

# MARKET REGIME CLASSIFICATION WITH SIGNATURES

PAUL BILOKON, ANTOINE JACQUIER, AND CONOR MCINDOE

ABSTRACT. We provide a data-driven algorithm to classify market regimes for time series. We utilise the path signature, encoding time series into easy-to-describe objects, and provide a metric structure which establishes a connection between separation of regimes and clustering of points.

## 1. INTRODUCTION

Market regimes are a clear feature of market data time series, with notions such as bull markets, bear markets, periods of calm and those of turmoil being commonplace in discussions between practitioners. As the saying goes, liquidity begets liquidity; regimes may be self-reinforcing for some time before a market shift is observed. The dramatic market event of early 2020, brought about by the COVID-19 pandemic, brought with it high volatility and low liquidity. History does indeed seem to repeat itself: similar conditions were observed in many crashes over the last century, such as the 1929 Wall Street crash, the Black Monday event of 1987, the 2008 global crisis, and the 2015-2016 Chinese stock market crash.

The ability of an investor to recognise the underlying economic and market conditions and, ideally, to estimate the transition probabilities between market regimes, has long sought attention. Kritzman, Page and Turkington [10] used Markov-switching models to characterise regimes for portfolio allocation. Jiltsov [8] has used Hidden Markov models on equity data to identify market clusters to capture opinions on credit risk of large banks. This approach is partially data-driven, in that the number of regimes is initially asserted, and the resulting classifications are analysed once fitted. Another approach, as in [15], is to decide on some set of regimes and then segregate market data into the predetermined categories.

In this paper, we seek a framework which is a data-driven as possible, without specifying either the number or characteristics of the final regimes. To do so, we make intensive use of the signature of a path, which originated in [3] and was developed in the context of rough paths by Lyons and coauthors [2, 7, 12, 13, 14]. The path signature has recently received much attention, proving itself to be a natural language to encode time series data in a form suitable for machine learning tasks. The key idea of the present contribution is to use these path signatures as points in some suitable metric space that can then be classified using a clustering algorithm.

In Section 2, we present the Azran and Ghahramani clustering algorithm [1] and show how to apply it to finite-dimensional data. We recall in Section 3 the basic definitions and properties of path signatures. Finally, in Section 4, we show how to incorporate signatures as points in the clustering algorithm and how to define a suitable notion of distance between these points. We provide a numerical example on synthetic data as a demonstration of the algorithm.

---

*Date:* June 30, 2021.

The `Python` code related to the results in this paper is available at <https://github.com/mcindoe/regimedetection>.

## 2. DATA-DRIVEN CLUSTERING

We recall here the data-driven clustering algorithm by Azran and Ghahramani [1] (AG-algorithm) over arbitrary metric spaces. The algorithm is purely data-driven, in that the number and shape of clusters is left unspecified and both are suggested by the algorithm.

**2.1. The Azran-Ghahramani clustering.** We consider a given metric space with a distance function  $\mathfrak{d}$  and a collection of points  $\mathcal{X} = \{x_1, \dots, x_n\}$ .

**Definition 2.1.** A similarity function  $\mathfrak{w} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is a monotonically decreasing function. Given  $(\mathcal{X}, \mathfrak{d})$  as above, the similarity matrix is the matrix  $W = (w_{i,j})_{1 \leq i, j \leq n}$  defined as  $w_{i,j} := \mathfrak{w}(\mathfrak{d}(x_i, x_j))$ .

The  $(i, j)$ -entry of  $W$  is the similarity between points  $x_i$  and  $x_j$ . Natural candidates for the similarity function include the inverse function  $\mathfrak{w}(x) = x^{-1}$ , the squared inverse function  $\mathfrak{w}(x) = x^{-2}$  and the Gaussian  $\mathfrak{w}(x) = \exp(-x^2)$ . We further define the matrix  $P := D^{-1}W$ , where  $D$  is the diagonal matrix in  $\mathcal{M}_n(\mathbb{R})$  with  $D_{ii} := \sum_{j=1}^n w_{ij}$ , so that  $P$  corresponds to a transition matrix. We introduce the natural notion of cluster as a set of points that are close to each other:

**Definition 2.2.** A cluster of size  $k \leq n$  is a subset  $\mathcal{C} \subset \mathcal{X}$  such that  $\min_{x,y \in \mathcal{C}} \mathfrak{w}(\mathfrak{d}(x,y)) > \max_{x \in \mathcal{C}, y \in \mathcal{X} \setminus \mathcal{C}} \mathfrak{w}(\mathfrak{d}(x,y))$ .

We consider the random walk of  $n$  particles  $X_1, \dots, X_n$ , starting from  $x_1, \dots, x_n$  respectively, whose location evolves according to the homogeneous Markov chain with transition matrix  $P$ , so that  $P_{i,j}$  is the probability that the particle moves from  $x_i$  to  $x_j$  between two time steps. Let  $X_i(t)$  denote the (row vector) discrete distribution of the  $i^{\text{th}}$  particle after  $t$  steps. Clearly  $X_i(0)$  is a Dirac mass centered at  $x_i$ ; at any (discrete) time  $t \geq 1$ , the discrete distribution of the  $i^{\text{th}}$  particle is given by

$$(2.1) \quad X_i(t) = X_i(t-1)P = \dots = X_i(0)P^t = \begin{pmatrix} 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{pmatrix} P^t,$$

where the 1 is in the  $i^{\text{th}}$  position, as a placeholder for the position  $x_i$ .

In a well-clustered space, where similarities between points of the same cluster are high and between points of distinct clusters are low, we expect particles to mostly remain within their clusters throughout the random walk. So, after a sufficient number of steps, the distributions of particles beginning in these well-separated clusters should be similar. It follows then by (2.1) that the corresponding rows of  $P^t$  will be similar. Note that this establishes a correspondence between similar points in the metric space and similar rows in the matrix  $P^t$ . We summarise the following important properties of the matrix  $P$  [1, Lemma 1, Lemma 2]:

**Lemma 2.3.** *If  $W$  is full rank, then so is  $P$ . The spectrum of  $P$  is of the form  $1 = \lambda_1 \geq \dots \geq \lambda_n \geq -1$ . Let  $v_k$  be the eigenvector corresponding to  $\lambda_k$ , chosen with unit norm. Then, for any  $t \geq 1$ ,*

$$(2.2) \quad P^t = \sum_{k=1}^n \lambda_k^t A_k,$$

with  $A_k := \frac{v_k v_k^T}{v_k^T D v_k} D$  idempotent and orthogonal,  $\{A_k\}_{k=1}^n$  forming a basis of the space generated by  $\{P^k\}_{k \geq 1}$ .

From (2.2), we see that eigenvalues close to 1 correspond to more stable basis elements, whereas small eigenvalues quickly shrink to zero as time evolves. As discussed in [9, Assumption A1], the special case of  $K$  separated clusters with no connections between points in different clusters (i.e. zero similarity between such points) corresponds to the eigenvalues satisfying  $1 = \lambda_1 = \dots = \lambda_K > \lambda_{K+1}$ . In this case (2.2) implies that  $P^t$

converges to  $\sum_{k=1}^K A_k$  as  $t$  tends to infinity. This motivates the following definition, which gathers the essential tools required to build the clustering algorithm:

**Definition 2.4.** For  $t \geq 1$  and  $k = 1, \dots, K$ , define the  $k^{\text{th}}$  eigengap after  $t$  steps by  $\Delta_k(t) := \lambda_k^t - \lambda_{k+1}^t$  and let  $\mathcal{K}_t := \operatorname{argmax}_k \Delta_k(t)$  be the  $k$ -eigengap which is largest after  $t$  time steps, at which point we say  $k$  clusters are revealed. For a given number of clusters,  $k$ ,  $T_k := \{t : \mathcal{K}_t = k\}$  represents the set of all time steps at which  $k$  clusters are revealed. We call  $t_k := \operatorname{argmax}_{t \in T_k} \Delta_k(t)$  the  $k$ -cluster revealer, at which point the  $k$  clusters are best segregated. Finally, we call  $\Delta(t) := \max_{k \in \{1, \dots, n\}} \Delta_k(t)$  the maximal eigengap separation after  $t$  steps.

