Université de Rennes
École doctorale MathStic

Habilitation à Diriger des Recherches

Apprentissage statistique et séries temporelles

Romain TAVENARD

Laboratoire LETG, UMR CNRS 6554

Soutenue le XX YYY ZZZZ

Devant le jury composé de :

Rapporteurs	Florence D'ALCHÉ	Professeur, TelecomParisTech
	Panagiotis PAPAPETROU	Professeur, Stockholm University
	Nicolas THOME	Professeur, CNAM
Examinateurs	Élisa FROMONT	Professeur, Université de Rennes 1
	Hervé JÉGOU	*Senior Researcher*, Facebook AI Research
Directeur des Recherches	Thomas CORPETTI	Directeur de Recherches, CNRS

Contents

Synthèse en français

Ce document présente une synthèse de mes travaux de recherche consacrés au développement d'outils d'apprentissage statistique pour des données structurées, comme des graphes (*cf.* Sec. 2.3) ou des séries temporelles (dans le reste du document). On peut toutefois noter que l'un des aspects de ma contribution à ce domaine de recherche n'est pas abordé dans ce document (ou seulement de manière marginale dans sa version Jupyter book). Il s'agit du développement de logiciel *open source*, notamment à travers la création et la maintenance de la librairie `tslearn` [Tavenard 2020][1].

Je réalise en écrivant ces lignes que, au cours de ces travaux, j'ai traité les séries temporelles comme si elles étaient des objets protéiformes. Tout d'abord, en termes de domaine applicatif, j'ai travaillé avec des vidéos durant mon post-doctorat à l'Idiap puis je me suis intéressé aux données environnementales (que ce soit des niveaux de concentration en polluants dans les cours d'eau, des séries d'images satellites ou des trajectoires de bateaux) lorsque j'ai rejoint le laboratoire LETG (*Littoral, Environnement, Géomatique, Télédétection*) en 2013. Plus important encore, ces applications diverses ont donné lieu à des traitements variés. Notamment, un point central est la façon dont la dimension temporelle des données est intégrée dans les représentations utilisées. Durant le post-doctorat de Pierre Gloaguen [Gloaguen 2020], pour des raisons de complexité algorithmique, nous nous sommes basés sur un pré-*clustering* des données ignorant totalement l'information temporelle pour pouvoir, dans un second temps, modéliser des segments de trajectoires à l'aide d'un modèle plus fin en temps continu. À l'autre opposé du spectre, dans le cadre des thèses d'Adeline Bailly et de Mael Guillemé [Guilleme 2019, Tavenard 2017], nous avons mis en lumière l'importance de l'information temporelle en l'incluant directement dans la représentation utilisée pour la classification. Les approches basées alignement (telles que l'algorithme *Dynamic Time Warping*) se placent en quelque sorte entre ces deux extrêmes. En effet, elles ne prennent en compte que l'ordre (et non les temps d'apparition) des éléments contenus dans les séries pour estimer la similarité entre séries. Sur la prise en compte de l'aspect temporel, il est enfin à noter que, contrairement aux autres approches présentées dans ce manuscrit, les modèles convolutionnels présentés en Sec. 3.2 reposent sur une hypothèse additionnelle d'échantillonnage régulier (*i.e.* les observations composant les séries temporelles sont acquises à intervalle de temps régulier et cet intervalle est le même pour toutes les séries présentes dans la collection).

Je me suis intéressé, plus récemment, à d'autres types de données structurées comme les graphes et il m'est apparu que la question de l'encodage de l'information de structure est là aussi un aspect important. Dans le contexte de la thèse de Titouan Vayer, nous avons utilisé des distances issues du transport optimal dont la formulation est très proche de celle du problème de *Dynamic Time Warping* (même si les méthodes de résolution diffèrent).

Dans la suite de cette synthèse, je liste mes principales contributions qui sont présentées plus en

[1] `tslearn` est une librairie Python pour l'apprentissage statistique appliqué aux séries temporelles qui fournit des outils pour le pré-traitement, l'extraction de descripteurs, le *clustering* ou la classification de séries temporelles. J'ai initié ce projet en 2017.

détail dans le corps (en anglais) de ce document – et dans les publications scientifiques associées – en les organisant en deux familles : les contributions sur les métriques d'une part et celles concernant l'apprentissage de représentations d'autre part.

Définition de métriques adaptées aux données structurées

La définition de métriques adaptées aux objets à comparer est au coeur de nombreuses méthodes d'apprentissage statistique (plus proches voisins, méthodes à noyaux, *etc.*). Lorsque l'on s'intéresse à des objets complexes, il est important d'apporter un soin particulier au choix de ces métriques afin d'exprimer une notion de similarité adaptée. Dans ce cadre, j'ai proposé des méthodes pour la comparaison de séries temporelles ou de graphes.

* * *

Tout d'abord, comme detaillé en Sec. 2.1, nous avons proposé un noyau adapté à la comparaison de séries temporelles vues comme des distributions discrètes sur l'espace produit temps-*feature*. Dans ce cadre, nous avons notamment apporté un soin particulier à la complexité en temps de la méthode retenue. Pour réduire cette complexité, nous avons utilisé des méthodes d'approximation de noyaux permettant d'obtenir une complexité en temps linéaire en la taille des séries pour la phase hors ligne de calcul des représentations et linéaire en la dimension de l'espace de représentation pour le calcul de similarité entre représentations de séries temporelles, contre un habituel coût quadratique pour les méthodes basées alignement[2].

* * *

Ensuite, comme détaillé en Sec. 2.2, je me suis intéressé à l'apport des approches basées alignement. Dans ce contexte, nous avons proposé une méthode permettant de contraindre la taille de l'alignement résultant d'une comparaison entre séries, afin d'éviter certains types d'alignement pathologiques [Zhang 2017]. L'apport dans ce cas est principalement la proposition d'un algorithme exact de complexité temporelle cubique pour le nouveau problème contraint[3].

Nous avons également utilisé les approches basées alignement, et plus particulièrement l'algorithme de *Dynamic Time Warping* (DTW) pour une application de ré-échantillonnage non linéaire de séries temporelles multivariées. Plus précisément, notre méthode fait l'hypothèse qu'il existe une modalité de référence qui puisse être utilisée pour réaligner (temporellement) les autres modalités [Dupas 2015]. Cette approche peut être vue comme une version DTW d'autres travaux utilisant le transport optimal pour des problématiques d'adaptation de domaine [Courty 2017], à la différence près que ces derniers ne reposent pas sur l'utilisation d'une modalité de référence, qui a été guidée dans notre cas par le cadre applicatif visé. En effet, il s'agissait pour nous d'aligner des profils de concentration en polluants dans des cours d'eau d'un bassin versant et l'on avait à notre disposition une modalité "débit" qui permettait d'identifier l'évolution des crues[4].

Enfin, nous nous sommes plus récemment intéressés au problème de comparaison de séries temporelles hétérogènes. Dans ce contexte, on veut comparer des séries dont les *features* ne vivent pas dans les mêmes

[2]Ce travail a fait partie de la thèse de doctorat d'Adeline Bailly. Nous avons co-encadré cette thèse avec Laetitia Chapel.

[3]Ce travail fait partie de la thèse de doctorat de Zheng Zhang. Il a été réalisé durant le séjour de Zheng au LETG en 2015-2016. Je n'ai pas été impliqué dans l'encadrement de la thèse de Zheng.

[4]Ces travaux font partie de la thèse de doctorat de Rémi Dupas (en Sciences de l'Environnement). Je n'ai pas été impliqué dans l'encadrement de la thèse de Rémi.

espaces et on cherche à aligner à la fois les espaces de *features* et les dimensions temporelles des séries. Pour cela, nous avons proposé un cadre qui combine l'optimisation sur une transformation de l'espace des *features* d'une part et le *Dynamic Time Warping* d'autre part et nous avons proposé deux algorithmes pour la résolution de ce problème : l'un sous la forme de *Block Coordinate Descent* et l'autre sous la forme de descente de gradient projeté. Ces travaux sont disponibles sous la forme de *preprint* [Vayer 2020a][5].

* * *

Enfin, comme présenté en Sec. 2.3, nous nous sommes penchés sur le cas des graphes non orientés étiquetés. En les voyant comme des distributions discrètes dans l'espace structure-*features*, nous avons proposé une nouvelle distance de transport optimal pour comparer de tels objets. Plus précisément, cette distance, appelée Fused Gromov Wasserstein, interpole entre une distance de Wasserstein qui ne prendrait en compte que les étiquettes des noeuds du graphe et une distance de Gromov-Wasserstein qui se focaliserait, elle, sur la structure du graphe, oubliant ainsi les étiquettes associées aux noeuds. Nous avons proposé un algorithme de gradient conditionnel permettant de résoudre (de manière approchée) le problème d'optimisation sous-jacent. Nous avons également présenté une résolution en forme close et de complexité quasi-linéaire du problème de Gromov-Wasserstein dans le cas mono-dimensionnel.

Apprentissage de représentations pour les séries temporelles

Une autre piste de recherche à laquelle je me suis intéressé est l'apprentissage de représentations latentes pour les séries temporelles. J'ai dans ce cadre proposé des représentations issues de modèles de mélange (*cf.* Sec. 3.1) ou de couches intermédiaires dans des réseaux de neurones (comme dans les Sec. 3.2 et Sec. 3.3).

* * *

Les *topic models* sont des modèles de mélange qui permettent de manipuler des documents vus comme des ensembles de *features* quantifiées (ou sacs de mots) et qui permettent d'extraire des thèmes (*topics*) latents, un *topic* étant une distribution dans l'espace des *features*, à partir d'un corpus de documents. Dans ces méthodes, les séries temporelles sont donc vues comme des distributions discrètes dans l'espace produit temps-*features*.

Durant mon post-doctorat, nous avons proposé une extension du modèle *Hierarchical Dirichlet Latent Semantic Motifs* (HDLSM) introduit dans [Emonet 2011] (*cf.* Sec. 3.1.1). Ce modèle génératif se base sur l'extraction de motifs temporels (par opposition aux *topics* statiques). Un échantillonnage de Gibbs est utilisé pour estimer à la fois le contenu des motifs (de nombre inconnu) et leur localisation temporelle dans les documents. Notre variante supervisée [Tavenard 2013] se base sur le modèle génératif de HDLSM en y ajoutant une composante qui met en relation les motifs extraits et la classe à prédire.

Plus récemment, j'ai été impliqué dans un projet lié à la surveillance du traffic maritime. Dans ce contexte, un enjeu majeur est l'identification automatique de flux de navigation à partir de grands nombres de trajectoires observées (*cf.* Sec. 3.1.2)[6]. Le modèle que nous avons proposé diffère du précédent en plusieurs points :

[5] Ces travaux font partie de la thèse de doctorat de Titouan Vayer. Je co-encadre Titouan avec Laetitia Chapel et Nicolas Courty.

[6] Ces travaux font partie du post-doctorat de Pierre Gloaguen. Ils ont été réalisés en collaboration avec Laetitia Chapel et Chloé Friguet.

- Il s'agit ici d'une approche non supervisée ;
- On ne cherche pas ici de motifs localisés (avec une possibilité de recouvrements entre occurrences de motifs) mais plutôt une segmentation des trajectoires en modes de mouvements homogènes;
- Chaque mode de mouvement est décrit à l'aide d'un modèle en temps continu;
- Pour permettre un meilleur passage à l'échelle, un algorithme d'inférence variationnelle stochastique est utilisé en lieu et place de l'échantillonnage de Gibbs.

* * *

J'ai également proposé l'utilisation d'architectures convolutionnelles pour le traitement de séries temporelles.

Nous avons montré dans [Le Guennec 2016] que l'utilisation de techniques d'augmentation de données était une méthode efficace d'amélioration des capacités de généralisation des réseaux convolutionnels. Les stratégies d'augmentation de données que nous avons considéré dans ces travaux couvrent les déformations temporelles locales et l'extraction de sous-fenêtres[7].

Nous nous sommes également intéressés à l'apprentissage non supervisé de représentations. Dans ce contexte, nous avons cherché à apprendre des réseaux convolutionnels en fixant la contrainte que la distance obtenue dans l'espace des *feature maps* soit aussi proche que possible d'une similarité *Dynamic Time Warping* (DTW) entre les séries d'origines. Le modèle résultant est une instance de modèle siamois et nous avons montré dans [Lods 2017] qu'un tel modèle permettait de plonger les séries dans un espace dans lequel les méthodes habituelles d'apprentissage statistique (comme les k-means par exemple) peuvent opérer[8].

Dans la littérature de classification de séries temporelles, les modèles convolutionnels qui sont utilisés possèdent la plupart du temps une couche d'aggrégation qui fait disparaitre l'information temporelle. Dans [Guilleme 2019], nous nous sommes intéressés tout particulièrement à l'importance de cette information temporelle et avons proposé une approche à base de *shapelets* aléatoires localisées[9]. Cette approche nous a amené à introduire un nouveau type de régularisation pour les réseaux de neurones qui est une modification du Sparese-Group-Lasso appelée Semi-Sparse-Group-Lasso. Cette variante permet d'induire de la sparsité non pas sur toutes les variables individuelles mais sur une sous-partie seulement de ces variables, en plus de la régularisation par groupe de variables issue du Group-Lasso[10].

Autre point caractéristique de la littérature de classification de séries temporelles, les modèles à base de *shapelets* sont historiquement utilisés pour fournir des classifications explicables sous la forme de présence / absence de motifs discriminants dans les séries. Seulement, les premières approches étaient particulièrement coûteuses en temps de calcul en raison du long processus d'énumération des *shapelets* et les approches plus récentes ont fait disparaitre l'aspect interprétable au profit de meilleures performances en classification et d'une complexité réduite. Nous avons donc proposé une approche à base de réseaux convolutionnels permettant d'apprendre des filtres de convolution ressemblant à des morceaux de séries

[7] Ce travail a été réalisé dans le cadre du stage de Master 2 d'Arthur Le Guennec. J'ai co-encadré Arthur avec Simon Malinowski.

[8] Ce travail a été réalisé dans le cadre du stage de Master 2 d'Arnaud Lods. J'ai co-encadré Arnaud avec Simon Malinowski.

[9] Les *shapelets* sont des sous-séries, similaires dans leur usage à des filtres de convolution.

