions may immediately be made from this figure. Firstly, Utilities companies tend to be relatively self-similar amongst each other, far more than the other sectors. Equally, Energy companies are quite *dissimilar* to all other sectors (strikingly so compared to Consumer Staples companies), and also amongst themselves. We can also state that there are three sectors (Industrials, Financials, and Information Technology) that display relatively high cross-similarity amongst each other: in particular the relationship between Financial and IT companies is fairly close. This is somewhat as we expect given the composition of the S&P 500 at the time (and indeed now): the largest market capitalisations were almost all without fail either technology companies or financial services companies, and so market movements would be unreasonably influenced by their own movements (thereby linking the two sectors in this analysis). Further, we can say that in general, it is not the case that companies within their own sector are significantly more self-similar than they are to companies in other sectors.

We can proceed by attempting to get an idea of how strong our conclusions are, by now viewing the deviations on this data which we present within Figure 16.

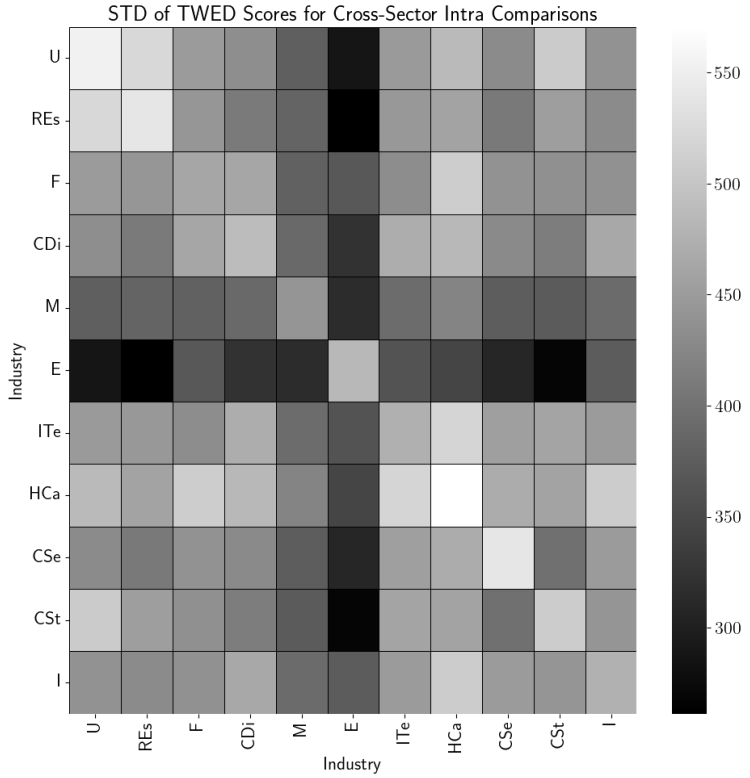


Figure 16: Cross-sector deviations on TWED scores using the intra-sector method.

We see that our conclusions drawn earlier for Utilities are relatively weak, but that for Energy companies they are relatively strong (i.e. that indeed we can say that Energy companies tend to be more dissimilar to other sectors than the norm). Further, we can see from Figure 16 that there are several sectors which have relatively high variances within themselves as in Health Care, and that there exist relationships like that between Financials and Health Care in which we see relatively high variances as well. We can also say that in general, the variances within the sectors are higher than they usually are across sectors, meaning our conclusions regarding companies not being inherently more self-similar within sectors are relatively weak.

Note that at this point we may also back-justify our methodology for choosing our choice of similarity measure by examining the cross-scores for the GICS *sub-sectors*, by considering the information which we present in Figure 17. Here, we see that generally, tickers are more similar to one another within their own sub-sectors than their sectors,

meaning that it is not a bad assumption to believe that tickers within the same sub-sector should appear within their top-n most similar lists.

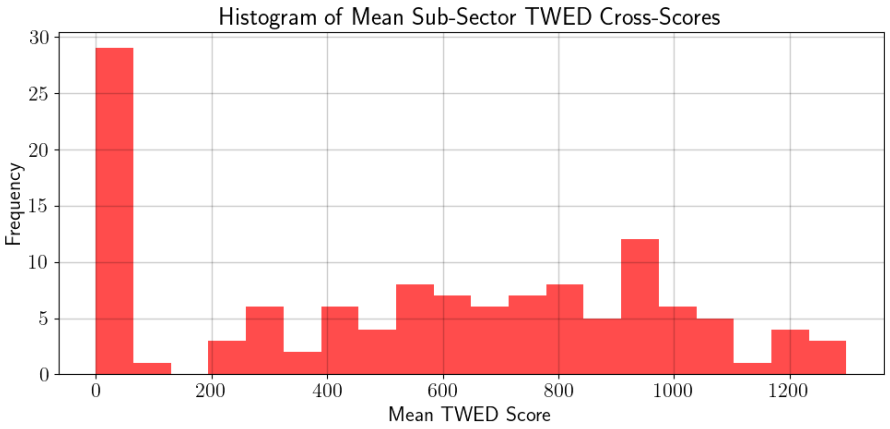


Figure 17: Cross-sub-sector mean TWED scores.

We may also analyse what *sectors* a given ticker is most similar to as in Figure 18, wherein we consider similar sectors to BLK.

