Symbolic representations of time series

Sylvain Combettes

université PARIS-SACLAY

Symbolic representations of time series

Représentations symboliques de séries temporelles

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 574, mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques aux interfaces
Graduate School : Mathématiques
Référent : ENS Paris-Saclay

Thèse préparée dans l'unité de recherche **Centre Borelli**
(Université Paris-Saclay, CNRS, ENS Paris-Saclay)
sous la direction de **Laurent OUDRE**, Professeur des Universités
et le co-encadrement de **Charles TRUONG**, Chercheur

Thèse soutenue à Paris-Saclay, le 8 janvier 2024, par

Sylvain COMBETTES

Composition du jury

Membres du jury avec voix délibérative

Mathilde MOUGEOT Professeure des Universités, ENSIIE, France	Présidente
Germain FORESTIER Professeur des Universités, Université de Haute-Alsace, France	Rapporteur & Examinateur
Romain TAVENARD Professeur des Universités, Université de Rennes 2, France	Rapporteur & Examinateur
Themis PALPANAS Professeur des Universités, Université Paris-Cité, France	Examinateur
Patrick SCHÄFER Chercheur, Humboldt-Universität zu Berlin, Allemagne	Examinateur

THESE DE DOCTORAT
NNT : 2024UPASM002

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre : Représentations symboliques de séries temporelles
Mots clés : reconnaissance de formes, approches symboliques, détection de ruptures, apprentissage de représentation

Résumé : Les objectifs de cette thèse sont de définir de nouvelles représentations symboliques et des mesures de distance adaptées aux séries temporelles pouvant être multivariées et non-stationnaires. De plus, elles doivent préserver l'information temporelle, être interprétables et rapides à calculer. Nous passons en revue les représentations symboliques de séries temporelles, ainsi que les mesures de distance sur séries temporelles, chaînes de caractères et séquences symboliques (qui résultent d'un processus de symbolisation).

Nous proposons deux contributions : ASTRIDE pour un ensemble de séries temporelles univariées, et d_{symb} pour un ensemble de séries temporelles multivariées. Nous avons également développé le d_{symb} playground, un outil interactif en ligne permettant aux utilisateurs d'appliquer d_{symb} à leurs données téléversées. ASTRIDE et d_{symb} sont pilotées par les données, car elles utilisent la détection de ruptures pour l'étape de segmentation, puis des quantiles ou un partitionnement par les K-moyennes pour l'étape de quantification. Enfin, elles appliquent la distance d'édition générale avec des coûts personnalisés entre les séquences symboliques obtenues.

Nous montrons les performances d'ASTRIDE, comparé à 4 autres représentations symboliques, sur des tâches de reconstruction, et lorsque cela s'applique, sur des tâches de classification. Pour d_{symb}, les expériences montrent à quel point la symbolisation est interprétable. De plus, comparée à 9 distances élastiques sur une tâche de partitionnement, d_{symb} atteint des performances compétitives tout en étant plusieurs ordres de grandeur plus rapide.

Title : Symbolic representations of time series
Keywords : change-point detection, pattern recognition, symbolic approaches, representation learning

Abstract : The objectives of this thesis are to define novel symbolic representations and distance measures that are suited for time series that can be multivariate and non-stationary. In addition, they should preserve the time information, be interpretable, and fast to compute. We review symbolic representations of time series (that transform a real-valued series into a shorter discrete-valued series), as well as distances measures on time series, strings, and symbolic sequences (that result from a symbolization process).

We propose two contributions : ASTRIDE for a data set of univariate time series, and d_{symb} for a data set of multivariate time series. We also developed the d_{symb} playground, an online interactive tool that allows users to apply d_{symb} to their uploaded data. ASTRIDE and d_{symb} are data-driven as they use change-point detection for the segmentation step, then either quantiles or a K-means clustering algorithm for the quantization step. Finally, they apply the general edit distance with custom costs between the resulting symbolic sequences.

We show the performance of ASTRIDE compared to 4 other symbolic representations on reconstruction and, when applicable, on classification tasks. For d_{symb}, experiments show how interpretable the symbolization is. Moreover, compared to 9 elastic distances on a clustering task, d_{symb} achieves a competitive performance while being several orders of magnitude faster.

Contents

Remerciements (en français)

Je n'aurais évidemment pas réussi cette thèse seul. Je tiens tout d'abord à chaleureusement et énormément remercier mes directeurs de thèse Laurent Oudre et Charles Truong. J'ai tant appris d'eux, que ce soit sur le domaine de l'apprentissage automatique pour les séries temporelles, sur la recherche scientifique et ses communautés, sur les bonnes pratiques en programmation Python, ainsi que sur le plan humain. Bref, je leur dois beaucoup (ils m'ont également appris à être concis !). Je remercie Myrto Limnios ainsi que Nicolas Vayatis pour la mise en relation.

Je souhaite également remercier les membres de mon jury de thèse : la Présidente Mathilde Mougeot, les rapporteurs Germain Forestier et Romain Tavenard, ainsi que les examinateurs Themis Palpanas et Patrick Schäfer. J'ai beaucoup apprécié leurs rapports intéressants sur mon manuscrit qui ont nourri ma réflexion, ainsi que leurs questions et remarques pertinentes lors de la soutenance. Ce fut pour moi un honneur de les avoir dans mon jury, je leur souhaite le meilleur pour la suite.

Mon laboratoire, le Centre Borelli, a également joué un rôle clé dans cette thèse. Tout d'abord, j'exprime ma profonde gratitude à Nicolas Vayatis et Damien Ricard pour leurs grandes qualités de directeurs de laboratoire et leur vision, ainsi que le secrétariat – Véronique Almadovar, Alina Müller, Gwladys Stouvenel, Annabelle Azan et Annabelle Bruneau – qui ont créé un environnement propice à une recherche de qualité. Je remercie également Mathilde Mougeot qui fait vivre notre laboratoire via la chaire industrielle IDAML (qui a co-financé mon doctorat, tout comme le programme UDOPIA financé par l'ANR-20-THIA-0013-01). Merci également à Argyris Kalogeratos pour l'organisation des séminaires hebdomadaires MLMDA que j'ai pu aider à co-organiser. C'était une expérience très enrichissante et j'espère que les séminaires MLMDA continueront à animer notre labo.

Je remercie également mes collègues du Centre Borelli pour cette atmosphère conviviale ainsi que pour les nombreuses discussions passionnantes, scientifiques ou non. Je fais un clin d'œil à mes co-doctorants et amis Alexandre Bois, Thibaut Germain et Sam Perochon avec qui j'ai effectué la quasi totalité de ces 3 ans et 4 mois de thèse. Plus globalement, je remercie le groupe signal où il fait bon vivre, que Laurent Oudre et Charles Truong ont su rassembler et animer avec brio : Antoine Mazarguil, Sylvain Jung, Quentin Laborde, Lucas Zoroddu, Marion Chauveau, Brian Tervil, Mona Michaud et Pierre Humbert. Un grand merci à mon brillant co-auteur Paul Boniol qui a rejoint notre équipe pour les derniers mois de ma thèse avec qui, cerveaux alignés, nous avons publié 2 papiers. Merci Paul, j'ai énormément appris de toi, la recherche en séries temporelles a de beaux jours avec des chercheurs comme toi. En dehors du groupe signal, j'exprime ma profonde gratitude à Myrto Limnios et Tina Nikoukhah pour leurs multiples conseils et leurs bonnes humeurs et énergies, ainsi que Harry Sevi pour les échanges passionnants et passionnés. Je remercie également Jean Vassoyan,

Ioannis Bargiotas, Etienne Boursier, Samuel Gruffaz, Quentin Bammey, Matthieu Serfaty, Anis Ben Mabrouk, Marie Garin, Firas Jarboui, Batiste Le Bars, Alejandro de la Concha Duarte, Perceval Beja-Battais, Gaëtan Serré, Gwendal Debaussart, Xavier Cassagnou et Anthea Merida.

Je remercie également le groupe de doctorants, rencontrés lors de divers séminaires, que nous avons assemblé et qui est devenu une communauté, à savoir les habituels du Centre Borelli : Alexandre Bois, Sam Perochon, Thibaut Germain, Antoine Mazarguil, Sylvain Jung, Jean Vassoyan, Matthieu Serfaty, et aussi des doctorants de toute la région parisienne : Linus Bleistein, Pierre Clavier, Bastien Batardière, Daniel Mimouni, Julien Jerphanion, Alice Lacan, Louis Vincent, Tom Dupuis et Maël Bompais. Parler à des collègues dans la même galère, devenus pour beaucoup des amis, a été réconfortant et motivant.

Pendant ma thèse, j'ai également pu vivre ma passion pour l'enseignement et la pédagogie. Je remercie Erwan Le Pennec de m'avoir permis d'enseigner la science des données à l'Ecole Polytechnique Executive Education (XEXED), c'était extrêmement enrichissant à tous les niveaux. Je remercie Antoine Bichat pour la mise en relation. A l'XEXED, j'ai rencontré Mathurin Massias, avec qui j'ai ensuite enseigné le cours de "Python for Data Science" dans le cadre du MScT Data Science for Business X/HEC, aux côtés de Julien Jerphanion. J'ai beaucoup appris aux côtés d'Erwan, Mathurin et Julien, que ce soit au niveau théorique et surtout en pratique, ces échanges passionnants m'ont grandement enrichi. Mathurin, merci encore pour ta bonne humeur, ta sympathie et tes conseils ; et bonne continuation dans ta recherche !

Je n'en serais pas là aujourd'hui sans mes enseignants précédents, je les remercie profondément. Je remercie Madame Bertaux et Monsieur Paul Gozlan du collège Jean Moulin pour les Mathématiques et Madame Baulleret pour la Physique-Chimie. Je remercie Monsieur Leroy du Lycée Louis-le-Grand pour les Mathématiques ainsi que Monsieur Wittmann et Monsieur Perez pour la Physique-Chimie. En classe préparatoire au Lycée Louis-le-Grand, je remercie Monsieur Thomas Lafforgue pour les Mathématiques et Madame Christelle Poux pour la physique. Voilà ce que j'ai indiqué à leurs égards dans une lettre de motivation en juin 2017 : "J'ai eu l'opportunité d'avoir d'excellents professeurs comme M. Lafforgue en mathématiques ou Mme Poux en physique pendant 2 ans qui ont révolutionné mes méthodes de travail. Je me suis efforcé de m'approprier leurs visions des sciences, consistant à simplifier toute notion et la rendre intuitive en choisissant le bon point de vue et ne laisser aucune zone d'ombre. J'ai alors pu véritablement apprécier la beauté des raisonnements. Ils m'ont fortement influencé dans ma décision de poursuivre mes études dans les sciences fondamentales." Jusqu'à ce jour et pour ma vie, je leur dois beaucoup pour ma manière d'appréhender des problèmes complexes, scientifiques ou non. Leur passion pour la pédagogie a été contagieuse.

Je remercie également les enseignants-chercheurs des Mines de Nancy : Antoine Henrot, Frédéric Sur, Christophe Cerisara, Parisa Rastin, Sandie Ferrigno, Anne Gégout-Petit, Rémi Peyre, Madalina Deaconu et Denis Villemonais. Ils m'ont initié au monde de la recherche et à la science des données, je remercie leur implication et leur pédagogie. Je remercie également Alexandre Voisin et Pierre Vallois qui ont encadré mon projet de Master 2 sur "Comparison of Empirical Probability Distributions" qui m'a permis de m'exercer à la recherche. Je remercie évidemment à nouveau Antoine Henrot d'avoir mis en place ces projets de recherche.

J'en profite pour remercier la France pour cette éducation publique (presque) gra-

tuite de grande qualité.

Je remercie également les entreprises qui m'ont accueilli en stage. J'ai beaucoup appris lors de mon stage de fin de Master 2 dans le conseil en data chez Artefact auprès de Kasra Mansouri, Clément Ménassé, Eduardo Nischiguti, Joris Caloud, Robin Doumerc et Hanania Ouazan (la team stonks). J'ai été initié à la prédiction de séries temporelles et les bonnes pratiques du Data Scientist. Je remercie également Fabrice Couvelard, Romain Guillier et Antoine Hamon de Servier pour l'encadrement de mon stage de fin Master 1. J'ai pu plonger dans la recherche sur les generative adversarial networks (GANs).

Enfin, je remercie, de manière ardente et profonde mes proches, ma famille et mes amis. Je ne vous énumère pas ici, vous vous reconnaîtrez. Ces 3 années de thèse à Paris auraient été complètement différentes sans vous. J'ai une chance inouïe de vous avoir à mes côtés. Merci encore à tous ceux qui sont venus à ma soutenance de thèse ainsi qu'au pot (et aux autres réjouissances au cours de ces dernières années, c'était et c'est toujours une régalade).

Pour clôturer ces remerciements, je remercie tous ceux que j'ai pu oublier.

Résumé (en français)

Cette thèse traite du problème de la représentation et de la comparaison de signaux physiologiques, qu'ils soient univariés ou multivariés. Dans de nombreuses applications comme en neurologie comportementale, les chercheurs ont besoin d'analyser et de comparer de grands jeux de données de séries temporelles multivariées, de manière interactive et interprétable. Les objectifs de cette thèse sont de définir de nouvelles représentations symboliques et mesures de distances qui peuvent prendre en compte des signaux physiologiques ayant une structure complexe : multivariée et non-stationnaire. De plus, cette représentation doit préserver l'information temporelle, être interprétable et rapide à calculer.

Après avoir passé en revue les techniques de symbolisation (qui transforment une série à valeurs réelles en une série plus courte à valeurs discrètes) et avoir mené une revue de l'état de l'art sur les mesures de distance sur les séries temporelles, les chaînes de caractères et les séquences symboliques (qui résultent d'un processus de symbolisation), nous introduisons de nouvelles représentations symboliques et définissons des mesures de distance entre les séquences symboliques obtenues.

La première contribution, appelée ASTRIDE, est une représentation symbolique pour un jeu de données de séries temporelles univariées. Contrairement à la plupart des représentations symboliques, ASTRIDE est adaptative (i.e. pilotée par les données) durant l'étape de segmentation grâce à une détection de ruptures ainsi que durant l'étape de quantification en utilisant des quantiles. Au lieu de traiter chaque signal l'un après l'autre, ASTRIDE construit un dictionnaire de symboles qui est commun à tous les signaux d'un jeu de données. Nous introduisons également une nouvelle mesure de distance entre représentations symboliques qui est basée dans la distance d'édition générale, avec des poids personnalisés. Nous montrons les performances d'ASTRIDE, comparé à 4 autres représentations symboliques, sur des tâches de reconstruction, et lorsque cela s'applique, sur des tâches de classification.

La seconde contribution est une représentation symbolique pour un jeu de données de séries temporelles multivariées qui peuvent être non-stationnaires, appelée d_{symb}, qui est mise en œuvre au sein d'un outil d'exploration en ligne, appelé le d_{symb} playground. Contrairement à la plupart des mesures de distance sur des signaux multivariés, d_{symb} prend en compte leur non-stationnarité grâce à une étape de symbolisation. Cette étape est elle-même basée sur une détection de ruptures divisant un signal non stationnaire en plusieurs segments stationnaires, suivie d'une quantification à l'aide d'un partitionnement par l'algorithme des K-moyennes. La mesure de distance proposée est basée sur la distance d'édition générale. Les avantages de d_{symb} sont illustrés sur 3 jeux de données de signaux physiologiques multivariés. Les expériences montrent à quel point la symbolisation est interprétable : un simple coup d'œil aux séquences symboliques obtenues fournit une compréhension

instantanée et globale d'un jeu de données. De plus, comparée à 9 distances élastiques multivariées sur une tâche de partitionnement, d_{symb} atteint des performances compétitives tout en étant plusieurs ordres de grandeur plus rapide que les autres méthodes. Avec ces caractéristiques désirables, nous avons développé le d_{symb} playground, un outil en ligne, qui permet aux chercheurs d'appliquer d_{symb} aux données qu'ils auront téléversées.

Abstract

This work addresses the problem of representing and comparing physiological signals that can be univariate or multivariate. In many applications, such as behavioral neurology, researchers have to analyze and compare large amounts of multivariate time series in an interactive and interpretable way. The objectives of this thesis are to define novel symbolic representations and distance measures that can handle physiological signals with a complex structure: multivariate and non-stationary. Moreover, the representation should preserve the time information, be interpretable, and be fast to compute.

After reviewing symbolization techniques (that transform a real-valued series into a shorter discrete-valued series) and conducting a survey of distance measures on time series, strings, and symbolic sequences (that result from a symbolization process), we introduce novel symbolic representations and define a distance measure between the resulting symbolic sequences.

The first contribution is a symbolic representation for a data set of univariate time series called ASTRIDE. Unlike most symbolization procedures, ASTRIDE is adaptive (i.e. data-driven) during both the segmentation step by performing change-point detection and the quantization step by using quantiles. Instead of proceeding signal by signal, ASTRIDE builds a dictionary of symbols that is common to all signals in a data set. We also introduce a novel distance measure on symbolic representations that is based on the general edit distance with custom weights. We show the performance of ASTRIDE compared to 4 other symbolic representations on reconstruction and, when applicable, on classification tasks.

The second contribution is a symbolic representation for a data set of multivariate time series that can be non-stationary, called d_{symb}, along with an online exploration tool, called the d_{symb} playground. Unlike most distance measures on multivariate signals, d_{symb} takes into account their non-stationarity thanks to a symbolization step. This step is itself based on a change-point detection procedure that splits a non-stationary signal into several stationary segments, followed by quantization using K-means clustering. The proposed distance measure leverages the general edit distance. The advantages of d_{symb} are shown on 3 data sets of multivariate physiological signals. Experiments show how interpretable the symbolization is: a single glance at the symbolic sequences provides an immediate and comprehensive understanding of a data set. Moreover, compared to 9 multivariate elastic distances on a clustering task, d_{symb} achieves a competitive performance while being several orders of magnitude faster than the other methods. With these desirable characteristics, we developed the d_{symb} playground, an online tool, that allows researchers to apply d_{symb} to their uploaded data.

Introduction (en français)

Contents

Contexte, motivation et objectifs

L'objectif général de cette thèse est d'introduire de nouvelles représentations symboliques et mesures de distance pour les séries temporelles multivariées et non stationnaires.

Contexte

Cette thèse a été menée au Centre Borelli[1], un laboratoire de recherche académique de l'École Normale Supérieure Paris-Saclay, également affilié à l'Université Paris-Saclay, à l'Université Paris Cité, au CNRS, au SSA et à l'INSERM. La recherche au Centre Borelli s'articule autour des mathématiques appliquées, des neurosciences et de l'informatique, avec un accent particulier sur leurs interactions biomédicales et industrielles. Ainsi, une spécificité majeure du Centre Borelli est de faire collaborer étroitement des mathématiciens avec des ingénieurs, des médecins, des cliniciens et des experts de l'industrie.

[1] https://centreborelli.ens-paris-saclay.fr/fr

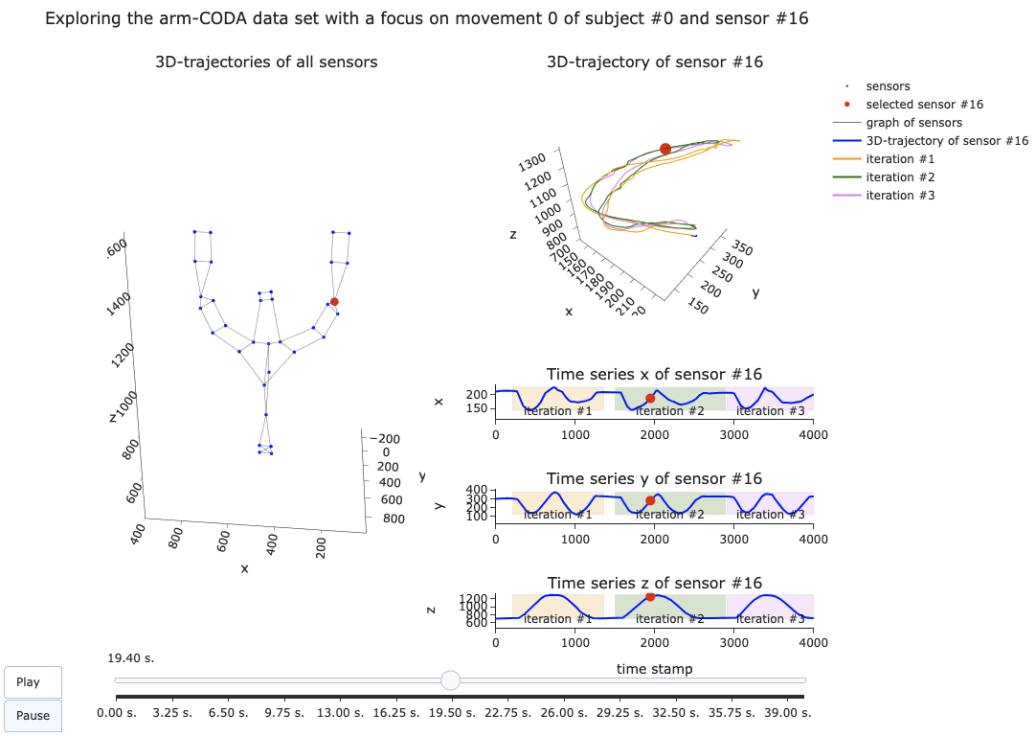


Figure 1: Exemple d'une série temporelle multivariée issue du jeu de données arm-CODA. À gauche de la Figure, le positionnement de plusieurs capteurs sur le membre supérieur d'un patient est affiché. À droite, un zoom sur la série temporelle multivariée, générée par l'un de ces capteurs, est fourni. Chaque mouvement (par exemple, l'élévation du bras) comprend trois itérations.

Pour ce qui est des neurosciences, le travail du Centre Borelli se concentre sur l'analyse du comportement humain et animal, avec deux objectifs principaux : le suivi longitudinal (étudier l'évolution d'un sujet au fil du temps) et la comparaison interindividuelle (comparer, souvent statistiquement, deux cohortes de sujets). Plusieurs projets sont actuellement en cours et visent à étudier la marche chez des sujets sains et pathologiques (par exemple, la sclérose en plaques) [VJ+19; Boi+22], le contrôle postural et la détection préventive du risque de chute [Bar+18], les mouvements des membres supérieurs pendant la rééducation après une blessure [Com+24a], les cycles respiratoires chez les souris [Ger+22], ou les états de conscience pendant l'anesthésie. Tous ces projets sont basés sur des capteurs pouvant être placés sur les sujets et permettant l'enregistrement de plusieurs signaux physiologiques (éventuellement synchronisés), tels que les électrocardiogrammes (ECG), les électroencéphalogrammes (EEG), ou les accélérations des pieds enregistrées avec des Inertial Measurement Units (IMUs). Le Centre Borelli a également participé à la construction de plusieurs protocoles cliniques et a généré des jeux de données en libre accès, tels qu'un jeu de données sur la marche humaine utilisant des IMUs [Tru+19] ou un jeu de données sur les mouvements des membres supérieurs enregistrés grâce à des capteurs de mouvement [Com+24a]. Un exemple de signaux physiologiques pouvant être enregistrés lors d'un protocole est présenté dans la Figure 1.

Du fait de la complexité des phénomènes que nous souhaitons observer, mobilisant parfois plusieurs fonctions physiologiques, les données collectées via différents

protocoles peuvent s'avérer difficiles à analyser. Tout d'abord, comme pour toute série temporelle, la première question est de savoir comment prendre en compte l'information temporelle dans les modèles. Intuitivement, dans une tâche de surveillance, la séquence et la chronologie des actions comportent des informations cruciales, qui doivent être préservées dans la chaîne de traitement. La deuxième question, qui est également très générale, concerne la nature bruitée des données issues de capteurs. En particulier, les études au niveau de la forme d'onde peuvent être rendues difficiles par le rapport signal/bruit parfois faible [KK03; Fu11; EA12]. De plus, d'autres questions découlent directement des protocoles utilisés :

1. **Nature multivariée** : L'étude d'un patient nécessite souvent des protocoles avec de nombreux capteurs afin d'obtenir une compréhension globale de l'état du patient. L'enregistrement de la position 3D d'une partie du corps au fil du temps génère une série temporelle multivariée de dimension 3. Cependant, lors de l'étude de l'élévation du bras, il est probable que plusieurs capteurs soient nécessaires (par exemple, sur les deux bras, aux poignets et aux coudes), ce qui donne une série temporelle de dimension bien plus grande (atteignant parfois des centaines de dimensions). Intuitivement, ces dimensions sont susceptibles d'être fortement corrélées, et il s'agit donc d'une information cruciale qui doit être prise en compte.

2. **Non-stationnarité** : Lorsqu'ils sont enregistrés sur de longues périodes de temps ou lors de protocoles complexes, les propriétés statistiques des signaux physiologiques évoluent souvent au cours du temps. Par exemple, si un sujet porte une montre connectée pendant toute une journée, en effectuant diverses activités intercalées avec des périodes de repos, le signal généré est généralement non stationnaire. La plupart des modèles statistiques couramment utilisés pour les séries temporelles nécessitent l'hypothèse de stationnarité au sens large et ne peuvent donc pas être utilisés dans ce contexte.

3. **Multimodalité** : Certains protocoles impliquent l'étude de différents capteurs qui enregistrent simultanément différents types de grandeurs, telles que des données d'accélérométrie, des ECG ou des EEG. Dans ce cas, le défi est beaucoup plus difficile car il nécessite l'étude conjointe de signaux physiologiques avec différentes propriétés physiques (fréquence d'échantillonnage, structure, etc).

Enfin, il y a la question de l'interprétabilité pour les cliniciens. Les travaux au Centre Borelli sont menés par des équipes pluridisciplinaires de mathématiciens et de cliniciens. Les outils d'analyse développés doivent donc leur permettre d'interagir avec les données, et la plupart des cliniciens ne sont pas formés à observer des formes d'ondes. Par conséquent, un défi de recherche fondamental est de créer des représentations mathématiques qui abstraient la complexité des données afin de les rendre sous une forme visuelle intuitive pour les cliniciens.

Questions scientifiques et positionnement

En ce qui concerne ces séries temporelles biomédicales complexes, cette thèse aborde les deux questions scientifiques suivantes :

1. Comment pouvons-nous représenter les signaux physiologiques avec une structure complexe ?
2. Comment pouvons-nous comparer ces séries temporelles ?

Il existe deux approches principales dans la littérature pour représenter et comparer des séries temporelles. La première consiste à extraire des caractéristiques à partir des séries temporelles brutes et à utiliser une représentation dite *bag of features*. Dans le contexte des signaux physiologiques, les caractéristiques couramment utilisées sont par exemple les coefficients de la transformée de Fourier discrète [AFS93; FRM94] ou la transformée en ondelettes discrète [CF99]. Ces approches extraient des caractéristiques des séries temporelles, souvent dimension par dimension, pour construire un vecteur de caractéristiques [BB21] qui est ensuite utilisé pour des tâches telles que la classification ou le partitionnement. Dans la plupart des cas, la comparaison entre les séries temporelles peut être effectuée en utilisant une simple distance euclidienne entre les vecteurs de caractéristiques. Une limitation majeure de ces méthodes est qu'elles perdent souvent l'information temporelle, étant donné qu'elles extraient des caractéristiques à l'échelle de la totalité de la série temporelle. De plus, si les séries temporelles sont non stationnaires, il est possible qu'une caractéristique définie sur toute la longueur du signal ne soit pas représentative.

La deuxième approche consiste à définir des distances applicables directement sur les formes d'onde. Parmi ces distances, il existe différentes techniques d'alignement temporel, à savoir les distances élastiques telles que la Dynamic Time Warping (DTW) [BC94; SY+17] ou les comparaisons de trajectoire [JCG20; Vay+22]. Ces méthodes travaillent directement sur la forme d'onde et projettent les séries temporelles dans des espaces géométriques qui peuvent être de grande dimension. Ces distances sont bien adaptées pour comparer de petits "snippets" de données, mais, par exemple, une comparaison brute des formes d'onde obtenues sur deux jours consécutifs est susceptible de produire des résultats non pertinents. Elles peuvent également être sensibles au bruit et avoir un coût de calcul élevé. De plus, ces distances sophistiquées impliquent des cadres mathématiques compliqués qui peuvent être difficiles à manipuler pour les cliniciens.

En se basant sur les idées développées précédemment, notre représentation devrait idéalement :

- Préserver l'information temporelle, c'est-à-dire la chronologie des événements ;
- Prendre en compte la nature multivariée et/ou multimodale des données ;
- Être interprétable et ergonomique : Les longues séries temporelles multivariées devraient être représentées de manière concise, où un simple coup d'œil à leur représentation fournirait toutes les informations essentielles qu'elle contient, par exemple en mettant l'accent sur les événements saillants ;
- Gérer la non-stationnarité des données : La caractérisation des phénomènes devrait être effectuée non pas au niveau de la totalité de la série temporelle, mais au niveau des *actions*, c'est-à-dire des phases stationnaires ;
- Être robuste au bruit.

De même, notre mesure de distance devrait idéalement :

- S'adapter aux phénomènes d'intérêt, c'est-à-dire aux types d'événements présents dans le jeu de données ;
- Effectuer la comparaison au niveau des *actions*, c'est-à-dire des phases stationnaires ;
- Être très rapide à calculer : Idéalement, le temps de calcul devrait être suffisamment faible pour qu'elle puisse être utilisée de manière interactive par des cliniciens ;
- Permettre des comparaisons interindividuelles et un suivi longitudinal.

Dans cette thèse, nous proposons de relever ces défis en nous appuyant sur une étape de représentation intermédiaire : la symbolisation des séries temporelles. Introduite au début des années 2000, la symbolisation vise à transformer des séries temporelles à valeurs réelles en séries plus courtes et à valeurs discrètes. L'une des représentations symboliques pionnière et très populaire est *Symbolic Aggregate approXimation (SAX)* [Lin+03; Lin+07]. Un exemple de représentation SAX pour une série temporelle univariée est illustré sur la Figure 4 en page 23. Grâce à l'effet de lissage induit par leur compression, les représentations symboliques sont largement utilisées dans les tâches de fouille de données, telles que la classification ou le partitionnement, où le choix de la représentation est fondamental. En particulier, une propriété souhaitable de ces techniques est qu'elles intègrent naturellement l'information temporelle et ont tendance à être robustes au bruit.

Dans les grandes lignes, la plupart des techniques de symbolisation suivent deux étapes : une étape de segmentation, où un signal à valeurs réelles est divisé en plusieurs segments, puis une étape de quantification, où chaque segment est attribué à une valeur discrète appelée un symbole. Par exemple, SAX utilise une segmentation uniforme puis quantifie les moyennes par segment en utilisant une hypothèse gaussienne. Ces séquences symboliques peuvent ensuite être comparées en utilisant des distances appropriées.

L'objectif de cette thèse est de créer une nouvelle représentation symbolique qui tient compte de tous les défis décrits précédemment (non-stationnarité, nature multivariée, interprétabilité, ...), mais aussi de construire une mesure de distance sur ces séquences symboliques qui soit rapide à calculer. Nos deux méthodes de symbolisation proposées sont ASTRIDE (décrite dans le chapitre IV) et d_{symb} (décrite dans le chapitre V). ASTRIDE transforme un jeu de séries temporelles univariées, tandis que d_{symb} transforme un jeu de séries temporelles multivariées. En plus de leur précision, les avantages clés sont l'interprétabilité et le faible temps de calcul.

Contributions et plan du manuscrit

Le manuscrit est organisé comme suit.

Chapitre II : Revue de la littérature sur les méthodes de symbolisation pour les séries temporelles

Dans le Chapitre II – Symbolic representation of time series, nous menons une revue exhaustive des méthodes de symbolisation qui ont été proposées dans la littérature. Nous examinons la première grande question scientifique de notre thèse d'un point

de vue symbolique : *Comment pouvons-nous représenter efficacement des séries temporelles avec une structure complexe ?*. Depuis l'introduction de SAX en 2003, il y a eu un intérêt prolifique pour la recherche autour des variantes de SAX et également d'autres catégories de symbolisation. Certaines revues ont été proposées il y a plus de 10 ans [DFT03; Lin+07; SW11]. Une revue plus récente [Wan+19] se concentre uniquement sur les variantes de SAX. Dans le Chapitre II, nous passons en revue plus de 60 méthodes de symbolisation.

Comme illustré dans la Figure 2, notre cadre est le suivant. Nous décomposons un processus de symbolisation en 3 étapes consécutives : la segmentation, l'extraction de caractéristiques et la quantification. En général, par rapport à SAX, les méthodes de symbolisation dans la littérature modifient une (ou plusieurs) étape(s) parmi les trois principales. Ce cadre n'est pas une grille stricte : certaines méthodes de symbolisation qui ne s'intègrent pas parfaitement dans ce cadre sont également décrites (par exemple, des méthodes qui n'utilisent pas exactement une étape de segmentation mais plutôt un sous-échantillonnage). Pour chaque étape, une revue détaillée est fournie avec le but de dégager des thèmes communs. Nous discutons également de la tâche de reconstruction qui consiste à reconstruire une série temporelle originale à partir de sa séquence symbolique. Enfin, nous discutons de la symbolisation de séries temporelles multivariées, un domaine de recherche plus récent. Les mesures de distance définies sur les séquences symboliques sont décrites dans le chapitre III.

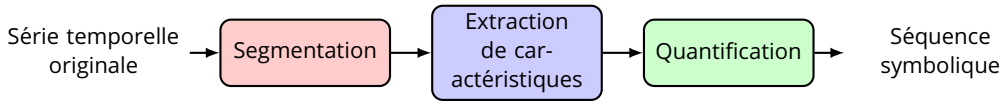


Figure 2: Les principales étapes de la symbolisation d'une série temporelle décrites dans le Chapitre II.

Chapitre III : Revue de la littérature sur les mesures de distance pour les séries temporelles, les chaînes de caractères et les séquences symboliques

Dans le Chapitre III – Distance measures on time series, strings, and symbolic sequences, nous passons en revue la deuxième grande question scientifique de cette thèse : *Comment pouvons-nous comparer efficacement les séries temporelles ?*. Nous passons en revue les mesures de distance sur les séries temporelles, les chaînes de caractères et les séquences symboliques trouvées dans la littérature. Les séquences symboliques sont des chaînes de caractères résultant de processus de symbolisation décrits dans le Chapitre II. Les mesures de distance sur les chaînes de caractères peuvent être appliquées aux séquences symboliques : la combinaison d'une méthode de symbolisation avec une mesure de distance sur les chaînes de caractères peut être considérée comme une mesure de distance sur les séries temporelles. Bien que des revues (y compris des récentes) sur les distances sur les séries temporelles [Wan+13; Shi+23; HMB23] et sur les chaînes de caractères [Kru83; Kuk92; WM92; Nav01] existent, à notre connaissance, elles n'abordent pas conjointement les séries temporelles et les chaînes de caractères. En effet, comme nous le verrons, il existe des points communs entre les mesures de distance sur les chaînes de caractères et celles sur les séries tem-

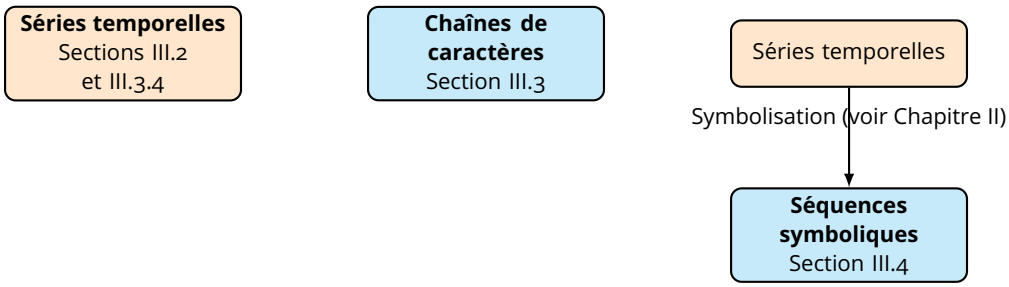


Figure 3: Aperçu des différents types de mesures de distance décrites dans le Chapitre III.

porelles. Dans ce chapitre, comme illustré dans la Figure 3, nous proposons une revue des séries temporelles et des chaînes de caractères, en mettant en évidence comment chaque domaine a inspiré l'autre. De plus, nous passons en revue les distances sur les séquences symboliques, obtenues après symbolisation, ce qui est novateur.

Pour les séries temporelles, nous passons en revue les distances d'alignement pas à pas ainsi que les distances dites élastiques. Lors de la comparaison de deux séries temporelles, les distances d'alignement pas à pas (telles que la distance euclidienne) ne peuvent comparer les échantillons qu'en utilisant un alignement "un à un", tandis que les distances élastiques utilisent un alignement "un à plusieurs", ce qui les rend plus robustes aux décalages temporels. Les distances élastiques incluent la Dynamic Time Warping (DTW) ainsi que ses variantes et versions contraintes. Pour les chaînes de caractères, nous décrivons les distances d'édition telles que la Longest Common SubSequence (LCSS). Nous examinons également l'extension des distances d'édition aux séries à valeurs réelles. Ensuite, nous décrivons les mesures de distance spécifiquement définies pour les séquences symboliques. Enfin, nous couvrons les extensions multivariées des distances sur les séries temporelles.

Chapter IV : Présentation d'ASTRIDE, une méthode de symbolisation adaptative pour un jeu de séries temporelles univariées

Dans le chapitre IV – ASTRIDE: Adaptive Symbolization for Time Series Databases, nous présentons une solution qui aborde simultanément les deux aspects scientifiques clés de cette thèse : la représentation et la distance, à travers la symbolisation efficace d'un jeu de séries temporelles univariées. Notre solution est une méthode de symbolisation appelée *ASTRIDE (Adaptive Symbolization for Time seRIes DatabasEs)* [CTO23b], qui est accompagnée d'une variante accélérée appelée *FASTRIDE (Fast ASTRIDE)* ainsi que d'une mesure de distance compatible appelée *D-GED (Dynamic General Edit Distance)*.

ASTRIDE et FASTRIDE sont de nouvelles représentations symboliques pour un jeu de séries temporelles univariées. Contrairement à la plupart des procédures de symbolisation, telles que la populaire SAX [Lin+03], ASTRIDE est adaptative (i.e. pilotée par les données) à la fois lors de l'étape de segmentation en effectuant une détection des points de rupture et lors de l'étape de quantification en utilisant des quantiles. Plus précisément, la segmentation détecte les changements de moyenne, où le nombre de ruptures est défini par l'utilisateur. La segmentation et la quantification adaptatives sont toutes deux apprises au niveau du jeu des signaux : les points de rupture, ainsi

que les quantiles (pour la quantification), sont estimés en utilisant tous les signaux du jeu de données. Ainsi, le dictionnaire de symboles d'ASTRIDE est le même pour tous les signaux, ce qui le rend efficace en mémoire. Une illustration comparant la représentation ASTRIDE avec SAX, sur un même signal univarié, est fournie dans la Figure 4. En plus de la symbolisation, nous introduisons également D-GED, une nouvelle mesure de distance sur les représentations symboliques basée sur la distance d'édition générale (décrite dans le Chapitre III). Définie sur des chaînes de caractères, la distance d'édition permet des substitutions, des suppressions et des insertions. À notre connaissance, ASTRIDE est la seule représentation symbolique offrant une discrétisation adaptative sur les dimensions temporelle et d'amplitude à l'échelle d'un jeu de données tout en ayant une mesure de distance compatible et une procédure de reconstruction efficace en mémoire.