[10] Ces travaux font partie de la thèse de Mael Guillemé. Je n'ai pas été impliqué directement dans l'encadrement de la thèse de Mael.

tout en conservant de bonnes performances en discrimination. Pour cela, nous utilisons un réseau adversaire dont le but est de discriminer entre les filtres appris et de vrais morceaux de séries. Ce travail est disponible sous forme de *preprint* [Wang 2020][11].

* * *

Enfin, je me suis intéressé à une tâche d'apprentissage statistique qui est spécifique aux séries temporelles : la classification précoce. La classification précoce de séries temporelles consiste à prendre une décision (attribuer une classe prédite à une série temporelle observée) le plus tôt possible dans un problème de classification.

Dans ce contexte, les travaux de Dachraoui *et al.* [Dachraoui 2015] reposent sur l'optimisation d'un coût de classification pénalisé. Leur approche utilise un *clustering* des données pour estimer une espérance de coûts futurs. Nous montrons en Sec. 3.3 que l'utilisation de ce *clustering* peut mener à des cas pathologiques. Nous avons tout d'abord travaillé sur un passage de cette méthode au cas limite dans lequel on aurait un exemple d'apprentissage par cluster [Tavenard 2016].

Plus récemment, nous nous sommes focalisés sur la proposition d'un modèle de classification précoce appris de bout en bout par descente de gradient. La méthode proposée permet un certain nombre d'améliorations par rapport aux méthodes de l'état de l'art. Tout d'abord, elle est peu gourmande en temps de calcul puisqu'elle ne nécessite pas d'apprendre un classifieur différent pour chaque taille de série. Ensuite, ce modèle, doté de deux sorties (l'une permettant de décider s'il est temps ou non de prendre une décision, l'autre permettant de prendre la décision effective le moment venu), est poussé à apprendre une représentation latente qui soit plus riche car permettant de nourrir les deux sorties du modèle. Finalement, nous proposons une nouvelle fonction de coût qui permet d'éviter certains écueils des fonctions de coût usuelles pour la classification précoce : celles-ci ont en effet facilement tendance à pousser les modèles soit à prédire ridiculement tôt soit à attendre sans raison. Ces travaux sont disponibles sous forme de *preprint* [Rußwurm 2019a][12].

Perspectives

Cette présentation de mes travaux passés me mène à dresser quelques perspectives, pour des travaux en cours ou à venir.

Tout d'abord, comme expliqué plus haut, nous avons commencé à nous intéresser à la comparaison de séries temporelles hétérogènes. La suite naturelle de ces travaux consiste à offrir la possibilité de comparer des séquences d'objets quelconques (et donc pas forcéments des séquences de vecteurs de $\mathbb{R}^p$), comme des graphes évoluant dans le temps par exemple. Ensuite, en écho aux parallèles dressés dans ce manuscrit entre DTW et distance de Wasserstein, il me semble que la DTW pourrait être un outil efficace pour aborder des problématiques d'adaptation de domaine temporel, qui sont par exemple très présentes dès lors que l'on s'intéresse à la télédétection.

À plus long terme, il me semble que la question (générale) de l'apprentissage d'une bonne manière de comparer les séries temporelles d'un jeu de données à partir des données elles-mêmes (plutôt que de pré-supposer un modèle de comparaison immuable) est une perspective importante du domaine de

[11] Ce travail fait partie de la thèse de Yichang Wang. Je co-encadre Yichang avec Élisa Fromont, Rémi Emonet et Simon Malinowski.

[12] Ces travaux font partie de la thèse de doctorat de Marc Rußwurm. Marc est un doctorant de TU Munich qui est venu en France pour une période de 4 mois en 2018-2019. J'ai co-encadré Marc avec Nicolas Courty et Sébastien Lefèvre durant son séjour en France.

la classification de séries temporelles. Enfin, l'apprentissage faiblement supervisé de représentations est également un outil central qui pourrait bénéficier à de nombreuses applications pour lesquelles la collecte de données étiquetées est coûteuse et donc peu réaliste à large échelle. Il me semble que la structure des données (temporelle comme structure de graphe) peut être dans ce cadre un guide intéressant pour apprendre des représentations pertinentes sans devoir faire appel à un volume de données étiquetées trop important.

Chapter 1

Introduction

This document is a summary of my recent work related to the design of machine learning methods specifically tailored to handle structured data such as graphs (in Sec. 2.3) or time series (in the rest of the document). Note however that one of my contributions to the field is not tackled in this document (or just marginally in its Jupyter book form). It concerns open source software development, especially through the creation and maintenance of the `tslearn` [Tavenard 2020] library.[1]

I realize while writing this document that, over the past few years, I have treated time series as if they were several different things. First, from an application point of view, I have worked with video recordings during my post-doc at Idiap and moved to earth observation time series (be it pollutant levels in water streams, satellite image time series or ship trajectories) when I joined the LETG lab (*Littoral, Environnement, Géomatique, Télédétection*) in 2013. Most importantly, these diverse applications have lead to different views over what time series can be and these views are connected to how the temporal nature of the data is included (or not) in the representation. In Pierre Gloaguen's post-doctoral work [Gloaguen 2020], for the sake of efficiency, we have relied on a fully non-temporal pre-clustering of the data so as to be able, in a refinement step, to model series segments using a continuous-time model (hence re-introducing temporal information at the sub-segment level). At the other end of the spectrum, during Adeline Bailly and Mael Guillemé's PhDs [Guilleme 2019, Tavenard 2017], we have postulated that temporal localization information was key for prediction. In these works, we hence use timestamps as additional features of the input data. Elastic alignment-based approaches (such as the well-known Dynamic Time Warping algorithm) somehow belong somewhere in-between those two extremes. Indeed, they rely solely on temporal ordering (not on timestamps) to assess similarity between series. Note also that, compared to other approaches considered in this document, convolutional models presented in Sec. 3.2 make an extra assumption about the regularity of the sampling process (*i.e.* observations in a time series are supposed to be acquired at a fixed time interval and this interval is the same for all time series in the considered collection).

I have, more recently, turned my focus to other structured data such as graphs, and it appears that choosing an adequate encoding for the structural information in this context is also a very important topic. In the context of Titouan Vayer's PhD, we have relied on the use of Optimal Transport distances that, surprisingly or not, use formulations that are very similar in spirit to that of Dynamic Time Warping.

In this document, my contributions are organized in two parts, the first one being dedicated to the design of adequate similarity measures between structured data (*i.e.* graphs and time series), and the second one focusing on methods that learn latent representations for temporal data.

[1] `tslearn` is a general-purpose Python machine learning library for time series that offers tools for pre-processing time series and extracting features from them as well as dedicated models for clustering, classification and regression. I initiated this project in 2017.

Notations

Throughout this document, the following notations are used.

A time series is a set of n timestamped features:

$$\mathbf{x} = \{(x_0, t_0), \ldots, (x_{n-1}, t_{n-1})\} \tag{1.1}$$

where all x_i lie in the same ambient space $\mathbb{R}^p$ and t_i are their associated timestamps. Time series datasets are denoted $(\mathbf{X}, \mathbf{y})$ (or just $\mathbf{X}$ for unsupervised methods) where $\mathbf{X} = \left(\mathbf{x}^{(0)}, \cdots, \mathbf{x}^{(N-1)}\right)$ is a vector of N time series (that do not necessarily share the same length) and $\mathbf{y}$ is a vector of N target values.

When subseries have to be considered, we denote by $\mathbf{x}_{t_1 \to t_2}$ the subseries extracted from $\mathbf{x}$ that starts at time index t_1 and stops at time index t_2 (excluded), and $\mathbf{x}_{\to t} = \mathbf{x}_{0 \to t}$ is a shortcut notation for the subseries that covers the first timestamps up to time index t.

Chapter 2

Defining Adequate Metrics for Structured Data

Contents

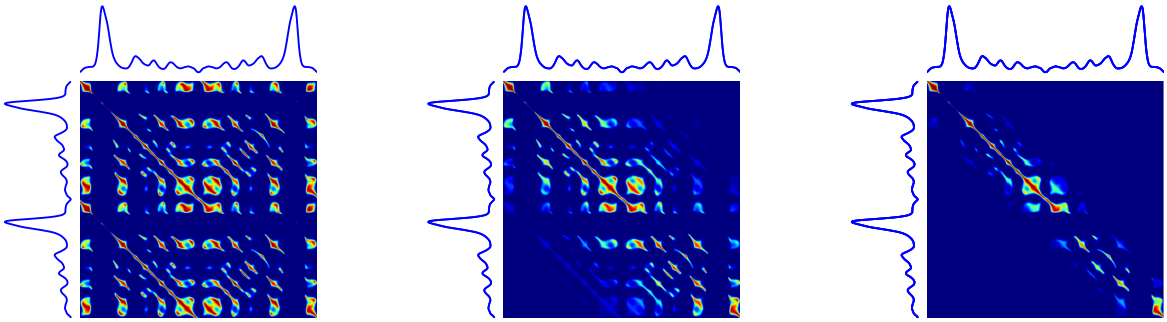
The definition of adequate metrics between objects to be compared is at the core of many machine learning methods (*e.g.*, nearest neighbors, kernel machines, *etc.*). When complex objects are involved, such metrics have to be carefully designed in order to leverage on desired notions of similarity.

This section covers my works related to the definition of new metrics for structured data such as time series or graphs. Three tracks are investigated. First, in Sec. 2.1, time series are seen as discrete distributions over the feature-time product space and a kernel that efficiently compares such representations is defined. Second, in Sec. 2.2, time series are seen as sequences, which means that only ordering is of importance (time delay between observations is ignored) and variants of the Dynamic Time Warping algorithm are used. Finally, in Sec. 2.3, undirected labeled graphs are seen as discrete distributions over the feature-structure product space and optimal transport distances are used.

2.1 A Temporal Kernel for Time Series

The method presented in this section consists in defining a kernel between sets of timestamped objects (typically features). This allows, in particular, to consider the case of irregularly sampled time series.[1]

[1] This work was part of Adeline Bailly's PhD thesis. We were co-supervising Adeline together with Laetitia Chapel.



(a) $\gamma_t = 0$ (b) Medium γ_t value (c) Large γ_t value

Figure 2.1: Effect of the γ_t parameter on the kernel matrix on which SQFD relies.

2.1.1 Match kernel and Signature Quadratic Form Distance

Our method relies on a kernel $k(\cdot, \cdot)$ between local features. Based on this local kernel, one can compute the match kernel [Bo 2009] between sets of local features as:

$$K(\mathbf{x}, \mathbf{x}') = \sum_i \sum_j k(x_i, x'_j) \tag{2.1}$$

and the Signature Quadratic Form Distance (SQFD, [Beecks 2009]) is the distance between feature sets $\mathbf{x}$ and $\mathbf{x}'$ embedded in the Reproducing Kernel Hilbert Space (RKHS) associated with K:

$$SQFD(\mathbf{x}, \mathbf{x}') = \sqrt{K(\mathbf{x}, \mathbf{x}) + K(\mathbf{x}', \mathbf{x}') - 2K(\mathbf{x}, \mathbf{x}')}\,. \tag{2.2}$$

2.1.2 Local Temporal Kernel

We introduce a time-sensitive local kernel defined as:

$$k_t((x_i, t_i), (x'_j, t'_j)) = e^{\gamma_t (t'_j - t_i)^2} k(x_i, x'_j). \tag{2.3}$$

This kernel is positive semi definite (psd), as the product of two psd kernels and, if k is the radial basis function (RBF) kernel, it can be written as:

$$k_t((x_i, t_i), (x'_j, t'_j)) = k(g(x_i, t_i), g(x'_j, t'_j)). \tag{2.4}$$

with

$$g(x_i, t_i) = \left(x_{i,0}, \ldots, x_{i,d-1}, \sqrt{\frac{\gamma_t}{\gamma_f}} t_i\right) \tag{2.5}$$

where $x_{i,l}$ denotes the l-th feature of the i-th observation in $\mathbf{x}$.

Figure 2.1 illustrates the impact of the ratio $\sqrt{\frac{\gamma_t}{\gamma_f}}$ on the kernel matrix (larger γ_t leads to paying less attention to off-diagonal elements).

k_t is then a RBF kernel itself, and Random Fourier Features [Rahimi 2008] can be used in order to approximate it with a linear kernel.

If ϕ is a feature map such that

$$k_t((x_i, t_i), (x'_j, t'_j)) \approx \left\langle \phi(g(x_i, t_i)), \phi(g(x'_j, t'_j)) \right\rangle, \tag{2.6}$$

then

$$SQFD(\mathbf{x}, \mathbf{x}') \approx \left\| \underbrace{\frac{1}{n} \sum_i \phi(g(x_i, t_i))}_{b_\phi(\mathbf{x})} - \underbrace{\frac{1}{m} \sum_j \phi(g(x'_j, t'_j))}_{b_\phi(\mathbf{x}')} \right\|. \tag{2.7}$$

In other words, once feature sets are embedded in this finite-dimensional space, approximate SQFD computation is performed through (i) a barycenter computation $b_\phi(\cdot)$ in the feature space (which can be performed offline) followed by (ii) a Euclidean distance computation with a time complexity of $O(D)$, where D is the dimension of the feature map ϕ. Overall, we have a distance between timestamped feature sets, and its precision / complexity tradeoff can be tuned via the map dimensionality D.

2.1.3 Evaluation

In order to evaluate the method presented above, we used the UCR Time Series Classification archive [Bagnall 2018], which, at the time, was made of monodimensional time series only. We decided not to work on raw data but rather extract local features to describe our time series. We chose to rely on temporal SIFT features, that we had introduced in [Bailly 2015, Bailly 2016b]. These features are straight-forward 1D adaptations of the Scale-Invariant Feature Transform (SIFT) framework introduced in Computer Vision [Lowe 2004].[2]

We show in [Tavenard 2017] that kernel approximation leads to better trade-offs in terms of computational complexity *vs.* kernel approximation than a pre-processing of the feature sets that would rely on k-means clustering. We also show that the obtained distance, once embedded in a Support Vector Machine with Gaussian kernel, leads to classification performance that is competitive with the state-of-the-art.

2.2 Dynamic Time Warping

This section covers works related to Dynamic Time Warping for time series.

2.2.1 Definition

Dynamic Time Warping (DTW, [Sakoe 1978]) is a similarity measure between time series. Consider two time series $\mathbf{x}$ and $\mathbf{x}'$ of respective lengths n and m. Here, all elements x_i and x'_j are assumed to lie in the same p-dimensional space and the exact timestamps at which observations occur are disregarded: only their ordering matters.

Optimization Problem

In the following, a path π of length K is a sequence of K index pairs $((i_0, j_0), \ldots, (i_{K-1}, j_{K-1}))$.

