Laboratoire : Institut Elie Cartan de Lorraine
Ecole doctorale : IAEM Lorraine

THÈSE

présentée et soutenue publiquement
pour l'obtention du titre de

DOCTEUR DE L'UNIVERSITÉ DE LORRAINE

Mention Mathématiques

Par :
Raphaël MIGNOT

Sous la direction de :
Marianne CLAUSEL
Konstantin USEVICH

Multivariate time series analysis with the signature method

---

le 9 octobre 2024 à Nancy

Membres du jury :

Président	M. Xavier Pennec	Directeur de Recherche, INRIA, Sophia Antipolis
Rapporteurs	M. Nicolas Le Bihan	Directeur de Recherche, CNRS, Grenoble
	M. Badih Ghattas	Professeur, Université d'Aix–Marseille, Marseille
Examinateurs	Mme Agathe Guilloux	Directrice de Recherche, INRIA–INSERM, Paris
	Mme Emilie Devijver	Chargée de Recherche, CNRS, Grenoble
	M. Max Pfeffer	Junior Professor, Universität Göttingen, Göttingen
	M. El-Hadi Djermoune	Professeur, Université de Lorraine, Nancy
Invité	M. Georges Oppenheim	Professeur Emérite, Université Gustave Eiffel, Paris
Directeurs de thèse	Mme Marianne Clausel	Professeure, Université de Lorraine, Nancy
	M. Konstantin Usevich	Chargé de Recherche, CNRS, Nancy

Abstract

The analysis of sequential data, or time series, is key in numerous field of applications, e.g., engineering, sociology, medicine or econometrics. Often, linear models are not sufficient to account for the complex nature of data. This has created a need for interpretable and nonlinear methods for time series analysis. In this thesis, we analyze multidimensional time series through the lens of their integrals of various moment orders, constituting their signatures, a novel method for time series analysis. Under mild conditions, signatures characterize time series uniquely, up to time reparametrization and translation, into a set of features. Due to their ability to encode nonlinear dependencies in data, signature features have improve the current state-of-the-art in a broad range of Machine Learning applications, such as distribution regression, anomaly/novelty detection, human action recognition.

Signature features lie in a nonlinear space, making their manipulation challenging from a practical perspective. First, we introduce a method to average signature features which takes into account the geometry of the space, through a finite iterative algorithm. In addition, we present a strategy to effectively reduce the dimension of signature features by adapting the Principal Component Analysis (PCA). Our approaches rely on the algebraic manipulation of signatures and local approximations. We show that this dimension reduction method allows for stability of performances while using much fewer signature features. Then, we demonstrate how signatures can be highly effective as a multiscale tool for anomaly detection, with competitive runtimes. Finally, in the last chapter, we deal with clustering of time series under perturbations and introduce similarity measures in the space of signatures that we couple with usual distance-based clustering methods.

Keywords: Time series, Iterated Integrals Signatures, Learning on manifolds, Unsupervised learning.

Résumé

L'analyse de données séquentielles, ou séries temporelles, est essentielle dans de nombreux domaines d'application, tels que l'ingénierie, la sociologie, la santé ou l'économétrie. Souvent, les modèles linéaires ne suffisent pas à rendre compte de la nature complexe des données. Cela a créé un besoin de méthodes interprétables et non linéaires pour l'analyse des séries temporelles. Dans cette thèse, nous analysons les séries temporelles multidimensionnelles sous l'angle de leurs intégrales de différents ordres de moments, constituant leurs signatures, une nouvelle méthode d'analyse des séries temporelles. Sous des hypothèses non contraignantes, les signatures caractérisent les séries temporelles de manière unique, à reparamétrisation temporelle et translation près, en un ensemble de caractéristiques. En raison de leur capacité à encoder des dépendances non linéaires dans les données, les signatures ont dépassé les performances des meilleures méthodes sur un large éventail d'applications d'apprentissage automatique, telles que la régression de lois de probabilités, la détection d'anomalies, la reconnaissance d'actions humaines.

Les signatures sont des points sur un espace non linéaire, ce qui rend leur manipulation difficile d'un point de vue pratique. Tout d'abord, nous introduisons une méthode de calcul de moyennes de signatures qui tient compte de la géométrie de l'espace, par le biais d'un algorithme itératif fini. En outre, nous présentons une stratégie permettant de réduire efficacement la dimension des signatures en adaptant l'Analyse en Composantes Principales (ACP). Nos approches reposent sur la manipulation algébrique des signatures ainsi que sur des approximations locales. Nous montrons que cette méthode de réduction de dimension permet de stabiliser les performances tout en utilisant beaucoup moins de caractéristiques de signature. Ensuite, nous démontrons comment les signatures peuvent être très efficaces en tant qu'outil multi-échelle pour la détection d'anomalies, avec des temps d'exécution compétitifs. Enfin, dans le dernier chapitre, nous traitons du partitionnement de séries temporelles soumises à des perturbations et nous introduisons des mesures de similarités dans l'espace des signatures que nous combinons aux méthodes classiques de partitionnement.

Mots clés : Séries temporelles, Signatures, Apprentissage sur variétés, Apprentissage non supervisé.

Remerciements

Je tiens à exprimer ici ma sincère reconnaissance envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de cette thèse. Leur soutien, leurs conseils et leur encouragement ont été essentiels tout au long de ce parcours.

En particulier, je tiens à remercier mes directeurs de thèse Marianne et Konstantin. Forts de leur expérience, ils m'ont offert un soutien technique et m'ont orienter vers les bonnes directions tout au long du projet. De plus, ils ont su rendre cette aventure agréable sur le plan personnel.

Je remercie Nicolas Le Bihan et Badih Ghattas qui ont accepté de prendre le temps de rapporter ce manuscrit, ainsi que tous les examinateurs : Xavier Pennec, Agathe Guilloux, Emilie Devijver, Max Pfeffer, El-Hadi Djermoune et Georges Oppenheim.

D'autre part, je tiens à remercier tous les membres du projet international EDDA dans lequel s'inscrit ma thèse. En particulier, je suis reconnaissant envers Nozomi et Jonathan, qui m'ont chaleureusement accueilli dans leur laboratoire de Yokosuka au Japon, ainsi que toute la bienveillante équipe du Global Ocean Observation Research Center. J'y ai passé un été formidable et fait de belles rencontres. Jonathan, j'espère que tu peux enfin jouer à la Switch sans mukadé sur le tatami. Je tiens également à remercier Leonard et Joscha de Greifswald, avec qui il a été très agréable de collaborer et d'organiser notre mini-conférence en Allemagne. Je remercie Georges pour ses nombreuses interrogations sur mon travail, qui incite à examiner un problème sous toutes ses coutures. Merci par ailleurs à Laure, notamment pour le court séjour dans son labo à Toulouse en début de thèse. J'espère qu'on trouvera une date pour s'organiser ce fameux cassoulet.

Plus généralement, merci aux collègues de Lyon : Stéphane et Rémi, pour leur accueil et bien sûr, pour les moments de convivialité à Pau et en Allemagne. Je remercie Pierre de l'Université du Luxembourg, qui m'a invité et surtout qui a tenté de m'initier à la géométrie sous-Riemannienne. Merci également à Elina pour son accueil chaleureux à Vancouver. Je suis reconnaissant envers Yannick Deleuze, qui m'a encadré en stage chez Veolia, et Michel Broniatowski, mon directeur d'études à Sorbonne Université, qui m'ont tous les deux inspiré et donné l'envie de me lancer dans une thèse.

A Nancy, je remercie tous les doctorants de l'Institut Elie Cartan, sans qui la thèse aurait été beaucoup plus rude. Mention spéciale à ma co-bureau Anouk, avec qui j'ai pu partager les moments de la vie de doctorant ainsi que les conférences (et notamment un accrobranche caniculaire à Fourvière). Merci à Pierre, Thomas, Christophe, Benjamin, Valentin, Victor, Pierrick, Nathan, Yann, David, Clara, Rodolphe. Merci pour toutes ces discussions, ces soirées jeux, ces parties de tennis, ces sorties. J'ai pu y créer des amitiés durables. A Nancy toujours, merci à l'équipe Simul du CRAN, très attentionnée envers mon travail. Je suis reconnaissant envers tout le personnel de l'IECL, en particulier Nathalie et Laurence pour leur assistance dans le labyrinthe administratif.

Enfin, je remercie tous mes proches : ma famille et mes amis, qui ont joué un rôle prépondérant dans cette aventure.

Contents

Résumé détaillé en français

Dans de nombreuses applications et des contextes variés, des données sont observées au cours du temps (Figure 1). Ces données se présentent sous la forme d'une série temporelle, contenant des observations échantillonnées à des instants successifs. Plusieurs quantités (par exemple, à la fois une pression et une température) peuvent être mesurées simultanément, ce qui produit des séries temporelles multivariées. Récemment, le terme plus général de « flux de données » a été proposé, car les données peuvent se présenter sous diverses formes : à valeur dans $\mathbb{R}$, comme un relevé de pression ; à valeurs discrètes, comme le nombre d'occurrences d'un événement ; ou encore textuel, comme des rapports d'examens médicaux. Ces données peuvent être imparfaites, car elles peuvent être échantillonnées de manière irrégulière. Par exemple, les examens médicaux d'un patient sont généralement effectués après le déclenchement d'un événement, tel que des effets secondaires. Les données peuvent également comporter des valeurs manquantes, qui peuvent survenir à la suite d'une défaillance d'un capteur, ou être censurées, par exemple lorsqu'un patient d'une étude clinique déménage à l'étranger.

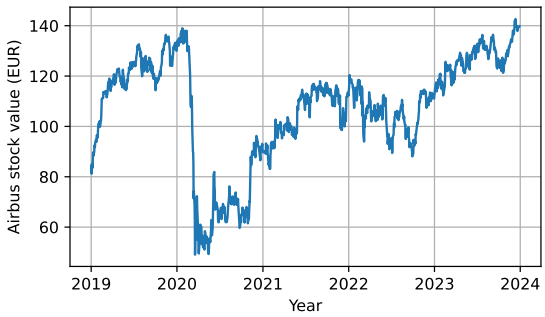


FIGURE 1: Un exemple de série temporelle, l'évolution de la valeur de l'action de l'entreprise Airbus sur les cinq dernières années.

L'analyse de ces flux de données est devenue essentielle dans divers domaines tels que l'ingénierie, la sociologie, la médecine et l'économétrie. Plusieurs tâches peuvent se présenter, notamment la prévision, la modélisation, la détection d'anomalies, le partitionnement, la classification et l'inférence causale. Pour traiter ces tâches, la boîte à outils actuelle contient un large éventail de méthodes, telles que l'analyse de l'autocorrélation, les modèles de régression des valeurs du temps présent par rapport aux valeurs à des temps passés (famille des modèles autorégressifs : ARMA, ARIMA, etc.), les fonctions qui extraient les composantes en fréquence (transformée de Fourier), les composantes en temps–fréquence (transformée en ondelettes), et les méthodes de décomposition, avec notamment la décomposition Saisonnalité–Tendance par LOESS, la décomposition en modes empiriques, et l'analyse spectrale singulière. Les architectures d'apprentissage profond telles que les réseaux neuronaux récurrents font également partie de la boîte à outils. Nous pourrions également distinguer les stratégies linéaires des stratégies non linéaires, les méthodes univariées des méthodes multivariées. Une introduction aux méthodes classiques

d'analyse de séries temporelles est fournie dans [BD16], et le cas multidimensionnel est présenté dans [Lüt05].

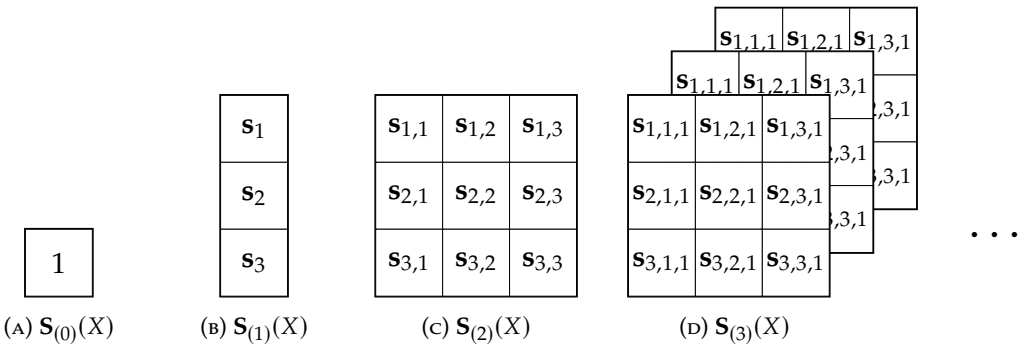
L'objectif de cette thèse est d'étudier la nouvelle méthode des signatures pour l'analyse des séries temporelles. D'une manière générale, les signatures peuvent être considérées comme des caractéristiques non linéaires décrivant une trajectoire continue en fournissant des caractéristiques invariantes à la translation et au rééchantillonnage. Souvent, le taux d'échantillonnage auquel le flux est enregistré n'est pas utile. Par exemple, pour la reconnaissance de l'écriture manuscrite, la vitesse à laquelle un caractère est écrit n'est pas essentielle. De même, la position du flux peut ne pas être importante (mais la rotation peut l'être, par exemple pour distinguer un « 6 » d'un « 9 »). La signature conserve l'information essentielle, à savoir l'ordre dans lequel les événements ont eu lieu et oublie la paramétrisation dans le temps.

La signature est apparue pour la première fois dans [Che57] et constitue l'un des outils centraux de la théorie des trajectoires rugueuses [Lyo98]. Récemment, de nombreux travaux ont suggéré de l'utiliser pour des tâches d'apprentissage automatique [CK16]. Plus de détails sont donnés dans la suite et dans un aperçu récent [LM24]. Voir également la thèse récente [Fer21].

L'intérêt particulier de cette thèse, par rapport à d'autres travaux sur le sujet, est que nous exploitons la structure géométrique différentielle et la structure de groupe de Lie de l'espace des signatures, pour des tâches d'apprentissage automatique. De manière générale, notre travail est ancré dans le domaine de l'apprentissage automatique (analyse des séries temporelles) et implique des éléments de géométrie différentielle, d'algèbre et d'analyse tensorielle.

Dans la suite, nous décrivons brièvement la transformation de la signature et donnons un aperçu des contributions.

Définition en termes simples. La méthode de la signature peut être considérée comme une version temporelle de la méthode des moments pour les variables aléatoires. En substance, la signature est une fonction $\mathbf{S}$ qui prend en entrée une série temporelle multivariée $X(t) = (X^1(t), \dots, X^d(t))$ et produit une collection de tenseurs $\mathbf{S}(X) = \{\mathbf{S}_{(0)}(X), \mathbf{S}_{(1)}(X), \dots\}$, comme en Figure 2, qui codent les dépendances d'ordre élevé entre les composantes X^i, pour $i = 1, \dots, d$.



(A) $\mathbf{S}_{(0)}(X)$ (B) $\mathbf{S}_{(1)}(X)$ (C) $\mathbf{S}_{(2)}(X)$ (D) $\mathbf{S}_{(3)}(X)$

FIGURE 2: La signature $\mathbf{S}(X)$ est une collection de tenseurs de tailles croissantes ; la signature de niveau L est un tenseur à L directions de dimension d. Nous utilisons la notation $\mathbf{s}_{i_1,\dots,i_L} := \left[\mathbf{S}_{(L)}(X)\right]_{i_1,\dots,i_L}$ avec X de dimension $d = 3$.

Une interprétation des tenseurs de signature $\mathbf{S}_{(L)}(X)$ peut être donnée pour les premiers niveaux. Par exemple, $\mathbf{S}_{(1)}(X)$ (niveau $L = 1$) est le changement global

$X(T) - X(0)$. Une combinaison linéaire des coefficients du tenseur $\mathbf{S}_{(2)}(X)$ (niveau $L = 2$) donne l'aire comprise entre la courbe et sa corde, comme le montre la Figure 3. Elle est également liée à la corrélation croisée [DR19]. En effet, si nous supposons que $X(0) = 0$, alors

$$[\mathbf{S}_{(2)}(X)]_{1,2} - [\mathbf{S}_{(2)}(X)]_{2,1} = \mathrm{Corr}(X^2, X^1)_1 - \mathrm{Corr}(X^1, X^2)_1 \tag{1}$$

où $\mathrm{Corr}(x, y)_1 := \sum_{t=0}^{T} x(t+1)y(t)$ est la corrélation croisée à 1-décalage, pour toute série temporelle x, y de longueur T. Il convient de noter que l'interprétation des signatures de niveau supérieur (niveaux $L \geq 3$) est moins évidente. Une interprétation en termes de géométrie sous-Riemannienne peut être trouvée dans [LCL07, p. 38].

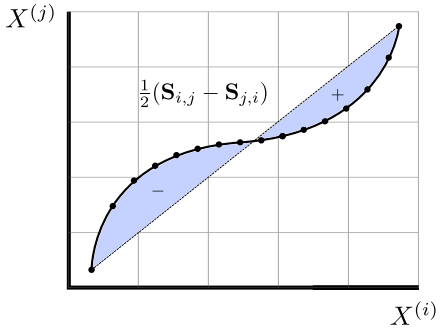


FIGURE 3: Interprétation géométrique de la signature de niveau $L = 2$.
La combinaison linéaire indiquée donne l'aire en bleu.

Une définition formelle de la signature sera donnée dans le Chapitre 2 ainsi que les principales propriétés. Un défi majeur réside dans le fait que les signatures appartiennent à une variété différentielle (en d'autres termes, un espace qui peut être localement bien approximé par un espace Euclidien) qui peut être équipé d'une structure de groupe ; cette structure est appelée un groupe de Lie. Par conséquent, la manipulation des signatures nécessite l'utilisation d'outils de géométrie différentielle, notamment pour adapter des méthodes initialement conçues pour les espaces Euclidiens. Il est intéressant de noter que dans cet espace, les opérations de groupe peuvent être liées à des opérations sur les séries temporelles, comme nous le verrons dans le Chapitre 2. En outre, cet espace possède un logarithme défini partout, le logarithme étant une fonction qui permet de naviguer entre l'espace des signatures et l'espace tangent. Pour avoir une vue d'ensemble de la théorie des signatures, nous nous référons à [CK16], [LCL07, Chapitre 2] et [FV10, Chapitre 7]. Pour plus de détails sur le cadre algébrique, voir [Reu93].

En raison de leur capacité à coder les dépendances non linéaires dans les séries temporelles multivariées, les caractéristiques de signature ont été utilisées avec succès pour aborder de nombreuses tâches d'apprentissage automatique au cours de la dernière décennie, par exemple, la classification [Mor+20; PA+18; GLM19; Mor+21], la détection d'anomalies [Arr+24; Sha+20], la prédiction [PA+18], la régression de lois de probabilités [Lem+21], la reconnaissance de mouvements et de formes [LZJ17; Yan+22; Gra13] (ce dernier est le vainqueur du défi ICDAR 2013 sur la reconnaissance des caractères chinois en ligne). Pour une étude exhaustive de la méthode de signature pour l'analyse des séries temporelles, nous nous référons à [LM24].

Dans la suite de ce résumé, nous introduisons brièvement les contributions principales présentées dans cette thèse.

Moyennes de signatures (Chapitre 3). Il est essentiel de pouvoir calculer la moyenne de données, c'est-à-dire de voir les choses à une échelle plus grossière, et ce pour de multiples raisons. Tout d'abord, le calcul de la moyenne résume ou simplifie l'information contenue dans un ensemble de points en une seule valeur. Cette valeur fournit une description pertinente des données avec un résumé global et, si nous cherchons à regrouper des points proches les uns des autres, les moyennes locales peuvent être utilisées pour classer les données (ce qui est l'idée centrale des stratégies d'apprentissage automatique telles que les K-moyennes). En d'autres termes, nous pouvons mettre en évidence des relations entre des ensembles de données en comparant les moyennes. En outre, il est possible d'obtenir des informations sur le temps, car les moyennes peuvent être calculées sur des périodes fixes et comparées (par exemple, en calculant une valeur moyenne pour chaque mois). Cette méthode peut être utilisée pour diverses tâches telles que le suivi de l'évolution de tendances, la détection des cycles, la surveillance (par exemple, le contrôle qualité pour procédés de fabrication), les prévisions et la réduction du bruit.

Dans le Chapitre 3, nous considérons la tâche consistant à calculer la moyenne d'un ensemble de N points $\mathbf{x}_1, \dots, \mathbf{x}_N$ situés dans l'espace des signatures G. Comme nous le verrons plus en détails dans le Chapitre 2, l'espace des signatures est une variété différentielle. Sur un tel espace *courbe*, le calcul de la statistique la plus simple, la moyenne, peut demander plus de travail que sur un espace Euclidien. En effet, la définition du barycentre pour un espace euclidien $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ ne peut pas être utilisée sur une variété, car dans de nombreux cas $\bar{x}$ peut ne pas appartenir à la variété. Prenons par exemple deux points du cercle unitaire : $x_1 = (1, 0)$ et $x_2 = (0, 1)$. Le barycentre Euclidien des deux points $\bar{x} = (1/2, 1/2)$ n'appartient pas au cercle.

Une généralisation du barycentre Euclidien aux variétés est la moyenne de Fréchet : soit $(\mathcal{M}, d)$ un espace métrique. Étant donné un ensemble de points $x_1, \dots, x_N \in \mathcal{M}$, la moyenne de Fréchet est le point $\mu \in \mathcal{M}$ tel que

$$\mu = \arg\min_{\mu \in \mathcal{M}} \sum_{i=1}^{N} d^2(\mu, x_i). \tag{2}$$

Cette définition peut être utilisée pour les groupes de Lie. De plus, si $d(.,.)$ est une métrique Riemannienne bi-invariante, alors μ est stable par les opérations de groupe : multiplication à gauche et à droite, inversion. Par exemple, la stabilité pour la multiplication à droite signifie que μy est la moyenne de Fréchet de $\{x_i y\}_{i=1,\dots,N}$. Cependant, si $d(.,.)$ n'est pas bi-invariant, la stabilité de μ n'est pas assurée. Pour de tels cas, les auteurs de [PL20] ont défini une notion de barycentre sur les groupes de Lie appelée moyenne de groupe, définie comme suit : pour un ensemble de N points $\mathbf{x}_1, \dots, \mathbf{x}_N$, la moyenne de groupe $\boldsymbol{\mu}$ est la solution de l'équation suivante :

$$\frac{1}{N}\sum_{i=1}^{N} \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i) = 0\ , \tag{3}$$

c'est-à-dire que les vecteurs $\mathbf{v}_i$ dans l'espace tangent à l'identité $T_{\mathbf{1}}G$ ont une moyenne nulle, où $\mathbf{v}_i := \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i)$, comme le montre la Figure 4. Il convient de noter que la moyenne de groupe peut ne pas exister dans le cas où le logarithme n'est pas défini partout.

Dans le Chapitre 3, les principales contributions sont :

- Dans la Section 3.3, nous prouvons l'existence et l'unicité de la moyenne de groupe pour les signatures.

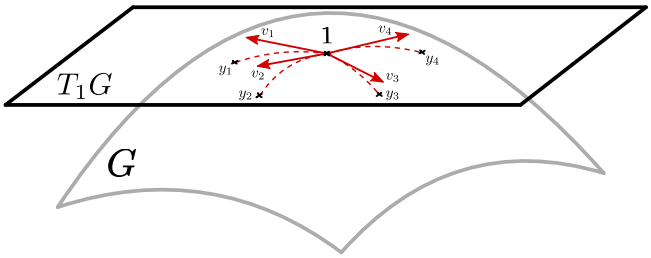


FIGURE 4: La moyenne de groupe μ sur un groupe de Lie G est telle que la somme des $\mathbf{v}_i$ vaut zéro, avec $\mathbf{v}_i := \log(\mathbf{y}_i) = \log(\mu^{-1}\mathbf{x}_i)$ vecteurs de l'espace tangent T_1G. Les lignes en pointillés sont les géodésiques sur G qui partent de l'origine avec vélocité $\mathbf{v}_i$.

- Dans la Section 3.4, nous fournissons un algorithme en temps fini pour calculer exactement le barycentre d'un ensemble fini de signatures et une implémentation en `Python`, qui repose essentiellement sur les propriétés algébriques de la signature.

Réduction de dimension de signatures (Chapitre 4). Étant donné une série temporelle à d dimensions $X = (X^1, \dots, X^d)$, sa signature jusqu'au niveau L, $\mathbf{S}_{(\leq L)}(X)$, a une dimension de $\frac{d(d^L-1)}{d-1}$, comme nous le verrons dans le Chapitre 2. En d'autres termes, le nombre de caractéristiques de la signature croît de manière exponentielle en fonction du niveau de troncature L. D'un point de vue numérique, l'utilisation d'un tel nombre de caractéristiques peut entraîner des ralentissements de calculs, en particulier si d est grand et s'il n'existe pas de méthodes efficaces pour effectuer la tâche en aval, par exemple du partitionnement. Notre objectif est de développer une méthode de réduction de dimension fidèle pour les signatures. Pour ce faire, nous généraliserons la célèbre Analyse en Composantes Principales (ACP), présentée ci-dessous, à l'espace des signatures.

Étant donné des échantillons $x_1, \dots, x_N$ dans $\mathbb{R}^d$, l'ACP fournit une suite de meilleures approximations linéaires des données, pour tous les rangs $K \leq d$. Dénotons $y_i := x_i - \mu$ les points après recentrage, où μ est la moyenne Euclidienne de l'ensemble $x_1, \dots, x_N$. Nous calculons une suite de vecteurs $v_1, \dots, v_K$ en résolvant successivement, pour tout $k = 1, \dots, K$,

$$v_k = \underset{\substack{\|v\|=1 \\ v \perp v_1, \dots, v_{k-1}}}{\arg\min} \sum_{i=1}^{N} \|y_i - \pi_v(y_i)\|^2 \tag{4}$$

où π_v est la projection orthogonale sur $\mathrm{vect}(v)$. Les $(v_k)_k$ sont appelés directions principales. Nous pouvons compresser les données en fixant $K < d$ et en les projetant sur $\mathrm{vect}(v_1, \dots, v_K)$. Le problème de minimisation (4) peut être résolu à l'aide de la décomposition en valeurs singulières.

Il existe de nombreuses généralisations de l'ACP à des ensembles de formes. Nous nous concentrons sur l'approche de l'Analyse en Géodésiques Principales (AGP) [FLJ03], qui est formulée comme suit. Soit $x_1, \dots, x_N$ des points sur une variété $\mathcal{M}$

avec une moyenne de groupe $\boldsymbol{\mu}$. L'AGP minimise

$$\underset{v \in T_\mu(\mathcal{M}), t_k \in \mathbb{R}}{\arg\min} \sum_{i=1}^{N} d(\gamma_v(t_k), x_k) \tag{5}$$

où γ_v est la géodésique qui part de μ.

Cependant, il n'est pas simple d'utiliser cette définition de l'AGP pour les signatures. La raison principale est l'absence d'une méthode canonique pour définir une métrique Riemannienne (bi-invariante).

Dans le Chapitre 4, nous proposons les solutions suivantes :

- Nous proposons une nouvelle version de l'AGP qui utilise un type spécial de divergence adapté à l'espace des signatures.
- Nous présentons deux algorithmes : un qui approxime sur l'espace tangent et un qui résout le problème d'optimisation sur la variété, ainsi que des implémentations en `Python`.
- Nous appliquons les deux méthodes de réduction de dimension pour des tâches classiques sur des données synthétiques et réelles et montrons que les performances restent élevées après réduction de dimension.

Détection d'anomalies avec signatures multi-échelles (Chapitre 5). En détection d'anomalies (DA) pour séries temporelles, l'objectif est de détecter un X_i dans un ensemble de séries temporelles $X_1, \ldots, X_N$ qui se comporte *anormalement*, c'est-à-dire qui prend des valeurs différentes du reste des données. Par exemple, dans le contexte de la surveillance maritime, il existe des voies principales de navigation et un bateau empruntant une trajectoire différente pourrait être considéré comme une anomalie.

Dans ce contexte de DA pour les séries temporelles multivariées, nous montrons qu'une analyse multi-échelle utilisant les signatures permet d'obtenir des résultats compétitifs. Outre l'amélioration des performances de détection, nous mettons en évidence l'efficacité numérique de l'analyse multi-échelle basée sur la signature. En particulier, nous évitons l'énorme charge de calcul qui apparaîtrait avec d'autres mesures de similarité. La méthode de signature multi-échelle peut être 100 fois plus rapide que le *Dynamic Time Warping* (DTW) à échelle unique. Cela est dû à la combinaison des propriétés de l'espace des signature avec l'utilisation d'interpolations linéaires sur les séries temporelles. En effet, comme indiqué plus loin dans la Section 2.3.4, la signature d'une trajectoire sur un segment entier est généralement calculée en combinant les signatures de sous-segments plus petits. Le stockage de ces sous-signatures et leur utilisation pour la DA ont le même coût de calcul que l'utilisation de la signature de toute la série temporelle. Ainsi, dans la segmentation dyadique présentée dans la Figure 5, nous stockerions la signature de chaque sous-segment (qui sont de longueurs 2, 4 et 8). Notre méthode de DA avec la signature repose sur ce résultat, qui nous permet d'augmenter la performance de détection des signatures grâce à l'opération multi-échelle, tout en conservant de faibles temps de calcul. Notre méthode est comparée à des stratégies classiques telles que le *Local Outlier Factor* (LOF) avec des mesures de similarité Euclidienne / DTW.

Partitionnement avec la signature (Chapitre 6). L'objectif du partitionnement de séries temporelles est de détecter des groupes de trajectoires *similaires* sans disposer

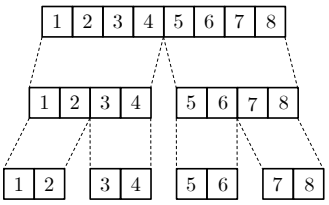


FIGURE 5: Segmentation dyadique d'une série temporelle de longueur 8.

d'informations préalables sur les données (apprentissage non supervisé). Cela nous permet de détecter des structures sous-jacentes cachées dans les données. Prenons l'exemple de l'évolution du prix des matières premières (pétrole, blé, sable, etc.). Une analyse de partitionnement pourrait permettre de détecter des relations entre deux matières premières, ce qui fournirait des informations précieuses, par exemple, pour la gestion de la chaîne d'approvisionnement ou pour les fournisseurs d'énergie cherchant à anticiper la demande.

Dans ce travail, nous montrons comment les signatures peuvent être utilisées pour regrouper des séries temporelles. Tout d'abord, nous étudions les mesures de similarité qui peuvent être utilisées dans ce contexte et nous introduisons plusieurs mesures de similarité qui sont invariantes par rapport à des transformations basiques des données. Nous comparons ces mesures de similarité aux distances usuelles sur les séries temporelles : Euclidienne, DTW, corrélation. Des jeux de données réelles de formes diverses sont analysés, par exemple, période temporelle d'observation courte/longue, petit/grand nombre de dimensions.

Ensuite, nous montrons comment la notion de barycentre des signatures (Chapitre 3) peut être utilisée pour étendre les méthodes d'apprentissage automatique. Par exemple, la méthode de partitionnement des K-moyennes peut être utilisée avec les signatures, en utilisant la notion de moyenne de groupe.

Nous introduisons des perturbations dans les données (vacillement, bruitage) et comparons le comportement des similarités sur les signatures par rapport aux similarités usuelles. Les signatures fournissent des performances fluctuantes tout en ayant des temps d'exécution très compétitifs, en particulier pour les séries temporelles observées sur de longues périodes.

Chapter 1

General introduction

1.1 Time series analysis

In many applications and contexts, data is observed over time (Figure 1.1). Such data come in a form of a time series, containing observations sampled at successive time instants. Several quantities (e.g., pressure and temperature) can be measured at the same time, leading to multivariate time series. Recently, the more general term "data stream" was proposed, as data can take many forms: real-valued, such as a pressure; discrete, such as the number of occurrences of an event; textual, including medical records. This data can be imperfect, as it may be sampled irregularly. For instance, medical examinations of a patient are usually conducted after the triggering of an event, such as side effects. Data can also include missing values, which might occur following a sensor failure, or be censored, such as when a patient in a panel moves abroad.

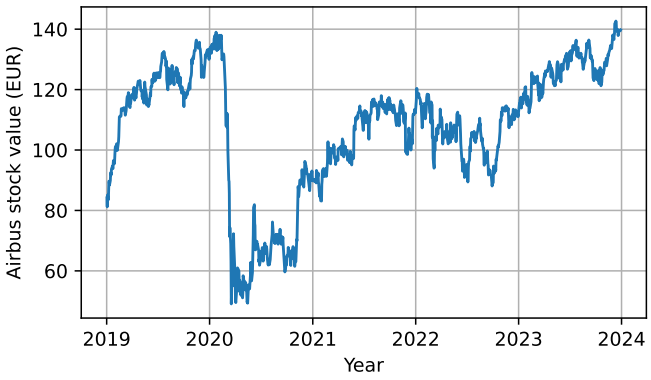


FIGURE 1.1: An example of time series, the evolution of the stock value of company Airbus over the last five years.

The analysis of such data streams has become key in diverse fields such as engineering, sociology, medicine, and econometrics. Several tasks may arise, including forecasting, modeling, anomaly detection, clustering, classification and causal inference. To address those tasks, the current toolbox includes a wide range of methods, such as autocorrelation analysis, regression models of present values against past values (with the autoregressive family of models—ARMA, ARIMA, etc.), mappings that extract frequency components (Fourier transform), time–frequency components (wavelet transform), and decomposition methods, including Seasonal–Trend Decomposition using LOESS, Empirical Mode Decomposition, and Singular Spectrum Analysis. Deep Learning architectures such as Recurrent Neural Networks are also part of the toolkit. Alternatively, we could distinguish linear from nonlinear strategies, univariate from multivariate methods. An introduction to classical time series

analysis methods is provided in [BD16], and the multidimensional case is presented in [Lüt05].

The objective of the thesis is to study the novel method of signatures for time series analysis. Broadly speaking, signatures can be seen as nonlinear features describing a continuous trajectory by providing features invariant to translation and resampling. Often, the sampling rate at which the stream is recorded is not useful. For instance, for handwriting recognition the speed at which a character is written is not essential. Similarly, the position of the stream might not be important (but the rotation might, e.g., to discriminate a "6" from a "9"). The signature keeps the key information, which is the order in which events took place and forget the parametrization in time.

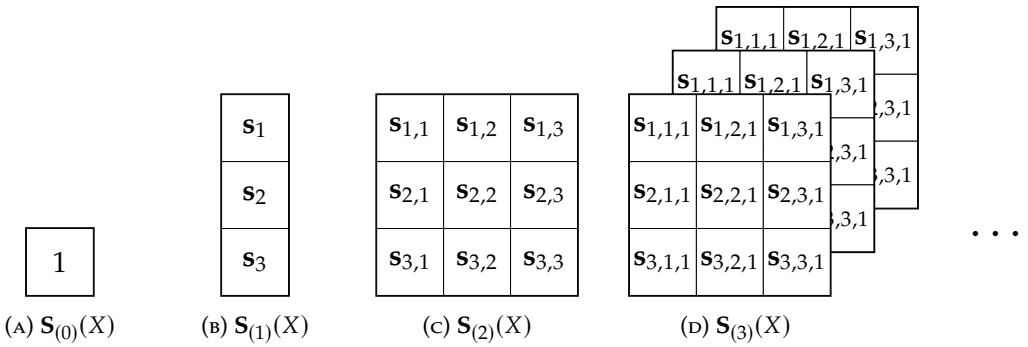
Signatures originated in [Che57] and is one of the central tool in the theory of rough paths [Lyo98]. Recently, numerous works have suggested to use it for machine learning tasks [CK16]. See more details in the next subsection, and in a recent overview [LM24]. See also the recent thesis [Fer21].

The particular focus of this thesis, compared to other works on the topic, is that we exploit the differential geometric and Lie group structure of the signature space, for machine learning tasks. Overall, our work is anchored in the field of machine learning (time series analysis) and involves elements of differential geometry, algebra, and tensor analysis.

In the next subsection, we briefly describe the signature transform, and give an overview of contributions.

1.2 Definition in plain words

The signature method can be understood as a time series version of the method of moments for random variables. Loosely speaking, the signature is a mapping $\mathbf{S}$ that takes as input a multivariate time series $X(t) = (X^1(t), \dots, X^d(t))$ and outputs a collection of tensors $\mathbf{S}(X) = \{\mathbf{S}_{(0)}(X), \mathbf{S}_{(1)}(X), \dots\}$, as in Figure 1.2, which encode high order dependencies between the components X^i, $i = 1, \dots, d$.



(A) $\mathbf{S}_{(0)}(X)$ (B) $\mathbf{S}_{(1)}(X)$ (C) $\mathbf{S}_{(2)}(X)$ (D) $\mathbf{S}_{(3)}(X)$

FIGURE 1.2: The signature $\mathbf{S}(X)$ is a collection of tensors of increasing sizes—the signature at level L is an L-ways tensor of dimension d. Here we denote $\mathbf{s}_{i_1,\dots,i_L} := \left[\mathbf{S}_{(L)}(X)\right]_{i_1,\dots,i_L}$ with X of dimension $d = 3$.

An interpretation of signature tensors $\mathbf{S}_{(L)}(X)$ can be given for the first levels. For instance, $\mathbf{S}_{(1)}(X)$ (level $L = 1$) is the global change $X(T) - X(0)$. A linear combination of the coefficients of tensor $\mathbf{S}_{(2)}(X)$ (level $L = 2$) gives the enclosed area between the curve and its chord, as shown in Figure 1.3. It is also related to the cross-correlation [DR19]. Indeed, if we assume that $X(0) = 0$, then

$$[\mathbf{S}_{(2)}(X)]_{1,2} - [\mathbf{S}_{(2)}(X)]_{2,1} = \mathrm{Corr}(X^2, X^1)_1 - \mathrm{Corr}(X^1, X^2)_1 \tag{1.1}$$

where $\mathrm{Corr}(x, y)_1 := \sum_{t=0}^{T} x(t+1)y(t)$ is the lag-one cross correlation for any time series x, y of length T. Note that the interpretation of higher level signatures (levels $L \geq 3$) is less straightforward. An interpretation in terms of sub-Riemannian geometry can be found in [LCL07, p. 38].

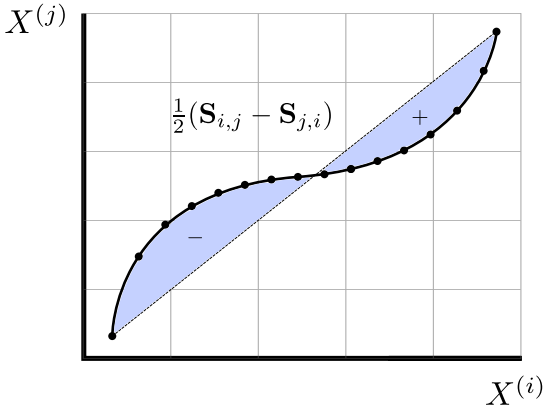


FIGURE 1.3: Geometrical interpretation of the signature at level $L = 2$. The linear combination gives the signed blue shaded area.

A formal definition of the signature will be given in Chapter 2 along with the main properties. A primary challenge lies in the fact that signatures belong to a differential manifold (in other words, a space that can be locally well approximated by a Euclidean space) which can be equipped with a group structure—this structure is called a Lie group. Therefore, manipulating signatures requires the use of differential geometry tools, especially for adapting methods initially designed for Euclidean spaces. Interestingly, in this space group operations can be related to operations on time series as we will see in Chapter 2. In addition, it has a globally defined logarithm, which is a function that allows to map elements of the signature space to elements of its tangent space. To have a comprehensive overview of the signature theory, we refer to [CK16], [LCL07, Chapter 2] and [FV10, Chapter 7]. For details regarding the algebraic setting, see [Reu93].

Due to their ability to encode non linear dependencies in multivariate time series, signature features have been successfully used to address numerous Machine Learning tasks in the last decade, e.g., classification [Mor+20; PA+18; GLM19; Mor+21], anomaly/novelty detection [Arr+24; Sha+20], forecasting [PA+18], distribution regression [Lem+21], motion and pattern recognition [LZJ17; Yan+22; Gra13] (the latter is the winner of ICDAR 2013 challenge on online Chinese character recognition). For an exhaustive survey on the signature method for time series analysis, we refer to [LM24].

In the next subsection, we introduce briefly the main results that we will present in the following chapters. In particular, we outline our contributions and the corresponding publications are given in Section 1.3.5.

1.3 Contributions

1.3.1 Averaging signatures (Chapter 3)

To be able to average data, that is, to view things on a coarser scale, is crucial for multiple reasons. First, averaging summarizes or simplifying the information contained in a set of points into a single value. This value provides an insightful description of data with a global summary and, if we are looking to group points

that are close to each other, local averages can be used to classify the data (which is the core idea of machine learning strategies such as K-means). That is, we can unveil relationships between datasets by comparing averages. Additionally, insights over time can be obtained as averages can be computed over fixed periods and compared (e.g., calculating an average value for each month). This can be used for various tasks such as tracking the evolution of global trends, detecting cycles, monitoring (e.g., quality control in manufacturing), forecasting, and noise reduction.

In Chapter 3, consider the task of averaging a set of N points $\mathbf{x}_1, \dots, \mathbf{x}_N$ lying in the signature space G. As we will see in more details in Chapter 2, the space of signatures is a manifold. On such *curved* space, the computation of the simplest statistics, the mean, can require more work than it does on a Euclidean space. Indeed, the definition of barycenter for Euclidean space $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ cannot be used for manifolds, since in many cases $\bar{x}$ might not belong to the manifold. Take for instance two points on the unit circle: $x_1 = (1, 0)$ and $x_2 = (0, 1)$. The Euclidean barycenter of the two points $\bar{x} = (1/2, 1/2)$ does not belong to the circle.

A generalization of the Euclidean barycenter to manifolds is the Fréchet mean: let $(\mathcal{M}, d)$ be a metric space. Given a set of points $x_1, \dots, x_N \in \mathcal{M}$, the Fréchet mean is the point $\mu \in \mathcal{M}$ such that

$$\mu = \arg\min_{\mu \in \mathcal{M}} \sum_{i=1}^{N} d^2(\mu, x_i). \tag{1.2}$$

This definition can be used for Lie groups. Also, if $d(.,.)$ is a bi-invariant Riemannian metric, then μ is stable by group operations: left and right multiplication, inversion. For instance, stability for the right multiplication means that μy is the Fréchet mean of $\{x_i y\}_{i=1,\dots,N}$. However, if $d(.,.)$ is not bi-invariant, the stability of μ is not ensured. For such cases, the authors of [PL20] have defined a notion of barycenter on Lie groups called the group mean, defined as follows: for a finite set of N points $\mathbf{x}_1, \dots, \mathbf{x}_N$, the group mean $\boldsymbol{\mu}$ is the solution of the following equation:

$$\frac{1}{N}\sum_{i=1}^{N} \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i) = 0\ , \tag{1.3}$$

that is, vectors $\mathbf{v}_i$ in the tangent space at the identity T_1G have mean zero, where $\mathbf{v}_i := \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i)$, as shown in Figure 1.4. Note that the group mean may not exist in the case when the logarithm is not defined globally.

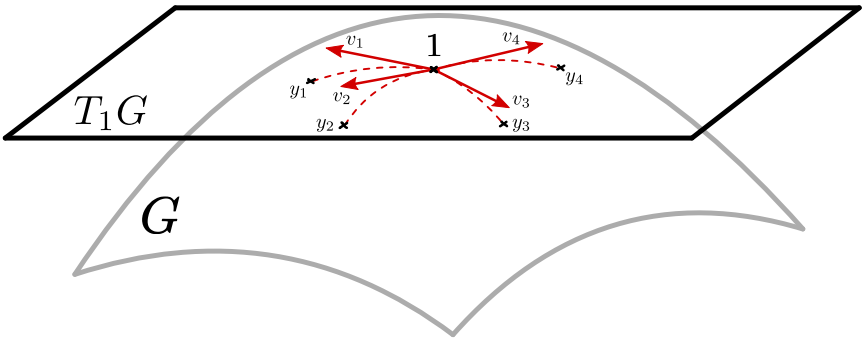


FIGURE 1.4: The group mean $\boldsymbol{\mu}$ on a Lie group G is such that the sum of the $\mathbf{v}_i$ is zero where we have denoted $\mathbf{v}_i := \log(\mathbf{y}_i) = \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i)$ vectors in the tangent space T_1G. Dotted lines are the geodesics on the Lie group G starting from the origin with initial velocity $\mathbf{v}_i$.

In Chapter 3, the main contributions are:

- In Section 3.3, we prove the existence and uniqueness of the group mean for signature features.
- In Section 3.4, we provide a finite time algorithm to exactly compute the barycenter of a finite set of signatures and an implementation in Python, which essentially relies on algebraic properties of the signature.