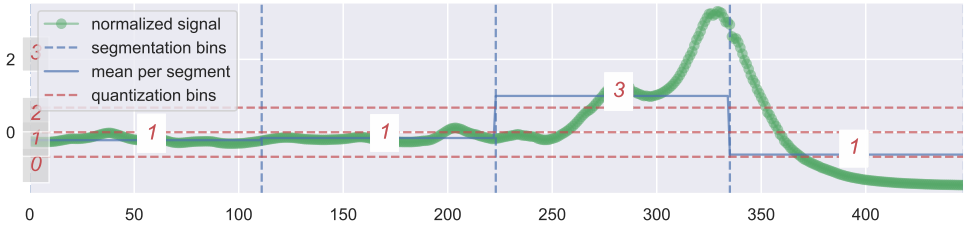
Afin d'évaluer la pertinence de nos solutions, nous les comparons avec des représentations symboliques populaires (décrites dans le Chapitre II) sur la tâche de reconstruction et, lorsque c'est applicable, en classification. Les algorithmes étudiés sont évalués sur 86 jeux de signaux univariés de taille égale provenant de la UCR Time Series Classification Archive [Dau+19] qui est largement utilisée. Cette archive est composée de séries temporelles issues du monde réel (audio, mouvement, etc), et aussi des séries simulées. Les performances des représentations ASTRIDE et FASTRIDE sont comparées à celles de SAX, 1d-SAX [Mal+13], *SFA (Symbolic Fourier Approximation)* [SH12], et *ABBA (Adaptive Brownian Bridge-based Aggregation)* [EG20a]. Pour la classification, notre comparaison est limitée aux méthodes directement basées sur des symbolisations, car notre objectif est d'évaluer la pertinence de cette étape en elle-même et non pas d'atteindre des performances d'état de l'art en classification de séries temporelles. Par conséquent, nous excluons les classifieurs construits sur des représentations symboliques, à savoir les algorithmes dits "bag-of-words" et les méthodes ensemblistes [SM13; Sch15; SL17; Ngu+19; Mid+20]. Les résultats montrent qu'ASTRIDE fournit une représentation symbolique intuitive qui surpasse l'état de l'art en termes de taux de classification par plus proche voisin et obtient des résultats compétitifs en reconstruction de signal. Un dépôt GitHub en libre accès[2] est disponible pour reproduire toutes les expériences en Python.

Chapitre V : Présentation de d_symb, une mesure de distance, basée sur la symbolisation, interprétable et rapide pour séries temporelles multivariées

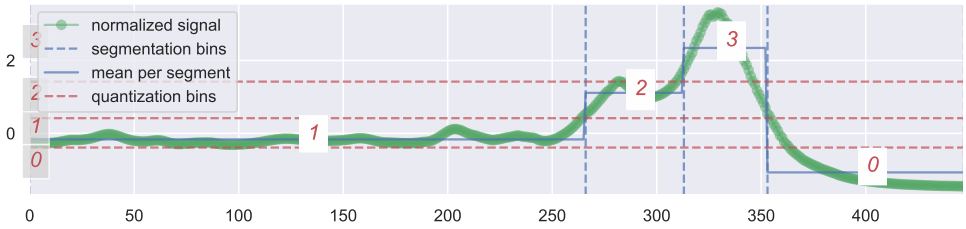
Dans le Chapitre V – d_symb: an interpretable distance measure for multivariate signals, nous présentons d_{symb} [CTO23a], une méthode qui traite les séries temporelles multivariées du point de vue de la représentation et aussi de celui de la distance, tout en étant interprétable, précise et rapide à calculer. De plus, d_{symb} est mis en œuvre dans un outil interactif en ligne appelé le d_{symb} playground [Com+24b]. Cet outil est destiné à être utilisé par des cliniciens ou des experts du domaine pour interpréter et comparer rapidement leurs volumineux jeux de données de séries temporelles multivariées non stationnaires.

d_{symb} est une nouvelle mesure de distance permettant de comparer des signaux multivariés non stationnaires. Contrairement à la plupart des mesures de distance sur les signaux multivariés telles que les variantes de la Dynamic Time Warping

[2] https://github.com/sylvaincom/astride



(a) Répresentation SAX



(b) Représentation ASTRIDE

Figure 4: Exemple de représentations SAX et ASTRIDE pour le même signal univarié et les mêmes paramètres d'entrée. La séquence symbolique résultante est 1131 pour SAX et 1230 pour ASTRIDE.

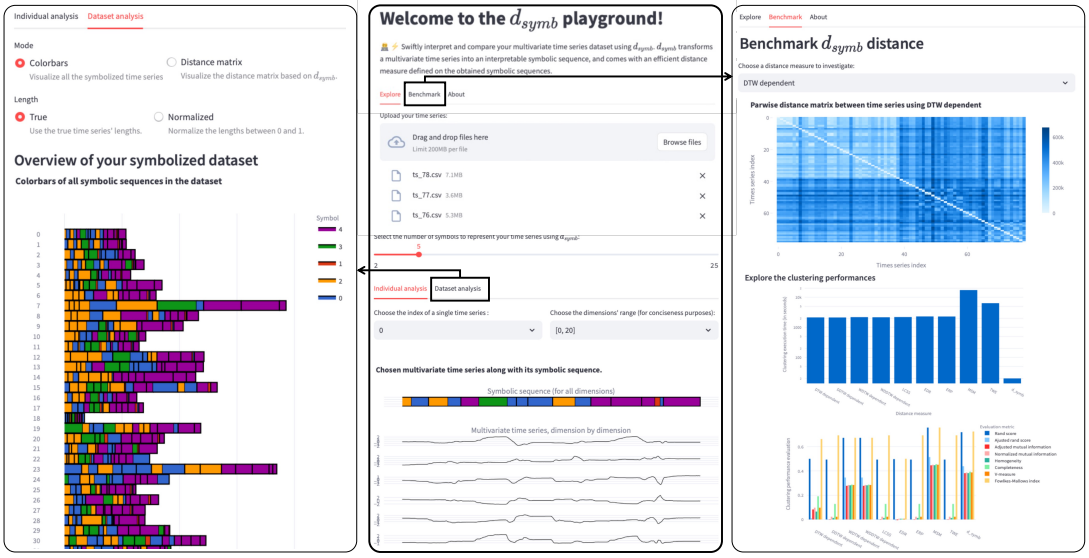
(DTW) [BC94; SY+17], d_{symb} peut prendre en compte la non-stationnarité des signaux grâce à une étape de segmentation adaptative. Cette étape repose sur une procédure de détection de ruptures qui divise un signal non stationnaire en plusieurs segments stationnaires. d_{symb} suit les mêmes étapes générales qu'ASTRIDE (introduite dans le Chapitre IV), mais avec les modifications suivantes : la segmentation de d_{symb} est appliquée à chaque signal multivarié séparément, le nombre de segments est trouvé automatiquement par une formulation pénalisée de la détection des ruptures, et l'étape de quantification utilise un partitionnement par les K-moyennes au lieu des quantiles. Enfin, la mesure de distance d_{symb} exploite la distance d'édition générale et est appliquée aux séquences symboliques.

Les avantages de d_{symb} sont démontrés sur trois jeux de signaux physiologiques : le jeu de données JIGSAWS [Gao+14], qui enregistre des chirurgiens utilisant des bras et des pinces robotisés, le jeu de données sur la marche humaine [Tru+19], et le jeu de données armCODA [Com+24a], qui enregistre les mouvements des membres supérieurs humains. Les expériences montrent à quel point la symbolisation est interprétable, comme illustré sur les données de la marche dans la Figure 5. En effet, la symbolisation détecte les segments qui correspondent aux comportements saillants, et chaque symbole correspond à un régime spécifique de la marche humaine, tel que faire demi-tour ou marcher tout droit. D'un simple coup d'œil sur les frises de couleur, la symbolisation fournit une compréhension immédiate et complète d'un jeu de séries temporelles multivariées. De plus, comparé à neuf distances élastiques multivariées sur une tâche de partitionnement, d_{symb} obtient des performances compétitives tout en étant plusieurs ordres de grandeur plus rapide que les autres méthodes. Un dépôt GitHub en libre accès[3], codé en Python, est disponible.

[3] https://github.com/sylvaincom/d-symb

Figure 5: Séquences symboliques (représentées sous forme de frises de couleur) obtenues en utilisant la symbolisation d_{symb} pour 60 séries temporelles multivariées provenant du jeu de données de la marche [Tru+19] contenant 3 classes. Les classes sont séparées par des lignes horizontales blanches en pointillés. Chaque ligne représente la frise de couleur correspondant à une séquence symbolique.

Le d_{symb} playground[4][5], disponible en ligne, vise à explorer, interpréter et comparer rapidement plusieurs séries temporelles multivariées à partir d'un jeu de données. Cet outil, présenté dans la Figure 6, permet aux utilisateurs de téléverser et de visualiser leurs séries temporelles multivariées ainsi que leurs symbolisations d_{symb} à l'aide des frises de couleur. L'interprétabilité et l'interactivité du d_{symb} playground découlent de la pertinence des symboles et du faible temps de calcul de d_{symb}.



(A) Dataset analysis frame (B) Individual analysis frame (C) Benchmark frame

Figure 6: Illustration des trois principales interfaces du d_{symb} playground.

Liste des papiers

Papiers acceptés :

- S. W. Combettes, C. Truong, and L. Oudre. "SAX-DD : une nouvelle représentation symbolique pour séries temporelles." In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, Nancy, France, 2022.
- S. W. Combettes, C. Truong, and L. Oudre. "An Interpretable Distance Measure for Multivariate Non-Stationary Physiological Signals." In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, Shanghai, China, 2023.
- S. W. Combettes, P. Boniol, A. Mazarguil, D. Wang, D. Vaquero-Ramos, M. Chauveau, L. Oudre, N. Vayatis, P.-P. Vidal, A. Roren, and M.-M. Lefèvre-Colau. "Arm-CODA: A Data Set of Upper-limb Human Movement During Routine Examination." *Image Processing On Line*, 14:1-13, 2024.
- S. W. Combettes, P. Boniol, C. Truong, and L. Oudre. "d_{symb} playground: an interactive tool to explore large multivariate time series datasets." In *Proceedings of the International Conference on Data Engineering (ICDE)*, Utrecht, Netherlands, 2024.

[4] https://dsymb-playground.streamlit.app
[5] https://github.com/boniolp/dsymb-playground

Prépublication :

- S. W. Combettes, C. Truong, and L. Oudre. ”ASTRIDE: Adaptive Symbolization for Time Series Databases.” arXiv preprint arXiv:2302.04097, 2023.

Chapter I

Introduction (in English)

Contents

I.1 Context, motivation, and objectives

The general objective of this thesis is to introduce new symbolic representations and distance measures for multivariate non-stationary time series.

I.1.1 Context

This thesis has been conducted at Centre Borelli[1], an academic research laboratory of Ecole Normale Supérieure Paris-Saclay, also affiliated with Université Paris-Saclay, Université Paris Cité, CNRS, SSA, and INSERM. Research at Centre Borelli revolves around applied mathematics, neuroscience, and computing, with a special focus on their biomedical and industrial interactions. Hence, a major specificity of the Centre

[1] https://centreborelli.ens-paris-saclay.fr/en

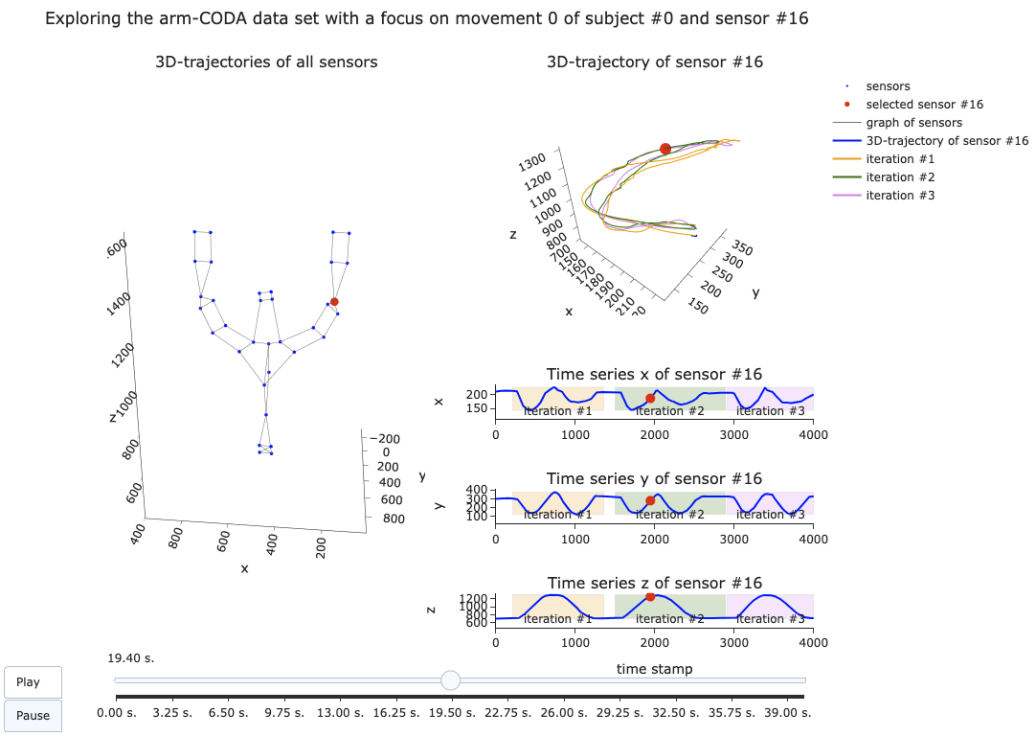


Figure I.1: Example of a multivariate time series from the armCODA data set. On the left of the Figure, the placement of multiple sensors on the upper limb of a patient is displayed. On the right, a focus on the multivariate time series generated by one of these sensors is provided. Each movement (for example, arm elevation) comprises three iterations.

Borelli is making mathematicians work closely with engineers, medical doctors, clinicians, and industry experts.

In terms of neuroscience, Centre Borelli's work focuses on the analysis of human and animal behavior, with two main aims: longitudinal follow-up (studying the evolution of a subject over time) and inter-individual comparison (comparing, often statistically, two cohorts of subjects). Several projects are currently underway to study gait in healthy and pathological subjects (e.g. multiple sclerosis) [VJ+19; Boi+22], postural control and early detection of fall risk [Bar+18], upper-limb movements during rehabilitation after injury [Com+24a], respiratory cycles in mice [Ger+22], or states of consciousness during anesthesia. All these projects are based on sensors that can be worn by the subjects and enable the recording of several physiological signals (possibly synchronized), such as electrocardiograms (ECGs), electroencephalograms (EEGs), or foot accelerations recorded with Inertial Measurement Units (IMUs). The Centre Borelli has also participated in the construction of several clinical protocols and has generated open-access data sets such as a human locomotion data set using IMUs [Tru+19] or an upper-limb human movement data set using motion capture [Com+24a]. An example of physiological signals that can be recorded during a protocol is shown in Figure I.1.

Due to the complexity of the phenomena we wish to observe, which sometimes mobilize several physiological functions, the data collected in the various protocols can be difficult to analyze. First of all, as with any time series, the first question is how

to consider the temporal information in the models. Intuitively, in a monitoring task, the sequence and chronology of actions carries crucial information, which must be preserved in the processing chain. The second question, which is also very general, relates to the noisy nature of sensor data. In particular, studies at the waveform level can be made difficult by the sometimes low signal-to-noise ratio [KK03; Fu11; EA12]. On the other hand, other questions arise directly from the protocols used:

1. **Multivariate nature**: Studying a patient often requires protocols with many sensors to get a complete understanding of the subject's condition. Recording the 3D position of a body segment over time results in a multivariate time series of dimension 3. However, when studying arm elevation, it is likely that multiple sensors will be needed (e.g., on both arms, at the wrists, and at the elbows), resulting in a time series of higher dimensions (possibly hundreds). Intuitively, these dimensions are likely to be highly correlated, and this constitutes crucial information that needs to be taken into account.

2. **Non-stationarity**: When recorded over long periods of time or during complex protocols, the statistical properties of physiological signals often change over time. For example, if a subject wears a connected watch for an entire day, performing various activities with periods of rest in between, the generated signal is typically non-stationary. Most popular statistical models for time series require the wide-sense stationary property and therefore cannot be used in this context.

3. **Multimodality**: Some protocols involve the study of different sensors that simultaneously record different types of quantities, such as accelerometry data, ECG, or EEG. In this case, the challenge is much more difficult because it requires the joint study of physiological signals with different physical properties (sampling frequency, structure, etc).

Last but not least, there is the question of interpretability for clinicians. Work at the Centre Borelli is carried out by multidisciplinary teams of mathematicians and clinicians. The developed analysis tools must, therefore, enable them to interact with the data, and most clinicians are not trained to observe waveforms. Therefore, a fundamental research challenge is to be able to create mathematical representations that abstract from the complexity of the data in order to render it in an intuitive visual form for clinicians.

I.1.2 Scientific questions and positioning

Regarding these challenging biomedical time series, this thesis addresses the two following scientific questions:

1. How can we represent physiological signals with a complex structure?
2. How can we compare these time series?

There are two main approaches in the literature for representing and comparing time series. The first is to extract features from the raw time series and use a *bag of features* representation. In the context of physiological signals, popular features

may include *Discrete Fourier Transform (DFT)* [AFS93; FRM94] or *Discrete Wavelet Transform (DWT)* [CF99] coefficients. These approaches extract features from the time series, often dimension by dimension, to build a vector of features [BB21] to be used for classification, clustering, and more. In most cases, the comparison between time series can be done using a simple Euclidean distance on the feature vectors. A major limitation of these methods is that they often lose the temporal information as they extract features at the scale of an entire time series. Furthermore, if the time series are non-stationary, it is likely that a feature defined over the entire length would not be representative.

The second approach is to define distances that can be applied directly to the waveforms. Among these distances, there is a variety of temporal alignment techniques with elastic distances such as Dynamic Time Warping (DTW) [BC94; SY+17] or trajectory comparisons [JCG20; Vay+22]. These methods work directly on the waveform and project the time series into geometric spaces, which can be high dimensional. These distances are well suited for comparing small snippets of data but, for example, a crude comparison of waveforms obtained over two consecutive days is likely to produce irrelevant results. They can also be sensitive to noise and have a high computational cost. In addition, sophisticated distances involve complicated mathematical frameworks that can be difficult for clinicians to use.

Based on the ideas developed previously, our ideal representation should:

- Preserve the time information, i.e. the chronology of the events;
- Handle the multivariate and/or multimodal nature of the data;
- Be interpretable and ergonomic: Long multivariate time series should be represented in a concise way, where a simple glance at the representation should provide all the essential information contained in it, for example, emphasizing on the salient events;
- Handle the non-stationarity of the data: The characterization of the phenomena should be done not at the level of the whole time series, but at the level of *actions*, i.e. stationary phases;
- Be robust to noise.

Similarly, our ideal distance measure should:

- Adapt to the phenomena of interest, i.e. to the types of events present in the data set;
- Perform the comparison at the level of *actions*, i.e. stationary phases;
- Be very fast to compute: Ideally, the complexity should be low enough so that it can be used interactively by clinicians;
- Allow us to perform both inter-individual comparisons or longitudinal follow-up.

In this thesis, we propose to address these challenges by relying on an intermediate representation step: the symbolization of time series. Introduced in the early 2000s, symbolization aims at transforming real-valued time series into shorter discrete-valued sequences. One of the pioneering and highly popular symbolic representations is *Symbolic Aggregate approXimation (SAX)* [Lin+03; Lin+07]. An example of SAX representation for a univariate time series is shown in Figure I.4 on page 34.

Thanks to the smoothing effect induced by their compression, symbolic representations are widely used in data mining tasks, such as classification or clustering, where the choice of the representation is fundamental. In particular, a desirable property of these techniques is that they naturally incorporate the time information and tend to be robust to noise.

In a nutshell, most symbolization techniques follow two steps: a segmentation step, where a real-valued signal is divided into several segments, then a quantization step, where each segment is mapped to a discrete value called a symbol. For example, SAX uses a uniform segmentation then quantizes the means per segment by using a Gaussian assumption. These symbolic sequences can then be compared using well-designed distances.

The goal of this thesis is to create a novel symbolic representation that addresses all the challenges described above (non-stationarity, multivariate nature, interpretability, ...) but also to build a distance measure on these symbolic sequences that is fast to compute. Our two proposed symbolization methods are ASTRIDE (described in Chapter IV) and d_{symb} (described in Chapter V). ASTRIDE transforms a data set of univariate time series, while d_{symb} transforms a data set of multivariate time series. Apart from their accuracy, key advantages are their interpretability and computational efficiency.

I.2 Contributions and outline

The manuscript is organized as follows.

I.2.1 Chapter II: Literature review of symbolization methods for time series

In Chapter II – Symbolic representation of time series, we conduct a comprehensive overview of symbolization methods that have been proposed in the literature. We review the first main scientific question of our thesis from a symbolic perspective: *How can we efficiently represent time series with a complex structure?* Since the introduction of SAX in 2003, there has been a prolific interest in research around SAX-like methods and other categories of symbolization methods. Some reviews have been proposed more than 10 years ago [DFT03; Lin+07; SW11]. A more recent one [Wan+19] focuses on SAX-like variants only. In Chapter II, we review more than 60 symbolization methods.

As illustrated in Figure I.2, our framework is the following: we break down a symbolization process into 3 consecutive steps: segmentation, feature extraction, and quantization. Typically, compared to SAX, symbolization methods in the literature modify one (or more) step(s) among the three main ones. This framework is not a strict grid: some symbolization methods that do not fit perfectly into this framework are also described (for example, methods that do not employ a segmentation step *per se* but rather down-sampling). For each step, a detailed overview is provided with the aim of identifying common themes. We also discuss the reconstruction task: reconstructing an original time series from its symbolic sequence. Finally, we discuss symbolization for multivariate time series, which is a more recent research area. Distance measures defined on the resulting symbolic sequences are described in Chapter III.

Figure I.2: Main steps for the symbolization of a time series described in Chapter II.

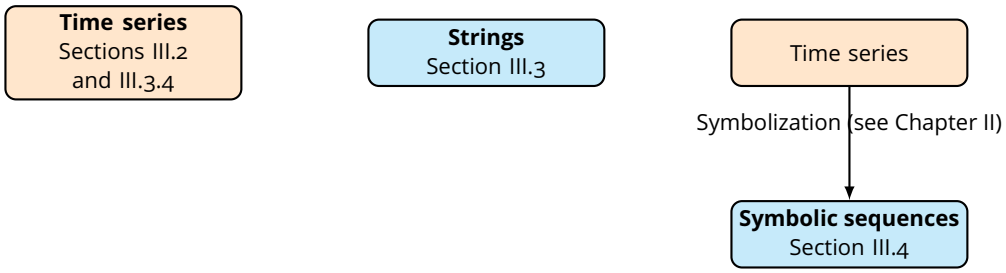


Figure I.3: Overview of types of distance measures described in Chapter III.

I.2.2 Chapter III: Literature review of distance measures on time series, strings, and symbolic sequences

In Chapter III – Distance measures on time series, strings, and symbolic sequences, we review the second main scientific question of this thesis: *How can we efficiently compare time series?* We survey distance measures on time series, strings, and symbolic sequences found in the literature. Symbolic sequences are strings resulting from symbolization processes described in Chapter II. Distance measures on strings could be applied to symbolic sequences: the combination of a symbolization method with a distance measure on strings can be considered as a distance measure on time series. While reviews (including recent ones) on distances on time series [Wan+13; Shi+23; HMB23] and strings [Kru83; Kuk92; WM92; Nav01] exist, to the best of our knowledge, they do not tackle time series and strings jointly. Indeed, as we shall see, there are common grounds for distances on strings and time series. In this chapter, as illustrated in Figure I.3, we propose a review of time series and strings while highlighting how one has inspired the other. Moreover, we survey distances on symbolic sequences obtained after symbolization, which has not been done before.

For time series, we review lock-step alignment distances as well as elastic ones. When comparing two time series, lock-step alignment distances (such as the Euclidean distance) can only compare samples using one-to-one alignment, while elastic distances use one-to-many alignment, thus are more robust to time-shifts. Elastic distances include Dynamic Time Warping (DTW) along with its variants and constrained versions. For strings, we describe edit distances such as the Longest Common SubSequence (LCSS). We also look into the extension of edit distances to real-valued series. Then, we describe distance measures specifically defined for symbolic sequences. Finally, we cover the multivariate extensions of distances on time series.

I.2.3 Chapter IV: Presentation of ASTRIDE, an adaptive symbolization method for a data set of univariate time series

In Chapter IV – ASTRIDE: Adaptive Symbolization for Time Series Databases, we introduce a solution that simultaneously addresses the two key scientific aspects of this

thesis: representation and distance, with a focus on efficiently symbolizing a dataset of univariate time series. Our solution is a symbolization method called *ASTRIDE (Adaptive Symbolization for Time seRIes DatabasEs)* [CTO23b] that comes with an accelerated variant named *FASTRIDE (Fast ASTRIDE)* as well as a compatible distance measure called *D-GED (Dynamic General Edit Distance)*.

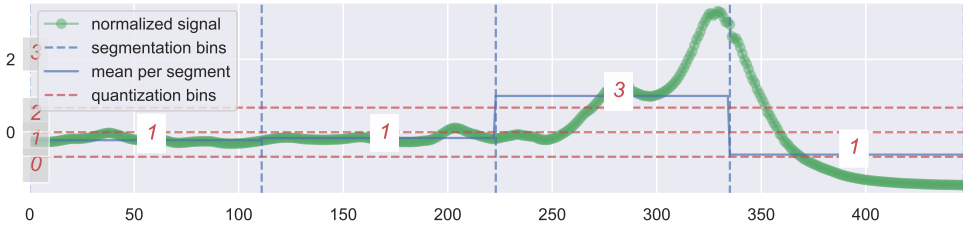
ASTRIDE and FASTRIDE are novel symbolic representations for a data set of univariate time series. Unlike most symbolization procedures, such as the popular SAX [Lin+03], ASTRIDE is adaptive (i.e. data-driven) during both the segmentation step by performing change-point detection and the quantization step by using quantiles. More precisely, the segmentation detects mean-shifts, where the number of changes is set by the user. Both adaptive segmentation and quantization are learned at the level of the data set of signals: the change-points, as well as the quantiles (for the quantization), are estimated using all signals in the data set. Hence, ASTRIDE's dictionary of symbols is the same for all signals, and is thus memory-efficient. An illustration comparing the ASTRIDE representation with SAX, on a single univariate signal, is provided in Figure I.4. Along with the symbolization, we also introduce D-GED, a novel distance measure on symbolic representations based on the general edit distance (reviewed in Chapter III). Defined on strings, the edit distance allows substitutions, deletions, and insertions. To the best of our knowledge, ASTRIDE is the only symbolic representation offering adaptive discretization on both the time and amplitude dimension at the scale of a data set while having a compatible distance measure and a reconstruction procedure that is memory-efficient.

In order to assess the relevance of our solutions, we benchmark them with popular symbolic representations (described in Chapter II) on reconstruction and, when applicable, on classification tasks. The studied algorithms are evaluated on 86 univariate equal-size data sets from the widely-used UCR Time Series Classification Archive [Dau+19], which is composed of real-world time series from several domains such as audio and motion and simulated series. The performance of the ASTRIDE and FASTRIDE representations is compared to SAX, 1d-SAX [Mal+13], *SFA (Symbolic Fourier Approximation)* [SH12], and *ABBA (Adaptive Brownian Bridge-based Aggregation)* [EG20a]. For classification, our comparison is limited to techniques directly based on symbolizations since our objective is to evaluate the relevance of this step itself and not to achieve state-of-the-art performance on time series classification. Hence, we exclude classifiers that are built on top of symbolic representations, namely bag-of-words and ensemble-based algorithms [SM13; Sch15; SL17; Ngu+19; Mid+20]. Results show that ASTRIDE provides an intuitive symbolic representation that outperforms the symbolization state of the art in nearest-neighbor classification accuracy and achieves competitive results in signal reconstruction. An open source GitHub repository[2] is available to reproduce all the experiments in Python.

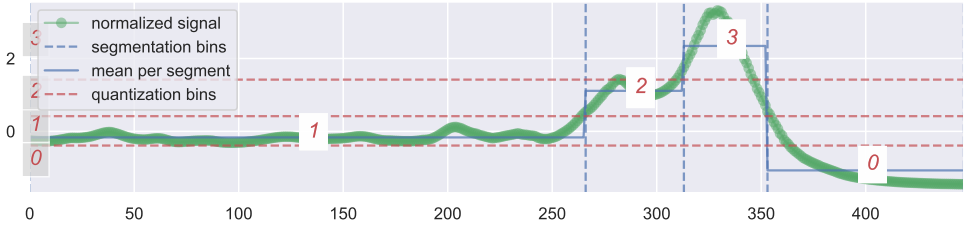
I.2.4 Chapter V: Presentation of d_symb, an interpretable and fast distance measure for multivariate time series based on symbolization

In Chapter V – d_symb: an interpretable distance measure for multivariate signals, we introduce d_{symb} [CTO23a] a solution that addresses multivariate time series for both the representation and the distance aspects while being interpretable, accurate, and fast to compute. Moreover, d_{symb} is showcased in an interactive online tool called

[2] https://github.com/sylvaincom/astride



(a) SAX representation



(b) ASTRIDE representation

Figure I.4: Example of SAX and ASTRIDE representations for the same univariate signal and the same input parameters. The resulting symbolic sequence is 1131 for SAX and 1230 for ASTRIDE.

the d_{symb} playground [Com+24b], to be used by clinicians or field experts to quickly interpret and compare their typically large data sets of non-stationary multivariate time series.

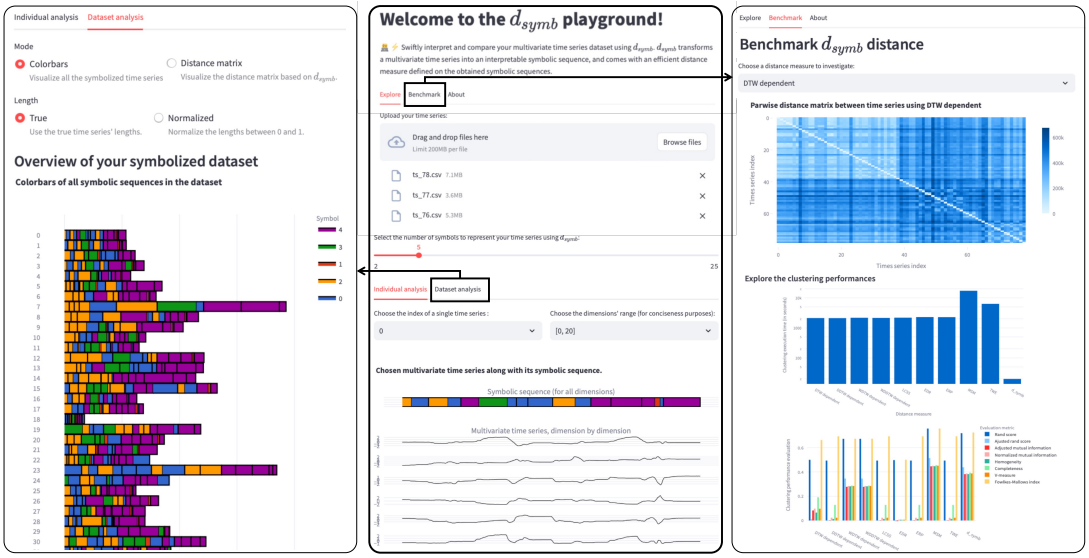
d_{symb} is a novel distance measure for comparing multivariate non-stationary signals. Unlike most distance measures on multivariate signals such as variants of Dynamic Time Warping (DTW) [BC94; SY+17], d_{symb} can take into account their non-stationarity thanks to an adaptive segmentation step. This step is based on a change-point detection procedure that splits a non-stationary signal into several stationary segments. d_{symb} follows the same overall steps as ASTRIDE (introduced in Chapter IV), but the d_{symb} segmentation is applied to each multivariate signal separately, the number of segments is found automatically by a penalized formulation of the change-point detection procedure, and the quantization step uses K-means clustering instead of quantiles. Finally, the d_{symb} distance measure leverages the general edit distance and is applied to the symbolic sequences.

The advantages of d_{symb} are shown on three data sets of physiological signals: the JIGSAWS data set [Gao+14] which monitors surgeons using robotic arms and grippers, the human locomotion data set [Tru+19], and the armCODA data set [Com+24a] which records human upper-limb movement. Experiments show how interpretable the symbolization is, as illustrated on gait data in Figure I.5. Indeed, the symbolization detects the segments that correspond to salient behaviors, and each symbol corresponds to a specific regime of human locomotion, such as turning around or walking in a straight line. With a single glance at the color bars, the symbolization provides an immediate and comprehensive understanding of a data set. Moreover, compared to nine multivariate elastic distances on a clustering task, d_{symb} achieves a competitive performance while being several orders of magnitude faster than the other methods.

Figure I.5: Symbolic sequences (represented as color bars) obtained using the d_{symb} symbolization for 60 multivariate time series from the gait data set [Tru+19] containing 3 classes. Classes are separated by white dashed horizontal lines. Each row is the color bar corresponding to a symbolic sequence.

An open source GitHub repository[3], written in Python, is available.

The d_{symb} playground[4][5], available online, aims at quickly exploring, interpreting, and comparing multiple multivariate time series from a data set. This tool, displayed in Figure I.6, allows users to upload and visualize their multivariate time series and their d_{symb} symbolizations using the color bars. The interpretability and interactivity of the d_{symb} playground stem from the symbols' relevance and the computational efficiency of d_{symb}.



(A) Dataset analysis frame (B) Individual analysis frame (C) Benchmark frame

Figure I.6: Illustration of the three main frames of the d_{symb} playground.

I.3 List of papers

Accepted papers:

- S. W. Combettes, C. Truong, and L. Oudre. "SAX-DD : une nouvelle représentation symbolique pour séries temporelles." In *Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI)*, Nancy, France, 2022.
- S. W. Combettes, C. Truong, and L. Oudre. "An Interpretable Distance Measure for Multivariate Non-Stationary Physiological Signals." In *Proceedings of the International Conference on Data Mining Workshops (ICDMW)*, Shanghai, China, 2023.
- S. W. Combettes, P. Boniol, A. Mazarguil, D. Wang, D. Vaquero-Ramos, M. Chauveau, L. Oudre, N. Vayatis, P.-P. Vidal, A. Roren, and M.-M. Lefèvre-Colau. "ArmCODA: A Data Set of Upper-limb Human Movement During Routine Examination." *Image Processing On Line*, 14:1-13, 2024.
- S. W. Combettes, P. Boniol, C. Truong, and L. Oudre. "d_{symb} playground: an interactive tool to explore large multivariate time series datasets." In *Proceedings*

[3] https://github.com/sylvaincom/d-symb
[4] https://dsymb-playground.streamlit.app
[5] https://github.com/boniolp/dsymb-playground

of the International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 2024.

Preprint:

- S. W. Combettes, C. Truong, and L. Oudre. ”ASTRIDE: Adaptive Symbolization for Time Series Databases.” arXiv preprint arXiv:2302.04097, 2023.

Chapter II

Symbolic representation of time series

This chapter is an overview of symbolization methods that have been proposed in the literature. A symbolization process transforms real-valued time series into shorter discrete-valued sequences. We break down a symbolization process into 3 consecutive steps: segmentation, feature extraction, and quantization. We also discuss the reconstruction task, that is, reconstructing an original time series from its symbolic sequence. Finally, we discuss the symbolization of multivariate time series, which is a more recent research area.

Contents

II.1 Introduction

Over the past decades, the increasing amount of available time series data has led to a rising interest in time series data mining. To cope with the complexity of such data, researchers have designed adapted representations that encapsulate signals' characteristics and that are easier to manipulate, e.g., shorter, interpretable, structured, etc. Among many time series representations [Rat+10; Fu11; EA12; Wan+13; BR14], symbolic representations constitute a tool of choice [Lin+07]. Symbolic representations of time series are used for data mining tasks such as time series visualization [LKL05; Lin+07; Fu11; Rut+19], classification [Lin+07; Esm+12; LKL12; SM13; Sch15; NGI17; Ngu+19; FCG22; LLP23], clustering [Lin+07; BTT21a], indexing [Lin+07; SK08; Cam+10; Cam+14; Yag+17], anomaly detection [KLF05; Lon+06; WKX06; Yan+07; RK13; EG20a; KR20; BTT21a], rule discovery [PJ20], motif discovery [Sen+18], and forecasting [EG20b]. The domain applications include finance [LSK06; BABO12], healthcare [SW10; SP+17], and industry [Esm+12; PJ20; KR20] to name a few.

Most symbolization techniques follow three steps: a segmentation step where a real-valued signal $y = (y_1, \dots, y_n)$ of length n is split into w segments, a feature extraction step where features of interest are extracted for each segment, then a quantization step where each segment (through its extracted features) is mapped to a discrete value $\hat{y}_i$ taken from a set $\{\mathtt{a}_1, \dots, \mathtt{a}_A\}$ of A symbols. The resulting symbolic representation is the discrete-valued signal (or symbolic sequence) $\hat{y} = (\hat{y}_1, \dots, \hat{y}_w)$. The set of symbols $\{\mathtt{a}_1, \dots, \mathtt{a}_A\}$ is usually called an *alphabet* or *dictionary*, and A is the *alphabet size*. The length w of the symbolic representation is called the *word length*. While there exist many high-level representations for time series, the two main advantages of symbolic representations are a reduced memory usage and competitive performances on data mining tasks thanks to the smoothing effect induced by compression [Lin+07].

First of all, let us explain in detail the principle of symbolization through a widely-used symbolization technique called *Symbolic Aggregate ApproXimation (SAX)* [Lin+03; Lin+07]. Introduced in 2003, SAX paved the way for many other symbolic representations, which are often variants or extensions of SAX. In SAX, as in most symbolic representations, the symbolization process has two parameters: the word length w and the alphabet size A. For instance, in the symbolic sequence `abbcaabc`, the parameters are $w = 8$ (length of the sequence) and $A = 3$ (number of possible symbols). Each signal is centered and scaled to unit variance, then split into w segments of equal length. Next, the means of all segments are clustered in bins and each segment is represented by the bin where its mean falls into. The bin boundaries are chosen so that all symbols are equiprobable under the assumption that the means follow a standard Gaussian distribution. A SAX transformation of a signal taken from the UCR Time Series Classification Archive [Dau+19] is shown in Figure II.1. The larger w and A, the better the quality of the SAX representation, but the lower the compression. Optimal values of w and A are highly dependent on the application and the data set. In the experiments for classification in the SAX paper [Lin+07], $w \in [\![2, n/2]\!]$ (where n is the length of the original time series) and $A \in [\![3, 10]\!]$. SAX has been applied to many data mining tasks, such as clustering, classification, query by content, anomaly detection, motif discovery, and visualization.

Since the introduction of SAX in 2003, there has been a prolific interest in research around SAX-like methods and other categories of symbolization methods. Some reviews have been proposed [Liu+02; DFT03; Lin+07; SW11] more than a decade ago. A

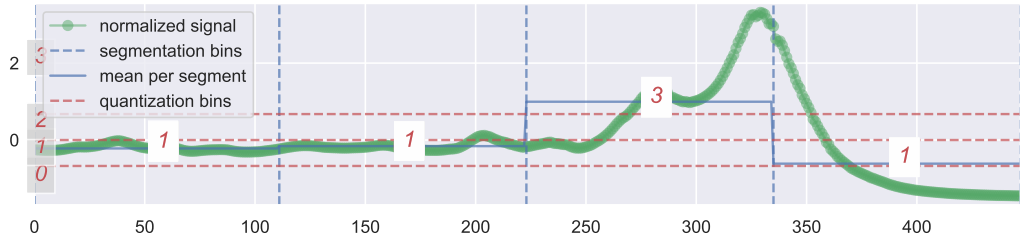


Figure II.1: Example of a SAX representation of a signal from the Meat data set (UCR Time Series Classification Archive). The original length of the signal is $n = 448$, and we use $w = 4$ and $A = 4$. The resulting symbolic sequence is 1131.

Figure II.2: Main steps for symbolization of a time series.

more recent one [Wan+19] focuses on SAX-variants only. In this chapter, we describe more than 60 symbolization methods, extending previous reviews.

Outline We conduct a comprehensive overview of symbolization methods that have been proposed in the literature. We break down a symbolization process into 3 consecutive steps illustrated in Figure II.2: segmentation, feature extraction, and quantization. Typically, compared to the popular SAX, symbolization methods in the literature modify one (or more) step(s) among the three main ones. A few symbolization methods that do not fit perfectly into this framework are also described (for example, methods that do not employ a segmentation step *per se* but rather down-sampling). We also discuss the reconstruction task: reconstructing an original time series from its symbolic sequence. A synthetic summary is provided in Table II.1 on page 56. Finally, we discuss the symbolization of multivariate time series, which is a more recent research area.

II.2 Segmentation

The first step of symbolization, called *segmentation*, splits a time series into several segments that can be of equal length or not. There are two ways to perform segmentation: *uniform segmentation* where each segment has the same length, and *adaptive segmentation* when the segmentation is data-driven. A taxonomy of segmentation techniques (in the symbolization literature) is displayed on Figure II.3.