DTW between $\mathbf{x}$ and $\mathbf{x}'$ is formulated as the following optimization problem:

[2] Note that the use of such handcrafted features was already outdated in the computer vision community at the time of this work. However, in our small data context, they proved useful for the task at hand.

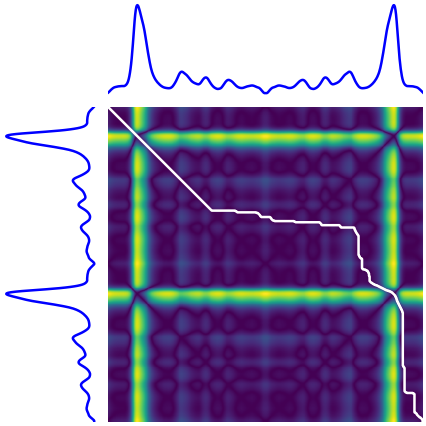


Figure 2.2: DTW path (in white) for a given pair of time series, shown on top of the cross-similarity matrix that stores $d(x_i, x'_j)$ values.

$$DTW(\mathbf{x}, \mathbf{x}') = \min_{\pi \in \mathcal{A}(\mathbf{x}, \mathbf{x}')} \sqrt{\sum_{(i,j) \in \pi} d(x_i, x'_j)^2} \tag{2.8}$$

where $\mathcal{A}(\mathbf{x}, \mathbf{x}')$ is the set of all admissible paths, *i.e.* the set of paths π such that:

- π is a sequence $[\pi_0, \ldots, \pi_{K-1}]$ of index pairs $\pi_k = (i_k, j_k)$ with $0 \leq i_k < n$ and $0 \leq j_k < m$
- $\pi_0 = (0, 0)$ and $\pi_{K-1} = (n-1, m-1)$
- for all $k > 0$, $\pi_k = (i_k, j_k)$ is related to $\pi_{k-1} = (i_{k-1}, j_{k-1})$ as follows:
  - $i_{k-1} \leq i_k \leq i_{k-1} + 1$
  - $j_{k-1} \leq j_k \leq j_{k-1} + 1$

Here, a path can be seen as a temporal alignment of time series and the optimal path (as presented in Figure 2.2) is such that Euclidean distance between aligned (*ie.* resampled) time series is minimal.

Algorithmic Solution

There exists an $O(mn)$ algorithm to compute the exact optimum for this problem, assuming computation of $d(\cdot, \cdot)$ is $O(1)$ (see Algorithm 1).

Properties

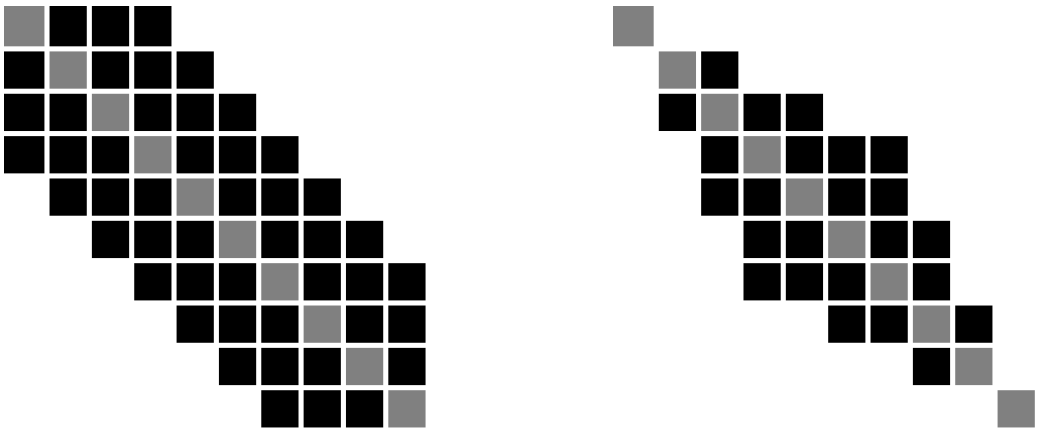
Dynamic Time Warping holds the following properties:

- $\forall \mathbf{x}, \mathbf{x}', DTW(\mathbf{x}, \mathbf{x}') \geq 0$
- $\forall \mathbf{x}, DTW(\mathbf{x}, \mathbf{x}) = 0$

However, mathematically speaking, DTW is not a valid metric since it satisfies neither the triangular inequality nor the identity of indiscernibles.

Algorithm 1: DTW algorithm. For the sake of simplicity, out-of-bound accesses to C are assumed to return ∞.

Data: $(\mathbf{x}, \mathbf{x}')$: a pair of time series
for $i = 0..n-1$ **do**
  for $j = 0..m-1$ **do**
    dist $= d(x_i, x'_j)^2$
    if $i == 0$ **and** $j == 0$ **then**
      $C_{i,j} =$ dist
    else
      $C_{i,j} =$ dist $+ \min(C_{i-1,j}, C_{i,j-1}, C_{i-1,j-1})$
    end
  end
end
return $\sqrt{C_{n-1,m-1}}$



(a) Sakoe-Chiba band of radius $r = 3$

(b) Itakura parallelogram of slope $s = 2$

Figure 2.3: Global constraints for Dynamic Time Warping.

Setting additional constraints

The set of temporal deformations to which DTW is invariant can be reduced by imposing additional constraints on the set of acceptable paths. Such constraints typically consist in forcing paths to stay close to the diagonal.

The Sakoe-Chiba band is parametrized by a radius r (number of off-diagonal elements to consider, also called warping window size sometimes), as illustrated in Figure 2.3a. The Itakura parallelogram sets a maximum slope s for alignment paths, which leads to a parallelogram-shaped constraint (see Figure 2.3b).

2.2.2 Constrained Dynamic Time Warping

In this section, we present a method to regularize Dynamic Time Warping by setting constraints on the length of the admissible warping paths [Zhang 2017].[3]

[3]This work is a part of Zheng Zhang's PhD thesis. It was performed during Zheng's stay at LETG in 2015-2016. I was not directly involved in the supervision Zheng's PhD thesis.

Algorithm 2: LDTW algorithm. For the sake of simplicity, out-of-bound accesses to C are assumed to return ∞.

Data: $(\mathbf{x}, \mathbf{x}')$: a pair of time series, $K_{\max}$: an upper bound on the path length
for $i = 0..n-1$ **do**
  for $j = 0..m-1$ **do**
    `// Set infinite cost for non-admissible lengths:`
    $C_{i,j,:} = (\infty, \cdots, \infty)$
    dist $= d(x_i, x'_j)^2$
    `// The core difference with DTW is the following loop:`
    for $l \in$ admissible_lengths(i, j, K_{max}) **do**
      if $i == 0$ **and** $j == 0$ **then**
        $C_{i,j,l} =$ dist
      else
        $C_{i,j,l} =$ dist $+ \min(C_{i-1,j,l-1}, C_{i,j-1,l-1}, C_{i-1,j-1,l-1})$
      end
    end
  end
end
return $\sqrt{\min_k C_{n-1,m-1,k}}$

Formulation and Optimization

As discussed above, a common way to restrict the set of admissible temporal distortions for Dynamic Time Warping consists in forcing paths to stay close to the diagonal through the use of Sakoe-Chiba band or Itakura parallelogram constraints. A limitation of these global constraints is that they completely discard some regions of the alignment matrix *a priori* (*i.e.* regardless of the data involved).

To alleviate this limitation, we propose Limited warping path length DTW (LDTW) that adds a path length constraint to the DTW optimization problem such that a path is said admissible for our method iff:

- it is an admissible DTW path;
- its length K is lower or equal to a user-defined bound $K_{\max}$.

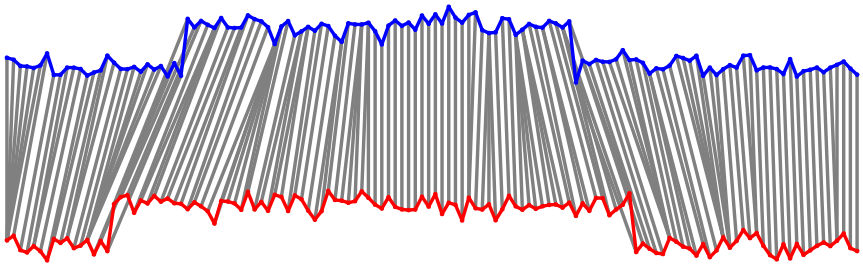
We have proposed an algorithm that stores, at each step (i, j), optimal alignment scores for all admissible alignment path lengths. This gives the general LDTW algorithm presented in Algorithm 2.

The question is then to compute the set `admissible_lengths`$(i, j, K_{\max})$. We have shown that this set can be computed explicitly and that its cardinal is $O(\min(i, j))$. Overall, we have a $O(mn^2 + nm^2)$ complexity for this exact algorithm.

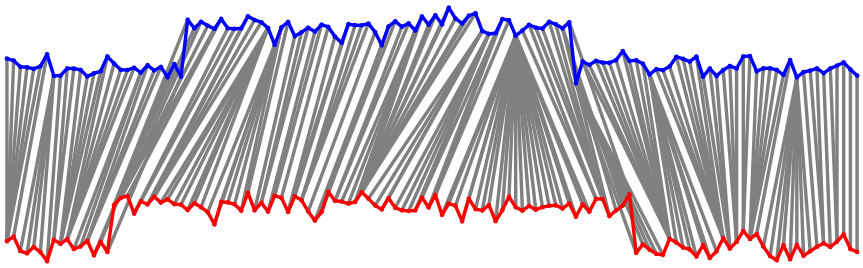
Empirical Observations

First, one can see in Figure 2.4 that the resulting alignment has indeed fewer singularities than the one obtained from DTW.

Moreover, our experiments on UCR Time Series Datasets [Bagnall 2018] show that this similarity measure, when used in a 1-Nearest Neighbor Classifier, leads to a higher accuracy than other constrained DTW variants (Sakoe-Chiba band and Itakura parallelogram).



(a) LDTW matches



(b) DTW matches

Figure 2.4: Compared matches obtained with DTW and its warping-length-constrained variant LDTW. Note the comparatively smaller temporal distortions induced by LDTW.

2.2.3 DTW Alignment as an Adaptive Resampling Strategy

In this section, we present a method that uses Dynamic Time Warping (DTW) on multimodal time series, *i.e.* time series that are made of several features recorded over time. The method relies on the assumption that one of the considered modalities (called reference modality in the following) can be used as a reference to (temporally) realign other modalities [Dupas 2015]. It has been used in the context of hydrological measurements to align pollutant concentration profiles based on discharge time series.[4]

This approach can be seen as the DTW counterpart of other works that rely on Optimal Transport for Domain Adaptation [Courty 2017]. One significant difference, however, is that it relies on a reference modality for alignment. This design choice is guided by our application context.

Motivating Use Case

Phosphorus (P) transfer during storm events represents a significant part of annual P loads in streams and contributes to eutrophication in downstream water bodies. To improve understanding of P storm dynamics, automated or semi-automated methods are needed to extract meaningful information from ever-growing water quality measurement datasets.

Clustering techniques have proven useful for identifying seasonal storm patterns and thus for increasing knowledge about seasonal variability in storm export mechanisms (*e.g.*, [Aubert 2013]). Clustering

[4]This work is a part of Rémi Dupas' PhD thesis (in Environment Sciences). I was not directly involved in the supervision of Rémi's PhD thesis.

techniques usually require calculating distances between pairs of comparable points in multiple time series. For this reason, direct clustering (without using hysteresis-descriptor variables) of high-frequency storm concentration time series is usually irrelevant because the lengths of recorded time series (number of measurement points) might differ and/or measurement points may have different positions relative to the hydrograph (flow rise and recession); hence, it is difficult to calculate a distance between pairs of comparable points.

The aim of this study was to develop a clustering method that overcomes this limit and test its ability to compare seasonal variability of P storm dynamics in two headwater watersheds. Both watersheds are ca. 5 km^2, have similar climate and geology, but differ in land use and P pressure intensity.

Alignment-based Resampling Method

In the above-described setting, we have access to one modality (discharge, commonly denoted Q) that is representative of the evolution of the flood. Temporal realignment based on this modality allows to overcome three difficulties that can arise when comparing storm-event data. Indeed, time series can have

1. different starting times due to the discharge threshold at which the samplers were triggered,
2. different lengths, and
3. differences in phase that yield different temporal localizations of the discharge peak.

To align time series, we use the path associated with DTW. This matching path can be viewed as the optimal way to perform point-wise alignment of time series.

For each discharge time series $\mathbf{x}_{\mathrm{Q}}^{(i)}$, we compute the matching path π_{Q} and use it to find the optimal alignment wrt. a fixed reference discharge time series $\mathbf{x}_{\mathrm{Q}}^{\mathrm{ref}}$. The reference discharge time series used in this study is chosen as a storm event with full coverage of flow rise and flow recession phases. Alternatively, one could choose a synthetic idealized storm hydrograph.

We then use barycentric mapping based on the obtained matches to realign other modalities to the timestamps of the reference time series, as shown in Figure 2.5.

At this point, each time series is transformed to series of n p-dimensional measurements, where n is the length of the reference discharge time series and p is the number of water quality parameters considered in the study (*i.e.* all modalities except the discharge). In a second step, a standard k-means algorithm is used to cluster realigned time series. Note that a Euclidean distance can be used for clustering since time series have already been temporally realigned; hence, time-sensitive metrics (such as DTW) are no longer needed.

This method proved useful to extract meaningful clusters and an *a posteriori* analysis of the clusters allowed to identify the export dynamics of pollutants in different geographical areas of the study sites, which then led to management recommendations, as detailed in [Dupas 2015].

2.2.4 DTW with Global Invariances

We now turn our focus to the problem of comparing time series while taking into account both feature space transformation and temporal variability. We have proposed a framework that combines a latent global transformation of the feature space with the widely used Dynamic Time Warping (DTW). This work is available as a preprint [Vayer 2020a].[5]

[5]This work is part of Titouan Vayer's PhD thesis. We are co-supervising Titouan together with Laetitia Chapel and Nicolas Courty.