1.3.2 Dimension reduction of signatures (Chapter 4)

Given a d-dimensional time series $X = (X^1, \dots, X^d)$, its signature up to level L, $\{\mathbf{S}_{(0)}(X), \dots, \mathbf{S}_{(L)}(X)\}$, has a dimension of $\frac{d(d^L-1)}{d-1}$, as it will be shown in Chapter 2. In other words, the number of signature features grows exponentially in the level of truncation L. Numerically, using such a number of features can result in slow computations, especially if d is large and if there are no efficient methods to perform the downstream task, e.g., clustering. Our goal is to develop a faithful dimension reduction method for signatures. For this, we will generalize the well-known Principal Component Analysis (PCA), presented below, to the space of signatures.

Given samples $x_1, \dots, x_N$ in $\mathbb{R}^d$, PCA provides a sequence of best linear approximations to the data, for all ranks $K \leq d$. Denote as $y_i := x_i - \mu$ the centered data points, where μ is the Euclidean mean of the set $x_1, \dots, x_N$. We compute a sequence of vectors $v_1, \dots, v_K$ successively by solving, for all $k = 1, \dots, K$,

$$v_k = \underset{\substack{\|v\|=1 \\ v \perp v_1, \dots, v_{k-1}}}{\arg\min} \sum_{i=1}^{N} \|y_i - \pi_v(y_i)\|^2 \tag{1.4}$$

where π_v is the orthogonal projection onto $\mathrm{span}(v)$. The $(v_k)_k$ are called Principal Directions. We can compress the data by setting $K < d$ and projecting it on $\mathrm{span}(v_1, \dots, v_K)$. The minimization problem (1.4) can be solved using the singular value decomposition.

Many generalizations of PCA to manifolds exist. We focus on the Principal Geodesic Analysis (PGA) [FLJ03] approach, which is typically formulated as follows. Let $x_1, \dots, x_N$ be points on a manifold $\mathcal{M}$ with group mean $\boldsymbol{\mu}$. Then, the PGA minimizes

$$\underset{v \in T_\mu(\mathcal{M}), t_k \in \mathbb{R}}{\arg\min} \sum_{i=1}^{N} d(\gamma_v(t_k), x_k) \tag{1.5}$$

where γ_v is a geodesic starting at μ.

However, is not straightforward to use this definition of PGA for the signatures. The main reason is the absence of a canonical way to define a (bi-invariant) Riemannian metric.

In Chapter 4, we propose the following solutions:

- We propose a new version of PGA that uses a special type of divergence adapted to the signature space.
- We present two algorithms: one that approximate on the tangent space and one that solve the optimization problem on the manifold, along with implementations in Python.

- We apply both dimension reduction methods for classical tasks on both synthetic and real-life data and show that performances are still high while keeping much less features.

1.3.3 Anomaly detection with multiscale signatures (Chapter 5)

In Anomaly Detection (AD) for time series, the goal is to detect a X_i in a dataset of time series $X_1, \dots, X_N$ that behaves *abnormally*, i.e., takes values different from the rest of the data. For instance, in the context of maritime surveillance, there exists major routes of navigation and a boat using a different trajectory could be considered as an anomaly.

In this context of AD for multivariate time series, we show that a multiscale analysis using signature features leads to state-of-the-art results. In addition to improved detection results, we put in evidence the numerical effectiveness of the multiscale signature based analysis. Notably, we avoid huge computational burden, that would appear with other similarity measures. The signature multiscale method can be 100 times faster than single-scale Dynamic Time Warping (DTW). This comes from the combination of group properties with linear interpolations of time series. Indeed, as shown later on in Section 2.3.4, the signature of a whole segment is usually computed by combining signatures of smaller sub-segments. Storing those sub-signatures and using them for the AD has the same computational cost as using only the signature of the whole time series. That is, in the dyadic segmentation presented in Figure 1.5 (coined as hierarchical dyadic windowing in Section 2.3.2), we would store the signature of every sub-segment (which are of lengths 2, 4 and 8). Our method for AD with

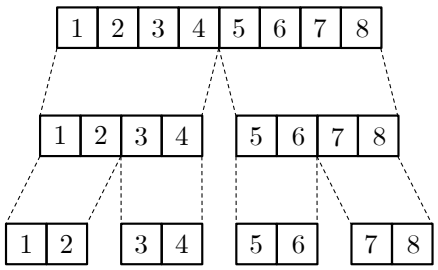


FIGURE 1.5: Dyadic segmentation of a time series of length 8.

the signature relies on this result, which allows us to increase the detection performance of the signature features thanks to the multiscaling operation, while keeping low computational overheads. Our method is compared to standard strategies such as Local Outlier Factor (LOF) with Euclidean / DTW similarity measures.

1.3.4 Clustering with signatures (Chapter 6)

The goal of time series clustering is to detect groups (clusters) of trajectories without having any prior information on the data (unsupervised learning). This allows us to detect hidden underlying structures in the data. As an example, consider the evolution of prices of raw materials (oil, wheat, sand, etc.). A clustering analysis could detect relationships between two materials, providing valuable insights to, e.g., supply chain management or energy providers looking to forecast the demand.

In this work, we show how signatures can be used for clustering time series. First, we investigate similarity measures that can be used in this context, and introduce several similarity measures that are invariant to basic transformations of data. We benchmark those similarity measures against standard distances on time

series: euclidean, DTW, correlation. Real datasets of various shapes are analyzed, e.g., short/long period of recording, small/large number of dimensions.

Second, we show how the notion of barycenter of signatures (Chapter 3) can be used to extend ML methods. For instance, the k-means clustering method can be used with signatures, using the notion of group mean.

We introduce perturbations in the data (jittering, noising) and compare the behavior of similarities on signature features compared to the standard similarities. Signature features provide fluctuating performances while having very competitive computational runtimes, especially for time series recorded over long periods of time.

1.3.5 Publications

Part of the material presented in this thesis has been reviewed. References corresponding to each chapter are given in the following list.

- Chapters 1 and 2 contain introductory material for the thesis.
- Chapter 3 has been accepted for publication in *SIAM Journal on Applied Algebra and Geometry*. Also, early work has been accepted for publication in *ESAIM: Proceedings and Surveys*.
  - M. Clausel, J. Diehl, R. Mignot, L. Schmitz, N. Sugiura and K. Usevich. "The barycenter in free nilpotent Lie groups and its application to iterated-integrals signatures" In: *SIAM Journal on Applied Algebra and Geometry*. 2024. To appear. https://arxiv.org/abs/2305.18996.
  - J. Cugliari, E. Devijver, A. Meynaoui and R. Mignot. "Some recent developments on functional data analysis". In: *ESAIM: Proceedings and Surveys*, EDP Sciences. 2024. To appear.
- Chapter 4 is in the process of submission.
  - R. Mignot, K. Usevich, M. Clausel and N. Sugiura. "Principal Geodesic Analysis for time series encoded with signature features". 2024. https://hal.science/hal-04392568.
- Chapter 5 has been accepted for publication:
  - R. Mignot, V. Mangé, K. Usevich, M. Clausel, J.-Y. Tourneret and F. Vincent. "Anomaly Detection Using Multiscale Signatures". In: *Proceedings of the 32nd European Signal Processing Conference (EUSIPCO)*. Lyon, France. 2024. To appear.
- Chapter 6 is unsubmitted/ongoing work.

Chapter 2

The signature method in Machine Learning

This thesis is anchored in the domain of time series analysis, where the goal is to provide insights on dynamic data sampled at discrete times. We focus on a particular method called the signature, which encodes dependencies among the components of multivariate time series.

This chapter is divided into two parts. First, we define the signature, a mapping of continuous functions, and explain its fundamental properties. In the second part, we move away from the continuous case and address the signature of time series (i.e., discretely sampled data), which will be the main focus of this thesis.

Before diving into the theory of signatures, we establish some notations that will be used throughout the thesis.

2.1 Notations

Throughout the thesis, we use the following notations.

- $\mathbb{R}^{d^k}$ — tensors of size $\underbrace{d \times \cdots \times d}_{k \text{ times}}$ with values in $\mathbb{R}$.

- Bold symbols, e.g., $\mathbf{u}$ — Tensors (elements of $\mathbb{R}^{d^k}$) or elements of the tensor algebra (see Definition 2.10).

- $\otimes$ — Outer product between tensors, that is, given $\mathbf{u} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathbf{v} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$, $\mathbf{w} := \mathbf{u} \otimes \mathbf{v}$ is the element of $\mathbb{R}^{I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M}$ such that

$$\mathbf{w}_{i_1,\ldots,i_N,j_1,\ldots,j_M} = \mathbf{u}_{i_1,\ldots,i_N} \mathbf{v}_{j_1,\ldots,j_M}. \tag{2.1}$$

  We denote as $\mathbf{u}^k := \underbrace{\mathbf{u} \otimes \ldots \otimes \mathbf{u}}_{k \text{ times}}$.

- $\bullet_n$ — mode-n product between a tensor and a matrix, that is, given $\mathbf{u} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $M \in \mathbb{R}^{J \times I_n}$, $\mathbf{u} \bullet_n A$ is the element of $\mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_J \times I_{n+1} \times \cdots \times I_N}$ such that

$$(\mathbf{u} \bullet_n A)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} \mathbf{u}_{i_1 i_2 \ldots i_N} A_{j i_n}. \tag{2.2}$$

- $\mathbf{u}_{(n)}$ — mode-n matricization of tensor $\mathbf{u}$.

- $\odot$ — Hadamard product

- $\|.\|_F$ — Frobenius norm on tensors, that is, let $\mathbf{u} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$,

$$\|\mathbf{u}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} u_{i_1 i_2 \ldots i_N}^2}. \tag{2.3}$$

- $\|.\|_{\mathrm{TV}}$ — Total variation norm, see Definition 2.1.
- $\star$ — Concatenation of functions, defined as: $Z : [0,T] \to \mathbb{R}^d; t \mapsto (X \star Y)(t)$ is the continuous trajectory such that

$$Z(t) = \begin{cases} X(t), & 0 \le t < u \\ Y(t) + X(u) - Y(u), & u \le t \le T \end{cases}. \tag{2.4}$$

  In other words, we translate Y so that it starts at the end point of X.

- $G_{\le L}$ — Space of signatures.
- $\mathfrak{g}_{\le L}$ — Lie algebra associated to the Lie group $G_{\le L}$.
- $[.,.]$ — Lie bracket, as defined in Equation (2.26).

Definition 2.1 (Total variation norm)**.** *The total variation norm $\|.\|_{\mathrm{TV}}$ of a function $X : [0,T] \to \mathbb{R}^d$ is*

$$\|X\|_{\mathrm{TV}} := \sup_{\mathcal{P}} \sum_{\substack{t_i \in \mathcal{P} \\ i \neq 0}} \|X(t_i) - X(t_{i-1})\| \tag{2.5}$$

where the supremum runs over sets $\mathcal{P} = \{0 = t_0 < t_1 < \cdots < t_r = T\}$ partitions of $[0,T]$. Functions of finite total variation are also referred to as functions of bounded variation or functions of finite 1-variation.

2.2 The signature transform

2.2.1 Definition and examples

We now formally define the signature transform. Throughout Section 2.2, we will consider the signature of continuous functions and their main properties. Then, we will introduce the signature of time series in Section 2.3. All integrals are defined in the Riemann sense.

Definition 2.2. *Let X be a continuous function of finite total variation $X : [0,T] \to \mathbb{R}^d$, with T a positive real value. The signature of level $L \in \mathbb{N}$ of X, denoted as $\mathbf{S}_{(L)}(X)$, is a tensor defined as*

$$\mathbf{S}_{(L)}(X) := \int_{0 \le t_1 \le \cdots \le t_L \le T} \cdots \int \dot{X}(t_1) \otimes \ldots \otimes \dot{X}(t_L) dt_1 \ldots dt_L \tag{2.6}$$

where $\otimes$ is the vector outer product (see Notation (2.1)) and $\dot{X}(t) := \frac{d}{dt} X(t)$. For clarity purposes, the bounds $[0,T]$ of the integrals are not denoted on the left hand side. Implicitly, it will always be the definition set of input X. We call signature of X, denoted as $\mathbf{S}(X)$, the infinite collection of signatures at all levels:

$$\mathbf{S}(X) = \{1, \mathbf{S}_{(1)}(X), \mathbf{S}_{(2)}(X), \ldots\}. \tag{2.7}$$

where 1 *is a convention. In the literature, the level* L *is also referred to as the order or the depth of the signature.*

Note that $\mathbf{S}_{(L)}(X)$ is a tensor of size d^L, see Figure 1.2. Indeed, denote the components of X as $X = (X^1, \dots, X^d)$, then coordinate $(i_1, \dots, i_L)$ of tensor $\mathbf{S}_{(L)}(X)$ is

$$\left[\mathbf{S}_{(L)}(X)\right]_{i_1,\dots,i_L} := \int_{0\leq t_1\leq\cdots\leq t_L\leq T} \cdots \int \dot{X}^{i_1}(t_1)\dots\dot{X}^{i_L}(t_L)dt_1\dots dt_L \tag{2.8}$$

for any $1 \leq i_1, \dots, i_L \leq d$.

Now, we show in the following Examples 2.3 and 2.4 how to compute the signature of affine mappings and mappings with values in $\mathbb{R}$ (i.e., one-dimensional).

Example 2.3. *Let the multivariate mapping* $X : [0,T] \to \mathbb{R}^d; t \mapsto (X^1(t), \dots, X^d(t))$ *be an affine trajectory* $X(t) = \mathbf{a}t + \mathbf{b}$*. Equation* (2.6) *gives*

$$\left[\mathbf{S}_{(L)}(X)\right]_{i_1,\dots,i_L} = \frac{1}{L!}\prod_{k=1}^{L}(X^{i_k}(T) - X^{i_k}(0)) = \frac{\mathbf{a}_{i_1}\dots\mathbf{a}_{i_L}}{L!}T^L \tag{2.9}$$

for all $1 \leq i_1, \dots, i_L \leq d$*. In tensor notation:* $\mathbf{S}_{(L)}(X) = \frac{1}{L!}(X(T) - X(0))^{\otimes L}$.

This last example is especially interesting from a computational perspective. Indeed, we will show in Section 2.3.4 how Equation (2.9) appears in the numerical calculation of signatures.

Example 2.4. *Given a continuous mapping of finite total variation* $X : [0,T] \to \mathbb{R}$ *such that* $X(0) = 0$*, we have* $\mathbf{S}_{(1)}(X) = X(T)$ *and*

$$\begin{aligned}\mathbf{S}_{(2)}(X) &:= \int_0^T \int_0^t \dot{X}(u)du\dot{X}(t)dt \\ &= \int_0^T X(t)\dot{X}(t)dt \\ &= \frac{1}{2}(X(T))^{\otimes 2}\end{aligned}$$

where we recall that $(X(T))^{\otimes 2}$ *is the tensor notation (squared with respect to the outer product, see Notation* (2.1)*). By induction, for any level* $L \in \mathbb{N}$*, we have*

$$\begin{aligned}\mathbf{S}_{(L)}(X) &:= \int_0^T \mathbf{S}_{(L-1)}(X_{[0,t]})\dot{X}(t)dt \\ &= \int_0^T \frac{1}{(L-1)!}(X(t))^{L-1}\dot{X}(t)dt \\ &= \frac{1}{L!}(X(T))^{\otimes L}.\end{aligned}$$

Therefore, the signature of a mapping with values in $\mathbb{R}$ *simply gives the successive powers of its global change.*

2.2.2 Main properties

We now present the main properties of the signature function of a continuous function of finite total variation $X : [0,T] \to \mathbb{R}^d$. Throughout Section 2.2.2, proofs are either postponed to the Appendix or provided as references for increased readability.

We start this section with a crucial result in the signature theory, since it demonstrates that the signature is invariant to a reparametrization of time. In other words, signature features are not sensitive to the speed at which sequential data is read.

Proposition 2.5. *Let $X : [0,T] \to \mathbb{R}^d$ be a continuous function of finite total variation. Given a non decreasing continuous surjection φ on $[0,T]$, denote as $\tilde{X} : [0,T] \to \mathbb{R}^d; t \mapsto X(\varphi(t))$. Then*

$$\mathbf{S}(X) = \mathbf{S}(\tilde{X}). \tag{2.10}$$

This is illustrated in Figure 2.1.

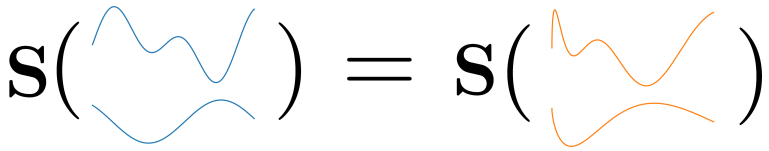


FIGURE 2.1: Illustration of Proposition 2.5. In blue, a two dimensional function X and in orange, a reparametrization $\widetilde{X} = X \circ \varphi$.

In fact, the signature characterizes trajectories of finite total variation up to a certain class of equivalence. To be more precise, we need the following definition of tree-like functions [HL10, Definition 1.3].

Definition 2.6 (Tree-like)**.** *A continuous function of finite total variation $X : [0,T] \to \mathbb{R}^d$ is said to be tree-like if there exists a continuous function $h : [0,T] \to [0,\infty)$ such that $h(0) = h(T) = 0$ and such that*

$$\|X(t) - X(s)\| \leq h(s) + h(t) - 2 \inf_{u \in [s,t]} h(u). \tag{2.11}$$

If h is of finite total variation, we say that X is a Lipschitz tree-like function.

The above formal definition is required to deal with finite variation functions. However, we can think of tree-likeness, geometrically speaking, as two trajectories are tree-like equivalent if they are equal modulo *backtracking* pieces. This is illustrated in Figure 2.2. In fact, tree-like equivalence includes translation and time reparametrization in addition to backtracking.

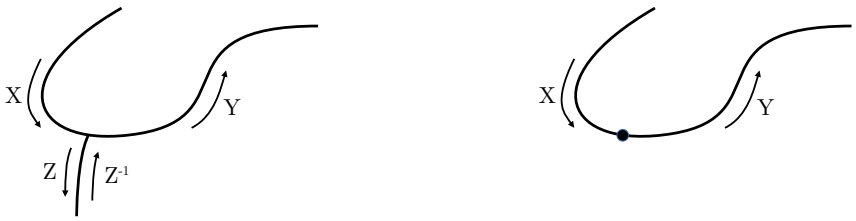


FIGURE 2.2: The trajectory on the left $X \star Z \star Z^{-1} \star Y$ is tree-like equivalent to the trajectory on the right $X \star Y$ ($\star$ is defined in Notation (2.4)).

The class of equivalence characterized by the signature mapping is given in the following result.

Proposition 2.7. *Let $X, Y : [0,T] \to \mathbb{R}^d$ be two functions of finite total variation. Then,*

$$\mathbf{S}(X) = \mathbf{S}(Y) \quad \Leftrightarrow \quad X \star Y^{\leftarrow} \text{ is a Lipschitz tree-like function} \tag{2.12}$$

where $Y^{\leftarrow}(t) = Y(T-t)$ for all $t \in [0,T]$ and where $\star$ is the concatenation operation (see Notation (2.4)).

Proof. See [HL10, Corollary 1.5]. □

A more practical version of Proposition 2.7 is given in the following result.

Corollary 2.8. *Let $X : [0,T] \to \mathbb{R}^d$ be a continuous function of finite total variation with at least one monotonous component. Then $\mathbf{S}(X)$ uniquely characterizes X up to translations.*

Note that the monotonicity can be ensured by adding a dummy component by considering the augmented function $\widetilde{X}(t) := (X(t), t)$. To sum up, signatures are invariant to the starting position of the function, its time parametrization and backtracking parts.

Another attractive feature of the signature transform is the following. Observe from Definition 2.2 that the dimension of the signature at any level L does not depend on the time horizon T, but only on the dimension d of the input trajectory: the signature up to order L has size

$$\sum_{k=1}^{L} d^k = \frac{d(d^L - 1)}{d - 1}. \tag{2.13}$$

That is, the signature transform compresses heavily signals recorded on long durations (T large).

We conclude this section with a proposition that illustrates the universality of the signature transform, i.e., its capacity to approximate linearly a broad range of functions. A lot of practical work related to signature features rely on this property.

Proposition 2.9. *Let D be a compact set of trajectories $X : [0,1] \to \mathbb{R}^d$, such that $\|X\|_{\mathrm{TV}} < \infty$ and $X(0) = 0$. Denote $\tilde{X} = (X(t), t)^T$ the time augmented trajectory of X. Let $f : D \to \mathbb{R}$ be a continuous function. Then, for any $\varepsilon > 0$, it exists $L \in \mathbb{N}$ and $\beta \in T(\mathbb{R}^d)$ such that for all $X \in D$,*

$$\|f(X) - \langle \beta, \mathbf{S}(\tilde{X}) \rangle\| \leq \varepsilon \tag{2.14}$$

with $\langle ., . \rangle$ the Euclidean inner product.

Proof. See [LLN16, Theorem 3.1]. □

Now, in order to manipulate signatures algebraically, we need to introduce notions related to the space of signatures.

2.2.3 Algebraic structure and topology of the signature space

In this section, we give the main tools which will allows us to handle data points in the space of signature features. The first notion that we introduce is the larger space in which the signature space is embedded, called the tensor algebra.

Definition 2.10. *The tensor algebra of vector space $\mathbb{R}^d$, truncated at level $L \in \mathbb{N}$, is*

$$T_{\leq L}(\mathbb{R}^d) := \mathbb{R} \oplus \mathbb{R}^d \oplus (\mathbb{R}^d \otimes \mathbb{R}^d) \oplus (\mathbb{R}^d \otimes \mathbb{R}^d \otimes \mathbb{R}^d) \oplus \cdots \oplus (\underbrace{\mathbb{R}^d \otimes \cdots \otimes \mathbb{R}^d}_{L \text{ times}}) \tag{2.15}$$

and elements of $T_{\leq L}(\mathbb{R}^d)$ are denoted $\mathbf{a} = (\mathbf{a}_0, \ldots, \mathbf{a}_L)$. $T_{\leq L}(\mathbb{R}^d)$ is endowed with the following operations : let $\mathbf{a} = (\mathbf{a}_0, \ldots, \mathbf{a}_L)$, $\mathbf{b} = (\mathbf{b}_0, \ldots, \mathbf{b}_L)$ be two elements of $T_{\leq L}(\mathbb{R}^d)$ and $\lambda \in \mathbb{R}$,

$$\mathbf{a} + \mathbf{b} = (\mathbf{a}_0 + \mathbf{b}_0, \mathbf{a}_1 + \mathbf{b}_1, \ldots, \mathbf{a}_L + \mathbf{b}_L) \tag{2.16}$$

$$\lambda \mathbf{a} = (\lambda \mathbf{a}_0, \lambda \mathbf{a}_1, \ldots, \lambda \mathbf{a}_L) \tag{2.17}$$

$$\mathbf{ab} =: \mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_L) \text{ with } \mathbf{c}_K := \sum_{k=0}^{K} \mathbf{a}_k \otimes \mathbf{b}_{K-k}, \text{ for all } K = 0, \dots, L \tag{2.18}$$

and with neutral element, with respect to $\otimes$,

$$\mathbf{1} := (1, 0_{\mathbb{R}^d}, 0_{\mathbb{R}^d \otimes \mathbb{R}^d}, \dots, 0_{\mathbb{R}^d \otimes \dots \otimes \mathbb{R}^d}) \tag{2.19}$$

and inverse with respect to $\otimes$

$$\mathbf{a}^{-1} := \sum_{k=0}^{L} (-1)^k (\mathbf{a} - \mathbf{1})^k. \tag{2.20}$$

For a visual representation of $T_{\leq L}(\mathbb{R}^d)$, see Figure 1.2.

Proposition 2.11. *$T_{\leq L}(\mathbb{R}^d)$ endowed with above operations is a (non-commutative associative) algebra.*

We have the following result that links the inverse operation with an operation on trajectories.

Proposition 2.12. *Let $X : [0, T] \to \mathbb{R}^d$ be a continuous function of finite total variation. Denote $X^{\leftarrow}$ the the trajectory run backwards, i.e., $X^{\leftarrow}(t) := X(T - t)$, for all $t \in [0, T]$. Then*

$$\mathbf{S}(X^{\leftarrow}) = (\mathbf{S}(X))^{-1} \tag{2.21}$$

with inverse operation given in (2.20).

Proof. See [LCL07, Proposition 2.14]. □

In the following, the space of signatures truncated at level $L \in \mathbb{N}$ will be denoted as

$$G_{\leq L} = \left\{ \mathbf{S}_{\leq L}(X) \text{ with } X : [0, T] \to \mathbb{R}^d \text{ continuous and } \|X\|_{\mathrm{TV}} < \infty \right\}. \tag{2.22}$$

In the literature, $G_{\leq L}$ is known as the nilpotent free Lie group of step L over $\mathbb{R}^d$. This space is a closed Lie subgroup.

Proposition 2.13. *$(G_{\leq L}, \otimes)$ is a Lie group which means that $G_{\leq L}$ is a smooth manifold with a smooth group structure, that is mappings $G \times G \to G, (\mathbf{g}, \mathbf{h}) \mapsto \mathbf{gh}$ (we omit the $\otimes$ notation) and $G \to G, \mathbf{g} \mapsto \mathbf{g}^{-1}$ are smooth. Moreover, it is embedded into the tensor algebra*

$$G_{\leq L} \subset T_{\leq L}(\mathbb{R}^d). \tag{2.23}$$

Proof. See [LCL07, Proposition 2.25]. □

We now give a few results regarding the geometry of the signature space. Note that a few elements and references on differential geometry are given in Appendix C. In particular, those elements will be used in Chapter 4.

Denote as $T_{\mathbf{1}}(G_{\leq L})$ the tangent space at the identity of $G_{\leq L}$. Note that this tangent space can be identified with the associated Lie algebra of $G_{\leq L}$, usually denoted as $\mathfrak{g}_{\leq L}$. The group exponential $\exp : T_{\mathbf{1}}(G_{\leq L}) \to G_{\leq L}$ is

$$\exp(\mathbf{v}) = \sum_{k=0}^{L} \frac{\mathbf{v}^k}{k!}. \tag{2.24}$$

The group logarithm $\log : G_{\leq L} \to T_{\mathbf{1}}(G_{\leq L})$ is defined everywhere on $G_{\leq L}$ and it verifies

$$\log(\mathbf{g}) = \sum_{k=1}^{L} \frac{(-1)^{k+1}}{k}(\mathbf{g} - \mathbf{1})^k. \tag{2.25}$$

Those two important mappings, exp and log, allows us to go from the signature space to its tangent space at identity and vice-versa. Note that the tangent space at the identity of a Lie group is a specific space, called its Lie algebra. The Lie algebra plays a crucial role in understanding the structure and properties of the corresponding Lie group. Elements of the Lie algebra of signatures are called logsignatures, see Section 2.2.4 below.

The following results asserts that $G_{\leq L}$ is globally diffeomorphic to its Lie algebra via the exponential mapping.

Proposition 2.14. *The exponential map of $G_{\leq L}$ is a global diffeomorphism.*

Proof. See [LCL07, Lemma 2.24]. □

As an associative algebra, $T(\mathbb{R}^d)$ has a Lie bracket, defined as follows:

$$[\mathbf{a}, \mathbf{b}] = \mathbf{ab} - \mathbf{ba}. \tag{2.26}$$

The Lie bracket reflects the non-commutative structure and appears in the following classical result, the Baker–Campbell–Hausdorff (BCH) theorem, which gives an explicit formula for the product of two exponentials of elements of the Lie algebra of signatures. Note that the theorem is valid in the Lie algebra of any Lie group.

Theorem 2.15 (BCH formula)**.** *Let $\mathbf{u}, \mathbf{v} \in \mathfrak{g}$ be two elements of the Lie algebra of signatures. We have*

$$e^{\mathbf{u}} e^{\mathbf{v}} = e^{\mathbf{w}} \tag{2.27}$$

where

$$\mathbf{w} = \mathbf{v} + \int_0^1 H(e^{t\mathrm{ad}_{\mathbf{u}}} e^{\mathrm{ad}_{\mathbf{v}}})\mathbf{u}\,dt \tag{2.28}$$

with $\mathrm{ad}_{\mathbf{u}}$ defined as $\mathrm{ad}_{\mathbf{u}}(\mathbf{z}) = [\mathbf{u}, \mathbf{z}]$ and

$$H(\mathbf{z}) := \sum_{k \geq 0} \frac{(-1)^k}{k+1}(\mathbf{z} - 1)^k. \tag{2.29}$$

for all $\mathbf{z} \in \mathfrak{g}$.

Proof. See [FV10, Theorem 7.24]. □

The first terms of the BCH formula are the following:

$$\log(e^{\mathbf{u}} e^{\mathbf{v}}) = \mathbf{u} + \mathbf{v} + \frac{1}{2}[\mathbf{u}, \mathbf{v}] + \frac{1}{12}[\mathbf{u}, [\mathbf{u}, \mathbf{v}]] - \frac{1}{12}[\mathbf{v}, [\mathbf{u}, \mathbf{v}]] + \dots \tag{2.30}$$

Remark 2.16. *For elements $\mathbf{u}, \mathbf{v}$ in the Lie algebra of truncated signatures $\mathfrak{g}_{\leq L}$, the BCH formula expand into a finite number of terms. This is useful to obtain explicit formulas and finite iterative algorithms, as it will be done in Chapters 3 and 4.*

Now, we introduce a proposition that characterizes elements of the signature space inside the ambient tensor algebra. For this, we need the notion of shuffle product.

Definition 2.17 (Shuffle product). *A permutation σ of $(1, \dots, N+M)$ is called a (N, M)-shuffle if $\sigma^{-1}(1) < \cdots < \sigma^{-1}(N)$ and $\sigma^{-1}(N+1) < \cdots < \sigma^{-1}(N+M)$.*

Let $I = (i_1, \dots, i_N)$ and $J = (j_1, \dots, j_M)$ be two sets of integers such that $1 \leq i_1, \dots, i_N \leq d$ and $1 \leq j_1, \dots, j_M \leq d$. We call shuffle product of I and J, denoted as $I \sqcup\!\sqcup J$,

$$I \sqcup\!\sqcup J := \left\{(k_{\sigma(1)}, \dots, k_{\sigma(N+M)}) \text{ such that } \sigma \in (N, M)\text{-shuffle}\right\}, \tag{2.31}$$

with $(k_1, \dots, k_N, k_{N+1}, \dots, k_{N+M}) = (i_1, \dots, i_N, j_1, \dots, j_M)$.

That is, $I \sqcup\!\sqcup J$ is the set of all words formed by the shuffling of words I and J without changing the order of letters. Or using a game analogy: we perform the riffle shuffle of two decks of cards.

The notion of shuffle product is best understood through an example. Using a card game analogy, the shuffle product is the set of all possible riffle shuffles of two decks of cards. Also, see the following example.

Example 2.18. *Let $d = 3$ and let $I = (3, 1)$ and $J = (2)$ be two set of indices. Then,*

$$I \sqcup\!\sqcup J = \{(3, 1, 2), (3, 2, 1), (2, 3, 1)\}. \tag{2.32}$$

This shuffle notion allows us to introduce the following result, which characterizes elements of the signature space inside the ambient tensor algebra.

Proposition 2.19. *An element $a \in T_{\leq L}(\mathbb{R}^d)$ is an element of $G_{\leq L}$ if and only if we have, for any set of indices I and J,*

$$a_I a_J = \sum_{K \in I \sqcup\!\sqcup J} a_K. \tag{2.33}$$

Proof. See [LG20, Theorem 33]. □

Another central aspect of signatures is given in the following proposition. It shows that the product of two signatures $\mathbf{S}(X)$ and $\mathbf{S}(Y)$ is related to the concatenation of the underlying trajectories X and Y. One crucial use of Proposition 2.20 is to compute the signature of discrete functions, as it will be shown in Section 2.3.4.

Proposition 2.20 (Chen identity). *Let $X : [0, u] \to \mathbb{R}^d$ and $Y : [u, T] \to \mathbb{R}$ be two continuous functions with finite total variation. We have*

$$\mathbf{S}(X \star Y) = \mathbf{S}(X)\mathbf{S}(Y) \tag{2.34}$$

where $\star$ is the concatenation operation (see Notation (2.4)). Equation (2.34) is known as Chen identity and is illustrated in Figure 2.3.

Proof. See [LCL07, Theorem 2.9]. □

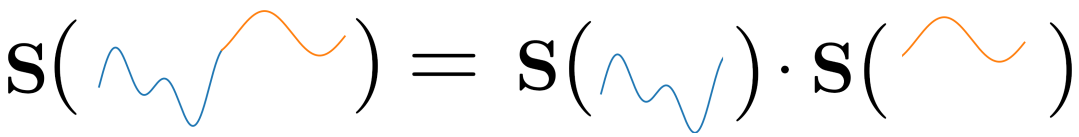


FIGURE 2.3: Illustration of Equation (2.34) with a one dimensional trajectory. X is represented by the blue curve and Y by the orange curve.

In the following, we introduce the usual metrics on the tensor algebra $T(V)$ and on the space of signatures $G_{\leq L}$. On $T(V)$, we can naturally extend classical

inner products. For instance, we have the standard Frobenius inner product on $V^{\otimes k}$ defined as, for any $\mathbf{a}_k, \mathbf{b}_k \in V^{\otimes k}$ (using notation $V^{\otimes k} := \underbrace{V \otimes \ldots \otimes V}_{k \text{ times}}$),

$$\langle \mathbf{a}_k, \mathbf{b}_k \rangle_F := \sum_{i_1=1}^{d} \sum_{i_2=1}^{d} \cdots \sum_{i_k=1}^{d} [\mathbf{a}_k]_{i_1,\ldots,i_k} [\mathbf{b}_k]_{i_1,\ldots,i_k}. \tag{2.35}$$

Then, an inner product on $T(V)$ is defined as, for any $\mathbf{a}, \mathbf{b} \in T(V)$,

$$\langle \mathbf{a}, \mathbf{b} \rangle_F := \sum_{k \geq 0} \langle \mathbf{a}_k, \mathbf{b}_k \rangle_F . \tag{2.36}$$

The standard metric on $G_{\leq L}$ is the Carnot-Carathéodory norm which we define now.

Definition 2.21. *Given $g \in G_{\leq L}$, the Carnot-Carathéodory norm is*

$$\|g\|_{\mathrm{CC}} := \inf \left\{ \mathrm{length}(X) : X \text{ continuous}, \|X\|_{\mathrm{TV}} < \infty \text{ and } \mathbf{S}_{\leq L}(X) = g \right\}. \tag{2.37}$$

where $\mathrm{length}(X) := \int_0^T |dX|$.

An important result associated to the Carnot-Carathéodory norm is given in the following result.

Theorem 2.22. *Let $g \in G_{\leq L}$. The Carnot-Carathéodory norm is finite and attained, that is, it exists a continuous X^* with finite total variation such that $\|g\|_{\mathrm{CC}} = \mathrm{length}(X^*)$ and $\mathbf{S}_{\leq L}(X^*) = g$. Moreover, X^* can be chosen to be Lipschitz continuous and of constant speed $|\dot{X}^*(t)| = |\dot{X}^*(0)|$.*

Proof. See [FV10, Theorem 7.32]. □

In practice, this norm is difficult to compute and we rely on other metrics, such as the Euclidean norm or homogeneous norms. As it will be shown in Chapter 6, learning performances with signature features depend heavily on the choice of the norm.

Now we introduce results related to the continuity of the signature mapping.

Proposition 2.23 (Continuity)**.** *For all $L \geq 1$, the following signature mapping*

$$X \mapsto \mathbf{S}_{\leq L}(X) \in T_{\leq L}(\mathbb{R}^d) \tag{2.38}$$

is continuous, where $X : [0, T] \to \mathbb{R}^d$ is of finite total variation.

Proof. This comes from the fact that the signature mapping is the solution of a differential equation [LCL07, Lemma 2.10]. □

The following result shows that if two paths are close in the total variation norm, then their signature at level L are close in the Frobenius norm.

Proposition 2.24. *Let $X : [0, T] \to \mathbb{R}^d$ be a continuous function of finite total variation. Then for all $L \geq 1$,*

$$\|\mathbf{S}_{(L)}(X)\|_F \leq \frac{1}{L!} (\|X\|_{\mathrm{TV}})^L \tag{2.39}$$

that is

$$\|\mathbf{S}(X)\|_F \leq \exp(\|X\|_{\mathrm{TV}}). \tag{2.40}$$

Proof. See [LCL07, Proposition 2.2]. □

Proposition 2.25 below demonstrates how Carnot-Carathéodory norm on signatures is related to the total variation norm on trajectories. For algorithmic considerations, Proposition 2.24 might be favored as it relies on the Frobenius norm which is more practical than the Carnot-Carathéodory norm.

Proposition 2.25 (Modulus of continuity)**.** *Let $X, Y : [0,T] \to \mathbb{R}^d$ be two continuous functions of finite total variation. Then, for all $L \geq 1$, it exists C_L s.t. for all $0 \leq s < t \leq T$,*

$$\left\|\mathbf{S}_{(L)}(X|_{[s,t]}) - \mathbf{S}_{(L)}(Y|_{[s,t]})\right\|_{CC} \leq C_L \alpha^{L-1} \left\| X|_{[s,t]} - Y|_{[s,t]} \right\|_{TV} \tag{2.41}$$

where $\alpha \geq \max\{\|X|_{[s,t]}\|_{TV}, \|Y|_{[s,t]}\|_{TV}\}$. In particular, for $s = 0$, $t = T$ and if $\|X\|_{TV}$ and $\|Y\|_{TV}$ are less or equal to one, then

$$\left\|\mathbf{S}_{(L)}(X) - \mathbf{S}_{(L)}(Y)\right\|_{CC} \leq C_L \left\|X - Y\right\|_{TV}. \tag{2.42}$$

Proof. See [FV10, Proposition 7.63] which rely mainly on Equation (2.39). □

2.2.4 Logsignature transform

We conclude Section 2.2 with some remarks regarding the log operation in the signature space.

The logarithm of a signature, or logsignature, stores the same information which is contained in the signature, but with less features. However, it does not verify the nonlinearity approximation result (Proposition 2.9). Formally, the dimension of the logsignature truncated at level L is

$$\sum_{\ell=1}^{L} \frac{1}{\ell} \sum_{a|\ell} \mu(a) d^{l/a} \tag{2.43}$$

where μ denotes the Möbius function. In Table 2.1, we present the dimension of both the signature and the logsignature for various truncation level L and trajectory dimension d. For instance, for $d = 3$ and $L = 5$, the signature $\mathbf{S}_{\leq L}(X)$ of a d-dimensional time series X has 364 coefficients when $\log \mathbf{S}_{\leq L}(X)$ is composed of only 80 coefficients.

	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$
$L=2$	7; 3	13; 6	21; 10	31; 15	43; 21	57; 28
$L=3$	15; 5	40; 14	85; 30	156; 55	259; 91	400; 140
$L=4$	31; 8	121; 32	341; 90	781; 205	1555; 406	2801; 728
$L=5$	63; 14	364; 80	1365; 294	3906; 829	9331; 1960	19608; 4088

TABLE 2.1: Dimension of (signature ; logsignature) for various dimension d and truncation level L.

However, it has been shown that in practice signature features perform better than logsignatures [Mor+21], at the cost of a higher storage complexity.

Now, we move from continuous to discrete representations, i.e., time series.

2.3 Time series analysis with signature features

In this section, we present the signature mapping from a machine learning perspective, emphasizing the computational aspects.

2.3.1 From continuous mappings to time series

Now that we have presented the signature of continuous functions, we introduce the signature of time series. Time series are functions $X(t) \in \mathbb{R}^d$ with t taking values in a finite set $\{t_1, \dots, t_T\}$. In other words, a time series is an ordered set of points of a space $\mathcal{V}$, with t_i usually being time indices. In the following, we will only consider $\mathcal{V} = \mathbb{R}^d$, although we could also consider more general structures, such as matrices $\mathcal{V} = \mathbb{R}^{n\times m}$ or tensors $\mathcal{V} = \mathbb{R}^{I_1\times\cdots\times I_n}$, as long as we have an ordering on the considered set.

To define the signature of a time series $\mathbf{S}(X)$, we switch from a discrete representation of the time series to a continuous one, as the signature mapping $\mathbf{S}$ is defined for continuous functions. For this, we augment X by computing the linear interpolation between each pair of successive observations $(X(t_i), X(t_{i+1}))$. Then, by combining the signature of a linear function, see Equation (2.9) and the Chen identity, see Equation (2.34), we can obtain a closed form expression of the (piecewise linear) interpolated time series. Thereafter, if X is a time series, $\mathbf{S}(X)$ denotes the signature of the linearly interpolated X.

2.3.2 Preprocessing and augmentations

The signature as a feature in Machine Learning tasks has shown to perform better when input time series are preprocessed in specific ways. We now detail preprocessing strategies which, in our experience, lead to better performances and also which are often used in the literature. A benchmark study of preprocessing strategies for the signature method and their impact on performances for a downstream classification task was established in [Mor+21].

The main preprocessing strategies are the following.

Time augmentation. We consider $\widetilde{X}(t) = (X(t), t)$ instead of X. Note that this is a way to remove the signature invariance to time reparametrization, see Proposition 2.5, which might be necessary for specific real life applications where the speed of time series gives crucial insights on the data.

Lead-lag augmentation. We consider the following transformation

$$\widetilde{X} = \begin{pmatrix} X(t_0) & X(t_1) & X(t_1) & X(t_2) & X(t_2) & \dots & X(t_T) & X(t_T) \\ X(t_0) & X(t_0) & X(t_1) & X(t_1) & X(t_2) & \dots & X(t_{T-1}) & X(t_T) \end{pmatrix}, \tag{2.44}$$

that is, if $X(t) \in \mathbb{R}^d$ then $\widetilde{X}(t) \in \mathbb{R}^{2d}$. The idea is that, since $\mathbf{S}(X)$ encodes dependencies between components, $\mathbf{S}(\widetilde{X})$ encodes dependencies between a component and the lagged values of another component, which may be valuable in a further analysis of the signature coefficients.

Note that the transformation in Equation (2.44) is the first order lead-lag augmentation and we can construct in the same way the k-th order lead-lag transformation of X.

Sliding windows. This consists in sliding a fixed width window over the time series as shown in Figure 2.4. This window is used for moving averages or Convolutional Neural Networks.

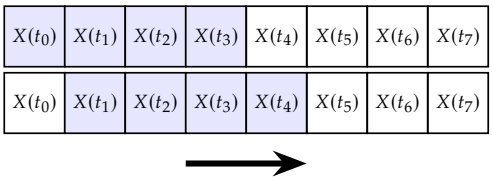


FIGURE 2.4: Sliding window.

Expanding windows. This consists in keeping the left bound of the window fixed and the right bound increases, as shown in Figure 2.5.

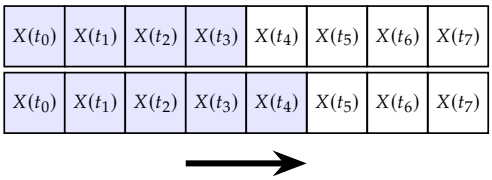


FIGURE 2.5: Expanding window.

Hierarchical dyadic windows. This consists in dividing the time series in two halves. Then dividing the first half and the second half each in two halves, etc., as shown in Figure 2.6.

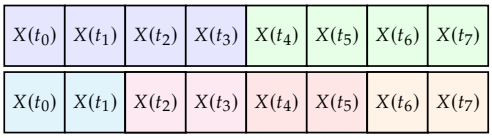


FIGURE 2.6: Hierarchical dyadic window.

Normalizations. Time series or signature features can be scaled in order to improve learning performances. Time series can be scaled in the following way: for each X in the dataset, we apply

$$X \leftarrow \frac{X}{\|X\|_{TV}}. \tag{2.45}$$

In addition, signature coefficients can also be scaled: let $a = (a_1, \dots, a_L)$ be a signature, then we can apply $a_i \leftarrow (a_i)^{1/i}$ or $a_i \leftarrow \frac{1}{i!}a_i$ for all $i = 1, \dots, L$. Normalization of signature coefficients is discussed in further details in Chapter 6.

2.3.3 Application of the signature method for time series analysis

We now detail examples in the literature of applications where the signature has achieved state-of-the-art performances. For a more exhaustive overview of applications of the signature methods in Machine Learning, we refer to [LM24].

Linear regression of signature features. The signature has been used in regression models in several articles. In [Fer22], the author applies a linear regression directly on signature features to forecast the next value of a time series. This work relies on the following identity from Proposition 2.9,

$$\|f(X) - \langle \beta, \mathbf{S}(X) \rangle\| \leq \varepsilon \tag{2.46}$$

where we approximate any transformation f of time series X with a linear combination of signature coefficients.

In [Lem+21], the authors introduce the Signature of Expected Signatures (SES) features for distribution regression. This time, instead of using Proposition 2.9 to approximate functions of time series, they use a result, see [Lem+21, Theorem 3.2], which shows that SES features approximate functions of distributions of time series. We use a method derived from this one is used in Chapter 4.