We use the term  $k$ -clustering for a partition of  $\mathcal{X}$  into  $k$  subsets, and say that a  $k'$ -clustering is better revealed than a  $k$ -clustering after  $t$  steps if  $\Delta_{k'}(t) > \Delta_k(t)$ .

The AG-algorithm suggests  $k$ -clusterings for values of  $k$  for which there exists some number of steps  $t$ , at which  $k$  clusters is better revealed than any other number of clusters  $k'$ . If, however, there exists some  $k' \neq k$  for which  $\Delta_k(t_k) < \Delta_{k'}(t_k)$ , then  $k$  is not considered a suitable number of clusters for the data. We are therefore interested in computing the set of time steps  $\mathcal{T} := \{t_1, \dots, t_m\}$  where  $\Delta(\cdot)$  attains a local maxima, and then for each  $t_i \in \mathcal{T}$  computing the number of clusters,  $k_i := \mathcal{K}_{t_i}$ , where  $\Delta_{k_i}(t_i) = \Delta(t_i)$ . Finally, the suggested  $k_i$ -clustering of the points is inferred by finding a  $k_i$ -clustering of the rows of the matrix  $P^{t_i}$ . The value  $\Delta_{k_i}(t_i)$ , bounded above by 1, provides a measure of the separation of clusters in the returned partition, as motivated by the discussion before Definition 2.4.

We summarise the process in Algorithm 1, deferring for now the discussion of how to compute the  $k$ -clustering for given  $k$ .

---

**Algorithm 1:** Multiscale  $k$ -Prototypes Algorithm

---

**Input:** Metric space  $(\mathcal{X}, \mathfrak{d})$ , maximum number of steps  $T$

**Output:** Collection of suggested partitions with the corresponding eigengap separations

- (i) Compute  $P$  and its spectrum  $\lambda_1 \geq \dots \geq \lambda_n$ ;
  - (ii) Compute  $\Delta(t)$  for  $t \in \{1, \dots, T\}$ . Find the set of local maxima  $\mathcal{T} := \{t_1, \dots, t_m\}$ ;
  - (iii) For each  $t_i \in \mathcal{T}$ , find the number of clusters  $k_i$  best revealed by  $t_i$  steps;
  - (iv) For each  $k_i$ , compute the corresponding  $k_i$ -partitioning  $\mathcal{I}_{k_i}$ ;
  - (v) Return the final collection  $\{(\mathcal{I}_{k_1}, \Delta(t_{k_1})), \dots, (\mathcal{I}_{k_m}, \Delta(t_{k_m}))\}$ .
- 

In order to determine the  $k$ -clusterings  $\mathcal{I}_k$  in Algorithm 1, we make use of an algorithm which is similar to  $k$ -means clustering, called the  $k$ -prototypes algorithm (the word ‘prototype’, borrowed from [1], refers to the distribution vectors of the particles following the random walk). In  $k$ -means clustering, a distance between vectors is used to separate points into  $k$  clusters. Here we have distributions, and hence a slightly different approach is suggested in [1], making use of the Kullback-Leibler divergence which we now recall.

**Definition 2.5.** Given two probability distributions  $\mathbb{P}$  and  $\mathbb{Q}$  on  $\mathcal{X}$ , the Kullback-Leibler (KL) divergence from  $\mathbb{Q}$  to  $\mathbb{P}$  is defined as

$$\text{KL}(\mathbb{P} \parallel \mathbb{Q}) := \sum_{x \in \mathcal{X}} \mathbb{P}(x) \log \left( \frac{\mathbb{P}(x)}{\mathbb{Q}(x)} \right)$$

The Kullback-Leibler divergence is not a proper distance function since it is not symmetric, but nevertheless gives a notion of disparity between two probability distributions, and is hence used in the  $k$ -prototype algorithm [1] to compare the distributions of the particles. As argued in [1], the usual Euclidean distance is not

adapted here as it gives all elements the same weights. For a given number of clusters  $k$  and number of steps  $t$ , a suitable  $k$ -clustering is identified by minimising the function

$$\sum_{j=1}^k \sum_{m \in \mathcal{I}_j} \text{KL}(\mathbf{P}_m^t \| Q_j),$$

over all partitioning  $\mathcal{I} = (\mathcal{I}_1, \dots, \mathcal{I}_k)$  and distributions  $(Q_1, \dots, Q_k)$ . We refer to [1] for a precise detail of the recursive algorithm to solve this non-convex optimisation problem, together with some specification about the initial starting point of the algorithm. Their approach is summarised in Algorithm 2.

---

**Algorithm 2:**  $k$ -Prototypes Algorithm
 

---

**Input:** Transition matrix  $\mathbf{P}^t \in \mathbb{R}^{n \times n}$ , number of clusters  $k$ , initial matrix  $Q \in \mathbb{R}^{k \times n}$  of prototypes;

**Initialisation:**  $Q^{(old)} := Q$ ;

**Output:** Partition  $\mathcal{I}$ , of size  $k$ , of the indices  $\{1, \dots, n\}$ ;

(i) For  $j \in \{1, \dots, k\}$ ,  $\mathcal{I}_j^{(new)} := \left\{ m : j = \operatorname{argmin}_{j \in \{1, \dots, k\}} \text{KL}(\mathbf{P}_m^t \| Q_j^{(old)}) \right\}$ ;

(ii) For  $j \in \{1, \dots, k\}$ , define  $Q_j^{(new)} := |\mathcal{I}_j^{(new)}|^{-1} \sum_{m \in \mathcal{I}_j^{(new)}} \mathbf{P}_m^t$ ;

(iii) Set  $Q^{(old)} := Q^{(new)}$  and return to (i) until convergence or stop criterion.

---

Note that step (ii) of Algorithm 2 is not well-defined if any of the partition elements  $\mathcal{I}_j$  are empty. Our approach in this case is to first compute the prototypes  $Q_j$  for all  $j$  where  $|\mathcal{I}_j| > 0$ , and then iteratively define the remaining prototypes  $Q_m$  to be the row of the transition matrix  $\mathbf{P}^t$  for which the minimum KL-divergence to all currently-defined prototypes is maximised. This ensures the space remains well-covered by the cluster prototypes; a similar technique is discussed to initialise the matrix of prototypes in [1, Section 4.2]

**2.2. Gaussian Clouds.** We now turn to applications of the AG-algorithm, and begin with an example in  $\mathbb{R}^2$ , equipped with the Euclidean distance. For  $m \in \mathbb{R}^2$ ,  $\sigma > 0$ ,  $n \in \mathbb{N}$ , we denote by  $\mathcal{X}_m^\sigma(n)$  a Gaussian Cloud of size  $n$ , centre  $m$  and variance  $\sigma^2$ , that is a collection  $\{a_1, \dots, a_n\}$  where each  $a_i$  is Gaussian with mean  $m$  and covariance matrix  $\sigma^2 I_2$ . Figure 1 shows the generated points, with four cluster centres, each cluster consisting of 100 points.

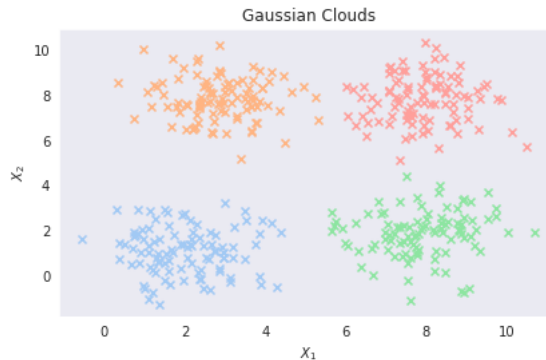


FIGURE 1. Gaussian Clouds where  $\sigma$  is set equal to one for all four clusters and the centers  $m$  are  $(2, 1)$ ,  $(3, 8)$ ,  $(8, 2)$ ,  $(8, 8)$ .