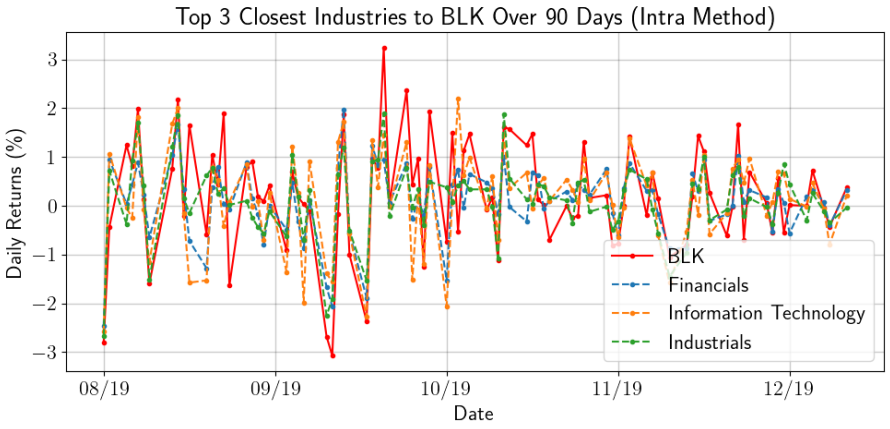


Figure 18: Top 3 similar sectors to BLK using the intra-scoring method.

These results are reassuring given that BLK (being a financial company) is shown to be most similar to other Financials and IT companies most, followed by Industrials; this then presents an opportunity to delve deeper to try and pre-emptively hedge against any shocks that may be occurring in any of these sectors. Such analysis may be helpful in seeking to provide insight into (for example) how to better predict the performance of a ticker if we know there exists some lead-lag effects with that ticker and a specific sector.

8.1 Low-Rank SSA, Hankel Matrices, & Chebyshev Expansions

We turn to discuss another avenue of exploration which arose throughout our wider work, delving deeper into the theoretical background of the analysis we have performed[15]. We start by posing two questions:

1. When is a given signal $s(t)$ such that a rank-r SSA truncation completely recovers the signal?

2. Can we say anything about the convergence of SSA to the optimal signal in terms of the rank of reconstruction?

To the first, the answer has been discussed in (V. N. Zhigljavsky and Anatoly A. 2001) that time series comprised of linear combinations of polynomials, sinusoidal, and exponentials permit Hankelised structures which are of finite rank. We discuss these three types of functions in turn and provide an alternative proof for series comprised of these functions in the case of polynomials.

Theorem 2 (Rank of time series comprised of a finite sum of polynomials)**.** *For a time series comprised of a polynomial $P(t)$ of degree p_0, consider the trajectory matrix formed by this polynomial, with window length $L = (T+1)/2$. The rank of this trajectory matrix is $p_0 + 1$.*

We construct the proof by induction.

Proof. The base case, with $p_0 = 1$, is trivial. For the next step in the inductive proof, assume that we need the vectors $(1, 1, \ldots, 1)^T, (0, 1, \ldots L-1)^T, \ldots (0, 1^k, \ldots, (L-1)^k)^T$ to represent the trajectory matrix for a polynomial of degree k, which we denote by $P_k(t)$. Now consider the trajectory matrix for a polynomial of degree $k+1$. If we consider this polynomial

$$P_{k+1}(t) = \alpha_0 + \alpha_1 t^1 + \ldots \alpha_{k+1} t^{k+1}, \tag{42}$$

we note that we can represent all but the last term in (42) via the basis vectors associated with $P_k(t)$, and we only lack a basis vector which can represent a term of the form t^{k+1}. It is then simple to see that the vector which does this is precisely $(0, 1^{k+1}, \ldots, (L-1)^{k+1})^T$; we have thus shown that we need $k+2$ basis vectors to represent the trajectory matrix associated with $P_{k+1}(t)$, completing the proof. □

Similarly, we may discuss series comprised of different sinusoids.

[15] We thank Professor Nakatsukasa significantly for his contributions towards this section, in particular for posing the guiding questions and for support suggesting ideas for experimentation.

Theorem 3 (Rank of time series comprised of a finite sum of sinusoids)**.** *Consider a time series of the form*

$$f(t) = \sum_{k=1}^{n} \alpha_k \sin(\omega_k t) + \beta_k \cos(\omega_k t).$$

When embedded into a trajectory matrix with $L = (T+1)/2$, the rank of this matrix is $2n$.

Proof. The proof is completed by simply noting that each frequency ω_k is associated with two basis vectors $\left\{B_1^k, B_2^k\right\}$, with the indices of these vectors defined such that

$$\left.\begin{aligned}(B_1^k)_i &= \alpha_k \sin(\omega_k \cdot t_i)\\ (B_2^k)_i &= \beta_k \cos(\omega_k \cdot t_i)\end{aligned}\right\} \text{ for } i = 1, \ldots, L.$$

□

Lastly, we discuss exponentials.

Theorem 4 (Rank of time series comprised of a finite sum of exponentials)**.** *Consider a time series of the form*

$$f(t) = \sum_{k=1}^{n} \alpha_k \exp\{\beta_k t\}.$$

When embedded into a trajectory matrix with $L = (T+1)/2$, the rank of this matrix is n.

The proof follows similarly.

Proof. Noting that $\exp\{\beta_k t + \Delta t\} = \exp\{\beta_k t\} \cdot \exp\{\Delta t\} = \delta \exp\{\beta_k t\}$, we immediately see that the columns of the trajectory matrix spanned by a single "mode" $\beta_k t$ are simply linear combinations of the first, since the increment in time along the columns is constant.

Next, note that the basis vectors $\left\{B^k\right\}$ defined by

$$(B^k)_i = \alpha_k \exp\{\beta_k \cdot t_i\},$$

where $i = 1, \ldots, L$, are linearly independent precisely due to the opposite argument, namely that there exists no constant $\tilde{\delta}$ relating any two columns. □

Discussing now the second question, wherein we seek insight into the convergence of SSA in terms of the rank of reconstruction chosen (provided we have a noisy low-rank signal), we first turn towards discussing the effect of the noise itself.