II.2.1 Uniform segmentation

Uniform segmentation is the most straightforward and commonly used in the literature on symbolization [Lin+03; LSK06; PLD10b; BABO12; MF12; Fua12; Esm+12; LZY12; Mal+13; LDH13; Bai+13; Sun+14; SP+17; Zha+18; AHWM19; Rua+20; LTN20; ZDX20; KR20; BTT21a; KR21]. *Piecewise Aggregate Approximation (PAA)* representation [Keo+01; YF00] is

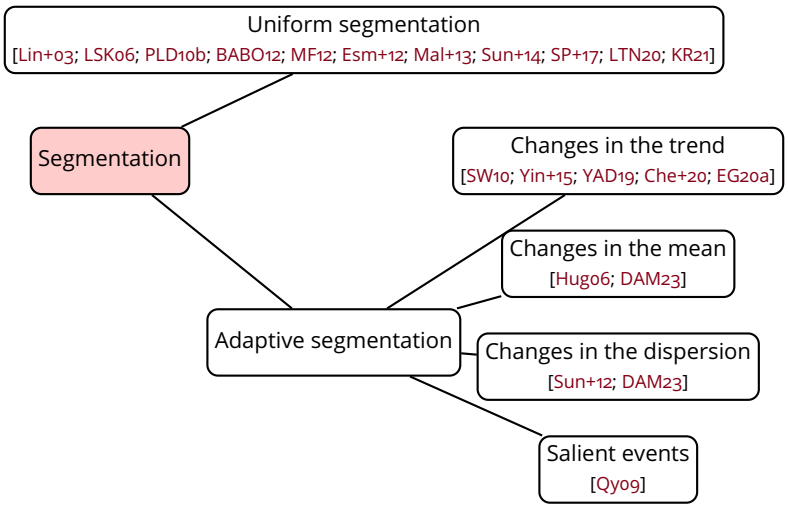


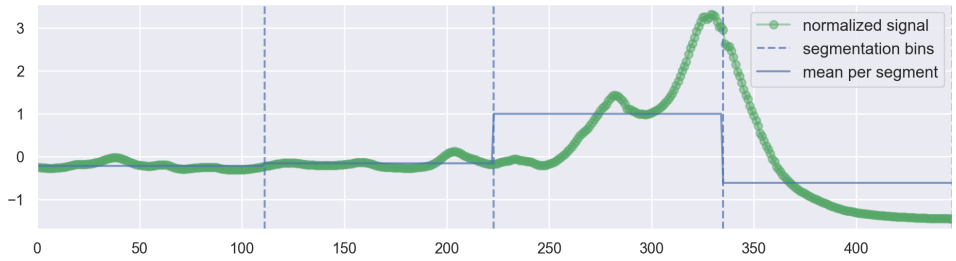
Figure II.3: Taxonomy of segmentation techniques in the symbolization literature. For conciseness, we do not display all the symbolization methods that use uniform segmentation.

an intermediate step of SAX: it uses uniform segmentation and then represents each segment by its mean. The number of segments w is set by the user. The larger w, the better the representation, but the larger the memory usage. The PAA representation, with several values of w, of a univariate signal is shown in Figure II.4. This signal is used as a running example and also appears in Figure II.5. When w increases, the mean value per segment better represents the signal's shape. Indeed, the peaks are better accounted for when $w = 16$ compared to $w = 4$.

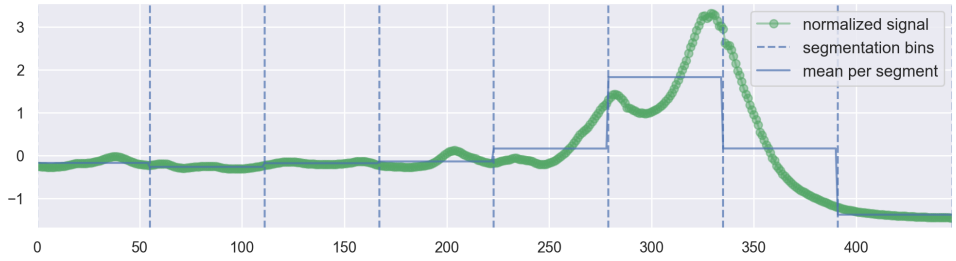
Finding an appropriate value for w and A is data-dependent and difficult. Some methods to automatically find it have been proposed [MABH11; CA15; ZY17]. *Harmony search SAX (HSAX)* [MABH11] is based on *Harmony Search (HS)* [GKL01] algorithms. An improved version of HSAX is *SAX++* [AABH14] which uses the relative frequency method. Instead of trying to find the best value of w, some classification methods use several values of w for a multiresolution representation in a supervised setting [NI22]. Moreover, some applications of symbolic methods replace the uniform segmentation by overlapping sliding windows, especially in classification [LKL12; SM13]. *Bag-Of-Patterns (BOP)* [LKL12] uses overlapping sliding windows and SAX is applied to each sliding window (for example fixing $w = 4$ and $A = 4$). BOP is a representation based on histograms of SAX word occurrence, similar to the bag-of-words representation in the text processing community. *SAX-VSM* [SM13], designed for classification, builds bag-of-words for each class using a sliding window.

II.2.2 Adaptive segmentation

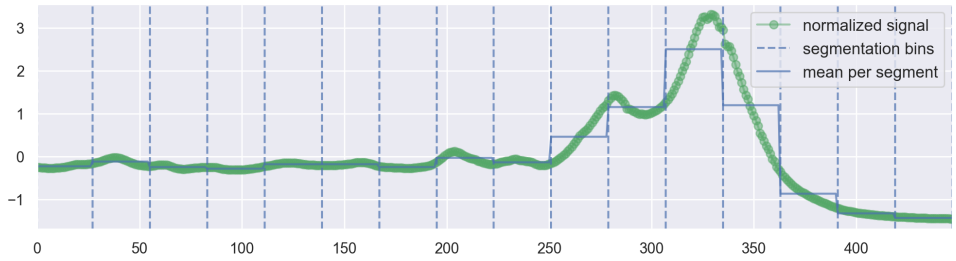
Contrary to uniform segmentation, adaptive segmentation is data-driven and adapts to the intrinsic properties of the signal. The number of segments w is either chosen by the user or controlled by another parameter (a threshold on the approximation error or a penalization). Adaptive segmentation is mostly based on *change-point detection* [TOV20], which finds unknown instants where some characteristics of the signal



(a) $w = 4$



(b) $w = 8$

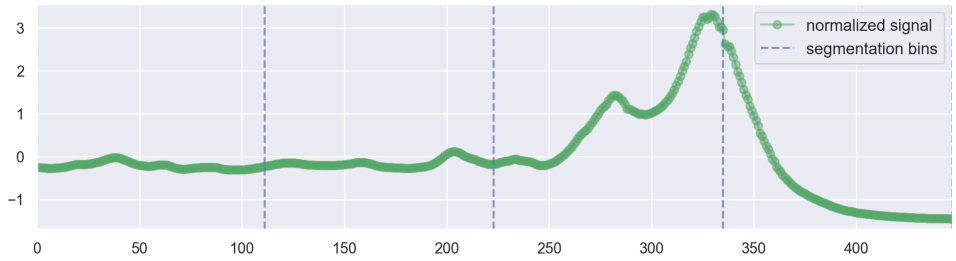


(c) $w = 16$

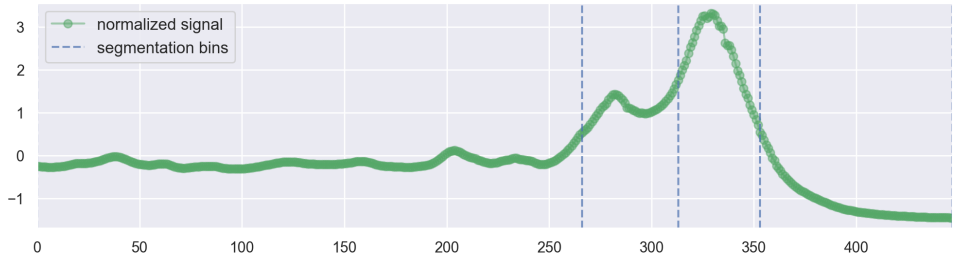
Figure II.4: PAA representation, based uniform segmentation, of a signal from the Meat data set (UCR Time Series Classification Archive), for several values of w. The original length of the signal is $n = 448$.

change abruptly. To illustrate, a comparison of uniform and adaptive segmentation, with the detection of changes in the mean or in the slope, is displayed in Figure II.5. Unlike uniform segmentation, the resulting segments of adaptive segmentation have varying lengths. The input time series are also allowed to have different lengths.

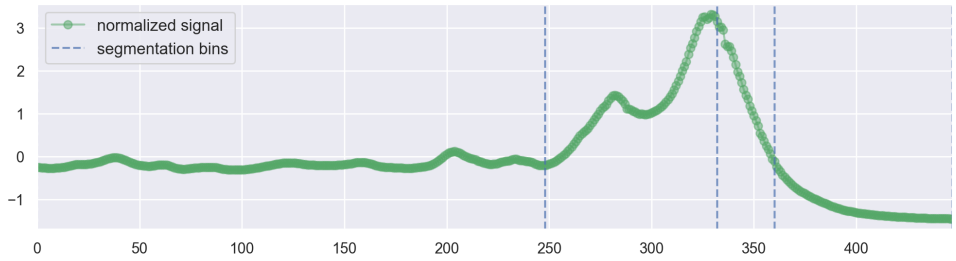
Detecting changes in trend. Adaptive segmentation can detect several kinds of change in a signal. In the symbolization literature, the most popular adaptive segmentation is perhaps on the trend [SW10; YAD19; EG20a; Yin+15; Che+20]. As PAA in SAX, the *Piecewise Linear Approximation (PLA)* [SZ96] a.k.a. *Piecewise Linear Representation (PLR)* [Keo+04] is used as an intermediate representation in some symbolization methods [SW10; EG20a]. A review of PLR [Keo+04] categorizes the algorithms into sliding windows, top-down, or bottom-up. This review also distinguishes *linear interpolation*, where the approximating line on each segment is simply the line connecting the starting and ending points, with *linear representation*, where the approximating line is the one that minimizes the least squares error. PLR is used in a symbol-based proce-



(a) Uniform segmentation.



(b) Adaptive segmentation with detection of changes in the mean.



(c) Adaptive segmentation with detection of changes in the slope.

Figure II.5: Comparing uniform segmentation with adaptive segmentation (detecting either changes in the mean or in the slope), with a fixed number of segments $w = 4$.

dure to detect phases of gait signals [SW10] and aims at detecting peaks in acceleration signals, which are related to events such as heel-strike and toe-off. *Adaptive Brownian Bridge-based Aggregation (ABBA)* [EG20a] focuses on the shape of the time series and its local up-and-down behavior, claiming that it corresponds to the human intuition of the summary of a signal. Each linear piece is chosen given a user-specified tolerance tol: when the value of tol increases, the resulting number of segments w decreases. *IEPF-TSR* [Che+20] employs the *Iterative End Point Fitting (IEPF)* [Ram72] algorithm that performs multiple iterations.

Still focusing on the trend but without using PLR, *SAX_CP* [YAD19] adds a penalization on the number of change-points. Using SAX_CP, for a single time series, the segments have different lengths, but all time series of a data set have the same change-points. These change-points are estimated on a training set when used in a supervised setting. SAX_CP claims that trend information is essential in fields such as finance, quality control, stocks, and service quality. On its side, *TFSA* [Yin+15] also detects changes in the trend and implements a two-step adaptive segmentation: the obtained segments during the first step are further divided into shorter segments in

the second step. This two-step mechanism aims at reducing the time complexity, especially for long time series, because the second segmentation is faster than the first and of the possibility of parallelization on each segment. TFSA proposes a method to find global key points, which is inspired by the cumulative sum control chart [YLV04]. TFSA states that the trend is an important feature in many domains, such as satellite monitoring, and that it corresponds to human intuition in finance or health.

Detecting changes in mean. Rather than detecting changes in the trend, another category of adaptive segmentation focuses on the mean [Hug06; DAM23]. *ASAX_SAE* [DAM23] uses a bottom-up approach to reduce the approximation error of the PAA representation. It also introduces a dynamic programming algorithm to improve the segmentation computation time. According to [DAM23], this method is suited for data sets with unbalanced distributions and for the similarity search task. *SBSR* [Hug06] does not apply change-point detection. SBSR can be viewed as a symbolic version of the *Adaptive Piecewise Constant Approximation (APCA)* representation [Cha+02], just as SAX can be viewed as the symbolic version of PAA [Keo+01; YF00]. APCA is based on the *Discrete Wavelet Transform (DWT)* [CF99].

Detecting changes in dispersion. Other methods focus on the dispersion that can be captured by the variance or the entropy [Sun+12; DAM23]. *VWSAX* [Sun+12] detects changes in the variance using a fixed-size sliding window. In the sliding window procedure, once a segment has enough total variance according to a threshold, it is transformed using SAX. *ASAX_EN* [DAM23] focuses on entropy and uses a top-down approach to find informative segments with high entropy.

Segmentation based on salient events. Instead of performing change-point detection based on a chosen feature, *KP_SAX* [Qyo9] finds so-called "Key Points". It has a two-step segmentation: the adaptive segmentation called *KP_SEG* finds potential change-points in the first step that are refined in the second step. The first step of KP_SEG finds the set of "Extreme Points", which are defined as points with a change in monotonicity. The second step of KP_SEG finds the "Key points" out of the "Extreme Points" of the first step. For an extreme point to be considered as a key point, the ratio between the segment length and the total length must be larger than a threshold or the angle between the line from the previous sample to the current sample and the line from the current sample to the next sample, must be smaller than an angle threshold. Finally, we quickly mention similar time series representations that perform adaptive segmentation, especially in finance: *Perceptually Important Points* [Chu+01], *Turning Points* [BY08], and *Important Points* [PSS08].

A summary of adaptive segmentation techniques used in the symbolization literature is given in Table II.1 on page 56, and a taxonomy was shared in Table II.3 on page 42. Now, let us describe feature extraction methods in detail.

II.3 Feature extraction

We now describe how to extract segment features after the segmentation step. In the literature on symbolization, extracted features can be the mean, the slope, the variance, the maximum, etc.

Note that, many methods, like SAX, only extract the segment mean [RKAJB05; MU05; Hug06; SK08; Qyo9; PLD10b; Sun+12; MF12; Fua12; Bai+13; KR20; KR21; DAM23]. In the following, we focus on methods that extract other features. A taxonomy of feature extraction techniques used in the symbolization literature is presented in Figure II.6.

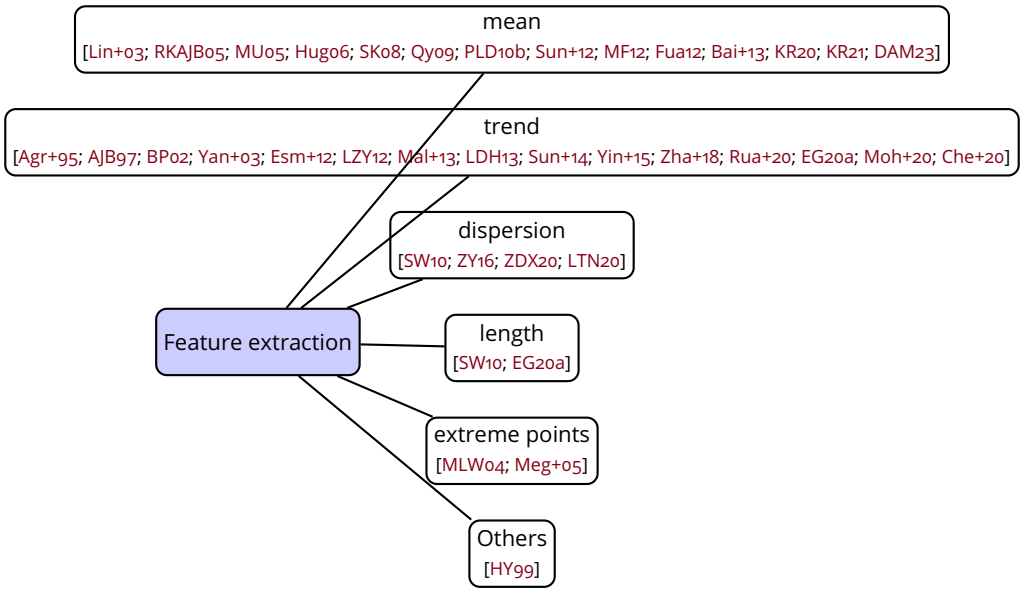


Figure II.6: Taxonomy of feature extractions in the symbolization literature. For conciseness, we do not display all symbolizations that extract the mean.

II.3.1 Extracting the trend

In the symbolization literature, the main alternative to extracting the mean per segment is the trend [HA07; LZY12]. In the literature, the trend can be extracted in several ways: the slope [Esm+12; Mal+13; Yin+15; Zha+18; Rua+20; Che+20], the direction (convex, concave, or linear) [LDH13], the increment [EG20a], or some ad-hoc features [Sun+14]. These different ways of encoding trend information may impact performance: ABBA [EG20a] claims that extracting the increment instead of the slope allows their procedure to be independent of any pre-processing.

Mostly, variants do not replace the mean symbol with a trend one but rather add the trend symbol to the mean symbol, obtained as in SAX, leading to at least two symbols per segment [Esm+12; LDH13; Sun+14; Zha+18; Rua+20; Che+20]. For instance, for the symbolized slope, *TVA* [Esm+12] and *TSAX* [Zha+18] only consider upward trend ($\nearrow$), a downward trend ($\searrow$), and a straight trend ($\rightarrow$). In order to represent the global trend more accurately, TSAX adds two trend symbols to the mean: each PAA segment

is further split into 3 segments to determine these two trend symbols. For example, on a PAA segment, the two trend symbols can be ↗ and then ↘. The slope can also be quantized using angle values, as in *TrSAX* [Rua+20] and *TSX* [LZY12]. In TSX, each segment is represented by 4 symbols. TSX first defines a line called *trend line* that connects the starting and ending points of a segment. The *Most Peak point (MP)* is above the trend line and has the largest distance to the trend line. The *Most Dip point (MD)* is below the trend line and has the largest distance to the trend line. TSX draws three trend lines connecting the four following key points of a segment: the starting and ending points, as well as the MP and MD. Finally, TSX holds 4 values per segment: the symbolized mean and the three slope values. Some methods use the trend as well the starting or ending point, especially when using adaptive segmentation [Yin+15; Che+20]. IEPF-TSR [Che+20] uses the symbolized mean, the slope, and the starting point. TFSA [Yin+15] is quite different from the other methods as it does not use the symbolized mean but the following three features: the symbolized trend (upward, downward, flat following upward, and flat following downward), the slope, and the end point. Its segmentation is adaptive and detects key points, as described in Section II.2.2. Instead of having several symbols per segment, *1d-SAX* [Mal+13] represents two features with only one symbol per segment. It uses linear regression to compute the mean and the slope of each segment, then discretizes the mean and the slope separately using the same Gaussian assumption as in SAX. The final segment symbol is the combination of the mean symbol and the slope symbol.

Ordinal patterns [BP02] is a symbolization approach that differs from previously described SAX variants: it only describes the up and down trends. There is no segmentation: the original time series is down-sampled and a delay parameter defines the equal-size duration between each sample. Let us consider that the down-sampled time series is $(2.24, 1.23, 5.42, 4.21)$ which contains 4 values. For the sample of value 5.42, the ordinal pattern of order $n_{\text{order}} = 3$ is $(1, 0, 2)$ because of the ordering of the two previous values and itself. As another example, the ordinal pattern of sample 4.21 is $(0, 2, 1)$. For an ordinal pattern of order $n_{\text{order}} = 3$, there exists $n_{\text{order}}!$ possible values. *Permutation Entropy (PE)* is often applied to these ordinal patterns. PE measures the complexity of a time series: it is high when the obtained ordinal patterns are random. Hence, ordinal patterns can be seen as symbols themselves, while PE on ordinal patterns can be seen as feature extraction. Similarly to ordinal patterns, [Yan+03] uses binary symbols ($A = 2$) to encode an increasing or decreasing trend. For each sample of the time series, if its value is larger than the previous sample, then the attributed symbol is 1, and 0 otherwise. The chosen number of successive pairs of values is the word length, hence there is no segmentation *per se*. It was initially used for challenging physiological signals. A similarity measure on the symbolic sequences is introduced and uses rank order statistics.

Finally, *Shape Definition Language (SDL)* [Agr+95] defines a specific alphabet for the trend. It was originally applied to query time series. SDL allows fuzzy matchings where the user is more interested in the overall shape rather than specific details. SDL defines a specific alphabet where each symbol represents an event, such as slightly increasing transition and highly increasing transition. There is no segmentation. Each event is defined by lower and upper bounds on the variation between the consecutive points and constraints on these two points (being zero, non-zero, or else). *Shape Description Alphabet (SDA)* [AJB97] is based on SDL [Agr+95] and rewrites the proposed alphabet.

II.3.2 Extracting the dispersion

Some symbolization techniques use the dispersion. The dispersion is either the standard deviation [SW10; ZY16], the entropy [ZDX20], or the complexity estimate [LTN20]. Apart from [SW10] (described in Section II.3.3), these methods use the symbolized mean and the real-valued dispersion, and thus hold two values per segment. The idea is that the mean and the dispersion provide complementary information. The *complexity estimate* CE of a time series x of length n is defined in [Bat+14] as

$$CE(x) = \sqrt{\sum_{i=1}^{n-1} (x_i - x_{i+1})^2}, \tag{II.1}$$

and corresponds to the L_2-norm of the finite differences vector. *SAX_SD* [ZY16], which extracts the symbolized mean and the symbolized standard deviation, has been improved by the same authors into *autoSAXSD_S* and *autoSAXSD_M* [ZY17] that automatically estimate the parameters of SAX and SAX_SD. autoSAXSD_S chooses the best value of w using Shannon's sampling theorem. Alternatively, autoSAXSD_M applies adaptive segmentation based on the change in the mean. Both autoSAXSD_S and autoSAXSD_M estimate the best value of A by investigating the distribution of the means values, and especially its skewness; it iterates over several values of A.

II.3.3 Extracting the length

When using adaptive segmentation, each segment usually has a different length and is considered a feature in certain works. A method that focuses on the analysis of acceleration signals [SW10] extracts five real-valued features: the variance, the mean along two axes, the trend information, and the segment length. ABBA [EG20a] aims to represent the shape of a time series and incorporates two features: the increment and the length. A weighting parameter enables the user to promote either the increment or the length, depending on the application.

II.3.4 Extracting extreme points

Rather than using the trend, the dispersion, or the length, *Extended SAX (ESAX)* [LSK06] focuses on extreme points and represents each segment by its mean, minimum, and maximum values. It was initially crafted for financial data. On each segment, the symbols for the mean, minimum, and maximum are ordered according to their time position in the segment.

II.3.5 Others

Finally, let us describe a quite unique symbolization technique called *IMPACTS* [HY99]. IMPACTS was designed for indexing, and we only describe its symbolization part, which serves as a preprocessing step for indexing. Its segmentation amounts to the uniform one with a fixed number of segments as input. Then, a unique symbol is mapped to each segment: the alphabet size A is the same as the word length w. Hence, no specific feature is extracted: all the sample values in a segment are used.

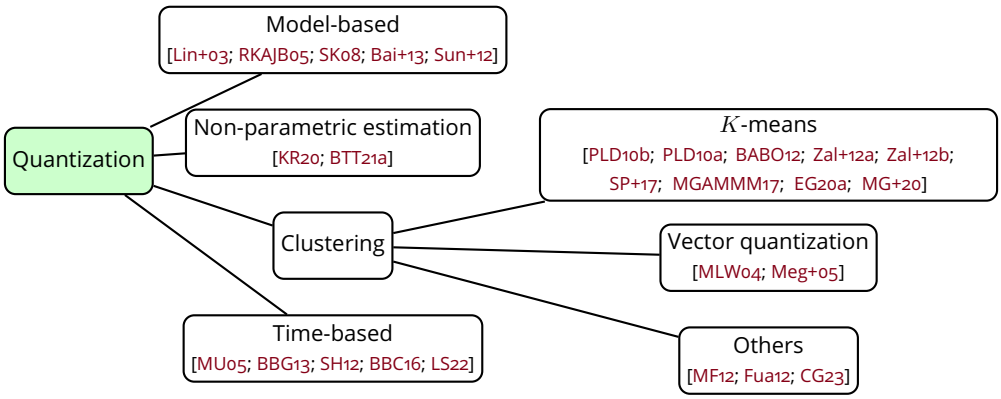


Figure II.7: Taxonomy of quantization techniques in the symbolization literature. For conciseness, we do not display the symbolization methods that use the exact same quantization as SAX.

II.4 Quantization

Following segmentation and feature extraction, quantization performs a mapping from the extracted features to a discrete value called a symbol. Quantization methods fall into several categories. Model-based methods assume a data model to determine the quantization bins; for example, SAX assumes that the segment means follow a Gaussian distribution. Conversely, non-parametric approaches estimate the quantization bins without model assumption. Usually, these two approaches solely deal with one feature and when there are several features, each one is quantized independently. On the contrary, clustering methods can directly input several features. Finally, we review methods incorporating time information while encoding the symbols. Note that some simple and straightforward quantization techniques were described in Section II.3.1 about extracting the trend feature and are not mentioned in this section.

A taxonomy of quantization methods used in the symbolization literature is presented in Figure II.7.

II.4.1 Model-based

As their name suggests, model-based quantization techniques assume a data model to determine the quantization bins. Most methods use the same Gaussian assumption as SAX: they are mentioned in Table II.1 on page 56 and in previous sections, but are not further described here. Other approaches extend the Gaussian bins of SAX [SK08; Bai+13]. In *indexable SAX (iSAX)* [SK08], an iSAX word is a SAX word where each symbol is represented using binary numbers (instead of alphabetical letters or integers). The quantization of iSAX also uses Gaussian bins, but it changes the cardinality of the symbols. When the cardinality (alphabet size) is a power of 2, using binary numbers enables one to change the cardinality of each iSAX word (into another power of 2) and the cardinality of each symbol inside an iSAX word. Basically, it lowers the alphabet size by aggregating close symbols. In the end, each symbol of an iSAX word can have a different alphabet size, which is always a power of 2. iSAX words of different cardinalities can be compared, and this multi-resolution trick allows one to index

time series much faster. An improvement of iSAX has been proposed: *Auto-iSAX* [CA15] estimates the best parameters of iSAX [SK08]: the best w per iSAX word, and the best value of A per symbol inside an iSAX word. Auto-iSAX tunes w by using the complexity estimate defined in Formula (II.1) on page 48 and tunes A by using the standard deviation. As for *Random Shifting based SAX (rSAX)* [Bai+13], it slightly modifies the Gaussian quantization bins of SAX by applying random shifting. The goal of rSAX is to have "soft borders": points that are similar but on the other side of Gaussian borders will have a higher probability of being mapped to the same symbol. Thus, the quantization bins are more intuitive.

Instead of using a Gaussian distribution, *Weibull-SAX (W-SAX)* [AHWM19] uses a Weibull distribution that is more suitable for predictive maintenance tasks where time series are composed of healthy states and degraded ones. W-SAX uses learned parameters for the distribution. Finally, the *clipped representation* [RKAJB05] is a bit level representation of time series ($A = 2$): the unique quantization bin is the time series mean. If the value of a sample is above the mean of the time series, its symbol is 1, otherwise 0. There is no segmentation step: all samples are quantized. As it is recommended that the time series are normalized [KK03], the mean should be zero. The clipped representation uses run-length encoding whilst taking advantage of the binary nature of the symbolic sequences, and also applies numerosity reduction.

II.4.2 Non-parametric estimation

Rather than using a predefined distribution, some methods have tried to estimate it in a data-adaptive fashion. A straightforward approach is to use quantiles [SH12; Zal+12a], the number of quantiles being determined by the number of symbols. *Distribution-Wise SAX (dwSAX)* [KR20] tackles non-Gaussian distributions. dwSAX estimates a data distribution of the PAA values using *Kernel Density Estimation (KDE)*. KDE requires the choice of a kernel function and a bandwidth parameter. After KDE, dwSAX finds the quantization bins using the *Probability Density Function (PDF)* so that they create equal-sized areas under the curve. An improved version called *edwSAX* [KR21] has been proposed by the same authors. *SAX using Kullback-Leibler (SAX-KL)* [BTT21a] is an anomaly detection based on a modified version of SAX and the Kullback-Leibler goodness-of-fit. The modified version of SAX performs adaptive quantization by estimating the PDF using KDE as in dwSAX [KR20], then a modified version of the Lloyd-Max algorithm to better detect the modes is applied to obtain the quantization bins. The symbolization step amounts to *probabilistic SAX (pSAX)* [BTT21b] by the same authors, which comes with a high computational cost.

II.4.3 Clustering

Symbolization methods widely use clustering to map the extracted features to symbols (see Figure II.7 on page 49). All points in the same cluster are attributed the same symbol. The K-means algorithm, called *Lloyd's algorithm* in one-dimension, is often used, where the number of clusters K equals the number of symbols A. In that case, the symbols are the cluster labels obtained by the clustering algorithm. Most symbolization techniques use K-means clustering (see Figure II.7). *Adaptive SAX (aSAX)* [PLD10b; PLD10a] uses a uniform segmentation and K-means clustering for the quantization. K-means is applied on a training set of PAA transformations from sev-

eral time series. aSAX has several applications for data mining tasks. aSAX has been applied to indexing with *iaSAX* [PLD10b] which is the adaptive version of iSAX [SK08]. aSAX has been used for anomaly detection with *HOT aSAX* [PLD10a] which is the adaptive version of HOT SAX [KLF05]. In *R-Kmeans* [SP+17], which is similar to aSAX, the clustering quantization step is done per class, as this method is applied to time series classification. [SP+17] also introduces *SAX-Kmeans* which is based on SAX and R-Kmeans, and *ESAX-Kmeans* which is based on ESAX [LSK06] and R-Kmeans. Rather than applying K-means directly on the PAA representation, [Zal+12a] computes the first-order differences of PAA values, then applies K-means clustering to obtain the symbolic sequences. Some approaches hold more than one feature per segment in order to preserve more information about the segments [BABO12; SW10]. *ENhanced SAX (ENSAX)* [BABO12] clusters the mean, the minimum, and the maximum. A symbol-based procedure to detect phases of gait signals [SW10] uses piecewise linear segmentation, then K-means clustering on the five features (described in Section II.3.3) per segment to get the symbols.

In ABBA [EG20a], following a PLR adaptive segmentation described in Section II.2.2, the K-means clustering inputs tuples of the increment over the segment and the segment length. ABBA uses a scaling parameter scl that calibrates the importance of the length in relation to its increment: the clustering is performed on the increments alone for $scl = 0$, while the clustering is done on both the length and increment with the same importance when $scl = 1$. Hence, the input parameters are the tolerance tol, the scaling scl, and the alphabet size A. When A is not set by the user, ABBA does several runs of the K-means algorithm to get the optimal value of A, resulting in a higher computational cost. A recent faster variant of the ABBA method, *fABBA* [CG23], replaces the K-means clustering by a sorting-based aggregation procedure that does not require the user to specify the alphabet size. *ABBA-LSTM* [EG20b] combines ABBA [EG20a] with LSTM for time series forecasting: it converts real-valued time series into symbolic sequences, then a LSTM predicts the symbols that are converted back to real values.

Another category of symbolization methods based on clustering employs *Vector Quantization (VQ)* [GG92]. *PVQA* [MLW04] uses uniform segmentation to obtain w segments, then VQ for the quantization. Each segment is attributed to a symbol which is its closest codeword taken from a codebook (or alphabet). The codebook is obtained from a training set of segments by applying the generalized Lloyd algorithm [Llo82], which is similar to K-means clustering. The authors of PVQA [MLW04] also introduced *MVQ* [Meg+05], which uses codebooks with different resolutions by using codebooks of different sizes.

Other algorithms are stochastic approaches and use genetic algorithms to do the clustering. *Genetic Algorithms-based SAX (GASAX)* [MF12] works as SAX, but the bin boundaries are determined through a genetic algorithm (which is a class of optimization procedures). GASAX does not require any specific distribution of the data. *DESAX* [Fua12] works like GASAX, but uses differential evolution instead of genetic algorithms to find the breakpoints. Quite differently, *eMODiTS* [MG+20], which enhances MODiTS [MGAMMM17], uses evolutionary programming, a multi-objective algorithm to have an alphabet size and quantization bins for each uniform segment.

II.4.4 Time-based

Apart from the symbol-based procedure to detect phases of gait signals [SW10] and ABBA [EG20a] that incorporate directly the length feature when creating their symbols (using clustering), some quantization methods incorporate the temporal information differently. In this section, we describe three independent methods where a symbol considers the temporal information without explicitly using the length in the feature extraction.

The *Persist algorithm* [MU05] is based on the persistence score of symbols which are considered as states, given A states. This persistence score is based on the symmetric *Kullback-Leibler (KL)* divergence of the non-self and self-transition probability distributions of the symbols according to a first-order Markov model. The more likely it is to observe the same segment as the previous one (self-transition is more probable than non-self-transition), the larger the persistence score based on KL. Persist is reviewed and experimentally evaluated in [SW11].

Symbolic Aggregate approXimation Optimized by data (SAXO) [BBG13; BBC16] is a parameter-free and adaptive time series symbolization. SAXO was initially used to represent electricity consumption, where the behavior changes drastically at night compared to the day. Hence, a good trade-off between compression and loss of information should have symbols that jointly represent the time and the values. To that aim, SAXO applies an unsupervised regularized Bayesian co-clustering method called *Minimum Optimized Description Length (MODL)* [Bou06]. On each obtained time segment, the number of symbols and their distribution is different. As a result, SAXO performs a joint adaptive segmentation and quantization that comes with a large time complexity cost.

Symbolic Fourier Approximation (SFA) [SH12] is based on the discrete Fourier transform. First, SFA selects the w Fourier coefficients of lowest frequencies, and second, uses a procedure called *Multiple Coefficient Binning (MCB)* to quantize them. In detail, MCB computes a user-defined number A of quantiles per Fourier coefficient across all signals of a data set, and each Fourier coefficient is represented by the bin (based on quantiles) to which it belongs. In a supervised data mining task, the MCB bins are learned on a training set. SFA naturally provides a low-pass filtering that reduces the influence of noise. Also, no distance on SFA's symbolic representations is described. Note that SFA does not go through a segmentation step but still has the w parameter that determines the length of the symbolic sequences. SFA has been widely used for dictionary-based time series classification, for example *Bag of SFA Symbols (BOSS)* [Sch15] and some extensions [Sch16; MVB19; Lar+19; SL17; SL23; Mid+20]. A recent variant of SFA called *SFFA* [LS22] applies the fractional Fourier transform instead of the discrete Fourier transform. Similarly to SAX-VSM [SM13], the authors have developed *SFFA-VSM*.

II.5 Reconstruction

Reconstruction is the inverse transformation of symbolization: the original signal is inferred from its symbolic sequences. Symbolization can be viewed as compression, while reconstruction can be viewed as decompression. Moreover, comparing the distance between the original time series and its reconstruction can give a general idea of the quality of the time series symbolization. Only a few papers on symbolic rep-

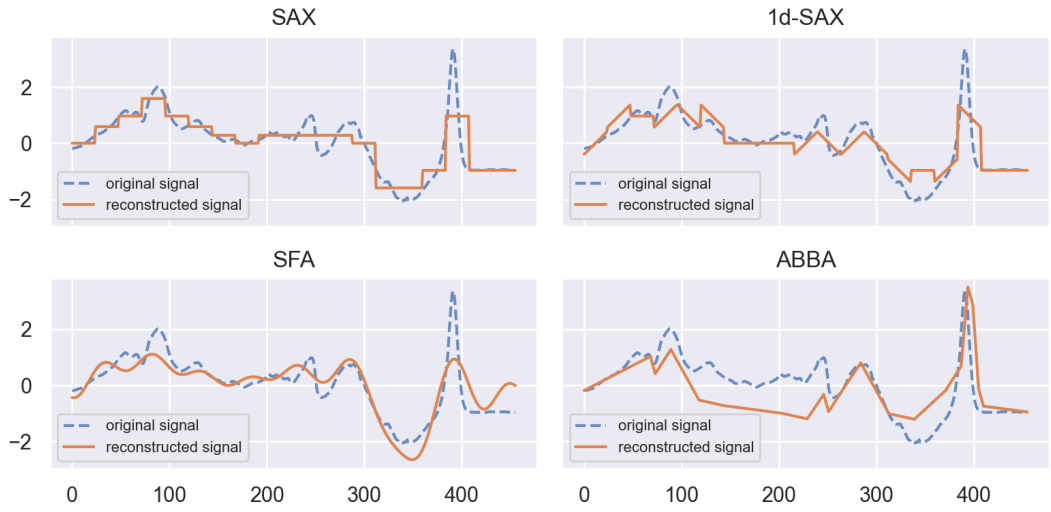


Figure II.8: Example of reconstruction of a single signal from the Beef data set (UCR Time Series Classification Archive) of original length $n = 470$ for several methods, with $w = 19$ and $A = 9$.

resentation tackle the signal reconstruction task. ABBA [EG20a] addresses the signal reconstruction task specifically. Papers on SAX, 1d-SAX, and SFA for example, do not tackle signal reconstruction. However, it is easy to infer a reconstruction procedure for these methods. As done in the `tslearn` Python package, for SAX[1] and 1d-SAX[2], the sample values on each segment of the reconstructed signal are based on the Gaussian bins of the look-up tables. For SFA, the reconstructed signal is the Fourier reconstruction based on the quantized Fourier coefficients. The reconstruction is quite smooth and provides low-pass filtering. For ABBA, as described in its original paper [EG20a], the reconstruction holds 3 steps for a symbolic representation. First of all, each symbol is associated with its corresponding cluster center. Then, as the lengths encoded in the cluster centers may not be integers, a trick aims at rounding them. Finally, a procedure reconstruction of the piecewise linear continuous approximation. Figure II.8 compares the reconstruction from these four symbolization methods for the same original time series.

II.6 Symbolic representations for multivariate time series

In this section, we describe a main challenge that is still an active area of research: extending symbolization methods to multivariate time series. While most symbolization techniques focus on univariate time series, some methods have extended procedures for the multivariate case. As illustrated in Figure II.9, there are three main strategies to tackle multivariate time series symbolization: dimensionality reduction, the independent strategy, and the dependent strategy.

[1] https://tslearn.readthedocs.io/en/stable/gen_modules/piecewise/tslearn.piecewise.SymbolicAggregateApproximation.html

[2] https://tslearn.readthedocs.io/en/stable/gen_modules/piecewise/tslearn.piecewise.OneD_SymbolicAggregateApproximation.html

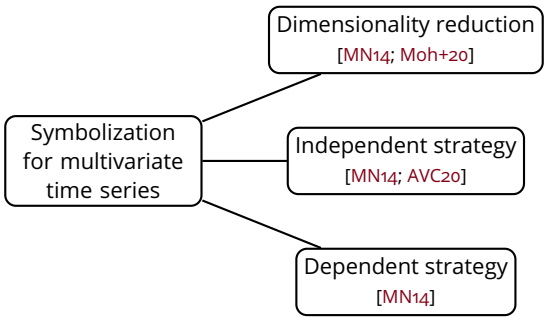


Figure II.9: Taxonomy of symbolization techniques for multivariate time series.

Dimensionality reduction. The dimensionality reduction techniques reduce the multivariate time series into a univariate time series and then use classic symbolization. *SAX-PCA* [MN14] applies PCA on the time series that are z-normalized on each dimension, then applies SAX to the projection of the time series on the first principal component, which is one-dimensional. Similarly, multivariate ordinal patterns [Moh+20] extend the ordinal pattern representation, which is based on the trend and described in Section II.3.1, to multivariate time series. It uses PCA to transform the multivariate time series into univariate ones, then applies the usual ordinal patterns on which the permutation entropy is then computed.

Independent strategies. The independent strategy symbolizes each channel independently and then aggregates them to return a single symbolic sequence. *SAX-REPEAT* [MN14] applies SAX on each dimension separately, then concatenates the multiple symbolic sequences obtained from each dimension into a single long string. *Multivariate SAX (MSAX)* [AVC20] applies PAA on each channel, and then a Gaussian distribution is associated with each variable. Then, a multivariate Gaussian distribution is formed and used for the quantization to obtain a univariate symbolic sequence. In the independent strategies, when there is an alphabet of size A for each dimension, then the total number of symbols is A^d, where d is the number of dimensions. Hence, the number of symbols does not scale well with the number of dimensions. For instance, MSAX was applied to trajectories where $d = 2$. Moreover, in these large alphabets, many symbols are not used.