Signature and Kernel methods. For an introduction to kernel methods in Machine Learning, we refer to [HSS08] and [HTF09, Sections 5.8 and 12.3]. The signature kernel is introduced in [KO19], which we give the definition now. Denote as $\mathcal{P}_{\mathcal{X}}$ the space of mappings (paths) of the form $X : [0,1] \to \mathcal{X} = \mathbb{R}^d$ and consider the kernel $k_X : \mathcal{P}_{\mathcal{X}} \to \mathcal{P}_{\mathcal{H}}$ defined as $k_X = (k(X(t), .))_{0 \leq t \leq 1}$ where $\mathcal{H}$ is the corresponding Reproducing Kernel Hilbert Space (RKHS). For any $h \in \mathcal{P}_{\mathcal{H}}$, we have $\mathbf{S}(h) \in \mathcal{H}' := (\mathcal{H}^{\otimes 0} \times \mathcal{H}^{\otimes 1} \times \mathcal{H}^{\otimes 2} \times \dots)$. The signature kernel is the mapping $k^{\oplus} : \mathcal{P}_{\mathcal{X}} \times \mathcal{P}_{\mathcal{X}} \to \mathbb{R}$ defined as

$$k^{\oplus}(X, Y) = \langle \mathbf{S}(k_X), \mathbf{S}(k_Y) \rangle_{\mathcal{H}'} . \tag{2.47}$$

The main result of [KO19] is that there is a closed formula for $k^{\oplus}$ and thus we do not compute $\mathbf{S}(k_X)$ nor $\langle ., . \rangle_{\mathcal{H}'}$ (coined as kernel trick in the literature on kernel methods).

An especially interesting outcome is that it allows us to extend algorithms for static data that rely on a kernel method to algorithms for time series by using the signature kernel. For instance, in [CO22], the authors develop a Mean Maximum Discrepancy (MMD) test, which is a standard kernel based statistical test to determine whether two given samples are issued from the same distribution, i.e., $H_0 : \mathcal{L}(X) = \mathcal{L}(Y)$. In [TO20], the authors use gaussian processes with a signature kernel as covariance to perform learning on time series with Bayesian tools.

Signature and deep learning. In [Bon+19], the authors develop a neural network architecture that contains a signature layer, that is a layer that takes as input time series and outputs signatures. They show that their model performs better than classical deep architectures (Recurrent Neural Networks, Gated Recurrent Units, Long Short-Term Memory) for the estimation of the Hurst parameter of fractional Brownian motions, which are important mappings, for instance in financial modeling. Note that backpropagation for signatures is implemented in Python packages `iisignature` (CPU oriented) and `Signatory` (GPU oriented), both mentioned later on in Section 2.3.6.

Anomaly detection. There are several works involving the signature to detect abnormal time series in a dataset. For instance, in [Sha+20], the authors develop a semi-supervised learning method called `SigMahaKNN` that computes anomaly scores using a generalized Mahalanobis distance on signature features. Novelty detection, i.e., given a non polluted dataset detect whether a new instance is an outlier, is tackled in [Arr+24]. In Chapter 5, we introduce a signature method that rely on the dyadic windowing of time series. Other works are presented in the survey [LM24, Section 14].

2.3.4 Time and storage complexities

From Example 2.3, we have that for any affine function $X : [0, t_1] \to \mathbb{R}^d$, that is X is of the form $X(t) = \mathbf{a}t + \mathbf{b}$ for all $t \in [0, t_1]$ and fixed $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$,

$$\mathbf{S}(X) = \exp\left(X(t_1) - X(0)\right) \tag{2.48}$$

where $X(t_1) - X(0)$ is to be seen as an element of $T(\mathbb{R}^d)$: $(1, X(t_1) - X(0), 0, 0, \dots)$. By combining Equation (2.48) with Chen identity (Proposition 2.20), we have that for any piecewise affine function $X : [0, T] \to \mathbb{R}^d$, that is X is affine on each segment

$[t_i, t_{i+1}]$ where $0 = t_0 < t_1 < \cdots < t_r = T$,

$$\mathbf{S}(X) = e^{X(t_1)-X(0)} e^{X(t_2)-X(t_1)} \dots e^{X(T)-X(t_{r-1})}. \tag{2.49}$$

From this last equation, we see that the computation of the signature of a time series is a succession of $(T-1)$ outer products. On the truncated tensor algebra $T_{\leq L}(\mathbb{R}^d)$, the product operation $\mathbf{a}e^Z$ is $O(d^L)$. Thus, the time complexity of the signature computation is $O(Td^L)$. As pointed out in Equation (2.13), the storage complexity of the signature is $O(d^L)$ and does not depend on T.

2.3.5 Reconstruction of a time series given its signature

The task of reconstructing a time series X given its signature $\mathbf{S}_{\leq L}(X)$, up to translation and time parametrization, is challenging and has been tackled in several studies.

One method, coined as the insertion algorithm [CL19; Fer+23], computes iteratively the slopes of each affine piece of the time series. This method has a serious limitation, as it requires the signature up to level $L+1$ in order to reconstruct a time series of length L. A Python implementation of this reconstruction method is available in package `Signatory` [KL21].

Another approach has been developed in [LX18], but it requires even more signature levels ($\sim 2dL^3 \log L$) to reconstruct a time series of length L.

2.3.6 Existing softwares

To handle signatures, we need an efficient implementation of the operations of the tensor algebra $T(\mathbb{R}^d)$, such as the product $\otimes$, on which rely the calculation of both the exponential, see Equation (2.24), and the signature, see Equation (2.49). Several softwares (C++ wrapped in Python) exist to compute the signature transform of a time series and handle such tensor operations. For the experiments of this thesis, we use `iisignature` [RG20]. Practical users might also be interested in `RoughPy` (the most active library in 2024), `Signatory` [KL21] (GPU compatible) and `signax` (JAX compatible), all available on the official Python Package Index (PyPI) repository. Also, note that the signature, as well as some preprocessing operations such as those presented in Section 2.3.2, have recently been incorporated into the wider project `sktime` [Lön+19], a scikit-learn compatible interface for Machine Learning with time series.

2.3.7 Connection with Rough paths theory

We extend the notion of total variation with the notion of p-variation.

Definition 2.26. *Let $p \geq 1$ be a real value. The p-variation norm $\|.\|_{p-\text{var}}$ of a function $X : [0,T] \to \mathbb{R}^d$ is*

$$\|X\|_{p-\text{var}} := \left(\sup_{\mathcal{P}} \sum_{t_i \in \mathcal{P},\, i \neq 0} \|X(t_i) - X(t_{i-1})\|^p \right)^{1/p} \tag{2.50}$$

where the supremum runs over sets $\mathcal{P} = \{0 = t_0 < t_1 < \cdots < t_r = T\}$ partitions of $[0,T]$. Note that for $p = 1$, we get the total variation norm (Definition 2.1).

Given a continuous function $X : [0, T] \to \mathbb{R}^d$, the signature is the solution of the following differential equation:

$$d\mathbf{S}_{[0,t]}(X) = \mathbf{S}_{[0,t]}(X) \otimes dX_t \ , \qquad \mathbf{S}_{[0,0]}(X) = \mathbf{1}. \tag{2.51}$$

In order to give meaning to integrals of the form

$$\int_0^t Y_s dX_s \tag{2.52}$$

for $X : [0, T] \to \mathbb{R}^d$ continuous of finite p-variation, $Y : [0, T] \to \mathbb{R}^d$ continuous of finite q-variation and $\frac{1}{p} + \frac{1}{q} > 1$, we can use the notion of Young integral [You36], which is a continuous generalization of the Riemann-Stieltjes integral. However, for $p = q = 2$ (e.g., X a Brownian motion), Young integrals can't be used and another framework is needed to define Equation (2.52). That is the purpose of the rough paths theory, a subfield of stochastic calculus initiated by Terry Lyons in the 1990s [Lyo98], which provides a framework to rigorously study the integration of functions with respect to *rough* functions ($p \geq 2$) and in a way, extend the classical integration theory. It is a natural extension of integrals against α-Hölder continuous functions with $\alpha \in \left(\frac{1}{3}, \frac{1}{2}\right]$, in the same way Young integrals is the natural notion of integral against α-Hölder continuous functions with $\alpha \in \left(\frac{1}{2}, 1\right]$. For further details, a recent introduction to rough paths theory can be found in [FH20].

In our work, we do not consider rough functions as we solely focus on the study of multivariate time series. Those time series are linearly interpolated and thus the input of the signature transform will always be a continuous piecewise linear function. Thus, in our work, the signature is defined with Riemann-Stieltjes integrals.

Chapter 3

Barycenter of signatures

In this chapter, we establish the well-definedness of the barycenter (in the sense of Buser and Karcher) for every integrable measure on the space of signatures (formally, the free nilpotent Lie group of step L over $\mathbb{R}^d$). We provide two algorithms for computing it, using methods from Lie theory (namely, the Baker–Campbell–Hausdorff formula introduced in Section 2.2.3).

3.1 Introduction

In this chapter, we want to view things on a coarser scale by averaging, that is by summarizing or simplifying the information contained in a set of points into a single value. As mentioned in Section 1.3, this value provides an insightful description that can be used in local strategies (e.g., K-means), or for unveiling relationships between datasets, tracking the evolution of global trends, detecting cycles, monitoring (e.g., quality control in manufacturing), forecasting, and noise reduction.

Consider the task of averaging a set of N points $\mathbf{x}_1, \dots, \mathbf{x}_N$ lying on the signature space G. The space of signatures is a manifold. On such *curved* space, the computation of the simplest statistics, the mean, can require more work than it does on a Euclidean space. Indeed, the definition of barycenter for Euclidean space $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ cannot be used for manifolds, since in many cases $\bar{x}$ might not belong to the manifold. Take for instance two points on the unit circle: $x_1 = (1, 0)$ and $x_2 = (0, 1)$. The Euclidean barycenter of the two points $\bar{x} = (1/2, 1/2)$ does not belong to the unit circle.

Lie group barycenters. A generalization of the Euclidean barycenter to manifolds is the Fréchet mean: let $(\mathcal{M}, d)$ be a metric space. Given a set of points $x_1, \dots, x_N \in \mathcal{M}$, the Fréchet mean is the point $\mu \in \mathcal{M}$ such that

$$\mu = \arg\min_{\mu \in \mathcal{M}} \sum_{i=1}^{N} d^2(\mu, x_i). \tag{3.1}$$

This definition can be used for Lie groups. Also, if $d(.,.)$ is a bi-invariant Riemannian metric, then μ is stable by group operations: left and right multiplication, inversion. For instance, stability for the right multiplication means that μy is the Fréchet mean of $\{x_i y\}_{i=1,\dots,N}$. However, if $d(.,.)$ is not bi-invariant, the stability of μ is not ensured. For such cases, the authors of [PL20] have defined a notion of barycenter on Lie groups called the group mean. Historically, the concept of a Lie group barycenter goes back to [BK81]. The group mean is put in context with other concepts of geometric means in [PL20] and barycenters in the group of rotations are treated in [Moa02].

Averaging signatures. In this chapter, we are interested in computing the group mean of signatures data. Note that the task of averaging signatures has been previously tackled with the concept of *expected signature* for stochastic processes [Faw02; Ni12] which has proven to be useful. For example, it often characterizes the law of the stochastic process [CL16; Boe+21], a fact that has been successfully applied in data science [SO21; CO22; Sug21]. Maybe surprisingly, the expected signature of several classes of stochastic processes is available either in closed form or can be computed by solving a fixed-point equation [LN15; FHT22]. However, the signature of a continuous mapping or of a time series takes value in a nonlinear space (the Lie group G) as we have seen in Chapter 2. Averaging in the aforementioned works is taken in the tensor algebra $T_{\leq L}(\mathbb{R}^d)$, i.e., the linear ambient space and the obtained expected signature is not an element of the group of signatures anymore (except for a Dirac measure) while the group mean is.

Contributions. Our contributions are as follows:

- We establish unique existence of the barycenter for arbitrary integrable measures on grouplike elements, Theorem 3.9. Previous results needed the assumption of compact support, which is not necessary in the "free" case.
- We show that, for discrete measures (e.g., a collection of grouplike "samples"), there is a finite-time algorithm to compute the barycenter (unlike existing approaches relying on iterative methods, such as fixed-point iteration [PA12]). For this, compare Corollary 3.16 and Proposition 3.29, together with implementations in `SageMath` and `python`.[1]

Notation

We use the following variable names throughout:

- d — dimension of multivariate time series;
- N — number of time series we want to compute the mean of;
- τ — (maximal) length of time series;
- L — truncation level (of the free Lie algebra or the tensor algebra).

3.2 Background

The following definitions and results can be found, for example, in [FV10, Chapter 7]. A recent exposition, with a notation similar to the one used here, can be found in [Die+22, Section 2].

3.2.1 Free Lie algebras and iterated-integral signatures

For a fixed dimension d, we consider the alphabet $A := \{\mathbf{1}, \dots, \mathbf{d}\}$. A word w on A is a (possibly infinite) ordered set of letters (elements of A), e.g., $w = \mathbf{31}$ is a two letters word on A, distinct from the word $\tilde{w} = \mathbf{13}$. Denote as A^* the set of words on A (including the empty word $\mathbf{1}$). The *length* of a word w is denoted by $|w|$. The tensor algebra over $\mathbb{R}^d$ can be realized as the $\mathbb{R}$-vector space over A^*,

$$T(\mathbb{R}^d) := \bigoplus_{k=0}^{\infty} (\mathbb{R}^d)^{\otimes k} \cong \mathrm{span}_{\mathbb{R}}(A^*).$$

[1] Source code is available at `https://github.com/diehlj/free-nilpotent-lie-group-barycenter`.

Elements of it are finite, formal sums of words

$$\sum_{w \in A^*} c_w \, w, \tag{3.2}$$

where the coefficients $c_w \in \mathbb{R}$ are zero, for all but finitely many words w. Such sums are also called (noncommutative) *polynomials*. It becomes an $\mathbb{R}$-algebra when taking the *concatenation* product of words (and extending it linearly). We shall also use the space of formal tensor series, $T((\mathbb{R}^d))$, which contains *all* formal sums of words of form (3.2) with possibly infinite number of non-zero coefficients $c_w \in \mathbb{R}$. Define the two-sided ideal

$$T_{>L}(\mathbb{R}^d) := \left\{ \left(\sum_{w \in A^*} c_w \, w \right) \in T(\mathbb{R}^d) \mid \forall w \in A^* : |w| \leq L \Rightarrow c_w = 0 \right\}.$$

The *truncated tensor algebra* $T_{\leq L}(\mathbb{R}^d)$ defined in Section 2.2.3 can equivalently be defined as the quotient algebra

$$T_{\leq L}(\mathbb{R}^d) := {}^{T(\mathbb{R}^d)}\!/_{T_{>L}(\mathbb{R}^d)}.$$

It can be realized as the space of formal sums (3.2) satisfying $c_w = 0$ for $|w| > L$. The product on it is the concatenation product, where the product of two words w, v with $|w| + |v| > L$ is set to zero. We will use this identification from now on. Like every associative algebra, the tensor algebra, as well as its truncation, is endowed with a Lie bracket given by the commutator $[v, w] = vw - wv$.

The *free Lie algebra* (over $\mathbb{R}^d$) , $\mathfrak{g}(\mathbb{R}^d)$, can be realized as the smallest Lie subalgebra of $T(\mathbb{R}^d)$ containing the letters A.

The *free, step-L nilpotent Lie algebra* (over $\mathbb{R}^d$), $\mathfrak{g}_{\leq L}(\mathbb{R}^d)$, can be realized as the smallest Lie subalgebra of $T_{\leq L}(\mathbb{R}^d)$ containing the letters A.

The *free, step-L nilpotent Lie group* (over $\mathbb{R}^d$), $G_{\leq L}(\mathbb{R}^d)$, can be realized as the image of $\mathfrak{g}_{\leq L}(\mathbb{R}^d)$ under the exponential map (defined in Section 2.2.3). Its product is given by the restriction of concatenation. As the name suggests, $G_{\leq L}(\mathbb{R}^d)$ is a Lie group. Moreover, its Lie algebra is realized by $\mathfrak{g}_{\leq L}(\mathbb{R}^d)$ and the map exp realizes the Lie group exponential. Moreover $\exp : \mathfrak{g}_{\leq L}(\mathbb{R}^d) \to G_{\leq L}(\mathbb{R}^d)$ is a *global* diffeomorphism, with inverse given by the logarithm mapping (see Section 2.2.3).

3.2.2 Baker–Campbell–Hausdorff (BCH) formula

We will use the following classical result (see [FV10, Theorem 7.24] for a proof in a notation close to ours).

Theorem 3.1 (Baker–Campbell–Hausdorff (BCH) formula)**.** *Let X, Y be non-commuting dummy variables (i.e., we work in the free Lie algebra over two letters X, Y). Then*

$$\log(\exp(X)\exp(Y)) = Y + \int_0^1 \Theta\left(\exp(t \operatorname{ad}_X)\exp(\operatorname{ad}_Y)\right) X \mathrm{d}t.$$

Here, ad_X is the linear map, $\operatorname{ad}_X(Z) = [X, Z]$, and

$$\Theta(z) := \sum_{n \geq 0} \frac{(-1)^n}{n+1} (z-1)^n.$$

Spelling out the first few terms of the resulting Lie series:

$$\begin{aligned}\log(\exp(X)\exp(Y)) = X + Y &+ \frac{1}{2}[X,Y] + \frac{1}{12}[X,[X,Y]] - \frac{1}{12}[Y,[X,Y]] \\ &- \frac{1}{24}[Y,[X,[X,Y]]] \\ &- \frac{1}{720}([[[[X,Y],Y],Y],Y] + [[[[Y,X],X],X],X]) \\ &+ \frac{1}{360}([[[[X,Y],Y],Y],X] + [[[[Y,X],X],X],Y]) \\ &+ \frac{1}{120}([[[[Y,X],Y],X],Y] + [[[[X,Y],X],Y],X]) + \cdots\end{aligned}$$

The BCH formula for permuted arguments is simply

$$\mathsf{BCH}(Y,X) := \log(\exp(Y)\exp(X)) = -\log((\exp(Y)\exp(X))^{-1}) = -\mathsf{BCH}(-X,-Y). \tag{3.3}$$

For multiple arguments, a BCH formula can be obtained by composition, which is associative via

$$\mathsf{BCH}(X,\mathsf{BCH}(Y,Z)) = \log(\exp(X)\exp(Y)\exp(Z)) = \log(\exp(\mathsf{BCH}(X,Y))\exp(Z)). \tag{3.4}$$

In the following subsection, we recall the Lyndon basis and its dual, as they are useful to compute $\mathsf{BCH}(X,Y)$.

3.2.3 Basis of the truncated Lie algebra and its dual

A word w over d symbols $\{\mathbf{1},\dots,\mathbf{d}\}$ is a *Lyndon word* if and only if it is nonempty and lexicographically strictly smaller than any of its proper suffixes, that is $w < v$ for all nonempty words v such that $w = uv$ and u is nonempty. For all Lyndon words of length at least 2 (i.e., non-letter Lyndon words), there is a unique choice of u and v, called the *standard factorization* of w, in which v is as long as possible and both u and v are Lyndon words. For every Lyndon word w we obtain a polynomial $\mathcal{B}_w \in T(\mathbb{R}^d)$ via the recursive definition

$$\mathcal{B}_w = \begin{cases} \mathbf{j}, & \text{if } w = \mathbf{j}, \\ [\mathcal{B}_u, \mathcal{B}_v], & \text{if } w \text{ has standard factorization } w = uv. \end{cases}$$

The Lyndon words, sorted[2] by length first and then lexicographically within each length class, form an infinite sequence $(w_i)_{i\in\mathbb{N}}$. For convenience we set $\mathcal{B}_i := \mathcal{B}_{w_i}$ for every $i \in \mathbb{N}$. Let $\mathfrak{g} = \mathfrak{g}(\mathbf{1},\dots,\mathbf{d}) = \mathfrak{g}(\mathbb{R}^d)$ be the free Lie algebra over $\mathbb{R}$ which is generated by $\{\mathbf{1},\dots,\mathbf{d}\}$. Then $(\mathcal{B}_i)_{i\in\mathbb{N}}$ forms an $\mathbb{R}$-basis for $\mathfrak{g}$ ([Gar90, Section 3, 4]).

For the entire section, we fix a truncation level L. The L-truncated Lie algebra $\mathfrak{g}_{\le L}$ with nested Lie brackets bounded by depth L is an $\mathbb{R}$-vector space ([Gar90, Proposition 3.1]) with

$$B := B_{L,d} := \dim_{\mathbb{R}}(\mathfrak{g}_{\le L}) = \sum_{1\le \ell \le L} \frac{1}{\ell} \sum_{a \mid \ell} \mu(a) d^{\frac{\ell}{a}}, \tag{3.5}$$

[2] This order is also known as the shortlex order.

where μ denotes the Möbius function. The expression (3.5) is known as the necklace polynomial. An $\mathbb{R}$-basis is given by the L-truncated ordered Lyndon basis $\mathcal{B}_{1\le b\le B}$ recalled above.

Example 3.2. *For $L = 3$ and $d = 2$ we obtain $B = 5$ and*

$$\mathcal{B}_{1\le b\le B} = \begin{pmatrix} 1 & 2 & [1,2] & [1,[1,2]] & [[1,2],2] \end{pmatrix} \in \mathfrak{g}^5_{\le L}.$$

On the corresponding B-dimensional dual space

$$\hom(\mathfrak{g}_{\le L}, \mathbb{R}) = \{f : \mathfrak{g}_{\le L} \to \mathbb{R} \mid f \ \mathbb{R}\text{-linear}\}$$

we use the dual basis $\mathcal{B}^*_{1\le b\le B}$, satisfying for all $b, j \le B$,

$$\mathcal{B}^*_j(\mathcal{B}_b) = \begin{cases} 1, & \text{if } j = b, \\ 0, & \text{elsewhere.} \end{cases} \tag{3.6}$$

3.2.4 BCH in the truncated Lie algebra

With some abuse of notation, we define for $X, Y \in \mathfrak{g}_{\le L}$ the L-truncated Lie series

$$\mathsf{BCH}(X, Y) := \log(\exp(X)\exp(Y)) \in \mathfrak{g}_{\le L}$$

derived from the BCH formula (Theorem 3.1).

Definition 3.3. *We define a binary operation $\star : \mathbb{R}^B \times \mathbb{R}^B \to \mathbb{R}^B$ of coefficient vectors via*

$$(u \star v)_i := \mathcal{B}^*_i \circ \mathsf{BCH}\left(\sum_{j=1}^{B} u_j \mathcal{B}_j, \sum_{j=1}^{B} v_j \mathcal{B}_j\right)$$

for every $i \le B$ and $u, v \in \mathbb{R}^B$.

Example 3.4. *For $L = d = 2$ we have $B = B_{2,2} = 3$ and*

$$\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \star \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1 + v_1 & u_2 + v_2 & u_3 + v_3 + \frac{1}{2}(u_1 v_2 - u_2 v_1) \end{bmatrix}$$

according to Definition 3.3.

Lemma 3.5. *$(\mathbb{R}^B, \star, 0_B)$ is a group.*

Proof. It is easy to see that $0_B \in \mathbb{R}^B$ is the neutral element with respect to $\star$ thanks to the property

$$\mathsf{BCH}(X, 0) = X = \mathsf{BCH}(0, X).$$

Next, for every $u \in \mathbb{R}^B$, its inverse with respect to $\star$ is nothing but $-u$ thanks to

$$\mathsf{BCH}(X, -X) = 0 = \mathsf{BCH}(-X, X).$$

Finally, associativity with respect to $\star$ follows from (3.4). □

This operation $\star$ is precisely the multiplication in the free Lie group $G_{\le L}(\mathbb{R}^d)$, which can be identified with R^B via a chosen basis.

Example 3.6. *In the setting of Example 3.4, that is $L = d = 2$ and $B = 3$, the group $(\mathbb{R}^3, \star, 0_3)$ according to Lemma 3.5 is isomorphic to the Heisenberg group*

$$H := \left\{ \begin{bmatrix} 1 & a_1 & a_3 \\ 0 & 1 & a_2 \\ 0 & 0 & 1 \end{bmatrix} \in \mathrm{GL}_3(\mathbb{R}) \mid a_1, a_2, a_3 \in \mathbb{R} \right\}$$

with its multiplication inherited from $\mathrm{GL}_3(\mathbb{R})$.

Proof. Define $\Phi : \mathbb{R}^3 \to H$ via

$$\Phi\left(\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}\right) := \begin{bmatrix} 1 & u_1 & u_3 + \frac{1}{2}u_1u_2 \\ 0 & 1 & u_2 \\ 0 & 0 & 1 \end{bmatrix}$$

for every $u_1, u_2, u_3 \in \mathbb{R}$. It is bijective, and it respects the group law with

$$\begin{aligned} \Phi(\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix})\Phi(\begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}) &= \begin{bmatrix} 1 & u_1 + v_1 & u_3 + v_3 + u_1v_2 + \frac{1}{2}(u_1u_2 + v_1v_2) \\ 0 & 1 & u_2 + v_2 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \Phi\left(\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix} \star \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}\right) \end{aligned}$$

for all $v_1, v_2, v_3 \in \mathbb{R}$. Therefore, Φ is an isomorphism of groups. □

3.3 The barycenter in the nilpotent Lie group

3.3.1 Definition and properties

Definition 3.7 ([BK81, Definition 8.1.4],[PL20, Definition 11])**.** *Let H be a Lie group with globally defined logarithm[3] and ν a probability measure on it. We say that $\mathbf{m}_\nu \in H$ is a* ***barycenter*** *or* ***group mean*** *of ν if*

$$0 = \int_G \log(\mathbf{m}_\nu^{-1}\mathbf{x})\, \nu(\mathrm{d}\mathbf{x}). \tag{3.7}$$

The notion of barycenter was introduced in [PL20] using the Cartan-Schouten connection. Informally speaking, the barycenter looks for a point $\mathbf{m}_\nu$ so that the logarithm[4]

$$\log_{\mathbf{m}_\nu} \mathbf{x} = \mathbf{m}_\nu \log(\mathbf{m}_\nu^{-1}\mathbf{x})$$

with respect to $\mathbf{m}_\nu$ has expectation 0. This notion is different from the so-called naive mean, which simply averages the logarithms of the points at the identity, namely

$$\mathbf{m}_\nu^{\text{naive}} := \exp\left(\int_H \log(\mathbf{x})\, \nu(\mathrm{d}\mathbf{x})\right).$$

Note that, in general, $\mathbf{m}_\nu^{\text{naive}} \neq \mathbf{m}_\nu$ and $\mathbf{m}_\nu^{\text{naive}}$ does not possess invariance properties.

Remark 3.8.

[3] In the case that we are interested in, this condition is satisfied, and we can thus omit the usual assumption that ν is supported in a neighborhood of the identity, where the logarithm is well-defined.

[4] The left-hand side is the abstract logarithm at a basepoint in a Lie group; the right-hand side is its concrete realization inside the tensor algebra.

i. *For compact Lie groups, the notion of barycenter corresponds to the Riemannian center of mass for the corresponding bi-invariant Riemannian metric. However, in our case, it is impossible to define the bi-invariant Riemannian metric [PL20].*

ii. *The barycenter of Definition 3.7 formally fits into the framework of proper scoring rules and Bayes acts [Goo52; BG20; BO21]. In that setting, a Bayes act is defined as*

$$a_\mu := \arg\min_a \mathbb{E}_{X\sim\mu}\left[L(a,X)\right],$$

for some loss function L. If H is a Riemannian manifold, if the Riemannian logarithm coincides with the Lie group logarithm, and if we set

$$L(a,X) := ||\log_a(\mathbf{S}(X))||^2,$$

then the condition for the minimum is

$$0 = \mathbb{E}_{X\sim\mu}[\partial_a||\log_a(\mathbf{S}(X))||^2] = -2\mathbb{E}_{X\sim\mu}[\log_a(\mathbf{S}(X))],$$

which, modulo the irrelevant prefactor, is exactly condition (3.7).

Now, as just mentioned, our H of interest is not *Riemannian (it can only be endowed with a compatible sub-Riemannian geometry) and therefore this formal argument does* not *apply. It would be nonetheless interesting to explore what ideas, in particular the concept of "elicitation", from that literature can be applied in our setting.*

Our main results show that for the free Lie groups, the barycenter exists and is unique under standard conditions.

Theorem 3.9. *Let $H = G_{\leq L}(\mathbb{R}^d)$ be the free, step-L nilpotent Lie group (over $\mathbb{R}^d$). Let ν be a probability measure on H such that this measure is integrable when considered as a measure on the ambient linear space $T_{\leq L}(\mathbb{R}^d)$. Then the group mean $\mathbf{m}_\nu$ of ν exists and is unique.*

We state some corollaries of Theorem 3.9 before providing its proof of in Section 3.3.3.

Remark 3.10. *For compactly supported measures, the uniqueness of the barycenter follows from [BK81, Example 8.1.8] (see also [PL20, Theorem 5.16]). For this, note that the free nilpotent Lie group of dimension d of step L is simply connected since it is diffeomorphic to the free Lie algebra of dimension d of step L [FV10].*

Theorem 3.9 is stronger in the sense that it also covers measures that are not compactly supported, and moreover our proof will provide a constructive way to compute the barycenter in a recursive fashion.

Remark 3.11. *If μ is a measure on $\mathcal{G}(\mathbb{R}^d)$ (with all moments well-defined), such that the pushforward $\mu_{\leq L}$ under the projection onto levels $\leq L$ has the appropriate integrability conditions for all L, then the proof shows that level n of the barycenter is independent of L for $L \geq n$. We thus get, projectively, a well-defined barycenter for all of μ.*

Theorem 3.12 (Bi-invariance of the barycenter)**.** *Under assumptions of Theorem 3.9, for $\mathbf{g} \in H$ denote by $(L_\mathbf{g})_*\nu$ the push-forward under left multiplication, i.e., for any bounded function f,*

$$\int_H f(\mathbf{y})\,((L_\mathbf{g})_*\nu)(\mathrm{d}\mathbf{y}) = \int_H f(\mathbf{gx})\,\nu(\mathrm{d}\mathbf{x}).$$

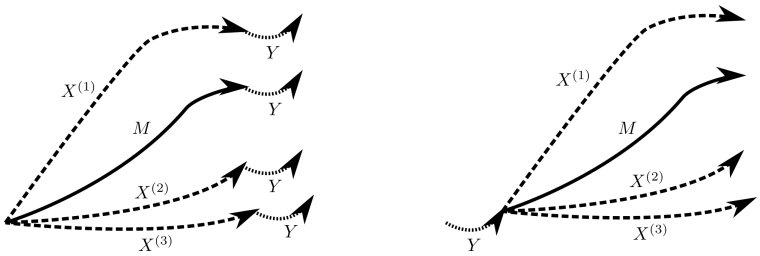


FIGURE 3.1: On the left the mean of the signatures $\mathbf{S}(X^{(i)})$ of three paths is assumed to be given by the signature $\mathbf{S}(M)$ of some path M. If we attach to all paths $X^{(i)}$ a new path segment Y, since the mean is right invariant, this corresponds to attaching that path segment to M. On the right, we see the analogous visualization for left invariance. Due to Chen identity, attaching path corresponds to the group product.

Let $(R_{\mathbf{g}})_*\nu$ be the push-forward under right multiplication. Then $(L_{\mathbf{g}})_*\nu, (R_{\mathbf{g}})_*\nu$ are also integrable in the required sense and

$$\mathbf{m}_{(L_{\mathbf{g}})_*\nu} = \mathbf{g}\mathbf{m}_\nu, \qquad \mathbf{m}_{(R_{\mathbf{g}})_*\nu} = \mathbf{m}_\nu\mathbf{g}.$$

*That is, the barycenter is **bi-invariant** with respect to left and right multiplication, illustrated by Figure 3.1.*

The proof of Theorem 3.12 mainly follows from [PL20, Theorem 5.13], but we provide it below for completeness.

Proof of Theorem 3.12. First,

$$\int \log((\mathbf{g}\mathbf{m}_\nu)^{-1}\mathbf{x})\,(L_{\mathbf{g}})_*\nu(\mathrm{d}\mathbf{x}) = \int \log((\mathbf{g}\mathbf{m}_\nu)^{-1}\mathbf{g}\mathbf{x})\,\nu(\mathrm{d}\mathbf{x}) = \int \log(\mathbf{m}_\nu^{-1}\mathbf{x})\,\nu(\mathrm{d}\mathbf{x}) = 0.$$

Hence $\mathbf{m}_{(L_{\mathbf{g}})_*\nu} = \mathbf{g}\mathbf{m}_\nu$. Further

$$\begin{aligned}\int \log((\mathbf{m}_\nu\mathbf{g})^{-1}\mathbf{x})\,(R_{\mathbf{g}})_*\nu(\mathrm{d}\mathbf{x}) &= \int \log((\mathbf{m}_\nu\mathbf{g})^{-1}\mathbf{x}\mathbf{g})\,\nu(\mathrm{d}\mathbf{x}) = \int \log(\mathbf{g}^{-1}\mathbf{m}_\nu^{-1}\mathbf{x}\mathbf{g})\,\nu(\mathrm{d}\mathbf{x}) \\ &= \mathbf{g}^{-1}\left(\int \log(\mathbf{m}_\nu^{-1}\mathbf{x})\,\nu(\mathrm{d}\mathbf{x})\right)\mathbf{g} = 0.\end{aligned}$$

Here we used

$$\log(\mathbf{g}^{-1}\mathbf{x}\mathbf{g}) = \mathbf{g}^{-1}\log(\mathbf{x})\mathbf{g},$$

which can, for example, be verified by expanding the power series for log. □

Finally, we show that in the special case of a "centered" probability measure, the barycenter coincides with the naive mean.

Lemma 3.13. *In the setting of Theorem 3.9 we have*

$$\mathbf{m}_\nu = \mathbf{1} \iff \mathbf{m}_\nu^{\mathrm{naive}} = \mathbf{1}.$$

Proof. Assume $\mathbf{m}_\nu^{\mathrm{naive}} = \mathbf{1}$. Then

$$0 = \log(\mathbf{1}) = \log(\mathbf{m}_\nu^{\mathrm{naive}}) = \int_H \log(\mathbf{x})\,\nu(\mathrm{d}\mathbf{x}) = \int_H \log(\mathbf{1}^{-1}\mathbf{x})\,\nu(\mathrm{d}\mathbf{x}).$$

Hence, by uniqueness of $\mathbf{m}_\nu$, $\mathbf{m}_\nu = \mathbf{1}$.

Assume $\mathbf{m}_\nu = \mathbf{1}$. Then

$$0 = \int_H \log(\mathbf{x})\, \nu(\mathrm{d}\mathbf{x}),$$

and hence $\mathbf{m}_\nu^{\text{naive}} = \exp(0) = \mathbf{1}$. □

3.3.2 Key lemma

The key idea in the proof of theorem 3.9 is that the BCH formula implies the following polynomial relations on the coefficients.

Let $x, y \in \mathfrak{g}_{\leq L}(\mathbb{R}^d)$ be as follows,

$$x = \sum_{j=1}^{B} u_j \mathcal{B}_j, \quad y = \sum_{j=1}^{B} v_j \mathcal{B}_j,$$

where the order of the chosen basis $\mathcal{B}$ of $\mathfrak{g}_{\leq L}(\mathbb{R}^d)$ respects the length. Then, by plugging x, y as X, Y in the BCH formula, we get

$$\exp(x)\exp(y) = \exp\left(\sum_{j=1}^{B} \left(u_j + v_j + p_j(u_1, \ldots, u_{j-1}, v_1, \ldots, v_{j-1})\right) \mathcal{B}_j\right),$$

where p_j are some polynomials that are globally defined for fixed L, d, and the basis $\mathcal{B}$.

Formally, let

$$R := \mathbb{R}[M_1, \ldots, M_B, C_1, \ldots, C_B]$$

be the polynomial algebra over $2B$ symbols and $\mathfrak{g}^{\mathsf{symb}} := \mathfrak{g}^{\mathsf{symb}}(\mathbf{1}, \ldots, \mathbf{d}) := \mathfrak{g}(R^d)$ be the free Lie algebra with coefficients taken from R. Its truncation $\mathfrak{g}^{\mathsf{symb}}_{\leq L}$ is a B-dimensional free R-module with L-truncated Lyndon basis $\mathcal{B}$. On the corresponding dual space

$$\hom(\mathfrak{g}^{\mathsf{symb}}_{\leq L}, R) = \{f : \mathfrak{g}^{\mathsf{symb}}_{\leq L} \to R \mid f \ R\text{-linear}\}$$

we use the dual basis $\mathcal{B}^*$ as in (3.6). We now apply the BCH formula as in the group law of Definition 3.3, but for elements from $\mathfrak{g}^{\mathsf{symb}}_{\leq L}$.

Lemma 3.14. *Consider the following elements of* $\mathfrak{g}^{\mathsf{symb}}_{\leq L}$,

$$X := \sum_{b=1}^{B} M_b \mathcal{B}_b, \quad Y := \sum_{b=1}^{B} C_b \mathcal{B}_b.$$

Then, for every $j \leq B$, there exist (uniquely determined) $p_j \in \mathbb{R}[M_1, \ldots, M_{j-1}, C_1, \ldots, C_{j-1}]$ such that

$$\mathcal{B}_j^* \circ \mathsf{BCH}(X, Y) = M_j + C_j + p_j. \tag{3.8}$$

Proof. The Lyndon words are sorted by length, and each Lie bracket strictly increases the depth of nested Lie brackets. Therefore, together with Theorem 3.1, we have

$$\mathcal{B}_j^* \circ \mathsf{BCH}(X, Y) = M_j + C_j + \mathcal{B}_j^* \left(\mathsf{BCH}(X_{<j}, Y_{<j}) - (M_j + C_j)\mathcal{B}_j\right) \tag{3.9}$$

for all $j \le B$, where

$$X_{<j} := \sum_{b=1}^{j-1} M_b \mathcal{B}_b, \quad Y_{<j} := \sum_{b=1}^{j-1} C_b \mathcal{B}_b.$$

The claim follows with $p_j := \mathcal{B}_j^* \left(\mathsf{BCH}(X_{<j}, Y_{<j}) - (M_j + C_j)\mathcal{B}_j \right)$. □

Example 3.15. *In the setting of Example 3.2 with $L = 3$ and $d = 2$,*

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{2}C_2M_1 - \frac{1}{2}C_1M_2 \\ -\frac{1}{12}C_1C_2M_1 + \frac{1}{12}C_2M_1^2 + \frac{1}{12}C_1^2M_2 - \frac{1}{12}C_1M_1M_2 + \frac{1}{2}C_3M_1 - \frac{1}{2}C_1M_3 \\ \frac{1}{12}C_2^2M_1 - \frac{1}{12}C_1C_2M_2 - \frac{1}{12}C_2M_1M_2 + \frac{1}{12}C_1M_2^2 - \frac{1}{2}C_3M_2 + \frac{1}{2}C_2M_3 \end{bmatrix} \in \mathbb{R}^5.$$

3.3.3 Proof of the main theorem

Proof of Theorem 3.9. Let $\mathcal{B}$ denote the L-truncated Lyndon basis. Assume there is a group element

$$\mathbf{m} = \exp\left(\sum_{j=1}^{B} m_j \mathcal{B}_j \right)$$

which satisfies

$$\begin{aligned} 0 &= \int_H \log(\mathbf{m}^{-1}\mathbf{x})\, \nu(\mathrm{d}\mathbf{x}) \\ &= \int_H \log\left(\exp\left(-\sum_{j=1}^{B} m_j \mathcal{B}_j \right) \exp\left(\sum_{j=1}^{B} c_j^{(\mathbf{x})} \mathcal{B}_j \right) \right) \nu(\mathrm{d}\mathbf{x}) \end{aligned}$$

with suitable coefficients $c_j^{(\mathbf{x})}$ for $\mathbf{x} \in G$ and $1 \le j \le B$. Then via Theorem 3.1,

$$\begin{aligned} 0 &= \int_H \log \circ \exp\left(\sum_{j=1}^{B} \left(p_j(-m_1, \dots, -m_{j-1}, c_1^{(\mathbf{x})}, \dots, c_{j-1}^{(\mathbf{x})}) - m_j + c_j^{(\mathbf{x})} \right) \mathcal{B}_j \right) \nu(\mathrm{d}\mathbf{x}) \\ &= \sum_{j=1}^{B} \left(\int_H \left(p_j(-m_1, \dots, -m_{j-1}, c_1^{(\mathbf{x})}, \dots, c_{j-1}^{(\mathbf{x})}) - m_j + c_j^{(\mathbf{x})} \right) \nu(\mathrm{d}\mathbf{x}) \right) \mathcal{B}_j \end{aligned}$$

with polynomials p_j according to Lemma 3.14. Since $\mathcal{B}$ is an $\mathbb{R}$-basis, we obtain

$$m_j = \int_H p_j(-m_1, \dots, -m_{j-1}, c_1^{(\mathbf{x})}, \dots, c_{j-1}^{(\mathbf{x})}) + c_j^{(\mathbf{x})}\, \nu(\mathrm{d}\mathbf{x}), \tag{3.10}$$

and thus iteratively the components of m.[5] Hence, $\mathbf{m}$ is unique.

It is immediate to see that *defining* $\mathbf{m}$ via (3.10) also yields existence. □

[5] This integral is well-defined, by the integrability assumption.

A very common case is when the measure ν is discrete, i.e., it is supported on N points with weights $w^{(i)}$, where

$$\nu(\{\mathbf{x}^{(1)}\}) = w^{(1)}, \ \cdots, \ \nu(\{\mathbf{x}^{(N)}\}) = w^{(N)}, \quad \sum_{i=1}^{N} w^{(i)} = 1. \tag{3.11}$$

In this case, the following corollary provides an algorithm for iterative computation.

Corollary 3.16. *For the discrete measure (3.11), where the points have expansions*

$$\mathbf{x}^{(i)} = \exp\left(\sum_{j=1}^{B} c_j^{(i)} \mathcal{B}_j\right), \quad c^{(i)} \in \mathbb{R}^B, \quad 1 \le i \le N,$$

the coefficient vector m of the mean according to Definition 3.7 can be computed recursively using the closed formula

$$m_j = \sum_{i=1}^{N} w^{(i)} \left(c_j^{(i)} + p_j(-m_1, \ldots, -m_{j-1}, c_0^{(i)}, \ldots, c_{j-1}^{(i)})\right) \tag{3.12}$$

for every $j \le B$, see Algorithm 1.

Proof. Due to Lemma 3.5,

$$\begin{aligned} 0 &= \sum_{i=1}^{N} w^{(i)} \log(\mathbf{m}^{-1}\mathbf{x}^{(i)}) = \sum_{i=1}^{N} w^{(i)} \sum_{j=1}^{B} ((-m) \star c^{(i)})_j \mathcal{B}_j \\ &= \sum_{j=1}^{B} \sum_{i=1}^{N} w^{(i)} \left(-m_j + c_j^{(i)} + p_j(-m_1, \ldots, -m_{j-1}, c_1^{(i)}, \ldots, c_{j-1}^{(i)})\right) \mathcal{B}_j \end{aligned}$$

with polynomials $p_j \in \mathbb{R}[M_1, \ldots, M_{j-1}, C_1, \ldots, C_{j-1}]$ according to Lemma 3.14, and where $p_1 = \cdots = p_d = 0$.

□

Algorithm 1: GROUP MEAN

Input: A set of N coefficient vectors $x^{(i)} \in \mathbb{R}^B$ for group elements $(\mathbf{x}^{(i)})_{1 \le i \le N}$
Output: The coefficients $m \in \mathbb{R}^B$ for the group mean $\mathbf{m}$

1 Precompute polynomials $(p_j)_{1 \le j \le B}$
2 **for** $j = 1, \ldots, B$ **do**
3 Compute m_j using Equation (3.12)
4 **return** $m = \begin{bmatrix} m_1 & \ldots & m_B \end{bmatrix}$

Example 3.17. *Continuing Example 3.15, and assuming $w^{(i)} = \frac{1}{N}$ for simplicity,*

$$m_1 = \frac{1}{N}\sum_{i=1}^{N} c_1^{(i)}, \quad m_2 = \frac{1}{N}\sum_{i=1}^{N} c_2^{(i)}, \quad m_3 = \frac{1}{N}\sum_{i=1}^{N}\left(-\frac{1}{2}c_2^{(i)}m_1 + \frac{1}{2}c_1^{(i)}m_2 + c_3^{(i)}\right),$$

$$m_4 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{12}c_1^{(i)}c_2^{(i)}m_1 + \frac{1}{12}c_2^{(i)}m_1^2 - \frac{1}{12}c_1^{(i)}c_1^{(i)}m_2 - \frac{1}{12}c_1^{(i)}m_1m_2 - \frac{1}{2}c_3^{(i)}m_1 + \frac{1}{2}c_1^{(i)}m_3 + c_4^{(i)}\right),$$

$$m_5 = \frac{1}{N}\sum_{i=1}^{N}\left(-\frac{1}{12}c_2^{(i)}c_2^{(i)}m_1 + \frac{1}{12}c_1^{(i)}c_2^{(i)}m_2 - \frac{1}{12}c_2^{(i)}m_1m_2 + \frac{1}{12}c_1^{(i)}m_2^2 + \frac{1}{2}c_3^{(i)}m_2 - \frac{1}{2}c_2^{(i)}m_3 + c_5^{(i)}\right),$$

which gives all the steps of Algorithm 1.