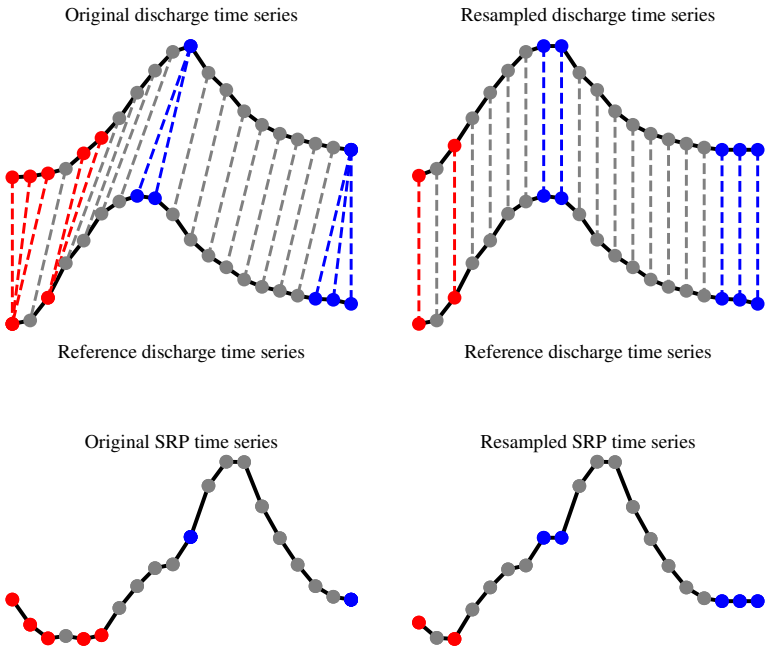


Figure 2.5: Adaptive resampling strategy. Top row: A discharge time series is resampled so that its timestamps match those of a reference one. Bottom row: The same temporal transformation is applied to all other modalities (*e.g.* SRP concentration) of the given sample.

Definition

Let $\mathbf{x}$ and $\mathbf{x}'$ be two time series of respective lengths n and m. Here, the features from the two time series are not assumed to lie in the same ambient space, but it is assumed that features from $\mathbf{x}$ lie in $\mathbb{R}^p$ while features from $\mathbf{x}'$ lie in $\mathbb{R}^{p'}$. In the following, we assume $p \geq p'$ without loss of generality. In order to allow comparison between time series $\mathbf{x}$ and $\mathbf{x}'$, we optimize on a family of functions $\mathcal{F}$ that map features from $\mathbf{x}'$ onto the feature space in which features from $\mathbf{x}$ lie. $\mathcal{F}$ is hence the family of registration functions. More formally, we define Dynamic Time Warping with Global Invariances (DTW-GI) as the solution of the following joint optimization problem:

$$\text{DTW-GI}(\mathbf{x}, \mathbf{x}') = \min_{f \in \mathcal{F}, \pi \in \mathcal{A}(\mathbf{x}, \mathbf{x}')} \sqrt{\sum_{(i,j) \in \pi} d(x_i, f(x'_j))^2}\,. \tag{2.9}$$

This similarity measure estimates both temporal alignment and feature space transformation between time series simultaneously, allowing the alignment of time series when the similarity should be defined up to a global transformation. Time series do not have to lie in the same ambient space, as presented in Figure 2.6.

Optimization

Optimization of the quantity in Equation (2.9) can be performed *via* Block Coordinate Descent. In a nutshell, the optimization process alternates between the following two steps:

1. for a fixed f, the optimal alignment path π is obtained through the DTW algorithm;

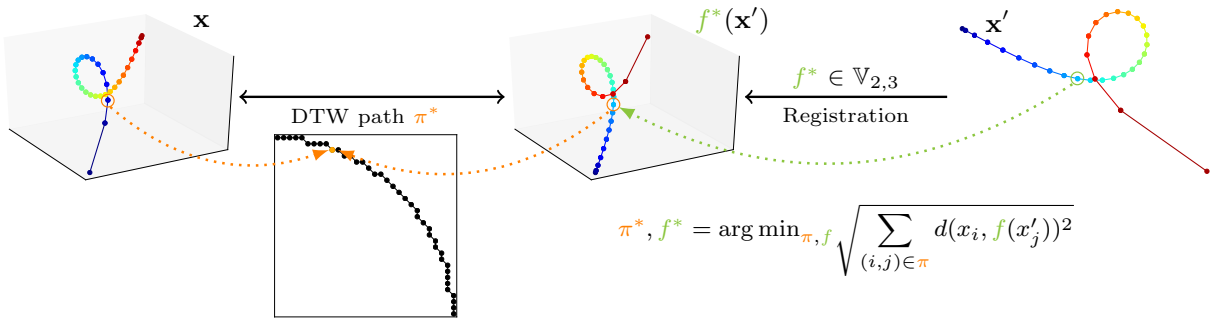


Figure 2.6: DTW-GI aligns time series by optimizing on temporal alignment (through Dynamic Time Warping) and feature space transformation (denoted f here). Time series represented here are color-coded trajectories, whose starting (resp. end) point is depicted in blue (resp. red).

2. for a fixed path π, the optimal map f (when $\mathcal{F}$ is the Stiefel manifold) is obtained through Singular Value Decomposition.

Interestingly, this optimization strategy where we alternate between time series alignment, *i.e.* time correspondences between both time series, and feature space transform optimization can be seen as a variant of the Iterative Closest Point (ICP) method in image registration [Chen 1992], in which nearest neighbors are replaced by matches resulting from DTW alignment.

We also introduce soft counterparts following the definition of softDTW from [Cuturi 2017]. In this case, we optimize on the resulting loss using projected gradient descent, and a wider variety of feature space transformation families can be considered.

To illustrate the interest of our approach, Figure 2.7 presents examples of barycenters obtained with various DTW-based barycenter computation methods. In [Vayer 2020a], we validate the utility of these similarity measures on real world datasets on the tasks of human motion prediction (where motion is captured under different points of view) and cover song identification (where song similarity is defined up to a key transposition). In both these settings, we observe that joint optimization on feature space transformation and temporal alignment improves over standard approaches that consider these as two independent steps.

2.3 Optimal Transport for Structured Data

This section covers our works related to Optimal Transport distances for structured data such as graphs. In order to compare graphs, we have introduced the Fused Gromov Wasserstein distance that interpolates between Wasserstein distance between node feature distributions and Gromov-Wasserstein distance between structures.[5] In the following, we first introduce both Wasserstein and Gromov-Wasserstein distances and some of our results concerning computational considerations related to the latter.

2.3.1 Wasserstein and Gromov-Wasserstein Distances

Let $\mu = \sum_i h_i \delta_{x_i}$ and $\mu' = \sum_i h'_i \delta_{x'_i}$ be two discrete distributions lying in the same metric space (Ω, d). The p-Wasserstein distance is defined as:

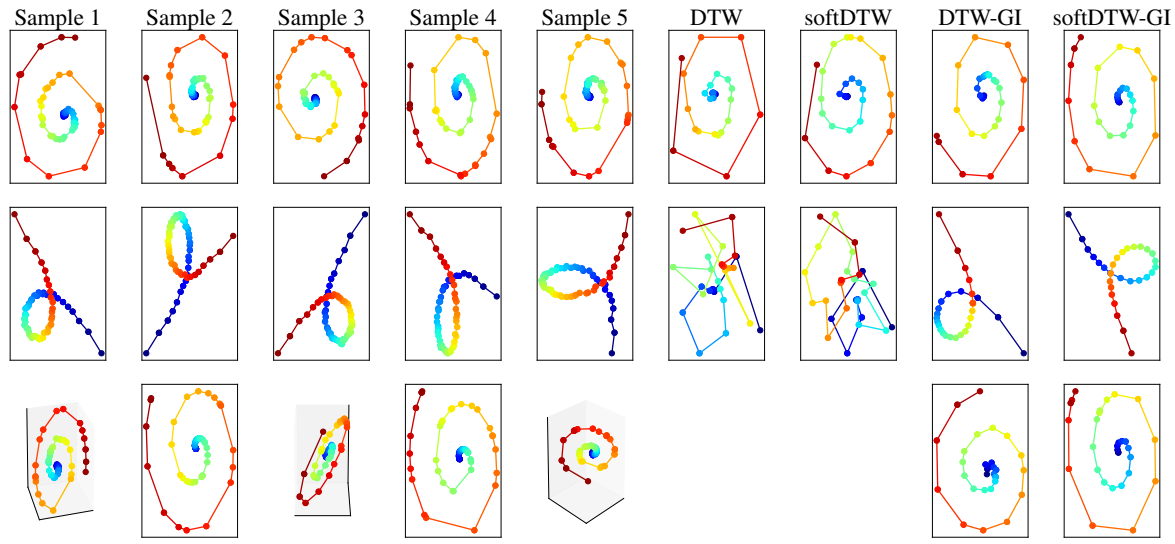


Figure 2.7: Barycenter computation using (i) DTW and softDTW baseline approaches, (ii) their rotation-invariant counterparts DTW-GI and soft-DTW-GI. Each row correspond to a different dataset, and the latter one contains both 2d and 3d trajectories, hence cannot be tackled by any baseline method. Trajectories are color-coded from blue (beginning of the series) to red (end of the series).

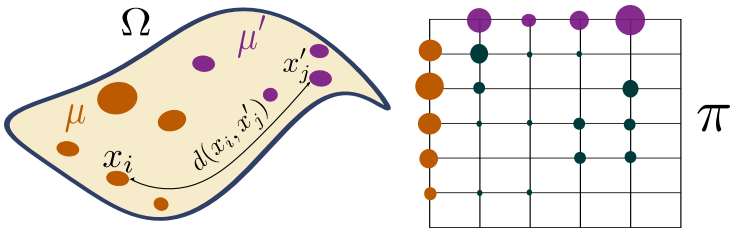


Figure 2.8: Wasserstein distance. Both distributions μ and μ' lie in the same ambient space (left) and the obtained coupling is required to meet marginal constraints (right).

$$W_p(\mu, \mu') = \min_{\pi \in \Pi(\mu,\mu')} \left(\sum_{i,j} d(x_i, x'_j)^p \pi_{i,j} \right)^{\frac{1}{p}} \tag{2.10}$$

where $\Pi(\mu, \mu')$ is the set of all admissible couplings between μ and μ' (*ie.* the set of all matrices with marginals h and h').[6] This distance is illustrated in Figure 2.8.

When distributions μ and μ' lie in distinct ambient spaces $\mathcal{X}$ and $\mathcal{X}'$, however, one cannot compute their Wasserstein distance. An alternative that was introduced in [Mémoli 2011] relies on matching intra-domain distances, as illustrated in Figure 2.9.

The corresponding distance is the Gromov-Wasserstein distance, defined as:

[6] Note that the 2-Wasserstein distance is very similar in its formulation to the Dynamic Time Warping similarity presented in Sec. 2.2. The only difference lies in the constraints that are enforced in the optimization problems. For Wasserstein, a coupling needs to meet marginal constraints to be considered valid while for Dynamic Time Warping, a path shall (i) not break the order of the sequences at stake and (ii) enforce alignment of complete series (from beginning to end).

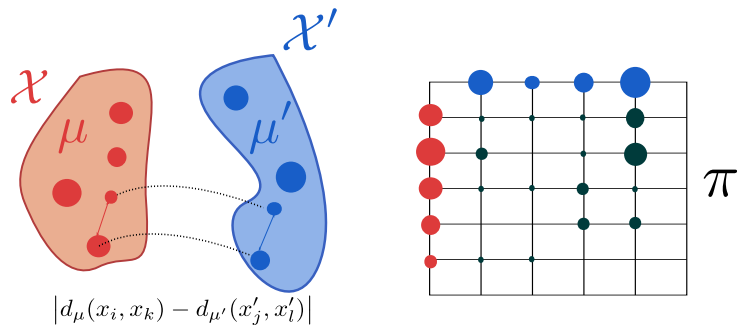


Figure 2.9: Gromov-Wasserstein distance. When distributions do not lie in the same ambient space, one can rely on intr-domain distances (left). The obtained coupling is still required to meet the same marginal constraints as for Wasserstein distance (right).

$$GW_p(\mu, \mu') = \min_{\pi \in \Pi(\mu,\mu')} \left(\sum_{i,j,k,l} \left| d_\mu(x_i, x_k) - d_{\mu'}(x'_j, x'_l) \right|^p \pi_{i,j} \pi_{k,l} \right)^{\frac{1}{p}} \tag{2.11}$$

where d_μ (resp. $d_{\mu'}$) is the metric associated to $\mathcal{X}$ (resp. $\mathcal{X}'$), the space in which μ (resp. μ') lies.

2.3.2 Sliced Gromov-Wasserstein

The computational complexity associated to the optimization problem in Equation (2.11) is high in general. However, we have shown in [Vayer 2019b] that in the mono-dimensional case, this problem can be seen as an instance of the Quadratic Assignment Problem [Koopmans 1957]. We have provided a closed form solution for this instance. In a nutshell, our solution consists in sorting mono-dimensional distributions and either matching elements from both distributions in order or in reverse order, leading to a $O(n \log n)$ algorithm that exactly solves this problem.

Based on this closed-form solution, we were able to introduce a Sliced Gromov-Wasserstein distance that, similarly to the Sliced Wasserstein distance [Rabin 2011], computes similarity between distributions through projections on random lines.

2.3.3 Fused Gromov-Wasserstein

Here, we focus on comparing structured data composed of a feature and a structure information. More formally, we consider undirected labeled graphs as tuples of the form $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \ell_f, \ell_s)$ where $(\mathcal{V}, \mathcal{E})$ are the set of vertices and edges of the graph. $\ell_f : \mathcal{V} \to \Omega_f$ is a labelling function that maps each vertex $v_i \in \mathcal{V}$ to a feature $a_i = \ell_f(v_i)$ in some feature metric space (Ω_f, d). We will denote by *feature information* the set of all the features $(a_i)_i$ of the graph. Similarly, $\ell_s : \mathcal{V} \to \Omega_s$ maps a vertex v_i from the graph to its structure representation $x_i = \ell_s(v_i)$ in some structure space (Ω_s, C) specific to each graph. $C : \Omega_s \times \Omega_s \to \mathbb{R}_+$ is a symmetric application which measures the similarity between the vertices in the graph. Unlike the feature space, however, Ω_s is implicit and in practice, knowing the similarity measure C is sufficient. With a slight abuse of notation, C will be used in the following to denote both the structure similarity measure and the matrix that encodes this similarity between pairs of nodes in the graph $\{C(i,k) = C(x_i, x_k)\}_{i,k}$. Depending on the context, C can either encode the neighborhood information of the nodes, the edge information of the graph or more generally it can model a distance between pairs of vertices such as the shortest path distance. When C is a metric, such as the shortest-path distance, we naturally equip the

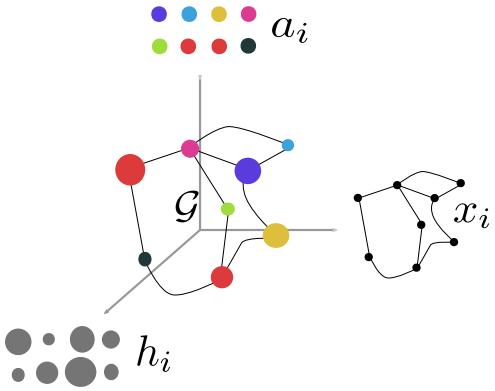


Figure 2.10: A structured object can be described by a labelled graph with $(a_i)_i$ the feature information of the object and $(x_i)_i$ the structure information. If we enrich this object with a histrogram $(h_i)_i$ for measuring the relative importance of the nodes we can represent the structured object as a fully supported probability measure μ over the product space of feature and structure.

structure with the metric space (Ω_s, C). We denote by *structure information* the set of all the structure embeddings $(x_i)_i$ of the graph. We propose to enrich the previously described graph with a histogram which serves the purpose of signaling the relative importance of the vertices in the graph. To do so, we equip graph vertices with weights $(h_i)_i$ that sum to 1.