We make use of the Gaussian similarity function:

$$(2.3) \quad \mathbf{w}^\xi(x) := \exp\left(-\frac{x}{\xi^2}\right).$$

The optimal choice of  $\xi$  is not entirely obvious and, following [1], we choose it to be the 1% low-value quantile of the non-zero distances in the space. In Figure 2, we demonstrate in the left-hand plot the curves  $\Delta_k(\cdot)$  for some small values of  $k$ . The right-hand plot is the curve  $\Delta(\cdot)$ , which we recall as the corresponding maximum over all  $k$ . The local maxima of this latter curve represents points at which some clustering is best revealed.

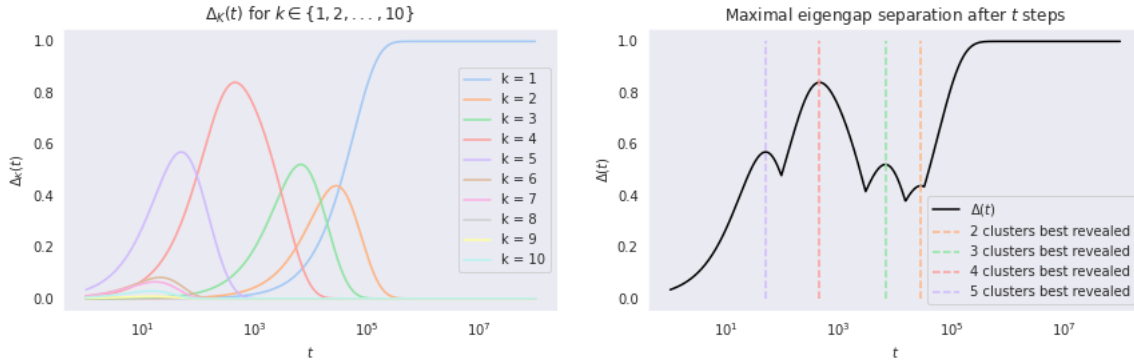


FIGURE 2. Gaussian Clouds -  $\Delta_k(t)$  and  $\Delta(t)$ .

The suggested 3-, 4- and 5-clusterings are displayed in Figure 3. The suitability of the resulting partitions, as indicated by the eigengap separation, suggests that the preferred partition of the points is the 4-clustering, followed by the 5-clustering and finally the 3-clustering. The recommended 4-clustering is almost the same but not identical to the original problem specification.

### 3. CLUSTERING PATHS VIA SIGNATURES

We now move on to the key part of the paper, demonstrating how the AG-algorithm above can be applied to time series. The obvious two hurdles to overcome are the infinite-dimension feature of time series and a suitable notion of distance. To do so, we summarise in Section 3.1 the information contained in a series in its so-called signature, and show then how this helps us extend the AG-algorithm.

**3.1. An overview of signatures.** We provide an overview of signatures of paths for completeness. They date back to Chen [3], but have received widespread investigation in the context of rough paths [2, 7]. Signatures can be defined for large classes of functions of bounded variation [5], but we restrict our analysis for simplicity to the case of piecewise smooth functions. A path  $\gamma$  is a continuous map from  $[a, b]$  to  $\mathbb{R}^d$  ( $d \in \mathbb{N}$ ). If the map  $t \mapsto \gamma_t$  is differentiable, the integral of a function  $f : \mathbb{R}^d \rightarrow \mathbb{R}^p$  along  $\gamma$  is classically defined by  $\int_a^b f(\gamma_t) d\gamma_t := \int_a^b f(\gamma_t) \dot{\gamma}_t dt$ . In the case of piecewise differentiable curves  $\mathcal{P}_{a,b}^d$  from  $[a, b]$  to  $\mathbb{R}^d$ , we may extend this to

$$\int_a^b f(\gamma_t) d\gamma_t := \int_{x_1}^{x_2} f(\gamma_t) d\gamma_t + \int_{x_2}^{x_3} f(\gamma_t) d\gamma_t + \dots + \int_{x_{n-1}}^{x_n} f(\gamma_t) d\gamma_t \in \mathbb{R}^p,$$

where  $a = x_1 < x_2 < \dots < x_n = b$  is a partition of  $[a, b]$  with  $\gamma$  differentiable over each interval  $(x_i, x_{i+1})$ .

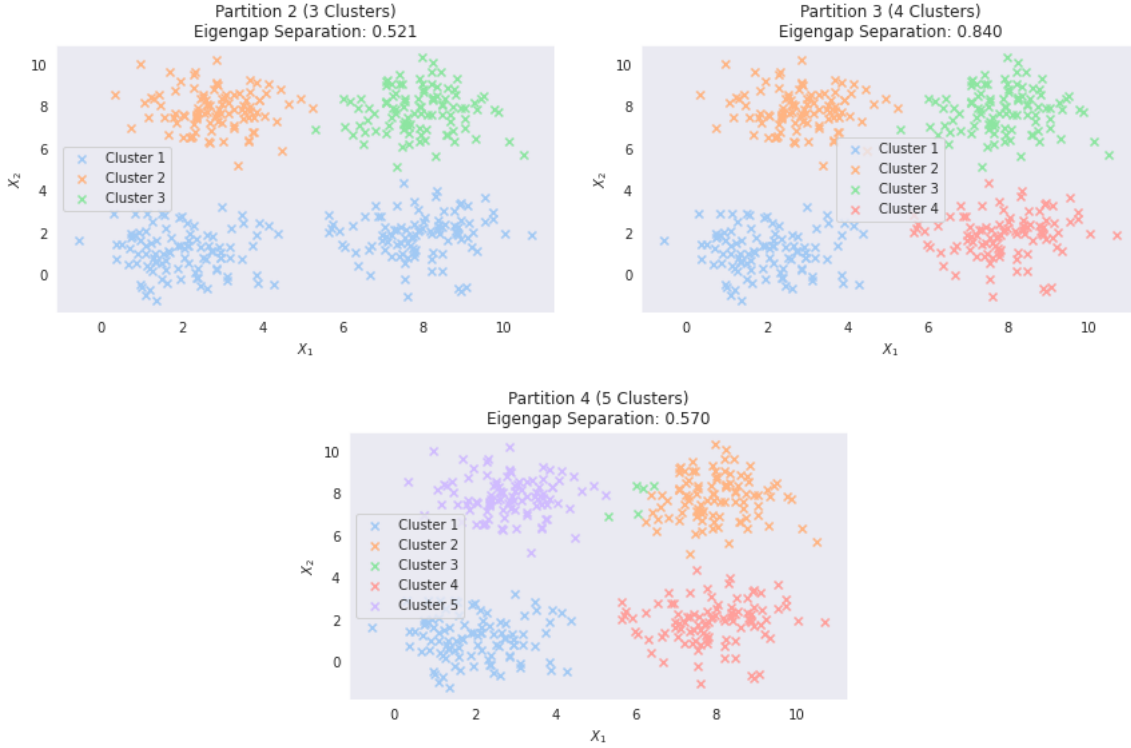


FIGURE 3. Gaussian Clouds - 3 suggested clusterings

**Definition 3.1.** Let  $\gamma \in \mathcal{P}_{a,b}^d$ . For  $k \in \mathbb{N}$  and  $\mathbf{i} = (i_1, \dots, i_k) \in \{1, \dots, d\}^k$ , we define recursively, for  $t \in (a, b)$ ,

$$\begin{aligned} \mathcal{S}(\gamma)_{a,t}^{\mathbf{i}} &:= \int_{(a,t)} d\gamma_s^{\mathbf{i}}, \quad \text{for } i = 1, \dots, d, \\ \mathcal{S}(\gamma)_{a,t}^{\mathbf{i}} &:= \int_{a < t_1 < \dots < t_k < t} d\gamma_{t_1}^{i_1} \dots d\gamma_{t_k}^{i_k} = \int_{a < s < t} \mathcal{S}(\gamma)_{a,s}^{i_1, \dots, i_{k-1}} d\gamma_s^{i_k}. \end{aligned}$$