Given a truncation level L and the dimension d of the inputs, we obtain $B = B_{L,d}$ according to Equation (3.5). Then the computational complexity of Algorithm 1 is given by the following lemma.

Lemma 3.18. *There is a Q_L such that for all d and $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)} \in G_{\leq L}(\mathbb{R}^d)$ we can compute the empirical group mean according to Definition 3.7 in less than*

$$NB_{L,d}\,(1 + LQ_L)$$

basic operations. The storage complexity is trivial, i.e., we only have to store the $B_{L,d}$ coefficients m_j during runtime.

Proof. For every $1 \leq j \leq B = B_{L,d}$ let $Q_{L,d,j}$ denote the number of terms in p_j and set

$$Q_L := \max_{\substack{1\leq j\leq B\\ 1\leq i\leq L}} Q_{L,i,j} \leq \dim_{\mathbb{R}}(\mathbb{R}[x_1,\dots,x_{2L^{L+1}}]_{\deg\leq L}) = \binom{2L^{L+1}+L}{2L^{L+1}} \tag{3.13}$$

where the inequality holds with $\deg(p_j) \leq L$ and $B_{L,i} \leq L^{L+1}$. Now assume $d > L$. Every Lyndon word $w = v_1 \dots v_\ell$, with $v_t \in \{1, \dots, d\}$ and $\ell \leq L$, can be written as $\varphi(v_1)\dots\varphi(v_\ell)$ with $\varphi(v_t) \in \{1, \dots, L\}$, and where φ is an homomorphism of monoids which preserves the order of letters. Clearly w and $\varphi(w)$ lead to polynomials with the same number of summands, bounded by Q_L. Hence, we can evaluate (3.12) in

$$N\sum_{1\leq j\leq B} 1 + \deg(p_j)Q_{L,d,j} \leq NB(1 + LQ_L)$$

basic operations due to (3.13). □

Remark 3.19. *The family of polynomials $(p_j)_{1\leq j\leq B}$ based on Corollary 3.16 can be precomputed symbolically, e.g., with* `SageMath` *using* `sage.algebras.lie_algebras.bch` *that allows a polynomial base ring through the class*

`sage.algebras.lie_algebras.lie_algebra.LieAlgebra`

and thus a remarkably light-weighted implementation of Equation (3.8). With the procedure `monomial_coefficient` *applied on our truncated Lie series, we can apply the dual basis to obtain p_j for every $1 \leq j \leq B$.*

Remark 3.20. *In Proposition 3.31 (Section 3.4) we provide a complexity analysis for an alternative algorithm. Clearly $B_{L,d} \in O(d^L)$ with (3.5). In the limit the computational complexity of Lemma 3.18 is thus not better than in Proposition 3.31.*

In the following subsection we use an antisymmetrized BCH formula to improve the rough bound (3.13) of Q_L for the practically relevant cases $L = 2, 3, 4, 5$ by 0, 3, 9 and 43, respectively. In such cases, the number of required operations given by Lemma 3.18 becomes comparable with the cost of computing the naive mean in $\mathbb{R}^{B_{L,d}}$.

3.3.4 Reducing the number of terms with an antisymmetrized BCH formula

In this section, we delve into the cancelations that emerge from the BCH formula and terms linear in C_j. These explicit cancelations result in polynomials with a comparable number of terms as in the previous section.

For this, we use the BCH formula with graded components

$$\log(\exp(X)\exp(Y)) = \mathsf{BCH}(X,Y) = \sum_{k=1}^{\infty} \mathsf{BCH}_k(X,Y) \tag{3.14}$$

expanded in the Lyndon basis ([CM09, Table 2, page 12] or [CM]) of the free Lie algebra over the two-letter alphabet $\{X,Y\}$. The first 6 components $\mathsf{BCH}_k(X,Y)$ are given by:

$$\begin{aligned}
\mathsf{BCH}_1(X,Y) &= X+Y, & \mathsf{BCH}_2(X,Y) &= \frac{1}{2}[X,Y],\\
\mathsf{BCH}_3(X,Y) &= \frac{1}{12}[[X,Y],Y] + \frac{1}{12}[X,[X,Y]], & \mathsf{BCH}_4(X,Y) &= \frac{1}{24}[X,[[X,Y],Y]],\\
\mathsf{BCH}_5(X,Y) &= -\frac{1}{720}[[[[X,Y],Y],Y],Y] + \frac{1}{360}[[X,[X,Y]],[X,Y]]\\
&\quad + \frac{1}{120}[[X,Y],[[X,Y],Y]] + \frac{1}{180}[X,[[[X,Y],Y],Y]]\\
&\quad + \frac{1}{180}[X,[X,[[X,Y],Y]]] - \frac{1}{720}[X,[X,[X,[X,Y]]]],\\
\mathsf{BCH}_6(X,Y) &= -\frac{1}{1440}[X,[[[[X,Y],Y],Y],Y]] + \frac{1}{720}[X,[[X,[X,Y]],[X,Y]]]\\
&\quad + \frac{1}{240}[X,[[X,Y],[[X,Y],Y]]] + \frac{1}{360}[X,[X,[[[X,Y],Y],Y]]]\\
&\quad - \frac{1}{1440}[X,[X,[X,[[X,Y],Y]]]].
\end{aligned}$$

Remark 3.21 ([CM09, Section IV.C]). *For even k, all the terms in $\mathsf{BCH}_k(X,Y)$ have necessarily the form $[X,Z]$, where Z goes over all Lyndon basis elements of degree $k-1$, except for the element $[X,[X,\dots,[X,Y]]]$ which does not appear.*

For such Lyndon words as in Remark 3.21, we define a map

$$\mathsf{D}_X([X,Z]) := Z.$$

This helps us to introduce an asymmetrized BCH formula $\mathsf{aBCH}(X,Y)$ as follows.

Lemma 3.22. *The Lie series*

$$\mathsf{aBCH}(X,Y) := \sum_{k=1}^{\infty} \mathsf{aBCH}_k(X,Y) := \mathsf{D}_X(\mathsf{BCH}(X,Y) - \mathsf{BCH}(Y,X)) \tag{3.15}$$

is well-defined and has graded components

$$\mathsf{aBCH}_k(X,Y) = \begin{cases} 0, & \text{if } k \text{ is even},\\ 2\,\mathsf{D}_X(\mathsf{BCH}_{k+1}(X,Y)), & \text{if } k \text{ is odd}; \end{cases} \tag{3.16}$$

for example, the first 3 nonzero graded components are given by

$$\mathsf{aBCH}_1(X,Y) = Y, \quad \mathsf{aBCH}_3(X,Y)) = \frac{1}{12}[[X,Y],Y],$$
$$\mathsf{aBCH}_5(X,Y) = -\frac{1}{720}[[[[X,Y],Y],Y],Y] + \frac{1}{360}[[X,[X,Y]],[X,Y]]$$
$$+ \frac{1}{120}[[X,Y],[[X,Y],Y]] + \frac{1}{180}[X,[[[X,Y],Y],Y]] - \frac{1}{720}[X,[X,[[X,Y],Y]]].$$

Proof of Lemma 3.22. From (3.3), we get that

$$\mathsf{BCH}(X,Y) - \mathsf{BCH}(Y,X) = \mathsf{BCH}(X,Y) + \mathsf{BCH}(-X,-Y).$$

Splitting the equation by degrees and using the fact that

$$\mathsf{BCH}_{k+1}(-X,-Y) = (-1)^{k+1}\mathsf{BCH}_{k+1}(X,Y),$$

we get

$$\mathsf{BCH}_{k+1}(X,Y) - \mathsf{BCH}_{k+1}(Y,X) = \begin{cases} 0, & k \text{ is even}, \\ 2\mathsf{BCH}_{k+1}(X,Y), & k \text{ is odd}. \end{cases}$$

Finally, by Remark 3.21 the operator D_X can be applied to non-zero terms, which proves Equation (3.16). □

As we will show next, the asymmetrized version of the BCH formula can be used in Lemma 3.14.

Theorem 3.23. *i. For the discrete measure supported on $\mathbf{x}^{(i)}$, $i \in \{1,\dots,N\}$ (as in Corollary 3.16), let $C^{(i)} = \log(\mathbf{x}^{(\mathbf{i})})$. Then the logarithm of the barycenter $M = \log(\mathbf{m})$ satisfies*

$$M = \sum_{i=1}^{N} w^{(i)} \mathsf{aBCH}(-M, C^{(i)}). \tag{3.17}$$

ii. Let m_j and $c_j^{(i)}$, $j \in \{1,\dots,B\}$ be the coordinates of $\mathbf{m}$ and $\mathbf{x}^{(i)}$ in the basis $\mathcal{B}$. Then m_j can be computed recursively:

$$m_j = \sum_{i=1}^{N} w^{(i)} \left(c_j^{(i)} + \widetilde{r}_j(m_1,\dots,m_{j-1},c_1^{(i)},\dots,c_{j-1}^{(i)}) \right). \tag{3.18}$$

where the polynomial $\widetilde{r}_j$ is defined for X and Y, expanded as in Lemma 3.14, as follows

$$\widetilde{r}_j := \mathcal{B}_j^* \left(\sum_{\substack{k=3,\dots,L \\ k \text{ odd}}} \mathsf{aBCH}_k(-X,Y) \right) \in R.$$

Remark 3.24. *In Section 3.4.4, we show that the numbers of terms $Q_{L,L}$ (0, 3, 9 and 43) for $L = 2,3,4$ and 5, respectively, coincide with the number of terms given by an alternative calculation in the ambient space that uses the antisymmetrized BCH. This implies an upper bound for the number of terms in one of the polynomials $\widetilde{r}_j$ and we conjecture that this bound is valid for other polynomials as well.*

The proof of Theorem 3.23 relies on some useful facts from [Reu93] which we recall below. We denote $\mathrm{ad}_X(Z) := [X, Z]$ and the "power" of ad_X

$$\mathrm{ad}_X^k(Y) = \overbrace{[X, [X, \dots, [X}^{k \text{ times}}, Y] \dots]].$$

The following lemma is key for this subsection.

Lemma 3.25 ([Reu93, page 80]). *We expand the BCH formula as follows:*

$$\mathsf{BCH}(X, Y) = X + \underbrace{H_{1,X}(Y)}_{\text{linear in } Y} + \quad (\text{order } \geq 2 \text{ terms in } Y). \tag{3.19}$$

Then the linear term can be expressed as

$$H_{1,X}(Y) = Y + \frac{1}{2}[X, Y] + \sum_{n=1}^{\infty} \frac{b_{2n}}{(2n)!} \mathrm{ad}_X^{2n}(Y),$$

where b_{2n} are Bernoulli numbers. Alternatively, we can write

$$H_{1,X}(Y) = g(\mathrm{ad}_X)(Y), \quad \text{for} \quad g(t) = \frac{t}{1 - e^{-t}},$$

where $g(\mathrm{ad}_X)$ means substitution of t^k with ad_X^k in the power series expansion

$$g(t) = 1 + g_1 t + g_2 t^2 + \cdots.$$

Remark 3.26. *i. With $\mathrm{ad}_X(X) = 0$, $H_{1,X}$ leaves X invariant, i.e., $H_{1,X}(X) = X$.*

ii. $H_{1,X}$ is invertible, and its inverse is $H_{1,X}^{-1}(Z) = f(\mathrm{ad}_X)(Z)$, where

$$f(t) = \frac{1}{g(t)} = \frac{1 - e^{-t}}{t}.$$

Proof of Theorem 3.23. Part 2 follows directly from part 1, therefore we are going to prove part 1. First, we show that

$$\mathsf{aBCH}(X, Y) = H_{1,X}^{-1}((\mathsf{BCH}(X, Y) - X) \tag{3.20}$$

Indeed, by Remark 3.26, we have

$$\begin{aligned} H_{1,X}^{-1}((\mathsf{BCH}(X, Y) - X) &= \frac{1 - e^{-\mathrm{ad}_X}}{\mathrm{ad}_X}(\log(e^X e^Y) - X) \\ &= \mathsf{D}_X \left(\log(e^X e^Y) - X - e^{-\mathrm{ad}_X}(\log(e^X e^Y) - X) \right) \\ &= \mathsf{D}_X \left(\log(e^X e^Y) - X - (\log(e^Y e^X) - X) \right) = \mathsf{D}_X(\mathsf{BCH}(X, Y) - \mathsf{BCH}(Y, X)), \end{aligned}$$

where for the last but one equality we use the well-known property

$$e^{\mathrm{ad}_A}(Z) = e^A Z e^{-A},$$

to get

$$e^{-\mathrm{ad}_X}(\log(e^X e^Y) - X) = e^{-X} \log(e^X e^Y) e^X - e^{-X} X e^X = \log(e^Y e^X) - X. \tag{3.21}$$

Recall that M is the logarithm of the barycenter if and only if

$$\sum_{i=1}^{N} w^{(i)} \mathsf{BCH}(-M, C^{(i)}) = 0. \tag{3.22}$$

Thus, using that $\sum_{i=1}^{N} w^{(i)} = 1$, we can add M on both sides of the equation:

$$M = \sum_{i=1}^{N} w^{(i)} \left(\mathsf{BCH}(-M, C^{(i)}) + M \right) \tag{3.23}$$

and using that $H_{1,-M}$ is linear and $H_{1,-M}(M) = M$, this gives

$$M = \sum_{i=1}^{N} w^{(i)} H_{1,-M}^{-1} \left(\mathsf{BCH}(-M, C^{(i)}) + M \right). \tag{3.24}$$

Now, by injecting Equation (3.20), we obtain Equation (3.17) and this concludes the proof. □

3.3.5 Recursive updates of group means

The following formula allows an update of a computed mean if a new data point is inserted. Use Taylor expansion and obtain the following online update formula.

Lemma 3.27. *Let* $\mathbf{m}'$ *be the group mean of* $N-1$ *points* $(\mathbf{x}^{(i)})_{1 \le i \le N-1}$ *and* $\mathbf{m}$ *be the group mean of those same* $N-1$ *points plus an additional incoming point* $\mathbf{x}^{(N)}$*. The computation of* $\mathbf{m}$ *can be derived from the value of* $\mathbf{m}'$ *via*

$$\begin{aligned} m_j = \frac{1}{N} \tilde{p}_j(m_1, \dots, m_{j-1}, c_1^{(N)}, \dots, c_j^{(N)}) + \frac{N-1}{N} m'_j \\ + \frac{1}{N} \sum_{i=1}^{N-1} \sum_{1 \le |\alpha| \le j-1} \frac{(\Delta m)^\alpha}{\alpha!} \frac{\partial^\alpha \tilde{p}_j(m'_1, \dots, m'_{j-1}, c_1^{(i)}, \dots, c_j^{(i)})}{(\partial m)^\alpha}, \end{aligned} \tag{3.25}$$

where $\tilde{p}_j := p_j + C_j$ *with* p_j *due to Lemma 3.14 and* $\Delta m := m - m'$*. Equation (3.25) requires to compute at most* $\binom{j-1+\deg(p_j)}{j-1} = \binom{j-1+\deg(p_j)}{\deg(p_j)}$ *partial derivatives.*

Proof. Using Equation (3.12) and denoting $\tilde{p}_j := p_j + C_j$, we have

$$\begin{aligned} m_j &= \frac{1}{N} \sum_{i=1}^{N} \tilde{p}_j(m_1, \dots, m_{j-1}, c_1^{(i)}, \dots, c_j^{(i)}) \\ &= \frac{1}{N} \tilde{p}_j(m_1, \dots, m_{j-1}, c_1^{(N)}, \dots, c_j^{(N)}) + \frac{1}{N} \sum_{i=1}^{N-1} \tilde{p}_j(m_1, \dots, m_{j-1}, c_1^{(i)}, \dots, c_j^{(i)}). \end{aligned}$$

Now apply Taylor expansion to the term inside the sum. We have

$$\begin{aligned} \tilde{p}_j(m'_1 + (\Delta m)_1, \dots, m'_{j-1} + (\Delta m)_{j-1}, c_1^{(i)}, \dots, c_j^{(i)}) = \tilde{p}_j(m'_1, \dots, m'_{j-1}, c_1^{(i)}, \dots, c_j^{(i)}) \\ + \sum_{1 \le |\alpha| \le j-1} \frac{(\Delta m)^\alpha}{\alpha!} \frac{\partial^\alpha \tilde{p}_j(m'_1, \dots, m'_{j-1}, c_1^{(i)}, \dots, c_j^{(i)})}{(\partial m)^\alpha}. \end{aligned}$$

Inject that back into m_j to obtain Equation (3.25). □

Remark 3.28. *The usefulness of the recursive update is mostly theoretical since that with Lemma 3.27 we would still need to perform at least $N-1$ polynomial evaluations.*

Note that Lemma 3.27 can also be applied to Equation (3.12). As an example, for $L=4$ and $d=2$, we have $B_{4,2}=8$ and $Q_{4,2}=3$. That is, we would have $\sum_{j=1}^{8}\binom{j-1+3}{3}=330$ partial derivatives to compute to update $\mathbf{m}'$ to $\mathbf{m}$.

3.4 Algorithm using updates in the ambient space

In this section, we leverage the embedding of the space of signatures in the truncated tensor algebra $T_{\leq L}(\mathbb{R}^d)$. We develop an iterative procedure for obtaining the successive levels of the group mean in terms of lower levels, that is $\mathbf{m}_K \in (\mathbb{R}^d)^{\otimes K}$ as a mapping of $(\mathbf{m}_0,\dots,\mathbf{m}_{K-1})$ and the data. This mapping involves two polynomial maps $\mathbf{p}_K$ and $\mathbf{q}_K$ that are obtained from the truncated version of the logarithm mapping in $T_{\leq L}(\mathbb{R}^d)$.

Notably, this approach is applicable to any dimension d. It also does not need to precompute the maps $\mathbf{p}_K$ and $\mathbf{q}_K$ (unlike in the algorithms in Section 3.3), at the expense of dealing with higher-dimensional tensor spaces. The corresponding algorithm essentially relies on the computation of tensor products and we provide a Python implementation.

First, we present the result and the corresponding algorithm. Then, examples illustrate the proof. Finally, we look at time and memory complexities.

Notation. Throughout this section, lower indices of elements of tensor algebra $T_{\leq L}(\mathbb{R}^d)$ denote levels: $\mathbf{g}=(\mathbf{g}_0,\dots,\mathbf{g}_L)$, $\mathbf{g}_K\in(\mathbb{R}^d)^{\otimes K}$. Note that $\mathbf{g}_0=1$ for all $\mathbf{g}\in G_{\leq L}(\mathbb{R}^d)$. Finally, in this notation, the identity element is $\mathbf{1}=(1,0,\dots,0)$.

3.4.1 Main result

We have a truncated version (by nilpotency) of the inverse and of the logarithm mapping presented in Section 2.2.3. For any $(\mathbf{1}+\mathbf{g})\in G_{\leq L}(\mathbb{R}^d)$,

$$(\mathbf{1}+\mathbf{g})^{-1}:=\sum_{k=0}^{L}(-1)^k\mathbf{g}^k,\qquad \log(\mathbf{1}+\mathbf{g}):=\sum_{k=1}^{L}\frac{(-1)^{k+1}}{k}\mathbf{g}^k. \tag{3.26}$$

Recall that the group operation in tensor algebra for

$$\mathbf{a}=(\mathbf{a}_0,\mathbf{a}_1,\dots,\mathbf{a}_L),\quad \mathbf{x}=(\mathbf{x}_0,\mathbf{x}_1,\dots,\mathbf{x}_L)$$

is given by

$$(\mathbf{ax})_L=\sum_{\ell=0}^{L}\mathbf{a}_\ell\otimes\mathbf{x}_{L-\ell}. \tag{3.27}$$

With this notation, we can prove the following result.

Proposition 3.29. *Let $\{X^{(1)},\dots,X^{(N)}\}$ be a batch of N d-variate time series and $\mathbf{x}^{(i)}:=\mathbf{S}_{\leq L}(X^{(i)})$. Let $\{w^{(1)},\dots,w^{(N)}\}$ be a set of real values such that $\sum_{i=1}^{N}w^{(i)}=1$. Denote $\mathbf{m}$ the group mean of the dataset with weights $w^{(i)}$ and $\mathbf{a}:=\mathbf{m}^{-1}$. Then we have that*

$$\mathbf{a}_1=-\sum_{i=1}^{N}w^{(i)}(\mathbf{x}^{(i)})_1 \tag{3.28}$$

and for any $K = 2, \dots, L$,

$$\mathbf{a}_K = -\sum_{i=1}^{N} w^{(i)} \Bigg(\mathbf{q}_K \left(\mathbf{a}_0, \dots, \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \dots, (\mathbf{x}^{(i)})_K \right) + \mathbf{p}_K \left(\mathbf{a}_0, \dots, \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \dots, (\mathbf{x}^{(i)})_{K-1} \right) \Bigg) \tag{3.29}$$

where $\mathbf{p}_K$ *and* $\mathbf{q}_K$ *are (noncommutative) polynomial maps, whose definitions are given in the proof (see Equations* (3.33) *and* (3.35)*).*

Note that the computation on the right-hand side solely relies on the values of $\mathbf{a}_0, \dots, \mathbf{a}_{K-1}$ and the input data $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$. In other words, Proposition 3.29 enables an iterative approach for calculating the successive values of $\mathbf{a}_K$ for increasing values of K.

Before proving Proposition 3.29, we begin with a preliminary consideration.

Lemma 3.30. *For any* $0 \leq K < j$ *and* $1 \leq i \leq N$*, denote* $\mathbf{v}^{(i)} := \mathbf{a}\mathbf{x}^{(i)} - \mathbf{1}$ *and* $\mathbf{v}^{(i,j)} := (\mathbf{v}^{(i)})^j$ *be the* j*-th group power of* $\mathbf{v}^{(i)}$*. Then,*

$$(\mathbf{v}^{(i,j)})_K = 0. \tag{3.30}$$

Proof. By induction on j. We have $(\mathbf{v}^{(i)})_0 = (\mathbf{a}\mathbf{x}^{(i)})_0 - 1 = 0$. Now, suppose that for a fixed j we have $(\mathbf{v}^{(i,j-1)})_0 = \cdots = (\mathbf{v}^{(i,j-1)})_{j-2} = 0$.

If $K < j - 1$ then, from Equation (3.27)

$$(\mathbf{v}^{(i,j)})_K = (\mathbf{v}^{(i)}\mathbf{v}^{(i,j-1)})_K = \sum_{k=0}^{K} (\mathbf{v}^{(i)})_{K-k} \otimes (\mathbf{v}^{(i,j-1)})_k = 0$$

using the induction hypothesis.

If $K = j - 1$ then

$$(\mathbf{v}^{(i,j)})_K = \sum_{k=0}^{K} (\mathbf{v}^{(i)})_{K-k} \otimes (\mathbf{v}^{(i,j-1)})_k = (\mathbf{v}^{(i)})_0 \otimes (\mathbf{v}^{(i,j-1)})_{j-1} = 0$$

using the induction hypothesis and that $(\mathbf{v}^{(i)})_0 = 0$. □

Proof of Proposition 3.29. By Definition 3.7, the group mean $\mathbf{m}$ with weights $w^{(i)}$ verifies

$$0 = \sum_{i=1}^{N} w^{(i)} \log(\mathbf{m}^{-1}\mathbf{x}^{(i)}) = \sum_{i=1}^{N} \left(w^{(i)}\mathbf{v}^{(i)} + w^{(i)} \sum_{j=2}^{L} \frac{(-1)^{j+1}}{j} \mathbf{v}^{(i,j)} \right) \tag{3.31}$$

where the last equality is obtained from Equation (3.26) (definition of the logarithm) and the fact that $\mathbf{m}^{-1}\mathbf{x}^{(i)} = \mathbf{a}\mathbf{x}^{(i)} = \mathbf{1} + \mathbf{v}^{(i)}$. Denote $\mathbf{z}$ the last right-hand side of Equation (3.31). Using Lemma 3.30, when $\mathbf{z}$ is evaluated at level 1, we have

$$\mathbf{z}_1 = \sum_{i=1}^{N} w^{(i)} (\mathbf{v}^{(i)})_1 = \sum_{i=1}^{N} w^{(i)} (\mathbf{a}_1 + (\mathbf{x}^{(i)})_1) = 0,$$

therefore $\mathbf{a}_1 = -\sum_{i=1}^{N} w^{(i)}\mathbf{x}^{(i)}$. Using Lemma 3.30, for any $K = 2, \dots, L$, when z is evaluated at level K, the sum stops at K:

$$\mathbf{z}_K = \sum_{i=1}^{N} \left(w^{(i)}(\mathbf{v}^{(i)})_K + w^{(i)} \sum_{j=2}^{K} \frac{(-1)^{j+1}}{j} (\mathbf{v}^{(i,j)})_K \right) = 0. \tag{3.32}$$

Now remark that $\sum_{j=2}^{K} \frac{(-1)^{j+1}}{j}(\mathbf{v}^{(i,j)})_K$ depends only on $(\mathbf{v}^{(i)})_1, \dots, (\mathbf{v}^{(i)})_{K-1}$. Therefore, we can denote

$$p_K \left(\mathbf{a}_0, \dots, \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \dots, (\mathbf{x}^{(i)})_{K-1} \right) := \sum_{j=2}^{K} \frac{(-1)^{j+1}}{j} (\mathbf{v}^{(i,j)})_K. \tag{3.33}$$

From definition of $\mathbf{v}^{(i)}$ and from (3.27)

$$(\mathbf{v}^{(i)})_K = (\mathbf{a}\mathbf{x}^{(i)})_K = \mathbf{a}_K + \mathbf{q}_K \left(\mathbf{a}_0, \dots \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \dots, (\mathbf{x}^{(i)})_K \right), \tag{3.34}$$

where

$$\mathbf{q}_K \left(\mathbf{a}_0, \dots \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \dots, (\mathbf{x}^{(i)})_K \right) := \sum_{k=0}^{K-1} \mathbf{a}_k \otimes (\mathbf{x}^{(i)})_{K-k}. \tag{3.35}$$

Injecting Equations (3.33)–(3.34) into Equation (3.32) gives (3.29). □

3.4.2 Algorithm

Let $\alpha_L := \dim T_{\leq L}(\mathbb{R}^d) = \sum_{i=0}^{L} d^i$, for any integer $L \geq 1$. For convenience, the group elements of $G_{\leq L}(\mathbb{R}^d) \subset T_{\leq L}(\mathbb{R}^d)$ are implemented as a long array of size α_L. The procedure is detailed in Algorithm 2 with corresponding nomenclature shown in Table 3.1. Then, we derive the corresponding time and space complexities in Proposition 3.31.

Symbol	Meaning	Tensor order	Size
$\mathbf{x}^{(i)}$	Signatures of input time series $X^{(i)}$	2	$N\alpha_L$
$\mathbf{m}$	Group mean	1	α_L
$\mathbf{a}$	Group inverse of $\mathbf{m}$	1	α_L
$\mathbf{p}^{(i)}$	Evaluation of Equation (3.33) for $\mathbf{x}^{(i)}$	2	$N\alpha_L$
$\mathbf{q}^{(i)}$	Evaluation of Equation (3.35) for $\mathbf{x}^{(i)}$	2	$N\alpha_L$
$\mathbf{v}^{(i)}$	$\mathbf{a}\mathbf{x}^{(i)} - \mathsf{e}$	2	$N\alpha_L$
$\mathbf{v}^{(i,j)}$	Group powers $(\mathbf{v}^{(i)})^j$ for $j = 2, \dots, L$	3	$N(L-1)\alpha_L$
$w^{(i)}$	Weights of the group mean	1	N

TABLE 3.1: Nomenclature for tensors in Algorithm 2. Index i varies between 1 and N.

Proposition 3.31. *Time complexity of Algorithm 2 is $O(Nd^{L-1}L)$ and storage complexity is $O(Nd^L L)$.*

Proof. For any $K = 2, \dots, L$, computation of $\mathbf{q}_K$ in Equation (3.29) is $O((K-1)d^K)$, given Equation (3.35). Computation of $\mathbf{p}_K$ in Equation (3.29) is $O(d^{L-1}L)$ for a fixed i, since computation of $(\mathbf{v}^{(i,k)})_\ell$ is $O(d^{L-1}L)$. Indeed, let us fix $1 \leq k \leq \ell \leq L$ ($k > \ell$ is covered in Lemma 3.30). The computation of $(\mathbf{v}^{(i,k)})_\ell$ requires $(\ell - k + 1)d^{\ell-k+1}$

Algorithm 2: GROUP MEAN USING UPDATES IN AMBIENT SPACE

Input: A batch of N signatures $\mathbf{x}^{(i)} := \mathbf{S}_{\leq L}(X^{(i)})$ and N weights $w^{(i)}$.
Output: An array $\mathbf{m}$ with the values of the group mean.

Initialize $\mathbf{a}_0 \leftarrow 1$, $\mathbf{a}_1 \leftarrow -\sum_{i=1}^{N} w^{(i)}(\mathbf{x}^{(i)})_1$ and $\mathbf{q}_1^{(i)} \leftarrow (\mathbf{x}^{(i)})_1$.
for $K = 2, \ldots, L$ **do**
  for $i = 1, \ldots, N$ **do**
    $(\mathbf{v}^{(i)})_{K-1} \leftarrow \mathbf{a}_{K-1} + \mathbf{q}_{K-1}^{(i)}$
    $\mathbf{q}_K^{(i)} \leftarrow \mathbf{q}_K(\mathbf{a}_0, \ldots, \mathbf{a}_{K-1}, (\mathbf{x}^{(i)})_1, \ldots, (\mathbf{x}^{(i)})_K)$
    for $j = 2, \ldots, K$ **do**
      $(\mathbf{v}^{(i,j)})_K \leftarrow \sum_{k=j-1}^{K} (\mathbf{v}^{(i,j-1)})_k \otimes (\mathbf{v}^{(i)})_{K-k}$
    $\mathbf{p}_K^{(i)} \leftarrow \sum_{j=2}^{K} \frac{(-1)^{j+1}}{j} (\mathbf{v}^{(i,j)})_K$
  $\mathbf{a}_K \leftarrow -\sum_{i=1}^{N} w_i \left(\mathbf{q}_K^{(i)} + \mathbf{p}_K^{(i)}\right)$
$\mathbf{m} \leftarrow$ `GroupInv(a)` `// Computed by implementing Equation (3.26)`
return $\mathbf{m}$

operations using that

$$(\mathbf{v}^{(i,k)})_\ell = (\mathbf{v}^{(i)}\mathbf{v}^{(i,k-1)})_\ell = \sum_{j=1}^{\ell-(k-1)} (\mathbf{v}^{(i)})_j \otimes (\mathbf{v}^{(i,k-1)})_{\ell-(k-1)-j}$$

since $\mathbf{v}^{(i,k-1)}$ has $k-1$ leading zeros. Now, the computation for any $1 \leq k \leq \ell \leq L$ is $O(d^{L-1}L)$. To obtain the time complexity of Algorithm 2, we have to take into account the batch size N and we obtain $O(Nd^{L-1}L)$.

Regarding space complexity, we have to store in memory $\mathbf{v}^{(i,K)}$ for all observation indices $1 \leq i \leq N$ and powers $1 \leq K \leq L$. Thus, the storage complexity is $O(Nd^L L)$. □

Remark 3.32. *In practical applications such as the analysis of a set of time series, the computation of the signature must be taken into account, especially when benchmarking against other methods. Let $X : [a, b] \to \mathbb{R}^d$ be a linear process, observe that*

$$\mathbf{S}_{\leq L}(X) = \exp(X(b) - X(a)). \tag{3.36}$$

Now consider $X : [0, \tau] :\to \mathbb{R}^d$ to be a piecewise linear process where each piece is defined on intervals $[t, t+1]$ with t integer. Using Equation (3.36) and Chen's identity, the signature can be computed iteratively:

$$\mathbf{S}_{\leq L}(X) = \exp(X(2) - X(1)) \exp(X(3) - X(2)) \ldots \exp(X(\tau) - X(\tau - 1)). \tag{3.37}$$

The product operation Ae^Z is $O(d^L)$. Thus, the time complexity of the signature computation is $O(\tau d^L)$. Combining this with the complexity of Algorithm 2, the overall complexity of the approach is $O(Nd^L(\tau + L))$.

3.4.3 Examples

To have a better grasp of the idea behind the algorithm, we show here the computations for the first two levels $L = 1, 2$. As stated before, the computation stands for

any dimension d.

Example 3.33. *At level 1, starting from the value of* $\mathbf{a}_1 = -\sum_{i=1}^N w^{(i)}(\mathbf{x}^{(i)})_1$ *computed in the proof above and since for any* $\mathbf{g} \in G_{\leq L}(\mathbb{R}^d)$, $(\log \mathbf{g})_1 = \mathbf{g}_1$ *and Equation (3.26), we obtain*

$$(\log \mathbf{m})_1 = \sum_{i=1}^N w^{(i)}(\log \mathbf{x}^{(i)})_1. \tag{3.38}$$

Example 3.34. *At level 2, we have,*

$$\begin{aligned}
\mathbf{a}_2 &= -\sum_{i=1}^N w^{(i)} \left(\mathbf{q}_2\Big(\mathbf{a}_0, \mathbf{a}_1, (\mathbf{x}^{(i)})_1, (\mathbf{x}^{(i)})_2\Big) + \mathbf{p}_2\Big(\mathbf{a}_0, \mathbf{a}_1, (\mathbf{x}^{(i)})_1\Big)\right) \\
&= -\sum_{i=1}^N w^{(i)} \left((\mathbf{x}^{(i)})_2 + \mathbf{a}_1 \otimes (\mathbf{x}^{(i)})_1 - \frac{1}{2}(\mathbf{v}^{(i)})_1 \otimes (\mathbf{v}^{(i)})_1\right) \\
&= -\sum_{i=1}^N w^{(i)}(\mathbf{x}^{(i)})_2 - \mathbf{a}_1 \otimes \mathbf{a}_1 + \frac{1}{2}\sum_{i=1}^N w^{(i)}(\mathbf{a}_1 + (\mathbf{x}^{(i)})_1) \otimes (\mathbf{a}_1 + (\mathbf{x}^{(i)})_1) \\
&= \sum_{i=1}^N w^{(i)} \left(\frac{1}{2}(\mathbf{x}^{(i)})_1 \otimes (\mathbf{x}^{(i)})_1 - (\mathbf{x}^{(i)})_2\right) + \frac{1}{2}\mathbf{a}_1 \otimes \mathbf{a}_1,
\end{aligned}$$

where the last equality follows from (3.28). Using the fact that for any $\mathbf{g} \in G_{\leq L}(\mathbb{R}^d)$, $(\log \mathbf{g})_2 = \mathbf{g}_2 - \frac{1}{2}\mathbf{g}_1 \otimes \mathbf{g}_1$ *and Equation (3.26), we get that*

$$(\log \mathbf{m})_2 = \sum_{i=1}^N w^{(i)}(\log \mathbf{x}^{(i)})_2. \tag{3.39}$$

Remark 3.35. *As we have seen using the BCH formula in Example 3.17, the first two levels of* $\log \mathbf{m}$ *correspond to the Euclidean mean of* $\{\log \mathbf{x}^{(i)}\}_{i=1,\dots,N}$.

3.4.4 Expressions in the ambient space using the asymmetrized BCH formula

We conclude this section by noting that we can also find the explicit expressions in the ambient space using the symmetrized BCH formula developed in Section 3.3.4. With some abuse of notation, we denote by $\mathbf{b} = \log(\mathbf{m}^{-1})$ and $\mathbf{c}^{(i)} = \log(\mathbf{x}^{(i)})$, and we view them as elements of tensor algebra, split them by orders:

$$\begin{aligned}
\mathbf{b} &= (0, \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_L, \dots) \in T((\mathbb{R}^d)), \\
\mathbf{c}^{(i)} &= (0, \mathbf{c}_1^{(i)}, \mathbf{c}_2^{(i)}, \dots, \mathbf{c}_L^{(i)}, \dots) \in T((\mathbb{R}^d)).
\end{aligned} \tag{3.40}$$

Next, we recall that Lie brackets for $\mathbf{x}_k \in (\mathbb{R}^d)^k$ and $\mathbf{y}_\ell \in (\mathbb{R}^d)^\ell$ can be computed as

$$[\mathbf{x}_k, \mathbf{y}_\ell] := \mathbf{x}_k \otimes \mathbf{y}_\ell - \mathbf{y}_\ell \otimes \mathbf{x}_k \in (\mathbb{R}^d)^{\otimes(k+\ell)}.$$

If $\mathbf{x}, \mathbf{y} \in T((\mathbb{R}^d))$, then the L-th level of the Lie bracket is expressed as

$$([\mathbf{x}, \mathbf{y}])_L = \sum_{\ell=1}^{L-1}[\mathbf{x}_\ell, \mathbf{y}_{L-\ell}] = \sum_{\ell=1}^{L-1}(\mathbf{x}_\ell \otimes \mathbf{y}_{L-\ell} - \mathbf{x}_{L-\ell} \otimes \mathbf{y}_\ell), \tag{3.41}$$

analogously to (3.27). Then we have the following corollary of Theorem 3.23.

Corollary 3.36. *The elements* $\mathbf{b}_L \in (\mathbb{R}^d)^{\otimes L}$ *of the tensor series of* $\mathbf{b} = \log(\mathbf{m}^{-1})$ *(see* (3.40)*) can be computed as*

$$\mathbf{b}_L = -\sum_{i=1}^{N} w_i(\mathbf{c}_L^{(i)} + R_L(\mathbf{b}, \mathbf{c}^{(i)})),$$

where R_L *is a tensor product analogue of the polynomial defined in Equation* (3.18)*, i.e.,*

$$R_L(\mathbf{b}, \mathbf{c}) = R_L(\mathbf{b}_1, \dots, \mathbf{b}_{L-1}, \mathbf{c}_1, \dots, \mathbf{c}_{L-1}) := \Big(\sum_{\substack{k=3,\dots,L \\ k \text{ odd}}} \mathsf{aBCH}_k(\mathbf{b}, \mathbf{c}) \Big)_L. \tag{3.42}$$

Example 3.37. *For* $L \leq 2$*, we have* $R_L = 0$ *in* (3.42)*, which agrees with Equations* (3.38) *to* (3.39)*. For* $L = 3$*, we can apply* (3.41) *to* $([[\mathbf{b}, \mathbf{c}], \mathbf{c}])_3$ *to get*

$$R_3(\mathbf{b}, \mathbf{c}) = \frac{1}{12}[[\mathbf{b}_1, \mathbf{c}_1], \mathbf{c}_1] = \frac{1}{12}(\mathbf{c}_1 \otimes \mathbf{c}_1 \otimes \mathbf{b}_1 + \mathbf{b}_1 \otimes \mathbf{c}_1 \otimes \mathbf{c}_1 - 2\mathbf{c}_1 \otimes \mathbf{b}_1 \otimes \mathbf{c}_1).$$

For order 4*, similarly, we get*

$$R_4(\mathbf{b}, \mathbf{c}) = \Big(\frac{1}{12}[[\mathbf{b}, \mathbf{c}], \mathbf{c}]\Big)_4 = \frac{1}{12}(\underbrace{[[\mathbf{b}_2, \mathbf{c}_1], \mathbf{c}_1]}_{I} + \underbrace{[[\mathbf{b}_1, \mathbf{c}_2], \mathbf{c}_1] + [[\mathbf{b}_1, \mathbf{c}_1], \mathbf{c}_2]}_{II}) \tag{3.43}$$

$$= \frac{1}{12}(\mathbf{c}_1\mathbf{c}_1\mathbf{b}_2 + \mathbf{b}_2\mathbf{c}_1\mathbf{c}_1 - 2\,\mathbf{c}_1\mathbf{b}_2\mathbf{c}_1) \tag{I}$$

$$+ \frac{1}{12}(\mathbf{c}_1\mathbf{c}_2\mathbf{b}_1 + \mathbf{b}_1\mathbf{c}_1\mathbf{c}_2 + \mathbf{c}_2\mathbf{c}_1\mathbf{b}_1 + \mathbf{b}_1\mathbf{c}_2\mathbf{c}_1 - 2\,\mathbf{c}_2\mathbf{b}_1\mathbf{c}_1 - 2\,\mathbf{c}_1\mathbf{b}_1\mathbf{c}_2). \tag{II}$$

where in (I) *and* (II) *we omitted the tensor products for short.*

Note that R_3 and R_4 have 3 and 9 terms ("monomials") respectively, and R_5 (provided in Supplementary Materials) has 43 terms. A general bound on the number of terms is given in the following lemma.

Proposition 3.38. *The number of terms in* R_L *defined as* (3.42) *is bounded by*

$$\widetilde{Q}_L = \begin{cases} 0, & L < 3, \\ 3^{L-1} + (-1)^{L-1} - (L+5)2^{L-3} + 1, & L \geq 3. \end{cases}$$

Proof. We start by counting the number of terms that can appear in $(\mathsf{aBCH}_k(\mathbf{b}, \mathbf{c}))_L$ given $3 \leq k \leq L$. From (3.41), the only terms which can appear are of the form

$$\mathbf{a}_{\delta_1, i_1} \otimes \cdots \otimes \mathbf{a}_{\delta_k, i_k},$$

where $\delta_j \in \{0, 1\}$ such that $2 \leq \delta_1 + \cdots + \delta_k \leq L - 1$ and i_j are positive integers that satisfy

$$i_1 + \cdots + i_k = L.$$

and the vectors $\mathbf{a}_{\delta_j, i_j}$ are given by

$$\mathbf{a}_{\delta, i} = \begin{cases} \mathbf{b}_i, & \delta = 0, \\ \mathbf{c}_i, & \delta = 1. \end{cases}$$

L	3	4	5	6	7	8	9	10
$\widetilde{Q}_L$	3	9	43	155	539	1771	5667	17763

TABLE 3.2: Values of $\widetilde{Q}_L$ depending on L.

The condition on δ_k are implied by the fact that the terms should contain at least two $\mathbf{c}_{(\cdot)}$ vectors and at least one $\mathbf{b}_{(\cdot)}$ vector.

We can count the number of tuples $(i_1, \dots, i_k)$ by the number of compositions of L, which is given by $\binom{L-1}{k-1}$. Thus the total number of terms in R_L is bounded by

$$\sum_{\substack{3 \le k \le L \\ k \text{ odd}}} \underbrace{(2^k - k - 2)}_{\#of(\delta_1,\dots,\delta_k)} \underbrace{\binom{L-1}{k-1}}_{\#of(i_1,\dots,i_k)} = \widetilde{Q}_L,$$

which follows from straightforward computations. □

The values of $\widetilde{Q}_L$ for the first several orders of L are given in Table 3.2. This suggests that for low values of L the expansions of R_L can be precomputed and used for computations. We also note that $\widetilde{Q}_L$ provides a bound for the number of terms in Equation (3.18).

Remark 3.39. *Note that, by [Reu93, Theorem 5.3], for any $\mathbf{a}_L \in (\mathbb{R}^d)^L$, which is also in the Lie algebra, we have that $\mathbf{a}_{1,\dots,L}$ is exactly the coefficient for the element $[1, [2, [\dots, L]\dots]$ for $\mathbf{a}$ expanded in the Lyndon basis. This implies that $\widetilde{Q}_L$ is an upper bound for the number of terms of the corresponding polynomial $\widetilde{r}_j$ in (3.18).*

3.5 Open questions / Outlook

- Time series data is usually discrete-in-time, and the recently introduced iterated-sums signature (or discrete signature) provides a natural way to deal with such data, without the need to interpolate [DEFT20]. For an appropriately truncated version, one is again in the setting of a free nilpotent group. What is different in that setting is that not all group elements can be realized as the signature of a time series [DEFT20, p. 279]. Can all barycenters be realized, though?

- Are there other ways to define a bi-invariant group mean in the signature space? The essential property for bi-invariance was conjugation-equivariance of the logarithm,

  $$\log(\mathbf{g}^{-1}\mathbf{x}\mathbf{g}) = \mathbf{g}^{-1}\log(\mathbf{x})\mathbf{g}.$$

  Is this the only such map from grouplike elements to the Lie algebra, that is invertible?

Chapter 4

Principal Geodesic Analysis for signatures

We analyze multidimensional time series through the lens of their integrals of various moment orders, constituting their signature. The contribution of this chapter is to adapt the Principal Geodesic Analysis (PGA), the counterpart for manifolds of the Principal Component Analysis (PCA), to signature features which form a Lie group, by setting an appropriate connection structure. We show that, on both simulations and real data, our dimension reduction approach allows us to keep state-of-the-art performances with much less number of features.

4.1 Introduction

In many scenarios, data is naturally recovered in the form of time series, that is the observation of a (possibly multidimensional) process at different times. The analysis of such stream of data has become key in various fields, e.g., engineering, sociology and economics. Multiple tasks arise such as time series decomposition (trend, seasonality), modeling, forecasting, anomaly detection, correlation/auto-correlation analysis and causal inference to cite a few. For an exhaustive approach to time series analysis, we refer to [BD16] and to [Lüt05] for the multivariate case.