All in all, we define *structured data* as a tuple $\mathcal{S} = (\mathcal{G}, h_\mathcal{G})$ where $\mathcal{G}$ is a graph as described above and $h_\mathcal{G}$ is a function that associates a weight to each vertex. This definition allows the graph to be represented by a fully supported probability measure over the feature-structure product space $\mu = \sum_{i=1}^{|\mathcal{V}|} h_i \delta_{(x_i, a_i)}$, where δ is the Dirac measure. This probability measure describes the entire structured data, as shown in Figure 2.10.

Distance Definition and Properties

Let $\mathcal{G}$ and $\mathcal{G}'$ be two graphs with their respective weight vectors h and h', described respectively by their probability measure $\mu = \sum_{i=1}^{|\mathcal{V}|} h_i \delta_{(x_i, a_i)}$ and $\mu' = \sum_{i=1}^{|\mathcal{V}'|} h'_i \delta_{(x'_i, a'_i)}$. Their structure matrices are denoted C and C', respectively.

We define a novel Optimal Transport discrepancy which we call the Fused Gromov-Wasserstein distance. It is defined, for a trade-off parameter $\alpha \in [0, 1]$ and order q, as

$$FGW_{q,\alpha}(\mu, \mu') = \min_{\pi \in \Pi(\mu, \mu')} E_q(\mathcal{G}, \mathcal{G}', \pi) \tag{2.12}$$

where π is a transport map (*i.e.* it has marginals h and h') and

$$E_q(\mathcal{G}, \mathcal{G}', \pi) = \sum_{i,j,k,l} (1-\alpha) d(a_i, a'_j)^q + \alpha |C(i,k) - C'(j,l)|^q \pi_{i,j} \pi_{k,l}. \tag{2.13}$$

The FGW distance looks for the coupling π between vertices of the graphs that minimizes the cost E_q which is a linear combination of a cost $d(a_i, a'_j)$ of transporting feature a_i to a'_j and a cost $|C(i,k) - C'(j,l)|$ of transporting pairs of nodes in each structure. As such, the optimal coupling tends to associate pairs of feature and structure points with similar distances within each structure pair and with similar features. As an important feature of FGW, by relying on a sum of (inter- and intra-)vertex-to-vertex distances,

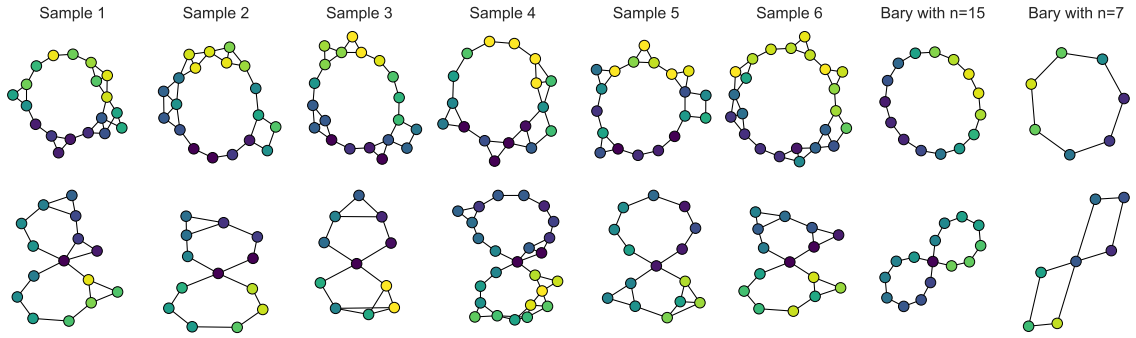


Figure 2.11: Example barycenters computed using FGW as a metric for two datasets of noisy labeled graphs. Node colors represent their mono-dimensional features and barycenter structured is recovered by thresholding the structure matrix.

it can handle structured data with continuous attributed or discrete labeled vertices (depending on the definition of d) and can also be computed even if the graphs have different numbers of nodes.

We have shown in [Vayer 2019a] that FGW holds the following properties:

- it defines a metric for $q = 1$ and a semi-metric for $q > 1$;
- varying α between 0 and 1 allows to interpolate between the Wasserstein distance between the features and the Gromov-Wasserstein distance between the structures;

In [Vayer 2020b], we also define a continuous counterpart for FGW which comes with a concentration inequality. We present a Conditional Gradient algorithm for optimization on the above-defined loss. We also provide a Block Coordinate Descent algorithm to compute graph barycenters *w.r.t.* FGW, such as the ones presented in Figure 2.11.

Results

We have shown that such barycenters can be used for graph clustering. Finally, we have exhibited classification results for FGW embedded in a Gaussian-kernel SVM which leads to state-of-the-art performance (even outperforming graph neural network approaches) on a wide range of graph classification problems.

Chapter 3

Learning Sensible Representations for Time Series

Contents

Another track of research I have been following over the past years is the learning of latent representations for time series. These latent representations can either be mixture coefficients (*cf.* Sec. 3.1) – in which case time series are represented as multinomial distributions over latent topics – or intermediate neural networks feature maps (as in Sec. 3.2 and Sec. 3.3) – and then time series are represented through filter activations they trigger.

More specifically, in Sec. 3.3, we focus on the task of early classification of time series. In this context, a method is introduced that learns an intermediate representation from which both the decision of triggering classification and the classification itself can be computed.

3.1 Temporal Topic Models

Topic models are mixture models that can deal with documents represented as bags of features (BoF) and that can extract latent topics (a topic being a distribution over features) from a corpus of documents. For these methods, time series are hence seen as bags of timestamped features. In the methods presented here, the temporal dimension is either included in the BoF representation (Sec. 3.1.1) or added in a refinement step (Sec. 3.1.2).

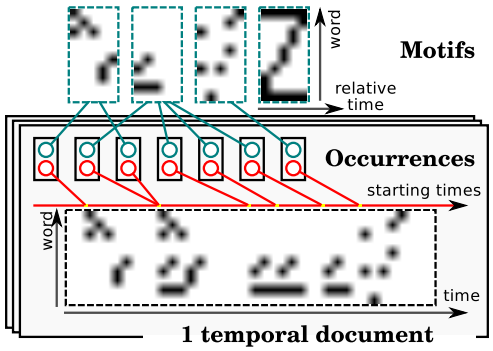


Figure 3.1: In the HDLSM model, a document is seen as a mixture of motif occurrences and motifs are shared across documents.

3.1.1 Supervised Hierarchical Dirichlet Latent Semantic Motifs

In this work, we build upon the Hierarchical Dirichlet Latent Semantic Motifs (HDLSM) topic model that was first introduced in [Emonet 2011]. This generative model relies on the extraction of motifs that encapsulate the temporal information of the data. It is able to automatically discover both the underlying number of motifs needed to model a given set of documents and the number and localization of motif occurrences in each document, as shown in the Figure 3.1.

The HDLSM model takes as input a set of quantized time series (*a.k.a.* temporal documents). More specifically, a time series is represented as a contingency table that informs, for each pair (w, t), whether word (or quantized feature) w was present in the time series at time index t (in fact, it can also account for the *amount* of presence of word w at time t).

HDLSM is a generative model. Its generative process can be described as follows:

1. Generate a list of motifs, each motif k being a 2D probability map indicating how likely it is that word w occurs at relative time t_r after the beginning of the motif.

2. For each document j, generate a list of occurrences, each occurrence having a starting time t_o and an associated motif k.

3. For each observation i in document j:

   (a) Draw an occurrence from the list,

   (b) Draw a pair (w, t_r) from the associated motif,

   (c) Generate the observation of word w at time $t = t_o + t_r$.

As stated above, motifs are represented as probabilistic maps. Each map is drawn from a Dirichlet distribution. This model makes intensive use of Dirichlet Processes (DP) to model the possibly infinite number of motifs and occurrences.

To learn the parameters of the model, Gibbs sampling is used, in which it is sufficient to re-sample motif assignments for both observations and occurrences as well as occurrence starting times. Other variables are either integrated out or deduced, when a deterministic relation holds.

Our supervised variant relies on the same generative process except that an extra component is added that maps motifs (denoted z) to classes (y) in a supervised learning context. Therefore, this mapping needs to be learned and, once the model is trained, classifying a new instance $\mathbf{x}$ consists in (i) extracting motif probabilities $P(z|\mathbf{x})$ and (ii) deriving class probabilities as:

$$P(y|\mathbf{x}) = \sum_z P(y|z)P(z|\mathbf{x}) \tag{3.1}$$

We have used this model in the context of action recognition in videos [Tavenard 2013]. Here, our *words* are quantized spatio-temporal features and each time series is the encoding of a video in which a single action is performed. In this context, we show that our model outperforms standard competitors that operate on the same quantized features.

3.1.2 Two-step Inference for Sequences of Ornstein Uhlenbeck Processes

More recently, I have been involved in a project related to the surveillance of the maritime traffic. In this context, a major challenge is the automatic identification of traffic flows from a set of observed trajectories, in order to derive good management measures or to detect abnormal or illegal behaviors for example.[1]

The model we have proposed in this context differs from the one described above in several aspects:

- The setting is unsupervised, we have no labelled data at our disposal and our goal will rather be to extract meaningful trajectory clusters;
- We are not looking for motifs to be localized in time series (with a possible overlap between motifs, as in the method described above) but rather in the segmentation of trajectories into homogeneous *movement modes*;
- Each movement mode is described using a continuous time model;
- In order to scale to larger datasets, stochastic variational inference is used (in place of Gibbs sampling) for inference.

Motivating Use Case

The monitoring of maritime traffic relies on several sources of data, in a rising context of maritime big data [Garnier 2016]. Among these sources lies the Automatic Identification System (AIS), which automatically collects messages from vessels around the world, at a high frequency. AIS data basically consist of GPS-like data, together with the instantaneous speed and heading, and some vessel specific static information. These data are characterized by their diversity as they (1) are collected at different frequencies (2) have different lengths (3) are not necessarily regularly sampled (4) represent very different behaviors, (5) share common trends or similar subparts (*movement modes*).

One major challenge in this context is the extraction of movement patterns emerging from the observed data, considering trajectories that share similar movement modes. This issue can be restated from a machine learning point of view as a large-scale clustering task. This task involves the definition of clustering methods that can handle such complex data while being efficient on large datasets, and that both cluster trajectories as a whole and detect common sub-trajectories.

[1] This work was part of Pierre Gloaguen's postdoc. This is joint work with Laetitia Chapel and Chloé Friguet.

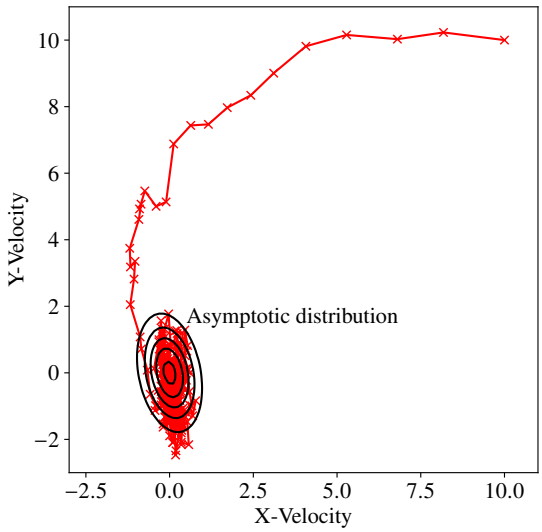


Figure 3.2: Simulation of an OUP process. The starting point is in the top-right corner and level sets of the asymptotic distribution are shown.

Model

We define a parametric framework to model trajectory data, *i.e.* sequences of geographical positions recorded through time. The modeling framework aims to account for two levels of heterogeneity possibly present in trajectory data:

1. heterogeneity of a vessel's movement within a single trajectory, and
2. heterogeneity between observed trajectories of several vessels.

Following a common paradigm, we assume that a moving vessel's trajectory is a heterogeneous sequence of patterns that we call *movement modes*. Different movement modes along a trajectory refer to different ways of moving in terms of velocity distribution, reflecting different behaviors, activities, or routes. It is assumed that a given movement mode can be adopted by several vessels.

As done in [Gurarie 2017], we characterize movement modes using a specific correlated velocity model, defined in a continuous-time framework, namely the Ornstein-Uhlenbeck Process [Uhlenbeck 1930] (OUP). One important property of the OUP is that, under mild conditions, the velocity process is an asymptotically stationary Gaussian Process, which can be visualized in Figure 3.2.

Parameter estimation

In order to perform scalable parameter inference and clustering of both trajectories and GPS observations (into movement modes), we adopt a pragmatic two step approach that takes advantage of the inherent properties of the OUP:

1. A first clustering is performed based on a simpler independent Gaussian Mixture Model, in order to estimate potential movement modes and trajectory clusters: it removes within mode autocorrelation

in the inference, and therefore facilitates the computations, yet it does not rely on any temporal or sequential information. Here again, we use a Hierarchical Dirichlet Process as a model for this two-level clustering, hence allowing for infinite mixtures of both movement modes and trajectory clusters. The Gaussian hypothesis in this case is in line with our choice of the OUP as our velocity process, since the OUP stationary distribution is Gaussian (see Figure 3.2).