We first note that the "noise" component added to this matrix nearly ensures that the corresponding trajectory matrix is full-rank. To show this, consider a low-rank Hankel

matrix H. As the rank of the matrix is deficient, we have $R(H_r) = R(H_{r+1})$, where H_r is the minimal full-rank matrix, and H_{r+1} is the same matrix with another column of L observations. We thus have the relationship (for all $\alpha_i \neq 0$ and h_i the columns of H_{r+1})

$$\alpha_1 h_1 + \alpha_2 h_2 + \ldots + \alpha_r h_r + \alpha_{r+1} h_{r+1} = 0. \tag{43}$$

Therefore, by the structure of the Hankel matrix we must have

$$\alpha_1 s_i + \alpha_2 s_{i+1} + \ldots + \alpha_{r+1} s_{i+r} = 0, \tag{44}$$

for all $i \in [1, N-r]$, where s_i are the measurement of the time series. In general, we expect that a noisy matrix will *not* satisfy (44), meaning that it cannot be low-rank.

We can thus experimentally explore the effect that noise has on the optimal rank by taking twenty one different low-rank functions, adding Gaussian noise of varying levels (centred with $\mu = 0$ and $\sigma \in [1 \times 10^{-3}, 1 \times 10^{1}]$), and then proceeding to embed these functions within trajectory matrices. After recording the rank of the trajectory matrix that would have been formed by the noise-free function, r_0, we then perform SSA truncations to all ranks $r \leq \min\{L, K\}$ on our noisy trajectory matrices. For each of these truncations, we record the Euclidean distance between the reconstruction and the noise-free series[16], and then plot this data in Figure 19.

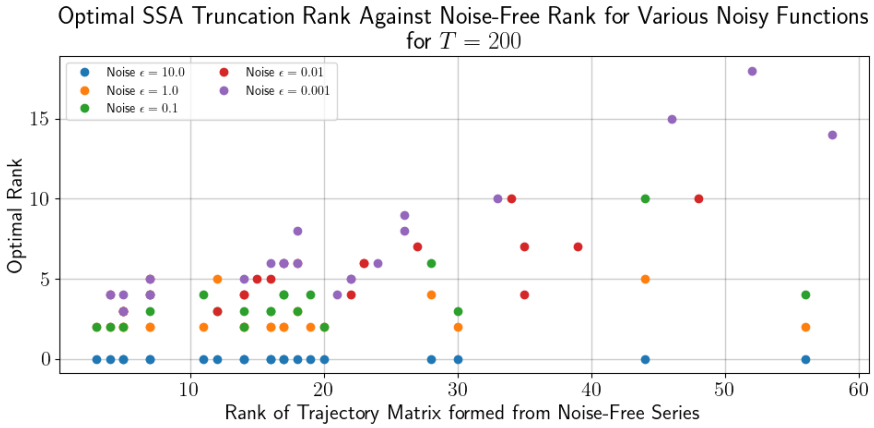


Figure 19: Examination of optimal SSA truncation rank against initial rank for various noisy functions with varying noise levels.

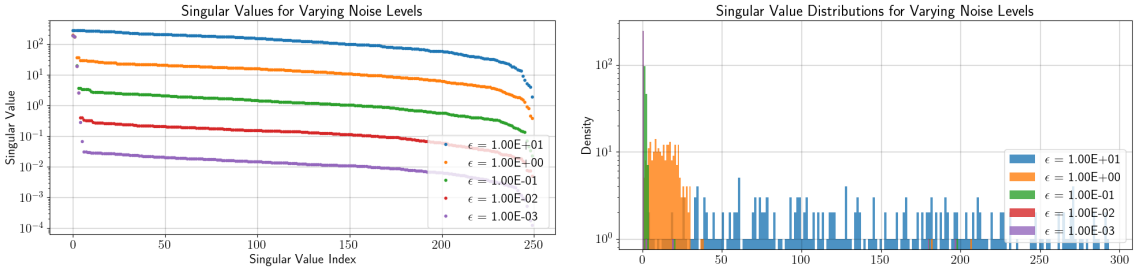
Here, we can see that as we increase the level of noise, we suppress the optimal truncation rank; as we decrease the level of noise we get closer to the point where

[16] We use the Euclidean distance since we assess a point-by-point reconstruction.

the optimal truncation rank is equivalent to the initial noise-free rank. We can explain this behaviour by the fact that when noise is relatively low, truncations closer to the rank of the underlying noise-free signal perform better (as we would intuitively expect). However, when noise is high, this may potentially cause over-fitting as the underlying signal becomes more difficult to discern. Thus, truncations at even lower ranks would perform better as they apply more conservative "estimates" of what the underlying signal would be, by considering only the most dominant of the singular values as corresponding to the signal.

We can examine this in Figure 20, where we plot the singular value distributions of the trajectory matrices formed by adding varying levels of Gaussian noise to a low-rank function. Figure 20(a) presents the overall singular values of these noisy matrices, where we can see that as noise decreases we suppress the latter singular values (i.e. we have a much clearer demarcation between noise and signal visually in our singular value plot), and as noise increases we see that it is more difficult to differentiate between which singular values correspond to noise and which to signal.

Figure 20(b) presents this information in the form of histograms pertaining to the singular values associated with the noise levels, we can clearly see that as noise decreases the bulk of the singular values become "small" and we retain a few "significant" singular values, as opposed to the case with large noise in which the singular value distribution is much more spread out.



(a) Singular values for varying noise levels on a low-rank function.

(b) Singular value distributions with varying noise for a low-rank function.

Figure 20: Examination of the effect of noise levels on singular values for low-rank functions.

Lastly, we present some work which we propose provides novel insight into the relationships between Chebyshev polynomials and Hankel matrices. Consider a function $f(x), x \in [-1, 1]$ such that the trajectory matrix formed from this function is of low

rank. We also know that such a function permits a Chebyshev expansion of the form

$$f(x) = \sum_{i=1}^{n} c_i T_i(X),$$

where $T_i(x)$ represents the ith Chebyshev polynomial. We consider the functions

$$\sigma = \exp\{m_\sigma I_\sigma + \gamma_\sigma\},$$
$$C = \exp\{m_C I_C + \gamma_C\},$$

where σ are the singular values associated with the corresponding trajectory matrix and C represents the Chebyshev coefficients c_i. Looking at a log-scale plot of a random function in Figure 21, we note that both C and σ appear to decay exponentially, but at different rates. We thus pose the question of whether there exists some underlying relationship between the decays of the singular values and the Chebyshev coefficients.