In this chapter, we analyze multivariate time series through their signatures. A characteristic of the signature is that the number of signature features grows exponentially with respect to the order. To deal with this, we provide a dimension reduction method for signature features, that is analogous to the Principal Component Analysis (PCA) for vector valued data.

Principal Component Analysis. Given samples $x_1, \dots, x_N$ in $\mathbb{R}^d$, PCA is a method for dimension reduction that provides a sequence of best linear approximations to the data, for all ranks $K \leq d$. Denote as $y_i := x_i - \mu$ the centered data points, where μ is the Euclidean mean of the set $x_1, \dots, x_N$. We compute a sequence of vectors $v_1, \dots, v_K$ successively by solving, for all $k = 1, \dots, K$,

$$v_k = \underset{\substack{\|v\|=1 \\ v \perp v_1, \dots, v_{k-1}}}{\arg\min} \sum_{i=1}^{N} \|y_i - \pi_v(y_i)\|^2 \tag{4.1}$$

where π_v is the orthogonal projection onto $\mathrm{span}(v)$ and where v_1 does not have an orthogonality requirement. The $(v_k)_k$ are called Principal Directions. We can compress the data by setting $K < d$ and projecting it on $\mathrm{span}(v_1, \dots, v_K)$. To solve Equation (4.1), we inject the explicit expression of the orthogonal projection π and the closed form solution is given using the singular value decomposition of $x = (x_1, \dots, x_N)$, see [HTF09, Section 14.5].

PCA for time series and manifolds. Consider N time series $x_1(t), \dots, x_N(t)$ evolving in $\mathbb{R}^d$ and sampled at discrete times $t = t_1, \dots, t_T$. That is, each x_i is a matrix of size $T \times d$. A straightforward extension of PCA to time series is the following: first flatten each matrix into vectors of size Td and then inject those vectors into the usual PCA [Rao58] [Tuc58]. Comprehensive details regarding this approach can be found in [RS05, Chapter 8]. A drawback of this approach is that we loose the multivariate structure inherent to the data.

Another strategy is to consider meaningful representations of time series instead of the raw data and then perform a PCA on those representations. Two examples of representations are signatures, used in the following, and power spectral densities. Power spectral densities of time series are considered in [CRT21], but only with stationary time series. Here with the signature we will not have such restriction. Note that representations of time series might provide features lying on a manifold. Thus, the usual PCA cannot be applied as it is.

An extension of the PCA for data lying on a manifold, called Principal Geodesic Analysis (PGA) have been developed in [FLJ03]. The problem is defined similarly as in Equation (4.1), but π_v is now the projection on a geodesic (starting from the origin with initial velocity v). Contrary to the Euclidean case, the resulting optimization problem does not have a closed form solution, if no further information on the manifold is given. For instance, the projection π_v might require to be approximated. Because of this, most of the work involving numerical calculation of the PGA relies on an approximation of it: the tangent PGA, also introduced in [FLJ03]. It consists in projecting the data onto the tangent space at the origin and performing a classical PCA. Also, note that the PCA is applied on centered data, thus a notion of barycenter must be defined and computed beforehand. In the end, PGA provides another way to extend the PCA to time series.

Contributions. Our contributions are as follows:

- We define an extension of the PCA for the signature space, Proposition 4.9. Our approach relies on the unique properties inherent to this space.
- We present two algorithms: one that approximate and one that exactly solve the resulting optimization problem, Algorithms 3 and 4, along with implementations in Python.
- We use both our dimension reductions methods for classical tasks on both synthetic and real-life data and show that performances are still high while keeping much less features than without dimension reduction, Section 4.5.2.

Notations. Throughout the chapter, we use the following notations:

- N — Number of multivariate time series.
- d — Number of components (features/channels) of each multivariate time series.
- T — Length of time series (number of timestamps).
- L — Truncation level of the signature feature.

Structure of the chapter. First, in Section 4.2, we present some properties of the signature mapping. Then, in Section 4.3, we introduce a similarity measure for signatures (a divergence) and the usual method to extend the PCA into a PGA for data in a Lie group. In Section 4.4, we adapt this extension to the specific case of signatures. Finally, in Section 4.5, we perform experiments with simulated and real data. Conclusion and perspectives are given in Section 4.6.

4.2 The signature space and its Lie group structure: connection and Riemannian metric

Properties of the signature mapping and the algebraic structure of the signature space $G_{\leq L}$ were presented in Sections 2.2.2 and 2.2.3, where we have globally defined the exponential and logarithm mappings. Now, we discuss the choice of a connection on $G_{\leq L}$. Note that elements of differential geometry are given in Appendix C.

For the signature space, there is no bi-invariant Riemannian metric available [PL20, Theorem 5.12, Proposition 3], as it only exists for Lie groups that are a direct product of compact and abelian groups. This fact makes it difficult to define a distance in a canonical way. However, we can measure distances using the notions of connection and parallel transport, briefly described below.

The affine connection is a tool to connect tangent spaces of the manifold and for instance translate a vector $\mathbf{v}$ defined on the tangent space at the identity $T_{\mathbf{1}}(G_{\leq L})$ to another tangent space $T_{\mathbf{g}}(G_{\leq L})$. This is called parallel transport (see Appendix C). In addition, setting up the following specific connection gives us closed form expression of the geodesics, which are generalization of straight lines on manifolds (see Appendix C). Thus, setting up a connection is crucial.

Among the natural family of bi-invariant connections suggested in [Car26], we choose the canonical Cartan Schouten (CCS) connection. The CCS connection is the most natural one because when there exists a bi-invariant metric on the Lie group, the canonical Cartan Schouten (CCS) connection is the Levi-Civita connection of that metric and when there is not, the CCS connection still exists. The CCS connection is the connection such that geodesics are one parameter subgroups of the form

$$\gamma_{\mathbf{g},\mathbf{v}}(t) := \mathbf{g}\exp(t\mathbf{v}) \tag{4.2}$$

for any $\mathbf{g} \in G$ and $\mathbf{v} \in T_{\mathbf{g}}(G)$. For the CCS connection the parallel transport is linked in a canonical way to left and right translations.

Definition 4.1. *For any $\mathbf{g} \in G$, we define the left translation $L_{\mathbf{g}} : G \longrightarrow G, \mathbf{h} \mapsto \mathbf{gh}$ and right translation $R_{\mathbf{g}} : G \longrightarrow G, \mathbf{h} \mapsto \mathbf{hg}$.*

Remark 4.2. *For any $\mathbf{h} \in G$, the differential*

$$\begin{aligned} dL_{\mathbf{g}} : T_{\mathbf{h}}(G) &\longrightarrow T_{\mathbf{gh}}(G) \\ \mathbf{u} &\mapsto (dL_{\mathbf{g}})_{\mathbf{h}}(\mathbf{u}) \end{aligned}$$

gives a natural identification of tangent spaces, where we have denoted as $T_{\mathbf{h}}(G)$ the tangent space at point $\mathbf{h}$.

We have the following result from [PL20, Section 5.3.3].

Proposition 4.3. *Let $G_{\leq L}$ be the signature space equipped with the CCS connection. The derived geodesics going through the identity with initial velocity $\mathbf{v}$ are of the form $\exp(t\mathbf{v})$ and the parallel transport along $\exp(t\mathbf{v})$ is, for any $\mathbf{v}, \mathbf{w} \in \mathfrak{g}_{\leq L}$,*

$$\Pi_{\mathbf{1}\to\exp(\mathbf{v})}\mathbf{w} = (dL_{\exp(\mathbf{v}/2)})_{\exp(\mathbf{v}/2)}(dR_{\exp(\mathbf{v}/2)})_{\mathbf{1}}\mathbf{w}. \tag{4.3}$$

The group exponential at $\mathbf{g} \in G_{\leq L}$ is, for any $\mathbf{v} \in T_{\mathbf{g}}(G_{\leq L})$,

$$\exp_{\mathbf{g}}(\mathbf{v}) := \mathbf{g}\exp((dL_{\mathbf{g}^{-1}})_{\mathbf{g}}\mathbf{v}) \tag{4.4}$$

and the group logarithm at $\mathbf{g} \in G_{\leq L}$ *is, for any* $\mathbf{h} \in G_{\leq L}$,

$$\log_{\mathbf{g}}(\mathbf{h}) = (dL_{\mathbf{g}})_{\mathbf{1}} \log(\mathbf{g}^{-1}\mathbf{h}). \tag{4.5}$$

For further details on connections, see Appendix C.

4.3 Mean and PGA in Lie groups

The first step of the PCA consists in centering the data (retracting the mean). Thus, for the PGA we will use the definition for the mean introduced in Chapter 3. Then, we present the PGA for Lie groups and how we can adapt it to the signature space.

4.3.1 Divergence on the signature space

In order to generalize Equation (4.1) to the space of signatures, we need to have a notion of distance. In this chapter, we start with an inner product defined on the tangent space at identity $T_{\mathbf{1}}(G_{\leq L})$, and extend it to any tangent space $T_{\mathbf{g}}(G_{\leq L})$ using the notion of connection and parallel transport as stated in the following.

Remark 4.4. *From any given inner product* $\langle ., . \rangle_{\mathbf{1}}$ *defined on* $T_{\mathbf{1}}(G)$*, we can define an inner product* $\langle ., . \rangle_{\mathbf{g}}$ *on* $T_{\mathbf{g}}(G)$ *for any* $\mathbf{g} \in G$ *using the parallel transport* Π *of the chosen connection:*

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{g}} := \left\langle \Pi_{\mathbf{g} \to \mathbf{1}} \mathbf{u}, \Pi_{\mathbf{g} \to \mathbf{1}} \mathbf{v} \right\rangle_{\mathbf{1}} , \tag{4.6}$$

for any $\mathbf{u}, \mathbf{v} \in T_{\mathbf{g}}(G)$*. The norm associated to* $\langle ., . \rangle_{\mathbf{g}}$ *is denoted* $\|.\|_{\mathbf{g}}$.

In the following, we will typically set $\langle ., . \rangle_{\mathbf{1}}$ as $\langle ., . \rangle_F$ from Equation 2.36, the inherited norm of the ambient space. Note that the value of $\langle ., . \rangle_F$ depends on the choice of the basis on $T_{\mathbf{1}}(G)$, here set to a basis of the ambient space—the tensor algebra $T(\mathbb{R}^d)$.

The inner product allows us to define a divergence, introduced below, to measure distances between points on the space of signatures.

Definition 4.5 (Divergence [AC10])**.** *Let* $\mathcal{M}$ *be a manifold. A function* $D(x : y)$ *is called a divergence if, for all* $x, y \in \mathcal{M}$*, the following two conditions holds.*

i. $D(x : y) \geq 0$ *with equality if and only if* $x = y$.

ii. $D(x : y)$ *is differentiable and the Hessian with respect to* y *at* $y = x$ *is positive definite.*

Note that a divergence is not a distance metric as it is not necessary symmetric and it might not satisfy the triangle inequality.

Proposition 4.6. *The function* $D : G \times G \to \mathbb{R}$ *defined as, for any* $\mathbf{g}, \mathbf{h} \in G$,

$$D(\mathbf{g} : \mathbf{h}) = \|\log_{\mathbf{g}} \mathbf{h}\|^2_{\mathbf{g}} = \|\Pi_{\mathbf{g} \to \mathbf{1}} \log_{\mathbf{g}} \mathbf{h}\|^2_{\mathbf{1}} \tag{4.7}$$

where $\log_{\mathbf{g}}$ *is defined in Proposition 4.3, is a divergence. In particular, the associated Riemannian metric coincides with* $\langle \cdot, \cdot \rangle_{\mathbf{g}}$ *up to a global constant.*

Proof. See Appendix A.2. □

An explicit expression for $D(\mathbf{g} : \mathbf{h})$ in terms of group operations is given in Section 4.4.2.

4.3.2 Barycenter and relation to divergence

In this subsection, we highlight the relation of the signature barycenter (Chapter 3) to the introduced notion of divergence. We first recall the case of Euclidean space and bi-invariant Riemannian metric.

The usual definition of mean in Euclidean space $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ cannot be used for manifolds, since in many cases $\bar{x}$ might not belong to the manifold. A generalization of the Euclidean barycenter to manifolds is the Fréchet mean: let $(\mathcal{M}, d)$ be a metric space. Given a set of points $x_1, \dots, x_N \in \mathcal{M}$, the Fréchet mean is the point $\mu \in \mathcal{M}$ such that

$$\mu = \arg\min_{\mu} \sum_{i=1}^{N} d^2(\mu, x_i). \tag{4.8}$$

This definition is frequently used for Lie groups. If $\mathcal{M}$ is a Lie group that admits a a bi-invariant Riemannian metric, then μ, defined with respect to the corresponding distance $d(.,.)$, is invariant by group operations (left and right multiplication) and is compatible with inversion. For instance, invariance with respect to right multiplication means that μy is the Fréchet mean of $\{x_i y\}_{i=1,\dots,N}$. However, non-compact Lie groups do not admit a bi-invariant Riemannian metric, that is why we have introduced the concept of group mean in Chapter 3. For a finite set of points, the group mean is

$$0 = \sum_{i=1}^{N} \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i). \tag{4.9}$$

In other words, for a set of N points $\mathbf{x}_1, \dots, \mathbf{x}_N$, we look for $\boldsymbol{\mu}$ such that vectors $\mathbf{v}_i$ in the tangent space at the identity $T_{\mathbf{1}}(G)$ have mean zero, where $\mathbf{v}_i := \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i)$, see Figure 4.1.

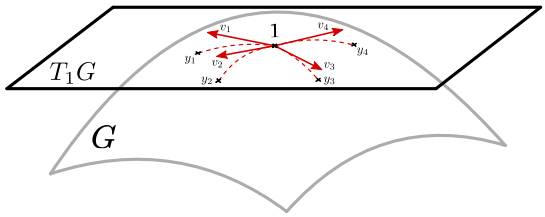


FIGURE 4.1: The group mean $\boldsymbol{\mu}$ on a Lie group G is such that the sum of the $\mathbf{v}_i$ is zero where we have denoted $\mathbf{v}_i := \log(\mathbf{y}_i) = \log(\boldsymbol{\mu}^{-1}\mathbf{x}_i)$ vectors in the tangent space $T_{\mathbf{1}}(G)$. Dotted lines are the geodesics on the Lie group G starting from the origin with initial velocity $\mathbf{v}_i$.

This notion of barycenter is bi-invariant (with respect to left and right multiplication). For compact Lie groups, the barycenter coincides with the Fréchet mean. For the group of signatures $G_{\leq L}$ (which is non-compact), it was shown in [Cla+24] that the barycenter is globally defined and is unique. Moreover, there is an explicit method to compute the group mean of a finite set of signatures. The bi-invariance of the barycenter, in the case of the signature space, is related to the concatenation of trajectories since we have Proposition 2.20. This is illustrated in Figure 4.2.

We conclude this section with establishing the link between the barycenter and the notion of divergence. We first introduce the operation of dilation of a signature $\mathbf{a} = (\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_L)$ as follows:

$$\delta_\lambda(\mathbf{a}) = (\mathbf{a}_0, \lambda\mathbf{a}_1, \lambda^2\mathbf{a}_2, \dots, \lambda^L\mathbf{a}_L).$$

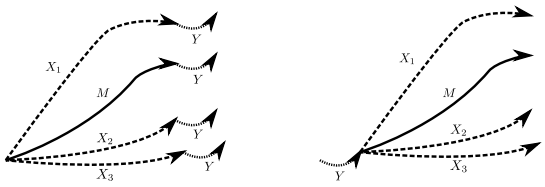


FIGURE 4.2: On the left, the mean of the signatures $\mathbf{S}(X_i)$ of three trajectories is assumed to be given by the signature $\mathbf{S}(M)$ of some trajectory M. If we attach to all trajectories X_i a new trajectory segment Y, since the mean is right invariant, this corresponds to attaching that trajectory segment to M. On the right, we see the analogous visualization for left invariance.

The dilation corresponds to the signature of the path rescaled by λ, that is $\mathbf{S}(\lambda X) = \delta_\lambda(\mathbf{S}(X))$. Then the following proposition holds true.

Proposition 4.7. *There exists λ_0 such that for all $0 < \lambda < \lambda_0$, the optimization problem*

$$\min_{\mathbf{m}\in G} f(\mathbf{m}), \quad \text{where } f(\mathbf{m}) = \sum_{i=1}^{N} D(\delta_\lambda(\mathbf{m}) : \delta_\lambda(\mathbf{x}_i)) \tag{4.10}$$

has a unique solution $\mathbf{m}(\lambda)$, which converges to the group mean ($\mathbf{m}(\lambda) \to \boldsymbol{\mu}$) as $\lambda \to 0$.

Proof. See Appendix A.2. □

Proposition 4.7, in fact, means that we can obtain the barycenter by jointly rescaling the input paths and minimizing the sum of divergences.

4.3.3 PGA in Lie groups

Principal Geodesic Analysis has first been introduced in [FLJ03]. The computation of the PGA components involve an optimization problem, shown below, that is dealt with using a linear approximation (tangent PGA). Here, the idea is to go beyond this linearization, which might lead to a too crude approximation of the geometry of the considered manifold, by solving the optimization problem with an exact computation.

Now, we introduce the Principal Geodesic Analysis in the specific context of Lie groups, as presented in [Sai+07]. The goal is to generalize the PCA, that is the optimization problem in Equation (4.1) defined on Euclidean spaces, to Lie groups (i.e., manifolds with a group structure). Let $\mathbf{x}_1, \dots, \mathbf{x}_N \in G_{\leq L}$ be N signatures with group mean $\boldsymbol{\mu}$. Denote $\mathbf{y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ where $\mathbf{y}_i := \boldsymbol{\mu}^{-1}\mathbf{x}_i$ is the centered data. PGA is the following optimization problem:

$$\underset{\substack{\mathbf{v}\in T_{\mathbf{1}}(G_{\leq L}) \\ \|\mathbf{v}\|=1}}{\arg\min} F_{\mathbf{y}}(\mathbf{v}) \tag{4.11}$$

where $F_{\mathbf{y}} : T_{\mathbf{1}}(G_{\leq L}) \to \mathbb{R}$ is defined as

$$F_{\mathbf{y}}(\mathbf{v}) := \sum_{i=1}^{N} \min_{t\in\mathbb{R}} D(\gamma_{\mathbf{v}}(t) : \mathbf{y}_i) \tag{4.12}$$

with $\gamma_{\mathbf{v}}$ the geodesic starting from identity element $\mathbf{1}$ with velocity $\mathbf{v}$.

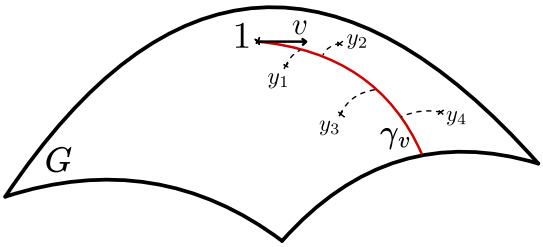


FIGURE 4.3: Illustration of Equation (4.12). To find the first principal direction, we search for an initial velocity $\mathbf{v}$ that minimizes the sum of the lengths of the dashed lines (distance to geodesic).

In other words, in Equation (4.11), we want to minimize the distance between the data points $\mathbf{y}_i$ and their projections onto the geodesic $\gamma_{\mathbf{v}}$ or equivalently, we want to maximize the variance of data points projected onto a geodesic, see Figure 4.3. In particular, if we replace $G_{\leq L}$ with $\mathbb{R}^d$, for some d, in the optimization problem defined in Equation (4.11), we obtain the PCA, as defined in Equation (4.1).

Similarly to the the PCA, the PGA successively finds principal directions $\mathbf{v}_1, \mathbf{v}_2, \ldots$. In the case of the PCA, we add the constraint for direction v_2 to be orthogonal to the first direction v_1. In the case of Lie groups, we do not use an orthogonality condition but instead we use the group structure in the following way:

i. The first step of PGA is to solve Equation (4.11) for data points $\mathbf{g}_i^{(1)} := \mathbf{y}_i$.

ii. Second step consists in finding a second principal geodesic by solving Equation (4.11) for
$$\mathbf{g}_i^{(2)} := (\mathbf{p}_i^{(1)})^{-1}\mathbf{g}_i^{(1)} , \tag{4.13}$$
where $\mathbf{p}_i^{(1)}$ is the projection of $\mathbf{g}_i^{(1)}$ on the geodesic $\gamma_{\mathbf{v}_1}$.

iii. The k-th step consists in finding the k-th geodesic. To this end, we solve Equation (4.11) for
$$\mathbf{g}_i^{(k)} := (\mathbf{p}_i^{(k-1)})^{-1}\mathbf{g}_i^{(k-1)} , \tag{4.14}$$
where $\mathbf{p}_i^{(k-1)}$ is the projection of $\mathbf{g}_i^{(k-1)}$ on geodesic $\gamma_{\mathbf{v}_{k-1}}$.

Note that we have the following reconstruction of the data points, for any integer k,
$$\mathbf{x}_i = \mu\mathbf{p}_i^{(1)}\mathbf{p}_i^{(2)}\ldots\mathbf{p}_i^{(k)}\mathbf{g}_i^{(k)} . \tag{4.15}$$

Related work. In [SLN14], the authors perform the PGA (without any linearization) on Lie groups that are also differentiable manifolds, in all generality, considering as distance in Equation (4.12) the Riemannian distance. Therefore, they have to use the geodesics written with implicit equations (with Christoffel symbols) to perform their calculations. In our case, we shall provide an explicit formula taking benefit only of the Lie group structure of the signature space which gives us an explicit closed form of the geodesic.

It has already be done in the work on Lie groups mentioned above [Sai+07]. In this case the Lie group distance involved in Equation (4.12) is a bi-invariant Riemannian metric allowing to simplify the expression of Equation (4.12). Unfortunately, in our setting, we do not have a bi-invariant Riemmanian metric on the signature space, hence the use of the CCS connection as mentioned above.

4.4 Extension of PCA for signature

4.4.1 Approximation in the tangent space

As pointed out in Section 4.1, PGA can be approximated easily with the tangent PGA (tPGA) method. The tPGA procedure performs the following three steps: center the data, project it into the tangent space at the identity and then apply the PCA. This procedure is valid on any manifold. To apply the tPGA to data on $G_{\leq L}$ the signature space, we simply adapt the first and second steps: data is centered using the group mean, see Definition 3.7, and the projection on the tangent space is performed using log mapping, see Equation (2.25). The tangent PGA method for signatures (SIG-tPGA) is presented in Algorithm 3. Although tPGA is a rough approximation of PGA, it has the benefit of having much shorter computation runtimes.

Algorithm 3: Tangent PGA for signatures (SIG-tPGA)

Input: $\mathbf{x}_1, \dots, \mathbf{x}_N$ set of N signatures

$\mu \leftarrow$ Group mean of $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ `// Using [Cla+24, Algorithm 3]`

for $i = 1, \dots, N$ **do**

  $\mathbf{y}_i \leftarrow \mu^{-1}\mathbf{x}_i$

  $\mathbf{u}_i \leftarrow \log \mathbf{y}_i$

Apply PCA to $\mathbf{u} := \{\mathbf{u}_1, \dots, \mathbf{u}_N\}$ (diagonalization of the covariance matrix of u)

Output: Principal directions $\mathbf{v}_1, \dots, \mathbf{v}_K$

4.4.2 Estimation of Principal Geodesics in the signature space

In view of Section 4.3.3, we see that the core of the PGA algorithm is solving the optimization problem defined in Equation (4.11). To solve Problem (4.11), we need to calculate the gradient of the objective function $F_{\mathbf{y}}$. Denote as $P_{\mathbf{v},\mathbf{y}}(t)$ the following

$$P_{\mathbf{v},\mathbf{y}}(t) := D(e^{t\mathbf{v}} : \mathbf{y}) := \|\log_{e^{t\mathbf{v}}} y\|^2_{e^{t\mathbf{v}}} \tag{4.16}$$

and

$$a(\mathbf{v}, t, \mathbf{y}) := \log\left(e^{-\frac{1}{2}t\mathbf{v}}\mathbf{y}e^{-\frac{1}{2}t\mathbf{v}}\right) . \tag{4.17}$$

Lemma 4.8. *Let the signature space $G_{\leq L}$ be equipped with the CCS connection. Then, we have for any $\mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$ and $t \in \mathbb{R}$,*

$$P_{\mathbf{v},\mathbf{y}}(t) = \|a(\mathbf{v}, t, \mathbf{y})\|^2_{\mathbf{1}} \tag{4.18}$$

where $\|\mathbf{u}\|^2_{\mathbf{1}} := \langle \mathbf{u}, \mathbf{u}\rangle_{\mathbf{1}}$ is an inner product on $T_{\mathbf{1}}(G_{\leq L})$.

Now, we introduce the main proposition.

Proposition 4.9. *Let L be a positive integer (truncation level) and $\mathbf{y}_1, \dots, \mathbf{y}_N$ be N elements in $G_{\leq L}$, where $G_{\leq L}$ is equipped with the CCS connection and $T_{\mathbf{1}}(G_{\leq L})$ is equipped with an inner product $\langle ., . \rangle_{\mathbf{1}}$. We have, for any $\mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$,*

$$\nabla F_{\mathbf{y}}(\mathbf{v}) = \sum_{i=1}^{N} \left\langle a(\mathbf{v}, t^*_{\mathbf{v},\mathbf{y}_i}, \mathbf{y}_i), \nabla_{\mathbf{v}} a(\mathbf{v}, t^*_{\mathbf{v},\mathbf{y}_i}, \mathbf{y}_i)\right\rangle_{\mathbf{1}} \tag{4.19}$$

where $a(\mathbf{v}, t, \mathbf{y})$ *is given in Equation* (4.17) *and*

$$t^*_{\mathbf{v},\mathbf{y}_i} := \arg\min_{t\in\mathbb{R}} P_{\mathbf{v},\mathbf{y}_i}(t) \tag{4.20}$$

with $P_{\mathbf{v},\mathbf{y}_i}$*, given in Equation* (4.16)*, is a polynomial function which expression can be computed explicitly.*

Before proving this proposition, we show how it is useful for our purpose. In order to solve Equation (4.11), we proceed as following. Initialize $\mathbf{v}_k^{(0)}$ such that $\|\mathbf{v}_k^{(0)}\| = 1$, then find $t_{k,0,i} := \arg\min_{t\in\mathbb{R}} P_{\mathbf{v}_k^{(0)},\mathbf{y}_i}(t)$ for all $i = 1, \dots, N$. Now, repeat the following two steps until convergence:

i. **Find k-th Principal Direction step.** Denote $T_{k,l} := \{t_{k,l,1}, \dots, t_{k,l,N}\}$ and $F_{T_{k,l}}(\mathbf{v}) := \sum_{i=1}^N d(\gamma_{\mathbf{v}}(t_{k,l,i}), \mathbf{y}_i)^2$ for any integers k, l. Perform one step of Gradient Descent (or of any gradient based optimization method):

$$\mathbf{v}_k^{(l+1)} \leftarrow \mathbf{v}_k^{(l)} - \alpha(\nabla F_{T_k})_{\mathbf{v}_k^{(l)}}$$
$$\mathbf{v}_k^{(l+1)} \leftarrow \frac{\mathbf{v}_k^{(l+1)}}{\|\mathbf{v}_k^{(l+1)}\|}$$

where index l denotes gradient descent steps.

ii. **Projection step.** For a fixed $\mathbf{v}_k^{(l)}$ and for all $i = 1, \dots, N$, find

$$t_{k,l,i} := \arg\min_{t\in\mathbb{R}} P_{\mathbf{v}_k^{(l)},\mathbf{y}_i}(t) \tag{4.21}$$

where the polynomial coefficients can be explicitly computed. This step is most efficiently performed using a numerical root finder for polynomials on $\frac{d}{dt}P_{\mathbf{v}_k^{(l)},\mathbf{y}_i}(t)$, as described later on in Section 4.4.4.

To sum up, we fix a direction $\mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$, project the data on the geodesic with velocity $\mathbf{v}$, then fix a new direction, project again, etc. The PGA algorithm for signatures (SIG-PGA) is detailed in Algorithm 4.

4.4.3 Proofs

The following two subsections contain the proof of Lemma 4.8 and Proposition 4.9 respectively.

Proof of Lemma 4.8

To prove Lemma 4.8, we use the properties of left/right translation and the adjoint representation from Appendix A.1.

Proof of Lemma 4.8. **Step 1: use parallel transport to give a meaning to inner product.** On $G_{\leq L}$ equipped with the CCS connection, we have used that $\log_{e^{t\mathbf{v}}}(\mathbf{y}) = (dL_{e^{t\mathbf{v}}})_{\mathbf{1}} \log(e^{-t\mathbf{v}}\mathbf{y})$, see Proposition 4.3. Thus,

$$D(e^{t\mathbf{v}} : \mathbf{y}) := \|\log_{e^{t\mathbf{v}}} \mathbf{y}\|^2_{e^{t\mathbf{v}}} \tag{4.22}$$
$$= \|(dL_{e^{t\mathbf{v}}})_{\mathbf{1}} \log(e^{-t\mathbf{v}}\mathbf{y})\|^2_{e^{t\mathbf{v}}} \tag{4.23}$$

Algorithm 4: Principal Geodesic Analysis of Signatures (SIG-PGA)

Input: $\{\mathbf{x}_i := \mathbf{S}_{\leq L}(X_i), i = 1, \dots, N\}$: a batch of N signatures up to level L.
K : integer, the number of Principal Geodesics to keep

$\boldsymbol{\mu} \leftarrow$ Group mean of $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ `// Using [Cla+24, Algorithm 3]`

for $i = 1, \dots, N$ **do**

  $\mathbf{g}_i^{(0)} \leftarrow \boldsymbol{\mu}^{-1}\mathbf{x}_i$ the centered data

for $k = 1, \dots, K$ **do**

  Initialize $\mathbf{v}_k$ (e.g., $\mathbf{v}_k \leftarrow \log(\mathbf{g}_i^{(k-1)})$ for a randomly chosen index i)

  Repeat the following two steps until convergence:

  1. $\mathbf{v}_k^{(l+1)} \leftarrow$ One step of Gradient Descent for $F_{\mathbf{p}_{\cdot,k}}$ at $\mathbf{v}_k^{(l)}$

  2. $\mathbf{p}_i^{(k)} \leftarrow$ Projection of $\mathbf{g}_i^{(k-1)}$ on $\gamma_{\mathbf{v}_k^{(l+1)}}$ the geodesic starting from identity with initial velocity $\mathbf{v}_k^{(l+1)}$

  $\mathbf{g}_i^{(k)} \leftarrow (\mathbf{p}_i^{(k)})^{-1}\mathbf{g}_i^{(k-1)}$

Output: $\mathbf{v}_1, \dots, \mathbf{v}_K$ the first K geodesic directions

Using the expression of the parallel transport of the CCS connection, see Proposition 4.3, we have for any $\mathbf{g} \in G_{\leq L}$, $\Pi_{\mathbf{g}\to\mathbf{1}} : T_{\mathbf{g}}(G_{\leq L}) \to T_{\mathbf{1}}(G_{\leq L})$ with

$$\Pi_{\mathbf{g}\to\mathbf{1}} = (dL_{\mathbf{g}^{-1/2}})_{\mathbf{g}^{1/2}}(dR_{\mathbf{g}^{-1/2}})_{\mathbf{g}} \tag{4.24}$$

where we have denoted $\mathbf{g}^{\alpha} := \exp(\alpha \log \mathbf{g})$ for any real value α. Thus, we have

$$\begin{aligned} D(e^{t\mathbf{v}} : \mathbf{y}) &= \|\Pi_{e^{t\mathbf{v}}\to\mathbf{1}}(dL_{e^{t\mathbf{v}}})_{\mathbf{1}} \log(e^{-t\mathbf{v}}\mathbf{y})\|_{\mathbf{1}}^2 && (4.25)\\ &= \|d(L_{e^{\frac{1}{2}t\mathbf{v}}})d(R_{e^{-\frac{1}{2}t\mathbf{v}}}) \log(e^{-t\mathbf{v}}\mathbf{y})\|_{\mathbf{1}}^2 && (4.26)\end{aligned}$$

where we have used the commutativity of dL and dR (Lemma A.1) in the last equation and that $dL_{\mathbf{g}} \circ dL_{\mathbf{h}} = dL_{\mathbf{gh}}$ (Lemma A.2). In other words, from the definition of Ad (Definition A.3),

$$D(e^{t\mathbf{v}} : \mathbf{y}) = \|\mathrm{Ad}(e^{\frac{1}{2}t\mathbf{v}}) \log(e^{-t\mathbf{v}}\mathbf{y})\|_{\mathbf{1}}^2 \tag{4.27}$$

Step 2: simplify the formula using the fact we have explicit expressions in the case of the signature. Using Lemma A.4 and that $\mathbf{g}\log(\mathbf{h})\mathbf{g}^{-1} = \log(\mathbf{ghg}^{-1})$, we have

$$\begin{aligned} \mathrm{Ad}(e^{\frac{1}{2}t\mathbf{v}}) \log(e^{-t\mathbf{v}}\mathbf{y}) &= e^{\frac{1}{2}t\mathbf{v}} \log(e^{-t\mathbf{v}}\mathbf{y})e^{-\frac{1}{2}t\mathbf{v}} && (4.28)\\ &= \log(e^{\frac{1}{2}t\mathbf{v}}e^{-t\mathbf{v}}\mathbf{y}e^{-\frac{1}{2}t\mathbf{v}}) && (4.29)\\ &= \log(e^{-\frac{1}{2}t\mathbf{v}}\mathbf{y}e^{-\frac{1}{2}t\mathbf{v}}), && (4.30)\end{aligned}$$

which completes the proof. □

Proof of Proposition 4.9

To prove Proposition 4.9, we need the two following results. The first one gives the formula of the product of two exponentials in the particular case of signatures and the second one asserts the existence of a global minimum of polynomial $P_{\mathbf{v},\mathbf{y}}$ defined before.

Lemma 4.10. *Let* $\mathbf{u}, \mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$. *Then,*

$$\log\left(e^{\mathbf{u}/2}e^{\mathbf{v}}e^{\mathbf{u}/2}\right) =: \mathrm{BCH}_{sym}(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{L} z_i E_i \tag{4.31}$$

where $z_i \in \mathbb{Q}$ *for all* $i \geq 1$ *and* $(E_i)_{i\geq 1}$ *are of the form* $E_1 = \mathbf{u}$, $E_2 = \mathbf{v}$ *and for all* $i \geq 3$, $E_i = [E_{i'}, E_{i''}]$ *for values* $i', i'' \leq i$. *Denote* $|i|$ *the homogeneous degree:* $|1| = |2| = 1$ *and* $|i| = |i'| + |i''|$ *for* $i \geq 3$. *Then, for terms of even degree* $|i|$, *we have* $z_i = 0$.

For instance, for $L = 3$ *or* $L = 4$, *Equation* (4.31) *is*

$$\mathrm{BCH}_{\mathrm{sym}}(\mathbf{u}, \mathbf{v}) = \mathbf{u} + \mathbf{v} - \frac{1}{24}[\mathbf{u}, [\mathbf{u}, \mathbf{v}]] - \frac{1}{12}[\mathbf{v}, [\mathbf{u}, \mathbf{v}]]. \tag{4.32}$$

Proof. See for instance [CM09, Section 4.1] for a proof of the BCH_{sym} formula in the general case, which gives a series instead of a sum on the RHS of Equation (4.31). Here, we only have L terms on the RHS thanks to the nilpotency of $T_{\mathbf{1}}(G_{\leq L})$, that is, if $\mathbf{u} \in T_{\mathbf{1}}(G_{\leq L})$ then $\mathbf{u}^k = 0$ for any $k > L$. Thus, if $\mathbf{u}, \mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$, then all Lie brackets with more than L terms vanish. □

Lemma 4.11. *Let the truncation order* $L \in \mathbb{N}$ *be fixed. For any* $\mathbf{v} \in T_{\mathbf{1}}(G_{\leq L})$ *and* $\mathbf{y} \in G_{\leq L}$. *Then, the polynomial* $t \mapsto P_{\mathbf{v},\mathbf{y}}(t)$ *defined in Equation* (4.16) *has a global minimum that we denote* $t^*_{\mathbf{v},\mathbf{y}}$.

Proof of Lemma 4.11. Denote as $p_{\mathbf{v},\mathbf{y}}$ the polynomial such that, for any $t \in \mathbb{R}$,

$$p_{\mathbf{v},\mathbf{y}}(t) := \frac{d}{dt}P_{\mathbf{v},\mathbf{y}}(t). \tag{4.33}$$

We have

$$\frac{d}{dt}\|a(\mathbf{v}, t, \mathbf{y})\|_{\mathbf{1}} = 2\left\langle a(\mathbf{v}, t, \mathbf{y}), \frac{d}{dt}a(\mathbf{v}, t, \mathbf{y})\right\rangle_{\mathbf{1}} \tag{4.34}$$

by definition of the derivative, continuity of the inner product and continuity of $t \mapsto a(\mathbf{v}, t, \mathbf{y})$. Now, using Lemma 4.10, we have

$$a_{\mathbf{v},t,\mathbf{y}} := \log\left(e^{-\frac{1}{2}t\mathbf{v}}\mathbf{y}e^{-\frac{1}{2}t\mathbf{v}}\right) \tag{4.35}$$

$$= \mathrm{BCH}_{sym}\left(-\frac{1}{2}t\mathbf{v}, \log\mathbf{y}\right). \tag{4.36}$$

Using the linearity in t of the Lie brackets in the BCH formula, we obtain that $t \mapsto a(\mathbf{v}, t, \mathbf{y})$ is a polynomial function, that we denote $\sum_{k=0}^{K}\alpha_k t^k$, where degree K depends on L, because of Lemma 4.10. Injecting this into p gives

$$p_{\mathbf{v},\mathbf{y}}(t) = 2\left\langle \sum_{i=0}^{K}\alpha_i t^i, \sum_{j=1}^{K} j\alpha_j t^{j-1}\right\rangle_{\mathbf{1}} \tag{4.37}$$

$$= 2\sum_{i=0}^{K}\sum_{j=1}^{K} j\left\langle \alpha_i, \alpha_j\right\rangle_{\mathbf{1}} t^{i+j-1} \tag{4.38}$$

That is, p is a polynomial which highest degree term is $2K\|\alpha_K\|_{\mathbf{1}}^2 t^{2K-1}$. Thus, $P_{\mathbf{v},\mathbf{y}}$ has highest degree term $\|\alpha_K\|_{\mathbf{1}}^2 t^{2K}$. $P_{\mathbf{v},\mathbf{y}}$ is a polynomial with even degree and positive highest degree term $\|\alpha_K\|_{\mathbf{1}}^2 > 0$. That is $\lim_{t\to\pm\infty} P_{\mathbf{v},\mathbf{y}}(t) = +\infty$. For any real value $M > 0$, it exists $R > 0$ such that $P_{\mathbf{v},\mathbf{y}}(t) > M$ for all $|t| > M$. That is $\min_{t\in\mathbb{R}} P_{\mathbf{v},\mathbf{y}}(t) =$

$\min_{-R\le t\le R} P_{\mathbf{v},\mathbf{y}}(t)$. Since $P_{\mathbf{v},\mathbf{y}}$ is continuous on the closed and bounded set $[-R, R]$, using the extreme value theorem we have that $P_{\mathbf{v},\mathbf{y}}$ attains its minimum. □

We now have all the tools to prove the main result.

Proof of Proposition 4.9. Using Lemmas 4.8 and 4.11, Equation (4.11) becomes

$$F_{\mathbf{y}}(\mathbf{v}) = \sum_{i=1}^{N} \|a(\mathbf{v}, t^*_{\mathbf{v},\mathbf{y}_i}, \mathbf{y}_i)\|^2_{\mathbf{1}}. \tag{4.39}$$

where the value of $t^*_{\mathbf{v},\mathbf{y}_i}$ can be calculated explicitly using a numerical root finder for polynomials, applied to $p_{\mathbf{v},\mathbf{y}}(t)$. Then, we differentiate this equation with respect to $\mathbf{v}$ and obtain the expression of $\nabla_{\mathbf{v}} F_{\mathbf{y}}$ as shown in Equation (4.19). □

4.4.4 Algorithmic details

In Proposition 4.9, the obtained expression of $\nabla F_{\mathbf{y}}$ depends on N implicitly defined values $t^*_{\mathbf{v},\mathbf{y}_i}$. In this section, we show how to derive closed form expressions for $t^*_{\mathbf{v},\mathbf{y}_i}$ for specific values of the truncation level L. Ultimately, it leads to greatly diminished computation times of SIG-PGA (Algorithm 4).

To this end, note that $t^*_{\mathbf{v},\mathbf{y}_i}$ is the arg min of polynomial function $P_{\mathbf{v},\mathbf{y}_i}$, i.e., $t^*_{\mathbf{v},\mathbf{y}_i}$ is a root of $p_{\mathbf{v},\mathbf{y}}(t)$ the derivative of $P_{\mathbf{v},\mathbf{y}_i}$. We have shown in Equation (4.37) that $p_{\mathbf{v},\mathbf{y}}(t)$ is a polynomial with degree K depending on L. Setting L to specific values allows us to derive the expressions of $p_{\mathbf{v},\mathbf{y}}(t)$ and then to compute its roots explicitly. This allows us to avoid using a generic root finder. In this section, we detail two specific cases to illustrate this: $L = 2$ and $L \in \{3, 4\}$.

In the following two examples, we denote as $\mathbf{u} := \log \mathbf{y}$.

Example 4.12. *If truncation level $L = 2$,*

$$a(\mathbf{v}, t, \mathbf{y}) = \mathrm{BCH}_{sym}\left(-\frac{1}{2}t\mathbf{v}, \mathbf{u}\right) = -t\mathbf{v} + \mathbf{u} \tag{4.40}$$

that is,

$$p_{\mathbf{v},\mathbf{y}}(t) = 2\langle -t\mathbf{v} + \mathbf{u}, -\mathbf{v}\rangle_{\mathbf{1}} \tag{4.41}$$

$$= 2\langle \mathbf{v}, \mathbf{v}\rangle_{\mathbf{1}}\, t - 2\left\langle \log \mathbf{y}, \mathbf{v}\right\rangle_{\mathbf{1}}. \tag{4.42}$$

Then, solving $p_{\mathbf{v},\mathbf{y}}(t^*) = 0$ gives

$$t^* = \frac{\left\langle \log \mathbf{y}, \mathbf{v}\right\rangle_{\mathbf{1}}}{\langle \mathbf{v}, \mathbf{v}\rangle_{\mathbf{1}}}. \tag{4.43}$$

Example 4.13. *If truncation level $L = 3$ or $L = 4$ (this is the same formula as terms of odd degrees vanish in BCH_{sym}),*

$$a(\mathbf{v}, t, \mathbf{y}) = \mathrm{BCH}_{sym}\left(-\frac{1}{2}t\mathbf{v}, \mathbf{u}\right) = -t\mathbf{v} + \mathbf{u} - \frac{1}{24}t^2[\mathbf{v}, [\mathbf{v}, \mathbf{u}]] - \frac{1}{12}t[[\mathbf{v}, \mathbf{u}], \mathbf{u}] \tag{4.44}$$

that is,

$$p_{\mathbf{v},\mathbf{y}}(t) = 2\Big\langle -t\mathbf{v} + \mathbf{u} - \frac{1}{24}t^2[\mathbf{v}, [\mathbf{v}, \mathbf{u}]] - \frac{1}{12}t[[\mathbf{v}, \mathbf{u}], \mathbf{u}],$$

$$\mathbf{v} + \frac{1}{12}[[\mathbf{v},\mathbf{u}],\mathbf{u}] + \frac{1}{12}t[\mathbf{v},[\mathbf{v},\mathbf{u}]]\Big\rangle_{\mathbf{1}} \tag{4.45}$$

$$= 2\left\langle \mathbf{u} + At + Bt^2, C + Dt\right\rangle_{\mathbf{1}} \tag{4.46}$$

$$= 2\langle B, D\rangle_{\mathbf{1}} t^3 + 2\left(\langle B, C\rangle_{\mathbf{1}} + \langle A, D\rangle_{\mathbf{1}}\right) t^2 + 2\left(\langle A, C\rangle_{\mathbf{1}} + \langle \mathbf{u}, D\rangle_{\mathbf{1}}\right) t + 2\langle \mathbf{u}, C\rangle_{\mathbf{1}} \tag{4.47}$$

where $A := -\mathbf{v} - \frac{1}{12}[[\mathbf{v},\mathbf{u}],\mathbf{u}]$, $B = -\frac{1}{24}[\mathbf{v},[\mathbf{v},\mathbf{u}]]$, $C = \mathbf{v} + \frac{1}{12}[[\mathbf{v},\mathbf{u}],\mathbf{u}]$ *and* $D = \frac{1}{12}[\mathbf{v},[\mathbf{v},\mathbf{u}]]$. *The real root* t^* *of* p *can be found using Cardano's formula, which is a classical method to find roots of polynomials of degree 3.*

4.5 Experiments

4.5.1 Implementation details

Python implementations of SIG-tPGA and SIG-PGA (Algorithms 3 and 4) along with a notebook containing the following experiments to be reproduced are available online at `https://github.com/Raph-AI/pga-signature`. The computation of the signature is done with library `iisignature` [RG20]. To center the data on the signature space, we use the algorithm for computing the group mean (Definition 3.7) introduced in our previous work [Cla+24]. To compute $\nabla_{\mathbf{v}} F_{\mathbf{y}}$ as shown in Proposition 4.9, we need the differential of $\nabla_{\mathbf{v}} P_{\mathbf{v},\mathbf{y}}(t)$, which is obtained through automatic differentiation, with library `jax` [Bra+18]. Then the update step is performed using the Adam optimization strategy [KB17]. The projection step is done through exact computations if $L \in \{1,2,3,4\}$ (see Section 4.4.4) and otherwise if $L \geq 5$, we use root finder for polynomials `numpy.roots`.