2. Among the estimated movement modes, only those meeting a temporal consistency constraint are kept. Parameters of these consistent movement modes are then estimated, and used to reassign observations that were assigned to inconsistent movement modes (*i.e.* movement modes that do not last long enough to be considered reliable). It ensures that only trajectory segments for which the stationary distribution is reached are kept to estimate movement modes.

The resulting consistent movement mode concept allows one to (1) have a good estimation of OUP parameters within a movement mode (as a consistent sequence will often be related to a large amount of points) and (2) filter out "noise" movement modes gathering few observations in a temporally inconsistent manner.

Parameter estimation for Step 1 described above is performed through stochastic variational inference (SVI) to allow scalability to large datasets of AIS data, and movement mode parameter estimation is performed using standard tools from the OUP literature.

The clustering step is predominant in the overall computational complexity at inference time, since the OUP parameter estimation can be performed independently for each movement mode. It is quasilinear in the number of observations and, as stochastic variational inference is used, parts of the computations involved can easily be distributed.

Results

We have provided a dataset of several million observations in the AIS context. This dataset is used in [Gloaguen 2020] to validate our model qualitatively (through visual analysis of extracted movement modes and trajectory clusters) and compare it to a standard k-means clustering. We intend to make this dataset a reference for future competitive methods to compare on a real-world large-scale trajectory dataset.

3.2 Shapelet-based Representations and Convolutional Models

In this section, we will cover works that either relate to the Shapelet representation for time series or to the family of (1D) Convolutional Neural Networks, since these two families of methods are very similar in spirit [Lods 2017].

3.2.1 Data Augmentation for Time Series Classification

We have shown in [Le Guennec 2016] that augmenting time series classification datasets was an efficient way to improve generalization for Convolutional Neural Networks. The data augmentation strategies that were investigated in this work are local warping and window slicing, and they both lead to improvements.[2]

[2]This work was part of Arthur Le Guennec's Master internship. We were co-supervising Arthur together with Simon Malinowski.

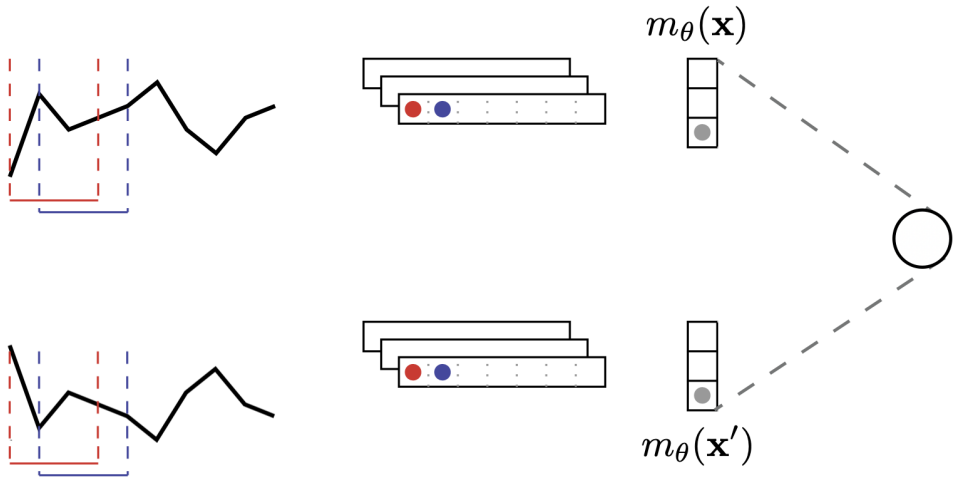


Figure 3.3: Shapelet extraction using a Siamese Network.

3.2.2 Learning to Mimic a Target Distance

Another track of research we have lead concerns unsupervised representation learning for time series. In this context, our approach has consisted in learning a representation in order to mimic a target distance.[3]

As presented in Sec. 2.2, Dynamic Time Warping is a widely used similarity measure for time series. However, it suffers from its non differentiability and the fact that it does not satisfy the metric properties. Our goal in [Lods 2017] was to introduce a Shapelet model that extracts latent representations such that the Euclidean distance between latent representations is a tight estimate of Dynamic Time Warping between original time series.

The resulting model is an instance of a Siamese Network, as illustrated in Figure 3.3, where $m_\theta(\cdot)$ is the feature extraction part of the model that maps a time series to its shapelet transform representation. The corresponding loss function on which we optimize is hence

$$\mathcal{L}(\mathbf{x}, \mathbf{x}', \beta, \theta) = \left(DTW(\mathbf{x}, \mathbf{x}') - \beta \|m_\theta(\mathbf{x}) - m_\theta(\mathbf{x}')\|_2\right)^2 \tag{3.2}$$

where β is an additional scale parameter of the model.

We have shown that the resulting model could be used as a feature extractor on top of which a k-means clustering could operate efficiently. We have also shown in [Carlini Sperandio 2018] that this representation is useful for time series indexing tasks.[4]

3.2.3 Including Localization Information

The shapelet transform, as defined above, does not contain localization information. Several options could be considered to add such information. First, the global pooling step could be turned into local pooling to keep track of local shapelet distances. In [Guilleme 2019], we rather focused on augmenting the feature representation with shapelet match localization features.[5]

Relying on a set of p random shapelets (shapelets that are extracted uniformly at random from the set of all subseries in the training set) $\{\mathbf{s_k}\}_{k<p}$, each time series is embedded into a $2p$-dimensional feature

[3] This work was part of Arnaud Lods' Master internship. We were co-supervising Arnaud together with Simon Malinowski.
[4] This work is part of Ricardo Carlini Sperandio's PhD thesis. I am not involved in Ricardo's PhD supervision.
[5] This work is part of Mael Guillemé's PhD thesis. I was not directly involved in Mael's PhD supervision.

that stores, for each shapelet, the shapelet distance $d_{\mathbf{s_k}}(\cdot)$ as well as optimal match localization $l_{\mathbf{s_k}}(\cdot)$:

$$d_{\mathbf{s_k}}(\mathbf{x}) = \min_t \|\mathbf{x}_{t\to t+L_k} - \mathbf{s_k}\|_2 \tag{3.3}$$

$$l_{\mathbf{s_k}}(\mathbf{x}) = \arg\min_t \|\mathbf{x}_{t\to t+L_k} - \mathbf{s_k}\|_2 \tag{3.4}$$

where L_k is the length of the k-th shapelet and $\mathbf{x}_{t\to t+L_k}$ is the subseries from $\mathbf{x}$ that starts at timestamp t and has length L_k.

In the random shapelet setting, a large number of shapelets are drawn and feature selection is used afterwards to focus on most useful shapelets. In our specific context, we have introduced a structured feature selection mechanism that allows, for each shapelet, to either:

- Discard all information (match magnitude and localization),
- Keep shapelet distance information and discard localization information, or
- Keep all information (match magnitude and localization).

To do so, we have introduced a modified Sparse-Group-Lasso (called Semi-Sparse-Group-Lasso) loss that enforces sparsity only on some individual variables:

$$\mathcal{L}^{\text{SSGL}}(\mathbf{x}, y, \boldsymbol{\theta}) = \mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}) + \alpha\lambda \|\mathbf{M}_{\text{ind}}\boldsymbol{\beta}\|_1 + (1-\alpha)\lambda \sum_{k=1}^{K} \sqrt{p_k} \left\|\boldsymbol{\beta}^{(k)}\right\|_2 \tag{3.5}$$

where $\mathcal{L}(\cdot,\cdot,\cdot)$ is the unpenalized loss function, $\mathbf{M}_{\text{ind}}$ is a diagonal indicator matrix that has ones on the diagonal for features that could be discarded individually (localization features in our random shapelet case), $\boldsymbol{\theta}$ is the set of all model weights, including weights $\boldsymbol{\beta}$ that are directly connected to the features (*i.e.*, these are weights from the first layer), that are organized in K groups $\boldsymbol{\beta}^{(k)}$ of size p_k ($p_k = 2$ in the random shapelet context, each group corresponding to a different shapelet). Finally, α and λ are hyper-parameters of the method that balance regularizations.

Figure 3.4 illustrates the benefit of our SSGL regularization scheme when groups of variables exist in the data and semi-sparse assumption holds. One can notice that SSGL slightly outperforms Sparse-Group-Lasso (SGL) in terms of both Mean Squared Error (MSE) and estimation of zero coefficients.

When applied to the specific case of random shapelets, we have shown that this lead to improved accuracy as soon as datasets are large enough for coefficients to be properly estimated.

3.2.4 Learning Shapelets that Look Like Time Series Snippets

Early works on shapelet-based time series classification relied on a direct extraction of shapelets as time series snippets from the training set. Selected shapelets could be used *a posteriori* to explain the classifier's decision from realistic features. However, the shapelet enumeration and selection processes were either very costly or the selection was fast but did not yield good performance. Jointly learning a shapelet-based representation of the series in the dataset and classifying the series according to this representation [Grabocka 2014] allowed to obtain discriminative shapelets in a much more efficient way.[6]

However, if the learned shapelets are definitively discriminative, they are often very different from actual pieces of a real series in the dataset. As such, these shapelets might not be suited to explain

[6]This work is part of Yichang Wang's PhD thesis. I am co-supervising Yichang with Élisa Fromont, Rémi Emonet and Simon Malinowski.

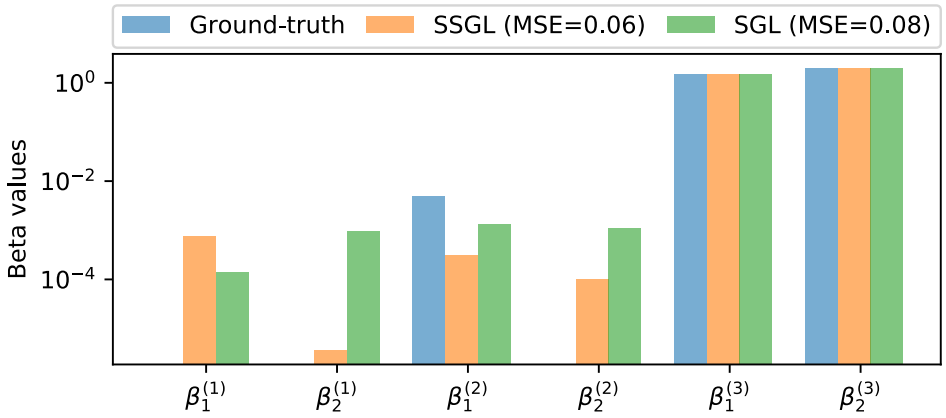


Figure 3.4: Semi-Sparse Group Lasso regularization. Coefficients learned using different regularization schemes for a linear regression problem. Ground-truth coefficients are reported in blue. Note that coefficients $\beta_1^{(1)}$, $\beta_2^{(1)}$ and $\beta_2^{(2)}$ are zero, hence not represented in logscale. For both SGL and SSGL, the goal is to estimate ground-truth coefficient, hence the closer their bars are to the blue one (or to 0 when the blue bar is not visible), the better.

a particular classifier's decision. In [Wang 2020], we rely on a simple convolutional network to classify time series and use an adversarial network that acts as a regularizer to ensure that learned shapelets are un-distinguishable from actual time series pieces from the training set.

3.3 Early Classification of Time Series

Early classification of time series is the task of performing a classification as early as possible for an incoming time series. I have worked on two methods for this task. The first one is a slight improvement over [Dachraoui 2015] and the second one relies on a representation-learning strategy.

3.3.1 Optimizing a Composite Loss for Early Classification

[Dachraoui 2015] introduces a composite loss function for early classification of time series that balances earliness and accuracy.

The cost function is of the following form:

$$\mathcal{L}(\mathbf{x}_{\rightarrow t}, y, t, \boldsymbol{\theta}) = \mathcal{L}_c(\mathbf{x}_{\rightarrow t}, y, \boldsymbol{\theta}) + \alpha t \tag{3.6}$$

where $\mathcal{L}_c(\cdot,\cdot,\cdot)$ is a classification loss and t is the time at which a decision is triggered by the system. In this setting, α drives the tradeoff between accuracy and earliness and is supposed to be a hyper-parameter of the method.

The authors rely on (i) a clustering of the training time series and (ii) individual classifiers $m_t(\cdot)$ trained at all possible timestamps, so as to be able to predict, at time t, an expected cost for all times $t+\tau$ with $\tau \geq 0$:

$$f_\tau(\mathbf{x}_{\rightarrow t}, y) = \sum_k \left[P(C_k|\mathbf{x}_{\rightarrow t}) \sum_i \left(P(y=i|C_k) \left(\sum_{j \neq i} P_{t+\tau}(\hat{y}=j|y=i, C_k) \right) \right) \right] + \alpha t \tag{3.7}$$

where:

- $P(C_k|\mathbf{x}_{\rightarrow t})$ is a soft-assignment weight of $\mathbf{x}_{\rightarrow t}$ to cluster C_k;
- $P(y = i|C_k)$ is obtained from a contingency table that stores the number of training time series of each class in each cluster;
- $P_{t+\tau}(\hat{y} = j|y = i, C_k)$ is obtained through training time confusion matrices built on time series from cluster C_k using classifier $m_{t+\tau}(\cdot)$.

At test time, if a series is observed up to time t and if, for all positive τ we have $f_\tau(\mathbf{x}_{\rightarrow t}, y) \geq f_0(\mathbf{x}_{\rightarrow t}, y)$, then a decision is made using classifier $m_t(\cdot)$.

Limitations of the Clustering

Relying on Equation (3.7) to decide prediction time can be tricky. We show in the following that in some cases (related to specific configurations of the training time confusion matrices), such an approach will lead to undesirable behaviors.[7]

Using Bayes' rule, Equation (3.7) can be re-written

$$f_\tau(\mathbf{x}_{\rightarrow t}, y) = \sum_k P(C_k|\mathbf{x}_{\rightarrow t}) \sum_i \sum_{j \neq i} P_{t+\tau}(\hat{y} = j, y = i|C_k) + \alpha t \tag{3.8}$$

$$= \sum_k P(C_k|\mathbf{x}_{\rightarrow t}) \underbrace{\sum_i 1 - P_{t+\tau}(\hat{y} = i, y = i|C_k)}_{A_{t+\tau}(C_k)} + \alpha t \tag{3.9}$$

where $A_{t+\tau}(C_k)$ is the sum of off-diagonal elements in the (normalized) training time confusion matrix built from time series in cluster k using classifier $m_{t+\tau}(\cdot)$.