Dependent strategies. The dependent strategy symbolizes all channels together and returns a single symbolic sequence. *SAX-ZSCORE* [MN14] starts by applying a multivariate version of the z-normalization step that uses the covariance matrix. Then, it applies a modified multivariate version of PAA: the mean per segment is a real value that corresponds to the average of the L_2-norms of each multidimensional sample.

In this section, we excluded symbolization methods that consist in multiple univariate symbolizations that are then handled by a data mining algorithm [Esm+12; Son+20; PJ20]. TVA [Esm+12] focuses on multivariate signal classification. *SAX-ARM* [PJ20] uses SAX to mine association rules efficiently among the deviant events of multivariate time series. They do not transform multivariate time series into a single symbolic sequence.

II.7 Conclusion

In this chapter, we conducted a survey of symbolization techniques. A synthetic summary of all univariate symbolization methods described in this review is provided in Table II.1 on page 56.

Symbolic representations are widely used when dealing with time series data for visualization and many data mining tasks such as classification. Symbolization can be used directly, for example, SAX with the MINDIST distance defined on its symbolic sequences (described in Section III.4.1) can be used in 1-nearest neighbor classification, or indirectly as an intermediate step, such as in SFA [SH12] which is used in the BOSS classifier [Sch15]. SAX is the most popular symbolic representation due to its simplicity and has paved the way for numerous SAX variants. Some variants focus on adaptive segmentation and/or adaptive quantization while extracting more relevant features than the mean per segment. Adaptive segmentation amounts to change-point detection: it looks for important points where there is a change in the mean or the slope, for example. Adaptive quantization uses distribution models, non-parametric estimation, or clustering to find quantization bins. They can also integrate the time information. These variants mainly focus on univariate time series, but some symbolization for multivariate time series have also been introduced more recently, some of them trying to deal with all channels simultaneously.

Table II.1: Summary of symbolization techniques for univariate time series found in the literature, ordered by year of publication and acronym.
For the feature extraction step, $\mathcal{A}$ indicates that the feature is extracted then later symbolized, while $\mathbb{R}$ indicates that the feature is used but not quantized afterwards (it remains a real value).
[†]Increment, not exactly the slope.

	Segmentation		**Feature extraction**										**Symbolization**	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
SDL (Shape Definition Language) [Agr+95]	✘	N/A	✘	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
SDA (Shape Description Alphabet) [AJB97]	✘	N/A	✘	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
IMPACTS (Interactive Matching of Patterns with Advanced Constraints in Time-Series databases) [HY99]	✘	N/A	N/A: no feature extraction										✔	model-based
Ordinal patterns [BP02]	✘	N/A	✘	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
Yang et al. [Yan+03]	✘	N/A	✘	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
SAX (Symbolic Aggregate approXimation) [Lin+03; Lin+07]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
PVQA (Piecewise Vector Quantized Approximation) [MLW04]	✘	N/A	N/A : vector quantization										✔	clustering

Continued on next page.

Table II.1 – continued from previous page

	Segmentation		Feature extraction										Symbolization	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
Clipped representation [RKAJB05]	N/A		$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	model-based
MVQ (Multiresolution Vector Quantized) [Meg+05]	✘	N/A	N/A : vector quantization										✔	clustering
Persist algorithm [MU05]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	time-based
ESAX (Extended SAX) [LSK06]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	$\mathcal{A}$	$\mathcal{A}$	✘	✘	✘	model-based
SBSR-Lo (adaptive Segmentation Based Symbolic Representations) [Hug06]	✔	mean	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
iSAX (indexable SAX) [SK08]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
KP_SAX (Key Points SAX) [Qyo09]	✔	salient points	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
Sant'Anna and Wickström [SW10]	✔	trend	$\mathcal{A}^2$	$\mathcal{A}$	✘	✘	$\mathcal{A}$	✘	✘	✘	✘	$\mathcal{A}$	✔	clustering
aSAX (adaptive SAX) [PLD10b; PLD10a]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
DESAX (Differential Evolution-Based SAX) [Fua12]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering

Continued on next page.

Table II.1 – continued from previous page

	Segmentation		Feature extraction										Symbolization	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
EFVD (Equal Fixed-Values Discretization) [Zal+12b]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
EN-SAX (ENhanced SAX) [BABO12]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	$\mathcal{A}$	$\mathcal{A}$	✘	✘	✔	clustering
GASAX (Genetic Algorithms-based SAX) [MF12]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
SFA (Symbolic Fourier Approximation) [SH12]	N/A : Fourier coefficients												✔	time-based
TSX (Trend-based Symbolic approximation) [LZY12]	✘	N/A	$\mathcal{A}$	$\mathcal{A}^3$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
TVA (Trend-based and Valued-based Approximation) [Esm+12]	✘	N/A	$\mathcal{A}$	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
VWSAX (Variance-Wise segmentation SAX) [Sun+12]	✔	variance	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
1d-SAX [Mal+13]	✘	N/A	$\mathcal{A}$	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
rSAX (Random shifting based SAX) [Bai+13]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	model-based

Continued on next page.

Table II.1 – continued from previous page

	Segmentation		Feature extraction										Symbolization	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
SAX_DR (SAX with Direction Representation) [LDH13]	✘	N/A	$\mathcal{A}$	✘	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	model-based
SAXO (Symbolic Aggregate approXimation Optimized by data) [BBG13; BBC16]	✔	N/A	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	time-based
SAX-TD (SAX-Trend Distance) [Sun+14]	✘	N/A	$\mathcal{A}$	$\mathbb{R}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
TFSA (Trend Feature Symbolic Approximation) [Yin+15]	✔	trend	✘	$\mathcal{A} \times \mathbb{R}$	✘	✘	✘	✘	✘	✘	$\mathbb{R}$	✘	✘	model-based
SAX_SD (SAX with Standard Deviation) [ZY16]	✘	N/A	$\mathcal{A}$	✘	✘	$\mathbb{R}$	✘	✘	✘	✘	✘	✘	✘	model-based
MODiTS (Multi-objective symbOlic Discretization for Time Series) [MGAMMM17]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
R-Kmeans (Representation Kmeans) [SP+17]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
TSAX (Trend-based SAX) [Zha+18]	✘	N/A	$\mathcal{A}$	$\mathcal{A}^2$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based

Continued on next page.

Table II.1 – continued from previous page

	Segmentation		Feature extraction										Symbolization	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
SAX_CP (SAX Change-Points) [YAD19]	✔	trend	$\mathcal{A}$	$\mathcal{A} \times \mathbb{R}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
W-SAX (Weibull-SAX) [AHWM19]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	non-param. est.
ABBA (Adaptive Brownian Bridge-based Aggregation) [EG20a]	✔	trend	✘	$\mathcal{A}^\dagger$	✘	✘	✘	✘	✘	✘	✘	$\mathcal{A}$	✔	clustering
CSAX (Complexity-invariant SAX) [LTN20]	✘	N/A	$\mathcal{A}$	✘	✘	✘	$\mathbb{R}$	✘	✘	✘	✘	✘	✘	model-based
eMODiTS (enhanced Multi-objective symbOlic Discretization for Time Series) [MG+20]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	clustering
EN_SAX (ENtropy-based SAX) [ZDX20]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	$\mathbb{R}$	✘	✘	✘	✘	✘	model-based
dwSAX (Distribution-Wise SAX) [KR20]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	non-param. est.
IEPF-TSR (Trend Segmentation Representation based on Iterative End Point Fitting algorithm) [Che+20]	✔	trend	$\mathcal{A}$	$\mathbb{R}$	✘	✘	✘	✘	✘	✘	$\mathbb{R}$	✘	✘	model-based

Continued on next page.

Table II.1 – continued from previous page

	Segmentation		Feature extraction										Symbolization	
Method	adaptive?	change-points type	mean	slope	direction	variance	complexity estimate	entropy	minimum	maximum	start / end point	length	adaptive?	type
TrSAX (Trend SAX) [Rua+20]	✘	N/A	$\mathcal{A}$	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
edwSAX [KR21]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	non-param. est.
SAX-KL (SAX using Kullback-Leibler) [BTT21a]	✘	N/A	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	non-param. est.
SFFA (Symbol Fractinal Fourier Approximation) [LS22]	N/A : Fourier coefficients												✔	time-based
ASAX_EN (Adaptive SAX based on ENtropy) [DAM23]	✔	entropy	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✔	model-based
ASAX_SAE (Adaptive SAX based on the Sum of Absolute Errors) [DAM23]	✔	mean	$\mathcal{A}$	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	model-based
fABBA (fast ABBA) [CG23]	✔	trend	✘	$\mathcal{A}^{\dagger}$	✘	✘	✘	✘	✘	✘	✘	$\mathcal{A}$	✔	clustering

Chapter III

Distance measures on time series, strings, and symbolic sequences

This chapter reviews popular distance measures on time series, strings, and symbolic sequences. We first describe distances on univariate time series, including the popular elastic distances and numerous variants of DTW. Afterwards, we describe distance measures on strings such as the edit distances. Then, we describe distance measures that have been defined on symbolic sequences, i.e. resulting from a symbolic representation. Finally, we describe how distances have been extended to multivariate time series.

Contents

III.1 Introduction

In this chapter, we survey distance measures on time series, strings, and symbolic sequences. First of all, let us define these objects.

- A *time series* is a series of real values indexed in time order.
- A *string* is a series of discrete values indexed in time order, the discrete values being non-ordered and taken from a fixed alphabet of characters.
- A *symbolic sequence* is a discrete sequence resulting from the transformation of a time series using a symbolization process (described in Chapter II).

In the following, $\mathcal{A}$ denotes the alphabet, that is, a set of symbols, e.g. $\mathcal{A} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ where a, b, c, ... are symbols. $A = |\mathcal{A}|$ is the alphabet size.

III.1.1 Definitions

Measuring the distance (or similarity) between series is key in many machine learning tasks [EA12]. A distance measure computes a real value that quantifies the similarity between two sets of values. For two series (with discrete or real values) x and y of respective lengths m and n, a distance (or similarity) measure D is defined as

$$\begin{array}{rccc} D & : & \mathbb{B}^m \times \mathbb{B}^n & \rightarrow & \mathbb{R} \\ & & (x, y) & \mapsto & D(x, y) \end{array}, \tag{III.1}$$

where $\mathbb{B}$ designates either the alphabet $\mathcal{A}$ in case of strings / symbolic sequences or the set of real number $\mathbb{R}$ in case of time series. The challenge of building a distance measure is to make it compatible with any series, whatever their nature, their size,

etc, and also to formalize the human intuition of what makes two series different or not, although they are not identical from a mathematical viewpoint [EA12].

A particular case of distance measure is the metric:

Definition III.1 (Metric)**.** *A measure D is a metric if it satisfies the three following fundamental properties, for any sequences (with discrete or real values) x, y, and z:*

1. Identity
$$D(x, y) = 0 \iff x = y; \tag{III.2}$$
2. Symmetry
$$D(x, y) = D(y, x); \tag{III.3}$$
3. Triangle inequality
$$D(x, y) \leq D(x, z) + D(z, y). \tag{III.4}$$

If any of these three is not verified, then the distance measure is not a metric.

Note that the three properties described in Definition III.1 imply the *non-negativity* property of a metric

$$D(x, y) \geq 0. \tag{III.5}$$

The triangle inequality property is also known as *subadditivity* [EA12].

III.1.2 Applications of distances measures

III.1.2.1 Data mining tasks

Distance measures are particularly useful in data mining tasks. Let us review the use of distance measures for each of the main data mining tasks on time series.

- Distance measures are omnipresent in indexing and similarity search [GD01; Cha+02; KK03; Din+08; EA12; Rak+12; TWP17]. Indeed, given a query time series, nearest neighbor search looks for the closest point out of a set of candidate points, according to a distance measure.
- Distances on time series are crucial in time series clustering, such as for K-means and agglomerative clustering. Reviews on time series clustering, including a recent one, are available at [War05; Ber06; Fu11; ASY15; HMB23].
- Classifiers such k-nearest neighbors classification require a distance measure. Reviews on univariate and/or multivariate time series classification are available at [Bag+17; AML19; Rui+21; MSB23; Shi+23] and apply the algorithms to the open-access UCR archive [Dau+19].
- For anomaly detection (also known as outlier detection), a distance on subsequences can be used [IP14; BBC18]. Anomaly detection is tackled by using clustering on subsequences, and considering that some groups are anomalies, while others are normal. Reviews including recent ones are available at [Wei04; CBK09; BG+21; Pap+22].
- Moreover, in the forecasting task, to assess the quality of a prediction, a distance measure is used to compare the ground truth values with predicted ones.

- Similarly, in the reconstruction task, a distance measure is needed to compare the original time series with its reconstruction (after a certain transformation).

The metric property defined in Definition III.1 is particularly useful for some data mining tasks. The triangle inequality can help time series indexing and can be used to accelerate the time series retrieval task. A lot of algorithms have been optimized to index and retrieve objects in metric spaces [Cho1]. For example, that is the case of the widely-used indexing framework called *GEMINI (GEneric Multimedia INdexIng)* [FRM94].

III.1.2.2 Lower-Bounding property

As stated in [FRM94; Keo+01; HW21], the Lower-Bounding (LB) property, in Definition III.2, is important when performing similarity search such as nearest neighbor search.

Definition III.2 (Lower-bound of a distance)**.** *A lower-bound LB of a distance D is an easy to compute approximation of D such that for all time series x and y*

$$LB(x, y) \leq D(x, y). \tag{III.6}$$

A lower-bound is particularly useful when looking for the nearest neighbor given a time series query x. Indeed, the search investigates each candidate iteratively: for a candidate $y \in \mathcal{C}$ (where $\mathcal{C}$ the set of time series candidates), if $LB(x, y) \geq D(x, y_{bsf})$ where y_{bsf} is the current nearest neighbor ("best-so-far" [Keo+09; SB16; Sil+18]), then we have $D(x, y) \geq LB(x, y) \geq D(x, y_{bsf})$ according to Formula (III.6), so candidate y can not be the nearest neighbor, and there is no need to compute $D(x, y)$. $D(x, y_{bsf})$ is known as the cut-off [HW21]. Ideally, a lower bound LB is faster to compute than its corresponding distance D. Typically, a lower bound would have a time complexity of one order of magnitude lower than its corresponding distance. Hence, using a lower-bound sometimes (e.g. if $LB(x, y) \geq D(x, y_{bsf})$) replaces long computations (e.g. $D(x, y)$) with faster ones (e.g. $LB(x, y)$), thus speed up the total search.

Moreover, the ideal lower-bound is tight. The *Tightness of Lower Bound (TLB)* [Keo+01], defined in Definition III.3, measures how close a lower bound is to its corresponding distance: the closer a TLB is to 1, the tighter (and better). Usually, there is a trade-off between the TLB and the computation time efficiency [Sil+18].

Definition III.3 (Tightness of Lower Bound)**.** *Given a distance measure D and a lower bound LB of D, the Tightness of Lower Bound (TLB) is defined as*

$$TLB = \frac{LB(x, y)}{D(x, y)} \leq 1. \tag{III.7}$$

Finally, the LB property ensures exact indexing of data in the sense that there will be completeness (no false negatives / dismissals) [FRM94].

III.1.3 Outline

A taxonomy of distance measures on time series, strings, and symbolic sequences, that will be described in this chapter, is displayed in Figure III.1. We first describe the distances on time series. Then, we focus on strings, mainly edit distances along with

their extension to time series. Next, we present the distances on symbolic sequence which are obtained after symbolization (described in Chapter II). Note that a distance measure on symbolic sequences can be viewed as a distance measure on strings, as well as a distance measure on time series when combined with a symbolization process. Finally, we describe the extensions of distances for multivariate time series. A summary table of distances on time series is available in Table III.2 on page 93, and a summary table of distances on strings is available in Table III.3 on page 94.

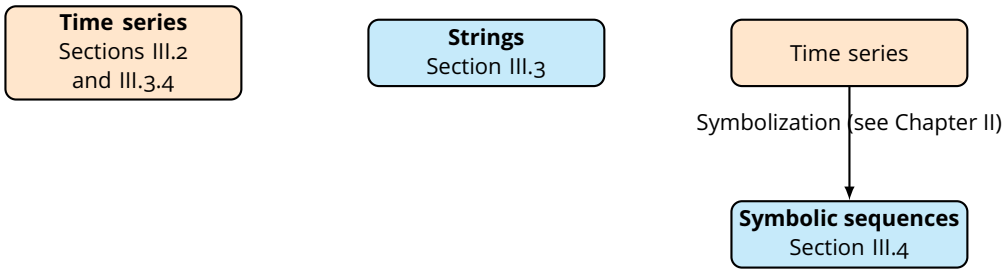


Figure III.1: Taxonomy of distance measures on time series, strings, and symbolic sequences described in this chapter.

III.2 Distance measures on time series

In this section, we describe the univariate measures, while the multivariate ones (which are actually extensions of the univariate ones) will be covered in Section III.5. More in-depth or complementary reviews on distance measures for time series can be found in [Shi+23; EA12; Fu11; HMB23; Wan+13; Rui+21; Rat+10; Cas+12]. Review [Shi+23] focuses on distance measures on multivariate time series but also describes univariate ones. Review [EA12] classifies the distance measures into four categories: shape-based, edit-based, feature-based, and structure-based. Some of these categories will not be covered. Shape-based distances will be covered with the Euclidean distance (and more generally L_p distances) in Section III.2.1, and Dynamic Time Warping (along with its variants) in Section III.2.2. [AML19] reviews distance-based time series classification, while [HMB23] reviews distance-based clustering.

Let us assume that we want to compute the distance between the two univariate real-valued time series $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ of respective lengths m and n, such as the ones depicted in Figure III.2.

III.2.1 Lp distances

Let us assume that x and y have the same length $m = n$. The most straightforward way to calculate a distance between two signals is to use the L_p distance [YF00] defined in Definition III.4.

Definition III.4 (L_p distance)**.** *The L_p distance where p is the order of the distance, also known as the Minkowski distance, between univariate time series x and y of same length*

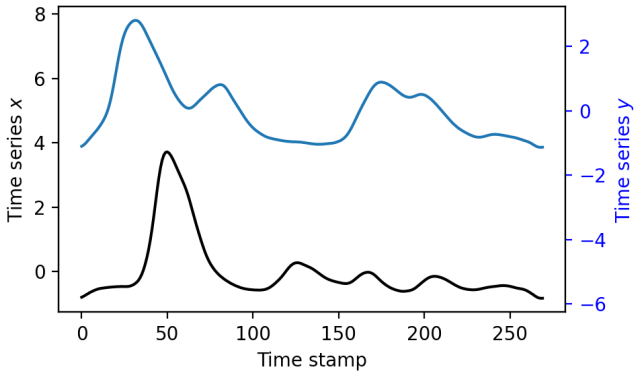


Figure III.2: Two equal-size univariate real-valued time series x and y. Note that there is separate amplitude axis for each time series.

n is defined by

$$L_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}. \tag{III.8}$$

The L_1 (Manhattan) and L_2 (Euclidean) distances are widely used. In particular, the Euclidean distance is often used as a baseline in data mining tasks, as it is well referenced, holds no parameter, and is easy to implement [KK03].

The time complexity of the L_p-norm (whatever p) is $\mathcal{O}(n)$. Thus, it is considered as one of the fastest in the community. For $p \geq 1$, the L_p distance is a metric. The L_p distance requires its inputs to have the same length, otherwise it is not defined.

An extension of the Euclidean distance is the *Complexity-Invariant Distance (CID)* [Bat+14] which is invariant to the complexity of a time series (for example a random walk is more "complex" than a linear line). The motivation is the following: complex time series are often considered close (by the Euclidean distance) to simple time series rather than other complex time series that actually bear a resemblance to them. To circumvent this issue, the CID introduces a correction factor to the Euclidean distance

$$D_{CID}(x, y) = L_2(x, y) \cdot D_{CF}(x, y), \tag{III.9}$$

where D_{CF} is a *complexity correction factor* defined as

$$D_{CF}(x, y) = \frac{\max\left(CE(x), CE(y)\right)}{\min\left(CE(x), CE(y)\right)} \geq 1, \tag{III.10}$$

and $CE(x)$ is the complexity estimate of time series x, as defined in Formula (II.1) on page 48. The D_{CF} term forces time series with very different complexities to have a larger distance according to D_{CID} (relatively to time series with similar complexities). If two time series have the same complexity, their CID corresponds to the classic Euclidean distance.

III.2.2 Dynamic Time Warping (DTW): an elastic distance measure

We now focus on *Dynamic Time Warping (DTW)* [SC71; SC78; BC94], a distance that can cope with signals of different lengths. Such a distance is called an elastic distance

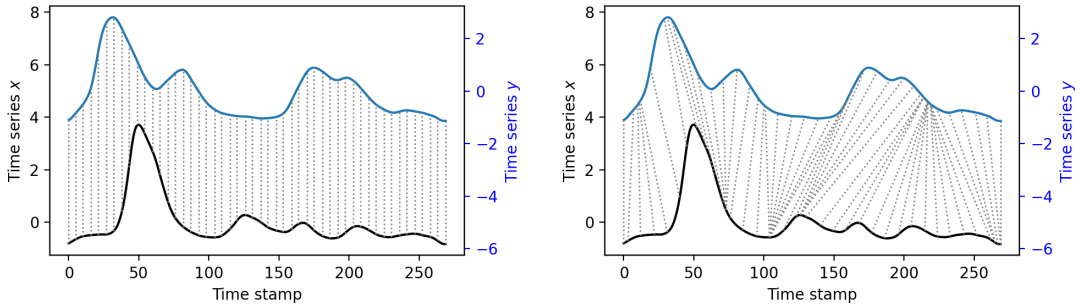
measure [AML19; Shi+23], as defined in Definition III.5. Let us assume that x and y have possibly different lengths $m \neq n$. An elastic measure is robust to time warping, which is a contraction or dilatation of the time axis because it is able to "stretch" or "shrink" [Mar09].

Definition III.5 (Elastic distance measure)**.** *An elastic distance measure is a distance measure D that can compare two time series x and y, possibly of different lengths m and n.*

Contrary to elastic alignments, L_p distances described in Section III.2.1 are called *lock-step alignment* [AML19] due to their one-to-one alignment.

DTW is the most popular elastic distance and has been used in numerous data mining tasks [Shi+23]. Historically, DTW was first used in the speech processing community [SC71; Ita75]. It has also been used in bioinformatics [AC01], health [Cai+98; Ger+22], and entertainment [ZS03], to give a few examples. *1 Nearest-Neighbor (1-NN)* classifier with DTW has long been considered as the traditional benchmark algorithm for time series classification [Bag+17]. Note that DTW can input time series of varying lengths, but there seems to be no significant difference in accuracies between using variable-length time series and equal-length time series (using reinterpolation) in DTW [RK05].

One important feature of an elastic distance measure such as DTW is its robustness to time warping. Contrary to L_p distances, DTW can perform warping, meaning one-to-many alignment between samples of two time series, as illustrated in Figure III.3b. L_p distances compute point-to-point differences between corresponding samples, so they require the timelines between the two signals being compared to have a perfect match. They cannot align two series that are misaligned in the time dimension (even if the signals have the same length).



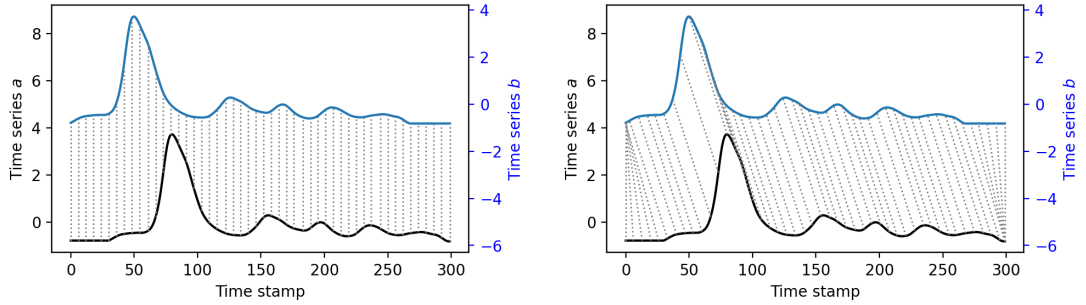
(a) Euclidean distance: one-to-one alignment. Sample x_i is associated with sample y_i.

(b) DTW distance: one-to-many alignment. Sample x_{i_k} is associated with sample y_{j_k}.

Figure III.3: Comparison of the alignments from the Euclidean and DTW distances for the x and y signals depicted in Figure III.2. In the DTW alignment, the first big peaks of each time series are aligned, which makes sense, but not in the Euclidean distance.

Moreover, let us look into a pathological example of bad warping by the Euclidean distance. Let us consider the univariate time series x of length 270 depicted in Figure III.2. Let us denote by a the time series of length 300 obtained from x by padding 30 times to the left the first value of x. Let us denote by b the time series of length 300 obtained from x by padding 30 times to the right the last value of x. These two signals,

along with their Euclidean and DTW alignments, are plotted Figure III.4. By construction, a and b are (almost) the same, but the Euclidean distance would attribute them a distance that is high, due to its imperfect fixed warping, contrary to DTW that is able to recognize that these time series are just shifted on the time axis. Indeed, the large peak is re-aligned by DTW. Moreover, the first value of b and the last value of a both have many alignments, because of the construction of a and b upon x. Hence, often, DTW is more suited than the Euclidean distance (including for equal-length signals) [Bag+17]. However, note that, for nearest neighbor search in large data sets, the Euclidean distance has been shown to be quite equivalent in accuracy to DTW [SK08]. Indeed, the larger a data set, the larger the probability that there is no need for warping for a close match to happen.



(a) The Euclidean distance can not re-align the time series.

(b) DTW can re-align the time series.

Figure III.4: Comparison of the alignments from the Euclidean and DTW distances of two time series a and b that are shifted in time and not synchronized.

An elastic distance measure such as DTW is computed using dynamic programming. *Dynamic programming* simplifies a "complicated" problem by breaking it down into "simpler" sub-problems. Note that dynamic programming is not the same as recursion: while recursion combined with memoization can be viewed as top-down dynamic programming, bottom-up dynamic programming does not involve recursion.

An elastic distance computes the alignment cost between two series that minimizes the cumulative cost of aligning their individual samples [HW21]. This mapping is non-linear. A *cumulative cost matrix* $C \in \mathcal{M}_{m,n}(\mathbb{R})$ stores each intermediate value: $C_{i,j}$ is the minimal cumulative cost of aligning the first i points of x with the first j points of y. As a consequence, the elastic distance measure D is

$$D(x, y) = C_{m,n}. \tag{III.11}$$

III.2.2.1 Dynamic time warping (DTW)

Now, let us describe in detail how DTW works. Let x and y be two time series of respective lengths m and n, with possibly $m \neq n$. As depicted in Figure III.3b, DTW computes a correspondence between the elements of x and those of y using some paths which are defined in Definition III.6.

Definition III.6 (Path for DTW)**.** *A path P is a mapping function*

$$P = ((i_1, j_1), \ldots, (i_{K_P}, j_{K_P})) \in (\mathbb{N} \times \mathbb{N})^{K_P} \quad K_P \in \mathbb{N} \tag{III.12}$$

such that for all $k \in [\![1, K_P]\!]$, $(i_k, j_k) \in P$ *if, and only if,* y_{j_k} *is matched with* x_{i_k}.

The length K_p of a path verifies the following bounds

$$\max\{m, n\} \leq K_P \leq m + n - 1. \tag{III.13}$$

An example of path, the one corresponding to the alignment in Figure III.3b, is given Figure III.5.

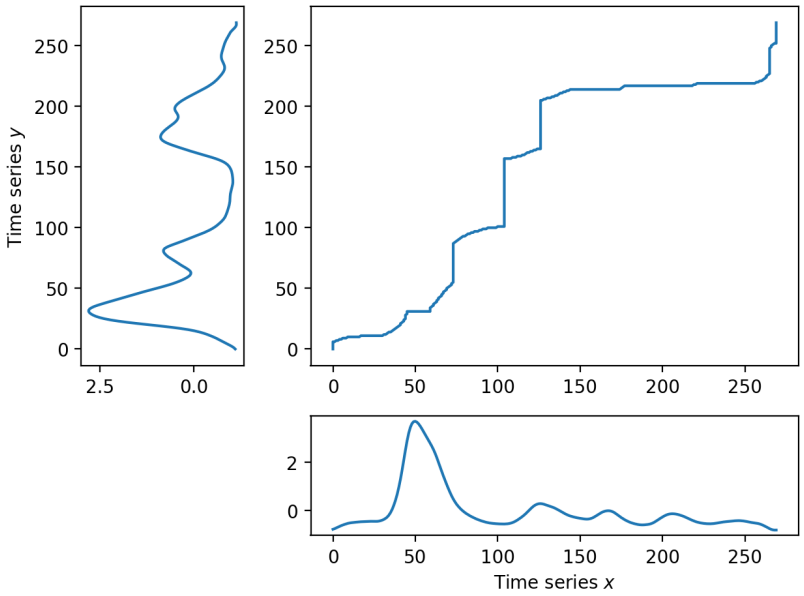


Figure III.5: Optimal warping path between two time series, corresponding to the alignment in Figure III.3b.

A path P is evaluated through the following cost function

$$\gamma_{\text{DTW}}(P) = \sum_{k=1}^{K_P} (x_{i_k} - y_{j_k})^2, \tag{III.14}$$

which is the squared Euclidean distance along the path. Out of the exponential number of possible warping paths, the final DTW distance corresponds to the optimal warping path and is computed as the minimum value for the cost function

$$D_{\text{DTW}}(x, y) = \sqrt{\min_{P \in \mathcal{P}} \gamma_{\text{DTW}}(P)}, \tag{III.15}$$

where $\mathcal{P}$ is the set of acceptable paths respecting the conditions defined in Definition III.7. As depicted in Figure III.7, the optimal path is the one that minimizes the total cost to go from the first time point (bottom left) to the last one (top right). Note that if $m = n = K_P$ and $i_k = j_k = k$ for all $k \in [\![1, K_p]\!]$, DTW is equal to the Euclidean distance (and the path would be the anti-diagonal). DTW is actually a generalization of the Euclidean distance.

Definition III.7 (Set of acceptable paths for DTW)**.** *Given two time series* x *and* y *of lengths* m *and* n*, the set of acceptable paths* $\mathcal{P}$ *must verify the three following conditions:*

1. Continuity

$$i_k - i_{k-1} \leq 1 \quad \textit{and} \quad j_k - j_{k-1} \leq 1 \tag{III.16}$$

At each step, continuity restricts the warping path to adjacent cells: it acts as a step size condition.

2. Monotonicity

$$i_{k-1} \leq i_k \quad \textit{and} \quad j_{k-1} \leq j_k \tag{III.17}$$

The path can only go up $(\uparrow)$ *and right* $(\rightarrow)$*, or diagonally up and right* $(\nearrow)$*: "time can only move forward".*

3. Boundary conditions

$$(i_1, j_1) = (1, 1) \quad \textit{and} \quad (i_{K_P}, j_{K_P}) = (m, n) \tag{III.18}$$

The path starts at the bottom left corner by matching together the first elements of both signals, then finishes at the top right corner by matching together to last elements of x *and* y*.*

Concretely, the three conditions for an acceptable path defined in Definition III.7 impose that, at each iteration, to get to (i_k, j_k), there is a limited set of indexes for the previous step (i_{k-1}, j_{k-1})

$$(i_{k-1}, j_{k-1}) = \begin{cases} & (i_k - 1, j_k) \\ \text{or} & (i_k, j_k - 1) \\ \text{or} & (i_k - 1, j_k - 1) \end{cases} . \tag{III.19}$$

An illustration is provided Figure III.6.

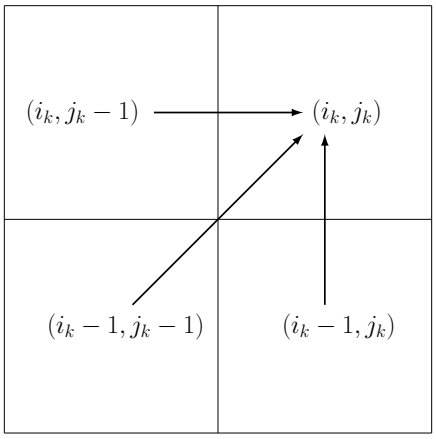


Figure III.6: Set of acceptable paths for DTW at an iteration. As in Figure III.5, signal x is on the x-axis and signal y on the y-axis.

In practice, DTW is solved using dynamic programming. It uses a cumulative cost matrix defined in Definition III.8.

Definition III.8 (Cumulative cost matrix of DTW)**.** *The* cumulative cost matrix $C \in \mathcal{M}_{m,n}(\mathbb{R})$ *of DTW is the dynamic programming cost matrix such that*

$$\forall i \in [\![1, m]\!] \quad C_{i,1} = 0 \tag{III.20}$$

$$\forall j \in [\![1, n]\!] \quad C_{1,j} = 0 \tag{III.21}$$

$$\forall (i,j) \in [\![2, m]\!] \times [\![2, n]\!] \quad C_{i,j} = M^2_{i,j} + \min \begin{cases} C_{i-1,j-1} \\ C_{i,j-1} \\ C_{i-1,j} \end{cases} . \tag{III.22}$$

where $M^2_{i,j}$ *is the squared Euclidean distance between* x_i *and* y_j*. The* point-to-point distance cost matrix $M \in \mathcal{M}_{m,n}(\mathbb{R})$ *is the matrix where each element* $M^2_{i,j}$ *is the cost of pairing* x_i *with* y_j

$$\forall (i,j) \in [\![1, m]\!] \times [\![1, n]\!] \quad M^2_{i,j} = (x_i - y_j)^2 . \tag{III.23}$$

The cumulative cost distance matrix of the two signals in Figures III.2, III.3b, III.5 is given in Figure III.7. For a fixed time stamp of x, one can look into where the path should go next. Contrary to the point-to-point distance cost matrix M which can precomputed, the cumulative cost matrix C is computed step by step. Note that the cumulative cost matrix C enables us to memorize intermediate computations in Formula (III.22), as the computations of $C_{i-1,j}$ and $C_{i,j-1}$ both require the computation of $C_{i-1,j-1}$.

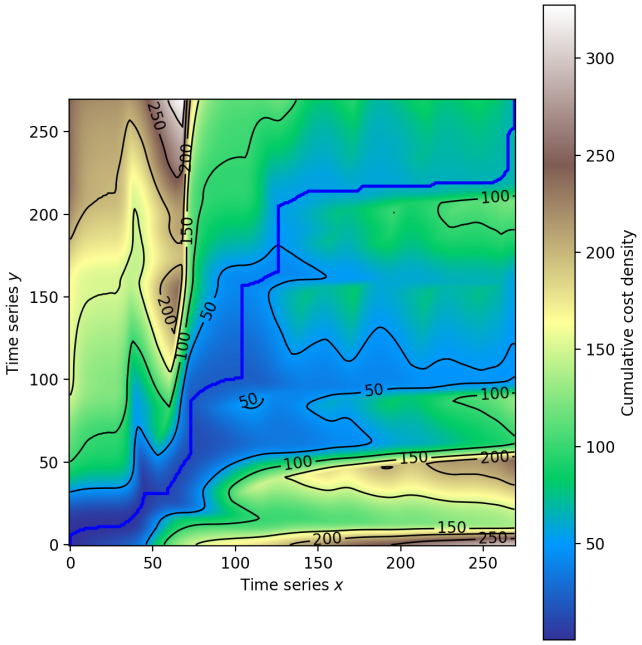


Figure III.7: DTW distance of the two signals in Figures III.2, III.3b, III.5: cumulative cost matrix C with the optimal warping path represented in blue.

The final DTW distance between x and y then corresponds to

$$D_{\text{DTW}}(x, y) = \sqrt{C_{m,n}}. \tag{III.24}$$

The DTW algorithm is recapitulated on Algorithm 1. The point-to-point distance matrix D is first computed. Then, the cumulative cost distance matrix C is computed (based on D).

Algorithm 1: Dynamic Time Warping

Data: Time series $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$
Result: DTW distance $D_{\text{DTW}}(x, y)$

$M \leftarrow 0_{m \times n}$;
$C \leftarrow 0_{m \times n}$;
for $i \leftarrow 1$ **to** m **do**
    for $j \leftarrow 1$ **to** n **do**
        $M(i, j) = (x_i - y_j)^2$;
$C(1, :) = M(1, :)$;
$C(:, 1) = M(:, 1)$;
for $i \leftarrow 2$ **to** m **do**
    for $j \leftarrow 2$ **to** n **do**
        $C(i, j) = M(i, j) + \min\{C(i-1, j-1), C(i-1, j), C(i, j-1)\}$;
$D_{\text{DTW}}(x, y) = \sqrt{C(m, n)}$;
return $D_{\text{DTW}}(x, y)$;

For time series x and y, DTW holds the following properties: $D_{\text{DTW}}(x, y) \geq 0$ (non-negativity) and $D_{\text{DTW}}(x, x) = 0$. However, DTW is not a metric since it does not satisfy the triangular inequality nor the identity.

Although the number of possible ways to align x and y is exponential in m and n, thanks to dynamic programming, DTW is quite efficient with $\mathcal{O}(mn)$. Still, having a quadratic complexity, a point of focus has been trying to optimize it.

In the following, we describe variants of DTW. An outline and taxonomy of these DTW variants is illustrated in Figure III.8.

III.2.3 Penalized variants of DTW

Looking at a cumulative cost density, such as the one in Figure III.7 on page 73, one can make the intuitive observation that, if the timelines of x and y are assumed to be approximately similar, a path would rarely go too far from the diagonal: points that are too far away are unlikely to be aligned. Moreover, DTW can lead to bad alignments where a relatively small part of one time series maps onto a large section of the other one [RK04]. In order to avoid these pathological matchings, some extensions of DTW add a weight to penalize paths that are far from the diagonal. Definition III.9 describes this general framework.

Definition III.9 (Penalized DTW with global or adaptive regions)**.** *For penalized DTW, when creating the point-to-point distance matrix M^2, a weight penalty, denoted $\text{weight}_{i,j}$, is applied. The algorithm of a penalized DTW is the same as unconstrained DTW defined in Section III.2.2.1, except that Formula* (III.23) *is extended into*

$$M^2_{i,j} = \text{weight}_{i,j} \cdot (x_i - y_j)^2 \,. \tag{III.25}$$

Penalization with global regions depend only on the lengths m and n, while penalization with adaptive regions depend on the actual values taken by x and y.

In the case of global regions, the weights are often symmetric, and the weight penalty is then denoted $\text{weight}_{|i-j|}$. There are several ways to set the weight penalty

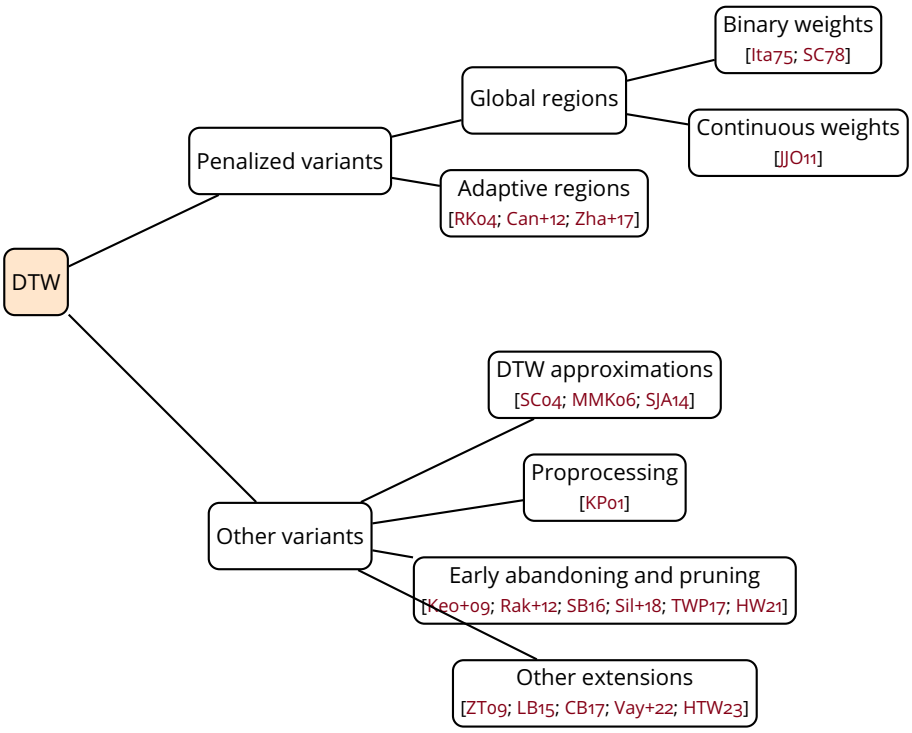


Figure III.8: Outline and taxonomy of DTW variants that are described in the remainder of this section.

in Definition III.9 (see Figure III.8): some weights are binary (discrete) while some are continuous ; some are global while others are adaptive. When the weights are binary (whether global or adaptive), penalized DTW is often called *constrained DTW (CDTW)* [RK04; Fu11; JJO11; HMB23]. Note that the classic DTW described in Section III.2.2.1 uses weights equal to 1 everywhere and is sometimes called *unconstrained DTW* [Gel+19]. Hence, penalized / constrained DTW is a generalization of (unconstrained) DTW.