4.5.2 Numerical results

In this section, we show how SIG-tPGA and SIG-PGA (Algorithms 3 and 4) solve various time series related tasks with performances close to the full signature (SIG) while using a much lower number of features. First, we solve a parameter estimation task for simulated fractional Ornstein–Uhlenbeck processes. Then, we forecast real data with a dataset of air quality measurements.

Simulated data

Parameter estimation in a pricing model.

The following method and example are adapted from [Lem+21]. We model a volatility process as $\sigma(t) := \exp(X(t))$ where X is a fractional Ornstein-Uhlenbeck (fOU) process:

$$dX(t) = -\alpha(X(t) - \mu)dt + \nu dW^H(t) \tag{4.48}$$

with real values $\alpha, \nu, \mu \geq 0$ and $W^H(t)$ a fractional Brownian Motion of Hurst exponent $H \in (0,1)$. In the following experiment, we simulate M datasets each composed of N time series $\{\hat{\sigma}^{i,j}(t)\}_{1\leq j\leq N}$ drawn from $\exp(\text{fOU}(\alpha_i))$ for $i = 1,\dots,M$ with α_i drawn from $\text{Unif}(0,1]$ and discrete time sampling $t = 1,\dots,T$. The goal is to estimate α_i from the observation of the N times series $\{\hat{\sigma}^{i,j}(t)\}_{1\leq j\leq N}$, for each $i = 1,\dots,M$. In the experiment we set $M = 50$, $N = 20$, $T = 200$. Also, we set $\mu = 0.5$, $\nu = 0.3$ and $H = 0.2$. Trajectories are time augmented, that is we consider the two dimensional time series $(\sigma(t), t)$. We show examples of simulated volatility time series in Figure 4.4.

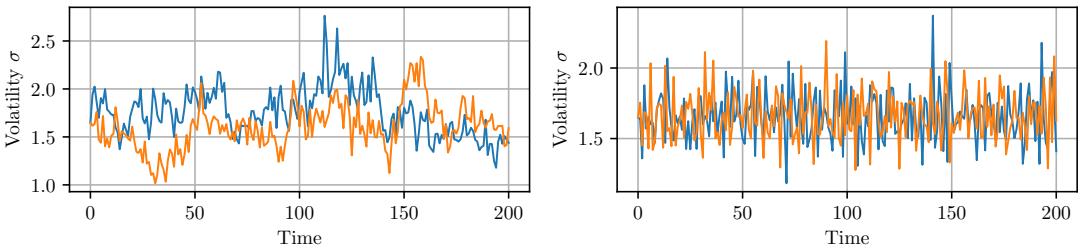


FIGURE 4.4: Simulated data from the pricing model. **Left**: $\alpha = 0.02$. **Right**: $\alpha = 0.98$.

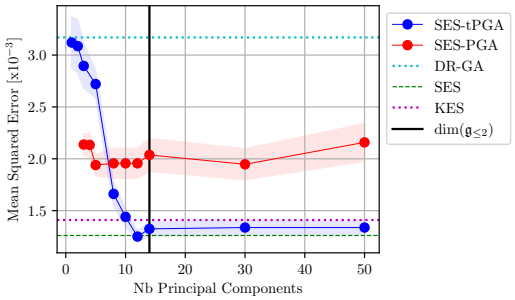


FIGURE 4.5: Results for the pricing model dataset. DR-GA, KES: adapted from [Lem+21].

To solve this parameter estimation task, we use the SES method (Signature of Expected Signature) introduced in [Lem+21]. The SES method proceeds as follows. Fix $1 \leq i \leq M$. We compute the signature on expanding windows applied to process $\hat{\sigma}^{i,j}(t)$, for $j = 1, \dots, N$. This gives $N \times (T-1)$ signatures. Then, we average those signatures and obtain $\bar{S}(t)$, a process composed of $T-1$ mean signatures:

$$t \mapsto \bar{S}(t) := \frac{1}{N} \sum_{j=1}^{N} \mathbf{S}_{\leq \ell}(W_{[1,t]}(\sigma^{i,j})) \tag{4.49}$$

where W is a windowing operation such that $W_{[1,t]}(x) := x|_{[1,t]}$ and the average of the signature is computed element-wise. Finally, we compute the signature of $\bar{S}$ up to level 2, i.e., $\mathbf{S}_{\leq 2}(\bar{S})$. In order to obtain α, we fit a linear regression model on $\mathbf{S}_{\leq 2}(\bar{S})$, with penalization (LASSO/Ridge).

To analyze the usefulness of the PGA, we add a dimension reduction step to the SES method: before fitting the linear model, we perform a PGA on $\mathbf{S}_{\leq 2}(\bar{S})$, for various number of components, and compare the results of the downstream estimation task to the results obtained without PGA. Further details on the SES method are presented in Appendix A.3.

Results for the pricing model dataset are shown in Figure 4.5. We can see that the MSE of SES and of PGA with 3 components are close. That is in comparison to DR-GA which is, among methods not using signature features, the one with the lowest MSE, and thus the only one displayed. In other words, signature features can be projected in a subspace of dimension 3 and be almost as insightful than 241 dimensions[1] for this estimation task. tPGA seems to yield similar results than PGA, even slightly improving when the number of components increases. Note that tPGA is faster to

[1]SES method best MSE was obtained with $\ell = 3$. We have $B_{3,2} = 15$ and $B_{2,15} = 241$ where $B_{L,d}$ is the dimension of the signature up to level L of a d-dimensional time series.

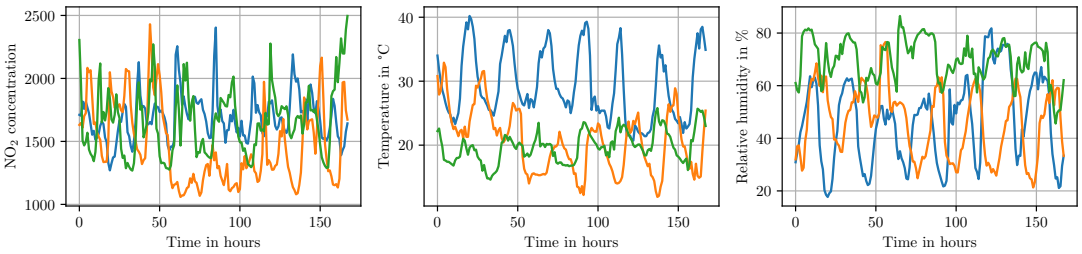


FIGURE 4.6: Air Quality data. Each color represents the same week of recording (168 hours).

compute than PGA and does not have convergence issues since tPGA relies on SVD whereas PGA relies on an optimization method on the tangent space $T_{\mathbf{1}}(G_{\leq L})$. Note that tPGA MSE after the vertical black line do not improve: that is because when we perform a tPGA, we project the data onto the tangent space and then perform a dimension reduction. Here, the tangent space is of dimension $\dim(\mathfrak{g}_{\leq 2})$.

Real data

Air quality forecasting.

The following example is adapted from [Fer22]. The Air Quality dataset [De +08] contains recordings of a sensor set in a polluted area in an Italian city. Data was recorded from March 2004 to February 2005 and is hourly averaged. Each time series have three components: nitrogen dioxide (NO_2) concentration, temperature and relative humidity. In Figure 4.6, we show three weeks of recordings.

The task is to predict the next value of NO_2 concentration, using the data from the previous week. To this end, we use the forecasting method introduced in [Fer22]. It consists in computing the signature of the data and then fit a linear regression model, with penalization (either LASSO or Ridge). This method can be seen as a simplification of the SES method presented in Section 4.5.2. The difference is that here we do not apply expanding windows and thus we do not average signatures. In our setting, we adapt this method by applying a PGA before performing the linear regression. Trajectories are time augmented, that is we consider the 4-dimensional time series $(X(t), t)$ where X is the air quality data, which has shown to perform better. The maximum number of Principal Components is $\min\{N, B_{L,4}\}$, where $B_{L,d}$ is the dimension of the signature up to level L of a d-dimensional time series. Since $N = 9189$ and we compute the signature up to order $L = 5$, the maximum number of Principal Components is $B_{5,4} = 1365$.

Forecasting results are shown in Figures 4.7 and 4.8. Along with the MSE of signature methods, we show the MSE of three other functional linear methods, as computed in [Fer22], each relying on the decomposition of the time series in one of the following basis: Fourier, B-Spline and fPCA (functional Principal Component Analysis). The number of basis elements is chosen, through cross-validation, between 1 and 13 for Fourier and B-Spline, and between 1 and 5 for fPCA. The fPCA is applied after a 7 B-Spline decomposition of the signals. In both figures, the vertical line is the number of features of the logsignature. That is, the lowest possible MSE of SIGℓ-PGA is obtained at #PCs = $\dim(\mathfrak{g}_{\leq \ell})$ and that is why the blue line stops at 10 (Figure 4.7) and 294 (Figure 4.8).

In Figure 4.7, SIG2 method uses 21 features. We show SIG2-PGA only up to 11 PCs because the optimization was not stable after with that many data points ($N \times B_{2,4} = 9189 \times 21$ that is 10^5 points).

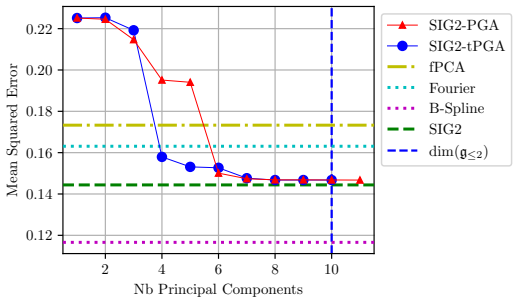


FIGURE 4.7: Results on the Air Quality dataset. Signature truncated at order 2.

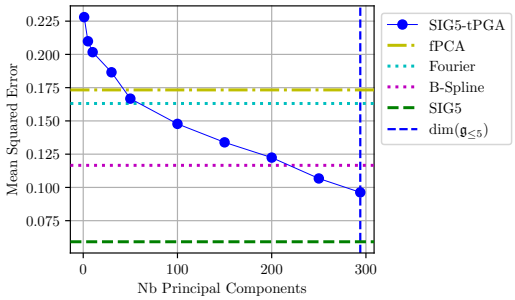


FIGURE 4.8: Results on the Air Quality dataset. Signature truncated at order 5.

In Figure 4.8, observe that with only 200 PCs, SIG5-tPGA gives performances as good as the best non signature method, B-Spline. While SIG5 method gives better performances, it uses 7 times more features (1365 features). Notice the gap in MSE between SIG5 and the best SIG5-tPGA result. This is because the logsignature sometimes gives poorer results compared to the signature and SIG5-tPGA MSE is bounded below by logsignature MSE, which is the point at the intersection of SIG5-tPGA and $\dim(\mathfrak{g}_{\leq 5})$. We do not show results for SIG5-PGA because with $N \times B_{5,4} = 9189 \times 1365 \approx 10^7$ data points, the method was too slow to compute.

Note that SIG-PGA methods perform better than fPCA, which is also a dimension reduction method for time series (with number of components tuned between 1 and 5 to obtain the best MSE). Thus, the SIG-PGA methods gives a better prediction,

Dimensions of the signature space $G_{\leq L}$ and of its tangent space $\mathfrak{g}_{\leq L}$ for various truncation levels L are shown in Appendix A.3.

4.6 Conclusion and perspectives

We have proposed an extension of the PCA for signature features of time series by means of an adaptation of the PGA. We have provided theoretical tools along with a numerical implementation to apply this new method. Also, we have shown through experiments, both on simulated and real data, that our approaches of dimension reduction are effective in that it keeps state-of-the-art performances while requiring less features.

Further work could be made on the dimension reduction problem for signatures, and especially the PGA. It would be interesting to have a computer algebra program that calculates the projection of a point on the geodesic with velocity $\mathbf{v}$ for any

truncation level $L \in \mathbb{N}$, that is, computes the real root of polynomial $p_{\mathbf{v},\mathbf{t}}$ as we have done by hand in Section 4.4.4 for $1 \leq L \leq 4$. It would allow for fast computations even for $L \geq 5$. Moreover, it would be interesting to implement a more efficient optimization method on $T_{\mathbf{1}}(G_{\leq L})$, which is the main bottleneck of our implementation of Algorithm 4.

Chapter 5

Anomaly detection using multiscale signatures

This chapter analyzes multidimensional time series through the lens of their integrals of various moment orders, constituting their signatures, a novel tool for detecting anomalies in time series. The proposed anomaly detection (AD) method is compared using classical distance-based methods such as Local Outlier Factor (LOF) and One-Class Support Vector Machine (OCSVM). These methods are investigated using different similarity measures: distance on signature features, Euclidean distance and Dynamic Time Warping (DTW). The combination of signature features with a specific segmentation of time series leads to a multiscale analysis tool that is competitive with respect to the state-of-the-art results, while maintaining low computational costs thanks to a property of the signature features.

5.1 Introduction

Context. Anomaly detection (AD) is a critical field of research with various applications in different fields (medical, telemetry, etc.) [Les+21; Pil+20]. In certain fields, the data to be analyzed is a set of time series. Along with the typical challenges of AD, anomalies in time series can appear in multiple forms such as global anomalies or contextual anomalies [Pil+20]. This raises the need for algorithms that can operate at multiple scales in order to identify the nature of anomalies and determine their exact location with a reasonable time complexity. Furthermore, specific challenges are related to the analysis of time series (irregularly sampled data, missing measurements, different recording lengths, etc.). These issues make the problem of AD in time series an active research field.

State-of-the-art. Several AD algorithms exist in the state-of-the-art [CBK09], such as Isolation Forest [LTZ12], Local Outlier Factor (LOF) [Bre+00] and One-Class SVM (OCSVM) [Sch+01]. These algorithms are well suited for comparing vectors in a space of fixed dimension. To compare time series with different lengths, similarity measures such as the Dynamic Time Warping (DTW) have been proposed in the literature [Men+19]. These techniques have already been incorporated in the standard AD algorithms for time series [Man+23]. However, they need to compute all the pairwise similarities, which is time consuming. This chapter proposes to analyze time series through the lens of their signature features.

Objectives, contributions and organization. An original approach for AD in time series is introduced in this chapter using standard AD algorithms and multiscale signatures features. The approach is compared to multiple state-of-the-art algorithms in terms of detection performance and time complexity. The contributions of this work are summarized below:

- We show that a multiscale analysis using signature features leads to state-of-the-art results for AD in multivariate time series.
- In addition to excellent detection results, we put in evidence the incredible numerical effectiveness of the multiscale signature based analysis. Notably, we avoid huge computational burden, that would appear with other similarity measures. The signature multiscale method can be 100 times faster than single-scale DTW.

Section 5.2 recalls the principles of LOF and OCSVM methods that are used in this chapter. Section 5.3 provides details on DTW and signature methods. Section 5.4 compares the proposed methods with state-of-the-art approaches on both synthetic and real datasets. Conclusions are reported in Section 5.5.

5.2 Anomaly Detection

The objective of AD is to detect abnormal behavior, i.e., data that deviates significantly from what is observed in the majority of cases. Abnormalities can represent different phenomena depending on the data that is analyzed. In general, abnormal data is scarce and cannot be used to describe all possible anomalies [CBK09]. Therefore, a model must be trained in an unsupervised way while considering an imbalance between the normal and abnormal classes. State-of-the-art AD algorithms include LOF and OCSVM that are considered in this chapter (other algorithms such as Isolation Forest and Density Based Spatial Clustering of Applications with Noise (DBSCAN) could be considered similarly).

5.2.1 Local Outlier Factor (LOF)

LOF [Bre+00] is an AD algorithm based on the density of the training data for classification. The more abnormal a data vector is, the larger its distance to its neighbors. Consider $N \geq 1$ vectors $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with $\mathbf{x}_n \in \mathbb{R}^D$, $n = 1, \dots, N$, D being the dimension of the vectors $\mathbf{x}_n$ and k a strictly positive integer. The distance between $\mathbf{x}_n$ and its k-th nearest neighbor is referred to as k-distance of $\mathbf{x}_n$ and denoted as $\mathrm{kd}(\mathbf{x}_n)$. The reachability distance between $\mathbf{x}_n$ and $\mathbf{x}_m$ is

$$\mathrm{rd}_k(\mathbf{x}_n, \mathbf{x}_m) = \max\{\mathrm{kd}(\mathbf{x}_n), \|\mathbf{x}_n - \mathbf{x}_m\|\} \,. \tag{5.1}$$

Therefore, all the points in the neighborhood of $\mathbf{x}_n$ have the same reachability but points that are further away will have a higher reachability. The local reachability density of $\mathbf{x}_n$ is the inverse of the average of all reachability distances of the k-nearest neighbors of $\mathbf{x}_n$:

$$\mathrm{lrd}_k(\mathbf{x}_n) = k \left(\sum_{\mathbf{x} \in V_n}^{N} \mathrm{rd}_k(\mathbf{x}, \mathbf{x}_n) \right)^{-1} \tag{5.2}$$

with V_n the subset containing all k-nearest neighbors of $\mathbf{x}_n$. The LOF score of $\mathbf{x}_n$ is the mean ratio of the local reachability densities of $\mathbf{x}_n$ with its k-nearest neighbors:

$$\mathrm{LOF}_k(\mathbf{x}_n) = \frac{1}{k} \sum_{\mathbf{x} \in V_n} \frac{\mathrm{lrd}_k(\mathbf{x})}{\mathrm{lrd}_k(\mathbf{x}_n)} \,. \tag{5.3}$$

Inliers have a LOF score close to 1 whereas outliers have a much higher LOF score.

5.2.2 One-Class SVM

Method

One-Class SVM is an AD algorithm, which learns a separating hyperplane close to the normal data in a certain sense. In its basic form, the data must be linearly separable in their vector space of the data. Consider a training set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ with $N \geq 1$ and $\forall i \in \{1, \dots, N\}, \mathbf{x}_i \in \mathbb{R}^D$ with D the number of data features. Finding a hyperplane separating the data can be expressed as

$$\begin{aligned} \underset{\omega, \rho, \xi_i}{\text{minimize}} \quad & \frac{1}{2}\|\omega\|^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i \\ \text{with} \quad & \langle \omega, \mathbf{x}_i \rangle \geq \rho - \xi_i, \forall i \in \{1, \dots, N\} \end{aligned} \tag{5.4}$$

where $\langle \omega, \mathbf{x}_i \rangle = \omega^T \mathbf{x}_i$ and ν is the maximum proportion of abnormal data in $\mathcal{X}$.

Kernel trick

When the data is not linearly separable, a reproducing kernel κ can be used. This kernel is associated with a scalar product in a higher dimensional space such that

$$\kappa(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \tag{5.5}$$

and $\forall i \in \{1, \dots, N\}, \Phi(\mathbf{x}_i) \in \mathbb{R}^F$ with $F \geq D$. The kernel function should be chosen in a way that the normal data is linearly separable from anomalies in the new space. The constraints in (5.4) are now applied to the transformed vectors:

$$\langle \omega, \Phi(\mathbf{x}_i) \rangle \geq \rho - \xi_i, \forall i \in \{1, \dots, N\} \tag{5.6}$$

with the new decision function

$$f(\mathbf{x}) = \text{sgn}(\langle \omega, \Phi(\mathbf{x}) \rangle - \rho) \tag{5.7}$$

where sgn denotes the “sign" function. The solution of (5.4) with the constraints (5.6) is known to be $\omega = \sum_{i=1}^{N} \alpha_i \Phi(\mathbf{x}_i)$, $0 \leq \alpha_i \leq \frac{1}{\nu N}$ s.t. $\sum_{i=1}^{N} \alpha_i = 1$ and the decision function is

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \rho\right) . \tag{5.8}$$

A common choice is the Gaussian kernel (or radial basis function) defined by $\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$ where σ is a hyper-parameter to be estimated. It can be shown that the corresponding function Φ projects the data into a space of infinite dimension.

5.2.3 Performance evaluation

Appropriate metrics must be defined to evaluate the performances of AD accurately. The measures of precision, recall and a combination of the two, the F_1 score, are usually considered for AD [SR15]. These measures are defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.9}$$

where TP, TN, FP and FN are the numbers of true positives, true negatives, false positives and false negatives. The F_1 score is the harmonic mean of precision and recall. Precision-Recall curves are often preferred to ROC curves for AD [SR15].

5.3 Comparing trajectories

5.3.1 Feature engineering: Signatures

Signatures can be useful for AD for at least two reasons:

- $\mathbf{S}(X)$ uniquely determines X up to translation and time reparametrization, except for pathological cases [HL10]. The invariance regarding time parametrization is in the following sense: given a non decreasing continuous surjection φ on $[0,T]$, denote as $\tilde{X} : [0,T] \to \mathbb{R}^d$; $t \mapsto X(\varphi(t))$. Then, $\mathbf{S}(X) = \mathbf{S}(\tilde{X})$. Thus the signature function $\mathbf{S}$ can be seen as a non linear filter, providing features invariant to translation and resampling.

- The dimension of $\mathbf{S}_{\leq L}(X)$ does not depend on the length T. Thus the signatures of two signals of different lengths T_1 and T_2 can be easily compared. Indeed, if one vectorizes the signatures at different levels, the associated signature vector, denoted as $\mathrm{vec}(\mathbf{S}_{\leq L}(X))$, is of dimension $\sum_{k=1}^{L} d^k = \frac{d(d^L-1)}{d-1}$.

Note that any non linear function of a multivariate time series can be arbitrarily well approximated by a linear function of its signature.

A similarity measure d_{sig} is defined by computing the Euclidean distance between two vectorized signatures, i.e.,

$$d_{\mathrm{sig}}(X,Y) = \|\mathrm{vec}(\mathbf{S}_{\leq L}(X)) - \mathrm{vec}(\mathbf{S}_{\leq L}(Y))\|_2 \,, \tag{5.10}$$

for two trajectories X and Y. Note that in some cases, normalizing the coefficients of the signature can improve AD performances. This normalization can be performed, e.g., by replacing $\mathbf{S}_{(k)}(X)$ with $(\mathbf{S}_{(k)}(X))^{1/k}$ where the exponent is applied element-wise, for any $k = 1, \ldots, L$.

Note that from a computational point of view, the computation of the signature is very simple thanks to Chen identity (Proposition 2.20). Indeed, one can compute the signature of a time series combining a piecewise linear approximation and Chen identity. More precisely, in order to handle time series (discrete set of points) a linear interpolation can be performed before calculating the signature. Because (2.34) depends on the concatenation operation, it is affected by the interpolation method and the linear interpolation leads to efficient numerical computations. Thereafter, for any time series X, $\mathbf{S}(X)$ denotes the signature of the linearly interpolated time series. Note that the signature has been used for AD in a semi-supervised context in [Sha+20] where an AD algorithm called `SigMahaKNN` (SMK) was based on anomaly scores using a generalized Mahalanobis distance on the signature features. This chapter proposes to detect anomalies in trajectories using signature features with a different and unsupervised approach, as explained below.

5.3.2 Multiscale signature feature

A powerful tool which allows for multiscale analysis is the hierarchical dyadic windowing operation defined in what follows. This operation is interesting in the case of signature features for two reasons: it has shown to improve classification results

[Mor+21] and it does not add computational overheads as explained below. Let $\{X(t_i)\}_{i=1,\dots,n}$ be a multivariate time series. Denote $\ell \in \mathbb{N}$ the hierarchical depth. For simplicity, assume that $2^{\ell-1}$ divides n. Denote as $W^i(X)$ the time series obtained after segmentation with a sliding window of length equal to $\frac{n}{2^{(i-1)}}$, for $i = 1, \dots, \ell$, as shown in Figure 5.1, which was presented in Section 2.3.2. Denote as $\mathrm{MSIG}(L) = \mathbf{S}_{\leq L} \circ W$ the

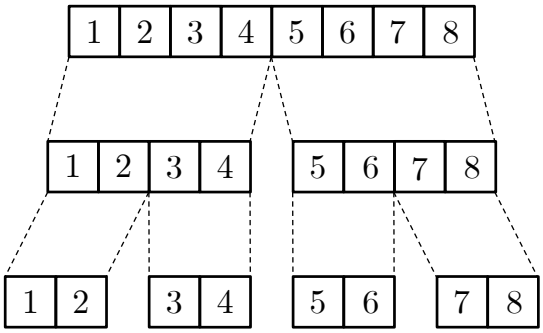


FIGURE 5.1: Dyadic windowing operation on a time series of length $n = 8$. We obtain the three rows by applying successively W^1, W^2 and W^3 to X.

signature computed on the segmented time series. The multiscale signature method for AD is defined by:

$$\mathrm{AD} \circ \mathrm{MSIG}(L). \tag{5.11}$$

The windowing operation W in MSIG does not add computational overheads since operations $\mathbf{S}_{\leq L}(X)$ and $(\mathbf{S}_{\leq L} \circ W)(X)$ have the same complexity. Indeed, the computation of the signature of X is done in two steps: 1) apply (2.9) on the $n/2$ pairs of the form $\{X(t_{2k}), X(t_{2k+1})\}$, 2) perform ℓ successive iterations where signatures are combined using (2.34). These two steps are illustrated in Figure 5.1 read from bottom to top: in the first iteration, pairs are combined by computing the product of the signatures of $\{X(t_1), X(t_2)\}$ and $\{X(t_3), X(t_4)\}$ using (2.34), which gives the signature of the time series $\{X(t_1), X(t_2), X(t_3), X(t_4)\}$. At the end of this process, the signature of the whole time series $\{X(t_i)\}_{i=1,\dots,n}$ is obtained. Therefore, to obtain $(\mathbf{S} \circ W)(X)$, it suffices to store all the signatures of sub time series calculated during the computation process of $\mathbf{S}(X)$.

Note that each time we go deeper in the dyadic segmentation, the more of the resampling invariance is lost.

5.3.3 Trajectory alignment : Dynamic Time Warping

Another way to compare two trajectories directly is to find which points of one trajectory *best* match the points of the other trajectory. This technique referred to as trajectory alignment requires to find associations or removes points from the two trajectories. A common tool used for trajectory alignment is DTW [KR05]. Consider $X = (X(1), \dots, X(n))$ and $Y = (Y(1), \dots, Y(m))$ two d-dimensional time series, i.e., $X(i) \in \mathbb{R}^d$ and $Y(j) \in \mathbb{R}^d$. The DTW cumulated similarity score between the first i columns of X and the first j columns of Y, denoted as $\mathrm{cs}(i, j)$, is defined as

$$\mathrm{cs}(i, j) = \min\{\mathrm{cs}(i-1, j-1), \mathrm{cs}(i, j-1), \mathrm{cs}(i-1, j)\} + \|X(i) - Y(j)\|^2. \tag{5.12}$$

5.4 Experiments

The proposed AD methods are based on computing signatures and applying the LOF or One-Class SVM methods with the distance d_{sig}. Two kinds of signatures are considered yielding the algorithms SIG4 (using $\mathbf{S}_{\leq L}(X)$ with $L = 4$) and MSIG4 (using $(\mathbf{S}_{\leq L} \circ W)(X)$ with $L = 4$). These methods are compared to DTW combined with LOF and DTW with One-Class SVM (OCSVM) to show the interest of using signatures, and to another AD algorithm `SigMahaKNN` (SMK). The number of neighbors in LOF was set to 9. Experiments can be reproduced using the notebook at `https://github.com/Raph-AI/anomaly-detection`.

5.4.1 Synthetic data

The first set of experiments consists in analyzing synthetic data representing trajectories in the context of abnormal ship behavior [PMF08]. The dataset contains $N_T = 1000$ sets of trajectories, each with 260 time series, 10 of them being abnormal according to the generated ground truth. The remaining 250 trajectories are divided into 5 groups (or railways) of 50 trajectories. The trajectories are each composed of 16 positions in a 2D space. Examples of trajectories are displayed in Figure 5.2. The method will not be compared to deep learning as there is too little data in this dataset for this method.

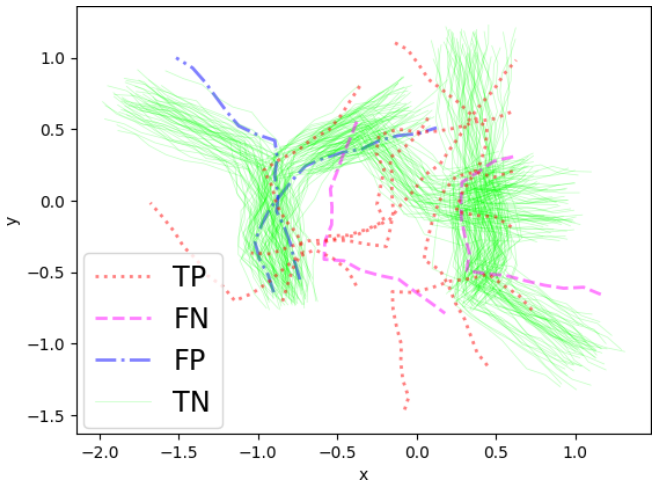


Figure 5.2: Examples of synthetic trajectories after prediction using LOF with SIG4 (2 FPs and 2 FNs).

Figure 5.3 shows the detection results using precision-recall curves for the different methods. The best detection results are obtained with LOF-DTW, LOF-MSIG4 and SMK4. However, MSIG4 is 10 times faster than DTW (see Section 5.4.3) and completely unsupervised whereas SMK4 is semi-supervised. Note that MSIG4 provides better results than SIG4, indicating that the windowing operation is indeed useful for AD. Results obtained with OCSVM from [Man+23] using DTW and SIG4 are also displayed showing a reduced detection performance.

5.4.2 Real data

This section analyzes data recovered from the unsupervised AD benchmark of the Harvard dataverse repository[1]. Dataset sizes and contamination rates are reported in

[1]Available at `https://doi.org/10.7910/DVN/OPQMVF`.

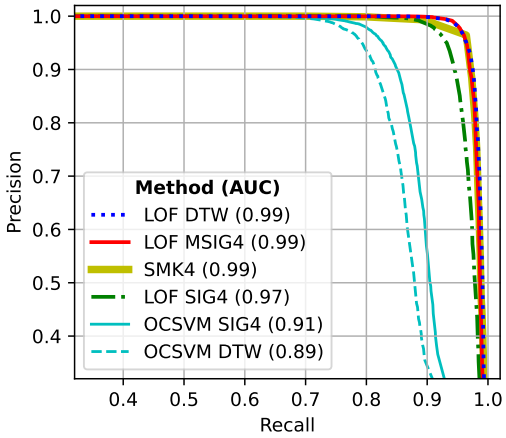


FIGURE 5.3: Precision-recall curves using LOF and OCSVM with various similarity measures.

Table 5.1. This dataset has been used in previous AD papers such as [Les+21] where a OCSVM with DTW baseline is provided. Table 5.2 shows the detection results of LOF using various similarity measures: Euclidean distance and DTW as baseline, and MSIG4. Overall, the MSIG provides promising AD results when compared to the baseline. Indeed, F_1 scores are larger for three datasets out of five, and comparable to the best score on the remaining two. Thus, MSIG seems to be consistently efficient for detecting anomalies in these datasets. Note that contamination rates have to be taken into account, as for instance, there are only ten outliers to detect in the Breast Cancer dataset. Thus the recall score can only take ten different values, leading to large discrepancies in the computed F_1 scores. The OCSVM method was also tested using the same similarity measures. Conclusions were similar to the ones obtained with LOF, but all detection scores were strictly inferior, similarly to the results of 5.4.1. An important aspect of the MSIG method is its low computational complexity, which is evaluated in what follows.

TABLE 5.1: Real datasets sizes and contamination rates.

Dataset name	**Nb time series**	**Nb points**	**Nb of anomalies**
ANN Thyroid	6916	21	250 (0.04%)
Breast Cancer	367	30	10 (2.72%)
Letter	1600	32	100 (6.25%)
Pen Global	809	16	90 (11.12%)
Satellite	5100	36	75 (1.47%)

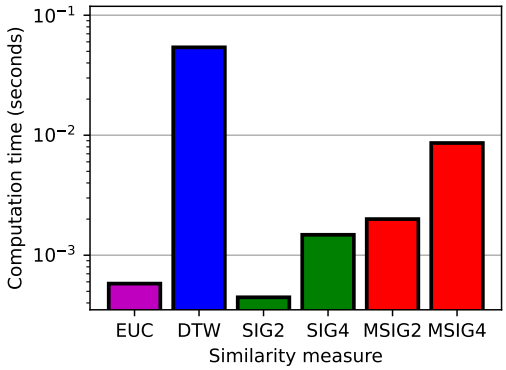
TABLE 5.2: F_1-scores using Local Outlier Factor (LOF) with 3 similarity measures and true contamination rates. Best in bold.

	EUC	DTW	MSIG4
ANN Thyroid	0.09	**0.22**	0.17
Breast Cancer	0.60	0.60	**0.70**
Letter	**0.55**	0.44	0.52
Pen Global	0.58	0.43	**0.59**
Satellite	**0.57**	0.55	**0.57**

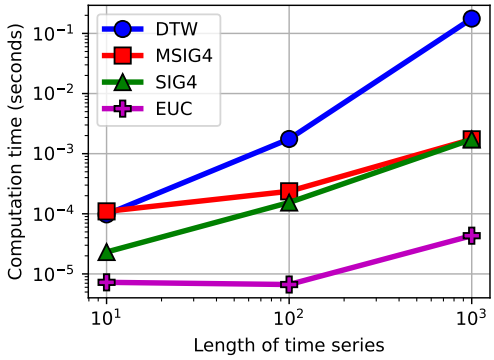
5.4.3 Computation time and memory space

Given a d-dimensional time series $(X(t))_{t=1,\dots,T}$ and a signature level $L \in \mathbb{N}$, the runtime complexity of the signature up to level L (i.e., of $\mathbf{S}_{\leq L}(X)$) is $O(Td^L)$ which has to be compared with the quadratic in length $O(T^2 d)$ runtime complexity of DTW. The space complexity of the signature is $O(d^L)$, which does not depend on T. Therefore, for large values of T, the signature can be viewed as a compression tool. Figure 5.4a displays runtimes for the computation of the pairwise similarity matrix. Note that the values shown for SIG/MSIG are the runtimes for the computation of the signature and the similarity matrix. The Euclidean distance (EUC) and SIG2 have complexities of the same order of magnitude. This is because the overhead of the computation of the signature is balanced with the Euclidean distance computation of a smaller number of signature coefficients. However, signature and multiscale signature methods are at least one order of magnitude faster than DTW.

Figure 5.4b compares runtimes on 10 randomly generated time series of dimension 2 for varying lengths $T \in \{10, 100, 1000\}$. As T increases, the overhead created by the multi-scaling vanishes explaining why SIG4 and MSIG4 have the same complexity. This figure also shows that the proposed AD approach based on the multiscale signature is numerically efficient, even for large datasets.



(A) Runtimes of pairwise distance matrix computations (synthetic data, 260 trajectories of size 2×16). SIG2 is the signature up to level 2. Log scale.



(B) Runtimes of pairwise distance matrix computation on randomly generated data. Each point: average runtime over 1000 iterations. Log-log scale.

FIGURE 5.4: Runtimes of the AD methods.

5.5 Conclusions and perspectives

This chapter studied a new anomaly detection (AD) method for time series based on their multiscale embedding in the space of signature features. A comparison between the proposed signature-based AD method and DTW on both simulated and real datasets showed similar detection performances. However, signature features can be computed multiple orders of magnitude faster than DTW, which is important for practical applications. The studied datasets had outliers that were globally abnormal. In the future, the proposed method should be applied to trajectories that contain collective anomalies (small abnormal segments), contextual anomalies or point anomalies [Pil+20]. A feature importance analysis could be used to detect which segmentation is most valuable to improve performances.

Chapter 6

Clustering multivariate time series with the signature

Clustering time series, i.e., detecting structures and patterns in temporal data is challenging for multiple reasons. For instance, methods have to take into account the temporal dynamic of the data, the possible high dimensionality and also artifacts such as variable lengths, noise, outliers. Several techniques have been developed to address these challenges, often categorized as raw-data-based, feature-based (e.g., wavelet transform, spectrum, signatures) and model-based (i.e., time series are assumed to be generated from an unknown underlying probability density function that we try to approximate). For each technique, the choice of the clustering algorithm (e.g., K-means, agglomerative hierarchical) and of the distance measure (e.g., Euclidean, correlation, DTW) is crucial. Surveys on time series clustering can be found in [Lia05; ASW15].

Following the recommended methodology on evaluation method for clustering algorithms [KK03], we compare signature-based clustering to raw-data-based and spectrum-based clustering methods, as those last two strategies when coupled with simple distance measures (Euclidean, correlation, DTW) often achieve state-of-the-art results and are favored for their interpretability, simplicity and stability. We discuss strategies to measure the similarity between two signatures and analyze their scaling or orthogonal transformation invariances. Also, we show through experiments how similarity measures on signatures perform compare to usual similarities on time series. In order to prevent implementation and data bias [KK03], we use a wide range of datasets, simulated and real, in which we artificially add perturbations (noise, jittering, warping) which are standard for learning tasks [IU21].

6.1 Similarity measures for signature features

In addition to the previous remarks made in Section 2.2, we would like to point out some other aspects of the signature that renders it especially suited for learning tasks. First, the signature transform is inherently suited for multidimensional time series, whereas a lot of time series analysis methods focus on the one-dimensional case and lacks of a generalization to higher dimensions. Here, we will only deal with multivariate time series. Moreover, the signature is well suited to several usual transformations, as shown in Table 6.1, such as the invariance to translation or the invariance to time reparametrizations, which are often needed as undesired translations (e.g., not well calibrated sensor) and time shifts (e.g., short sensor failure, *censored* data) can easily occur in real data. Note that time series of various lengths can be compared (e.g., sensor with various frequency of recording). In addition, we can set norms on signatures that are invariant to dilations or rotations, see Section 6.1.2.

Finally, if needed input time series can be preprocessed in order to avoid specific invariances. For instance, to loose the translation invariance, a time series $X = (X(t_1), \dots, X(t_T))$ can be transformed into $\widetilde{X} = (0, X(t_1), \dots, X(t_T))$. To loose the invariance to time reparametrization, it suffices to artificially add a new dimension to X which consists of $\widetilde{X}^{d+1} = (t_1, \dots, t_T)$. This makes the signature a versatile tool for time series clustering

Transformation	Signature after transformation
$\widetilde{X}(t) := X(t) + a$	$\mathbf{S}(\widetilde{X}) = \mathbf{S}(X)$
$\widetilde{X}(t) := X(t) + a(t)$	No reduced form
$\widetilde{X}(t) := X(t) \odot a(t)$	No reduced form
$\widetilde{X}(t) := X(t) \star a(t)$	$\mathbf{S}(\widetilde{X}) = \mathbf{S}(X)\mathbf{S}(a)$
$\widetilde{X}(t) := \alpha X(t)$	See section 6.1.2
$\widetilde{X}(t) := AX(t)$	$\mathbf{S}_{(k)}(\widetilde{X}) = \mathbf{S}_{(k)}(X) \bullet_1 A \bullet_2 \cdots \bullet_k A$
$\widetilde{X}(t) := QX(t)$	See section 6.1.2
$\widetilde{X}(t_i) := X(t_{T-i})$	$\mathbf{S}(\widetilde{X}) = \mathbf{S}^{-1}(X)$
$\widetilde{X}(t) := X(\varphi(t))$	$\mathbf{S}(\widetilde{X}) = \mathbf{S}(X)$

TABLE 6.1: Usual transformations of time series and the corresponding signature, where X is a d-dimensional time series, $a \in \mathbb{R}^d$, $a(t) \in \mathbb{R}^d$, $\alpha \in \mathbb{R}$, A matrix of size $d \times d$, Q orthogonal matrix of size $d \times d$, φ warping.

6.1.1 Definitions

We now introduce two measures of similarity for signature features, denoted as $d_{\text{SIG-A}}$ and $d_{\text{SIG-B}}$. We define $d_{\text{SIG-A}}$, for any d-dimensional time series X and Y, and for any signature level $L \in \mathbb{N}$, as

$$d_{\text{SIG-A}}(X, Y) := \sum_{k=1}^{L} \|\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y)\|_F^{1/k}. \tag{6.1}$$

and we define $d_{\text{SIG-B}}$ as

$$d_{\text{SIG-B}}(X, Y) := \|\phi_{sc}(\mathbf{S}(X)) - \phi_{sc}(\mathbf{S}(Y))\|_F \tag{6.2}$$

$$= \sqrt{\sum_{k=1}^{L} \|(\mathbf{S}_{(k)}(X))^{1/k} - (\mathbf{S}_{(k)}(Y))^{1/k}\|_F^2} \tag{6.3}$$

where the last equality follows from the definition of the norm on the tensor algebra $T(\mathbb{R}^d)$ (Equation (2.36)) and where $\phi_{sc} : T(\mathbb{R}^d) \to T(\mathbb{R}^d)$ is such that for any $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \dots) \in T(\mathbb{R}^d)$,

$$\phi_{sc}(\mathbf{a}) = (\mathbf{a}_1, (\mathbf{a}_2)^{1/2}, \dots, (\mathbf{a}_k)^{1/k}, \dots) \tag{6.4}$$

with $(\mathbf{a}_k)^{1/k}$ the element-wise scalar power.

In both definitions of $d_{\text{SIG-A}}$ and $d_{\text{SIG-B}}$, we do not rely on group operations because the computational cost would be too heavy. Indeed, using for instance $d(X, Y) = \|(\mathbf{S}(X))^{-1}\mathbf{S}(Y)\|$, as suggested in [FV10, Proposition 7.36], rely on the group product (Equation (2.18)) and the group inverse (Equation (2.20)). Those two group operations are multiple orders of magnitude slower than the subtraction in

the tensor algebra $\mathbf{S}(X) - \mathbf{S}(Y)$ (Equation (2.16)) that we use here, i.e., would not be competitive.

Instead, we rely on the usual generalization of the Frobenius inner product to the tensor algebra (see Equation (2.36)). For signature tensors, this generalization needs to be adapted since for any continuous X of finite variation, we have for any $k \in \mathbb{N}$,

$$\|\mathbf{S}_{(k)}(X)\|_F \leq \frac{(C_X)^k}{k!}, \tag{6.5}$$

with C_X a constant depending on X, see [Bon+19, Proposition A.5]. The scaling $\mathbf{a}_k \leftarrow k!\mathbf{a}_k$ can be applied (see [Mor+21, Section 3.4.]) but we have found that the $\mathbf{a}_k \leftarrow (\mathbf{a}_k)^{1/k}$ scaling leads to better results, especially when we only use the first levels of the signature, $L \leq 10$. In addition, it is the simplest way to obtain a similarity measure homogeneous to dilation [FV10, Example 7.37], see Section 6.1.2 below. Then, we can choose to apply the scaling before or after the computation of the norm. In Equation (6.1), the scaling is applied after ($d_{\text{SIG-A}}$) and in Equation (6.2), the scaling is applied before $d_{\text{SIG-B}}$.

6.1.2 Properties

The next proposition below analyzes the behavior of $d_{\text{SIG-A}}$ regarding dilations and rotations. Geometrically speaking, it is natural to require that our similarity measure stays constant as two elements are rotated together. For instance, the distance between two time series of handwritten digits should stay the same if the digits are all rotated.

Proposition 6.1. *For any positive scalar α and two time series X, Y, we have*

$$d_{SIG\text{-}A}(\alpha X, \alpha Y) = \alpha d_{SIG\text{-}A}(X, Y), \tag{6.6}$$

and for any Q orthogonal matrix of size $d \times d$,

$$d_{SIG\text{-}A}(QX, QY) = d_{SIG\text{-}A}(X, Y). \tag{6.7}$$

Proof. See Section 6.1.3. □

As with $d_{\text{SIG-A}}$, we also have the $\mathbb{R}$-linearity for $d_{\text{SIG-B}}$, see Proposition 6.2, but we loose the invariance to orthogonal transformations.

Proposition 6.2. *$d_{SIG\text{-}B}$ is homogeneous with respect to positive scalars, that is, for any positive scalar α and two time series X, Y, we have*

$$d_{SIG\text{-}B}(\alpha X, \alpha Y) = \alpha d_{SIG\text{-}B}(X, Y). \tag{6.8}$$

Proof. See Section 6.1.3. □

Remark 6.3. *In other words, in Equation (6.2), we scale the signature coefficients and then compute the norm whereas in Equation (6.1) the two operations are performed in the reverse order. One advantage of $d_{SIG\text{-}B}$ compared to $d_{SIG\text{-}A}$ is its numerical efficiency, because we do not need to loop over the signature levels $k = 1, \dots, L$ and apply a scaling. $d_{SIG\text{-}B}$ only rely on Euclidean distance implementations, which are very efficient (e.g., `pdist()` function of python library scipy).*

The two similarities $d_{\text{SIG-A}}$ and $d_{\text{SIG-B}}$ are compared to the usual similarities (Euclidean, correlation, DTW) in clustering experiments in the following section.