The path signature is then defined as the collection of all such iterated integrals. This can be stated in a more elegant way though, via the notion of alphabets. For a given alphabet  $A$ , i.e. a (finite or not) sequence of elements, a word  $x$  of length  $|x| \in \mathbb{N}$  is an ordered sequence  $x_1 x_2 \dots x_{|x|}$  with  $x_i \in A$  for  $i \in \{1, \dots, |x|\}$ . We denote by  $\mathcal{W}(A)$  the set of all possible words, including the empty word  $\varepsilon$  (of length zero). For example, the set of words of length 3 on  $A := \{a, b\}$  is  $\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$ , while the set of all words on  $A$  is  $\mathcal{W}(A) = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$ . If an ordering  $<$  exists on the alphabet, we can extend it recursively to  $\mathcal{W}(A)$ : the concatenation of  $x, y \in \mathcal{W}(A)$  is defined to be the word  $xy := x_1, \dots, x_{|x|}, y_1, \dots, y_{|y|}$ ; then by setting  $\varepsilon < x$  whenever  $|x| > 0$ , we say, for any  $a, b \in A$ , that  $ax < by$  if either  $a < b$  (in  $A$ ), or  $a = b$  and  $x < y$ .

**Definition 3.2.** For  $\gamma \in \mathcal{P}_{a,b}^d$ , the path signature  $\mathcal{S}(\gamma)_{a,b}$  is the collection of iterated integrals  $\mathcal{S}(\gamma)_{a,b}^{\mathbf{i}}$  with  $\mathbf{i} \in \mathcal{W}(\{1, \dots, d\})$  and  $\mathcal{S}(\gamma)_{a,b}^{\varepsilon} := 1$ . The truncated signature  $\mathcal{S}(\gamma)_{a,b}^{\leq n}$  up to level  $n \in \mathbb{N}$  is the restriction of  $\mathcal{S}(\gamma)_{a,b}^{\mathbf{i}}$  to  $|\mathbf{i}| \leq n$ . The terms in the signature appear in the same order as the corresponding words in the natural ordered alphabet:

$$\mathcal{S}(\gamma)_{a,b} := (\mathcal{S}(\gamma)_{a,b}^{\varepsilon}, \mathcal{S}(\gamma)_{a,b}^1, \dots, \mathcal{S}(\gamma)_{a,b}^d, \mathcal{S}(\gamma)_{a,b}^{11}, \mathcal{S}(\gamma)_{a,b}^{12}, \dots)$$

The following proposition demonstrates that the starting point of the path is lost when projecting to the signature: indeed, the signature is invariant to translations of the original path.

**Proposition 3.3** (Translation invariance). *For any  $\gamma \in \mathcal{P}_{a,b}^d$ ,  $c \in \mathbb{R}^d$  and  $\mathbf{i} \in \mathcal{W}(\{1, \dots, d\})$ , we have*

$$\mathcal{S}(\gamma + c)_{a,b}^{\mathbf{i}} = \mathcal{S}(\gamma)_{a,b}^{\mathbf{i}}$$

*Proof.* Write  $c = (c^1, \dots, c^d)$ ,  $\tilde{\gamma} := \gamma + c$ , and take  $i \in \{1, \dots, d\}$ . We have  $\tilde{\gamma}_t^i = \gamma_t^i + c^i$ , hence

$$\mathcal{S}(\tilde{\gamma})_{a,b}^i = \int_a^b d\tilde{\gamma}_t^i = \int_a^b \frac{d\tilde{\gamma}_t^i}{dt} dt = \int_a^b \frac{d\gamma_t^i}{dt} dt = \mathcal{S}(\gamma)_{a,b}^i,$$

which shows the result for any word of length one. Now suppose the result holds for any  $k$ -length word. Let  $i_1 \dots i_k i_{k+1}$  be a  $(k+1)$ -length word; we have

$$\mathcal{S}(\tilde{\gamma})_{a,b}^{i_1, \dots, i_k, i_{k+1}} = \int_a^b \mathcal{S}(\tilde{\gamma})_{a,b}^{i_1, \dots, i_k} \frac{d\tilde{\gamma}_t^{i_{k+1}}}{dt} dt = \int_a^b \mathcal{S}(\gamma)_{a,b}^{i_1, \dots, i_k} \frac{d\gamma_t^{i_{k+1}}}{dt} dt = \mathcal{S}(\gamma)_{a,b}^{i_1, \dots, i_k, i_{k+1}},$$

so the terms of the signatures of each path agree for any word in  $\mathcal{W}(\{1, \dots, d\})$ , hence for the entire signature.  $\square$

**3.1.1. Logsignatures.** The logsignature is a more concise representation of the information present in a signature. We introduce the vector space  $\mathcal{V}$  of non-commutative formal power series on a basis of symbols  $B = \{e_1, \dots, e_r\}$ , that is the set of elements of the form  $\sum_{w \in \mathcal{W}(B)} \lambda_w w$ ,  $\lambda_w \in \mathbb{R}$ , with possibly many null coefficients, where for now  $\lambda w$  is interpreted as a formal symbol rather than a multiplication. We endow  $\mathcal{V}$  with a vector space structure, namely for any  $\lambda, \mu \in \mathbb{R}$  and  $w \in \mathcal{W}(B)$ , both  $\lambda(\mu w) := (\lambda\mu)w$  and  $\lambda w + \mu w := (\lambda + \mu)w$  hold. We define  $\otimes : \mathcal{V} \times \mathcal{V} \rightarrow \mathcal{V}$  as the concatenation  $w \otimes v := wv$ , so that  $\mathcal{V}$  acquires the structure of an algebra. For a path  $\gamma \in \mathcal{P}_{a,b}^d$ , we may identify its signature with the non-commutative formal power series

$$(3.1) \quad \mathcal{S}(\gamma)_{a,b} = 1 + \mathcal{S}(\gamma)_{a,b}^1 e_1 + \dots + \mathcal{S}(\gamma)_{a,b}^d e_d + \mathcal{S}(\gamma)_{a,b}^{1,1} e_1 e_1 + \mathcal{S}(\gamma)_{a,b}^{1,2} e_1 e_2 + \dots$$

The  $=$  sign is a slight abuse of notation, but is justified as a one-to-one correspondence between signatures of  $d$ -dimensional paths and the non-commutative formal power series in  $d$  letters. Given a power series  $\omega \in \mathcal{V}$ , where the coefficient  $\lambda_0$  of the empty word is not null, we define

$$(3.2) \quad \log \omega := \log(\lambda_0) + \sum_{n \geq 1} \frac{(-1)^n}{n} \left(1 - \frac{\omega}{\lambda_0}\right)^{\otimes n},$$

and thus deduce the logsignature  $\log \mathcal{S}(\gamma)_{a,b}$  of  $\gamma$  of a path  $\gamma \in \mathcal{P}_{a,b}^d$ .