III.2.3.1 Global regions with binary weights

In the case of binary weights (taking values in $\{1, \infty\}$), when we have $\text{weight}_{|i-j|} = \infty$, the alignment is simply discarded and there is no need to compute term $(x_i - y_j)^2$ in Formula (III.25). Hence, these penalty weights are called "constraints". These constraints actually speed up the computation of DTW as defined in Algorithm 1: it is not necessary (nor recommendable) to compute all values of the cumulative cost distance matrix C. In general, computations far from the diagonal are avoided in order to force paths to be close to the diagonal. One can view these binary penalty weights, which are excluding some alignments, as a fourth condition in Definition III.7 of acceptable paths.

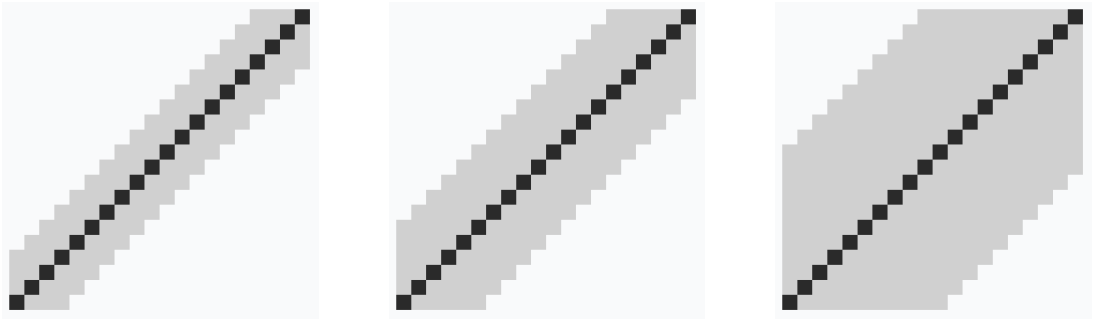
A constrained DTW variant uses the *Sakoe-Chiba band* [SC78] defined in Definition III.10.

Definition III.10 (Sakoe-Chiba band)**.** *The Sakoe-Chiba band sets the weight penalty ac-*

cording to a radius $r \in \mathbb{R}$ *such that*

$$\text{weight}_{|i-j|} = \begin{cases} 1 & \textit{if } |i-j| \leq r \\ \infty & \textit{otherwise} \end{cases} \tag{III.26}$$

In other words, using the Sakoe-Chiba band excludes all computations that are "far" (with a fixed radius r) from the diagonal, directly above or to the right. Figure III.9 shows examples of considered elements for several values of r. When r increases, more indexes become valid. Note that r is also known as the *warping window width* or just *window* [Sil+18; HW21]. The best value of r is data dependent [SB16]. When $r = 0$,

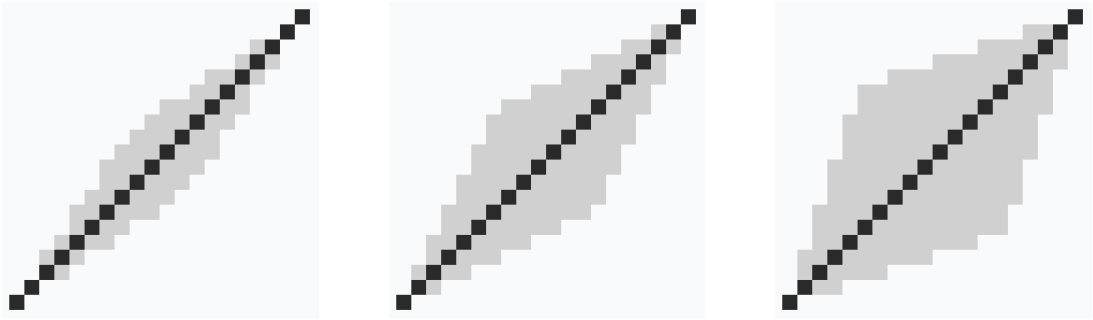


(a) $r = 3$ (b) $r = 5$ (c) $r = 10$

Figure III.9: Visualization of DTW global constraints: Sakoe-Chiba band for several values of radius r, with $m = n = 20$. Valid index pairs are colored. When r increases, more index pairs are valid. Source: [Tav21].

DTW with the Sakoe-Chiba band is the Euclidean distance. When $m = n = r$, DTW with the Sakoe-Chiba band amounts to unconstrained DTW. While most papers have used a Sakoe-Chiba Band with a 10% width, a wider r does not always lead to a better accuracy [RK05]. According to experiments [RK05], there is even a peak in accuracy that occurs at around 4% (on average) which suggests that narrow constraints are better. The best value of r depends upon the data set.

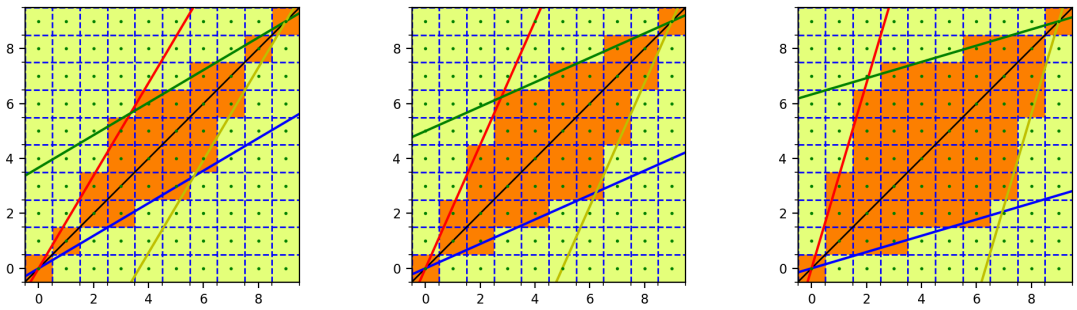
Another band is the *Itakura parallelogram* [Ita75] which sets a maximum slope s for alignment paths, which leads to a parallelogram-shaped constrain as depicted in Figure III.10. Figure III.11 provides more insights into how the Itakura parallelogram is



(a) $s = 1.5$ (b) $s = 2$ (c) $s = 3$

Figure III.10: Visualization of DTW constraints: Itakura parallelogram for several values of maximum slope s, with $m = n = 20$. Valid index pairs are colored. Source: [Tav21].

built: parameter s determines the slope of the steeper side and the slope of the other side is set to $1/s$, passing through the bottom left (start) and the top right (end). The path should not be too steep nor too shallow, so that extremely short subsequences do not match with extremely long ones.



(a) $s = 1.5$ (b) $s = 2$ (c) $s = 3$

Figure III.11: Visualization of DTW constraints: Itakura parallelogram for several values of maximum slope s, with $m = n = 10$. Figures are generated using a `pyts` [FJ20] example.

The Sakoe-Chiba band is more uniform than the Itakura one. A recent paper [Gel+19] compares the Sakoe-Chiba and Itakura bands to unconstrained DTW. It states that the bands not only speed the computation, but also lead to a better accuracy in classification than unconstrained DTW. The Sakoe-Chiba is said to be more accurate than the Itakura band, when averaging scores on 85 real-word data sets from the UCR archive [Dau+19]. However, it concludes with caution: the best band can vary according to the data set at hand.

Note that some lower bounds have been developed on unconstrained DTW or constrained DTW with Sakoe-Chiba or Itakura bands, such as *LB Kim* [KPC01], *LB Keogh* [KR05], *Lower Bounding distance measure with Segmentation (LBS)* [SYF05], and *LB Improved* [Lem09].

III.2.3.2 Global regions with continuous weights

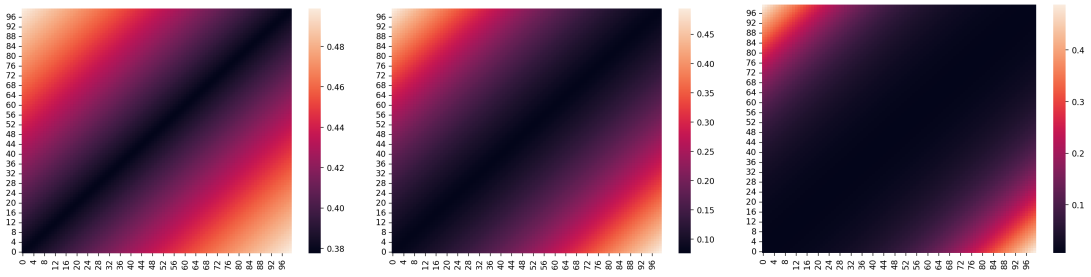
Another variant, known as *Weighted DTW (WDTW)* [JJO11], aims at avoiding large warpings by penalizing them using a non-linear multiplicative weight defined in Definition III.11. WDTW is not exactly a constrained extension of DTW, but a penalized extension, as constrained alignments are not plainly forbidden.

Definition III.11 (Weighted DTW)**.** *Weighted DTW (WDTW) penalizes large warpings by applying a non-linear weight to the warpings using the modified logistic function*

$$\text{weight}_{|i-j|} = \frac{\text{weight}_{max}}{1 + \exp\left(-g_{WDTW} \cdot \left(\frac{|i-j|-n}{2}\right)\right)} \tag{III.27}$$

where $|i-j|$ is the phase difference, weight_{max} is the upper bound on the weight (set to 1), n is the series length, and g_{WDTW} is the parameter that controls the penalty level for large warpings.

According to [JJO11], g_{WDTW} is usually chosen in $[0.01, 0.6]$ and its best value is data dependent. Smaller values of g_{WDTW} result in less penalty for further points in the sequence (meaning large values of $|i-j|$), thus WDTW behaves similarly to DTW. When $g_{\text{WDTW}} = 0$, the weight is constant: all the points have the same weight, and WDTW is classic (unpenalized) DTW. Larger values of g_{WDTW} impose higher penalty for further points, leading to a similar behavior to Euclidean distance. As can be observed in Figure III.12 and confirming Formula (III.27), larger values of g_{WDTW} increase the penalty for further points (relatively to closer points).



(a) $g_{\text{WDTW}} = 0.01$ (b) $g_{\text{WDTW}} = 0.05$ (c) $g_{\text{WDTW}} = 0.1$

Figure III.12: Visualization of the weighted penalty $\text{weight}_{|i-j|}$ in WDTW with $m = n = 100$. Lighter region indicates matching indexes that are more penalized. Note that for each figure, the range of the color bar changes.

III.2.3.3 Adaptive regions

Contrary to previously described global constraints (Sakoe-Chiba band, Itakura band, and WDTW), *adaptive regions* depend on the actual values taken by x and y, and not only on their lengths. Indeed, the optimal warping path may leave the specified global region.

The *Ratanamahatana-Keogh band (RK-band)* [RK04] is a generalization of the Sakoe-Chiba and Itakura bands, as it finds the optimal width and shape of the constrained band. As wider bands to do not always result in an increased accuracy, the R-K band is automatically learned from the data, using heuristic search algorithms. The R-K band offers a practical balance between the Sakoe-Chiba and Itakura bands, as they each have their specific applications. For instance, when dealing with speech recognition tasks, where most variations occur in the middle rather than at the beginning or end, the Itakura band is more suitable. The R-K band does not only aim to speed up DTW, but also to make it more accurate [RK05]. Contrary to globals regions, the R-K band has no reason to be symmetric.

Salient feature based DTW (sDTW) [Can+12] identifies salient temporal features in the time series in order to help the search of the optimal warping path. These features are robust to noise and are similar to *Scale-Invariant Feature Transform (SIFT)* [Low04] used in computer vision. They are used to match salient feature points from the two time series, on which the optimal warping path then relies on. The Sakoe-Chiba band is a particular case of sDTW.

DTW with Limited warping path Length (LDTW) [Zha+17] introduces an upper bound L_{UB} on the warping path length in order to avoid singularities. A *singularity* is defined as a data point from a time series that is matched with a large subsection of the other

time series, thus leading to pathological alignments. In a supervised learning setting, the best upper bound L_{UB} of warping path lengths is learned from the training set.

III.2.4 Other variants of DTW

III.2.4.1 DTW approximations

A *DTW approximation* finds an approximation of the optimal warping path of DTW: it favors speed over accuracy. According to [Sil+18], DTW approximations do not provide bounds for the approximation error.

FastDTW [SC04] is a DTW approximation, with linear time complexity thanks to a multilevel strategy. The time series are first down-sampled (*coarsening*) and an optimal warping path is found on this lower resolution. Next, this warping path is projected into a higher resolution (*projection*). Then, the optimal path in the neighborhood of the projected path is found (*refinement*). The considered neighborhood is controlled by a radius parameter. This multi-level process continues until a warping path is identified at the original resolution of the time series.

A similar approach is *Multiscale DTW (MsDTW)* [MMK06]. MsDTW states that a limitation of FastDTW is that an incorrect alignment on a low resolution level becomes increasingly inaccurate as it propagates to higher levels. Contrary to FastDTW, MsDTW iteratively combines global constraints with the multilevel strategy. The difference with FastDTW is that the projected path from a lower level is used to form a global constraint region in order to find the optimal warping path. MsDTW was first applied to music synchronization: aligning the note events of different interpretations of a same music.

Lucky Time Warping (LTW) [SJA14] is a DTW approximation with linear time complexity. It uses a greedy algorithm to accelerate the distance calculations: it only evaluates elements which are the most likely to be in the optimal warping path, resulting in a suboptimal warping path. It is faster than DTW, but less accurate in nearest neighbor classification.

III.2.4.2 Preprocessing

Some preprocessing can be done to transform the time series before feeding them to DTW (whether constrained or not).

Normalization As for the L_p distance, a commonly used preprocessing involves normalization. Normalization is used in order to improve the robustness to changes in offset (amplitude). Note that normalization may increase the sensitivity with respect to additive noise in time series. The need for normalization before measuring the distance between time series is described in [KK03]. z-normalization, where a signal is centered and scaled to unit variance, is the most popular.

Derivative Another preprocessing involves the derivative. One such variant is *Derivative DTW (DDTW)* [KP01] which applies DTW, not directly on the raw signals, but on their first derivative. The goal is to prevent unnatural warpings when there is variability in the signals. In DDTW, the derivative transformation x'_i of a univariate time

point x_i is defined as

$$x'_i = \frac{x_i - x_{i-1} + \frac{x_{i+1} - x_{i-1}}{2}}{2}, \tag{III.28}$$

where the first and last element of the signal are not defined. x'_i is the average of the slopes of the line passing through x_{i-1} and x_i, and of the line passing through x_{i-1} and x_{i+1}, and is considered more robust than an estimation using only two data points. The time complexity of DDTW is $\mathcal{O}(mn)$, same as DTW. When both Derivative DTW and Weighted DTW are combined, the variant is referred to as *Weighted Derivative DTW (WDDTW)* [JJO11].

III.2.4.3 Early abandoning and pruning extensions

Early abandoning and pruning both aim at making computations involving distances faster, but their strategies differ from the constrained bands described in Section III.2.3.

As stated in [HW21], *early abandoning* is the strategy that abandons a whole computation once it has been determined, through an "abandoning criterion", that an exact result is not necessary. Early abandoning is also known as "early stopping" [SYF05]. For example, the lower-bounding property used for nearest neighbor search, described in the introduction of this chapter, is a typical example of early abandoning: the whole computation between two time series $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_n)$ is not done. As a consequence, early abandoning does not compute an exact similarity score (as it stops when it has a partial score such as the lower bound). *Time Series Indexing (TSI)* [TWP17] combines priority search in a hierarchy of K-means clusterings with lower-bounding of DTW in order to index and classify trillions of satellite image time series. Given that DTW is not a metric, indexing it is challenging, as explained in the introduction of this chapter.

Similar yet different, *pruning* aims at identifying and avoiding unnecessary computations [HW21]. Most recent advances aiming at accelerating the DTW computation have focused on similarity search. However, data mining tasks such as classification and clustering require the full pairwise distance matrix which early abandoning does not provide. Pruning computes the exact similarity score of the full pairwise matrix: the computation itself of the distance between x and y is improved. Pruning is applied to constrained DTW with *PrunedDTW* [SB16]. When iteratively computing the cumulative cost matrix C of DTW, if C_{i_k,j_k} has a high value (according to a threshold), then the path passing through position (i_k, j_k) would probably not be part of the optimal full path. Hence, the pairwise distances $M_{i,j} = (x_i - y_j)^2$ of paths going through (i_k, j_k) do not need to be computed. Note that only partial computations $(x_i - y_j)^2$ are avoided, but not the whole computation $D_{\text{DTW}}(x, y)$ as in early abandoning. An improvement of PrunedDTW is suggested in [Sil+18].

The *UCR Suite* [Rak+12] introduces a set of accelerations, mainly lower-bounding and pruning methods, making subsequence similarity search using DTW faster than the Euclidean distance. [Keo+09] focuses on fast rotation-invariant search. *EAPruned (Early Abandoning and Pruned)* [HW21] combines both pruning and early abandoning. EAPruned can be applied to DTW, but also other elastic distances: unconstrained DTW, WDTW, ERP, MSM, and TWE (that will described in Section III.3.4).

III.2.4.4 Other extensions

DTW$^+$ and *ADTW*$^+$ [HTW23] modify the cost function for the point-to-point distance $M_{i,j}$ in DTW that was defined in Definition III.15 on page 71. They introduce a family of cost functions $c_\tau(x, y) = |x - y|^\tau$ with a parameter τ to be tuned for DTW, for time series classification. It claims that tuning τ improves the classification accuracy for 1-NN DTW and another popular classifier called Proximity Forest [Luc+19].

Canonical Time Warping (CTW) [ZT09] performs spatio-temporal alignment of times by combining Canonical *Correlation Analysis (CCA)* [SCF06] with DTW. CCA is a technique to extract common features from two sets of multivariate data. Originally, CTW addresses the issue of large temporal scale difference between humans actions and inter/intra subject variability, for example when aligning motion capture data. The original paper [ZT09] also introduces *Local Canonical Time Warping (LCTW)* that allows several local spatial deformations (in order to align long sequences). In addition to being able to compare time series of varying lengths, CTW can also compare time series of different dimensionality.

soft-DTW [CB17] is a differentiable extension of DTW: it uses a smoothed formulation of DTW with the soft-min operator. A limit of classic DTW is that it can not be differentiated everywhere because of its use of the min operator. For a distance, being differentiable is important in order to be used as a loss function in gradient-based optimization that is paramount in machine learning tasks. According to [JCG20], contrary to classic DTW, soft-DTW is not invariant to time-shifts. Some variants of soft-DTW have been proposed [JCG20; BMV21].

DTW with Global Invariances (DTW-GI) [Vay+22] aligns time series jointly in the temporal and feature spaces, and thus addresses two sources of variability that are commonly encountered when dealing with time series: time-shifts and distribution-shifts. The latter, feature space alterations, occur for example when sensors are switched during a protocol. While DTW is invariant to time-shifts, (soft)DTW-GI can handle both types of shifts thanks to a joint optimization formulation that can be extended for soft-DTW.

Some ensemble distances are based on the previously described elastic distance measures and also others that will be described in Section III.3.4. Indeed, using a diversity of distance measures for 1-NN classification is significantly more accurate than 1-NN with any single measure [LB15]. Ensemble methods help to reduce the variance of the model. *Elastic Ensemble (EE)* [LB15] combines eleven elastic measures (to be applied to 1-NN algorithms): Euclidean, DTWF (with full window), DTW (with leave-one-out cross-validated window), DDTWF, DDTW, WDTW, WDDTW, LCSS, ERP, MSM, and TWE. For each measure, EE fine tunes their parameters using cross validation. While being a relatively accurate classifier [Bag+17], EE is quite slow to train: its time complexity is $\mathcal{O}\left(pN^2n^2\right)$ where N is the number of time series which are of length n, and p is the total number of parameters.

III.3 Distance measures on strings

Let us assume that we want to compare two strings denoted by $\mathtt{x}$ and $\mathtt{y}$, of respective lengths m and n. In the string matching community, we are interested in *approximate string matching* which addresses string matching while allowing errors [Nav01]. For example, an error in a text is a typing or spelling error. We refer an interested reader

to [Kru83; Kuk92; WM92; Nav01; Shi+23] for extensive reviews on distances between strings. Note that some extensions for multivariate texts exist [Nav01; NBY99], but they are not described in this thesis.

III.3.1 The general edit distance framework

A popular distance measure is the edit distance: for two strings $\mathtt{x}$ and $\mathtt{y}$, it is the minimal cost of a sequence of elementary operations that transform $\mathtt{x}$ into $\mathtt{y}$. The allowed elementary operations, each one called an *edit operation* [YB07], are the following:

1. *Insertion* [Kru83; Nav01] of an elementary character in a string.

   For example, insert $\mathtt{d}$ (at the last position):

   $$\mathtt{abc} \rightarrow \mathtt{abcd} \tag{III.29}$$

2. *Deletion* [Kru83; Nav01] of an elementary character in a string.

   For example, delete $\mathtt{b}$:

   $$\mathtt{abc} \rightarrow \mathtt{ac} \tag{III.30}$$

3. *Substitution* [Kru83; Nav01] (a.k.a replacement [Kru83; Nav01] or mutation [BR02; Pin+13]) of elementary characters (with a different one) in both strings.

   For example, substitute $\mathtt{b}$ by $\mathtt{d}$ in the following string:

   $$\mathtt{abc} \rightarrow \mathtt{adc} \tag{III.31}$$

4. *Transposition* [Kru83; Nav01] (a.k.a. swapping [Kru83]): substitution of the form $\mathtt{ab} \rightarrow \mathtt{ba}$

   This operation is particularly interesting in the case of typing errors.

5. *Duplication* [Pin+13] (a.k.a. amplification [BR02] or expansion [Kru83])

   For example, amplify $\mathtt{b}$:

   $$\mathtt{abc} \rightarrow \mathtt{abbc} \tag{III.32}$$

6. *Contraction* [Pin+13] (a.k.a. compression [Kru83])

   For example, contract $\mathtt{b}$:

   $$\mathtt{abbc} \rightarrow \mathtt{abc} \tag{III.33}$$

Note that, in [Kru83], the expansion operation amplifies one element into two *or more*, and the compression operation contracts two *or more* elements into one. Moreover, in [Kru83], *indel* covers either *in*sertion or *del*etion.

To each edit operation corresponds a cost, and this cost also depends on the characters involved. The cost of a sequence of operations is the sum of the costs of the edit operations. An edit distance can compare strings of different lengths, if it allows for insertions, deletions, duplications, or contractions. Otherwise, it can only compare equal-length strings.

If all the elementary operations have a cost of 1, whatever the operation or the characters involved, it is called the *simple edit distance*. If the all authorized operations

have different costs and/or the costs depend on the characters involved, it is called the *general edit distance*.

Note that some of the edit operations can be obtained through a combination of the others. For example, a transposition can be seen as an insertion followed by a deletion. However, the main difference is the cost of the total operation. In the case of the simple edit distance, a transposition has a cost of 1, while insertion followed by deletion has a cost of 2. Hence, allowing transpositions reduces the impact of swapping errors. The same idea applies for substitution which can be viewed as an insertion followed by a deletion.

In addition, the main difference between insertion and duplication (and between deletion and contraction) is that, in duplication, the operation cost depends on the current character but also its adjacent ones. For example, inserting $\mathtt{a}$ at the middle of $\mathtt{aa}$ costs less than inserting $\mathtt{b}$ at the middle of $\mathtt{aa}$. On the other hand, an insertion can be viewed as a duplication followed by a substitution. Hence, duplications or contractions can allow an edit distance to be invariant to translations.

Let us formalize the general edit distance framework [YB07]. An elementary edit operation is written as $\mathtt{a} \to \mathtt{b}$ where $(\mathtt{a}, \mathtt{b}) \neq (\varnothing, \varnothing)$. γ_{ed} is the weight function which gives the cost of an edit operation $\mathtt{a} \to \mathtt{b}$. The forms $\varnothing \to \mathtt{a}$, $\mathtt{a} \to \mathtt{b}$, and $\mathtt{b} \to \varnothing$ respectively, represent insertions, substitutions, and deletions. The forms $\mathtt{ab} \to \mathtt{ba}$, $\mathtt{a} \to \mathtt{aa}$, and $\mathtt{bb} \to \mathtt{b}$ respectively, represent transpositions, duplications, and contractions. $T_{\mathtt{x},\mathtt{y}} = T_1 \circ T_2 \circ \ldots \circ T_{K_p}$ is the edit transformation of $\mathtt{x}$ into $\mathtt{y}$: it is a sequence of elementary edit operations transforming $\mathtt{x}$ into $\mathtt{y}$. Hence, an edit distance D_{ed} can be defined as

$$D_{\mathsf{ed}}(\mathtt{x}, \mathtt{y}) = \min \left\{ \gamma_{\mathsf{ed}}(T_{\mathtt{x},\mathtt{y}}) \right\}. \tag{III.34}$$

A parallel can be drawn with DTW, described in Section III.2.2, with the notion of path and a dynamic programming algorithm.

III.3.2 The various edit distances

Based on the general edit distance framework presented in Section III.3.1, several edit distances have been defined over the years, each one allowing a certain set of edit operations out of the 6 that are described in Section III.3.1. A summary of their definition is presented in Table III.3 on page 94, along with some properties. They are further described in the next paragraphs.

As the Levenshtein distance (an edit distance which will be described in Section III.3.2.1) is widely-used, we will describe it with further details compared to the other variants, as an illustrative example. In particular, we will study its simple version (where all costs are set to 1) as well as its general version where the costs are not uniform. For the other variants, we will only study their simple version.

III.3.2.1 Levenshtein distance

One of the most popular edit distance is the *Levenshtein distance* [Lev+66] which only allows insertions, deletions, and substitutions. Note that sometimes, the Levenshtein distance is simply referred to as the edit distance. Originally, its main application is to check for spelling errors.

The simple Levenshtein distance is the minimum number of insertions, deletions and substitutions to make both strings equal. In that case, insertions on $\mathtt{x}$ are the same as deletions in $\mathtt{y}$, and substitutions can be made in $\mathtt{x}$ or $\mathtt{y}$.

Similarly to DTW, the Levenshtein distance is solved using dynamic programming. The Levenshtein distance between two strings $\mathtt{x}$ and $\mathtt{y}$ of respective lengths m and n is defined by

$$\forall i \in [\![1, m]\!] \quad C_{i,1} = i \tag{III.35}$$

$$\forall j \in [\![1, n]\!] \quad C_{1,j} = j \tag{III.36}$$

$$\forall (i, j) \in [\![2, m]\!] \times [\![2, n]\!] \quad C_{i,j} = \min \begin{cases} C_{i-1,j-1} + W_{sub}(i, j) \\ C_{i,j-1} + W_{ins}(j) \\ C_{i-1,j} + W_{del}(i) \end{cases} \tag{III.37}$$

$$D_{\mathsf{gLev}}(\mathtt{x}, \mathtt{y}) = C(m, n) \tag{III.38}$$

Algorithm 2 details the implementation of the general edit distance. Comparing Algorithm 2 to Algorithm 1, one can observe that the general edit distance is the equivalent of DTW on strings and that their optimization problems are similar, both using dynamic programming and a warping path. The time complexity to compare two strings of lengths m and n is $\mathcal{O}(mn)$.

Algorithm 2: General Levenshtein distance, with $\mathcal{A}$ the alphabet of size A.

Data: Strings $\mathtt{x} \in \mathcal{A}^m$ and $\mathtt{y} \in \mathcal{A}^n$; deletion costs $W_{del} \in \mathbb{R}^A$, insertion costs $W_{ins} \in \mathbb{R}^A$, substitution costs $W_{sub} \in \mathbb{R}^{A \times A}$

Result: General Levenshtein distance $D_{\mathsf{gLev}}(\mathtt{x}, \mathtt{y})$

$C \leftarrow 0_{m \times n}$;
for $i \leftarrow 1$ **to** m **do**
  $C(i, 1) = i$;
for $j \leftarrow 1$ **to** n **do**
  $C(1, j) = j$;
for $i \leftarrow 2$ **to** m **do**
  for $j \leftarrow 2$ **to** n **do**
    $C(i, j) = \min \begin{cases} C(i, j-1) + W_{ins}(j) \\ C(i-1, j) + W_{del}(i) \\ C(i-1, j-1) + W_{sub}(i, j) \end{cases}$;
return $C(m, n)$;

The simple Levenshtein distance D_{sLev} satisfies all four fundamental properties of a metric. Hence, the simple Levenshtein distance is a metric. Moreover, we have a simple upper-bound

$$0 \leq D_{\mathsf{sLev}}(\mathtt{x}, \mathtt{y}) \leq \max(m, n) \tag{III.39}$$

for all strings $\mathtt{x}$ and $\mathtt{y}$. Note that some tighter bounds are described in [Navo1].

According to [YBo7], it has been shown that the general edit distance is a metric if the following conditions on the costs are satisfied:
$\forall \mathtt{a}, \mathtt{b} \in \mathcal{A} \cup \{\varnothing\}$,

- $\gamma_{\text{ed}}(\mathtt{a} \to \mathtt{a}) = 0$,
- $\gamma_{\text{ed}}(\mathtt{a} \to \mathtt{b}) > 0$ if $\mathtt{a} \neq \mathtt{b}$,
- $\gamma_{\text{ed}}(\mathtt{a} \to \mathtt{b}) = \gamma_{\text{ed}}(\mathtt{b} \to \mathtt{a})$.

III.3.2.2 Other edit distances

The *Longest Common SubSequence (LCSS)* [NW70; Hir77; AG87] allows only insertions and deletions. LCSS is widely used as it measures the length of the longest pairing of characters that can be between both strings, so that the pairings respect the order of the letters. The distance is the number of unpaired characters. The distance is symmetric, and it holds

$$0 \leq D_{\text{LCSS}}(\mathtt{x}, \mathtt{y}) \leq m + n \tag{III.40}$$

The *Hamming distance* [SM83] allows only substitutions. It can only be applied to strings of the same length $m = n$. The distance is symmetric. It holds

$$0 \leq D_{\text{Hamming}}(\mathtt{x}, \mathtt{y}) \leq m. \tag{III.41}$$

The *Episode distance* [Das+97] allows only insertions. It models the case where a sequence of events is sought, where all of them must occur within a short period. This distance is not symmetric. Note that it may not be possible to convert $\mathtt{x}$ (of length m) into $\mathtt{y}$ (of length n) in this case. Hence, $d_{\text{Episode}}(\mathtt{x}, \mathtt{y})$ is either $(n - m)$ or ∞.

Compared to the Levenshtein distance, the *Damerau–Levenshtein distance* [Dam64; Lev+66] adds the transposition operation. Its main application is spelling error correction. The Lowrance and Wagner algorithm [WL75] has a complexity of $\mathcal{O}(mn)$. According to [Bar07], it is a metric. Compared to the Levenshtein distance, the *Edit Distance with Duplications and Contractions (EDDC)* [BR02; Pin+13] adds the duplication and contraction operations. The *Jaro–Winkler distance* [Win90] allows only transposition. It is not a metric because it does not satisfy the triangle inequality.

Edit distances are used in bioinformatics: many algorithms are used to align DNA sequences of nucleotides, meaning strings composed of the letters `A`, `C`, `G`, and `T`. These algorithms often use the edit distance as a score. Exhaustive searches look for all possible alignments and retrieve the alignment(s) with the optimal score, which is very costly. An alternative is the *Needleman-Wunsch algorithm* [NW70] which uses dynamic programming for global sequence alignment. It determines the optimal alignment of all possible prefixes of the first sequence with all possible prefixes of the second sequence by going from the smallest to the largest prefixes. The *Smith and Waterman algorithm* [SW81] is a variation of the Needleman-Wunsch algorithm which performs local sequence alignment.

III.3.3 Normalization

Let us take the example of the Levenshtein distance. It lacks normalization with respect to the lengths of the compared strings. It is intuitive that errors occurring when comparing short strings are more crucial than when comparing long strings. Hence, according the the data mining task at hand, normalizing the Levenshtein distance can be important.

There exists several ways to normalize the Levenshtein distance [MV93; WF94]. They are based on the editing path lengths or the string lengths, but they do not verify the triangle inequality. The *Normalized Levenshtein Distance Metric* [YB07] is a normalized Levenshtein distance that is also a valid metric valued in $[0, 1]$ (under some conditions on the costs of the edit operations).

III.3.4 Extensions of edit distances to time series

Some previously described edit distances, originally defined on strings, have been extended to input real-valued time series [Shi+23], mainly thanks to thresholds. We will only describe the univariate cases, as the multivariate setting will be covered in Section III.5.

III.3.4.1 Longest Common SubSequence (LCSS)

As described in Section III.3.2.2, LCSS was originally defined on strings [Hir77]. It has then been extended to real-valued time series [VKG02] thanks to a threshold $\varepsilon \in \mathbb{R}$. Compared to LCSS on strings, two real values x_i and y_j are considered a match if

$$L_2\left(x_i, y_j\right) = |x_i - y_j| \leq \varepsilon. \tag{III.42}$$

The relaxed version of LCSS for real-valued signals is the following

$$C_{\text{LCSS}}(i,j) = \begin{cases} 0 & \text{if } i = 0 \\ 0 & \text{if } j = 0 \\ 1 + C_{\text{LCSS}}(i-1, j-1) & \text{if } L_2\left(x_i, y_j\right) < \varepsilon \\ \max\left(C_{\text{LCSS}}(i-1,j), C_{\text{LCSS}}(i,j-1)\right) & \text{otherwise} \end{cases}. \tag{III.43}$$

Then, we have

$$D_{\text{LCSS}}(x,y) = C_{\text{LCSS}}(m,n). \tag{III.44}$$

LCSS has a greater robustness against noise compared to DTW, as it allows certain elements within the time series to remain unmatched, all while preserving the matching order. LCSS verifies the non-negativity and symmetry properties, and $D_{\text{LCSS}}(x,x) = 0$ for all time series x. However, LCSS does not verify the triangle inequality, and thus is not a metric.

III.3.4.2 Edit distance with Real Penalty (ERP)

Edit distance with Real Penalty (ERP) [CN04] and *Edit Distance on Real sequence (EDR)* [COO05] were both introduced around the same time (with a common co-author in both). They are based on the Levenshtein distance on strings [Lev+66] (that allows insertions, deletions, and substitutions). In EDR, a threshold is used to consider a match, similarly to LCSS, but the triangle inequality is not respected. On the contrary, ERP is a metric. In the following, we focus on ERP rather than EDR.

Rather than using a delete operation, EDR considers a deletion in a time series (e.g. x) as a special symbol in another series (e.g. y). ERP calls it a *gap* element and its penalty parameter is β_{ERP}. According to [COO05], ERP is sensitive to noise. In ERP,

the Euclidean distance between elements is employed when there is no gap present, while a constant penalty is applied in cases where a gap exists:

$$C_{\mathsf{ERP}}(i,j) = \min \begin{cases} C_{\mathsf{ERP}}(i-1,j-1) + L_2\left(x_i, y_j\right)^2 \\ C_{\mathsf{ERP}}(i-1,j) + L_2\left(x_i, \beta_{\mathsf{ERP}}\right)^2 \\ C_{\mathsf{ERP}}(i,j-1) + L_2\left(y_j, \beta_{\mathsf{ERP}}\right)^2 \end{cases} . \tag{III.45}$$

Then, we have

$$D_{\mathsf{ERP}}(x,y) = C_{\mathsf{ERP}}(m,n). \tag{III.46}$$

Hence, the cost of insertion or deletion depends on the absolute magnitude of the value that is inserted or delete.

III.3.4.3 Move-split-merge (MSM)

Move-split-merge (MSM) [SAD13] is inspired by edit distances on strings. It verifies the properties of a metric. MSM states that, contrary to ERP, it has the particularity of being invariant to translations. It allows three operations: move, split, and merge. Actually, these three operations correspond to edit operations on strings described in Section III.3.1: move is equivalent to substitution, split to duplication, and merge to contraction.

The cost associated to a substitution is set by the pairwise distance between two points, and the cost of a duplication or a contraction depends on a parameter denoted by β_{MSM}

$$C_{\mathsf{MSM}}(i,j) = \min \begin{cases} C_{\mathsf{MSM}}(i-1,j-1) + L_2\left(x_i, y_j\right) \\ C_{\mathsf{MSM}}(i-1,j) + W_{\mathsf{MSM}}\left(x_i, x_{i-1}, y_j, \beta_{\mathsf{MSM}}\right) \\ C_{\mathsf{MSM}}(i,j-1) + W_{\mathsf{MSM}}\left(y_j, x_i, y_{j-1}, \beta_{\mathsf{MSM}}\right) \end{cases} , \tag{III.47}$$

where

$$W_{\mathsf{MSM}}\left(x_i, x_{i-1}, y_j, \beta_{\mathsf{MSM}}\right) = \begin{cases} \beta_{\mathsf{MSM}} & \text{if } x_{i-1} \leq x_i \leq y_j \\ \beta_{\mathsf{MSM}} & \text{if } x_{i-1} \geq x_i \geq y_j \\ \beta_{\mathsf{MSM}} + \min \begin{cases} |x_i - x_{i-1}| \\ |x_i - y_j| \end{cases} & \text{otherwise} \end{cases} . \tag{III.48}$$

The algorithm contracts two values or duplicates a value x_i if x_i is between two adjacent values (x_{i-1} and y_j). If a value x_i falls two consecutive values (x_{i-1} and y_j), the algorithm either performs contraction or duplication.

Then, we have:

$$D_{\mathsf{MSM}}(x,y) = C_{\mathsf{MSM}}(m,n). \tag{III.49}$$

III.3.4.4 Time Warp Edit (TWE)

Time Warp Edit Distances (TWED) [Mar09] is based on the edit distance on strings, however it has no straightforward equivalent on strings. Indeed, TWE combines (non-elastic) L_p norms with the (elastic) edit distance. TWED is also referred to as *TWE* [Shi+23] and is a metric.

TWE allows three operations called $match$, $delete_x$, and $delete_y$. When there is a match, the L_p distance is used, otherwise a constant penalty is added. The $delete_x$ (or $delete_y$) operation is used to remove an element from x (or y) to match y (or x).

The TWE dynamic programming algorithm is thus

$$C_{\text{TWE}}(i,j) = \min \begin{cases} C_{\text{TWE}}(i-1,j-1) + \gamma_M & match \\ C_{\text{TWE}}(i-1,j) + \gamma_x & delete_x \\ C_{\text{TWE}}(i,j-1) + \gamma_y & delete_y \end{cases}, \tag{III.50}$$

where

$$\begin{aligned} \gamma_M &= L_2\left(x_i, y_j\right)^2 + L_2\left(x_{i-1}, y_{j-1}\right)^2 + 2\nu && match \\ \gamma_x &= L_2\left(x_i, x_{i-1}\right)^2 + \nu + \beta_{\text{TWE}} && delete_x \\ \gamma_y &= L_2\left(y_j, y_{j-1}\right)^2 + \nu + \beta_{\text{TWE}} && delete_y \end{aligned} \tag{III.51}$$

with

- ν, the *stiffness* parameter, controls the elasticity of TWE. When $\nu = 0$, TWE is stiff, similarly to the L_p distance. When ν approaches infinity, TWE becomes less stiff and more elastic, similarly to DTW.
- β_{TWE} is the cost of a delete operation, either $delete_x$ or $delete_y$.