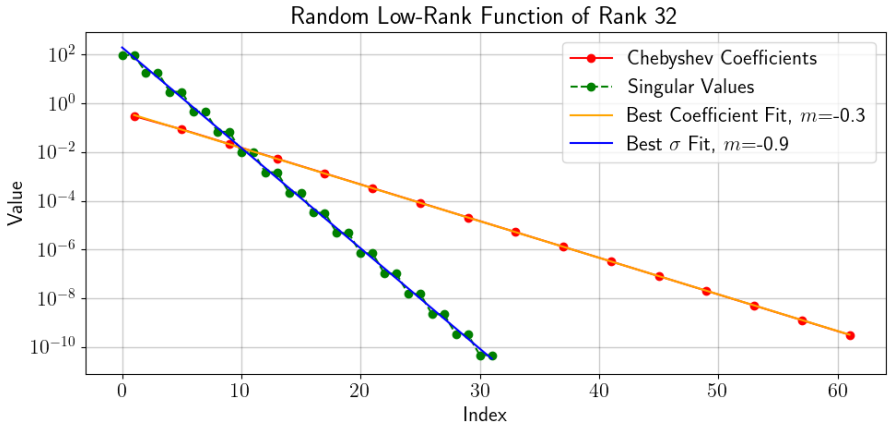


Figure 21: Decay of singular values and Chebyshev coefficients for a low-rank function.

We analyse this by generating the singular values and Chebyshev coefficients for various low-rank functions. For each function, we take a linear approximation to the decay of both these variables, and retain the gradients m_σ and m_C. We present the plot of these variables and the linear approximation of their relationship in Figure 22.

Note that we have also plotted the rank of the underlying trajectory matrix as well; the relationship which we observe is as we expect (i.e. that as rank increases we see slower decay of both the Chebyshev coefficients and the singular values) since higher rank trajectory matrices have "more" singular values through which to decay. There also appears to be a fairly strong relationship between the gradient of decay of the singular values and the gradient of the decay of the Chebyshev coefficients: in particular, the

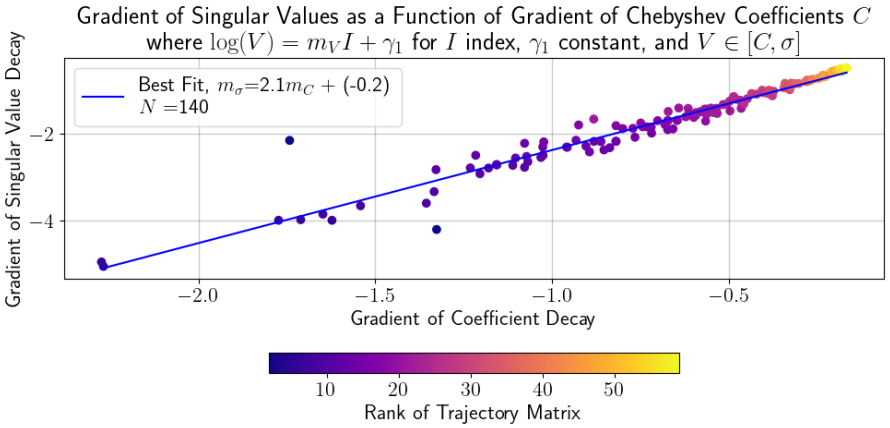


Figure 22: Relationship between the decays of singular values and Chebyshev coefficients for various functions.

former appears to be roughly twice the speed of the latter on a log graph, meaning a decay rate proportional to the square of the latter. To reformulate, we consider a function $f(x)$, with $x \in [-1, 1]$. Now consider the (bivariate) function $g(y, z) := f(y + z)$ for $y \in [-1, 0], z \in [0, 1]$. We seek to show that the singular values obtained from the representation of $g(y, z)$ into a matrix (which is Hankel) decay at roughly squared of the rate of the Chebyshev coefficients of the function $f(x)$. We conjecture that this is indeed the case given our experimentation, but ultimately leave the proof for future analysis.

9 Conclusions

Within this thesis, we have tackled the problem of analysing similarity within time series, particularly within the context of financial stock data. After delving into a review of the existing work that has been done on this topic in Section 2, we began with an overview of the data which we utilise: namely five year's worth of ticker information from nearly 500 different tickers from the S&P 500. This data set provides not only a very applicable real-world scenario within which to contextualise our results, but also presents real challenges in that there are no clear-cut trends inherent to all entries within the data set, as well as the inherent noisiness associated with financial data.

We then proceeded to discuss our main methodology for de-noising our data set: via Singular Spectrum Analysis. We discussed the outline of the methodology and also delve into the theoretical choice of the parameters associated with the method, justifying our choices based on both theoretical work and experimentally validated claims. In particular, we show how we can identify components within our model to real-world market behaviours, thereby allowing us to identify the dominant signals which we wish to capture and the noise which we wish to discard. Lastly, we discussed alternative methods and links to SSA within them, and further justify our choice of SSA and why we believe it to be the appropriate method to de-noise time series in such a context.

We then move towards answering the question of measuring similarity of two different time series. We provide an overview of various distance-based measures, in particular focusing on the "stiff" measure of Euclidean distance, and the "elastic measures" of Dynamic Time Warping and the Time Warp Edit Distance. We also discuss the differences between distances and similarity, which are often conflated in the literature and past analyses of time series, and construct an empirical test to decide upon a choice for similarity measure, which for our use case ends up being the Time Warp Edit Distance.