From Definition 3.1, the level-one term  $\mathcal{S}(\gamma)_{a,b}^i$  corresponds to the displacement of the coordinate path  $\gamma^i$  between times  $a$  and  $b$ . In fact, this  $i^{\text{th}}$  displacement fully determines the  $k$ -fold iterated integral over the  $i^{\text{th}}$  index:

**Proposition 3.4.** *For  $\gamma \in \mathcal{P}_{a,b}^d$ , the identity  $\mathcal{S}(\gamma)_{a,b}^{\overbrace{i, \dots, i}^{k \text{ times}}} = (\gamma_b^i - \gamma_a^i)^k / k!$  holds for any  $i \in \{1, \dots, d\}$ ,  $k \in \mathbb{N}$ .*

*Proof.* Let  $i \in \{1, \dots, d\}$ . The  $k = 1$  case is trivial. Suppose the result holds for  $k \in \mathbb{N}$ ; by induction

$$\mathcal{S}(\gamma)_{a,b}^{\overbrace{i, \dots, i}^{k+1 \text{ times}}} = \int_a^b \frac{(\gamma_t^i - \gamma_a^i)^k}{k!} \frac{d\gamma_t^i}{dt} dt = \left[ \frac{(\gamma_t^i - \gamma_a^i)^{k+1}}{(k+1)!} \right]_a^b = \frac{(\gamma_b^i - \gamma_a^i)^{k+1}}{(k+1)!}.$$

$\square$

This shows that the signature of a one-dimensional path is completely determined by its displacement. It should be noted however that the time series of price paths, as considered in this paper, are not one-dimensional but instead are two-dimensional processes with time in the first coordinate, namely  $\{(t_1, x_1), \dots, (t_n, x_n)\}$  instead of  $\{x_1, \dots, x_n\}$ .

Another important result connects second-order signature terms to the product of first-order terms. We provide an understanding of this result in the case where the curve is the concatenation of paths which are piecewise monotone, and postpone its proof to Appendix A.

**Lemma 3.5.** *The equality  $\mathcal{S}(\gamma)_{a,b}^{i,j} + \mathcal{S}(\gamma)_{a,b}^{j,i} = \mathcal{S}(\gamma)_{a,b}^i \mathcal{S}(\gamma)_{a,b}^j$  holds for any path  $\gamma \in \mathcal{P}_{a,b}^d$  and  $1 \leq i, j \leq d$ .*

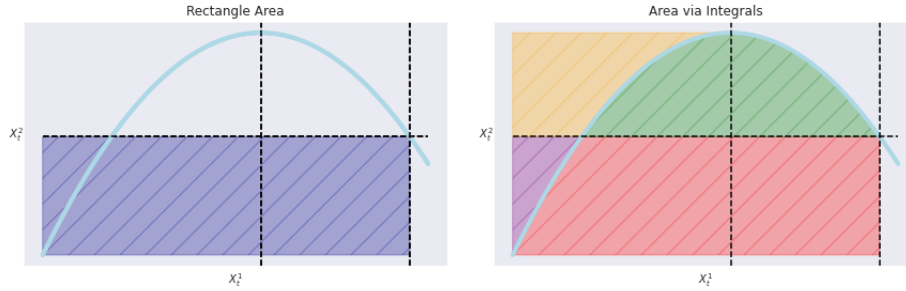


FIGURE 4. Inductive step for Lemma 3.5 - two methods to count the product of the displacements, depicted as the blue rectangle

A geometric interpretation of this result may be seen in Figure 4. The integral  $\mathcal{S}(\gamma)^{1,2}$  for example may be considered as the sum of the signed areas over each section where the curve is monotone. The statement then reads that the signed area of the blue rectangle in the left image may be constructed as follows: take the integral of  $\gamma^2 d\gamma^1$ , to contribute the positive green and red sections, then the integral of  $\gamma^1 d\gamma^2$  from zero until the maxima (with respect to the  $\gamma^1$  coordinate, the left-most black dotted line) contributes positive purple and yellow areas; the final integral from the maxima to the end point (second black dotted line) contributes negative green and negative yellow areas. The resulting sum of all these signed areas is the area of the blue rectangle.

**Remark 3.6.** Lemma 3.5 highlights that there is some redundancy in the vector representation of the signature which we have so far discussed. If the terms  $\mathcal{S}(\gamma)^i$ ,  $\mathcal{S}(\gamma)^j$  and  $\mathcal{S}(\gamma)^{i,j}$  are known, then  $\mathcal{S}(\gamma)^{j,i}$  may be inferred without an explicit record in the vector. In the same way, the terms corresponding to words in one letter have the representation of Proposition 3.4; we note that only one of these terms is required for each letter in order to compute all such terms.

Let us recall the notion of a Lie bracket operation  $[\cdot, \cdot]$  induced by  $\otimes$ :  $[x, y] := x \otimes y - y \otimes x$  for  $x, y \in \mathcal{V}$ . Combining (3.1) and (3.2) yields, for any path  $\gamma \in \mathcal{P}_{a,b}^d$ ,

$$(3.3) \quad \log \mathcal{S}(\gamma)_{a,b} = \sum_{k \geq 1} \sum_{i_1, \dots, i_k \in \{1, \dots, d\}} \lambda_{i_1, \dots, i_k} \left[ e_{i_1}, [e_{i_2}, \dots, [e_{i_{k-1}}, e_{i_k}] \dots] \right].$$

A vector representation of the logsignature therefore only requires entries corresponding to each of the basis elements of the form appearing in (3.3). This has considerably fewer terms (up to a given level) than the full signature in Definition 3.2. A five-dimensional path for example, up to the third level, has 155 terms in its signature and 55 in its logsignature (see [16] for formulae on the sizes of signatures). This makes the logsignature



particularly enticing from a computational point of view, not just for the reduction of required working memory, but also because convergence time of algorithms will benefit from the removal of redundancy in the feature set.

**3.2. Signature of data points.** Data, however, is not continuously observed and instead discrete observations  $\{(t_i, x_i)\}_{i=1, \dots, n}$  of values  $x_i$  at times  $t_i$  are more realistic. In order to associate a signature to this set of points, some interpolation is required. Among several approaches proposed in the literature, we present here the piecewise linear and the rectilinear interpolations used by Levin, Lyons and Ni [11].

**Definition 3.7** (Path interpolation). Let  $\mathcal{X} = \{(t_1, x_1), \dots, (t_n, x_n)\}$  be a set of observations.

- The piecewise linear interpolation  $X : [t_1, t_n] \rightarrow \mathbb{R}$  of  $\mathcal{X}$  is defined as

$$X(t) := \sum_{i=1}^{n-1} \left[ x_i + (x_{i+1} - x_i) \frac{t - t_i}{t_{i+1} - t_i} \right] \mathbf{1}_{\{t \in [t_i, t_{i+1})\}} + x_n \mathbf{1}_{\{t = t_n\}};$$

- The rectilinear interpolation path  $X' : [t_1, t_n] \rightarrow \mathbb{R}$  of  $\mathcal{X}$  is given by

$$X'(t) := \sum_{i=1}^{n-1} x_i \mathbf{1}_{\{t \in [t_i, t_{i+1})\}} + x_n \mathbf{1}_{\{t = t_n\}}.$$

Figure 5 shows the behaviour of these two interpolations over the set of points  $\{(0, 8), (2, 0), (3, 12), (6, 14)\}$ :

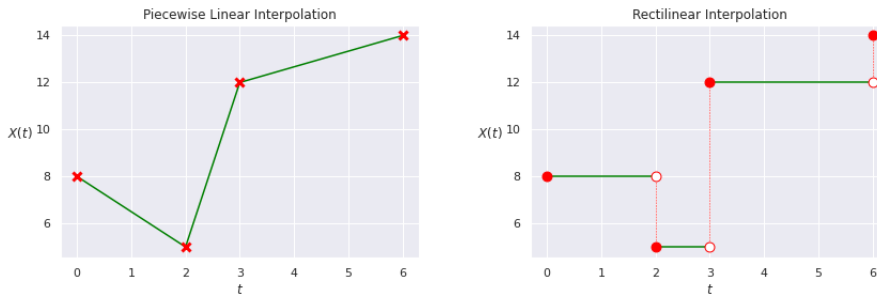


FIGURE 5. Piecewise linear and rectilinear interpolations

In Section 4, we will be using the piecewise linear interpolation of data points to form piecewise differentiable curves over which we may compute signatures. With this in place we may demonstrate how the AG-algorithm can be used to cluster market data time series.