Then, we have

$$D_{\text{TWE}}(x,y) = C_{\text{TWE}}(m,n). \tag{III.52}$$

III.4 Distance measures on symbolic sequences

In Chapter II, we reviewed symbolic representations on time series: they transform real-valued time series into discrete-valued ones called symbolic sequences. In order to use the learned symbolic representations for tasks such as classification or clustering, it is crucial to define a distance measure between symbolic sequences, which can be viewed as character strings. Defining an informative measure is a challenge that has received a lot of attention. In this section, we described distance measures defined on symbolic sequences obtained from symbolization processes that were described in Chapter II. Note that symbolic representations do not systematically define a distance measure on their symbolic sequences.

III.4.1 MINDIST

The most popular distance measure on symbolic sequences is the one introduced along with the SAX representation [Lin+03; Lin+07]: *MINDIST*. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two time series with n samples. The Euclidean distance between x and y, described in Section III.2.1, is given by

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{III.53}$$

The MINDIST distance measure between the resulting symbolic sequences $\hat{x}$ and $\hat{y}$, of lengths w, mimics the Euclidean distance

$$D_{\text{MINDIST}}\left(\hat{x}, \hat{y}\right) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(\text{dist}\left(\hat{x}_i, \hat{y}_i\right)\right)^2}, \tag{III.54}$$

where the dist function, based on a so-called look-up table, is illustrated in Table III.1. MINDIST requires the symbolic sequences to be of equal length. For a given value of

Table III.1: Example of look-up table for MINDIST with $A = 4$. For example, $\text{dist}(\texttt{a}, \texttt{d}) = \beta_3 - \beta_1 = 1.34$.

	a	b	c	d
a	0	0	$\beta_2 - \beta_1$	$\beta_3 - \beta_1$
b	0	0	0	$\beta_3 - \beta_2$
c	$\beta_2 - \beta_1$	0	0	0
d	$\beta_3 - \beta_1$	$\beta_3 - \beta_2$	0	0

	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

the alphabet size A, this table is calculated only once, and then stored for fast look-up. For all look-up tables, whatever the alphabet size, the value in the cell of indexes (i, j) is given by

$$\text{cell}_{i,j} = \begin{cases} 0 & \text{if } |i - j| \leq 1 \\ \beta_{\max(i,j)-1} - \beta_{\min(i,j)} & \text{otherwise} \end{cases}, \tag{III.55}$$

where the β_k are the boundaries of the bins used by SAX to quantize the segment means. MINDIST is not a true metric, as $\text{dist}(\texttt{a}, \texttt{b}) = 0$ for example (see Table III.1). The MINDIST distance measure on SAX symbolic sequences lower bounds the Euclidean distance on original signals, i.e.,

$$D_{\text{MINDIST}}\left(\hat{x}, \hat{y}\right) \leq L_2(x, y). \tag{III.56}$$

As emphasized in the introduction of this chapter, the lower-bounding property implies that the similarity matching in the reduced space maintains its meaning with regards to the original space. The ability of SAX to define a lower bound is one of the reasons why SAX is so popular against other time series representation techniques.

III.4.2 Extensions of MINDIST

In the literature, the majority of SAX-like symbolization methods use a MINDIST-like distance measure on their symbolic sequences.

III.4.2.1 Symbolization methods with one symbol per segment

Symbolization techniques with only one symbol per segment usually use straightforward variants of MINDIST [LSK06; SK08; Qyo9; MF12; Fua12; Mal+13; Bai+13; KR20; KR21; LS22; DAM23]. These MINDIST variants generalize equation (III.55) according to their modified symbolization(s) step(s): the β_k coefficients correspond to the obtained quantization bins. For example, SFFA [LS22], which uses a chi-square strategy for the quantization bins β_k, replaces the obtained quantization bins in the MINDIST

Formula (III.55). As for the *Updated Minimum Distance (UMD)* [MFM10], it updates the look up table of MINDIST, so that adjacent symbols do not have a null distance.

While having one symbol per segment, a few methods do not use MINDIST-like distances. SBSR [Hug06] explores distance measures that have additional information compared to MINDIST, namely the data set global information, the time series local information, and the episode local information.

III.4.2.2 Symbolization methods with at least two symbols per segment

Symbolization techniques that have more than one feature per segment (in addition to the mean) usually include additive term(s) in Formula (III.54) to take into account their added feature(s) [PLD10b; LZY12; LDH13; Sun+14; Zha+18; YAD19; ZDX20; LTN20; Che+20].

When the added feature is quantized, a specific predefined look-up table is often defined. For example, SAX_DR [LDH13] introduces the direction distance using a precomputed look-up table between the directions, and TSX [LZY12] uses a precomputed look-up table for the trends. To give a formalized example in more details, we describe the SAX_DR distance. SAX_DR symbolizes the mean and the direction for each segment. Let us consider a time series $x = (x_1, \ldots, x_n)$. We denote by $\widehat{x}_{\text{mean}} = (\widehat{x}_{\text{mean},1}, \ldots, \widehat{x}_{\text{mean},w})$ the symbolic sequence of the symbolized means (i.e. the SAX representation) and $\widehat{x}_{\text{dir}} = (\widehat{x}_{\text{dir},1}, \ldots, \widehat{x}_{\text{dir},w})$ the symbolic sequence of the symbolized directions, where w is the number of segments. $\widehat{x}$ is the total symbolic sequence, incorporating the means and the directions. The distance measure between two SAX_DR symbolic sequences $\widehat{x}$ and $\widehat{y}$ is thus

$$D_{\text{SAX\_DR}}(\widehat{x}, \widehat{y}) = D_{\text{MINDIST}}(\widehat{x}_{\text{mean}}, \widehat{y}_{\text{mean}}) + \sqrt{\frac{n}{w}} \sqrt{\sum_{i}^{w} \frac{\left(\text{dist}_{\text{dir}}\left(\widehat{x}_{\text{dir},i}, \widehat{y}_{\text{dir},i}\right)\right)^2}{w}}, \quad \text{(III.57)}$$

where the dist_{dir} function is based on a look-up table between symbolized directions.

When the added feature is not quantized and remains real-valued, the Euclidean distance can be used. For example, CSAX [LTN20], which extracts a real value for the complexity-invariant value CI (which is the normalized complexity estimate CE defined in Formula (II.1) on page 48) in addition to the symbolized mean, defines the following distance:

$$D_{\text{CSAX}}(\hat{x}, \hat{y}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(\text{dist}(\hat{x}_i, \hat{y}_i) + \left(CI(\hat{x}_i) - CI(\hat{y}_i)\right)^2\right)}, \quad \text{(III.58)}$$

where $CI(\hat{x}_i)$ denotes the complexity-invariant value on the corresponding segment. To give another example that is not using the Euclidean distance, SAX-TD [Sun+14] defines a distance measure called *TDIST* that includes the trend:

$$D_{\text{TDIST}}(\hat{x}, \hat{y}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(\left(\text{dist}(\hat{x}_i, \hat{y}_i)\right)^2 + \frac{w}{n}\left(\text{dist}_{\text{td}}(x, y)\right)^2\right)}, \quad \text{(III.59)}$$

where the trend distance called dist_{td} is defined on the real-valued time series and involves the starting and ending points of the segments.

On the contrary, some methods do not use additive MINDIST-like terms. For example, TFSA [Yin+15] which uses three features (without the mean): the symbolized trend, the real-valued slope, and the real-valued end point, has multiplicative terms. Moreover, as it uses clustering for the quantization, the pSAX [BTT21b] distance between symbolic sequences, which are sequences of cluster center labels, is the Euclidean distance between the vectors of cluster centers coordinates.

III.4.2.3 Lower-bounding property

When using a MINDIST-like distance measure, an important aspect is to retain the lower bound. Quite a few methods claim that their proposed distance lowers bounds the Euclidean distance [LDH13; Bai+13; Sun+14; Yin+15; Zha+18; YAD19; Che+20; LS22; DAM23]. It appears that the definition of the distance measure of SAX_SD [ZY16], which claims to guarantee the lower bound, is wrong: the properties of the square root are not respected when going from Equation (10) to Equation (11). Furthermore, some methods claim to have a tighter lower bound than SAX [LDH13; Bai+13].

III.4.3 Distance measures between extracted features

Other methods use extracted features to compare signals. (i) Some distances extract features from the symbolic sequences. The extracted features are mostly based on the frequency of symbolic words. SAX-VSM [SM13] uses a *Term Frequency - Inverse Document Frequency (TF-IDF)* weighting scheme, then the cosine similarity. TF-IDF was originally applied to natural language processing tasks. Other methods are based on histograms. BOSS [Sch15], a dictionary-based classifier, applies an overlapping sliding on which SFA [SH12] is applied. Two time series are then compared based on their histograms of symbolic words. Other methods focus on the permutation entropy, which is extracted from ordinal patterns applied to a time series [BP02]. The permutation entropy in each time series is then compared. The *Permutation Jensen-Shannon Distance (PJSD)* [Zun+22] combines ordinal patterns with the Jensen-Shannon divergence. (ii) There are a few works that use the intermediate feature extraction (the step following segmentation and preceding quantization) of the symbolization. For instance, EN-SAX [BABO12] uses the cosine similarity between vectors of the extracted features per segment (the mean, minimum, and maximum).

III.4.4 Edit distances

The literature on symbolic representations advocates that a major advantage of symbolic representations is their ability to leverage the richness of the bioinformatics and text processing communities [Lin+07; EG20a]. However, to the best of our knowledge, only a few symbolic representations make use of distance measures defined on strings. *Symbolic Vector Quantized Approximation (SVQA)* [WML05] extends PVQA [MLW04], described in Section II.4.3, and uses LCSS on the string symbolic sequences, where a symbol is a codeword index. Similarly, *SAXLCSS* and *1D-SAX*$_{LCSS}$ [TTK17] respectively use the SAX and 1d-SAX representations, then LCSS for the distance measure.

III.5 Distances on multivariate time series

Until now, we have focused on distance measures for univariate series. Indeed, most distances in the literature are univariate. Recently, several strategies have been designed to extend DTW to multivariate time series [SY+17]. More recently, a review [Shi+23] applies these same strategies to extend seven other elastic distances (that were described in Sections III.2 and III.3.4): DDTW [KP01], WDTW [JJO11], WDDTW [JJO11], LCSS [VKG02], ERP [CN04], MSM [SAD13], and TWE [Mar09]. Two popular approaches are used in practice: the independent and dependent strategies. In the following, for each distance, the I subscript indicates the multivariate distance using the independent strategy, and the D subscript indicates the dependent strategy. As stated in [Shi+23], the time complexities of all these multivariate distances increase linearly with the number of dimensions.

Let us illustrate these strategies on DTW. In [SY+17], each dimension of the time series is z-normalized in order for the distance measure to be invariant to offset and scale. Thus, the multivariate DTW can handle dimensions of different physical natures for example.

- In the *independent strategy* for multivariate DTW, called DTW-I, the univariate DTW is applied to each dimension separately, and the resulting distances on each dimension are summed. As its name suggests, all dimensions are treated independently and their warping is independent.
- The *dependent strategy* for multivariate DTW, called DTW-D, considers the multivariate series as a single series in which each timestamp is associated to a single multidimensional point. The DTW scheme is applied by using Euclidean distances between the multidimensional points of the two series in Formula (III.23) on page 73. There is a unique warping path that deals with all dimensions.

An empirical review of these two approaches is conducted in [SY+17] and concludes that, on the nearest neighbor classification tasks, DTW-D performs better than DTW-I on some data sets, while DTW-I outperforms DTW-D on some other data sets: in general, there is no definitive recommendation for a specific strategy to use. If the warping paths of DTW-I are all the same, then the warping path of DTW-D should be similar to the ones of DTW-I.

For the other elastic distances, the same multivariate strategies are employed. In the following, when the independent strategy is straightforward, it is not described.

- For WDTW, the derivatives are obtained separately on each dimension, then are fed to DTW-I to obtain DDTW-I, or to DTW-D to obtain DDTW-D.
- For WDTW-D, the weight is applied to the DTW-D scheme.
- For WDDTW, the derivatives are obtained separately on each dimension, then are fed to WDTW-I to obtain WDDTW-I, or to WDTW-D to obtain WDDTW-D.
- LCSS-I computes the LCSS for each dimension, but each dimension has its own threshold value. LCSS-D works as LCSS but computes the *squared* Euclidean distance between multidimensional points in Formula (III.43) on page 86.
- ERP-D works as ERP but computes the Euclidean distance between multidimensional points in Formula (III.45) on page 87 where the penalty parameter is now a vector.

- MSM-D works as MSM but computes the *squared* Euclidean distance between multidimensional points in Formula (III.47) on page 87 where W_{MSM} is now adapted to input vectors.
- TWE-D works as TWE but computes the Euclidean distance between multidimensional points in Formula (III.51) on page 88.

For more details, we refer an interested reader to the recent review [Shi+23], which also introduces *Multivariate Elastic Ensemble (MEE)*, the multivariate extension of EE [LB15] where each univariate elastic measure of EE is extended to its multivariate version.

III.6 Conclusion

In this chapter, we have reviewed distance measures that are defined on time series, strings, and symbolic sequences. These distances are at the core of many data mining tasks.

For distance measures on time series, if the input time series do not have the same length (or do not have the same time synchronization), then an elastic distance measure is more suitable. DTW is the most popular elastic distance measure, and a lot of variants have proposed to make it more accurate and/or faster (see Figure III.8 on page 75). Elastic distance measures, other than DTW, are based on the edit distance originally defined for strings. A summary of distance measures on time series is shared in Table III.2.

Table III.2: Summary of distance measures on time series of length n.
[§]Metric for $p \geq 1$.
[§§]r is the window size.
[§§§]s is the slope parameter.
[†]p is the number of parameters. The time complexity reported is for the training time.

Distance name	Elastic	Varying lengths	Constrained	Triangle inequality	Metric	Time complexity
L_p distance [YF00]	✘	✘	✘	✔	$\approx^{\S}$	$\mathcal{O}(n)$
Classic DTW [BC94]	✔	✔	✘	✘	✘	$\mathcal{O}\left(n^2\right)$
DTW with Sakoe-Chiba band [SC78]	✔	✔	✔	✘	✘	$O(nr)^{\S\S}$
DTW with Itakura band [Ita75]	✔	✔	✔	✘	✘	$O(ns)^{\S\S\S}$
WDTW [JJO11]	✔	✔	✔	✘	✘	$\mathcal{O}\left(n^2\right)$
LCSS [VKG02]	✔	✔	✘	✘	✘	$\mathcal{O}\left(n^2\right)$
ERP [CN04]	✔	✔	✘	✔	✔	$\mathcal{O}\left(n^2\right)$
MSM [SAD13]	✔	✔	✘	✔	✔	$\mathcal{O}\left(n^2\right)$
TWE [Mar09]	✔	✔	✘	✔	✔	$\mathcal{O}\left(n^2\right)$
EE [LB15]	✔	✔	✔	✘	✘	$\mathcal{O}\left(pn^2\right)^{\dagger}$

For distance measures on strings, they are based on the general edit distance framework. We have defined 6 edit operations. Each edit distance allows a subset of these edit operations in order to model different behaviors in the compared strings. Given the edit operations involved, the properties of each edit distance is impacted. A summary of these edit distances is shared in Table III.3.

Table III.3: Summary of edit distances on strings, with their authorized operations, whether they are a metric, whether they can input strings of varying lengths, and their time complexity.
†Depends on how the operation costs are set.

	Allowed edit operations						Properties			
Distance name	Insertion	Deletion	Substitution	Transposition	Duplication	Contraction	Symmetric	Metric	Varying lengths	Time complexity
LCSS [Hir77]	✔	✔	✘	✘	✘	✘	✔	✘	✔	$\mathcal{O}(mn)$
Hamming [SM83]	✘	✘	✔	✘	✘	✘	✔	✔	✘	$\mathcal{O}(m)$
Simple Levenshtein distance [Lev+66]	✔	✔	✔	✘	✘	✘	✔	✔	✔	$\mathcal{O}(mn)$
General Levenshtein distance [Lev+66]	✔	✔	✔	✘	✘	✘	$\approx^{\dagger}$	$\approx^{\dagger}$	✔	$\mathcal{O}(mn)$
Damerau-Levenshtein	✔	✔	✔	✔	✘	✘	✔	✔	✔	$\mathcal{O}(mn)$
Edit Distance with Duplications and Contractions (EDDC) [BR02; Pin+13]	✔	✔	✔	✘	✔	✔	✔	✔	✔	$\mathcal{O}\left(\lvert\mathcal{A}\rvert m^3\right)$

We have also reviewed distance measures on symbolic sequences, which are mainly based on MINDIST from SAX. A distance measure defined on symbolic sequences can be viewed as a distance measure on real-valued time series when combined with the symbolization technique. Surprisingly, few distance measures on symbolic sequences employ edit distances, and the exploration of the multivariate setting is still in its early stages.

Chapter IV

ASTRIDE: Adaptive Symbolization for Time Series Databases

We introduce ASTRIDE (Adaptive Symbolization for Time seRIes DatabasEs), a novel symbolic representation of time series, along with its accelerated variant FASTRIDE (Fast ASTRIDE). Unlike most symbolization procedures, ASTRIDE is adaptive during both the segmentation step by performing change-point detection and the quantization step by using quantiles. Instead of proceeding signal by signal, ASTRIDE builds a dictionary of symbols that is common to all signals in a data set. We also introduce D-GED (Dynamic General Edit Distance), a novel distance measure on symbolic representations based on the general edit distance. We demonstrate the performance of the ASTRIDE and FASTRIDE representations compared to SAX (Symbolic Aggregate approXimation), 1d-SAX, SFA (Symbolic Fourier Approximation), and ABBA (Adaptive Brownian Bridge-based Aggregation) on reconstruction and, when applicable, on classification tasks. These algorithms are evaluated on 86 univariate equal-size data sets from the UCR Time Series Classification Archive.

Contents

IV.1 Introduction

Over the past decades, the increasing amount of available time series data has led to a rising interest in time series data mining. In many applications, the collected data take the form of complex time series which can be multivariate, multimodal, or noisy. A fundamental issue is to adopt an actionable representation which takes into account temporal information. In this regard, symbolic representations constitute a tool of choice [Lin+07]. Symbolic representations of time series are used for data mining tasks such as classification [Lin+07; SM13; Sch15; Ngu+19], clustering [Lin+07], indexing [Lin+07; Cam+10], anomaly detection [EG20a; CG23], motif discovery [Sen+18], and forecasting [EG20b]. The domain applications include finance [LSK06; BABO12], healthcare [SW10], and manufacturing [PJ20].

Briefly, most symbolization techniques follow two steps: a segmentation step where a real-valued signal $y = (y_1, \dots, y_n)$ of length n is split into w segments, then a quantization step where each segment is mapped to a discrete value $\hat{y}_i$ taken from a set $\{\mathtt{a}_1, \dots, \mathtt{a}_A\}$ of A symbols. The resulting symbolic representation is the discrete-valued signal (or *symbolic sequence*) $\hat{y} = (\hat{y}_1, \dots, \hat{y}_w)$. The set of symbols $\{\mathtt{a}_1, \dots, \mathtt{a}_A\}$ is usually called an *alphabet* or *dictionary*, and A is the *alphabet size*; the length w of the symbolic representation is called the *word length*. While there exist many high-level representations for time series [Fu11], the two main advantages of symbolic representations are reduced memory usage, and often a better score on data mining tasks thanks to the smoothing effect induced by compression [Lin+07].

In the present chapter, we introduce *ASTRIDE (Adaptive Symbolization for Time seRIes DatabasEs)* [CTO23b], a novel symbolic representation of time series data bases, along with its accelerated variant *FASTRIDE (Fast ASTRIDE)*. Unlike most symbolization techniques, ASTRIDE is adaptive during both the segmentation step by performing change-point detection and the quantization step by using quantiles. As the segmentation and quantization are performed on the whole data set, a notable benefit of ASTRIDE is to define a common dictionary of symbols for all signals in the data set under consideration, thus further reducing memory usage. ASTRIDE comes with *DGED (Dynamic General Edit Distance)*, a new distance measure for symbolic sequences which is based on the general edit distance. As we shall see, ASTRIDE provides an intuitive symbolic representation which outperforms the state of the art in classification accuracy and achieves competitive results in signal reconstruction.

The remainder of the chapter is organized as follows. Section IV.2 provides an overview of symbolic representations and their distance measures, highlights their limits, and presents our main contributions. Section IV.3 introduces the novel ASTRIDE

and FASTRIDE symbolic representations, as well as the new D-GED distance measure. Section IV.4 contains an experimental evaluation of the accuracy of ASTRIDE and FASTRIDE for classification and for signal reconstruction compared to several state-of-the-art symbolization methods. Section IV.5 provides concluding remarks.

IV.2 Background and motivations

This section gives an overview of symbolic representations and their distance measures, then assesses their limits and presents our contributions. It also provides a summary of some symbolization methods in Table IV.2.

IV.2.1 Overview of symbolic representations

In 2003, a popular symbolic representation for time series was introduced: *Symbolic Aggregate approXimation (SAX)* [Lin+03; Lin+07]. In SAX, a symbolic sequence is a chain of characters, for example `abbcaabc` (or `01120012`). SAX has two parameters: the word length w and the alphabet size A. For instance, in the symbolic sequence `abbcaabc`, the parameters are $w = 8$ (length of the sequence) and $A = 3$ (number of possible symbols). The larger w and A, the better the quality of the SAX representation, but the lower the compression. Optimal values of w and A are highly dependent on the application and the data set. In SAX, each signal is centered and scaled to unit variance, then split into w segments of equal length. Next, the means of all segments are grouped together in bins and each segment is represented by the bin where its mean falls into. The bin boundaries are chosen so that all symbols are equiprobable under the assumption that the means follow a standard Gaussian distribution. A SAX transformation of a signal taken from the UCR Time Series Classification Archive [Dau+19] is shown in Figure IV.1.

Since the introduction of SAX, many variants and symbolization techniques have been proposed (see Chapter II). First of all, some variants focus on the feature(s) per segment. *Extended SAX (ESAX)* [LSK06] represents each segment by its mean, minimum, and maximum values. *1d-SAX* [Mal+13] represents two features with only one symbol per segment. It uses linear regression to compute the mean and the slope of each segment, then discretizes the mean (in A_{mean} symbols) and the slope (in A_{slope} symbols) separately using the same Gaussian assumption as in SAX. The final segment symbol is the combination of the mean symbol and the slope symbol. The alphabet size is therefore $A = A_{\text{mean}} \cdot A_{\text{slope}}$.

Some symbolization procedures perform an non-uniform segmentation in order to better adjust to the signal. Adaptive Segmentation Based Symbolic Representations (SBSR) [Hug06] can be viewed as a symbolic version of the *Adaptive Piecewise Constant Approximation (APCA)* representation [Cha+02], just as SAX can be viewed as the symbolic version of the *Piecewise Aggregate Approximation (PAA)* [Keo+01; YF00]. In SBSR, segment lengths adapt to the shape of the signal.

Some symbolic representations have an adaptive quantization step in order to relax the Gaussian assumption on the data. *Adaptive SAX (aSAX)* [PLD10a] uses a uniform segmentation and K-means clustering for the quantization. A symbol-based procedure to detect phases of gait signals [SW10] uses piecewise linear segmentation then K-means clustering on the features per segment to get the symbols. The features per

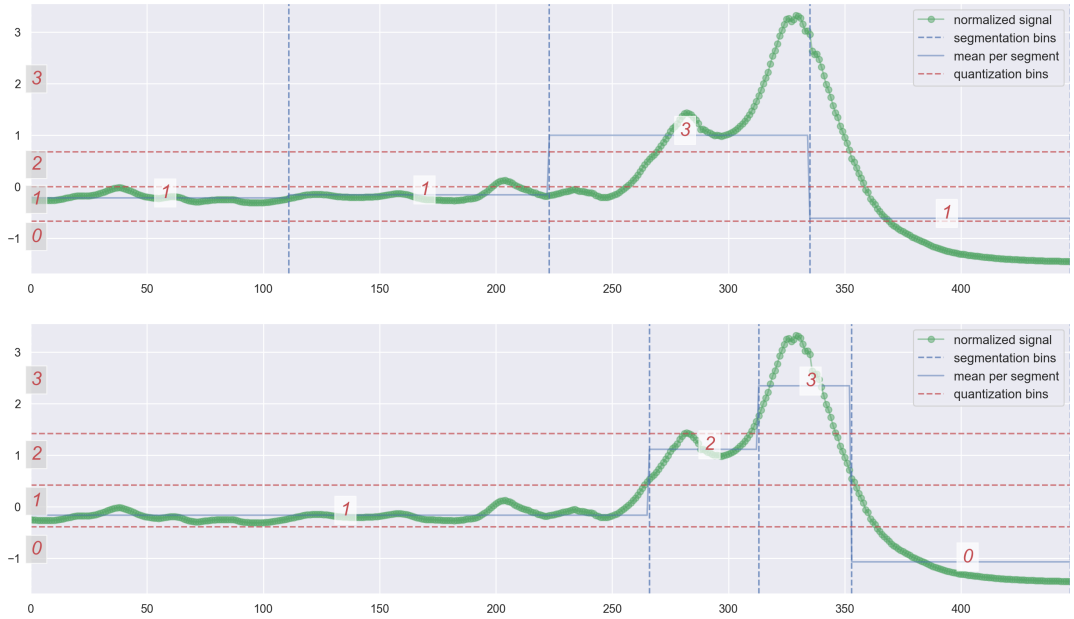


Figure IV.1: Example of a SAX (top) and ASTRIDE (bottom) representations of a signal from the Meat data set (UCR Time Series Classification Archive). The original length of the signal is $n = 448$, and we use $w = 4$ and $A = 4$. The resulting symbolic sequence is 1131 for SAX, and 1230 for ASTRIDE. (ASTRIDE is described in Section IV.3.)

segment include slope, length, mean, and variance. Symbolization is used in order to have a interpretable representation of gait signals. However, no distance measure is derived from this representation. *Adaptive Brownian Bridge-based Aggregation (ABBA)* [EG20a] is adaptive for both the segmentation and quantization steps. It also chooses w and A through a data-driven procedure. For the segmentation, adaptive piecewise linear continuous approximation of the signal is used. Each linear piece is chosen given a user-specified tolerance tol: when the value of tol increases, the resulting number of segments w decreases. The quantization step consists in a K-means clustering of the tuples of the increment over the segment and the segment length, where the number of clusters is set to A. ABBA uses a scaling parameter scl that calibrates the importance of the length in relation to its increment: the clustering is performed on the increments alone for $scl = 0$, while the clustering is done on both the length and increment with the same importance when $scl = 1$. Hence, the input parameters are the tolerance tol, the scaling scl, and the alphabet size A. Note that when A is not set by the user, ABBA does several runs the K-means algorithm to get the optimal value of A, resulting in a higher computational cost. ABBA focuses on signal reconstruction: there is no mention of a distance measure on symbolic representations. *Reconstruction* is the inverse transformation: the original signal is inferred from its transformation which is its symbolic sequence. A recent faster variant of the ABBA method, *fABBA* [CG23], replaces the K-means clustering by a sorting-based aggregation procedure that does not require the user to specify A.

While previously mentioned methods symbolize each signal independently, some procedures operate on a data set of signals. These methods share a dictionary of symbols across all signals of the considered data set. *Symbolic Fourier Approximation*

(SFA) [SH12] is based on the Discrete Fourier transform (DFT). First, SFA selects the w Fourier coefficients of lowest frequencies, and second, uses a procedure called *Multiple Coefficient Binning (MCB)* to quantize them. In detail, MCB computes a user-defined number A of quantiles per Fourier coefficient across all signals of a data set, and each Fourier coefficient is represented by the bin (based on quantiles) to which it belongs. In a supervised data mining task, the MCB bins are learned on a training set. SFA naturally provides a low-pass filtering that reduces the influence of noise. Also, no distance on SFA's symbolic representations is described. Note that SFA does not go through a segmentation step, but still has the w parameter that determines the length of the symbolic sequences.

Table IV.2 summarizes the main SAX variants as well as our novel ASTRIDE and FASTRIDE representations that will be presented in Section IV.3.

IV.2.2 Overview of distance measures on symbolic sequences

In order to use the learned symbolic representations for tasks such as classification or clustering, it is crucial to define a distance measure between symbolic sequences, which can be viewed as character strings. Distance measures on symbolic sequences were reviewed in Section III.4, while distances on strings were reviewed in Section III.3. Defining an informative measure is a challenge that has received a lot of attention.

SAX employs MINDIST, a distance measure on symbolic sequences. Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two real-valued time series with n samples. The Euclidean distance between x and y is given by

$$L_2(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{IV.1}$$

The *MINDIST* distance measure between the resulting symbolic sequences $\hat{x}$ and $\hat{y}$ mimics the Euclidean distance

$$\text{MINDIST}(\hat{x}, \hat{y}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (\text{dist}(\hat{x}_i, \hat{y}_i))^2} \tag{IV.2}$$

where the function $\text{dist}()$, based on a so-called look-up table, is illustrated in Table IV.1. MINDIST requires the symbolic sequences to be of equal length. For a given value of

Table IV.1: Example of look-up table for MINDIST with $A = 4$. For example, $\text{dist}(\texttt{a}, \texttt{d}) = 1.34$.

	a	b	c	d
a	0	0	0.67	1.34
b	0	0	0	0.67
c	0.67	0	0	0
d	1.34	0.67	0	0

the alphabet size A, this table is calculated only once, and then stored for fast look-up. For all look-up tables, whatever the alphabet size, the value in the cell of indexes (i, j)

is given by

$$\text{cell}_{i,j} = \begin{cases} 0, & \text{if } |i-j| \leq 1 \\ \beta_{\max(i,j)-1} - \beta_{\min(i,j)}, & \text{otherwise} \end{cases} \tag{IV.3}$$

where the β_k are the boundaries of the bins used by SAX to discretize the segment means. MINDIST is not a true metric, as $\text{dist}(\texttt{a}, \texttt{b}) = 0$ for example (see Table IV.1).

The literature on symbolic representations advocates that a major advantage of symbolic representations is their ability to leverage the richness of the bioinformatics and text processing communities [Lin+07; EG20a]. However, few symbolic representations employ distance measures defined on strings [WML05; TTK17]. A review on string matching is given in [Nav01]. A popular distance measure is the edit distance which is reviewed in Section III.3. For two strings `s1` and `s2`, it is the minimal cost of a sequence of operations that transform `s1` into `s2`. The edit distance is also called the Levenshtein distance [Lev+66]. If all the simple operations have a cost of 1, whatever the operation or the characters involved, it is called the *simple edit distance*. If the three authorized operations have different costs or the costs depend on the characters involved, it is called the *general edit distance*.

For each symbolization method presented in Section IV.2.1, Table IV.2 indicates whether it comes with a compatible distance measure.

IV.2.3 Limitations of existing symbolization methods

IV.2.3.1 The need for adaptive segmentation and quantization steps

As stated above, only a few methods are adaptive, and fewer are adaptive on both the segmentation and quantization steps. Let us understand with the SAX method why this can be an issue. Note that the same observations apply to SAX-like variants.

First, uniform segmentation has flaws. Most signals found in practice contain salient events that are crucial to perform tasks such as classification. However, uniform segmentation does not detect these events and neglects the phenomena of interest. As can be seen on the SAX representation of Figure IV.1, the two peaks around timestamps 280 and 330 are not detected. Uniform segmentation does not depend on the specific signal or data set at hand, but only on the input word length w. Moreover, because of it, SAX is restricted to input signals of equal length, while real-world signals are often of varying lengths [Tan+19].

As for quantization, the Gaussian assumption of SAX can be inappropriate for some data sets. SAX considers that the symbols obtained after quantization will be equiprobable because all normalized time series follow a Gaussian distribution. While normalized time series that are independent and identically distributed do tend to follow a Gaussian distribution, this in not the case for the means per segment [BK15]. To illustrate this point, we computed the means per segment for a data set from the UCR Time Series Classification Archive [Dau+19] and their histogram is displayed on Figure IV.2. As observed, the means per segment do not seem to follow a Gaussian distribution. We also performed the D'Agostino's K^2 normality test, whose null hypothesis is that the sample comes from a normal distribution. This test rejects the Gaussian assumption at the risk level $\alpha = 5\%$. In total, we computed the mean per segment for each of 86 univariate equal-size data sets from the UCR Time Series Classification Archive (that will be taken into account in the experiments of Section IV.4). All data sets reject the normal distribution hypothesis at the risk level $\alpha = 5\%$. Note that

Table IV.2: Synthetic comparison of some symbolization methods (including our proposed ASTRIDE and FASTRIDE) for a word length w and an alphabet size A, with a data set composed of N signals whose values are encoded on n_{bits} bits (for example $n_{\text{bits}} = 32$ bits or $n_{\text{bits}} = 64$ bits.) The A_{mean}, A_{min}, A_{max}, A_{slope}, A_{clusters}, $A_{\text{coefficients}}$ are respectively the number of symbols used to encode the mean, the minimum, the maximum, the slope, the number of clusters, and the number of Fourier coefficients.

†For ESAX, $A = A_{\text{mean}} = A_{\text{min}} = A_{\text{max}}$.

‡For SFA, there is no segmentation.

§For ASTRIDE, the lengths are common to all signals.

SAX variant	Features used per segment	Adaptive segmentation?	Adaptive quantization?	Shared dictionary?	Space (in bits) for a single symbolic sequence	Space complexity (in bits) of the dictionary of symbols for a data set of N signals	distance measure?
SAX [Lin+03]	means	✗	✗	✓	$w \left\lceil \log_2(A_{\text{mean}}) \right\rceil$	$n_{\text{bits}} A_{\text{mean}}$	✓
ESAX [LSK06]	maxs, mins, means	✗	✗	✓	$w \left\lceil \log_2(A_{\text{mean}} A_{\text{min}} A_{\text{max}}) \right\rceil$	$n_{\text{bits}} A^{\dagger}$	✓
1d-SAX [Mal+13]	means, slopes	✗	✗	✓	$w \left\lceil \log_2(A_{\text{mean}} A_{\text{slope}}) \right\rceil$	$n_{\text{bits}} A_{\text{mean}} A_{\text{slope}}$	✓
SBSR-Lo [Hug06]	means, change-points	✓	✗	✗	$w \left\lceil \log_2(A_{\text{clusters}}) \right\rceil$	$2N n_{\text{bits}} A_{\text{clusters}}$	✓
aSAX [PLD10a]	means	✗	✓	✓	$w \left\lceil \log_2(A_{\text{clusters}}) \right\rceil$	$n_{\text{bits}} A_{\text{clusters}}$	✓
Sant'Anna and Wickström [SW10]	ad hoc features	✓	✓	✗	$w \left\lceil \log_2(A_{\text{clusters}}) \right\rceil$	$5N n_{\text{bits}} A_{\text{clusters}}$	✗
SFA [SH12]	Fourier coefficients	N/A‡	✓	✓	$w \left\lceil \log_2(A_{\text{coefficients}}) \right\rceil$	$n_{\text{bits}} A_{\text{coefficients}}$	✗
ABBA [EG20a]	increments, lengths	✓	✓	✗	$w \left\lceil \log_2(A_{\text{clusters}}) \right\rceil$	$2N n_{\text{bits}} A_{\text{clusters}}$	✗
ASTRIDE	means, lengths§	✓	✓	✓	$w \left\lceil \log_2(A_{\text{mean}}) \right\rceil$	$n_{\text{bits}}(A_{\text{mean}} + w)$	✓
FASTRIDE	means	✗	✓	✓	$w \left\lceil \log_2(A_{\text{mean}}) \right\rceil$	$n_{\text{bits}} A_{\text{mean}}$	✓

the authors of SAX [Lin+07] emphasize that, if the normality assumption is not satisfied, the algorithm is less efficient but still correct due to the lower-bounding property.

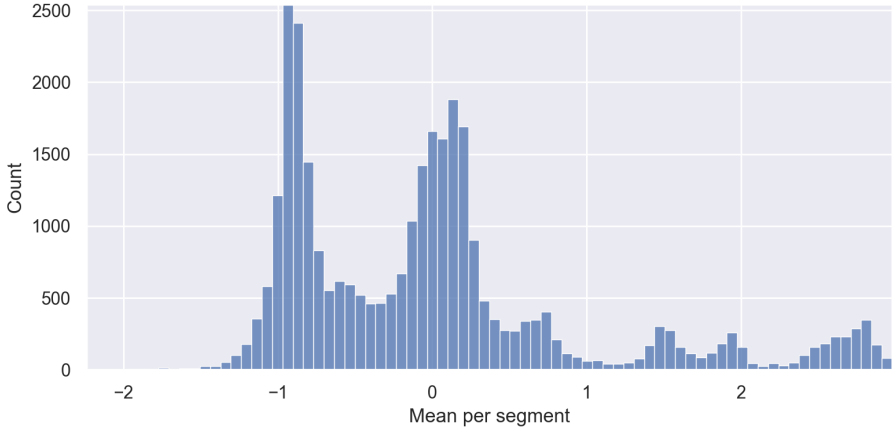


Figure IV.2: Example of histogram of the Strawberry data set (UCR Time Series Classification Archive) whose signals are of length $n = 235$ and for which the word length is set to $w = 32$. The obtained p-value for D'Agostino's K^2 normality test is 0: the means per segment do not come from a normal distribution.

IV.2.3.2 The need for a distance measure on symbolic sequences

As seen in Table IV.2, some symbolic representations do not provide a distance measure. Thus, they cannot be used directly for tasks such as classification or clustering. As stated in Section IV.2.2, most symbolization methods that have a distance measure are based on MINDIST from SAX. An issue of MINDIST is that it considers adjacent symbols to be equal. For example, the MINDIST measure between adjacent symbols `a` and `b` is null, hence symbols that are actually different will be considered equal by MINDIST, which can lead to misclassification of signals. Moreover, MINDIST is based on a Gaussian assumption. As a result, the MINDIST distance does not adapt to the signal. In addition, MINDIST is restricted to symbolic sequences of equal lengths. For example, MINDIST cannot be applied to the ABBA symbolic sequences.

IV.2.3.3 The need for a shared dictionary of symbols across the signals of a data set

As stated in the introduction, one of the goals of symbolization is to reduce the memory usage of the data. However, only a few papers mention that, in order to reconstruct a data set of N symbolic sequences, one needs to store the N symbolic sequences, but also the dictionary of A symbols for each signal. Denote by n_{bits} the number of bits needed to store a real value. A symbolic sequence with one symbol per segment requires $w \log_2(A)$ bits, resulting in $Nw \log_2(A)$ bits for N symbolic sequences. A dictionary of symbols with one real value per symbol needs $n_{\text{bits}}A$ per signal, resulting in $Nn_{\text{bits}}A$ bits for N symbolic sequences if the dictionary of symbols is not shared across signals.

Let us consider the SAX and the ABBA methods. SAX carries a shared dictionary of symbols across signals, while ABBA does not. For SAX, $Nw\log_2(A) + n_{\text{bits}}A$ bits are needed to reconstruct a data set of N symbolic sequences. For the dictionary of symbols of ABBA, each symbol is a cluster center, and each cluster center has two real values: the length and the increment. Hence, each symbol requires $2n_{\text{bits}}$ bits in memory. But, contrary to SAX, ABBA needs to store one dictionary of symbols per signal. As a result, we need $Nw\log_2(A) + 2n_{\text{bits}}NA$ bits for the whole data set. For the Meat data set with $N = 120$, $w = 10$, $A = 9$, and $n_{\text{bits}} = 64$ bits, the memory usage to encode the whole data set is $4{,}380$ bits for SAX, and $142{,}044$ bits for ABBA, thus 32 times more. For ABBA, encoding the dictionary of symbols costs 36 times more than encoding the symbolic sequences. As a result, ABBA requires much more memory usage than SAX because it is adaptive and its dictionary of symbols is not shared. The memory usage to reconstruct a data set of symbolic sequences for more methods are given Table IV.2.

IV.2.4 Contributions

To the best of our knowledge, the ASTRIDE method to be presented in Section IV.3 is the only symbolic representation offering adaptive segmentation and quantization, a shared dictionary of symbols as well as a compatible distance measure and a reconstruction procedure. Altogether, ASTRIDE circumvents the limitations of the methods described in Section IV.2.3.

Instead of using uniform segmentation, ASTRIDE performs adaptive segmentation, a.k.a. change-point detection [TOV20], in order to capture salient events. More precisely, we detect changes in the mean, where the number of changes is set by the user. Moreover, ASTRIDE does not rely on the Gaussian assumption for the quantization: this step is adaptive on the signals at hand. Consequently, ASTRIDE does not require any assumption on the distribution of the data.