Having found a method to analyse when two time series are similar, we then go into an analysis of two different methods to find out when two time series begin to become *dissimilar*. Often neglected in similar analyses performed on time series, this question is particularly relevant to our context if we seek to set up a strategy wherein we are automatically able to determine when time series become similar or dissimilar, since these events mark times when it might be optimal execute either selling or buying of certain stocks. However, it is easy to see how this generic concept can be important in other contexts too: such as in wanting to analyse when temperature in two locations is similar or not to suggest strategies to deal with climate changes.

The first strategy we discuss is conceptually and computationally simple, wherein

we simply take a moving TWED score of the normalised time series and highlight when the score rises beyond a threshold to indicate dissimilarity. The issue with this method arose from the fact that the moving TWED score must be normalised on a ticker-level basis, with the normalisation depending on a qualitative examination of when the cross-similarity of any given tickers has changed enough over a longer timescale such that the mean and variance of the moving TWED score should be updated. In our second strategy, via RuLSIF, we attempt to rectify this by computing the points of change of the non-normalised moving TWED score. This then means that we are able to automatically detect when we enter different "regimes" of similarity - but comes with the drawback that we then require qualitative examination of when two time series are classified as "dissimilar".

We next introduce the notion of sector-level data into our analysis, and construct a measure of analysis in which a given sector is represented by the tickers corresponding to that sector. We are able to demonstrate some insights, namely that the Energy sector is comprised of tickers which are quite dissimilar to other sectors, and that sectors such as Finance and Information Technology display relatively high cross-similarity. Further, using the methodologies outlined previously, we are able to discuss which tickers and sectors are most similar to a given ticker.

We then finish off with a theoretical analysis on Hankel matrices. We show that certain classes of time series lend themselves to low-rank embedding within these types of matrices, and also examine the effect noise plays in the optimal SSA truncation rank; we are able to show that if we expect even a moderate amount of noise we must perform quite conservative rank truncations.

Lastly, we introduce a novel aspect of discussion, namely the relationship between the decay of the Chebyshev coefficients of low-rank functions and the decay of these functions' trajectory matrices' singular values. We conjecture that the latter decays are approximately proportional to the square of the decays of the former, though we leave the proof for future work.

9.1 Future Work

Our discussion has left much room for future work. In particular, we have opted not to focus much on machine-learning based approaches, due to two primary issues:

1. We are lacking in pre-labelled training data with which to train supervised machine learning models, such as Siamese Neural Networks (Hou, Jin, and Z. Zhao 2019). Though these models have shown to be effective when compared to DTW (Pei,

Tax, and Maaten 2016), there has not been much exploration on the performance of these methods against either TWED or the similarity measures we discussed.

2. When running unsupervised methods such as K-Means clustering, the inherent variations of the time series over the timespan we consider render constructing the groupings of similar tickers very difficult.

Thus, there is room for further analysis on finding machine learning based approaches in classifying similarity. Furthermore, there may also be alternative methods for change point detection, such as in using algorithms alternative to RuLSIF; further exploration is necessary to determine what is best for our scenario. There is also much examination to be done in terms of implementing an end-to-end trading strategy using the methods discussed (namely, taking input data, extracting signal, and tracking (dis)similarities) and back-testing it on historical data to verify that this is indeed a viable methodology. We can also explore the implementation of these methods on other (financial) assets: we have currently only explored stocks, but we may wish to check (for example) commodity prices to see if similar conclusions can be made as well.

Lastly, there is much to be explored in terms of the theoretical properties of Hankel matrices. In Section 8.1, we have outlined several experiments which demonstrate certain properties of noisy, low-rank time series and their embedding. However, we have thus far been unable to provide proofs for many results, in particular the relationship between the Chebyshev coefficients and the singular values of the trajectory matrices. We conclude with some questions which may guide further work, both pertaining to these theoretical discussions and ones brought up during the course of this report:

1. What exactly is the quantitative relationship between the decays of Chebyshev coefficients of smooth functions and their corresponding induced Hankel matrix singular values?

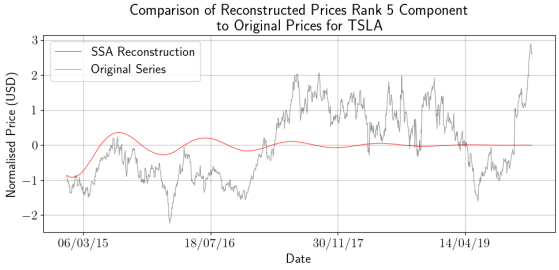
2. How does the rank of SSA truncation influence the reconstruction error of the final time series for low-rank, noisy time series?

3. What is the most effective methodology to detect when similar time series become dissimilar and vice-versa?

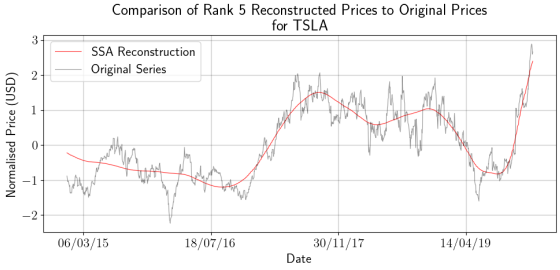
4. What is the best window size to set for these change point detection methods? Can be it automatically detected even for very noisy data?

We hope that these questions are useful in seeking launching points for further avenues of exploration.

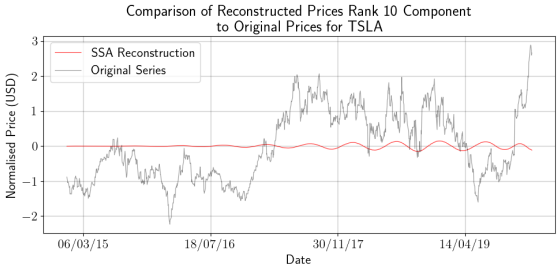
Appendix A Further Figures



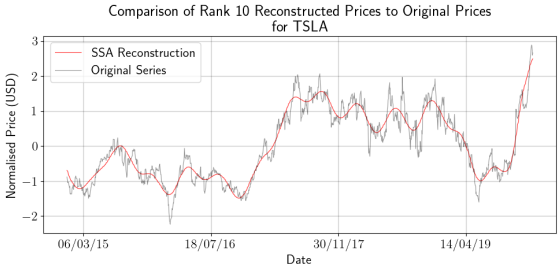
(a) Rank 5 SSA component of TSLA reconstruction.