#### 4. NUMERICAL ANALYSIS

We now work towards the clustering of market time series data. We start with simulated price paths following the Black-Scholes dynamics

$$(4.1) \quad S_t = S_0 \exp \left\{ \left( \mu - \frac{\sigma^2}{2} \right) t + \sigma W_t \right\},$$

where  $(W_t)_{t \geq 0}$  is a standard Brownian motion, and  $S_0 = 1$  so the paths represent returns. Here, a regime corresponds to a choice of  $(\mu, \sigma)$  within a possible set of parameters  $\mathfrak{M}$ . We demonstrate the clustering of Brownian paths by selecting four regimes according to the parameters in Table 1.

	Regime 1	Regime 2	Regime 3	Regime 4
$\mu$	5%	5%	2%	2%
$\sigma$	10%	20%	10%	20%

TABLE 1. Space  $\mathfrak{M}$  of parameters for the Brownian paths in Figure 6

**4.1. Regime points and point elements.** In this setting, we will understand a point as a collection of path signatures. The component paths are thought of as being returns of some collection of securities over some time horizon, whose prices evolve according to the dynamics of (4.1) with a common regime  $(\mu, \sigma)$  from Table 1. The paths are mapped to the interval  $[0, 1]$ . We then compute signatures, truncated to some level, and scale each of the level- $k$  terms by  $k!$ . By repeated application of Gronwall's Lemma [5, Lemma 3.2], the iterated integral of level  $k$  of a bounded variation path is equal to its 1-norm divided by  $k!$ , hence our scaling ensures that each level is comparable to the others. Algorithm 3 outlines the construction of the set of points  $\{\{\mathcal{X}_i^{\mu, \sigma}\}_{i=1, \dots, k}\}_{(\mu, \sigma) \in \mathfrak{M}}$ .

---

**Algorithm 3:** Generation of a regime-point

---

**Input:** Number of paths  $n$  to generate per point, signature truncation depth  $l$ , number of divisions  $m$  of the interval  $[0, 1]$ , parameters  $(\mu, \sigma) \in \mathfrak{M}$ .

**Output:** Single metric space point.

- (1) For each  $i \in \{1, \dots, n\}$ , simulate a path  $S_i = (S_i^1, \dots, S_i^m)$ , of length  $m$  according to (4.1);
  - (2) For each  $i$ , time-augment the path to obtain a two-dimensional path  $\{(1/m, S_i^1), \dots, (1, S_i^m)\}$ ;
  - (3) Construct the piecewise-differentiable function  $\tilde{S}_i : [0, 1] \rightarrow \mathbb{R}^2$  by linear interpolation;
  - (4) Let  $x_i = \mathcal{S}(\tilde{S}_i)_{\leq l}^1$  be the signature transform of the augmented path  $(\tilde{S}_i)$  up to level  $l$ ;
  - (5) Return  $\mathcal{X}^{\mu, \sigma} := \{x_1, \dots, x_n\}$
- 

**4.2. Distance between collections of signatures.** Consider the points  $\mathcal{X} = \mathcal{X}^{\mu, \sigma}$  and  $\tilde{\mathcal{X}} = \tilde{\mathcal{X}}^{\tilde{\mu}, \tilde{\sigma}}$ , generated according to Algorithm 3. We may think of the regime parameters  $(\mu, \sigma), (\tilde{\mu}, \tilde{\sigma})$ , as inducing distributions  $\mathbb{P}$  and  $\tilde{\mathbb{P}}$  of path signatures. The expected distance between the two collections may then be understood as a distance between these two distributions. Given a collection  $X = \{x_1, \dots, x_n\}$  of independent samples from a

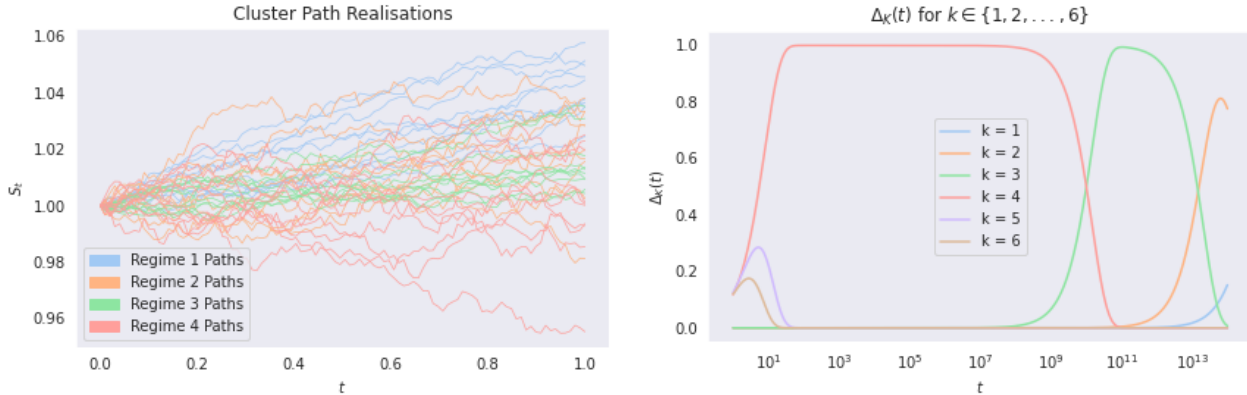


FIGURE 6. Synthetic paths - generated paths and eigengaps plot

population  $p$  and a collection  $Y = \{y_1, \dots, y_m\}$  of samples from a population  $\tilde{p}$ , the maximum mean discrepancy test [6] is a two-sample hypothesis test used to determine whether there is sufficient evidence at some significance level to reject the null hypothesis  $p = \tilde{p}$ . In order to define this statistic, we first recall the definition of a reproducing kernel Hilbert space (RKHS):

**Definition 4.1.** Let  $\mathfrak{X}$  be a set and  $\mathcal{H}$  a Hilbert space of functions from  $\mathfrak{X}$  to  $\mathbb{R}$  endowed with an inner product  $\langle \cdot, \cdot \rangle$ . For each  $x \in \mathfrak{X}$ , the evaluation functional  $\mathcal{L}_x : \mathcal{H} \rightarrow \mathbb{R}$  is defined by  $\mathcal{L}_x f := f(x)$ . We call  $\mathcal{H}$  a reproducing kernel Hilbert space (RKHS) if  $\mathcal{L}_x$  is continuous for every  $x \in \mathfrak{X}$ . In that case, for each  $x \in \mathfrak{X}$ , there exists  $K_x \in \mathcal{H}$  such that  $\mathcal{L}_x f = \langle f, K_x \rangle$ , for any  $f \in \mathcal{H}$  and the function  $k : \mathfrak{X} \times \mathfrak{X} \rightarrow \mathbb{R}$  such that  $k(x, y) := \langle K_x, K_y \rangle$  is called the reproducing kernel for  $\mathcal{H}$ . Finally,  $\mathcal{H}$  is said to be universal [17] if  $k(x, \cdot)$  is continuous for all  $x$  and  $\mathcal{H}$  is dense in  $\mathcal{C}(\mathfrak{X})$ , the space of continuous functions from  $\mathfrak{X}$  to  $\mathbb{R}$ .