We also introduce Dynamic General Edit Distance (D-GED), a new distance measure on symbolic representations which is based on the general edit distance. Moreover, unlike MINDIST, the symbolic sequences are not required to be of equal lengths.

Adaptive segmentation and quantization are learned at the level of the data set of signals: the change-points as well as the quantiles (for the quantization) are estimated using all signals in the data set. ASTRIDE's dictionary of symbols is the same for all signals (unlike ABBA), and is thus memory-efficient.

IV.3 The ASTRIDE method

ASTRIDE (Adaptive Symbolization Time seRIes DatabasEs) is a novel symbolic representation for data sets of signals. It is an offline method inputting univariate signals that are required to be of equal size. There are two parameters to be set by the user: the word length w and the alphabet size A. ASTRIDE comes with a new distance measure on symbolic sequences: D-GED (Dynamic General Edit Distance). After describing ASTRIDE and D-GED, we introduce FASTRIDE (Fast ASTRIDE), an accelerated version of ASTRIDE.

IV.3.1 ASTRIDE segmentation step

As a preprocessing step, all times series in the data set are centered and scaled to unit variance. Then, the N signals of length n are segmented. To that end, all signals are stacked, producing a single multivariate signal of length n and dimension N. ASTRIDE applies multivariate change-points detection with a fixed number of segments on this high-dimensional signal. When w segments are chosen, the segmentation provides $w-1$ change-points that are the same for each univariate signal. Since the change-points are common to all (univariate) signals, this allows ASTRIDE to be memory-efficient. The lengths of each resulting symbolic sequence are the same (equal to w). For a given multivariate signal $y = (y_1, \ldots, y_n)$ with n samples, change-point detection finds the $w-1$ unknown instants $t_1^* < t_2^* < \ldots < t_{w-1}^*$ where some characteristics (here, the mean) of y change abruptly. A recent review of such methods is given in [TOV20]. In the context of ASTRIDE, the number of changes $w-1$ is chosen by the user: it is the desired number of regimes, meaning the length of the resulting symbolic sequences. The change-point algorithm estimates $\hat{t}_1, \ldots, \hat{t}_{w-1}$ which are the minimizers of a discrete optimization problem

$$\left(\hat{t}_1, \ldots, \hat{t}_{w-1}\right) = \underset{(w, t_1, \ldots, t_{w-1})}{\arg\min} \sum_{k=0}^{w+1} \sum_{t=t_k}^{t_{k+1}-1} \|y_t - \bar{y}_{t_k:t_{k+1}}\|^2, \tag{IV.4}$$

where $\bar{y}_{t_k:t_{k+1}}$ is the empirical mean of $\{y_{t_k}, \ldots, y_{t_{k+1}-1}\}$. By convention, $t_0 = 0$ and $t_w = n$. Formulation (IV.4) seeks to reduce the error between the original signal and the best piecewise constant approximation. This problem is solved using dynamic programming which has a time complexity of $\mathcal{O}\left(Nwn^2\right)$ where N is the number of signals in the data set.

Figure IV.1 displays an example of an ASTRIDE representation of a signal, along with the SAX representation (for the same parameters w and A). Visually, compared to uniform segmentation, adaptive segmentation leads to more meaningful segments. For example, it detects that one segment is sufficient to approximate the signal from timestamp 0 to 250, and that there is a peak around timestamp 280 and another one around timestamp 330. It shows the importance of our adaptive segmentation scheme. Figure IV.3 depicts how the multivariate change-point detection works. The algorithm tries to find the abrupt changes in mean that are common to most (univariate) signals in the data set.

IV.3.2 ASTRIDE adaptive quantization step

After segmentation, the means of all segments are computed and grouped into bins based on the empirical quantiles. Each segment is then symbolized by the bin it belongs to. This quantization step is similar to the MCB (Multiple Coefficient Binning) procedure of SFA [SH12]. Since the segments found during the segmentation step correspond to mean-shifts, it is reasonable to represent each segment by its mean value. The $A-1$ quantiles are calculated on the means of all segments of all signals in the data set, leading to A symbols. The time complexity of the quantization step (computing the means, the quantiles, and applying the binning) is $\mathcal{O}\left(Nw\right)$, where N is the number of signals in the data set. By design, all symbols are equiprobable. Figure IV.1 shows an example of an ASTRIDE representation. Compared to SAX, the bins of ASTRIDE represent the quantiles of the means per segment and are quite different

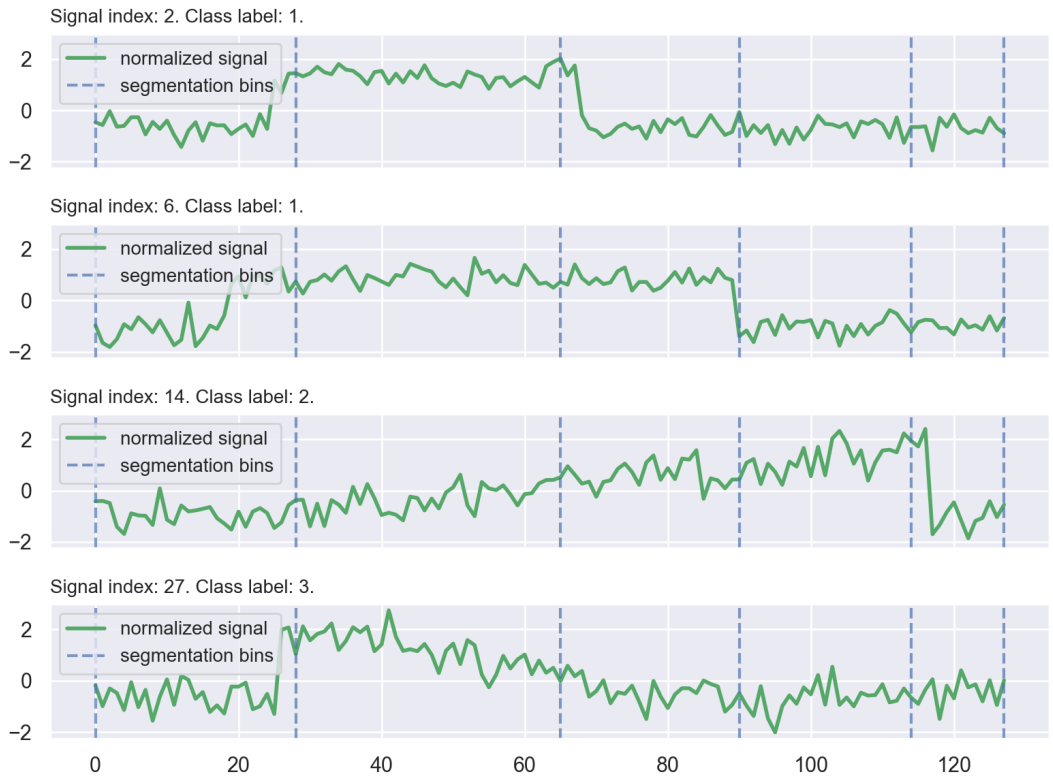


Figure IV.3: Multivariate change-point detection on (univariate) signals from the CBF data set (UCR Time Series Classification Archive). Here, $n = 128$ and $w = 5$. The change-points are obtained from the whole training set, but only a few signals are displayed.

from the ones of SAX. Recall that ASTRIDE is fitted on the whole training set (and not on the displayed signal only).

IV.3.3 The D-GED distance measure

We introduce Dynamic General Edit Distance (D-GED), a novel distance measure on symbolic representations. D-GED is compatible with symbolic sequences of equal or varying lengths. The distance measure D-GED is based on the general edit distance [Navo1]. D-GED sets the operation costs of the general edit distance so that they incorporate the distance between individual symbols as follows:

- The substitution cost $\mathrm{sub}(\mathtt{a}, \mathtt{b})$ for individual symbols $\mathtt{a}$ and $\mathtt{b}$ is the Euclidean distance between the mean $\mu_{\mathtt{a}}$ of the mean values attributed to symbol $\mathtt{a}$ and the mean $\mu_{\mathtt{b}}$ of the mean values attributed to symbol $\mathtt{b}$

$$\mathrm{sub}(\mathtt{a}, \mathtt{b}) = \|\mu_{\mathtt{a}} - \mu_{\mathtt{b}}\|_2 \,. \tag{IV.5}$$

- For all characters, the insertion and deletion costs are set to $\mathrm{sub}_{\max}$, where $\mathrm{sub}_{\max}$ is the maximum value of the modified substitute costs given in (IV.5).

For the substitution cost, the intuition is that if symbols $\mathtt{a}$ and $\mathtt{b}$ are "very different", then the difference between $\mu_{\mathtt{a}}$ and $\mu_{\mathtt{b}}$ will be wider, and substituting them will have a larger cost in D-GED. By setting the insertion and deletion costs to $\mathrm{sub}_{\max}$, D-GED favors substitutions over insertions and deletions. The worst-case complexity to compute the D-GED distance measure of two symbolic sequences of lengths w_1 and w_2 is $\mathcal{O}(w_1 w_2)$.

D-GED is not applied directly on the symbolic representation but on a replicated version. Indeed, when a method uses a non-uniform segmentation, the segments can have different lengths. Without taking into account the varying segment sizes, D-GED would compare (substitute or delete/insert) symbols corresponding to segments of different lengths. To prevent ASTRIDE from losing this information, we propose the following procedure. Denote by $\ell_1, \ldots, \ell_w$ the segment lengths obtained with our adaptive segmentation. By design, they are the same for all signals in the data set. Each segment length is divided by the minimum of all segments lengths and rounded to its nearest integer to obtain the *normalized segment lengths* $\widehat{\ell}_1, \ldots, \widehat{\ell}_w$. Then, the symbolic sequences are modified by replicating the symbol of the first segment $\widehat{\ell}_1$ times, then the symbol of the second segment $\widehat{\ell}_2$ times, etc. Finally, the D-GED measure between these replicated symbolic sequences is computed. As an example, consider the symbolic sequence from ASTRIDE depicted in Figure IV.1. The symbolic sequence without incorporating information about the segment lengths is 1230. The segment lengths are $(266, 47, 40, 95)$ before normalization (more details on the signal are given in Table IV.3). The smallest segment has 40 samples and, as a result, the normalized segment lengths are $(7, 1, 1, 2)$. The replicated symbolic sequence based on the normalized segment lengths is

$$\underbrace{1111111}_{\text{7 times}}\ \underbrace{2}_{\text{once}}\ \underbrace{3}_{\text{once}}\ \underbrace{00}_{\text{twice}} \tag{IV.6}$$

which is of total length 11.

Table IV.3: Details of the ASTRIDE representation of the signal displayed on Figure IV.1. The parameters of ASTRIDE are $w = 4$ and $A = 4$. (The quantized mean feature is described in Section IV.3.4.)

segment start	mean	symbol	quantized mean	length	normalized length
0	-0.17	1	-0.16	266	7
266	1.11	2	1.05	47	1
313	2.34	3	2.38	40	1
353	-1.07	0	-1.06	95	2

IV.3.4 Reconstruction of ASTRIDE symbolic sequences

In ASTRIDE, a signal is reconstructed from its symbolic sequence as follows. Each symbol of the symbolic representation is replicated ℓ_k times, where ℓ_k is the length of the associated segment. The length of the reconstructed signal is the same as the original one. Then, each symbol is replaced by the average of all segment means that belong to the associated bin (the quantized mean). This resulting real-valued signal is the reconstruction by ASTRIDE. As an example, consider the signal shown in Figure IV.1 whose symbolic representation is 1230. Details about the segment lengths, symbols, and quantized mean are given in Table IV.3. First, the symbols 1, 2, 3 and 0 are replicated 266, 47, 40 and 95 times respectively. Second, each symbol is mapped to its quantized mean, going from a symbolic signal to a real-valued signal:

$$\underbrace{\boxed{\text{-0.16}}\ldots\boxed{\text{-0.16}}}_{266\text{ times}}\underbrace{\boxed{\text{1.05}}\ldots\boxed{\text{1.05}}}_{47\text{ times}}\underbrace{\boxed{\text{2.38}}\ldots\boxed{\text{2.38}}}_{40\text{ times}}\underbrace{\boxed{\text{-1.06}}\ldots\boxed{\text{-1.06}}}_{95\text{ times}} \tag{IV.7}$$

The real-valued signal displayed in Formula (IV.7) is the reconstruction of the 1230 symbolic sequence. A reconstructed signal from ASTRIDE is a piecewise constant signal, as displayed in Figure IV.4. Notice how the reconstructed signal in Figure IV.4 is

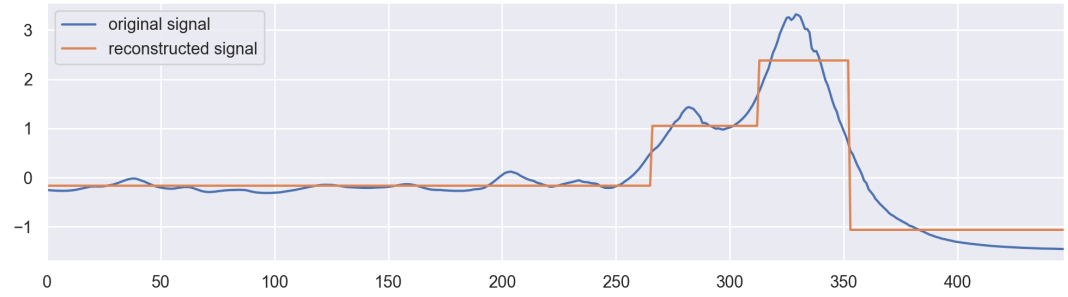


Figure IV.4: Example of reconstruction by ASTRIDE of the 1230 symbolic sequence (same original signal as in Figure IV.1). Here, $w = 4$ and $A = 4$. Note that ASTRIDE is fitted on the whole training set (and not on the displayed signal only).

different from the mean per segment representation in Figure IV.1, as the quantized mean is used, and not the mean.

The memory cost of ASTRIDE is easily derived. To reconstruct N symbolic sequences from the ASTRIDE representation with w segments and A symbols, one needs

to store $Nw\log_2(A)$ bits for the symbolic sequences. For the shared dictionary of symbols, one needs $n_{\text{bits}}A$ bits. In addition to storing the symbolic sequences with their shared dictionary of symbols, one also needs to store the w segment lengths that are the same for all symbolic sequences, resulting in $n_{\text{bits}}w$ bits for the whole data set. In total, $Nw\log_2(A) + (w + A)n_{\text{bits}}$ are required. To illustrate, let us take the example in Section IV.2.3.3: the Meat data set with $N = 120$, $w = 10$, $A = 9$, and $n_{\text{bits}} = 64$ bits. For ASTRIDE, the memory usage to encode the whole data set is $5{,}020$ bits. Recall that the memory usage is $4{,}380$ bits for SAX, and $142{,}044$ bits for ABBA.

IV.3.5 The FASTRIDE method

FASTRIDE (Fast ASTRIDE) is an accelerated variant of ASTRIDE. For the symbolization procedure, the only difference is its segmentation step which is uniform, like SAX. The reconstruction of FASTRIDE is performed in the same way as ASTRIDE. For the distance measure of FASTRIDE, we use D-GED but there is no need to replicate the symbolic sequences, as the segment lengths are equal due to the uniform segmentation. FASTRIDE is computationally faster than ASTRIDE because FASTRIDE skips the adaptive segmentation step, and the input symbolic sequences of the general edit distance are not replicated, thus are shorter.

IV.4 Experimental results

We compare ASTRIDE and FASTRIDE to several popular symbolic representations (SAX, 1d-SAX, SFA, ABBA) on a classification task and a reconstruction task. We show that ASTRIDE and FASTRIDE constitute the best compromises to address both these tasks. Indeed, as will be discussed, some of these methods can be used only on one task; in particular, SFA and ABBA do not possess a distance measure and, therefore, cannot be used as such for classification.

The adaptive segmentation step of ASTRIDE is implemented with the `ruptures` Python package [TOV20]. The general edit distance in D-GED uses the `weighted-levenshtein` Python package [Inf18]. SAX and 1d-SAX are implemented in the `tslearn` Python package [Tav+20]. SFA is implemented from scratch. ABBA is taken from the authors' GitHub repository[1]. A Python implementation of ASTRIDE and FASTRIDE, along with codes to reproduce the figures and scores in this chapter, can be found in a GitHub repository[2].

IV.4.1 Classification task

We first investigate the performances of our approaches on a classification task.

IV.4.1.1 Experimental setup

Data mining task and competitors Our methods ASTRIDE and FASTRIDE are compared to SAX and 1d-SAX. One-Nearest Neighbor (1-NN) classification is used to compare the quality of both the symbolizations and the distance measures, as often done

[1] https://github.com/nla-group/ABBA
[2] https://github.com/sylvaincom/astride

Table IV.4: Presentation of the 86 univariate equal-size data sets from the UCR Time Series Classification Archive considered in our classification experiment.

	No. of signals	Length	No. of classes
mean	1,357	644	10
min	40	128	2
50%	687	456	4
max	9,236	2,844	60

in the literature [Bag+17]. For ASTRIDE and FASTRIDE, the change-points and the quantization bins are learned on the training set.

Our comparison is limited to classification techniques based on symbolizations, since our objective is to evaluate the relevance of this step itself and not to achieve state-of-the-art performance on time series classification. Hence, we exclude classifiers that are built on top of symbolic representations, namely bag-of-words and ensemble-based algorithms. In particular, SAX-VSM (SAX and Vector Space Model) [SM13], BOSS (Bag-of-SFA-Symbols) [Sch15], Mr-SEQL (Multiple symbolic representations SEQuence Learner) [Ngu+19], WEASEL (Word ExtrAction for time SEries cLassification) [SL17], and TDE (Temporal Dictionary Ensemble) [Mid+20] are out of the scope on this study. More details on these techniques can be found in [Bag+17; Rui+21; Aga+21].

Hyperparameters The hyperparameters for all methods are:

- the word length w in $\{5, 10, 15, 20, 25\}$
- the alphabet size A in $\{4, 9, 16, 25\}$.

For 1d-SAX, $A \in \{4, 9, 16, 25\}$ corresponds to $(A_{\text{mean}}, A_{\text{slope}}) \in \{(2, 2), (3, 3), (4, 4), (5, 5)\}$.

Evaluation of the task The evaluation metric for the classification is the test accuracy: percentage of correctly classified signals.

Data sets Since SAX and other methods can be applied only to univariate and equal-size times series and we choose our signals to be of length at least 100 (as in the ABBA paper [EG20a]), the scope of our comparisons is restricted to 86 data sets of the UCR Times Series Classification Archive [Dau+19]. The data sets are both real-world and synthetic, and come with a default train / test split which is the one used in this study. Our experiments were launched on a total of 66,827,003 samples across all signals of all data sets. Table IV.4 recaps some key figures of the considered data sets. While averaging accuracies over different data sets, with different sizes and challenges, has some flaws, it is the best compromise to obtain a global key figure to assess the quality of a classifier on the UCR Time Series Classification Archive.

IV.4.1.2 Results

Figure IV.5 displays the accuracy scores as a function of the word length w averaged over the selected data sets, for several methods, and for different values of alphabet

size A. For each method and each alphabet size A, plotting the accuracy in relation to w tells us which method provides the best representation: the larger the accuracy for a given w, the better the symbolic representation.

Our results show that ASTRIDE and FASTRIDE perform better than both SAX and 1d-SAX on the classification task. Indeed, for each alphabet size A and each word length w, ASTRIDE and FASTRIDE have a higher accuracy than both SAX and 1d-SAX. This shows that the proposed adaptive symbolization process, combined with the D-GED distance measure, is relevant in this classification context.

Influence of the parameters We observe that, for all methods, the accuracy increases as the word length w increases. This was expected as the symbolization becomes more precise as each signal is represented with more bits. Interestingly, ASTRIDE and FASTRIDE achieve reasonable classification results even for a very small number of segments. For example, when $w = 5$ and $A = 16$, ASTRIDE has a score of 57%, while SAX and 1d-SAX have a score of 48%. This confirms the fact that using a more adaptive representation better captures the phenomenon observed in the signals, and thus better compresses the information.

For all methods, when the alphabet size A increases, the classification scores improve. Yet, ASTRIDE and FASTRIDE are less sensitive to the value of A. For $w = 20$, the accuracy of ASTRIDE is of 61%, 65%, 67%, and 68%, for $A = 4$, $A = 9$, $A = 16$, and $A = 25$ respectively. On the contrary, SAX seems to be very sensitive to the value of A: for $w = 20$, it reaches 45% for $A = 4$ and 62% for $A = 16$. Moreover, the performance of SAX is worse than 1d-SAX for small values of A, and is slightly better than 1d-SAX (and even largely surpasses it) for large values of A.

Importance of the adaptive approaches According to Figure IV.5, FASTRIDE achieves results similar to those of ASTRIDE, which suggests that the adaptive segmentation does not increase performances. As will be seen in the next section, the relevance of the segmentation phase is more acute in the reconstruction task.

The importance of the adaptive quantization process based on quantiles can been assessed by comparing the performances of FASTRIDE to those of SAX, which uses quantization bins based on the standard normal distribution instead of an empirical distribution. Based of the results, it appears that using the quantiles to build both the symbolization and the distance measure allows us to adapt it to the data set of interest, and to detect variations between signals that are not captured by the fixed MINDIST costs.

We also note that both FASTRIDE and ASTRIDE benefit from the newly introduced D-GED distance, which offers nice performances on this classification task.

IV.4.2 Reconstruction task

In this section, we investigate the performances of our approaches on a reconstruction task.

IV.4.2.1 Experimental setup

Data mining task and competitors Our ASTRIDE and FASTRIDE representations, SAX, 1d-SAX, SFA, and ABBA are compared on a reconstruction task. Except for ABBA,

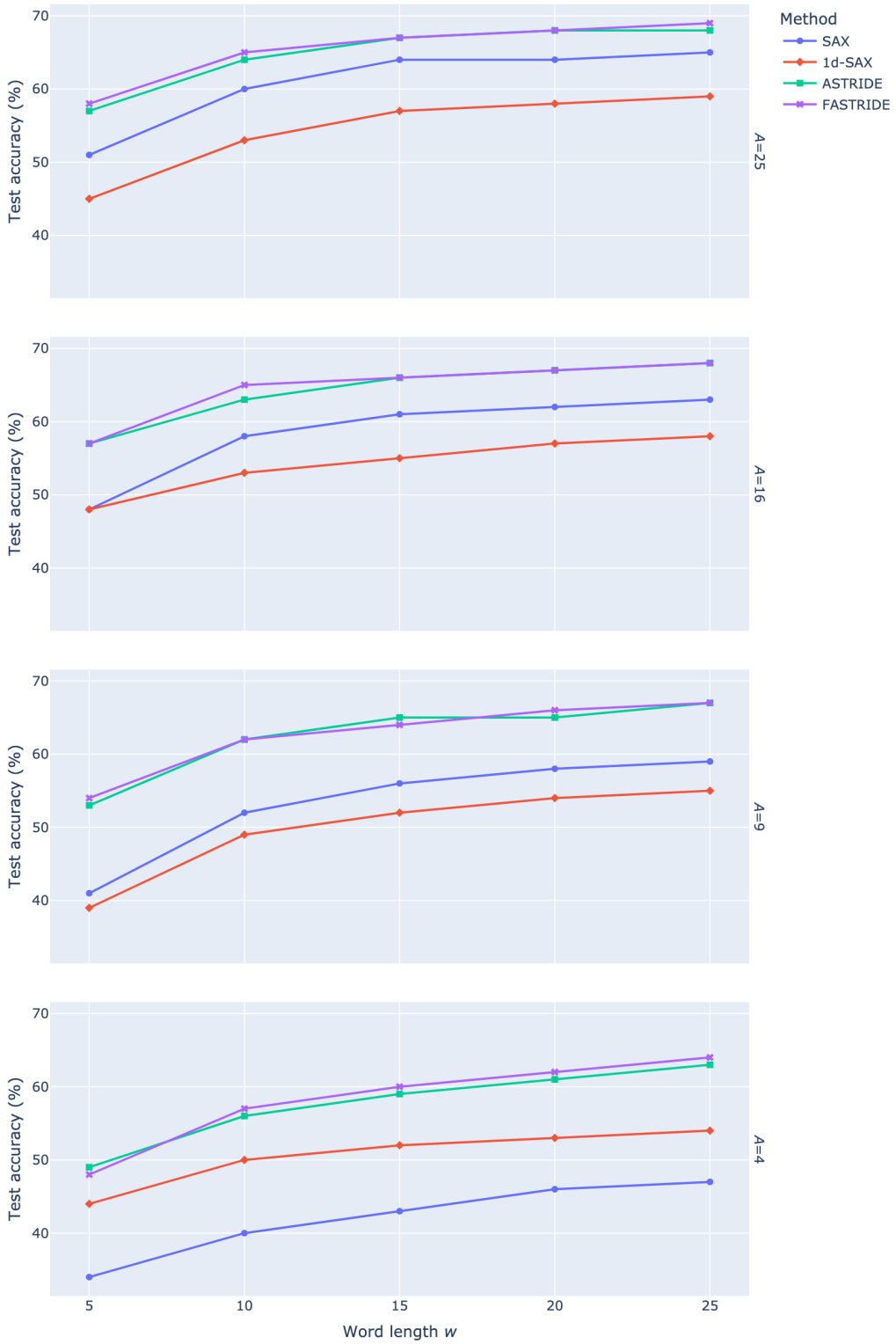


Figure IV.5: Accuracy of SAX, 1d-SAX, ASTRIDE, and FASTRIDE on the classification task versus the word length w, for several values of alphabet size A, averaged on 86 data sets from the UCR Time Series Classification Archive.

the papers about SAX, 1d-SAX, and SFA do not tackle signal reconstruction. However, it is easy to infer a reconstruction procedure for these methods. For SAX and 1d-SAX, the sample values on each segment of the reconstructed signal are based on the Gaussian bins. For SFA, the reconstructed signal is the Fourier reconstruction based on the quantized Fourier coefficients.

Hyperparameters The alphabet size is fixed to $A = 9$ for all methods, and the scaling parameter of ABBA is set to $scl = 1$, as is done in [EG20a][3]. The word length w results from the choice of the tolerance tol in ABBA. For fair comparison, this value of w is determined by running ABBA with a fixed value of tol. To compare the different approaches, we apply a protocol inspired from the ABBA paper [EG20a]. For each signal, ABBA is first run with a low tolerance $tol = 0.05$ and it returns the number of segments w to approximate the original signal at tolerance tol. If $w \geq n\tau_t$, where n is the length of the original signal and τ_t is a target memory usage ratio, we successively increase tol by 0.05 and rerun ABBA until $w \leq n\tau_t$. This last value is denoted by w_e. As in [EG20a], we exclude all signals leading to $w_e < 9$, because we choose $A = 9$.

We run this protocol for $\tau_t \in \{5\%, 6.7\%, 10\%, 16.7\%, 20\%, 25\%, 33.3\%\}$. Unlike ABBA, which works signal by signal, SFA, ASTRIDE, and FASTRIDE work on a whole data set, so their input word length w is the same for all signals in the data set. For the latter methods, we set w equal to the average $\overline{w}_e$ of the w_e's obtained for each signal by ABBA. As a result, each data set and each value of τ_t is associated with a word length $\overline{w}_e$.

In most cases, the empirical memory usage ratio $\tau_e = \overline{w}_e/n$ is smaller than the target memory usage ratio τ_t. The values of τ_t and their corresponding τ_e (averaged on all signals irrespective of their data set) are displayed in Table IV.5. In the protocol, if it is not possible to compress a signal at a given tol, τ_t, and A, then the whole data set is excluded from our benchmark, which explains why there are different numbers of compatible data sets in Table IV.5.

Evaluation of the tasks The evaluation metrics of the reconstruction task are the Euclidean and DTW (Dynamic Time Warping) which is robust to time-shifts. A data set of N signals is transformed into N symbolic sequences, then these N symbolic sequences are reconstructed. For each signal, we compute the reconstruction error: the distance between the original signal and its reconstruction. Recall that, for all methods, each signal is first centered and scaled to unit variance. The reconstruction error is between the scaled original signal and its reconstruction, as the normalization is important when conducting benchmarks [KK03].

Moreover, because we noticed that the SAX and 1d-SAX implementations from `tslearn (v0.5.2)` [Tav+20] poorly handles the last samples of the reconstructed signals when n is not divisible by w, the reconstruction error is computed between the truncated signals: both the original and reconstructed signals (for all methods) are truncated so that their length is $\lfloor n/w \rfloor w$.

Data sets As for the classification task, we use as input the UCR Time Series Classification Archive. The scope of our comparisons is restricted to equal-size data sets of length at least 100 from the UCR Times Series Classification Archive [Dau+19].

[3] Note that we obtain similar reconstruction error results with $scl = 0$.

Table IV.5: Empirical protocol to set the word length w per data set for the reconstruction benchmark, given a target memory usage ratio τ_t for ABBA.

Target memory usage ratio τ_t (%)	Empirical memory usage ratio τ_e (%)	Number of compatible data sets
5	4.0	44
6.7	5.3	49
10	7.6	56
16.7	11.7	64
20	13.4	65
25	15.8	67
33.3	19.3	69

IV.4.2.2 Results

Figure IV.6 displays the reconstruction error versus the memory usage ratio, averaged over the compatible data sets, for several methods, and a fixed alphabet size $A = 9$. We observe that, for both the Euclidean and DTW errors, SFA obtains the best performances, followed by ASTRIDE.

To complement the use of the mean in Figure IV.6, the box-plots in Figure IV.7 show the spread of the reconstruction errors and the outliers, for a fixed target memory usage ratio $\tau_t = 6.7\%$. The most extreme outliers are generated by ABBA, which shows that this method is not robust. A possible explanation is that ABBA is very sensitive to noisy signals because of its piecewise linear approximation step and its use of the quantized increments. On the contrary, SAX, 1d-SAX, SFA, ASTRIDE, and FASTRIDE seem quite robust.

Influence of the memory usage ratio For SAX, 1d-SAX, SFA, ASTRIDE, and FASTRIDE, the reconstruction error decreases as the memory usage ratio increases. Indeed, as the memory usage ratio increases, more segments are allowed in the symbolic representations, resulting in a higher quality of the reconstruction.

For very small memory usage ratios ($\tau_e \leq 6\%$), SFA and ASTRIDE have similar performances. Moreover, when $\tau_e \leq 6\%$, according to the DTW error, ABBA performs better than ASTRIDE and FASTRIDE. However, this observation is challenged by the box-plot in Figure IV.7, which shows that ABBA reaches very large errors and is much less robust than ASTRIDE. Moreover, the empirical memory usage ratio $\tau_e = \overline{w}_e/n$ in Figure IV.6 does not take into account the total memory usage: it ignores the dictionary of symbols and the fact that the reconstruction is done on a data set of signals (and not on a single signal). As emphasized in Section IV.2.3.3 and Section IV.3.4, the total memory usage of ABBA is much larger than those of ASTRIDE and FASTRIDE, because ABBA does not share a dictionary of symbols across signals.

Importance of the adaptive approaches According to Figure IV.6, ASTRIDE achieves better results than FASTRIDE, thus showing the relevance of the adaptive segmentation on the reconstruction task. Indeed, the segmentation phase allows to focus on the

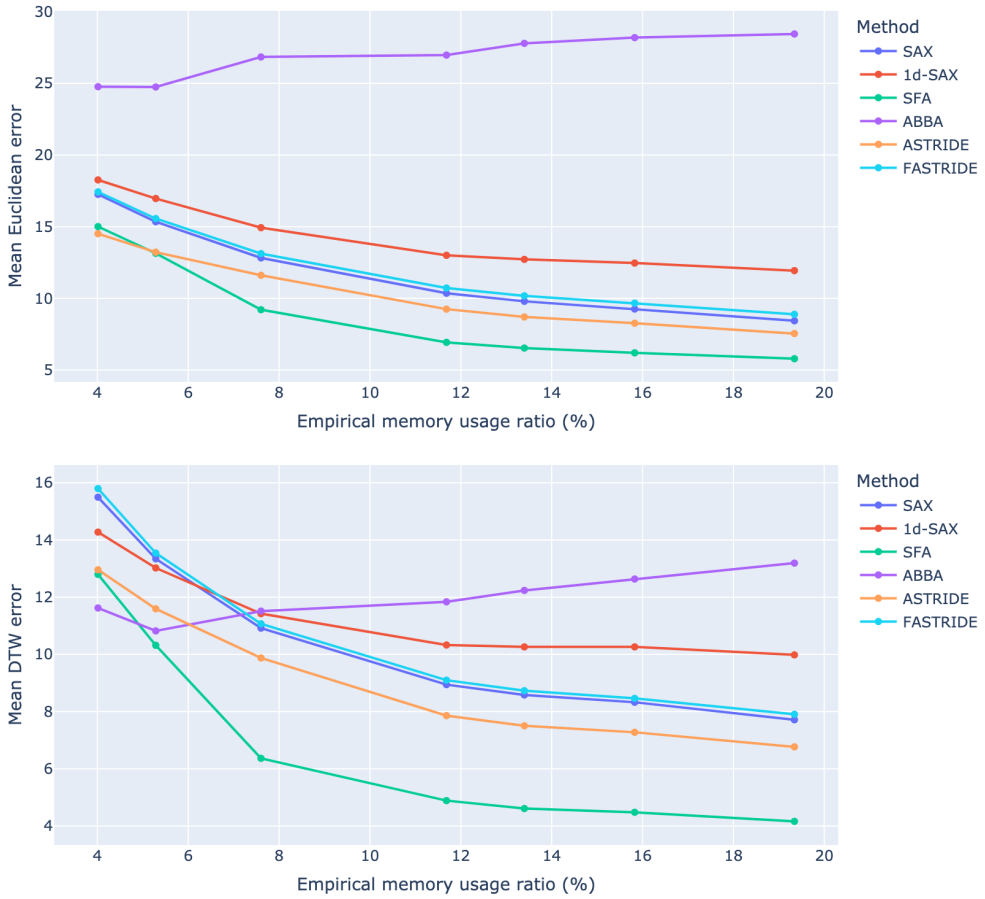


Figure IV.6: Reconstruction benchmark for several methods and several empirical memory usage ratios τ_e, averaged over the signals from various data sets from the UCR Time Series Classification Archive, for $A = 9$.

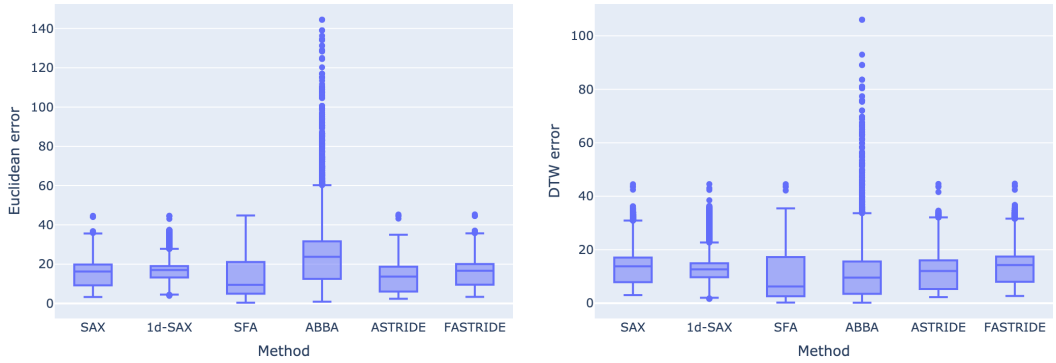


Figure IV.7: Box-plots of the reconstruction benchmark for several methods over the signals of 49 data sets from the UCR Time Series Classification Archive, for $A = 9$ and $\tau_t = 6.7\%$ (leading to $\tau_e = 5.3\%$). For both box-plots, the range of the y-axis is limited to the 99.99% quantile of the reconstruction error for visualization purposes.

events of interest, which are thus correctly reconstructed. Dedicating a memory size for the fine encoding of these events thus seems to be a good strategy to compress the information contained in the signals. Adaptive quantization based on quantiles does not appear particularly useful for signal reconstruction, as FASTRIDE performs similarly to SAX.

Comparison of the methods on a single signal Figure IV.8 gives an example of reconstruction of a single signal from the UCR Time Series Classification Archive, for several methods. Contrary to ASTRIDE, the change-points from ABBA are not exact but approximated from the cluster centers. Thus, they are not precise in the reconstruction phase, which explains why ABBA behaves better with the DTW error, which allows for time-shifts, than with the Euclidean error. Note that the authors of ABBA emphasize that their method does not focus on approximating the signal values at the exact timestamps, but rather on capturing the overall behavior. SFA tends to provide accurate global – rather than local – reconstruction: as shown in Figure IV.8: depending on the data set, this property can be an advantage or a drawback. Regarding ASTRIDE, we can see that the segmentation phase allows us to focus on the phenomenon of interest in the signal, thus to devote more memory to the encoding of salient events.

IV.4.3 Computational complexity

This section describes an important characteristic of the methods: the computational cost. The processing times of the different methods are compared on the 1-NN classification task applied to the ECG200 data set from the UCR Time Series Classification Archive, and are reported in Table IV.6. We ran the experiments using Python 3.10.6 on a laptop under macOS 13.0.1 with Apple M1 Chip 8-Core CPU and 7-Core GPU. All methods mentioned in Section IV.4.1 are compared (SAX, 1d-SAX, ASTRIDE, and FASTRIDE). For SAX, both our implementation and `tslearn` are tested. Two durations are reported: the time to compute the symbolization for all time series in the data set, and the time to perform the actual 1-NN classification from the symbolized time series.

Table IV.6: Processing times on the symbolization and 1-NN classification on the ECG200 data set (UCR Time Series Classification Archive) composed of 100 training signals and 100 test signals of length $n = 96$, with $w = 10$ and $A = 9$.

Method	Symbolization processing time (s)	1-NN classification processing time (s)
SAX	0.27	0.08
SAX (`tslearn`)	0.02	0.11
1d-SAX (`tslearn`)	0.42	0.21
ASTRIDE	0.30	0.17
FASTRIDE	0.26	0.07

First, as expected, the ASTRIDE symbolization is more time-consuming than the non-adaptive ones (SAX for example). An important remark is that the temporal segmentation is relatively fast: the computation times for ASTRIDE and its variant

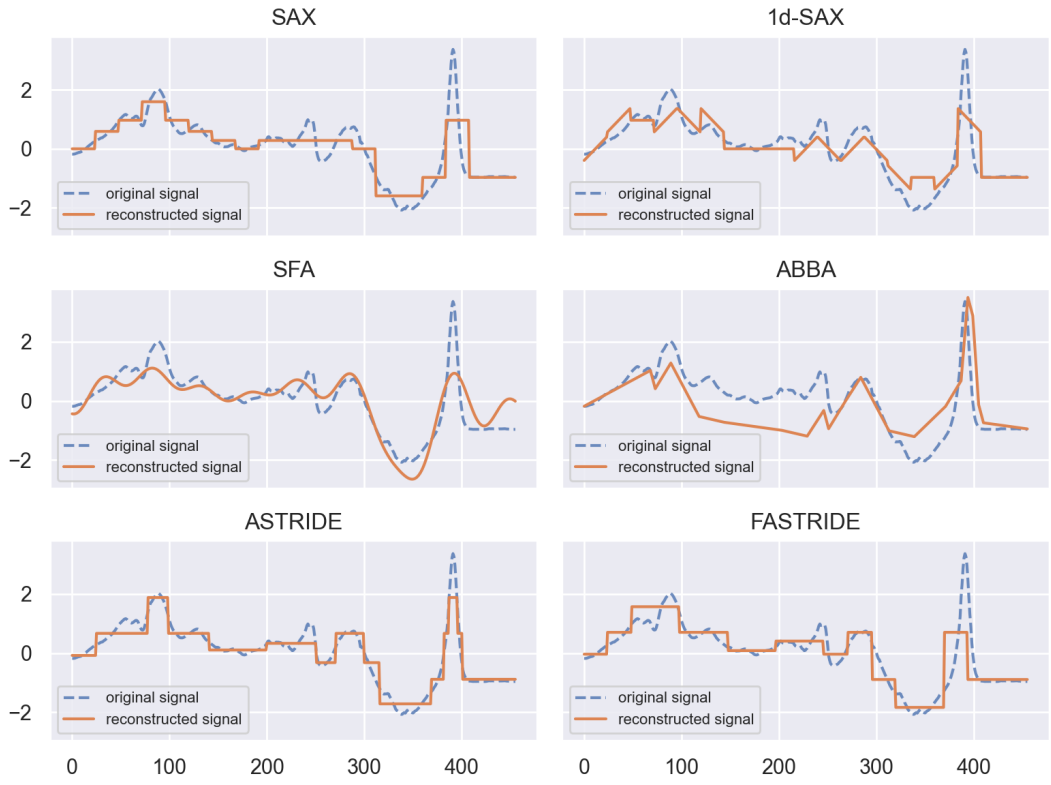


Figure IV.8: Example of reconstruction of a single signal from the Beef data set (UCR Time Series Classification Archive) of original length $n = 470$ for several methods, with $A = 9$ and $\tau_t = 5\%$ leading to $\overline{w}_e = 19$ and $\tau_e = 4.0\%$. The Euclidean error is respectively of 9.9, 10.7, 10.7, 18.1, 6.0, and 11.5 for SAX, 1d-SAX, SFA, ABBA, ASTRIDE, and FASTRIDE respectively. Note that SFA, ASTRIDE, and FASTRIDE are fitted on the whole training set (and not on the displayed signal only). All displayed signals are truncated (so that their length is $\lfloor n/w \rfloor \, w$).