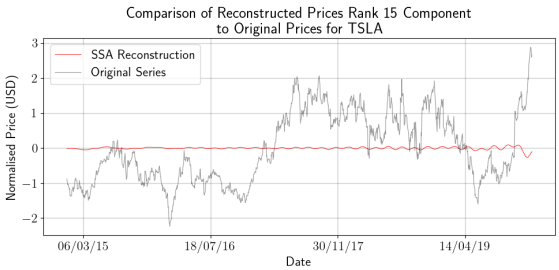
(b) Rank 5 SSA reconstruction of TSLA.



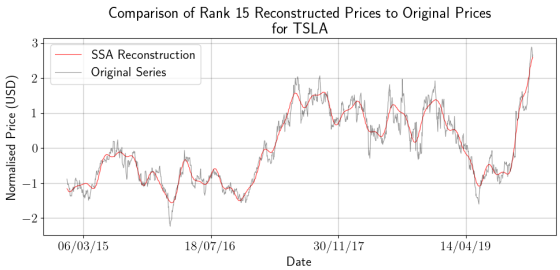
(c) Rank 10 SSA component of TSLA reconstruction.



(d) Rank 10 SSA reconstruction of TSLA.



(e) Rank 15 SSA component of TSLA reconstruction.



(f) Rank 15 SSA reconstruction of TSLA.

Figure 23: Comparison of SSA components and reconstructions of different ranks for TSLA. Note how as rank of reconstruction increases, the amplitudes of the movements fall, and the resolution of the reconstruction increases.

References

Aminikhanghahi, Samaneh and Diane J. Cook (May 1, 2017). "A survey of methods for time series change point detection". In: *Knowledge and Information Systems* 51.2, pp. 339–367. ISSN: 0219-3116. DOI: 10.1007/s10115-016-0987-z. URL: https://doi.org/10.1007/s10115-016-0987-z (visited on 07/19/2023).

Baig, Zella (July 1, 2023). *series_scorer: A Package for Time Series Smoothing and Similarity Scoring*. Coursework submitted for Python in Scientific Computing in the Department of Mathematics, University of Oxford as credit for an MSc MMSC.

Chen, Shihyen, Bin ma, and Kaizhong Zhang (Mar. 1, 2009). "On the Similarity Metric and the Distance Metric". In: *Theoretical Computer Science* 410, pp. 2365–2376. DOI: 10.1016/j.tcs.2009.02.023.

Elzinga, Cees H. (2014). "Distance, Similarity and Sequence Comparison". In: *Advances in Sequence Analysis: Theory, Method, Applications*. Ed. by Philippe Blanchard, Felix Bühlmann, and Jacques-Antoine Gauthier. Life Course Research and Social Policies. Cham: Springer International Publishing, pp. 51–73. ISBN: 978-3-319-04969-4. DOI: 10.1007/978-3-319-04969-4_4. URL: https://doi.org/10.1007/978-3-319-04969-4_4 (visited on 05/24/2023).

Elzinga, Cees H. and Matthias Studer (Nov. 1, 2019). "Normalization of Distance and Similarity in Sequence Analysis". In: *Sociological Methods & Research* 48.4. Publisher: SAGE Publications Inc, pp. 877–904. ISSN: 0049-1241. DOI: 10.1177/9124119867849. URL: https://doi.org/10.1177/0049124119867849 (visited on 05/24/2023).

Emms, Martin and Hector-Hugo Franco-Penya (2013). "On the Expressivity of Alignment-Based Distance and Similarity Measures on Sequences and Trees in Inducing Orderings". In: International Conference on Pattern Recognition Applications and Methods. Ed. by Pedro Latorre Carmona, J. Salvador Sánchez, and Ana L.N. Fred. Vol. 30. New York, NY: Springer New York, pp. 1–18. ISBN: 978-1-4614-5075-7 978-1-4614-5076-4. DOI: 10.1007/978-1-4614-5076-4_1. URL: http://link.springer.com/10.1007/978-1-4614-5076-4_1 (visited on 05/24/2023).

Frame, Peter and Aaron Towne (June 17, 2022). *Space-time POD and the Hankel matrix*. Number: arXiv:2206.08995. DOI: 10.48550/arXiv.2206.08995. arXiv: 2206.08995[cs,math]. URL: http://arxiv.org/abs/2206.08995 (visited on 07/15/2023).

Ghorbani, Behrooz, Song Mei, Theodor Misiakiewicz, and Andrea Montanari (Dec. 2021). "When do neural networks outperform kernel methods?*". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12. Publisher: IOP Publishing

and SISSA, p. 124009. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ac3a81. URL: https://dx.doi.org/10.1088/1742-5468/ac3a81 (visited on 08/11/2023).

GICS® - Global Industry Classification Standard (2023). URL: https://www.msci.com/our-solutions/indexes/gics (visited on 07/26/2023).

Goldstein, Louis (Dec. 5, 2015). *Lecture 26: Dynamic Time Warping - University of Southern California*. URL: https://sail.usc.edu/~lgoldste/Ling285/Slides/Lect26_handout.pdf (visited on 05/03/2023).

Golyandina, Nina (July 20, 2011). *On the choice of parameters in Singular Spectrum Analysis and related subspace-based methods*. Number: arXiv:1005.4374. arXiv: 1005.4374[stat]. URL: http://arxiv.org/abs/1005.4374 (visited on 05/01/2023).