We have the following formulation of the maximum mean discrepancy statistic [6]:

**Definition 4.2.** Let  $\mathcal{F}$  be a class of functions from  $\mathfrak{X}$  to  $\mathbb{R}$ , and let  $\mathbb{P}$  and  $\mathbb{Q}$  two distributions on  $\mathfrak{X}$ . The Maximum Mean Discrepancy of  $\mathbb{P}, \mathbb{Q}$  over  $\mathcal{F}$  is defined as

$$(4.2) \quad \mathfrak{D}^{\mathcal{F}}[\mathbb{P}, \mathbb{Q}] := \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{X \sim \mathbb{P}}[f(X)] - \mathbb{E}_{Y \sim \mathbb{Q}}[f(Y)] \right\}.$$

If, instead of observing  $\mathbb{P}$  and  $\mathbb{Q}$ , we have independent observations  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_n\}$  from  $\mathbb{P}$  and  $\mathbb{Q}$  respectively, then an empirical estimate of the  $\mathfrak{D}$  is given by

$$\mathfrak{D}^{\mathcal{F}}[X, Y] := \sup_{f \in \mathcal{F}} \left\{ \frac{1}{m} \sum_{i=1}^m f(x_i) - \frac{1}{n} \sum_{i=1}^n f(y_i) \right\}.$$

Let  $\mathcal{H}$  be a universal reproducing kernel Hilbert space with associated kernel  $k$  and  $\mathcal{F}$  the unit ball in  $\mathcal{H}$ . Then the empirical estimate can be computed in terms of the kernel  $k(\cdot, \cdot)$  as

$$(4.3) \quad \mathfrak{D}^{\mathcal{F}}[X, Y] = \left[ \frac{1}{m^2} \sum_{i_1, i_2=1}^m k(x_{i_1}, x_{i_2}) - \frac{2}{mn} \sum_{i=1}^m \sum_{j=1}^n k(x_i, y_j) + \frac{1}{n^2} \sum_{j_1, j_2=1}^n k(y_{j_1}, y_{j_2}) \right]^{1/2}.$$

We will use in particular the reproducing kernel Hilbert space induced by the Gaussian kernel (shown to be universal in [17])

$$k^{\sigma}(x, y) := \exp \left\{ -\frac{\|x - y\|^2}{2\sigma^2} \right\},$$

on compact subsets of  $\mathbb{R}^d$ , for fixed  $\sigma > 0$ . Note that the maximum mean discrepancy is not a distance function, since (4.2) depends on the order of  $\mathbb{P}$  and  $\mathbb{Q}$ . Nevertheless, if the induced Hilbert space  $\mathcal{H}$  is a universal RKHS, then  $\mathfrak{D}^{\mathcal{F}}[\mathbb{P}, \mathbb{Q}] = 0$  if and only if  $\mathbb{P} = \mathbb{Q}$ , as proved in [6]. Furthermore, the estimate (4.3) is symmetric. We show below that this choice of metric is sufficient to allow for the classification of regimes from Brownian paths.

**4.3. Results.** For each regime in Table 1, we generate 10 samples according to Algorithm 3, each consisting of 40 paths, each path coming from a uniform partition of  $[0, 1]$  with 100 time steps. A subset of the resulting paths (with 10 paths per regime) is shown in Figure 6 (left). For each point in the space, the signatures up to level 3 are computed. The second- and third-order terms are scaled by  $2!$  and  $3!$  respectively. The similarity function used was that of Equation (2.3). From this setup, we may proceed with a similar analysis as in the Gaussian Clouds examples. The maximal eigengap separation plot is presented in Figure 6 (right). As in the previous example, this structure is deemed successful at clustering the underlying points in the sense

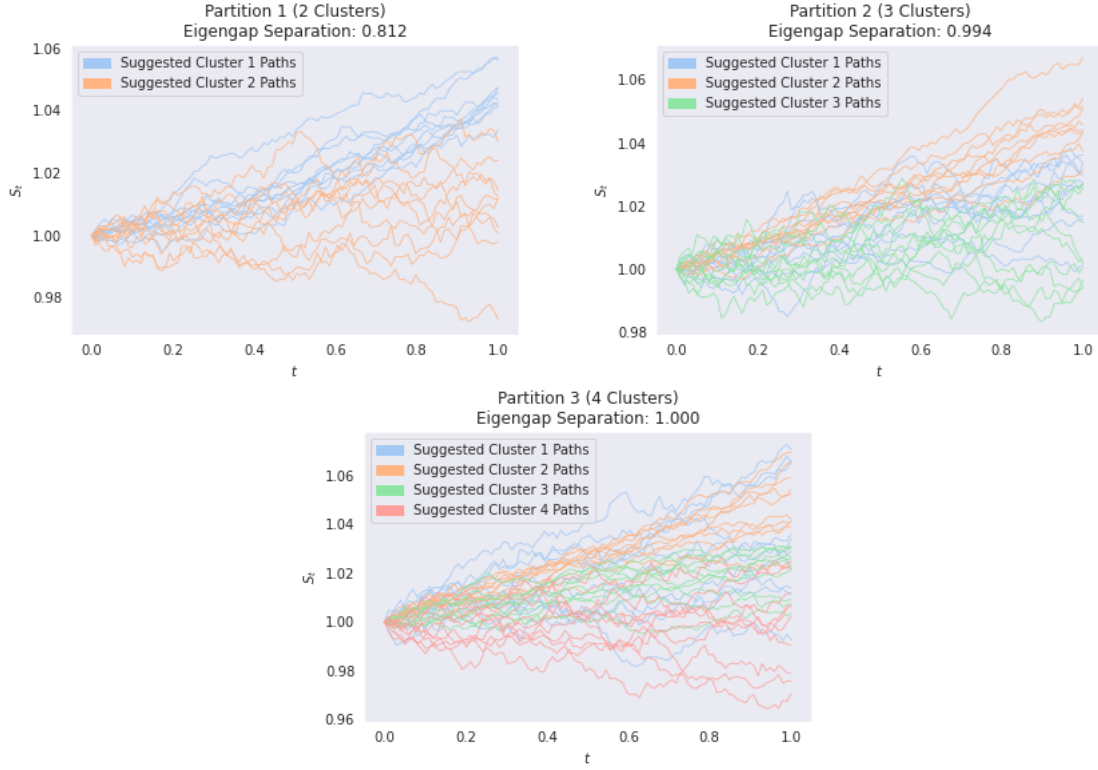


FIGURE 7. Synthetic paths example - suggested path clusterings

that the nontrivial clustering with the highest eigengap separation is the 4-clustering, presented in Figure 7. For comparison, the other suggested clusterings are also presented. Let  $\mathcal{R}_1, \dots, \mathcal{R}_4$  denote the point indices corresponding to the regimes of Table 1, with the same numberings. The clusters shown in Figure 7 are made precise in Table 2. In all of the suggested partitions, the clusters are preserved, with several clusters being combined to form larger suggested clusters in the  $k$ -clusterings with  $k < 4$ .

	Partition	Eigengap Separation
2 Clusters	$\mathcal{R}_1, \mathcal{R}_2 \cup \mathcal{R}_3 \cup \mathcal{R}_4$	0.8123
3 Clusters	$\mathcal{R}_1, \mathcal{R}_2 \cup \mathcal{R}_3, \mathcal{R}_4$	0.9941
4 Clusters	$\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$	1.000

TABLE 2. Synthetic data - suggested clusters

APPENDIX A. PROOF OF LEMMA 3.5

It suffices to prove the result for curves with  $\gamma_0^i = \gamma_0^j = 0$ . Indeed, then if  $\tilde{\gamma} \in \mathcal{P}_{a,b}^d$  is the translation of  $\gamma$  having  $\tilde{\gamma}_0^i = \tilde{\gamma}_0^j = 0$ , by Proposition 3.3 (translation invariance) we have

$$\mathcal{S}(\gamma)_{a,b}^{i,j} + \mathcal{S}(\gamma)_{a,b}^{j,i} = \mathcal{S}(\tilde{\gamma})_{a,b}^{i,j} + \mathcal{S}(\tilde{\gamma})_{a,b}^{j,i} = \mathcal{S}(\tilde{\gamma})_{a,b}^i \mathcal{S}(\tilde{\gamma})_{a,b}^j = \mathcal{S}(\gamma)_{a,b}^i \mathcal{S}(\gamma)_{a,b}^j.$$

So assume without loss of generality that  $\gamma_0^i = \gamma_0^j = 0$ . The result is clear when the function is monotone. We refer to Figure 8 below. The term on the right-hand side is the area of the bounding rectangle, and the term on the left is the sum of the shaded areas, which are the integrals in both directions.