6.1.3 Proofs

In order to prove Proposition 6.1, we need the following result.

Lemma 6.4. *Let* $\mathbf{u} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *be a tensor and* Q *be a* $d \times d$ *orthogonal matrix. For any* n*,*

$$\|\mathbf{u} \bullet_n Q\|_F = \|\mathbf{u}\|_F. \tag{6.9}$$

Proof. Denote $\mathbf{v} = \mathbf{u} \bullet_n Q$. We have $\mathbf{v}_{(n)} = Q\mathbf{u}_{(n)}$, see [KB09, page 461]. Thus,

$$\begin{aligned}
\|\mathbf{v}\|_F^2 &= \|\mathbf{v}_{(n)}\|_F^2 && (6.10)\\
&= \|Q\mathbf{u}_{(n)}\|_F^2 && (6.11)\\
&= (Q\mathbf{u}_{(n)})^T Q\mathbf{u}_{(n)} && (6.12)\\
&= \mathbf{u}_{(n)}^T Q^T Q\mathbf{u}_{(n)} && (6.13)\\
&= \mathbf{u}_{(n)}^T \mathbf{u}_{(n)} && (6.14)\\
&= \|\mathbf{u}\|_F^2. && (6.15)
\end{aligned}$$

□

Now, we can prove Proposition 6.1 and also Proposition 6.2.

Proof of Proposition 6.1. Using the linearity of the signature transform, we have

$$\begin{aligned}
d_{\text{SIG-A}}(\alpha X, \alpha Y) &= \sum_{k=1}^{L} \|\mathbf{S}_{(k)}(\alpha X) - \mathbf{S}_{(k)}(\alpha Y)\|_F^{1/k} && (6.16)\\
&= \sum_{k=1}^{L} \|\alpha^k (\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y))\|_F^{1/k} && (6.17)\\
&= \alpha d_{\text{SIG-A}}(X, Y) && (6.18)
\end{aligned}$$

which gives Equation (6.6). Moreover, we have for any integer k,

$$\begin{aligned}
\|\mathbf{S}_{(k)}(QX) - \mathbf{S}_{(k)}(QY)\|_F &= \|(\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y)) \bullet_1 Q \bullet_2 Q \cdots \bullet_k Q\|_F && (6.19)\\
&= \|\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y)\|_F && (6.20)
\end{aligned}$$

where we have used in the last equality that for any tensor $\mathbf{u}$ and orthogonal matrix Q, $\|\mathbf{u} \bullet_n Q\|_F = \|\mathbf{u}\|_F$ for any integer n (see Lemma 6.4). Thus,

$$\sum_{k=1}^{L} \|\mathbf{S}_{(k)}(QX) - \mathbf{S}_{(k)}(QY)\|_F^{1/k} = \sum_{k=1}^{L} \|\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y)\|_F^{1/k}. \tag{6.21}$$

□

Proof of Proposition 6.2. Denote the coordinates of $\mathbf{S}(X)$ as $\mathbf{S}(X) = (\mathbf{x}_1, \mathbf{x}_2, \dots)$. We have

$$\begin{aligned}
\phi_{sc}(\mathbf{S}(\alpha X)) &= \phi_{sc}((\alpha\mathbf{x}_1, \alpha^2\mathbf{x}_2, \dots)) && (6.22)\\
&= (\alpha\mathbf{x}_1, \alpha(\mathbf{x}_2)^{1/2}, \dots, \alpha(\mathbf{x}_k)^{1/k}, \dots) && (6.23)\\
&= \alpha\phi_{sc}(\mathbf{S}(X)). && (6.24)
\end{aligned}$$

□

6.2 Experiments

Before presenting the methodology and experiments, we introduce some operations that will be used to artificially add noise to the data.

6.2.1 Perturbation of the data

In this section, we present the transformations `crop`, `noise` and `warp` that will be used in the experiments to assess the robustness of the signature method. Those will be compared to results on the raw time series. Those transformations of the raw data are natural as they often occur in real life. For instance, a sensor can stop recording for a few seconds and thus produce missing values, or the starting time of recording can vary between sensors. Note that those transformations of time series are standard as an augmentation tool to improve learning performances of neural networks [IU21].

- `crop` : given a time series $X = \{X(1), \dots, X(T)\}$, we have

$$\texttt{crop}(X) = \{X(1+k), \dots, X(T-K+k)\} \tag{6.25}$$

  where K is a fixed integer (the cropping size, e.g. $K := \lfloor T/10 \rfloor$) and k is a random integer drawn uniformly between 0 and $K-1$. For each time series of the dataset, we draw a different k. This procedure has the effect to produce time shifts and thus remove time synchronization of each curve with respect to the others. The `crop` operation is illustrated in Figure 6.1.

- $\texttt{noise}_\sigma$: to each data point $X(t) \in \mathbb{R}^d$ we apply

$$\texttt{noise}_\sigma(X(t)) = X(t) + \varepsilon(t) \tag{6.26}$$

  where $\varepsilon(t) \sim N(0, \sigma I_d)$.

- $\texttt{warp}_\sigma$: we generate a random continuous bijection $\varphi_\sigma : [0,1] \to [0,1]$ where σ is an amplitude parameter and to X we apply

$$\texttt{warp}(X(t)) = X(\varphi_\sigma(t)) \,. \tag{6.27}$$

  This produce a time dilation (or shrinkage), and as with operation `crop`, it can remove synchronizations between time series of the dataset. It can also imitate the presence of missing values in the data (when no imputation has been performed). Note that a random warping can change the convexity of the input trajectory X, as shown in Figure 6.2.

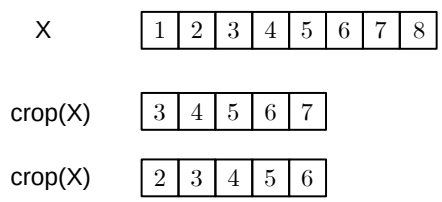


FIGURE 6.1: Operation `crop` performed two times on a time series composed of eight observations.

We now present our methodology and experiments.

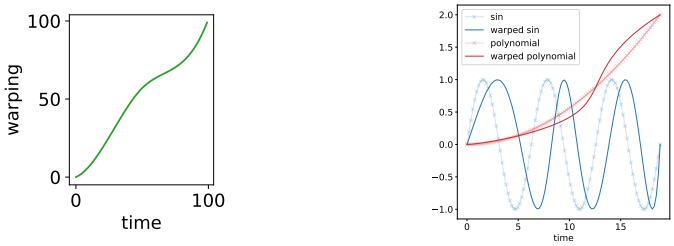


FIGURE 6.2: **Left**: a warping function φ. **Right**: Two functions X, Y and their corresponding warping trajectory $t \mapsto X(\varphi(t))$ and $t \mapsto Y(\varphi(t))$.

6.2.2 Clustering methodology

First, we clusterize data simulated from a vector autoregressive (VAR) model, then several real datasets. The clustering procedure goes as follows. Let $X_1, \dots, X_N$ be N time series such that $X_i(t_k) \in \mathbb{R}^d$, for all $i = 1, \dots, N$ and for all $k = 1, \dots, T$. To each time series X_i corresponds a label $y_i \in \mathbb{N}$. The goal is to retrieve the true groups with an unsupervised method. We proceed as follows. We perform either a spectral clustering or a K-means clustering procedure coupled with the following similarities: correlation, Euclidean distance, Dynamic Time Warping (DTW), $d_{\text{SIG-B}}$ and $d_{\text{SIG-A}}$. The first three similarities are computed directly on the time series whereas $d_{\text{SIG-B}}$ and $d_{\text{SIG-A}}$ are computed on signatures computed up to level $L \in \mathbb{N}$, L being a hyperparameter to learn. Details on spectral clustering and K-means are given in Appendix B. Note that the K-means procedure, when performed in the signature space, is adapted in order to use the signature averaging method introduced in Chapter 3. For each dataset, we clusterize the raw data (i.e., without any tinkering) and perturbed data, that is where we have used operations `crop`, `noise` and `warp` introduced in the previous section. The Adjusted Rand Index (ARI) is used as the main metric here, which is a classical time series clustering evaluation measure [ASW15, Section 6].

We use the following python libraries: DTW is computed using `fastdtw`, signatures are computed with `iisignature` [RG20], K-means is computed using `sklearn` [Ped+11] and `tslearn` [Tav+20]. We use our own implementations for spectral clustering and K-means with signatures.

6.2.3 Simulated data

We use the following vector autoregressive model of lag 1, i.e., VAR(1) model:

$$X(t+1) = AX(t) + U(t+1) \tag{6.28}$$

where A, the transition matrix, is of size $d \times d$ and $U(t)$ are iid Gaussian white noise vectors with covariance matrix ς such that diagonal coefficients $\varsigma_{i,i} = 1$ and non diagonal coefficients $\varsigma_{i,j} = \rho \in (0,1)$, $i \neq j$. The VAR model is useful to describe empirical data from various fields, e.g., sociology, economics, physics and clustering VAR data (as well as related models, e.g., VARMA, VARIMA) has proven to be challenging [KGP01].

In order to generate K distinct groups of time series using Model (6.28), we first construct K matrices $A^{(1)}, \dots, A^{(K)}$. This is done by randomly drawing coefficients of $A^{(k)}$: $A^{(k)}_{i,j} \sim \mathcal{N}(0,1)$, for any $1 \leq i, j \leq d$. Then, for cluster k, we generate N time series using Model (6.28) with transition matrix $A^{(k)}$.

We use the following parameters : $K = 3$ clusters, $N = 10$ time series per cluster, $d = 2$ dimensions, $T = 20$ time points, signature level $L = 4$. To illustrate our model, we show in Figure 6.3 one instance of each cluster. We also cluster the differentiated time series $\Delta X(t_i) = X(t_{i+1}) - X(t_i)$ for all $i = 1, \dots, T$.

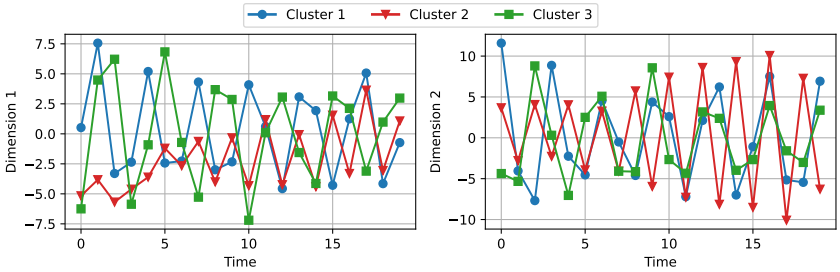


FIGURE 6.3: Bidimensional time series simulated using Model (6.28). Only one instance of each cluster is shown.

Clustering results are shown in Table 6.2 and in Table 6.3 for the differentiated data, with medians ARI (over 200 simulated sets).

We can see that the best results are achieved by spectral clustering with $d_{\text{SIG-B}}$. Interestingly, there is a large gap of performances between $d_{\text{SIG-A}}$ and $d_{\text{SIG-B}}$, which indicates that the choice of the similarity measure between signatures is crucial. The choice of the clustering method that we perform in the signature space is also crucial since performances with signature K-means are largely inferior.

Regarding the perturbed data, all the methods seem to be similarly impacted by operations `crop`, `noise` and `warp`. This indicates that the signature seems to be as robust as the standard methods on this type of data. Note that DTW is the only method that achieve better results on the differentiated data ΔX compared to results obtained on X.

Overall, the signature seems to be especially suited for VAR data compared with usual similarity measures for time series.

		raw	crop	noise_3	noise_{10}	$\text{warp}_{0.05}$	$\text{warp}_{0.5}$
Spectral clustering	Correlation	52	45	42	19	42	9
	Euclidean	15	14	14	8	16	15
	DTW	29	25	25	11	23	17
	SIG-B	90	81	51	17	55	13
	SIG-A	47	40	28	4	25	8
K-means	Euclidean	12	10	13	8	9	5
	DTW	25	25	23	11	17	12
	SIG-B	13	13	14	13	14	13
	SIG-A	8	8	8	9	8	9

TABLE 6.2: ARI (%) after clustering the VAR(1) data.

		raw	crop	noise_3	noise_{10}	$\text{warp}_{0.05}$	$\text{warp}_{0.5}$
Spectral clustering	Correlation	42	44	42	18	42	8
	Euclidean	25	13	14	8	16	15
	DTW	45	25	25	13	26	17
	SIG-B	90	86	49	19	52	15
	SIG-A	42	42	29	6	24	8
K-means	Euclidean	12	10	11	9	9	6
	DTW	27	23	24	11	16	10
	SIG-B	14	14	14	14	14	14
	SIG-A	9	9	8	9	9	9

TABLE 6.3: ARI (%) after clustering the **differentiated** VAR(1) data.

6.2.4 Real data

We use multivariate time series from the UEA repository [Bag+18][1]. As shown in Table 6.4, the datasets that we have selected for this analysis have various lengths (between 51 and 1197) and various number of dimensions (between 2 and 24).

Dataset	N	**Length** T	**Number of dimensions** d	**Number of clusters** K
ArticularyWordRecognition	575	144	9	25
Cricket	180	1197	6	12
ERing	300	65	4	6
Libras	360	45	2	15
NATOPS	350	51	24	6

TABLE 6.4: Datasets used for clustering.

In the following list, we give further information regarding the data. More exhaustive details can be found in [Bag+18].

- ArticularyWordRecognition: Electromagnetic Articulogram, or in other words, measurement of the movement of the tongue and lips during speech. Each class corresponds to the pronunciation of a different word. Out of the 12 tridimensional sensors, only 9 dimensions are kept.
- Cricket: recordings from tridimensional accelerometers placed on each wrist of cricket referees (i.e., 6 dimensions). Each of the 12 classes corresponds to a specific movement of the hands (signaling a foul, a request for video replay, etc.).
- ERing: finger ring with electric field sensing. Each class corresponds to a different posture of the hand.
- Libras: point mapping on a video recording. Each class corresponds to a specific hand movement.
- NATOPS: tridimensional recordings of sensors places on each hands, elbows, wrists and thumbs of an aircraft ground operator, i.e., $8 \times 3 = 24$ dimensions. Each class corresponds to a gesture command (spread wings, fold wings, etc.).

We use the following hyperparameters: signature level $L = 4$, lead-lag order 3 and time augmentation (see Section 2.3.2 for the definition of those parameters).

Results are shown in Tables 6.5 to 6.9. Note that for each method, two ARI values are given: each one corresponds to a different parameter K (number of clusters). In light of the results, several remarks are in order. First, for most raw data clustering results, DTW-based methods obtained the best ARI, but the signature (especially, spectral clustering with SIG-B) come in as a close second. In the case of Libras data, signature is first.

Signature-based methods are more robust to `crop` (i.e., jittering) than the correlation or the euclidean distance. Note that this is also the case of DTW, as expected since it was originally designed to handle desynchronization. Moreover, signature-based method are robust to noise only with the K-means procedure and not with the spectral clustering. In the case of the Libras data, the signature is the most robust

[1] Available at `https://timeseriesclassification.com/`.

method, especially to noise. Regarding the `warp` transformation, it seems to affect less signature-based methods than the other methods, even though the data have been time augmented (i.e., we have lost the signature reparametrization invariance). Note that results on the signature without time augmentation were largely inferior and thus not shown here.

If we compare signature-based methods, it seems that spectral clustering lead to the best results compared to K-means. However, as mentioned above, K-means is more robust in the presence of noise. Thus, a denoising procedure should be applied when using the former methods. Moreover, if we compare normalization procedures SIG-B and SIG-A, the former has the best ARI for AWR and Cricket data, and the latter has the best ARI for the ERing, Libras data. Thus, this indicates that the choice of normalization of signature features is important and depends on the data.

Regarding the normalization procedure of signature features, note that other methods were tried such as $\sum_{k=1}^{L} \|(\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y))^{1/k}\|_F^2$ and $\sum_{k=1}^{L} w_k \|\mathbf{S}_{(k)}(X) - \mathbf{S}_{(k)}(Y)\|_F^2$ with $w_k = \frac{1}{\|S_{(k)}(X)\|_F^2}$. Those normalization lead to poorer results than those presented here.

A few remarks regarding computation times can be made in light of the runtimes shown in Figure 6.4. We can see that clustering methods that rely on the DTW similarity measure are slow when the length T of time series is large (datasets ArticularyWordRecognition and Cricket) which is expected since the DTW algorithm has a quadratic complexity in the length $O(T^2 d)$. Signature-based methods have large runtimes when the number of dimensions d is high (NATOPS). This is due to the number of features that grows exponentially in the number of dimension: as stated earlier, the signature has complexity $O(T d^L)$ in the dimension, with L the signature level, here fixed to $L = 4$.

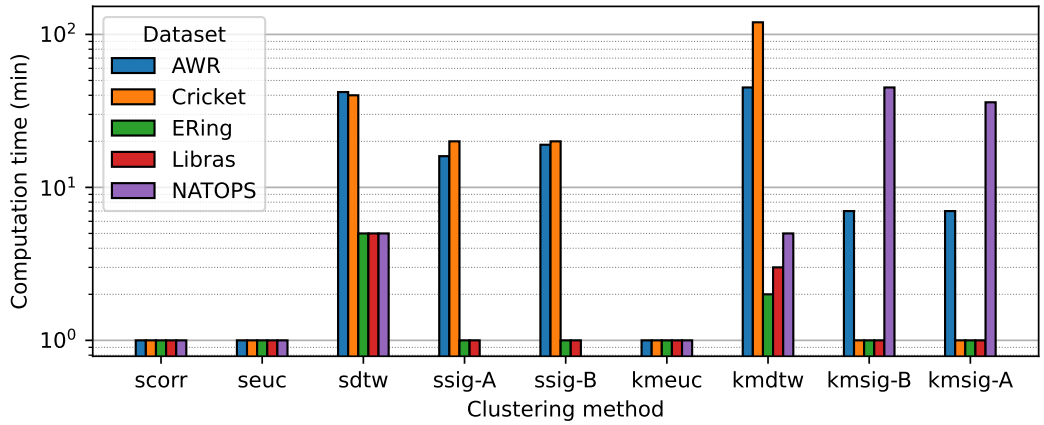


FIGURE 6.4: Computation times of clustering methods on real datasets. Spectral clustering is denoted with an 's' in the method name and K-means clustering with 'km'.

6.3 Conclusion

In this chapter, we have compared clustering procedure in time series space with clustering in the signature space. We have seen that for the standard VAR(1) model, learning in the signature space leads to better results. In the case of the real datasets,

TABLE 6.5: ARI (%) after clustering **ArticularyWordRecognition** data.

		raw	$\texttt{crop}_{10\%}$	$\texttt{noise}_{\sigma}$	$\texttt{noise}_{3\sigma}$	$\texttt{warp}_{0.05}$	$\texttt{warp}_{0.5}$
Spectral clustering	Correlation	80; 76	70; 67	78; 75	69; 61	77; 72	10; 9
	Euclidean	83; 85	77; 78	82; 82	65; 38	83; 81	10; 10
	DTW	96; 85	85; 85	93; 83	61; 49	92; 84	10; 14
	SIG-B	88; 78	84; 79	25; 29	0; 0	82; 77	20; 19
	SIG-A	76; 73	77; 79	36; 18	2; 1	77; 73	43; 39
K-means	Euclidean	82; 79	74; 75	77; 77	14; 15	85; 80	13; 10
	DTW	91; 88	94; 84	85; 79	29; 28	92; 86	9; 11
	SIG-B	37; 69	67; 62	55; 27	37; 69	48; 65	68; 68
	SIG-A	51; 63	64; 65	57; 45	51; 63	57; 68	58; 63

TABLE 6.6: ARI (%) after clustering **Cricket** data.

		raw	$\texttt{crop}_{10\%}$	$\texttt{noise}_{\sigma}$	$\texttt{noise}_{3\sigma}$	$\texttt{warp}_{0.05}$	$\texttt{warp}_{0.5}$
Spectral clustering	Correlation	58; 61	41; 40	45; 59	35; 30	54; 59	3; 2
	Euclidean	62; 58	49; 48	62; 56	54; 51	66; 58	4; 6
	DTW	100; 90	100; 87	98; 91	45; 43	98; 88	10; 12
	SIG-B	95; 82	95; 87	18; 21	0; 1	28; 27	6; 8
	SIG-A	73; 71	72; 74	15; 14	0; 0	32; 39	11; 11
K-means	Euclidean	62; 67	55; 49	59; 62	28; 36	62; 69	4; 4
	DTW	100; 92	100; 93	87; 92	69; 81	98; 88	10; 9
	SIG-B	17; 19	10; 12	15; 17	17; 13	17; 19	24; 15
	SIG-A	13; 18	17; 20	18; 20	19; 20	13; 18	15; 16

TABLE 6.7: ARI (%) after clustering **ERing** data.

		raw	$\texttt{crop}_{10\%}$	$\texttt{noise}_{\sigma}$	$\texttt{noise}_{3\sigma}$	$\texttt{warp}_{0.05}$	$\texttt{warp}_{0.5}$
Spectral clustering	Correlation	70; 58	53; 45	66; 57	26; 22	66; 47	11; 9
	Euclidean	89; 71	78; 63	85; 68	49; 35	81; 66	15; 19
	DTW	86; 69	85; 66	85; 70	59; 39	85; 74	14; 32
	SIG-B	39; 54	40; 52	26; 22	1; 1	34; 45	24; 21
	SIG-A	57; 47	51; 47	21; 17	2; 3	59; 44	50; 36
K-means	Euclidean	90; 70	77; 64	87; 65	44; 32	77; 66	21; 17
	DTW	91; 69	88; 66	76; 63	28; 18	87; 74	30; 22
	SIG-B	21; 34	21; 34	23; 26	17; 30	19; 29	19; 28
	SIG-A	46; 40	46; 40	45; 46	50; 48	48; 46	45; 36

TABLE 6.8: ARI (%) after clustering **Libras** data.

		raw	$\texttt{crop}_{10\%}$	$\texttt{noise}_{\sigma}$	$\texttt{noise}_{3\sigma}$	$\texttt{warp}_{0.05}$	$\texttt{warp}_{0.5}$
Spectral clustering	Correlation	34; 32	28; 31	24; 23	2; 2	31; 31	13; 11
	Euclidean	34; 32	32; 33	31; 24	4; 3	35; 31	15; 14
	DTW	38; 35	38; 35	31; 27	4; 4	39; 37	14; 11
	SIG-B	46; 47	47; 48	9; 10	2; 1	45; 46	24; 25
	SIG-A	54; 55	49; 53	10; 8	1; 1	50; 53	33; 33
K-means	Euclidean	32; 32	29; 31	30; 26	5; 5	31; 31	11;11
	DTW	33; 37	32; 35	30; 25	4; 3	35; 36	14; 16
	SIG-B	33; 34	33; 34	30; 28	32; 33	33; 30	31; 34
	SIG-A	29; 31	29; 31	23; 25	25; 26	26; 28	27; 25

TABLE 6.9: ARI (%) after clustering **NATOPS** data.

		raw	$\texttt{crop}_{10\%}$	$\texttt{noise}_{\sigma}$	$\texttt{noise}_{3\sigma}$	$\texttt{warp}_{0.05}$	$\texttt{warp}_{0.5}$
Spect. clust.	Correlation	32; 33	30; 35	27; 30	28; 25	32; 37	7; 8
	Euclidean	45; 45	40; 46	35; 44	34; 27	46; 46	22; 17
	DTW	69; 56	64; 57	56; 48	33; 37	68; 58	22; 17
K-means	Euclidean	36; 37	37; 42	38; 35	30; 21	42; 35	21; 16
	DTW	53; 56	53; 56	71; 55	40; 32	54; 56	17; 13
	SIG-B	14; 15	14; 15	16; 21	19; 17	17; 22	13; 16
	SIG-A	24; 30	24; 30	25; 28	27; 29	20; 30	29; 28

DTW provide the best results followed closely by signature-based methods results. But for the `warp` transformation, the less affected methods are signatures.

In addition, we have seen that the choice of the norm in the signature space can modify drastically the performances and that neither $d_{\text{SIG-B}}$ nor $d_{\text{SIG-A}}$ works better for all datasets. Also, the clustering procedure impacts largely the results with the spectral clustering giving the best results, except in the case of the `noise` perturbation. Regarding computation times, we have seen that our non optimal signature-based algorithms are up to two orders of magnitude faster than DTW.

Conclusion of the thesis

Summary. In this thesis, we designed a methodology to compute the barycenter of a set of signatures (Chapter 3), leveraging algebraic properties and tools from differential geometry. Using this averaging methodology, we introduced a dimension reduction procedure for signatures by adapting previous work on Lie groups to the space of signatures (Chapter 4). We demonstrated that this method allows us to use significantly fewer features than the full signature while maintaining almost the same performance. We then focused on two major tasks in time series analysis: anomaly detection (Chapter 5) and clustering (Chapter 6). For anomaly detection, we employed a multiscale approach that integrates efficiently with the signature computation procedure. For clustering, we introduced multiple similarity measures designed with computational efficiency in mind and benchmarked them against classical clustering algorithms.

Implications of the research. The averaging and dimension reduction methodologies introduced in Chapters 3 and 4 pave the way for generalizing strategies originally designed for static data to dynamic data (time series). An example of this is the K-means procedure discussed in Chapter 6. Another straightforward application could be extending Random Forest classification to the signature space. This can be achieved by using decision boundaries $\mathcal{P}_1 := \{x \mid d(x, \mu_1) < d(x, \mu_2)\}$ and $\mathcal{P}_2 := \{x \mid d(x, \mu_1) > d(x, \mu_2)\}$, where μ_1 and μ_2 are centroids obtained from a 2-means clustering procedure in the signature space, utilizing the signature averaging method (Chapter 3) and an appropriately chosen distance metric d (see the discussion in Chapter 6). This extension is inspired by [Cap+24], where Random Forest is applied to data in general metric spaces. Furthermore, dimension reduction can facilitate the use of signature-based learning with larger signature levels on higher-dimensional time series (longer and with more components).

Limitations. The methodologies developed in this thesis come with certain limitations. The algorithm for computing the mean is not particularly fast and does not characterize the measure as the expected signature. Similarly, Principal Geodesic Analysis is slow due to the gradient optimization process. Additionally, both the mean and the principal geodesics are defined in the signature space, which is quite abstract and may be challenging to interpret. Another limitation is in anomaly detection: in industrial applications, precise localization of anomalies is often crucial. However, our method detects anomalies at a global level and does not provide precise localization.

Future research directions. We have established a definition for the average, but exploring the concepts of median (quantiles) and covariance in this context could be highly beneficial. A research direction that would be valuable to the signature theory is the generalization of learning strategies initially designed for linear spaces to Lie groups (or manifolds), exploiting the manifold and group structures as done with the PGA. Note that a lot of work in this direction has already been done, in particular by

developers of softwares `ManOpt` [Bou+14] and `geomstats` [Mio+20]. Regarding Chapters 5 and 6, further investigation into the multiscale signature anomaly detection method is necessary, particularly to address local anomalies (collective, contextual, point) and to use feature importance. For clustering with the signature, it would be interesting to employ more advanced methods such as agglomerative hierarchical clustering and neural networks. Additionally, designing efficient numerical methods for optimization within the Lie group of signatures would be very useful. Furthermore, as discussed in Section 2.3.5, trajectory reconstruction from a signature is challenging, and current algorithms require a large number of signature features. Advancing this area, which is in relation to optimization on Lie groups, could be very beneficial. An intriguing outcome would be to interpolate between two time series, as it is done in [CRT21, Section 4] with a Wasserstein-Fourier distance, using a reconstruction of the signature barycenter or to reconstruct principal geodesics obtained from the PGA. Finally, another interesting line of research is to obtain limit theorems of $k \mapsto \mathbf{S}_{[0,k]}(X)$ for stationary time series under dependence conditions, as explored in [Kif24].

Appendix A

Supplementary material of Chapter 4

A.1 Background on Lie groups

In this section, we group some useful facts on Lie groups.

Lemma A.1. *For any* $\mathbf{g}, \mathbf{h} \in G$ *and* $\mathbf{v} \in T_{\mathbf{1}}(G)$*, we have*

$$(dL_{\mathbf{g}})_{\mathbf{h}} \circ (dR_{\mathbf{h}})_{\mathbf{1}} = (dR_{\mathbf{h}})_{\mathbf{g}} \circ (dL_{\mathbf{g}})_{\mathbf{1}}. \tag{A.1}$$

Proof of Lemma A.1. Let $\mathbf{g}, \mathbf{h}, \mathbf{p} \in G$. Using the associativity rule of the group, we have

$$(L_{\mathbf{g}} \circ R_{\mathbf{h}})(\mathbf{p}) = \mathbf{g}(\mathbf{ph}) = (\mathbf{gp})\mathbf{h} = (R_{\mathbf{h}} \circ L_{\mathbf{g}})(\mathbf{p}) \tag{A.2}$$

thus $L_{\mathbf{g}}$ and $R_{\mathbf{h}}$ commute for any $\mathbf{g}, \mathbf{h}$. Differentiating Equation (A.2), with $\mathbf{p} = \mathbf{1}$ and $\mathbf{v} \in T_{\mathbf{1}}(G)$, we have

$$((dL_{\mathbf{g}})_{\mathbf{h}} \circ (dR_{\mathbf{h}})_{\mathbf{1}})(\mathbf{v}) = ((dR_{\mathbf{h}})_{\mathbf{g}} \circ (dL_{\mathbf{g}})_{\mathbf{1}})(\mathbf{v}) \tag{A.3}$$

thus $dL_{\mathbf{g}}$ and $dR_{\mathbf{h}}$ commute for any $\mathbf{g}, \mathbf{h}$. Note that this stays true for any $\mathbf{p} \in G$. □

Lemma A.2. *For any* $\mathbf{g}, \mathbf{h} \in G$*, we have*

$$(dL_{\mathbf{gh}})_{\mathbf{1}} = (dL_{\mathbf{g}})_{\mathbf{h}} \circ (dL_{\mathbf{h}})_{\mathbf{1}}. \tag{A.4}$$

Proof. This identity comes from the differentiation of $L_{\mathbf{gh}} = L_{\mathbf{g}} \circ L_{\mathbf{h}}$. □

Definition A.3. *We call adjoint representation of* G *the function* $\mathrm{Ad} : G \longrightarrow \mathrm{Aut}(\mathfrak{g}), \mathbf{g} \mapsto \mathrm{Ad}_{\mathbf{g}}$ *where*

$$\mathrm{Ad}_{\mathbf{g}} : T_{\mathbf{1}}(G) \longrightarrow T_{\mathbf{1}}(G) \tag{A.5}$$

$$\mathbf{v} \mapsto (dL_{\mathbf{g}})_{\mathbf{g}^{-1}} \circ (dR_{\mathbf{g}^{-1}})_{\mathbf{1}}(\mathbf{v}) \tag{A.6}$$

The derivative of Ad *is called the adjoint representation of* $\mathfrak{g}$, $\mathrm{ad} : \mathfrak{g} \longrightarrow \mathrm{End}(\mathfrak{g}), \mathbf{v} \mapsto \mathrm{ad}_{\mathbf{v}} := d(\mathrm{Ad})_{\mathbf{1}}(\mathbf{v})$. *One can show that for any* $\mathbf{v}, \mathbf{w} \in \mathfrak{g}$*, we have* $\mathrm{ad}_{\mathbf{v}}(\mathbf{w}) = [\mathbf{v}, \mathbf{w}] := \mathbf{vw} - \mathbf{wv}$*, where* $[.,.]$ *is called the Lie bracket.*

Lemma A.4. *Let* $\mathbf{g} = e^{\mathbf{u}}$ *be an element of the signature space* $G_{\leq L}$ *and* $\mathbf{v}$ *an element of the signature Lie algebra* $\mathfrak{g}_{\leq L}$*. Then*

$$\mathrm{Ad}(\mathbf{g})(\mathbf{v}) = \mathbf{gvg}^{-1} = e^{ad_{\mathbf{u}}}\mathbf{v}. \tag{A.7}$$

Proof. See [Reu93, Theorem 3.2]. □

A.2 Proofs

Proof of Proposition 4.6. The first condition of Definition 4.5 is satisfied since $\mathbf{h} \mapsto \log_{\mathbf{g}} \mathbf{h}$ is locally a bijection from G to $T_{\mathbf{g}}(G)$ and the norm is nonnegative.

To verify the second condition, we consider the point $\mathbf{g} = e^{\mathbf{u}}$ and the mapping $f : T_{\mathbf{1}}(G) \to G; \mathbf{v} \mapsto e^{\mathbf{u}/2} e^{\mathbf{v}} e^{\mathbf{u}/2}$, which satisfies $f(0) = \mathbf{g}$. Moreover, by group and logarithm properties, we have $\mathbf{v} = \log(e^{-\mathbf{u}/2} f(\mathbf{v}) e^{-\mathbf{u}/2})$. Thus $f(\mathbf{v})$ is one-to-one, and can be taken as a chart of the manifold in a neighborhood of $\mathbf{g}$. Finally, following the same steps as in the proof of Lemma 4.8,

$$D(e^{\mathbf{u}} : e^{\mathbf{u}/2} e^{\mathbf{v}} e^{\mathbf{u}/2}) = \|\mathbf{v}\|_{\mathbf{1}}^2.$$

That proves, that in local coordinates given by the exponential map, the Hessian of the divergence with respect to the second variable is exactly the quadratic form associated with $\langle \cdot, \cdot \rangle_{\mathbf{1}}^2$, and thus is positive definite. □

Proof of Proposition 4.7. We split the proof in several steps. First, we show how to rephrase Equation (4.9). For this, let $\mathbf{v} = \log(\boldsymbol{\mu})$, we can see that we can apply $\mathrm{Ad}(e^{\frac{1}{2}\mathbf{v}})$ operator to (4.9) and show that $\boldsymbol{\mu}$ is a barycenter if and only if

$$\sum_{i=1}^{N} F_{\mathbf{x}_i}(\mathbf{v}) = 0,$$

where

$$F_{\mathbf{x}}(\mathbf{v}) = \log\left(e^{-\frac{1}{2}\mathbf{v}} \mathbf{x}_i e^{-\frac{1}{2}\mathbf{v}}\right).$$

Now let us rewrite the first-order conditions for the cost function in Equation (4.10). First, we look how the scaling affects the cost function. Denote $\mathbf{v} = \log(\boldsymbol{\mu})$. Since the group operation, logarithm, and inverses commute with dilation, we have that, using Lemma 4.8, the divergence between dilated vectors becomes

$$D(\delta_\lambda(\mathbf{m}) : \delta_\lambda(\mathbf{x})) = \|\log(e^{-\frac{1}{2}\delta_\lambda(\mathbf{v})} \delta_\lambda(\mathbf{x}) e^{-\frac{1}{2}\delta_\lambda(\mathbf{v})})\|_{\mathbf{1}}^2 = \|\delta_\lambda\left(F_{\mathbf{x}}(\mathbf{v})\right)\|_{\mathbf{1}}^2,$$

Next, we split the vector inside the norm of the divergence by the orders:

$$F_{\mathbf{x}}(\mathbf{v}) = (0, F_{\mathbf{x},1}(\mathbf{v}), \ldots, F_{\mathbf{x},L}(\mathbf{v})),$$

and let us write the gradient of the divergence between scaled parts $\nabla_{\mathbf{v}} D(\delta_\lambda(\mathbf{m}) : \delta_\lambda(\mathbf{x})) =$

$$2 \begin{bmatrix} 0 & 0 & \cdots & 0 \\ (\nabla_{\mathbf{v}_1} F_{\mathbf{x},1}(\mathbf{v}))^\top & (\nabla_{\mathbf{v}_1} F_{\mathbf{x},2}(\mathbf{v}))^\top & \cdots & (\nabla_{\mathbf{v}_1} F_{\mathbf{x},L}(\mathbf{v}))^\top \\ (\nabla_{\mathbf{v}_2} F_{\mathbf{x},1}(\mathbf{v}))^\top & (\nabla_{\mathbf{v}_2} F_{\mathbf{x},2}(\mathbf{v}))^\top & \cdots & (\nabla_{\mathbf{v}_2} F_{\mathbf{x},L}(\mathbf{v}))^\top \\ \vdots & \vdots & & \vdots \\ (\nabla_{\mathbf{v}_L} F_{\mathbf{x},1}(\mathbf{v}))^\top & (\nabla_{\mathbf{v}_L} F_{\mathbf{x},2}(\mathbf{v}))^\top & \cdots & (\nabla_{\mathbf{v}_L} F_{\mathbf{x},L}(\mathbf{v}))^\top \end{bmatrix} \begin{bmatrix} \lambda^2 & & & \\ & \lambda^4 & & \\ & & \ddots & \\ & & & \lambda^{2L} \end{bmatrix} \begin{bmatrix} F_{\mathbf{x},1}(\mathbf{v}) \\ F_{\mathbf{x},2}(\mathbf{v}) \\ \vdots \\ F_{\mathbf{x},L}(\mathbf{v}) \end{bmatrix}.$$

Note that, from the symmetric BCH formula,

$$F_{\mathbf{x},j}(\mathbf{v}) = -\frac{1}{2}\mathbf{v}_j + \underbrace{\cdots}_{\text{depending on } \mathbf{v}_1,\ldots,\mathbf{v}_{j-1}},$$

hence the Jacobian $(\nabla_{\mathbf{v}} F_{\mathbf{x}})^\top$ is upper triangular with scaled identity on the diagonal. By applying an inverse scaling, we have that (for example, by [BU21, Lemma 4.7])

$$\widetilde{F}_{\mathbf{x},\lambda}(\mathbf{v}) = \delta_{\lambda^{-2}}(\nabla_{\mathbf{v}} D(\delta_\lambda(\mathbf{m}) : \delta_\lambda(\mathbf{x}))) = -(\mathbf{I} + O(\lambda^2))F_{\mathbf{x}}(\mathbf{v}).$$

Therefore, the rescaled necessary condition $\delta_{\lambda^{-2}} \nabla_{\mathbf{v}} f(e^{\mathbf{v}}) = 0$ for the local minimum in (4.9) can be written as

$$\sum_{i=1}^{N} \widetilde{F}_{\mathbf{x}_i,\lambda}(\mathbf{v}) = 0.$$

Finally, denote by $\widetilde{F}(\lambda, \mathbf{v}) = \sum_{i=1}^{N} \widetilde{F}_{\mathbf{x}_i,\lambda}$. Note that we have the following:

- $\widetilde{F}(\lambda, \mathbf{v})$ is polynomial both in λ and $\mathbf{v}$;
- $\widetilde{F}(\lambda, \mathbf{v}) = \widetilde{F}(0, \mathbf{v}) + O(\lambda^2)$, in particular $\widetilde{F}(0, \mathbf{v}) = \sum_{i=1}^{N} F_{\mathbf{x}_i}(\mathbf{v})$;
- The equation $\widetilde{F}(0, \mathbf{v}) = 0$ has unique solution (which is a barycenter).
- The Jacobian $\nabla \widetilde{F}(0, \mathbf{v})$ is a triangular matrix with the constant value on the diagonal $(-N/2)$, and thus is nonsingular.

By the implicit function theorem, for small λ, $\widetilde{F}(\lambda, \mathbf{v}) = 0$ has a unique solution $\mathbf{v}(\lambda)$ in some neighborhood of the barycenter $\boldsymbol{\mu}$, and $\mathbf{v}(\lambda)$ is a continuous function of λ. We are left to prove that for sufficiently small fixed λ, this solution is unique globally (for $\mathbf{v} \in T_{\mathbf{1}}(G)$).

For this we note the following.

- First, since $\det(\nabla \widetilde{F}(0, \mathbf{v})) = const$, for all $\mathbf{v} \in T_{\mathbf{1}}(G)$ and $\widetilde{F}(\lambda, \mathbf{v})$ is polynomial, then there exist λ_0 such that for all $0 \le \lambda \le \lambda_0$ we have $\det(\nabla \widetilde{F}(\lambda, \mathbf{v})) \ge C_0 > 0$. This guarantees that the Jacobian is globally nonsingular for small λ.
- Second, we note that for any fixed $\mathbf{v}_0 \in T_{\mathbf{1}}(G)$, the function $\|\widetilde{F}(\lambda, \mathbf{v}_0 t)\|_{\mathbf{1}}^2 \to \infty$ as $t \to \infty$, due to the fact that it is a squared norm of some polynomial map, which is not constant in t (similarly to the argument in the proof of Lemma 4.11).

This implies by [Pal59, Corollary 4.3], that each $\widetilde{F}(\lambda, \mathbf{v}))$, for $\lambda \le \lambda_0$ is a global homeomorphism, thus the solution to $\widetilde{F}(\lambda, \mathbf{v}) = 0$ is unique for $0 \le \lambda \le \lambda_0$ (and coincides with $\mathbf{v}(\lambda)$ given by the implicit function theorem). □

A.3 Supplementary material of Section 4.5

We give details on the SES method [Lem+21] coupled with the PGA/tPGA in Figure A.1 and more thoroughly in Algorithm 5.

Also, we present the dimension of the Lie group $G_{\le L}$ and the corresponding Lie algebra $\mathfrak{g}_{\le L}$ for a fixed dimension $d = 4$ in Table A.1.

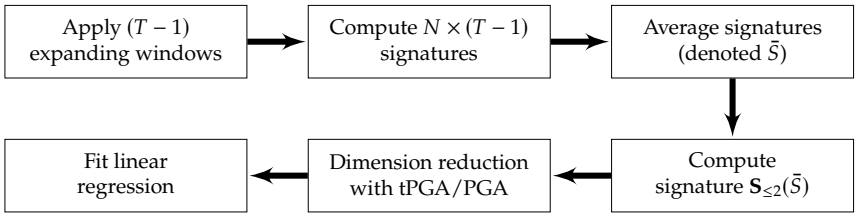


FIGURE A.1: SES method [Lem+21] with tPGA/PGA (SES-tPGA and SES-PGA).

Algorithm 5: SES METHOD [LEM+21] WITH PGA (SES-PGA)

Input: $\sigma = [\sigma^{i,j}[t]]_{i,j}$ a set of MN time series, with time indices $t = 1, \dots, T$
$\alpha = [\alpha^1, \dots, \alpha^M]$ a set of M real values to predict

Initialize an array SES of size $(M \times B_{2,d'})$ where $d' := B_{\ell,d}$
for $i = 1, \dots, M$ **do**
  Initialize an array $\bar{S}$ of size $(T-1, d')$
  for $k = 1, \dots, T-1$ **do**
    $\bar{S}[k,:] \leftarrow \frac{1}{N} \sum_{j=1}^{N} \mathbf{S}_{\leq \ell}\left(\sigma^{i,j}[1:k+1]\right)$
  $\mathrm{SES}[i,:] \leftarrow \mathbf{S}_{\leq 2}(\bar{S})$
$\widetilde{X} \leftarrow \mathrm{PGA}\Big(\mathrm{SES}[1], \dots, \mathrm{SES}[M]\Big)$
Split features $\widetilde{X}$ and target α into train and test sets
$\beta^* \leftarrow \mathrm{LinearRegression}(\alpha_{\mathrm{train}}, \widetilde{X}_{\mathrm{train}})$
$\hat{\alpha}_{\mathrm{test}} \leftarrow \beta^* \widetilde{X}_{\mathrm{test}}$
return $\hat{\alpha}_{\mathrm{test}}, \alpha_{\mathrm{test}}$

Truncation level L	2	3	4	5
$\dim(G_{\leq L}(\mathbb{R}^4))$	21	85	341	1365
$\dim(\mathfrak{g}_{\leq L}(\mathbb{R}^4))$	10	30	90	294

TABLE A.1: Dimensions of the signature space $G_{\leq L}(\mathbb{R}^4)$ and its tangent space $\mathfrak{g}_{\leq L}(\mathbb{R}^4)$ for various truncation levels L and fixed dimension $d = 4$.

Appendix B

Supplementary material of Chapter 6

B.1 Dynamic Time Warping for measuring similarities in time series

Dynamic Time Warping (DTW) [BC94] is a similarity measure between time series introduced to deal with asynchronicity. Consider $X = (X(1), \dots, X(n))$, $n \geq 1$ and $Y = (Y(1), \dots, Y(m))$, $m \geq 1$, two d-dimensional time series, respectively of lengths n and m. Thus, $X(i)$ and $Y(j)$ are both vectors of dimension d. The DTW cumulated similarity score between the first i columns of X and the first j columns of Y, denoted as $\mathrm{cs}(i, j)$, is defined as

$$\mathrm{cs}(i, j) = \min\{\mathrm{cs}(i-1, j-1), \mathrm{cs}(i, j-1), \mathrm{cs}(i-1, j)\} + \|X(i) - Y(j)\|^2. \tag{B.1}$$

Computing this similarity score gives us a trajectory alinement between X and Y. Note that several variants of DTW exists. For instance, the Sakoe-Chiba band allows for a faster computation of the cs matrix.