In practice, this means that if the sum of off-diagonal elements of confusion matrices is equal to the same $A_{t+\tau}$ for all clusters, then this method will make a decision on the most adequate prediction time without taking the data $\mathbf{x}_{\rightarrow t}$ into account:

$$f_\tau(\mathbf{x}_{\rightarrow t}, y) = \sum_k P(C_k|\mathbf{x}_{\rightarrow t}) A_{t+\tau} + \alpha t \tag{3.10}$$

$$= A_{t+\tau} + \alpha t \tag{3.11}$$

In other words, for this method to adapt the decision time t in a data-dependent fashion, it is important that accuracy differs significantly between clusters, which is a condition that is difficult to ensure *a priori*.

3.3.2 Pushing the Method to the Limit Case

In [Tavenard 2016], we pushed this method to its limit case where the number of clusters is equal to the number of training time series. In this case, the limitation exposed above does not hold anymore.

We showed superior loss optimization capabilities with this approach, at the cost of a larger computational complexity.

[7] This unpublished note is part of François Painblanc's PhD work. We are co-supervising François together with Laetitia Chapel and Chloé Friguet.

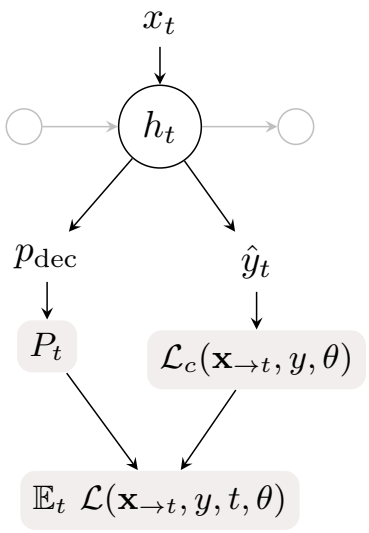


Figure 3.5: End-to-end differentiable early classification model. Grey cells correspond to quantities that are computed from the model outputs.

We also showed that in order to limit inference time complexity, one could learn a *decision triggering classifier* that, based on the time series $\mathbf{x}_{\to t}$, predicts whether a decision should be triggered or not. In this setting, the target values γ_t used to train this *decision triggering classifier* were computed from expected costs f_τ presented above:

$$\gamma_t(\mathbf{x}_{\to t}, y) = \begin{cases} 1 \text{ if } f_0(\mathbf{x}_{\to t}, y) = \min_{\tau \geq 0} f_\tau(\mathbf{x}_{\to t}, y) \\ 0 \text{ otherwise.} \end{cases} \tag{3.12}$$

In other words, decision making is here seen as a two-step process where a first classifier (*decision triggering classifier*) decides whether a decision should be made, in which case a second classifier is used to determine the class to be predicted (the latter classifier is $m_t(\cdot)$, the same as for other methods).

3.3.3 Representation Learning for Early Classification

The previous approach has several shortcomings. First, it requires to learn a classifier $m_t(\cdot)$ for each possible time series length t, which is very costly. Second, both classifiers (the one that decides whether a decision should be made, and the one that actually makes the decision) are seen as independent models, while they are, in practice, closely related. Finally, the loss function presented in Equation (3.6) requires a careful choice of hyper-parameter α that might not be easy to determine in practice.[8]

We have hence proposed a representation learning framework that addresses these three limitations [Rußwurm 2019a].

In more detail, we rely on a feature extraction module (that can either be made of causal convolutions or recurrent submodules) to extract a fixed-sized representation h_t from an incoming time series $\mathbf{x}_{\to t}$. An important point here is that this feature extractor should operate on time series whatever their length (and hence a different feature extractor need not to be learned for each time series length). Then, this feature is provided as input to two different heads, as shown in Figure 3.5:

[8] This work is part of Marc Rußwurm's PhD work. Marc is a PhD student from TU Munich who came to France for a 4-month period in 2018-2019. I was co-supervising Marc with Nicolas Courty and Sébastien Lefèvre during his stay.

- The first head (left) outputs a probability p_{dec} of making a decision at time t given that no decision has been made before: it plays the same role as the *decision triggering classifier* presented above and from the series of p_{dec} values, one can compute the probability P_t of making a decision at time t;
- The second head is the standard classification head that effectively produces a classification if the first head triggered it.

Hence, provided that we have a differentiable early classification loss function, we are able to learn all parameters of this model end-to-end. Our last contribution in this context is the design of a loss function that does not lead to useless optimal solutions (*e.g.*, trigger all classifications at the first time stamp, whatever the data). We introduced the following loss function:

$$\mathcal{L}(\mathbf{x}_{\to t}, y, t, \boldsymbol{\theta}) = \alpha \mathcal{L}_c(\mathbf{x}_{\to t}, y, \boldsymbol{\theta}) - (1-\alpha) P_{\boldsymbol{\theta}}(m_t(\mathbf{x}_{\to t}) = y) \left(\frac{T-t}{T}\right) \tag{3.13}$$

where $P_{\boldsymbol{\theta}}(m_t(\mathbf{x}_{\to t}) = y)$ is the probability (as assigned by the classification model) to generate y as an output and T is the total length of the time series (*i.e.* the maximum timestamp at which a decision can be made). The second part in this loss function is an earliness reward, which is taken into account iff the provided decision is sound (*i.e.* the correct class is predicted with non-zero probability). When the decision time is drawn from the multinomial distribution of parameters $\{P_t\}_{t \in [0, T-1]}$, the overall loss is now:

$$\mathbb{E}_{t \sim \mathcal{M}(P_0, \ldots, P_{T-1})} \mathcal{L}(\mathbf{x}_{\to t}, y, t, \boldsymbol{\theta}) \tag{3.14}$$

and gradients can be back-propagated through both heads of the model, hence allowing to jointly learn the early decision mechanism and the predictor.

We have shown that this model outperforms all known baselines in terms of both time complexity and earliness/accuracy tradeoff, especially for large scale datasets. Moreover, we have presented an application of this model to the monitoring of agriculture, and demonstrated its ability to trigger class-specific early decisions in this context in [Rußwurm 2019b].

Chapter 4

Perspectives

Contents

In this part, I will first describe some current and future works that we plan to investigate. Finally, I will discuss some more fundamental and general questions that I expect to be of importance to the future of machine learning for time series.

4.1 Current and Future Works

4.1.1 Dealing with Sequences of Arbitrary Objects

As described in Sec. 2.2.4, we have started to investigate the design of invariant alignment similarity measures. This work can be seen as a first attempt to accommodate time series alignments (such as Dynamic Time Warping) and optimal transport distances (and more specifically the work presented in [Alvarez-Melis 2019]).

One step forward in this direction is to take direct inspiration from the Gromov-Wasserstein distance presented in Sec. 2.3 for designing novel time series alignment strategies. While DTW-GI can deal with series of features that do not have the same dimension, this formulation would allow the comparison of sequences of arbitrary objects that lie in different metric spaces (not necessarily of the form $\mathbb{R}^p$), like, for example, graphs evolving over time.

Though this extension seems appealing, it would come with additional computing costs since the Bellmann recursion, which is at the core of the Dynamic Time Warping algorithm, cannot be used anymore. It is likely that approximate solvers will have to be used in this case. Also, one typical use-case for such a similarity measure would be to serve as a loss function in a forecasting setting, in which case the computational complexity would be an even higher concern which could necessitate to train dedicated Siamese networks (*e.g.* by taking inspiration from the method presented in Sec. 3.2.2).

4.1.2 Temporal Domain Adaptation

Another track of research that I am considering at the moment concerns temporal domain adaptation, that is the task of temporally realigning time series datasets in order to be able to transfer knowledge (*e.g.* a trained classifier) from one domain to the other.

In this context, and in close relation with application needs, several settings can be considered:

1. Time series can be matched with no particular constraint on temporal alignments (*i.e.* individual alignments are considered independent);
2. Time series are matched with the strong constraint that a single temporal alignment map is used for all time series comparison;
3. There exists a finite number of different temporal alignment patterns and one should extract these patterns, the matching between series of source to target datasets and the pattern used for each match.

In the first case, matching can be performed using optimal transport and DTW as the ground metric, and the method from [Courty 2017] can be used. One straightforward approach for the second case can be to alternate between (i) an optimal transport problem (finding time series pairs) for a fixed temporal realignment and (ii) a Dynamic Time Warping between synthetic series (that are built from the source and target datasets respectively) given a fixed series matching. The latter case is probably the most ambitious one, yet it is of prime importance in real-world settings such as the classification of satellite image time series. Indeed, in this context, images can contain pixels representing different land cover classes, which have different temporal responses to a given input (*e.g.* change in meteorological conditions). Hence each cluster of temporal response could be assigned a different temporal alignment pattern.

4.2 Broader Questions Related to Learning from Time Series

4.2.1 Learning the Notion of Similarity

As illustrated in this document, learning from time series can take very diverse forms depending on the invariants involved in the data. In case these invariants are known, dedicated methods can be used, yet it can be that very limited expert knowledge is available or that knowledge cannot easily guide the choice of a learning method. At the moment, this is handled through the use of ensemble techniques that cover a wide range of similarity notions [Lines 2018], yet this is at the cost of a significantly augmented complexity. More principled approaches are yet to be designed that could learn the notion of similarity from the data.

4.2.2 Structure as a Guide for Weakly-supervised Learning

Finally, learning meaningful representations in weakly supervised settings is probably one of the major challenges for the community in the coming years. Unsupervised representation learning has been overlooked in the literature up to now, despite recent advances such as [Franceschi 2019], which relies on contrastive learning.

In this context, I believe structure can be used as a guide. Typically, in the time series context, learning intermediate representations that are suited for structured prediction (*i.e.*, predicting future observations together with their emission times) is likely to capture the intrinsics of the data. Such approaches could

rely on the recent revival of time series forecasting models, such as in [Vincent 2019, Rubanova 2019]. A first step in this direction is the SOM-VAE model presented in [Fortuin 2019], which relies on a Markov assumption to model transitions between quantized latent representations.

Note that the great potential of structured prediction to learn useful representations from unsupervised datasets is not restricted to the time series context, it also holds for graphs and other kinds of structured data. Such a representation could then be used for various tasks with limited amount of supervision, in a few-shot learning fashion.

We have started investigating an instance of this paradigm in François Painblanc's PhD thesis that deals with the use of forecasting models for a better estimation of possible futures in the context of early classification.

Bibliography

[Alvarez-Melis 2019] David Alvarez-Melis, Stefanie Jegelka et Tommi S Jaakkola. *Towards optimal transport with global invariances*. In Proceedings of the International Conference on Artificial Intelligence and Statistics, 2019. (Cited on page 35.)

[Aubert 2013] Alice Aubert, Romain Tavenard, Rémi Emonet, Alban De Lavenne, Simon Malinowski, Thomas Guyet, René Quiniou, Jean-Marc Odobez, Philippe Mérot et Chantal Gascuel-Odoux. *Clustering Flood Events from Water Quality Time-Series using Latent Dirichlet Allocation Model*. Water Resources Research, vol. 49, no. 12, pages 8187–8199, 2013. (Cited on pages 15 and 46.)

[Bagnall 2018] Anthony Bagnall, Jason Lines, William Vickers et Eamonn Keogh. *The UEA & UCR Time Series Classification Repository*, 2018. (Cited on pages 11 and 14.)

[Bailly 2015] Adeline Bailly, Simon Malinowski, Romain Tavenard, Thomas Guyet et Laetitia Chapel. *Bag-of-Temporal-SIFT-Words for Time Series Classification*. In ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Porto, Portugal, Septembre 2015. (Cited on pages 11 and 45.)

[Bailly 2016a] Adeline Bailly, Damien Arvor, Laetitia Chapel et Romain Tavenard. *Classification of MODIS Time Series with Dense Bag-of-Temporal-SIFT-Words: Application to Cropland Mapping in the Brazilian Amazon*. In IEEE International Geoscience and Remote Sensing Symposium, Beijing, China, Juillet 2016. (Cited on page 46.)

[Bailly 2016b] Adeline Bailly, Simon Malinowski, Romain Tavenard, Laetitia Chapel et Thomas Guyet. *Dense Bag-of-Temporal-SIFT-Words for Time Series Classification*. In Advanced Analysis and Learning on Temporal Data. Springer, 2016. (Cited on pages 11 and 45.)

[Bailly 2017] Adeline Bailly, Laetitia Chapel, Romain Tavenard et Gustau Camps-Valls. *Nonlinear Time-Series Adaptation for Land Cover Classification*. IEEE Geoscience and Remote Sensing Letters, 2017. (Cited on page 46.)

[Beecks 2009] Christian Beecks, Merih Seran Uysal et Thomas Seidl. *Signature Quadratic Form Distances for Content-Based Similarity*. In Proceedings of the ACM International Conference on Multimedia, page 697–700, 2009. (Cited on page 10.)

[Bo 2009] Liefeng Bo et Cristian Sminchisescu. *Efficient Match Kernel between Sets of Features for Visual Recognition*. In Neural Information Processing Systems, pages 135–143. 2009. (Cited on page 10.)

[Carlini Sperandio 2018] Ricardo Carlini Sperandio, Simon Malinowski, Laurent Amsaleg et Romain Tavenard. *Time Series Retrieval using DTW-Preserving Shapelets*. In SISAP 2018 – 11th International Conference on Similarity Search and Applications, pages 257–270, Lima, Peru, Octobre 2018. Springer. (Cited on pages 28 and 46.)

[Chen 1992] Yang Chen et Gérard Medioni. *Object modelling by registration of multiple range images.* Image and Vision Computing, vol. 10, no. 3, pages 145 – 155, 1992. (Cited on page 18.)

[Courty 2017] Nicolas Courty, Rémi Flamary, Devis Tuia et Alain Rakotomamonjy. *Optimal Transport for Domain Adaptation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. (Cited on pages 2, 15 and 36.)

[Cuturi 2017] Marco Cuturi et Mathieu Blondel. *Soft-DTW: a differentiable loss function for time-series.* In Proceedings of the International Conference on Machine Learning, pages 894–903, 2017. (Cited on page 18.)

[Dachraoui 2015] Asma Dachraoui, Alexis Bondu et Antoine Cornuéjols. *Early classification of time series as a non myopic sequential decision making problem.* In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, pages 433–447. Springer, 2015. (Cited on pages 5 and 30.)