Grabocka, Josif and Lars Schmidt-Thieme (Dec. 19, 2018). *NeuralWarp: Time-Series Similarity with Warping Networks*. Number: arXiv:1812.08306. DOI: 10.48550/arXiv.1812.08306. arXiv: 1812.08306[cs,stat]. URL: http://arxiv.org/abs/1812.08306 (visited on 05/08/2023).

Guo, Mingjun, Weiguang Li, Qijiang Yang, Xuezhi Zhao, and Yalian Tang (Mar. 15, 2020). "Amplitude filtering characteristics of singular value decomposition and its application to fault diagnosis of rotating machinery". In: *Measurement* 154, p. 107444. ISSN: 0263-2241. DOI: 10.1016/j.measurement.2019.107444. URL: https://www.sciencedirect.com/science/article/pii/S0263224119313119 (visited on 06/30/2023).

Halko, N., P. G. Martinsson, and J. A. Tropp (Jan. 2011). "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In: *SIAM Review* 53.2. Publisher: Society for Industrial and Applied Mathematics, pp. 217–288. ISSN: 0036-1445. DOI: 10.1137/090771806. URL: https://epubs.siam.org/doi/10.1137/090771806 (visited on 06/26/2023).

Hamilton, James D. (1994). *Time Series Analysis*. Princeton University Press. DOI: 10.2307/j.ctv14jx6sm. URL: https://www.jstor.org/stable/j.ctv14jx6sm (visited on 06/06/2023).

Hassani, Hossein, Rahim Mahmoudvand, and Mohammad Zokaei (Sept. 1, 2011). "Separability and window length in singular spectrum analysis". In: *Comptes Rendus Mathematique* 349.17, pp. 987–990. ISSN: 1631-073X. DOI: 10.1016/j.crma.2011.07.012. URL: https://www.sciencedirect.com/science/article/pii/S1631073X11001993 (visited on 05/02/2023).

Hassani, Hossein and Dimitrios Thomakos (2010). "A review on singular spectrum analysis for economic and financial time series". In: *Statistics and Its Interface* 3.3. Publisher: International Press of Boston, pp. 377–397. ISSN: 1938-7997. DOI: 10.

4310/SII.2010.v3.n3.a11. URL: https://www.intlpress.com/site/pub/pages/journals/items/sii/content/vols/0003/0003/a011/abstract.php (visited on 07/15/2023).

Hassani, Hossein, Allan Webster, Emmanuel Sirimal Silva, and Saeed Heravi (Feb. 1, 2015). "Forecasting U.S. Tourist arrivals using optimal Singular Spectrum Analysis". In: *Tourism Management* 46, pp. 322–335. ISSN: 0261-5177. DOI: 10.1016/j.tourman.2014.07.004. URL: https://www.sciencedirect.com/science/article/pii/S0261517714001368 (visited on 05/01/2023).

Hassani, Hossein, Mohammad Reza Yeganegi, Atikur Khan, and Emmanuel Sirimal Silva (Sept. 2020). "The Effect of Data Transformation on Singular Spectrum Analysis for Forecasting". In: *Signals* 1.1. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, pp. 4–25. ISSN: 2624-6120. DOI: 10.3390/signals1010002. URL: https://www.mdpi.com/2624-6120/1/1/2 (visited on 05/10/2023).

Hou, Linshan, Xiaofeng Jin, and Zhenshuang Zhao (Oct. 2019). "Time Series Similarity Measure via Siamese Convolutional Neural Network". In: *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pp. 1–6. DOI: 10.1109/CISP-BMEI48845.2019.8966048.

Hushchyn, Mikhail and Andrey Ustyuzhanin (July 1, 2021). "Generalization of change-point detection in time series data based on direct density ratio estimation". In: *Journal of Computational Science* 53, p. 101385. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2021.101385. URL: https://www.sciencedirect.com/science/article/pii/S1877750321000740 (visited on 08/11/2023).

Imani, Shima, Eamonn Keogh, Alireza Abdoli, Ali Beyram, and Azam Imani (Aug. 14, 2021). "Multi-window-finder: domain agnostic window size for time series data". In: MileTS '21: 7th KDD Workshop on Mining and Learning from Time Series. Singapore. URL: https://kdd-milets.github.io/milets2021/papers/MiLeTS2021_paper_9.pdf (visited on 07/27/2023).

Iungo, Giacomo Valerio and Edoardo Lombardi (Oct. 1, 2011). "A procedure based on proper orthogonal decomposition for time-frequency analysis of time series". In: *Experiments in Fluids* 51.4, pp. 969–985. ISSN: 1432-1114. DOI: 10.1007/s00348-011-1123-1. URL: https://doi.org/10.1007/s00348-011-1123-1 (visited on 07/27/2023).

Keogh, Eamonn and Shruti Kasetty (Jan. 10, 2003). "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Mining and

Knowledge Discovery 7(4), 349-371". In: *Data Mining and Knowledge Discovery* 7, pp. 349–371. DOI: 10.1023/A:1024988512476.

Keogh, Eamonn and Chotirat Ann Ratanamahatana (Mar. 1, 2005). "Exact indexing of dynamic time warping". In: *Knowledge and Information Systems* 7.3, pp. 358–386. ISSN: 0219-3116. DOI: 10.1007/s10115-004-0154-9. URL: https://doi.org/10.1007/s10115-004-0154-9 (visited on 06/21/2023).

Khan, Atikur R. and Hossein Hassani (Oct. 1, 2019). "Dependence measures for model selection in singular spectrum analysis". In: *Journal of the Franklin Institute* 356.15, pp. 8906–8928. ISSN: 0016-0032. DOI: 10.1016/j.jfranklin.2019.08.033. URL: https://www.sciencedirect.com/science/article/pii/S0016003219306076 (visited on 05/03/2023).