B.2 K-means clustering method

The K-means method is a relatively simple strategy used to discriminate observations into K clusters. The method is the following. Let $x_1, \dots, x_N$ be N observations in $\mathbb{R}^d$. First, initialize K values $\mu_1^0, \dots, \mu_k^0$ in $\mathbb{R}^d$, called centroids. Then, the K-means method is the iteration of the following two steps, until convergence: at iteration k, do

i. Assignment step: assign each observation x_i to the cluster with the nearest mean μ_j^k.

ii. Update step: μ_j^{k+1} is the average of observations in cluster j.

The convergence condition can be a fixed number of iterations (e.g. 100 iterations), or stop when clusters do not change anymore between two iterations. Note that there have new versions of this algorithm, e.g., the k-means++ method or the X-means. Here, we only deal with the vanilla method as it is still widely used [ASW15, Table 4].

B.3 Spectral clustering method

Given data, we compute a similarity matrix S, where $S_{i,j}$ is the similarity between x_i and x_j. The spectral clustering method consists in the computation of the eigenvectors

of the normalized graph Laplacian associated to the similarity matrix. We present the spectral clustering strategy in Algorithm 6.

Algorithm 6: SPECTRAL CLUSTERING

Input: S the $N \times N$ similarity matrix; K number of clusters

1 $D \leftarrow$ the diagonal matrix such that $D_{i,i} := \sum_{j=1}^{N} S_{i,j}$.

2 $L \leftarrow D - S$

3 $L \leftarrow D^{-1/2} L D^{-1/2}$

4 $U \leftarrow$ the $N \times K$ matrix such that its k-th column is the k-th eigenvector of L.

5 $\widetilde{U} \leftarrow$ the matrix such that $\widetilde{U}_{i,j} = \frac{U_{i,j}}{\|U_{i,j}\|}$.

6 $y_i \leftarrow i$-th row of $\widetilde{U}$, $i = 1, \ldots, N$.

7 Apply the K-means algorithm to data $\{y_1, \ldots, y_N\}$ and obtain K clusters $C_1, \ldots, C_K$.

Output: Clusters $C_1, \ldots, C_K$

This algorithm can be efficiently implemented using standard linear algebra libraries, such as BLAS. Note that many variants exists, for instance, instead of using the complete matrix S in the algorithm, we can construct a sparse version of S where for each point x_i, only the similarity values of the n nearest neighbors are kept and the other values are set to zero. For further details on spectral clustering, its tuning strategy, interpretation with graph theory and variants, see [Lux07].

Appendix C

Differentiable geometry toolbox

The goal of this section is to present a few elements of differential geometry. In particular, those notions are used in Chapters 3 and 4.

Differentiable geometry is the study of smooth manifolds, see Appendix C.2 below. An extensively studied example are Riemannian manifolds, which are smooth manifolds with a notion of length, presented in Appendix C.3. A link with the signature space is made in Appendix C.4. To dive deeper into the theory of differentiable geometry and Riemannian manifolds, we refer to the following introducing textbooks: [Lee18], [Car92] and [Tu17]. An introduction to Lie groups can be found in [Hal03].

C.1 Standard notations

There are a lot of abuse of notation in differential geometry. To be consistent with textbooks, we will use the following notations. Let $\mathcal{M}$ be a manifold, $p \in \mathcal{M}$, $f : \mathcal{U} \to \mathbb{R}^n$ a smooth (C^∞) mapping, with $\mathcal{U}$ open subset of $\mathcal{M}$, and X a smooth (C^∞) vector field on $\mathcal{U}$. Denote as $a^i(p)$ the coordinates of X_p in basis $\left(\frac{\partial}{\partial x^1}, \dots, \frac{\partial}{\partial x^n}\right)$ of $T_p\mathcal{M}$.

- X_p — Mapping from $C^\infty(\mathcal{U}, \mathbb{R}^n)$ to $\mathbb{R}$ such that $X_p = X(p) := \sum_{i=1}^n a^i(p) \left.\frac{\partial}{\partial x^i}\right|_p$
- Xf — Differential of f in the direction of X,

$$Xf := \sum_{i=1}^n a^i \frac{\partial f}{\partial x^i} \tag{C.1}$$

  and for any $p \in \mathcal{M}$, we denote the differential of f in the direction of X at point p as

$$(Xf)(p) := \sum_{i=1}^n a^i(p) \left.\frac{\partial f}{\partial x^i}\right|_p . \tag{C.2}$$

- fX — Pointwise product of f and X, $(fX)(p) := f(p)X(p)$.

A particularly confusing notation is the Einstein convention: if an index appears twice in a formula, then there is implicitly a sum over it, e.g., YZ^kE_k denotes $\sum_k YZ^kE_k$ (where we intentionally don't give meaning to Y, Z and E here). We will not use this convention but note that it is standard in textbooks.

C.2 Manifolds, tangent space and connections

First, we introduce the notion of topological manifold, with the following two definitions.

Definition C.1 (Locally Euclidean space)**.** *A topological set $\mathcal{M}$ is locally Euclidean of dimension n if for all $p \in \mathcal{M}$, there exists a neighborhood $\mathcal{U}$ such that there is a homeomorphism ϕ defined from $\mathcal{U}$ to $\mathbb{R}^n$.*

Definition C.2 (Topological manifold)**.** *A topological set $\mathcal{M}$ is a topological manifold of dimension n if it is a Hausdorff, second countable, locally Euclidean space of dimension n.*

Loosely speaking, a manifold is a space such that for each point there exists a bijection between a neighborhood of this point and $\mathbb{R}^n$ for some integer n.

Definition C.3 (Chart)**.** *Let $\mathcal{M}$ be a set. Let $\phi : \mathcal{U} \subset \mathcal{M} \to \mathbb{R}^d$ be a bijection. $(\mathcal{U}, \phi)$ is called a d-dimensional chart of $\mathcal{M}$.*

Definition C.4 (Atlas)**.** *Let $(\mathcal{U}_\alpha, \phi_\alpha)$ be a collection of d-dimensional charts of $\mathcal{M}$ such that:*

- $\bigcup_\alpha \mathcal{U}_\alpha = \mathcal{M}$
- *For any pair α, β such that $\mathcal{U}_\alpha \cap \mathcal{U}_\beta \neq \emptyset$, both sets $\phi_\alpha(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$ and $\phi_\beta(\mathcal{U}_\alpha \cap \mathcal{U}_\beta)$ are open sets in $\mathbb{R}^d$ and mapping $\phi_\beta \circ \phi_\alpha^{-1}$ is C^∞.*

$(\mathcal{U}_\alpha, \phi_\alpha)$ is called a smooth (or C^∞) atlas of $\mathcal{M}$ into $\mathbb{R}^d$.

The second condition is to ensure that the charts overlap smoothly. Then, we define the notion of maximal smooth atlas as follows.

Definition C.5 (Maximal atlas)**.** *Given a smooth atlas on a topological space, a chart is said to be smoothly compatible with the atlas if the inclusion of the chart into the collection of charts of the atlas results in a smooth atlas.*

A smooth atlas determines a maximal smooth atlas, consisting of all charts which are smoothly compatible with the given atlas.

Definition C.6 (Smooth manifold)**.** *A smooth manifold is a Hausdorff and second countable topological space $\mathcal{M}$, together with a maximal differentiable atlas on $\mathcal{M}$.*

Remark C.7. *There are specific well-studied manifolds, that we will define in the following.*

- *Manifolds defined by the choice of the atlas, e.g., Riemannian manifolds.*
- *Manifolds equipped with an additional structure, e.g., smooth manifolds or Lie groups (which are smooth manifolds with a group structure).*

From now on, we only consider smooth manifolds.

A crucial point in differential geometry is that tangent spaces are linear maps, that we define now. For further details on differentiation on manifolds (and thus tangent spaces), the approach made in [Tu11, Section 8] is helpful.

Definition C.8 (Tangent space)**.** *Let $\mathcal{M}$ be a smooth manifold and $p \in \mathcal{M}$. A tangent vector at p is a linear map $v : C^\infty(\mathcal{M}) \to \mathbb{R}$ such that for all f, g smooth (C^∞) mappings on $\mathcal{M}$, it satisfies the product (or Leibniz) rule*

$$v(fg) = f(p)vg + g(p)vf \ . \tag{C.3}$$

The set of all tangent vectors at p is called the tangent space at p, denoted as $T_p\mathcal{M}$.

We can now define set of tangent vectors.

Definition C.9 (Vector field)**.** *Let $\mathcal{M}$ be a smooth manifold. We call vector field any mapping*

$$\begin{aligned} X : \mathcal{M} &\to T\mathcal{M} \\ p &\mapsto X_p \end{aligned}$$

where we have denoted as $T\mathcal{M}$ the following disjoint union:

$$T\mathcal{M} = \bigsqcup_{p\in\mathcal{M}} T_p\mathcal{M} \,. \tag{C.4}$$

In the following, we denote as $\mathfrak{X}(M)$ the set of all the vector fields defined on $\mathcal{M}$.

The following definition is central in differential geometry. It allows us to compare tangent vectors in different tangent spaces, which is not straightforward in manifolds as in linear spaces. From this notion, called connection, we can define derivatives on manifolds.

Definition C.10 (Connection)**.** *Let $\mathfrak{X}(\mathcal{M})$ be the set of vector fields on $\mathcal{M}$. An affine (or linear) connection is any map*

$$\begin{aligned} \nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) &\to \mathfrak{X}(\mathcal{M}) \\ (X, Y) &\mapsto \nabla(X, Y) =: \nabla_X Y \end{aligned}$$

such that the three following properties are verified:

- *Linearity over $C^\infty(\mathcal{M})$ with respect to X: for any $f \in C^\infty(\mathcal{M})$,*

$$\nabla_{fX} Y = f\nabla_X Y \,. \tag{C.5}$$

- *$\mathbb{R}$-linearity with respect to Y : for any real a, b,*

$$\nabla_X(aY + bZ) = a\nabla_X Y + b\nabla_X Z \,. \tag{C.6}$$

- *Product (or Leibniz) rule: for any $f \in C^\infty(\mathcal{M})$,*

$$\nabla_X(fY) = (Xf)Y + f\nabla_X Y \,. \tag{C.7}$$

$\nabla_X Y$ is called the covariant derivative of Y in the direction of X and ∇ is called connection or covariant derivative.

Remark C.11. *Note that in Definition C.10, we have introduced the notion of affine connection and not the general notion of connection, for clarity purposes. The difference between the definition of connection and affine connection is the set to which the Y belongs and also the output space of ∇. Denote $\pi : E \to \mathcal{M}$ a vector bundle over $\mathcal{M}$ (not defined) and denote $\mathcal{E}(\mathcal{M})$ the set of smooth sections of E (not defined). A connection (not necessarily affine) is a function*

$$\nabla : \mathfrak{X}(\mathcal{M}) \times \mathcal{E}(\mathcal{M}) \to \mathcal{E}(\mathcal{M}) \,. \tag{C.8}$$

Now, in order to introduce an important theorem and the notion of parallel vector, which allows us to translate vectors from one tangent space to another, we need the following definition of vector field along a curve.

Definition C.12 (Vector field along a curve)**.** *Let $c : [a,b] \to \mathcal{M}$. We call vector field along curve c any mapping*

$$V : [a,b] \to \bigsqcup_{a \le t \le b} T_{c(t)}\mathcal{M} \,. \tag{C.9}$$

Theorem C.13 (Covariant derivative along a curve)**.** *Let $\mathcal{M}$ be a smooth manifold and ∇ a connection in $T\mathcal{M}$. For each smooth (C^∞) mapping $c : [a,b] \to \mathcal{M}$, ∇ gives a unique mapping*

$$\frac{D}{dt} : \Gamma(T\mathcal{M}|_{c(t)}) \to \Gamma(T\mathcal{M}|_{c(t)}) \,, \tag{C.10}$$

where we denote as $\Gamma(T\mathcal{M}|_{c(t)})$ the space of vector fields along c, such that for any vector field V along curve c we have the following:

- *$\mathbb{R}$-linearity: for any $\lambda \in \mathbb{R}$,*
$$\frac{D(\lambda V)}{dt} = \lambda \frac{DV}{dt} \,. \tag{C.11}$$
- *Product (or Leibniz) rule:*
$$\frac{D(fV)}{dt} = \frac{Df}{dt} V + f \frac{DV}{dt} \,. \tag{C.12}$$
- *Compatibility with ∇:*
$$\frac{DV}{dt}(t) = \nabla_{c'(t)} \widetilde{V} \tag{C.13}$$
where $\widetilde{V}$ is the vector field such that $V(t) = \widetilde{V}(c(t))$.

The mapping $\frac{DV}{dt}$ is called the covariant derivative of V along c.

Proof. See [Lee18, Theorem 4.24]. □

Definition C.14 (Parallel vector field)**.** *Let $\mathcal{M}$ be a smooth manifold and ∇ connection in $T\mathcal{M}$. A vector field V along a curve c is said to be parallel along c with respect to ∇ if*

$$\frac{DV}{dt} \equiv 0 \,. \tag{C.14}$$

This is illustrated in Figure C.1.

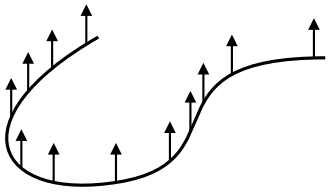


FIGURE C.1: A parallel vector field along a curve.

Definition C.15 (Parallel transport)**.** *Let $\gamma : I \subset \mathbb{R} \to \mathcal{M}$ be a smooth mapping, $t_0 \in I$, and $v \in T_{\gamma(t_0)}\mathcal{M}$. We call parallel transport of v along γ the unique parallel vector field V along γ such that $V(t_0) = v$, which existence and uniqueness is proved in [Lee18, Theorem 4.32].*

We call parallel transport the mapping

$$\Pi^{\gamma}_{t_0 \to t_1} : T_{\gamma(t_0)}\mathcal{M} \to T_{\gamma(t_1)}\mathcal{M} \tag{C.15}$$

such that $\Pi^{\gamma}_{t_0 \to t_1}(v) = V(t_1)$ *for each* $v \in T_{\gamma(t_0)}\mathcal{M}$*, where* V *is the parallel transport of* v *along* γ*.*

The notions of parallel transport and connection are closely related. The following proposition shows that the parallel transport determines the connection.

Proposition C.16. *Let* $\mathcal{M}$ *be a manifold,* ∇ *a connection on* $T\mathcal{M}$*. Let* X, Y *be smooth vector fields on* $\mathcal{M}$*. Then, for any* $p \in \mathcal{M}$*,*

$$\nabla_X Y|_p = \left.\frac{d}{dt}\right|_{t=0} \Pi^{\gamma}_{t \to 0} Y_{\gamma(t)} \tag{C.16}$$

where $\gamma : I \subset \mathbb{R} \to \mathcal{M}$ *is any smooth mapping such that* $\gamma(0) = p$ *and* $\dot{\gamma}(0) = X_p$*.*

Proof. See [Lee18, Corollary 4.35]. □

Now, we want to generalize the notion of straight line to curved spaces.

Definition C.17 (Geodesic)**.** *Let* $\mathcal{M}$ *be a manifold equipped with an affine connection* ∇*. A smooth mapping* $\gamma : I \subset \mathbb{R} \to \mathcal{M}$ *is called a geodesic if for all* $t \in I$*,*

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0 \,. \tag{C.17}$$

In other words, a geodesic is a curve which covariant derivative of its acceleration $\ddot{\gamma}$ is null. It is a generalization of straight lines to curved spaces.

C.3 Riemannian manifolds

We introduce the following definition of Riemannian metric, which is an intrinsic way of measuring lengths on a manifold. A Riemannian metric is an inner product on tangent space, whereas a metric is an abstract notion of distance. Note, that any Riemannian metric induces a metric, called the Riemannian distance.

Definition C.18 (Riemannian metric)**.** *Let* $(\langle ., . \rangle_{p \in \mathcal{M}})$ *be a collection of inner products (i.e., symmetric, bilinear and positive definite mappings) that varies smoothly in* p*, i.e., for any pair of smooth vector fields* X, Y *around* p*,*

$$p \mapsto \langle X_p, Y_p \rangle \tag{C.18}$$

is a smooth mapping. We call Riemannian metric the mapping $p \mapsto g_p$ *with*

$$\begin{aligned} g_p : T_p\mathcal{M} \times T_p\mathcal{M} &\to \mathbb{R} \\ (u, v) &\mapsto \langle u, v \rangle_p \end{aligned}$$

where we denote as $T_p\mathcal{M}$ *the tangent space at* p *of* $\mathcal{M}$*.*

Theorem C.19. *Let* $\mathcal{M}$ *be a smooth manifold. Then* $\mathcal{M}$ *admits a Riemannian metric.*

Proof. See [Car92, Proposition 2.10]. □

Definition C.20 (Riemannian manifold)**.** *Let* $\mathcal{M}$ *be a smooth manifold equipped with a Riemannian metric* g*.* $(\mathcal{M}, g)$ *is called a Riemannian manifold.*

In order to introduce the fundamental theorem of Riemannian geometry, we need the following two definitions.

Definition C.21 (Symmetric connection). *A connection ∇ is said to be symmetric if for all vector fields $X, Y \in \mathfrak{X}(\mathcal{M})$, we have*

$$\nabla_X Y - \nabla_Y X = [X, Y] \tag{C.19}$$

where $[X, Y]$ is the smooth vector field such that $[X, Y](f) = X(Y(f)) - Y(X(f))$ for all $f \in C^\infty(\mathcal{M})$.

Definition C.22 (Connection compatible with the Riemannian metric). *Let $(\mathcal{M}, \langle ., . \rangle)$ be a Riemannian manifold. A connection is said to be compatible with the Riemannian metric if for all vector fields $X, Y, Z \in \mathfrak{X}(\mathcal{M})$, we have*

$$Z \langle X, Y \rangle = \langle \nabla_Z X, Y \rangle + \langle X, \nabla_Z Y \rangle \ . \tag{C.20}$$

In other words, the angle between two vectors is constant when parallel transport is applied to the two vectors.

Theorem C.23 (Fundamental theorem of Riemannian geometry). *Let $(\mathcal{M}, g)$ be a Riemannian manifold. Then, there exists a unique symmetric connection ∇ on $T\mathcal{M}$ that is compatible with the metric g. It is called the Levi-Civita connection.*

Proof. See [Lee18, Theorem 5.10]. □

C.4 Signature space

The signature space (see Section 2.2.3) is a Lie group and also a sub-Riemannian manifold.

Definition C.24 (Lie group). *A Lie group is a smooth manifold equipped with a group structure such that the group operations of multiplication and inversion are smooth (C^∞) mappings.*

Example C.25. *Examples of Lie groups are:*

- $\mathrm{GL}(n)$ *the space of $n \times n$ real matrices with non zero determinant.*
- $O(n) = \{A \in \mathrm{GL}(n) \text{ s.t. } A^T A = I\}$ *the orthogonal group.*

Remark C.26. *Note that the space of signatures, denoted as $G_{\leq L}$ in Section 2.2.3, is a sub-Riemannian manifold (see [ABB19, Definition 3.2]), which is a more general structure than Riemannian manifolds and which requires more background, such as the notions of fiber, horizontal and bracket-generating vector field. For details on sub-Riemannian geometry, see [ABB19].*

Bibliography

[ABB19] Andrei Agrachev, Davide Barilari, and Ugo Boscain. *A comprehensive introduction to sub-Riemannian geometry*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2019. ISBN: 978-1-108-67732-5. DOI: 10.1017/9781108677325.

[AC10] Shun-ichi Amari and Andrzej Cichocki. "Information geometry of divergence functions". In: *Bulletin of the Polish Academy of Sciences Technical Sciences* 58.1 (2010), pp. 183–195. DOI: 10.2478/v10175-010-0019-1.

[Arr+24] Paola Arrubarrena, Maud Lemercier, Bojan Nikolic, Terry Lyons, and Thomas Cass. "Novelty detection on radio astronomy data using signatures". Preprint. 2024.

[ASW15] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering—a decade review". In: *Information Systems* 53 (2015), pp. 16–38. DOI: 10.1016/j.is.2015.04.007.

[Bag+18] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. "The UEA multivariate time series classification archive". Preprint. 2018.

[BC94] Donald J. Berndt and James Clifford. "Using dynamic time warping to find patterns in time series". In: *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (ICKDDM)*. Seattle, Washington, USA, 1994, pp. 359–370.

[BD16] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. 3rd. Springer Texts in Statistics. Springer Cham, 2016. ISBN: 978-3-319-29852-8; 978-3-319-29854-2. DOI: 10.1007/978-3-319-29854-2.

[BG20] Jonas R. Brehmer and Tilmann Gneiting. "Properization: constructing proper scoring rules via Bayes acts". In: *Annals of the Institute of Statistical Mathematics* 72 (2020), pp. 659–673. DOI: 10.1007/s10463-019-00705-7.

[BK81] Peter Buser and Hermann Karcher. "Gromov's almost flat manifolds". In: *Astérisque* 81 (1981).

[BO21] Patric Bonnier and Harald Oberhauser. "Proper scoring rules, gradients, divergences, and entropies for paths and time series". Preprint. 2021.

[Boe+21] Horatio Boedihardjo, Joscha Diehl, Marc Mezzarobba, and Hao Ni. "The expected signature of Brownian motion stopped on the boundary of a circle has finite radius of convergence". In: *Bulletin of the London Mathematical Society* 53.1 (2021), pp. 285–299. DOI: 10.1112/blms.12420.

[Bon+19] Patric Bonnier, Patrick Kidger, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. "Deep signature transforms". In: *Proceedings of 33rd International Conference on Neural Information Processing Systems (NeurIPS)*. Vol. 32. Vancouver, British-Columbia, Canada, 2019.

[Bou+14] Nicolas Boumal, Bamdev Mishra, Pierre-Antoine Absil, and Rodolphe Sepulchre. "Manopt, a Matlab toolbox for optimization on manifolds". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1455–1459.

[Bra+18] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018. URL: http://github.com/google/jax.

[Bre+00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Sander Jörg. "LOF: identifying density-based local outliers". In: *ACM Special Interest Group on Management of Data (SIGMOD) Record* 29.2 (2000), pp. 93–104. DOI: 10.1145/335191.335388.

[BU21] Simon Barthelmé and Konstantin Usevich. "Spectral properties of kernel matrices in the flat limit". In: *SIAM Journal on Matrix Analysis and Applications* 42.1 (2021), pp. 17–57. DOI: 10.1137/19M129677X.

[Cap+24] Louis Capitaine, Jérémie Bigot, Rodolphe Thiébaut, and Robin Genuer. "Fréchet random forests for metric space valued regression with non euclidean predictors". Preprint. 2024.

[Car26] Elie Cartan. "On the geometry of the group-manifold of simple and semi-groups". In: *Proceedings Akademie van Wetenschappen te Amsterdam* 29 (1926), pp. 803–815.

[Car92] Manfredo P. do Carmo. *Riemannian geometry*. 2nd. Mathematics: Theory & Applications. Birkhäuser Boston, Massachusetts, USA, 1992. ISBN: 978-0-8176-3490-2.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: a survey". In: *ACM Computing Surveys* 41.3 (2009), pp. 1–58. DOI: 10.1145/1541880.1541882.

[Che57] Kuo Tsai Chen. "Integration of paths, geometric invariants and a generalized Baker–Hausdorff formula". In: *Annals of Mathematics* 65.1 (1957), pp. 163–178. DOI: 10.2307/1969671.

[CK16] Ilya Chevyrev and Andrey Kormilitzin. "A primer on the signature method in machine learning". Preprint. 2016.

[CL16] Ilya Chevyrev and Terry Lyons. "Characteristic functions of measures on geometric rough paths". In: *Annals of Probability* 44.6 (2016), pp. 4049–4082. DOI: 10.1214/15-AOP1068.

[CL19] Jiawei Chang and Terry Lyons. "Insertion algorithm for inverting the signature of a path". Preprint. 2019.

[Cla+24] Marianne Clausel, Joscha Diehl, Raphael Mignot, Leonard Schmitz, Nozomi Sugiura, and Konstantin Usevich. "The barycenter in free nilpotent Lie groups and its application to iterated-integrals signatures". Preprint. 2024.

[CM] Fernando Casas and Ander Murua. *The BCH formula and the symmetric BCH formula up to terms of degree 20*. https://www.ehu.eus/ccwmuura/bch.html.

[CM09] Fernando Casas and Ander Murua. "An efficient algorithm for computing the Baker–Campbell–Hausdorff series and some of its applications". In: *Journal of Mathematical Physics* 50.3 (2009), p. 033513. DOI: 10.1063/1.3078418.

[CO22] Ilya Chevyrev and Harald Oberhauser. "Signature moments to characterize laws of stochastic processes". In: *Journal of Machine Learning Research* 23.176 (2022), pp. 1–42.

[CRT21] Elsa Cazelles, Arnaud Robert, and Felipe Tobar. "The Wasserstein-Fourier distance for stationary time series". In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 709–721. DOI: 10.1109/TSP.2020.3046227.

[De +08] Saverio De Vito, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. "On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario". In: *Sensors and Actuators B: Chemical* 129.2 (2008), pp. 750–757. DOI: 10.1016/j.snb.2007.09.060.

[DEFT20] Joscha Diehl, Kurusch Ebrahimi-Fard, and Nikolas Tapia. "Time-warping invariants of multidimensional time series". In: *Acta Applicandae Mathematicae* 170.1 (2020), pp. 265–290. DOI: 10.1007/s10440-020-00333-x.

[Die+22] Joscha Diehl, Rosa Preiß, Michael Ruddy, and Nikolas Tapia. "The moving-frame method for the iterated-integrals signature: orthogonal invariants". In: *Foundations of Computational Mathematics* 23 (2022), pp. 1273–1333. DOI: 10.1007/s10208-022-09569-5.

[DR19] Joscha Diehl and Jeremy Reizenstein. "Invariants of multidimensional time series based on their iterated-integral signature". In: *Acta Applicandae Mathematicae* 164.1 (2019), pp. 83–122. DOI: 10.1007/s10440-018-00227-z.

[Faw02] Thomas Fawcett. "Problems in stochastic analysis: connections between rough paths and noncommutative harmonic analysis". PhD thesis. University of Oxford, 2002.

[Fer21] Adeline Fermanian. "Learning time-dependent data with the signature transform". PhD thesis. Sorbonne université, 2021.

[Fer22] Adeline Fermanian. "Functional linear regression with truncated signatures". In: *Journal of Multivariate Analysis* 192 (2022), p. 105031. DOI: 10.1016/j.jmva.2022.105031.

[Fer+23] Adeline Fermanian, Jiawei Chang, Terry Lyons, and Gérard Biau. "The insertion method to invert the signature of a path". Preprint. 2023.

[FH20] Peter Friz and Martin Hairer. *A course on rough paths*. 2nd. Universitext. Springer Cham, 2020. ISBN: 978-3-030-41555-6; 978-3-030-41556-3. DOI: 10.1007/978-3-030-41556-3.

[FHT22] Peter Friz, Paul P. Hager, and Nikolas Tapia. "Unified signature cumulants and generalized Magnus expansions". In: *Forum of Mathematics, Sigma* 10 (2022), e42. DOI: 10.1017/fms.2022.20.

[FLJ03] Tom Fletcher, Conglin Lu, and Sarang Joshi. "Statistics of shape via principal geodesic analysis on Lie groups". In: *Proceeedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*. Madison, Wisconsin, USA, 2003. DOI: 10.1109/CVPR.2003.1211342.

[FV10] Peter Friz and Nicolas Victoir. *Multidimensional stochastic processes as rough paths: theory and applications*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010. ISBN: 978-0-511-84507-9. DOI: 10.1017/CBO9780511845079.

[Gar90] Adriano M. Garsia. "Combinatorics of the free Lie algebra and the symmetric group". In: *Analysis, et Cetera*. Academic Press, 1990, pp. 309–382. ISBN: 978-0-125-74249-8. DOI: 10.1016/B978-0-12-574249-8.50021-3.

[GLM19] John Gallacher, Terry Lyons, and Paul J. Moore. "Using path signatures to predict a diagnosis of Alzheimer's disease, for the Alzheimer's disease neuroimaging initiative". In: *PLoS ONE* 14.9 (2019), e0222212. DOI: 10.1371/journal.pone.0222212.

[Goo52] Irving John Good. "Rational decisions". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 14.1 (1952), pp. 107–114.

[Gra13] Benjamin Graham. "Sparse arrays of signatures for online character recognition". Preprint. 2013.

[Hal03] Brian C. Hall. *Lie groups, Lie algebras, and representations*. 1st. Graduate Texts in Mathematics. Springer, New York, 2003. ISBN: 978-1-4419-2313-4, 978-0-387-21554-9. DOI: 10.1007/978-0-387-21554-9.

[HL10] Ben Hambly and Terry Lyons. "Uniqueness for the signature of a path of bounded variation and the reduced path group". In: *Annals of Mathematics* 171.1 (2010), pp. 109–167. DOI: 10.4007/annals.2010.171.109.

[HSS08] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning". In: *Annals of Statistics* 36.3 (2008), pp. 1171–1220. DOI: 10.1214/009053607000000677.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. 2nd. Springer Series in Statistics. Springer New York, 2009. ISBN: 978-0-387-84857-0, 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7.

[IU21] Brian Kenji Iwana and Seiichi Uchida. "An empirical survey of data augmentation for time series classification with neural networks". In: *PLoS ONE* 16.7 (2021), pp. 1–32. DOI: 10.1371/journal.pone.0254841.

[KB09] Tamara G. Kolda and Brett W. Bader. "Tensor decompositions and applications". In: *SIAM Review* 51.3 (2009), pp. 455–500. DOI: 10.1137/07070111X.

[KB17] Diederik P. Kingma and Jimmy Ba. "Adam: a method for stochastic optimization". Preprint. 2017.

[KGP01] Konstantinos Kalpakis, Dhiral Gada, and Vasundhara Puttagunta. "Distance measures for effective clustering of ARIMA time-series". In: *Proceedings of the 1st International Conference on Data Mining (ICDM)*. San Jose, California, USA, 2001, pp. 273–280. DOI: 10.1109/ICDM.2001.989529.

[Kif24] Yuri Kifer. "Limit theorems for signatures". Preprint. 2024.

[KK03] Eamonn Keogh and Shruti Kasetty. "On the need for time series data mining benchmarks: a survey and empirical demonstration". In: *Data Mining and Knowledge Discovery* 7.4 (2003), pp. 349–371. DOI: 10.1023/A:1024988512476.

[KL21] Patrick Kidger and Terry Lyons. "Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU". Preprint. 2021.

[KO19] Franz J. Kiraly and Harald Oberhauser. "Kernels for sequentially ordered data". In: *Journal of Machine Learning Research* 20.31 (2019), pp. 1–45.

[KR05] Eamonn Keogh and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping". In: *Knowledge and Information Systems* 7.3 (2005), pp. 358–386. DOI: 10.1007/s10115-004-0154-9.

[LCL07] Terry Lyons, Michael Caruana, and Thierry Lévy. *Differential equations driven by rough paths*. Springer Berlin, Heidelberg, 2007. DOI: 10.1007/978-3-540-71285-5.

[Lee18] John M. Lee. *Introduction to Riemannian manifolds*. 2nd. Graduate Texts in Mathematics. Springer Cham, 2018. ISBN: 978-3-319-91754-2;978-3-030-80106-9. DOI: 10.1007/978-3-319-91755-9.

[Lem+21] Maud Lemercier, Cristopher Salvi, Theodoros Damoulas, Edwin Bonilla, and Terry Lyons. "Distribution regression for sequential data". In: *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 130. Held online, 2021, pp. 3754–3762.

[Les+21] Julien Lesouple, Cédric Baudoin, Marc Spigai, and Jean-Yves Tourneret. "How to introduce expert feedback in one-class support vector machines for anomaly detection?" In: *Signal Processing* 188 (2021), p. 108197. DOI: 10.1016/j.sigpro.2021.108197.

[LG20] Darrick Lee and Robert Ghrist. "Path signatures on Lie groups". Preprint. 2020.

[Lia05] T. Warren Liao. "Clustering of time series data—a survey". In: *Pattern Recognition* 38.11 (2005), pp. 1857–1874. DOI: 10.1016/j.patcog.2005.01.025.

[LLN16] Daniel Levin, Terry Lyons, and Hao Ni. "Learning from the past, predicting the statistics for the future, learning an evolving system". Preprint. 2016.

[LM24] Terry Lyons and Andrew D. McLeod. "Signature methods in machine learning". Preprint. 2024.

[LN15] Terry Lyons and Hao Ni. "Expected signature of Brownian motion up to the first exit time from a bounded domain". In: *Annals of Probability* 43.5 (2015), pp. 2729–2762. DOI: 10.1214/14-AOP949.

[Lön+19] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. "sktime: A unified interface for machine learning with time series". Preprint. 2019.

[LTZ12] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation-Based Anomaly Detection". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.1 (2012), pp. 1–39. DOI: 10.1145/2133360.2133363.

[Lüt05] Helmut Lütkepohl. *New introduction to multiple time series analysis*. 1st. Springer Berlin, Heidelberg, 2005. ISBN: 978-3-540-40172-8; 978-3-540-26239-8; 978-3-540-27752-1. DOI: 10.1007/978-3-540-27752-1.

[Lux07] Ulrike von Luxburg. "A tutorial on spectral clustering". In: *Statistics and Computing* 17.4 (2007), pp. 395–416. DOI: 10.1007/s11222-007-9033-z.

[LX18] Terry Lyons and Weijun Xu. "Inverting the signature of a path". In: *Journal of the European Mathematical Society* 20.7 (2018), pp. 1655 –1687. DOI: 10.4171/JEMS/796.

[Lyo98] Terry Lyons. "Differential equations driven by rough signals". In: *Revista Matemática Iberoamericana* 14.2 (1998), pp. 215–310.

[LZJ17] Chenyang Li, Xin Zhang, and Lianwen Jin. "LPSNet: a novel log path signature feature based hand gesture recognition framework". In: *Proceedings of the 16th International Conference on Computer Vision Workshops (ICCVW)*. Venice, Italy, 2017, pp. 631–639.

[Man+23] Valérian Mangé, Jean-Yves Tourneret, Francois Vincent, Laurent Mirambell, Fabio Manzoni Vieira, and Barbara Pilastre. "Détection de comportements anormaux dans des trajectoires de navires avec one-class SVM et dynamic time warping". In: *Proceedings of the 29th Conference of the Groupe de Recherche et d'Etudes de Traitement du Signal et des Images (GRETSI)*. Grenoble, France, 2023, pp. 49–52.

[Men+19] Fanrong Meng, Guan Yuan, Shaoqian Lv, Zhixiao Wang, and Shixiong Xia. "An overview on trajectory outlier detection". In: *Artificial Intelligence Review* 52.4 (2019), pp. 2437–2456. DOI: 10.1007/s10462-018-9619-1.

[Mio+20] Nina Miolane et al. "Geomstats: a python package for Riemannian geometry in machine learning". In: *Journal of Machine Learning Research* 21.223 (2020), pp. 1–9.

[Moa02] Maher Moakher. "Means and averaging in the group of rotations". In: *SIAM Journal on Matrix Analysis and Applications* 24.1 (2002), pp. 1–16. DOI: 10.1137/S0895479801383877.

[Mor+20] James Morrill, Andrey Kormilitzin, Alejo Nevado-Holgado, Sumanth Swaminathan, Samuel D. Howison, and Terry Lyons. "Utilization of the signature method to identify the early onset of sepsis from multivariate physiological time series in critical care monitoring". In: *Critical Care Medicine* 48.10 (2020), e976–e981. DOI: 10.1097/CCM.0000000000004510.

[Mor+21] James Morrill, Adeline Fermanian, Patrick Kidger, and Terry Lyons. "A generalised signature method for multivariate time series feature extraction". Preprint. 2021.

[Ni12] Hao Ni. "The expected signature of a stochastic process". PhD thesis. Oxford University, 2012.

[PA12] Xavier Pennec and Vincent Arsigny. "Exponential barycenters of the canonical Cartan connection and invariant means on Lie groups". In: *Matrix Information Geometry*. Springer Berlin Heidelberg, 2012, pp. 123–166. ISBN: 978-3-642-30231-2,978-3-642-30232-9. DOI: 10.1007/978-3-642-30232-9_7.

[PA+18] Imanol Perez Arribas, Guy M. Goodwin, John R. Geddes, Terry Lyons, and Kate E. Saunders. "A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder". In: *Translational Psychiatry* 8.1 (2018), pp. 1–7. DOI: 10.1038/s41398-018-0334-0.

[Pal59] Richard S. Palais. "Natural operations on differential forms". In: *Transactions of the American Mathematical Society* 92.1 (1959), pp. 125–141. DOI: 10.2307/1993171.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: machine learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830.

[Pil+20] Barbara Pilastre, Loïc Boussouf, Stéphane D'Escrivan, and Jean-Yves Tourneret. "Anomaly detection in mixed telemetry data using a sparse representation and dictionary learning". In: *Signal Processing* 168 (2020), p. 107320. DOI: 10.1016/j.sigpro.2019.107320.

[PL20] Xavier Pennec and Marco Lorenzi. "Beyond Riemannian geometry: The affine connection setting for transformation groups". In: *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press, 2020, pp. 169–229. ISBN: 978-0-128-14725-2. DOI: 10.1016/B978-0-12-814725-2.00012-1.

[PMF08] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. "Trajectory-based anomalous event detection". In: *IEEE Transactions on Circuits and Systems for Video Technology* 18.11 (2008), pp. 1544–1554. DOI: 10.1109/TCSVT.2008.2005599.

[Rao58] Calyampudi Radhakrishna Rao. "Some statistical methods for comparison of growth curves". In: *Biometrics* 14.1 (1958), pp. 1–17. DOI: 10.2307/2527726.

[Reu93] Christophe Reutenauer. *Free Lie algebras*. London Mathematical Society monographs. Clarendon Press, Oxford University Press, 1993. ISBN: 0-1985-3679-8; 978-0-198-53679-6. DOI: 10.1093/oso/9780198536796.001.0001.

[RG20] Jeremy F. Reizenstein and Benjamin Graham. "Algorithm 1004: the iisignature library: efficient calculation of iterated-integral signatures and log signatures". In: *ACM Transactions On Mathematical Software (TOMS)* 46.1 (2020), pp. 1–21. DOI: 10.1145/3371237.

[RS05] James O. Ramsay and Bernard W. Silverman. *Functional data analysis*. 2nd. Springer Series in Statistics. Springer New York, 2005. ISBN: 978-0-387-40080-8. DOI: h10.1007/b98888.

[Sai+07] Salem Said, Nicolas Courty, Nicolas Le Bihan, and Stephen J. Sangwine. "Exact principal geodesic analysis for data on SO(3)". In: *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*. Poznań, Poland, 2007, pp. 1701–1705.

[Sch+01] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. "Estimating the support of a high-dimensional distribution". In: *Neural computation* 13.7 (2001), pp. 1443–1471. DOI: 10.1162/089976601750264965.

[Sha+20] Zhen Shao, Ryan Sze-Yin Chan, Thomas Cochrane, Peter Foster, and Terry Lyons. "Dimensionless anomaly detection on multivariate streams with variance norm and path signature". Preprint. 2020.

[SLN14] Stefan Sommer, François Lauze, and Mads Nielsen. "Optimization over geodesics for exact principal geodesic analysis". In: *Advances in Computational Mathematics* 40.2 (2014), pp. 283–313. ISSN: 1572-9044. DOI: 10.1007/s10444-013-9308-1.

[SO21] Alexander Schell and Harald Oberhauser. "Nonlinear independent component analysis for continuous-time signals". Preprint. 2021.

[SR15] Takaya Saito and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PLoS ONE* 10.3 (2015), pp. 1–21. DOI: 10.1371/journal.pone.0118432.

[Sug21] Nozomi Sugiura. "Clustering global ocean profiles according to temperature salinity structure". Preprint. 2021.

[Tav+20] Romain Tavenard et al. “tslearn, a machine learning toolkit for time series data”. In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6.

[TO20] Csaba Toth and Harald Oberhauser. “Bayesian learning from sequential data using Gaussian processes with signature covariances”. In: *Proceedings of the 37th International Conference on Machine Learning (ICML)*. Held online, 2020.

[Tu11] Loring W. Tu. *An introduction to manifolds*. 2nd. Universitext. Springer New York, NY, 2011. ISBN: 978-1-441-97400-6. DOI: 10.1007/978-1-4419-7400-6.

[Tu17] Loring W. Tu. *Differential geometry. Connections, curvature, and characteristic classes*. 1st. Graduate Texts in Mathematics. Springer Cham, 2017. ISBN: 978-3-319-55082-4; 978-3-319-85562-2; 978-3-319-55084-8. DOI: 10.1007/978-3-319-55084-8.

[Tuc58] Ledyard R. Tucker. “Determination of parameters of a functional relation by factor analysis”. In: *Psychometrika* 23.1 (1958), pp. 19–23.

[Yan+22] Weixin Yang, Terry Lyons, Hao Ni, Cordelia Schmid, and Lianwen Jin. “Developing the path signature methodology and its application to landmark-based human action recognition”. In: *Stochastic Analysis, Filtering, and Stochastic Optimization*. Springer, Cham, 2022, pp. 431–464. ISBN: 978-3-030-98519-6. DOI: 10.1007/978-3-030-98519-6_18.

[You36] Laurence C. Young. “An inequality of the Hölder type, connected with Stieltjes integration”. In: *Acta Mathematica* 67 (1936), pp. 251–282. DOI: 10.1007/BF02401743.

Abstract

The analysis of sequential data, or time series, is key in numerous field of applications, e.g., engineering, sociology, medicine or econometrics. Often, linear models are not sufficient to account for the complex nature of data. This has created a need for interpretable and nonlinear methods for time series analysis. In this thesis, we analyze multidimensional time series through the lens of their integrals of various moment orders, constituting their signatures, a novel method for time series analysis. Under mild conditions, signatures characterize time series uniquely, up to time reparametrization and translation, into a set of features. Due to their ability to encode nonlinear dependencies in data, signature features have improve the current state-of-the-art in a broad range of Machine Learning applications, such as distribution regression, anomaly/novelty detection, human action recognition.

Signature features lie in a nonlinear space, making their manipulation challenging from a practical perspective. First, we introduce a method to average signature features which takes into account the geometry of the space, through a finite iterative algorithm. In addition, we present a strategy to effectively reduce the dimension of signature features by adapting the Principal Component Analysis (PCA). Our approaches rely on the algebraic manipulation of signatures and local approximations. We show that this dimension reduction method allows for stability of performances while using much fewer signature features. Then, we demonstrate how signatures can be highly effective as a multiscale tool for anomaly detection, with competitive runtimes. Finally, in the last chapter, we deal with clustering of time series under perturbations and introduce similarity measures in the space of signatures that we couple with usual distance-based clustering methods.

Keywords: Time series, Iterated Integrals Signatures, Learning on manifolds, Unsupervised learning.

Résumé

L'analyse de données séquentielles, ou séries temporelles, est essentielle dans de nombreux domaines d'application, tels que l'ingénierie, la sociologie, la santé ou l'économétrie. Souvent, les modèles linéaires ne suffisent pas à rendre compte de la nature complexe des données. Cela a créé un besoin de méthodes interprétables et non linéaires pour l'analyse des séries temporelles. Dans cette thèse, nous analysons les séries temporelles multidimensionnelles sous l'angle de leurs intégrales de différents ordres de moments, constituant leurs signatures, une nouvelle méthode d'analyse des séries temporelles. Sous des hypothèses non contraignantes, les signatures caractérisent les séries temporelles de manière unique, à reparamétrisation temporelle et translation près, en un ensemble de caractéristiques. En raison de leur capacité à encoder des dépendances non linéaires dans les données, les signatures ont dépassé les performances des meilleures méthodes sur un large éventail d'applications d'apprentissage automatique, telles que la régression de lois de probabilités, la détection d'anomalies, la reconnaissance d'actions humaines.

Les signatures sont des points sur un espace non linéaire, ce qui rend leur manipulation difficile d'un point de vue pratique. Tout d'abord, nous introduisons une méthode de calcul de moyennes de signatures qui tient compte de la géométrie de l'espace, par le biais d'un algorithme itératif fini. En outre, nous présentons une stratégie permettant de réduire efficacement la dimension des signatures en adaptant l'Analyse en Composantes Principales (ACP). Nos approches reposent sur la manipulation algébrique des signatures ainsi que sur des approximations locales. Nous montrons que cette méthode de réduction de dimension permet de stabiliser les performances tout en utilisant beaucoup moins de caractéristiques de signature. Ensuite, nous démontrons comment les signatures peuvent être très efficaces en tant qu'outil multi-échelle pour la détection d'anomalies, avec des temps d'exécution compétitifs. Enfin, dans le dernier chapitre, nous traitons du partitionnement de séries temporelles soumises à des perturbations et nous introduisons des mesures de similarités dans l'espace des signatures que nous combinons aux méthodes classiques de partitionnement.

Mots clés : Séries temporelles, Signatures, Apprentissage sur variétés, Apprentissage non supervisé.