[Damodaran 2017] Bharath Bhushan Damodaran, Nicolas Courty et Romain Tavenard. *Randomized Nonlinear Component Analysis for Dimensionality Reduction of Hyperspectral Images.* In IGARSS 2017 - IEEE International Geoscience and Remote Sensing Symposium, International Geoscience and Remote Sensing Symposium, pages 1–4, Houston, United States, Juillet 2017. (Cited on page 46.)

[Dupas 2015] Rémi Dupas, Romain Tavenard, Ophélie Fovet, Nicolas Gilliet, Catherine Grimaldi et Chantal Gascuel-Odoux. *Identifying seasonal patterns of phosphorus storm dynamics with dynamic time warping.* Water Resources Research, vol. 51, no. 11, pages 8868–8882, 2015. (Cited on pages 2, 15, 16 and 46.)

[Emonet 2011] Remi Emonet, Jagannadan Varadarajan et Jean-Marc Odobez. *Extracting and Locating Temporal Motifs in Video Scenes Using a Hierarchical Non Parametric Bayesian Model.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Juin 2011. (Cited on pages 3 and 24.)

[Fortuin 2019] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann et Gunnar Rätsch. *SOM-VAE: Interpretable Discrete Representation Learning on Time Series.* In Proceedings of the International Conference on Learning Representations, 2019. (Cited on page 37.)

[Franceschi 2019] Jean-Yves Franceschi, Aymeric Dieuleveut et Martin Jaggi. *Unsupervised scalable representation learning for multivariate time series.* In Advances in Neural Information Processing Systems, pages 4652–4663, 2019. (Cited on page 36.)

[Garnier 2016] Bernard Garnier et Aldo Napoli. *Exploiting the Potential of the Future "Maritime Big Data".* In Maritime Knowledge Discovery and Anomaly Detection Workshop, 2016. (Cited on page 25.)

[Gloaguen 2020] Pierre Gloaguen, Laetitia Chapel, Chloé Friguet et Romain Tavenard. Scalable clustering of segmented trajectories within a continuous time framework. Application to maritime traffic data. 2020. (Cited on pages 1, 7, 27 and 45.)

[Grabocka 2014] Josif Grabocka, Nicolas Schilling, Martin Wistuba et Lars Schmidt-Thieme. *Learning time-series shapelets.* In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 392–401, 2014. (Cited on page 29.)

[Guijo-Rubio 2019] David Guijo-Rubio, Pedro A Gutiérrez, Romain Tavenard et Anthony Bagnall. *A Hybrid Approach to Time Series Classification with Shapelets*. In Intelligent Data Engineering and Automated Learning – IDEAL, pages 137–144, Manchester, United Kingdom, Novembre 2019. (Cited on page 45.)

[Guilleme 2019] Maël Guilleme, Simon Malinowski, Romain Tavenard et Xavier Renard. *Localized Random Shapelets*. In International Workshop on Advanced Analysis and Learning on Temporal Data, pages 85–97, 2019. (Cited on pages 1, 4, 7, 28 and 45.)

[Gurarie 2017] Eliezer Gurarie, Christen H Fleming, William F Fagan, Kristin L Laidre, Jesús Hernández-Pliego et Otso Ovaskainen. *Correlated velocity models as a fundamental unit of animal movement: synthesis and applications*. Movement ecology, vol. 5, no. 1, page 13, 2017. (Cited on page 26.)

[Jégou 2011] Hervé Jégou, Romain Tavenard, Matthijs Douze et Laurent Amsaleg. *Searching in one billion vectors: re-rank with source coding*. In ICASSP 2011 - International Conference on Acoustics, Speech and Signal Processing, pages 861–864, Prague, Czech Republic, Mai 2011. IEEE. (Cited on page 46.)

[Koopmans 1957] Tjalling C Koopmans et Martin Beckmann. *Assignment problems and the location of economic activities*. Econometrica: journal of the Econometric Society, pages 53–76, 1957. (Cited on page 20.)

[Le Guennec 2016] Arthur Le Guennec, Simon Malinowski et Romain Tavenard. *Data Augmentation for Time Series Classification using Convolutional Neural Networks*. In ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, Riva Del Garda, Italy, Septembre 2016. (Cited on pages 4, 27 and 45.)

[Lines 2018] Jason Lines, Sarah Taylor et Anthony Bagnall. *Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles*. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 12, no. 5, page 52, 2018. (Cited on page 36.)

[Lods 2017] Arnaud Lods, Simon Malinowski, Romain Tavenard et Laurent Amsaleg. *Learning DTW-Preserving Shapelets*. In IDA 2017 - 16th International Symposium on Intelligent Data Analysis, volume 10584 of *Advances in Intelligent Data Analysis XVI*, pages 198–209, London, United Kingdom, Octobre 2017. springer International Publishing. (Cited on pages 4, 27, 28 and 45.)

[Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, Novembre 2004. (Cited on page 11.)

[Malinowski 2013] Simon Malinowski, Thomas Guyet, René Quiniou et Romain Tavenard. *1d-SAX: A Novel Symbolic Representation for Time Series*. In International Symposium on Intelligent Data Analysis, numéro 12, pages 273–284, United Kingdom, 2013. (Cited on page 45.)

[Mémoli 2011] Facundo Mémoli. *Gromov–Wasserstein distances and the metric approach to object matching*. Foundations of computational mathematics, vol. 11, no. 4, pages 417–487, 2011. (Cited on page 19.)

[Pauwels 2007] Eric Pauwels, Albert Ali Salah et Romain Tavenard. *Sensor Networks for Ambient Intelligence*. In IEEE Workshop on Multimedia Signal Processing, Chania, Greece, Octobre 2007. (Cited on page 46.)

[Rabin 2011] Julien Rabin, Gabriel Peyré, Julie Delon et Marc Bernot. *Wasserstein barycenter and its application to texture mixing.* In International Conference on Scale Space and Variational Methods in Computer Vision, pages 435–446. Springer, 2011. (Cited on page 20.)

[Rahimi 2008] Ali Rahimi et Benjamin Recht. *Random Features for Large-Scale Kernel Machines.* In Neural Information Processing Systems, pages 1177–1184. 2008. (Cited on page 10.)

[Rubanova 2019] Yulia Rubanova, Tian Qi Chen et David K Duvenaud. *Latent Ordinary Differential Equations for Irregularly-Sampled Time Series.* In Advances in Neural Information Processing Systems, pages 5321–5331, 2019. (Cited on page 37.)

[Rußwurm 2019a] Marc Rußwurm, Sébastien Lefevre, Nicolas Courty, Rémi Emonet, Marco Körner et Romain Tavenard. End-to-end Learning for Early Classification of Time Series. working paper or preprint, Juillet 2019. (Cited on pages 5, 32 and 45.)

[Rußwurm 2019b] Marc Rußwurm, Romain Tavenard, Sébastien Lefèvre et Marco Körner. *Early Classification for Agricultural Monitoring from Satellite Time Series.* In AI for Social Good Workshop at International Conference on Machine Learning (ICML), Long Beach, United States, 2019. (Cited on pages 33 and 46.)

[Sakoe 1978] Hiroaki Sakoe et Seibi Chiba. *Dynamic programming algorithm optimization for spoken word recognition.* IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 26, no. 1, pages 43–49, 1978. (Cited on page 11.)

[Salah 2010] Albert Ali Salah, Eric Pauwels, Romain Tavenard et Theo Gevers. *T-Patterns Revisited: Mining for Temporal Patterns in Sensor Data.* Sensors, vol. 10, no. 8, pages 7496–7513, 2010. (Cited on page 46.)

[Tavenard 2007] Romain Tavenard, Albert Ali Salah et Eric Pauwels. *Searching for Temporal Patterns in AmI Sensor Data.* In Constructing Ambient Intelligence, pages 53–62, Darmstadt, Germany, Novembre 2007. (Cited on page 46.)

[Tavenard 2009] Romain Tavenard, Laurent Amsaleg et Guillaume Gravier. *Model-based similarity estimation of multidimensional temporal sequences.* Annals of Telecommunications - annales des télécommunications, vol. 64, no. 5, pages 381–390, Juin 2009. (Cited on page 46.)

[Tavenard 2011] Romain Tavenard, Hervé Jégou et Laurent Amsaleg. *Balancing clusters to reduce response time variability in large scale image search.* In International Workshop on Content-Based Multimedia Indexing (CBMI 2011), Madrid, Spain, Juin 2011. (Cited on page 46.)

[Tavenard 2013] Romain Tavenard, Rémi Emonet et Jean-Marc Odobez. *Time-Sensitive Topic Models for Action Recognition in Videos.* In Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 2013. (Cited on pages 3, 25 and 45.)

[Tavenard 2015] Romain Tavenard et Laurent Amsaleg. *Improving the Efficiency of Traditional DTW Accelerators.* Knowledge and Information Systems (KAIS), vol. 42, no. 1, pages 215–243, Janvier 2015. (Cited on page 46.)

[Tavenard 2016] Romain Tavenard et Simon Malinowski. *Cost-Aware Early Classification of Time Series.* In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, pages 632–647, Riva del Garda, Italy, 2016. (Cited on pages 5, 31 and 45.)

[Tavenard 2017] Romain Tavenard, Simon Malinowski, Laetitia Chapel, Adeline Bailly, Heider Sanchez et Benjamin Bustos. *Efficient Temporal Kernels between Feature Sets for Time Series Classification*. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery, Septembre 2017. (Cited on pages 1, 7, 11 and 45.)

[Tavenard 2020] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar et Eli Woods. *Tslearn, A Machine Learning Toolkit for Time Series Data*. Journal of Machine Learning Research, vol. 21, no. 118, pages 1–6, 2020. (Cited on pages 1 and 7.)

[Uhlenbeck 1930] George E Uhlenbeck et Leonard S Ornstein. *On the theory of the Brownian motion*. Physical review, vol. 36, no. 5, page 823, 1930. (Cited on page 26.)

[Vayer 2019a] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard et Nicolas Courty. *Optimal Transport for structured data with application on graphs*. In Proceedings of the International Conference on Machine Learning, pages 1–16, Long Beach, United States, Juin 2019. (Cited on pages 22 and 45.)

[Vayer 2019b] Titouan Vayer, Rémi Flamary, Romain Tavenard, Laetitia Chapel et Nicolas Courty. *Sliced Gromov-Wasserstein*. In Neural Information Processing Systems, volume 32, Vancouver, Canada, Décembre 2019. (Cited on pages 20 and 45.)

[Vayer 2020a] Titouan Vayer, Laetitia Chapel, Nicolas Courty, Rémi Flamary, Yann Soullard et Romain Tavenard. *Time Series Alignment with Global Invariances*, 2020. (Cited on pages 3, 16, 18 and 45.)

[Vayer 2020b] Titouan Vayer, Laetitia Chapel, Remi Flamary, Romain Tavenard et Nicolas Courty. *Fused Gromov-Wasserstein Distance for Structured Objects*. Algorithms, vol. 13, no. 9, page 212, 2020. (Cited on pages 22 and 45.)

[Vincent 2019] LE Vincent et Nicolas Thome. *Shape and time distortion loss for training deep time series forecasting models*. In Advances in Neural Information Processing Systems, pages 4191–4203, 2019. (Cited on page 37.)

[Wang 2020] Yichang Wang, Rémi Emonet, Élisa Fromont, Simon Malinowski et Romain Tavenard. *Learning Interpretable Shapelets for Time Series Classification through Adversarial Regularization*. In Accepted for publication in the Proceedings of the International Conference on Tools with Artificial Intelligence, 2020. (Cited on pages 5, 30 and 45.)

[Zhang 2017] Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang et Thomas Corpetti. *Dynamic time warping under limited warping path length*. Information Sciences, vol. 393, pages 91–107, 2017. (Cited on pages 2, 13 and 45.)

Annex A

Summary of Research Themes

My research activities lie around the analysis of time series. I have investigated various tasks (clustering, classification, indexing, *etc.*) and part of the developed methods have been applied in the context of environmental data.

Representations for Time Series

One of my major research interests is in the design of sensible representations for time series. In [Malinowski 2013], we have introduced a novel quantized representation for time series data. At times, we have also relied on hand-crafted time series features [Bailly 2015, Bailly 2016b] for time series classification. More recently, we have turned our focus on the widely used shapelet transform for time series classification. One track we have followed in this regard consists in bridging the gap between shapelet mining and shapelet learning approaches [Guilleme 2019, Wang 2020, Guijo-Rubio 2019]. In [Le Guennec 2016] we have proposed some of the first data augmentation techniques for time series classification using convolutional neural networks. We have also investigated the specific task of early classification [Tavenard 2016] using end-to-end trainable models [Rußwurm 2019a]. Finally, we have investigated the use of temporal topic models in both supervised [Tavenard 2013] and unsupervised [Gloaguen 2020] settings.

Metrics for Time Series

We have introduced a time-sensitive kernel in [Tavenard 2017] that relies on match kernels for the comparison of time series. We have also proposed variants of the Dynamic Time Warping (DTW) algorithm that either discard pathological paths [Zhang 2017] or allow the comparison of series that do not necessarily lie in the same ambient space [Vayer 2020a]. Then, in [Lods 2017], we have attempted to bridge the gap between representations and metrics by learning a shapelet-based representation that approximates DTW.

Machine Learning and Optimal Transport

I have recently turned my focus to other structured data such as graphs. In this context, we have designed a novel optimal transport distance that takes into account both structural information and features attached to nodes in the graphs [Vayer 2020b, Vayer 2019a]. This distance relies on Wasserstein and Gromov-Wasserstein distances. We have also shown a closed-form solution for the monodimensional Gromov-Wasserstein problem which has lead to the definition of a sliced variant for higher dimensions [Vayer 2019b].

Machine Learning for Environmental Data

We have so far mainly turned our focus on two main types of environmental data. First, we have studied chemistry data in streams using topic models [Aubert 2013] or DTW alignment strategies [Dupas 2015]. Second, we have dealt with remote sensing data (and more specifically satellite image time series) for tasks as diverse as land cover classification [Bailly 2016a, Rußwurm 2019b], domain adaptation [Bailly 2017], or dimensionality reduction [Damodaran 2017]. Also, some of my earlier works were focused on pattern mining in smart environments [Tavenard 2007, Pauwels 2007, Salah 2010].

Indexing

We have also tackled the task of indexing, with a focus on feature vector indexing by using a residual quantization strategy in [Jégou 2011] or through the design of a balanced k-means clustering [Tavenard 2011]. More specific methods dedicated to temporal data have also been investigated that rely on the use of elastic distances [Carlini Sperandio 2018, Tavenard 2015] or on trained forecasting models [Tavenard 2009].