Liu, Song, Makoto Yamada, Nigel Collier, and Masashi Sugiyama (July 1, 2013). "Change-point detection in time-series data by relative density-ratio estimation". In: *Neural Networks* 43, pp. 72–83. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2013.01.012. URL: https://www.sciencedirect.com/science/article/pii/S0893608013000270 (visited on 07/19/2023).

Lütkepohl, Helmut and Fang Xu (2012). "The role of the log transformation in forecasting economic variables". In: *Empirical Economics* 42.3. Publisher: Springer, pp. 619–638. URL: https://ideas.repec.org//a/spr/empeco/v42y2012i3p619-638.html (visited on 05/10/2023).

Mahmoudvand, Rahim and Mohammad Zokaei (Feb. 1, 2011). "On the singular values of the Hankel matrix with application in singular spectrum analysis". In: *Chil. J. Stat.* 3, pp. 43–56.

Nakatsukasa, Yuji (May 3, 2023). *C6.1 Numerical Linear Algebra*. NLA Lecture notes. URL: https://courses.maths.ox.ac.uk/pluginfile.php/26688/mod_resource/content/19/NLA_lecture_notes.pdf (visited on 06/26/2023).

Pei, Wenjie, David M. J. Tax, and Laurens van der Maaten (Mar. 15, 2016). *Modeling Time Series Similarity with Siamese Recurrent Networks*. Number: arXiv:1603.04713. DOI: 10.48550/arXiv.1603.04713. arXiv: 1603.04713[cs]. URL: http://arxiv.org/abs/1603.04713 (visited on 05/08/2023).

Rakthanmanon, Thanawin, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh (Sept. 2013). "Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping". In: *ACM transactions on knowledge discovery from data* 7.3, p. 10. ISSN: 1556-4681. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6790126/ (visited on 05/11/2023).

Rodríguez-Aragón, Licesio and Anatoly Zhigljavsky (Jan. 1, 2010). “Singular spectrum analysis for image processing”. In: *Statistics and Its Interface* 3. DOI: 10.4310/SII.2010.v3.n3.a14.

Schoenberg, I. J. (1938). “Metric spaces and positive definite functions”. In: *Transactions of the American Mathematical Society* 44.3, pp. 522–536. ISSN: 0002-9947, 1088-6850. DOI: 10.1090/S0002-9947-1938-1501980-0. URL: https://www.ams.org/tran/1938-044-03/S0002-9947-1938-1501980-0/ (visited on 02/21/2023).

Serrà, Joan and Josep Lluis Arcos (Sept. 2014). “An Empirical Evaluation of Similarity Measures for Time Series Classification”. In: *Knowledge-Based Systems* 67, pp. 305–314. ISSN: 09507051. DOI: 10.1016/j.knosys.2014.04.035. arXiv: 1401.3973[cs,stat]. URL: http://arxiv.org/abs/1401.3973 (visited on 05/02/2023).

Vlachos, Michail (2017). “Similarity Measures”. In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, pp. 1163–1166. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_766. URL: https://doi.org/10.1007/978-1-4899-7687-1_766 (visited on 05/22/2023).

Wagner, Robert A. and Michael J. Fischer (Jan. 1, 1974). “The String-to-String Correction Problem”. In: *Journal of the ACM* 21.1, pp. 168–173. ISSN: 0004-5411. DOI: 10.1145/321796.321811. URL: https://dl.acm.org/doi/10.1145/321796.321811 (visited on 06/21/2023).

Wang, Rui, Hong-Guang Ma, Guo-Qing Liu, and Dong-Guang Zuo (Jan. 23, 2015). “Selection of window length for singular spectrum analysis”. In: *Journal of the Franklin Institute* 352. DOI: 10.1016/j.jfranklin.2015.01.011.

Wang, Xiaoyue, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh (Dec. 9, 2010). *Experimental Comparison of Representation Methods and Distance Measures for Time Series Data*. Number: arXiv:1012.2789. DOI: 10.48550/arXiv.1012.2789. arXiv: 1012.2789[cs]. URL: http://arxiv.org/abs/1012.2789 (visited on 06/21/2023).

Weiss, Julien (2019). “A Tutorial on the Proper Orthogonal Decomposition”. In: 2019 AIAA Aviation Forum. Dallas, Texas, United States. URL: https://depositonce.tu-berlin.de/handle/11303/9456 (visited on 06/28/2023).

Yahoo Finance – stock market live, quotes, business & finance news (2023). URL: https://uk.finance.yahoo.com/ (visited on 06/27/2023).

Yamada, Makoto, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama (May 1, 2013). “Relative Density-Ratio Estimation for Robust Distribution

Comparison". In: *Neural Computation* 25.5, pp. 1324–1370. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00442. URL: https://doi.org/10.1162/NECO_a_00442 (visited on 07/28/2023).

Zha, Daochen, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu (Jan. 6, 2022). *Towards Similarity-Aware Time-Series Classification*. Number: arXiv:2201.01413. DOI: 10.48550/arXiv.2201.01413. arXiv: 2201.01413[cs]. URL: http://arxiv.org/abs/2201.01413 (visited on 05/08/2023).

Zhao, Xuezhi and Bangyan Ye (Jan. 1, 2019). "Separation of Single Frequency Component Using Singular Value Decomposition". In: *Circuits, Systems, and Signal Processing* 38.1, pp. 191–217. ISSN: 1531-5878. DOI: 10.1007/s00034-018-0852-2. URL: https://doi.org/10.1007/s00034-018-0852-2 (visited on 06/13/2023).

Zhigljavsky, Vladimir Nekrutkin and Nina Golyandina Anatoly A. (Jan. 23, 2001). *Analysis of Time Series Structure: SSA and Related Techniques*. New York: Chapman and Hall/CRC. 320 pp. ISBN: 978-0-367-80168-7. DOI: 10.1201/9780367801687.