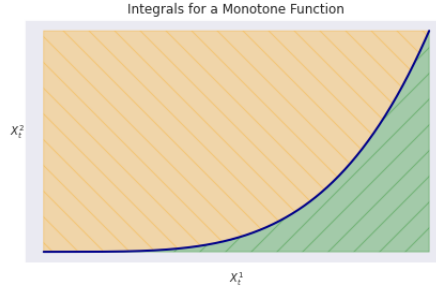


FIGURE 8. Monotone function step (base case) for Lemma 3.5

Next, suppose we can write  $\gamma$  as the concatenation of two paths,  $\gamma = \varphi * \phi$ , with  $\varphi \in \mathcal{P}_{a,t}^d$  and  $\phi \in \mathcal{P}_{t,b}^d$  both satisfying Lemma 3.5 (for example both are monotone) - we refer to Figure 4. Dropping the subscript  $a, b$  for brevity of notation, we may write

$$\mathcal{S}(\varphi)^i \mathcal{S}(\varphi)^j = \mathcal{S}(\varphi)^{i,j} + \mathcal{S}(\varphi)^{j,i} \quad \text{and} \quad \mathcal{S}(\phi)^i \mathcal{S}(\phi)^j = \mathcal{S}(\phi)^{i,j} + \mathcal{S}(\phi)^{j,i}.$$

A well-known result from Chen (see, for example, [4, Section 1.3.3]) establishes a connection between the signature of a concatenation of paths and the operation  $\otimes$  of Section 3.1.1, and states  $\mathcal{S}(\varphi * \phi) = \mathcal{S}(\varphi) \otimes \mathcal{S}(\phi)$ . Recall the non-commutative formal polynomial of  $\mathcal{S}(\varphi)$ :

$$\mathcal{S}(\varphi) = 1 + S^1(\varphi)e_1 + \dots + S^d(\varphi)e_d + S^{1,1}(\varphi)e_1e_1 + S^{1,2}(\varphi)e_1e_2 + \dots$$

and the similar representation for  $\mathcal{S}(\phi)$ . The coefficient of  $e_i$  in the product  $\mathcal{S}(\varphi) \otimes \mathcal{S}(\phi)$  is seen to be  $S^i(\varphi) + S^i(\phi)$ , that is  $\mathcal{S}(\gamma)^i = \mathcal{S}(\varphi * \phi)^i = \mathcal{S}(\varphi)^i + \mathcal{S}(\phi)^i$ . Note that the geometric interpretation here is simply that the displacement in the  $i^{\text{th}}$  coordinate path in the concatenation of  $\varphi$  and  $\phi$  is the sum of displacements in the path  $\varphi$  and  $\phi$ . We can also compute  $\mathcal{S}(\varphi * \phi)^{i,j}$  in a similar fashion, obtaining

$$\mathcal{S}(\varphi * \phi)^i = \mathcal{S}(\varphi)^i + \mathcal{S}(\phi)^i \quad \text{and} \quad \mathcal{S}(\varphi * \phi)^{i,j} = \mathcal{S}(\varphi)^i \mathcal{S}(\phi)^j + \mathcal{S}(\varphi)^{i,j} + \mathcal{S}(\phi)^{i,j},$$

from which we have

$$\begin{aligned} \mathcal{S}(\gamma)^{i,j} + \mathcal{S}(\gamma)^{j,i} &= \mathcal{S}(\varphi * \phi)^{i,j} + \mathcal{S}(\varphi * \phi)^{j,i} \\ &= \mathcal{S}(\varphi)^i \mathcal{S}(\phi)^j + \mathcal{S}(\varphi)^{i,j} + \mathcal{S}(\phi)^{i,j} + \mathcal{S}(\varphi)^j \mathcal{S}(\phi)^i + \mathcal{S}(\varphi)^{j,i} + \mathcal{S}(\phi)^{j,i} \\ &= \mathcal{S}(\varphi)^i \mathcal{S}(\phi)^j + \mathcal{S}(\varphi)^i \mathcal{S}(\phi)^j + \mathcal{S}(\varphi)^j \mathcal{S}(\phi)^i + \mathcal{S}(\phi)^i \mathcal{S}(\phi)^j \\ &= (\mathcal{S}(\varphi)^i + \mathcal{S}(\phi)^i)(\mathcal{S}(\varphi)^j + \mathcal{S}(\phi)^j) = \mathcal{S}(\varphi * \phi)^i \mathcal{S}(\varphi * \phi)^j = \mathcal{S}(\gamma)^i \mathcal{S}(\gamma)^j. \end{aligned}$$

Inductively, this proves the result for any curve composed of segments which are piecewise-monotone.  $\square$

#### REFERENCES

- [1] A. Azran and Z. Ghahramani. A new approach to data driven clustering. *Proceedings of the 23rd International Conference on Machine Learning*, 2006.
- [2] H. Boedihardjo, X. Geng, T. Lyons, and D. Yang. The signature of a rough path: uniqueness. *Advances in Mathematics*, 293:720–737, 2016.
- [3] K. T. Chen. Integration of paths - a faithful representation of paths by noncommutative formal power series. *Transactions of the American Mathematical Society*, 89(2):395–407, 1958.
- [4] I. Chevyrev and A. Kormilitzin. A primer on the signature method in machine learning. Unpublished, arXiv:1603.03788, 2016.
- [5] P. Friz and N. Victoir. *Multidimensional stochastic processes as rough paths*. Springer-Verlag, NY, 2010.

- [6] A. Gretton, K. M. Borgwardt, M. Rasche, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample problem. *Advances in NIPS*, 19, 2006.
- [7] B. Hambly and T. Lyons. Uniqueness for the signature of a path of bounded variation and the reduced path group. *Annals of Mathematics*, 171(1):109–167, 2010.
- [8] A. Jiltsov. Market regimes: how to spot them and how to trade them. *FX Markets*, 2020.
- [9] M. Jordan, A. Ng, and Y. Weiss. On spectral clustering: analysis and an algorithm. *Proceedings of the 14th NIPS International Conference*, pages 849–856, 2001.
- [10] M. Kritzman, S. Page, and D. Turkington. Regime shifts: Implications for dynamic strategies. *Financial Analysts Journal*, 68:22–39, 2012.
- [11] D. Levin, T. Lyons, and H. Ni. Learning from the past, predicting the statistic for the future, learning an evolving system. arXiv:1309.0260, 2013.
- [12] T. Lyons, M. Caruana, and T. Lévy. *Differential equations driven by rough paths*. Springer-Berlin Heidelberg, 2007.
- [13] T. Lyons and Z. Qian. *System control and rough paths*. Oxford University Press, 2002.
- [14] T. Lyons and W. Xu. Inverting the signature of a path. *Journal of the European Mathematical Society*, 20:1655–1687, 2018.
- [15] P. Nystrup, P. N. Kolm, and E. Lindström. Greedy online classification of persistent market states using realized intraday volatility features. *The Journal of Financial Data Science*, 2.3, 2020.
- [16] J. Reizenstein. Calculation of iterated-integral signatures and logsignatures. arXiv:1712.02757, 2017.
- [17] I. Steinward. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2001.

THE THALESIANS AND DEPARTMENT OF MATHEMATICS, IMPERIAL COLLEGE LONDON

*Email address:* paul@thalesians.com

DEPARTMENT OF MATHEMATICS, IMPERIAL COLLEGE LONDON AND THE ALAN TURING INSTITUTE

*Email address:* a.jacquier@imperial.ac.uk

DEPARTMENT OF MATHEMATICS, IMPERIAL COLLEGE LONDON

*Email address:* conormcindoe1@gmail.com