FASTRIDE without adaptive segmentation are quite similar. 1d-SAX is more expensive because it takes into account the mean as well as the slope, then has to combine them.

Second, as far as the classification step is concerned, it appears that the computation of D-GED for ASTRIDE is more expensive than for FASTRIDE because of the replication of the symbolic sequences which makes them longer. For ASTRIDE, using the normalized segment lengths instead of the raw segment lengths helps making the replicated symbolic sequences shorter, but a gap remains in comparison to FASTRIDE. Several improvements could further lower the computation time, such as using more advanced general edit distances (e.g., Weighted Symbols-Based Edit Distance [Bar+10]), which are optimized to deal with redundant series of symbols.

IV.5 Conclusion

We have introduced a new symbolic representation of time series, ASTRIDE, along with its accelerated variant FASTRIDE, as well as a novel distance measure on symbolic sequences, D-GED. ASTRIDE is the only symbolic representation offering adaptive discretization on both the time and amplitude dimension, at the scale of a data set, while having a compatible distance measure and a reconstruction procedure that is memory-efficient. Hence, ASTRIDE combined with D-GED alleviates the main drawbacks of existing symbolic representations. ASTRIDE uses change-point detection of mean-shifts instead of uniform segmentation, and adaptive quantization using quantiles in place of fixed Gaussian bins. Moreover, D-GED is based on the general edit distance which relies on the quantiles and deals with substitution, deletions and insertions, which are not handled by the MINDIST distance measure. In addition, thanks to the multivariate change-points and the quantiles learned from all signals in the data set, the dictionary of symbols is shared across all signals, thus reducing memory usage.

Our experiments show the quality of our symbolic representations. Indeed, both ASTRIDE and FASTRIDE give better accuracies than SAX and 1d-SAX on the classification task, for a same word length. This performance is mainly due to the adaptive quantization based on quantiles and the D-GED distance measure. For the reconstruction task, FASTRIDE and especially ASTRIDE give better errors than SAX, 1d-SAX and ABBA, for a same memory usage ratio. On the reconstruction task, the adaptive segmentation is particularly relevant, thus using ASTRIDE rather than FASTRIDE seems more appropriate.

Chapter V

d_symb: an interpretable distance measure for multivariate signals

In many applications, such as behavioral neurology, researchers have to compare and understand large amounts of multivariate time series in an interactive and interpretable way. We introduce d_{symb}, a novel distance measure for comparing multivariate non-stationary signals. Unlike most distance measures on multivariate signals, d_{symb} takes into account their non-stationarity thanks to a symbolization step. This step is based on a change-point detection procedure that splits a non-stationary signal into several stationary segments, followed by quantization using K-means clustering. The proposed distance measure leverages the general edit distance and is applied to the symbolic sequences. The advantages of d_{symb} are shown on three data sets of physiological signals. Moreover, we describe an online tool, called the d_{symb} playground, that we implemented to allow other researchers to apply d_{symb} to their uploaded data.

Contents

V.1 Introduction

In numerous applications, large data sets of time series are collected and then compared with each other. For instance, in behavioral neurology, subjects (human or animal) with various neurological conditions are monitored during long time periods.

Then, researchers want to compare subjects or groups of subjects or assess a subject's evolution (longitudinal study) using those signals. The recordings are often multivariate, as they are collected from one or several sensors. Moreover, because of operational constraints, subjects are monitored for prolonged periods, yielding large time series. For example, when monitoring elderly patients in hospitals [Jun+21], setting up the sensors is cumbersome and can only be done once. As a result, signals also contain several different regimes of interest. In such contexts, researchers focus on the dynamic or chronology of those regimes: in [Jun+21], medical doctors want to evaluate their patients' gait during a 10-minute protocol of several simple activities, e.g., walking and climbing stairs. In addition, they need intuitive and immediate feedback in order to make a diagnosis. An informative distance measure between these kinds of signals should consider this non-stationary structure. Such a setting –comparing long multivariate time series with switching regimes in a fast and interpretable way– is found in a great number of biomedical applications as well as industrial contexts.

Related work. Dynamic Time Warping (DTW) [BC94], which is arguably the most popular elastic distance measure, is often used in such situations [Shi+23]. One important feature of DTW is its robustness to time warping, that is, a contraction or dilatation of the time axis. First defined for univariate signals, it was recently extended to multivariate signals. Two popular approaches are often used in practice: the independent and dependent strategies [SY+17]. In the independent strategy, the univariate DTW is applied to each dimension separately, and the resulting distances on each dimension are summed. The dependent strategy considers the multivariate series as a single series in which each timestamp is associated with a single multidimensional point. The DTW scheme is then applied using Euclidean distances between the multidimensional points of the two series. Other variants of multivariate DTW exist (see Chapter III). One such variant is Derivative DTW [KP01] which applies DTW, not directly on the raw signals, but on their first derivative. Another variant, known as Weighted DTW [JJO11], uses custom weights to avoid large warpings. Both Derivative DTW and Weighted DTW can be combined into a variant called Weighted Derivative DTW. The time complexity of DTW-based distances is $\mathcal{O}(dmn)$ where m and n are the lengths of the compared time series and d is the dimension. In addition to the high computational cost, DTW-based distances are not always easy to interpret. For long signals, the warping path between two signals is as complex as the original data.

Another category of methods is based on symbolic representations (reviewed in Chapter II), on which a distance measure is defined. Symbolization transforms a real-valued signal y of arbitrary length n into a discrete-valued signal $\hat{y}$ of smaller length $w \leq n$, called a symbolic sequence. A common symbolic representation for univariate signals is *Symbolic Aggregate approXimation (SAX)* [Lin+03; Lin+07] that has successfully been used in several data-mining tasks such as classification [Lin+07; Ngu+19], clustering [Lin+07] or indexing [Cam+10]. However, extending a symbolic representation to the multivariate case remains a challenge. A naive approach consists in symbolizing each dimension independently and creating meta-symbols that are combinations of the first symbols. The total number of symbols is then A^d, where A is the number of symbols for each dimension and d is the number of dimensions. However, this approach does not scale well when d increases. For instance, MSAX [AVC20], which is a multivariate extension of SAX, is applied to data sets of trajectories ($d = 2$) only. As a comparison, the physiological signals that we use in the experiments can have

hundreds of dimensions. Large alphabets are also less interpretable, even for small values of A and d.

Finally, distances based on feature extraction often lose the temporal aspect that is essential for time series. Indeed, such methods (reviewed in Section III.4.3) extract features that are mostly based on the frequency of symbolic words [SM13; Sch15]. For example, in BOSS [Sch15], symbolic words are computed on overlapping sliding windows, then signals are compared based on their histograms of symbolic words.

Contributions. We propose a symbolic representation, with a compatible distance measure on its symbolic sequences, for a data set of multivariate time series that can be non-stationary, called d_{symb} [CTO23a]. Unlike most distance measures on multivariate signals, d_{symb} takes into account their non-stationarity thanks to a symbolization step. This step is based on a change-point detection procedure that splits a non-stationary signal into several stationary segments, followed by quantization using K-means clustering. The proposed distance measure leverages the general edit distance with custom costs. The advantages of d_{symb} are shown on three data sets of physiological signals. Experiments show how interpretable the symbolization is: a single glance at the symbolic sequences provides an immediate and informative description of a data set. Moreover, compared to nine multivariate elastic distances on a clustering task, d_{symb} preserves a competitive performance while being several orders of magnitude faster than the other methods. With these desirable characteristics, we developed the d_{symb} playground [Com+24b], an online tool that allows researchers to apply d_{symb} to their uploaded data.

Organization of the chapter. In the remainder of this chapter, we describe our d_{symb} method in Section V.2 and apply it to three multivariate physiological data sets in Section V.3. Finally, in Section V.4, we present our online tool, the d_{symb} playground.

V.2 The d_symb method

Our method, denoted d_{symb}, is a novel distance measure on multivariate signals of possibly different lengths. This distance measure is designed to handle non-stationarity, to be interpretable, and to run fast to allow interactivity. The symbolization is computed using the same steps as ASTRIDE: signal segmentation, feature extraction, and quantization. However, there are several noteworthy modifications in the segmentation and the quantization steps. Each step, of the symbolization and the resulting distance, is briefly described in the following. Let $y = (y_1, \dots, y_n)$ be a multivariate signal.

1. Each multivariate signal is partitioned into stationary segments using a change-point detection procedure. Signals are treated individually, unlike ASTRIDE, which deals with the whole data set simultaneously.

   Contrary to ASTRIDE, where the number of change-points is fixed by the user, in d_{symb}, the number of changes is controlled by a penalty parameter denoted λ. The change-point estimates $\hat{t}_1, \dots, \hat{t}_{\hat{w}}$ ($\hat{w}$ is the number of detected changes)

are the minimizers of a discrete optimization problem:

$$\left(\hat{w}, \hat{t}_1, \ldots, \hat{t}_{\hat{w}}\right) = \underset{(w,t_1,\ldots,t_w)}{\arg\min} \sum_{k=0}^{w+1} \sum_{t=t_k}^{t_{k+1}-1} \|y_t - \bar{y}_{t_k:t_{k+1}}\|^2 + \lambda w, \tag{V.1}$$

where $\bar{y}_{t_k:t_{k+1}}$ is the empirical mean of $\{y_{t_k}, \ldots, y_{t_{k+1}-1}\}$ and $\lambda > 0$ is a penalization parameter. (By convention, $t_0 := 0$ and $t_{w+1} := n$.) The penalized formulation (V.1) seeks a compromise between the reconstruction error given by the sum of quadratic errors and the complexity given by the number of change-points. Problem (V.1) is solved using the *Pruned Exact Linear Time* (PELT) algorithm [KFE12], which is shown to have $\mathcal{O}(n)$ complexity under the assumption that the segment lengths are randomly drawn from a uniform distribution.

Intuitively, the λ parameter penalizes the introduction of a new change-point: when λ is small, many change-points are detected. Once the user chooses a penalty λ, the segmentation procedure returns the segment bins and the estimated number of segments.

2. Each signal segment is summarized by its mean vector.

3. All segments of all signals are then pooled together and assigned a symbol through K-means clustering. The user-defined alphabet size A is the number of clusters. Each signal segment is symbolized by its cluster. A complete signal is symbolized by the sequence of symbols, yielding a symbolic representation. This is different from ASTRIDE, which uses quantiles to discretize univariate time series.

4. The final distance d_{symb} is computed as the general edit distance between the symbolic version of the signals. The distance is the same as for ASTRIDE, except that substituting one symbol for another has a cost equal to the Euclidean distance between their associated cluster centers.

The differences with ASTRIDE are the following: the time series are segmented individually with a penalty (in Step 1), and K-means is used for the quantization (Step 3). As a result, compared to ASTRIDE that has the same segment bins for a whole data set, d_{symb} loses some compression properties, but the resulting segmentation is better adapted to each multivariate signal, and the symbolic representation contains more information, as will be seen in the experiments (Section V.3). As each symbol is related to a cluster; each cluster center represents an average behavior that is encoded by the symbol. We will use this property extensively in the experiments to interpret the symbols produced by d_{symb}.

The theoretical complexity of the segmentation step is $\mathcal{O}(n)$. Similarly to ASTRIDE, each symbolic representation is down-sampled. Therefore, the general edit distance is applied to much shorter sequences than the raw time series. Since the complexity of the general edit distance between two sequences is $\mathcal{O}(mn)$ where m and n are the sequences' lengths, this produces an important speed-up.

As for the calibration, the penalty parameter can be derived using the well-known Bayesian Information Criterion (BIC) [Yao88]. The alphabet size A is task-dependent but should be chosen small enough for interpretation. In our experiments, we found that with more than 10 different symbols, the analysis becomes difficult.

V.3 Applications of d_symb

In this section, we apply our distance measure d_{symb} on several data sets of multivariate physiological signals. d_{symb}'s ability to separate clusters is quantitatively assessed on the JIGSAWS data set. Then, we illustrate how the symbolic representations of d_{symb} can provide insights into two behavioral neurology tasks: human gait analysis and upper-limb movement analysis. An open source GitHub repository[1], written in Python, is available.

V.3.1 Application on the JIGSAWS data set

Data and task. In order to evaluate the performance of d_{symb}, we apply it to a clustering task on the real-world JIGSAWS data set [Gao+14]. In this data set, eight surgeons have been monitored while performing "simple" surgical tasks with robotic arms and grippers. The signals are the kinematic data, e.g. positions and angular velocities, of the surgical tools that they manipulated during the trial. Here, we consider two surgical gestures: *Knot Tying* (39 multivariate time series) and *Needle Passing* (40 multivariate time series). The time series have 76 dimensions, are sampled at 30 Hz, and last around 1.5 minutes on average. The objective is to use d_{symb} to discriminate between the two gestures.

For comparison, nine other distances are applied to the time series: DTW (`DTW`) [BC94], Derivative DTW (`DDTW`) [KP01], Weighted DTW (`WDTW`) [JJO11], Weighted Derivative DTW (`WDDTW`) [JJO11], Move-Split-Merge (`MSM`) [SAD13], Time Warp Edit (`TWE`) [Mar09], Longest Common Sub-Sequence (`LCSS`) [VKG02], Edit distance with Real Penalty (`ERP`) [CN04], and Edit Distance on Real sequence (`EDR`) [COO05]. Each of these distances is extended to its multivariate version by using the dependent strategy.

Metric and calibration. For a given distance function, the distance between all pairs of signals is computed and fed to an agglomerative clustering algorithm, which sequentially merges similar clusters together. The number of final clusters is set to two. Note that this method is unsupervised. The result is compared to the true gesture labels by using the *Adjusted Mutual Information (AMI)*. For two partitions $U = \{U_1, U_2, \dots\}$ and $V = \{V_1, V_2, \dots\}$ of $\{1, 2, \dots, N\}$, the mutual information is defined by

$$\sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|}.$$

The AMI is an adjustment for chance of the mutual information; it is equal to 1 if $U = V$ and around 0 if the two partitions are random. All signals are centered and each dimension is scaled to unit variance. To calibrate the parameters of d_{symb} (alphabet size A and penalty λ), we use a training set of 10 signals. The AMI is computed on the remaining 69 signals. Results are obtained using a computer with Quad-Core Intel Core i5 (2.3 GHz) with 16 GB of RAM.

Results. The clustering performances and execution times are reported on Table V.1. Overall, `WDTW` and `WDDTW` have the best AMI, followed by d_{symb}. Remaining distances

[1] https://github.com/sylvaincom/d-symb

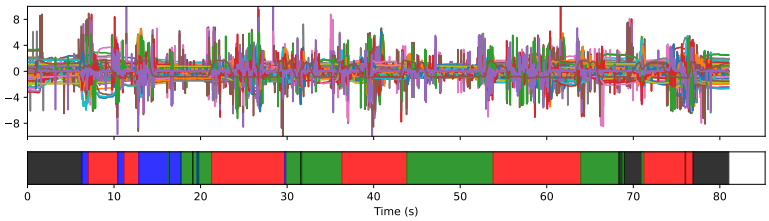
Table V.1: Results on the JIGSAWS data set

Distance	AMI	Timings
d_{symb}	0.21	38 s
DTW	0.19	35 min
DDTW	-0.00	35 min
WDTW	0.34	36 min
WDDTW	0.34	36 min
MSM	0.13	4 h 48 min
TWE	-0.00	1 h 46 min
LCSS	-0.00	37 min
ERP	-0.00	38 min
EDR	-0.01	38 min

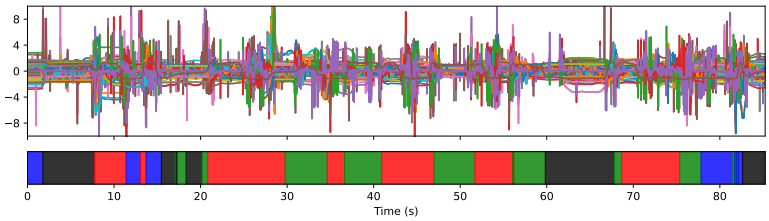
(DDTW, MSM, TWE, LCSS, ERP, EDR) have markedly worse AMI than d_{symb}. While our approach has not the best clustering score, it remains competitive –for instance, it is on par with DTW– and more importantly, it does it for a fraction of the time needed by other methods. Indeed, on average, it takes less than a minute to process 79 signals of dimension 76 and length 2700. Even though our distance has a worst-case complexity, which is quadratic in the number of samples (because of the change-point detection), in practice, it is much faster. Then, after the symbolization, the Levenshtein distance is applied to far shorter signals. To summarize, d_{symb} strikes a trade-off between clustering performance and speed, making it adapted for interactive but still informative use.

Furthermore, the symbolic representation associated with d_{symb} can be insightful. See, for instance, Figure V.1 where three signals from JIGSAWS are shown. Evidently, the raw multivariate time series are difficult to interpret. However, it is easier to observe on the symbolic representations that the first two signals (Figure V.1-a and Figure V.1-b) are similar: both roughly start and end with Symbol 4, and have an alternation of Symbol 1 and 2 in the middle. The third signal (Figure V.1-c) has more occurrences of Symbol 3 during the trial. Note that the first two signals represent the same gesture (Needle Passing), while the third represents Knot Tying.

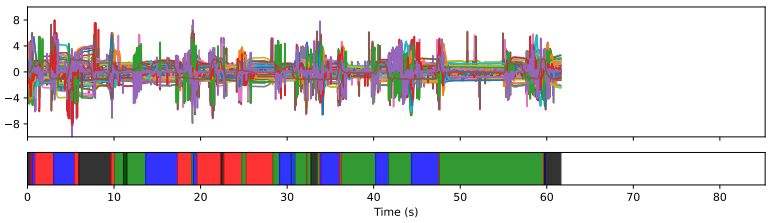
Moreover, it is possible to interpret each symbol. Since symbols are associated with a cluster of signal segments, we can study the clusters' centroids to understand the average behavior that the symbols account for. For each cluster's centroid, Figure V.2 shows the average positions, linear velocity, and angular velocity of the robotic arms (left and right) used by the surgeon. For instance, looking at Symbol 4, we can observe that it is associated with motions where both arms are far from each other, compared to other symbols. Moreover, the right arm has the largest linear velocity and the left arm has the largest (in absolute value) angular velocity. Thus, Symbol 4 is associated with a gesture where both arms are distant, the right arm moves across the space and the left arm does not move as much but instead turns. We can similarly analyze Symbols 1 and 2. First, they are located in different parts of the space (Figure V.2-a), and in Symbol 1, the arms are close together, which is not the case of Symbol 2. Second, looking at the angular velocity (Figure V.2-c), those two symbols have the largest velocities (in absolute value) for the right arm, with opposite signs. Hence, when alternating between Symbols 1 and 2, the robotic arms change location and distance from each other, and the right arm changes rotation sign.



(a)

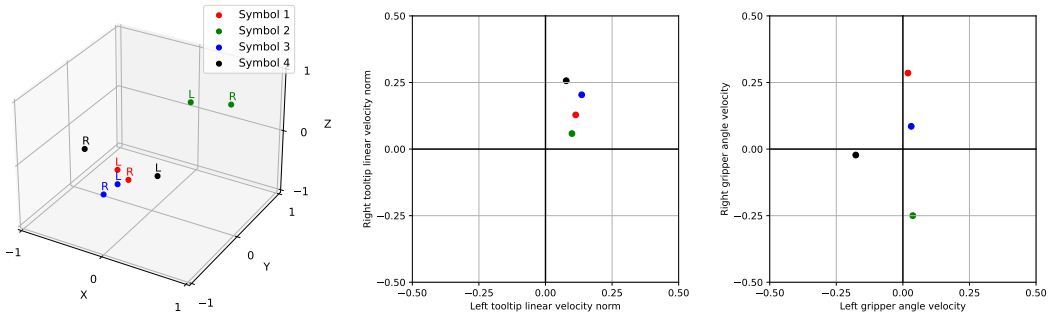


(b)



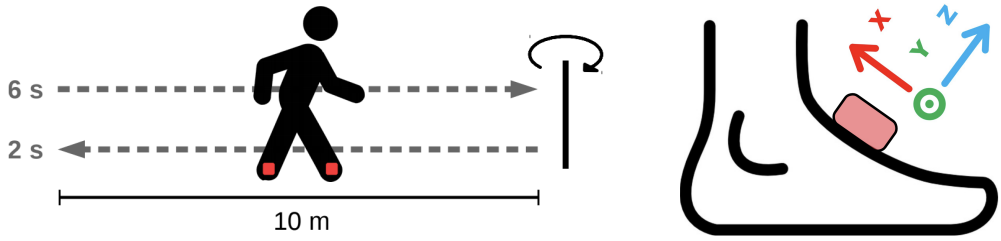
(c)

Figure V.1: Three signal examples from the JIGSAWS data set. Signals in (a) and (b) are close to each other according to d_{symb}, and are far from (c). Below the raw signals, the symbolic representation associated with d_{symb} is shown. There are four symbols: Symbol 1 ■, Symbol 2 ■, Symbol 3 ■, Symbol 4 ■. In (a) and (b), the surgical gesture is Needle Passing; in (c), it is Knot Tying.



(a) (b) (c)

Figure V.2: (a) Positions (x, y, z) of the left (L) and right (R) robotic arms for each symbol centroid. (b) Idem for the linear velocity norm. (c) Idem for the angular velocity. All features are expressed in normalized units since all signals are centered and scaled before applying d_{symb}.



(a) (b)

Figure V.3: (a) Schematic protocol recorded by the sensors that are located with red squares [Tru+19]. (b) Axis definition in the sensor's frame.

V.3.2 Application on the human locomotion data set

Human gait is a complex phenomenon that can be altered by many neurological disorders. Consequently, medical researchers try to objectively measure gait characteristics to analyze patients' walking patterns [Oud+18]. The human locomotion data set [Tru+19] consists of signals from subjects wearing accelerometers on their feet (sampling frequency: 100 Hz). All subjects underwent the same protocol (depicted in Figure V.3-a): standing still, walking 10 meters, then walking back to where they started, and standing still. For this illustrative study, we only keep the angular velocity around the (Oy) axis (see Figure V.3-b for axis definition) of each foot.

Since locomotion is an activity that has a strong periodic component, it is common in the literature to process such signals in the time-frequency domain. For each univariate gait signal, we compute its Short Time Fourier Transform (STFT), with a window length equal to 300 samples (3 seconds) and an overlap of 299 samples. Only the 0–5 Hz frequency band, where phenomena of interest are contained, is kept. The d_{symb} symbolic representation is computed on those 16-dimensional signals. The number of symbols is set to $A = 5$, and the change-point detection penalty is set to $\lambda = 2$.

Four signal examples are shown in Figure V.5. Note that since the subjects perform the protocol at different speeds, each recording has a different duration. The first three time series are from healthy subjects, and the last one is from a subject with neurological impairment. Simply looking at the raw signals does not provide any insight into the gait dynamics. However, as we shall see, the symbolic representation is better adapted to see that all four subjects walk differently. First, note that since d_{symb} is computed on the time-frequency representation of the raw signals, cluster centroids are actually power spectral densities, which are shown in Figure V.4. Symbol 3 represents the absence of movement (subject standing still). Symbol 1 represents walking with lower-amplitude footsteps. The remaining symbols represent different walking patterns.

From the symbolic representations of Figure V.6, we clearly see the activity sequence of the protocol (standing, walking, turning, walking, standing). Moreover, walking often starts and ends with one or several low amplitude footsteps (Symbol 1), a well-known fact in gait analysis studies [BM+16]. The U-turn strategies, which are an informative biomarker [BM+17], are also different even among healthy subjects. For instance, Subject 1 has a short U-turn; Subject 2 has an asymmetric U-turn with a pause on the left foot; and Subject 3 does long pauses while turning. When looking at the walking phases, they are homogeneous for Subject 1 and less consistent for the other

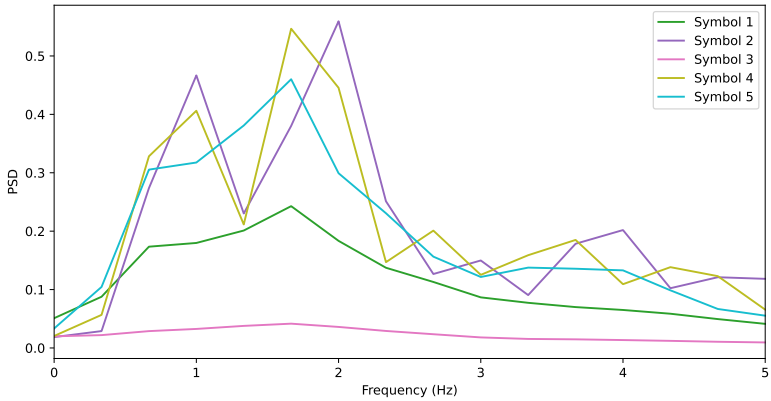


Figure V.4: Power Spectral Density (PSD) for each symbol centroid of the symbolic representation of Figure V.6.

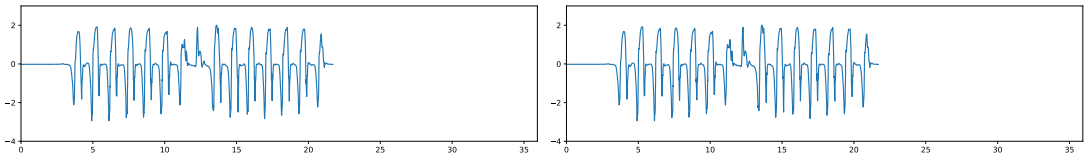
subjects. We observe a change between walking forward and walking back for Subject 3 that might result from tiredness. Here, in this limited sample, the main distinction between healthy and neurologically impaired subjects is the irregularity of the walking phase. Indeed, Subject 4 alternates more often between Symbols 4 and 5 during the walk. All those observations are made possible thanks to the symbolic representation and the interpretability of the symbols.

V.3.3 Application on the upper-limb movement analysis

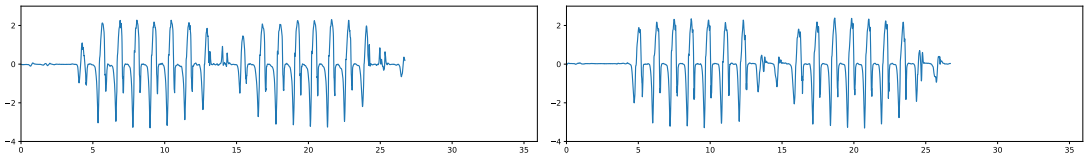
Similarly to human gait, upper-limb movement is an extensively studied biomechanical and neurological phenomenon that can be modified by many different medical disorders. Quantifying such movement is a central question, and researchers often use 3D position sensors to assess several key features, e.g., smoothness or symmetry. The armCODA data set [Com+24a] contains around 2.5 hours of multivariate time series collected from healthy subjects performing pre-defined simple movements. An online interactive tool[2] is available to download and explore this data set. More precisely, the subjects were asked to perform several types of movements, including elevation movements of the right arm, the left arm, and both arms simultaneously. In order to track these movements, 34 Cartesian Optoelectronic Dynamic Anthropometer (CODA) motion system 3D position markers are placed on the upper-limb of the participants, each marker recording its positions over time in the 3D space.

An example of signal is shown in Figure V.7. For this trial, the subject remained seated and raised both arms vertically from a resting position (arms along the body) to above the head. This movement was repeated three times. The signal dimension is $34 \times 3 = 102$. In its raw form, the signal is not interpretable, therefore, we use the symbolization procedure from d_{symb} to gain more insights. Here, the number of symbols is manually set to 7. For this illustrative example, we only display the symbolic sequences of a single subject performing four different movements. These movements are shown in Figure V.8. The first symbolic sequence (Figure V.8-a) corresponds to the raw signal in Figure V.7. The three other representations correspond to three other movements: raising both arms while standing (Figure V.8-b), raising the right arm (Figure V.8-c), and the left arm (Figure V.8-d) while standing. On these representations, the

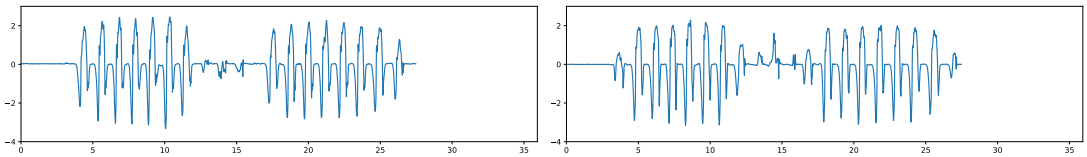
[2] https://www.ipol.im/pub/art/2024/494/



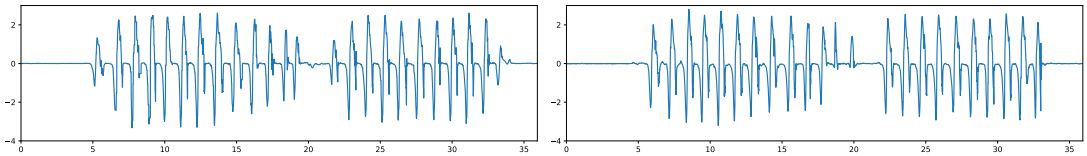
(a) Subject 1



(b) Subject 2

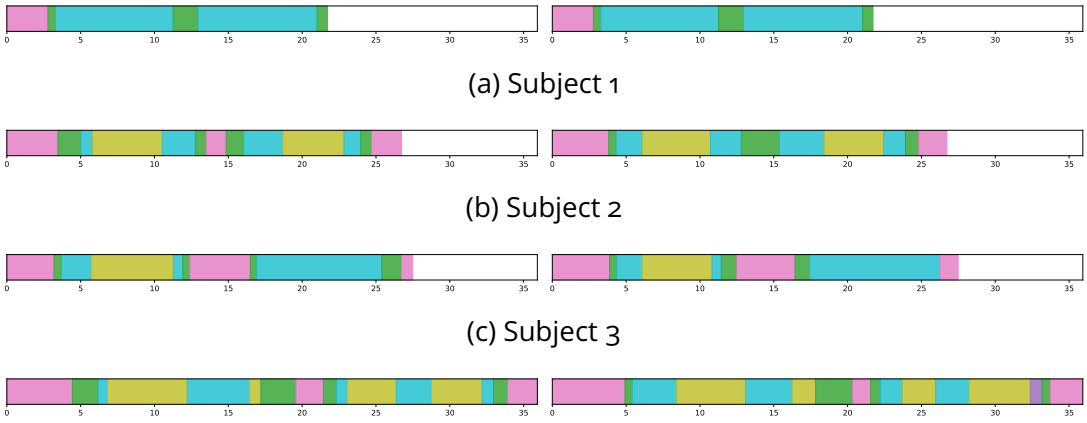


(c) Subject 3



(d) Subject 4

Figure V.5: Signal examples from the Human Gait data set. Each row is one trial with the left foot signal on the left and the right foot signal on the right. The x-axis is the time in seconds. The first three trials are from healthy subjects, and the last one is from a subject with a neurological pathology.



(a) Subject 1

(b) Subject 2

(c) Subject 3

(d) Subject 4

Figure V.6: Symbolic representations of the signals shown on Figure V.5. There are 5 symbols: Symbol 1 , Symbol 2 , Symbol 3 , Symbol 4 , Symbol 5 . The x-axis is the time in seconds.

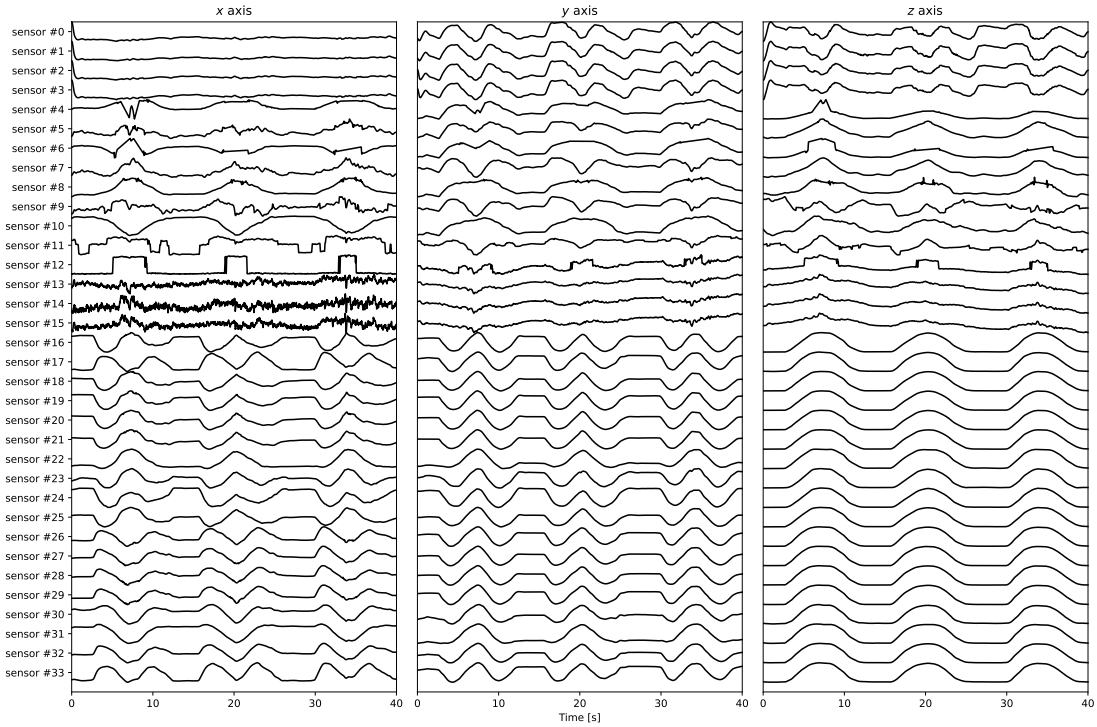
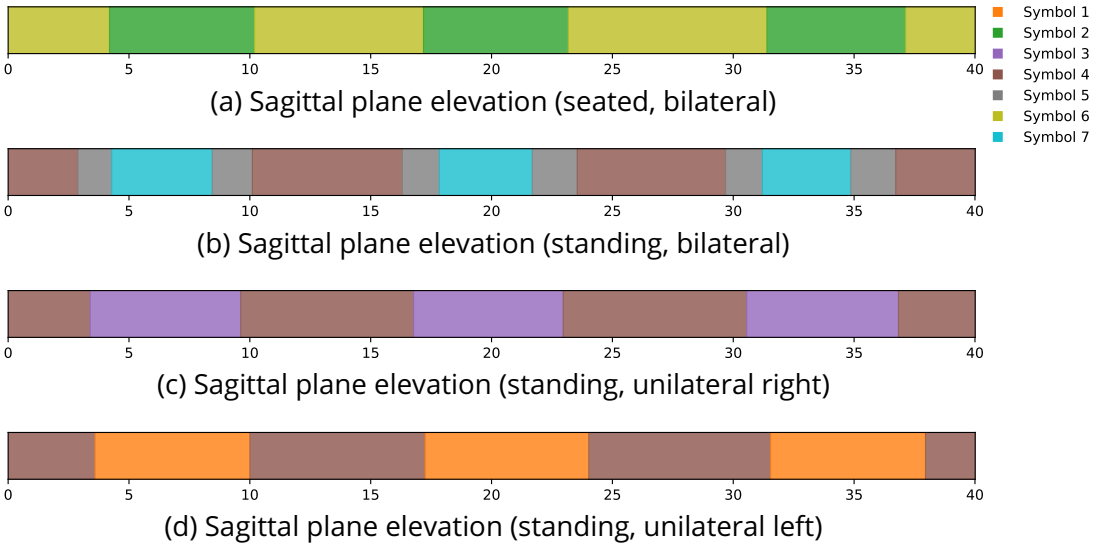


Figure V.7: Raw signal example from the armCODA data set. The movement is the sagittal plane elevation (seated and bilateral).



(a) Sagittal plane elevation (seated, bilateral)

(b) Sagittal plane elevation (standing, bilateral)

(c) Sagittal plane elevation (standing, unilateral right)

(d) Sagittal plane elevation (standing, unilateral left)

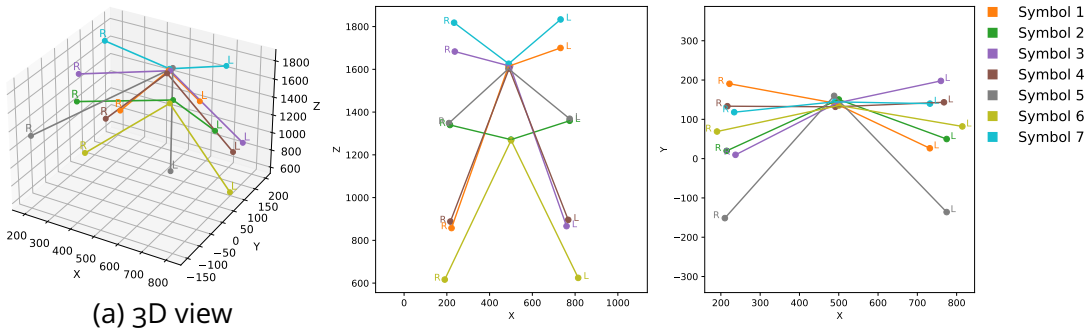
Figure V.8: Symbolic representations of signals from the armCODA data set. They belong to the same subject and each one is the repetition of a single movement. The first representation (a) is the signal shown in Figure V.7.

three movement repetitions (as demanded by the protocol), with in-between rest, can be easily observed. There is a rest symbol for the seated state (Symbol 6) and another for the standing state (Symbol 4). Moreover, each movement has its own symbol, and there is a last symbol (Symbol 5) for an intermediate state between resting and both arms up (Figure V.8-b). Recall that our symbolization procedure is unsupervised but is still able to recover the salient events and classify each state correctly. A closer look at the learned symbols in Figure V.9 confirms this observation. Since the cluster centers computed by d_{symb} are average body positions, it is possible to plot them in the 3D space. (Note that only the average positions of the head and forearms are shown for readability, even though the subject has been monitored with 34 sensors.) The front view is particularly revealing. The two symbols seen on the seated movement –Symbol 2 and 6– have the head sensors at around 1.3m high and the arms either up (Symbol 2) or down (Symbol 6). For the standing position, the head is around 1.6m high, and three positions are easily interpretable: both arms up (Symbol 7), left arm up (Symbol 1), or right arm up (Symbol 4). To summarize, the observation of this representation is sufficient to discriminate between different types of movements and is interpretable.

V.4 The d_symb playground

In this section, we describe the d_{symb} playground, available online[3][4], and built using Python 3.9 and the Streamlit framework [Str]. The d_{symb} playground, summarized in Figure V.10, is a web interactive tool to explore and compare large multivariate time series data sets. This interactive tool allows users to upload and visualize their multivariate time series and their d_{symb} symbolizations using the colorbars. With a single

[3] https://dsymb-playground.streamlit.app
[4] https://github.com/boniolp/dsymb-playground



(a) 3D view (b) Front view (c) Top view

Figure V.9: Positions (x, y, z) (in cm and in the laboratory frame) of the head, left forearm (L), and right forearm (R) for each symbol centroid.

glance at the colorbars, the symbolization provides an immediate and comprehensive understanding of a data set. Users can also visualize the d_{symb} pairwise distance matrix between the symbolic sequences. Furthermore, users can assess the relevance of the d_{symb} distance measure with regards to 9 elastic distance measures, including variants of DTW.

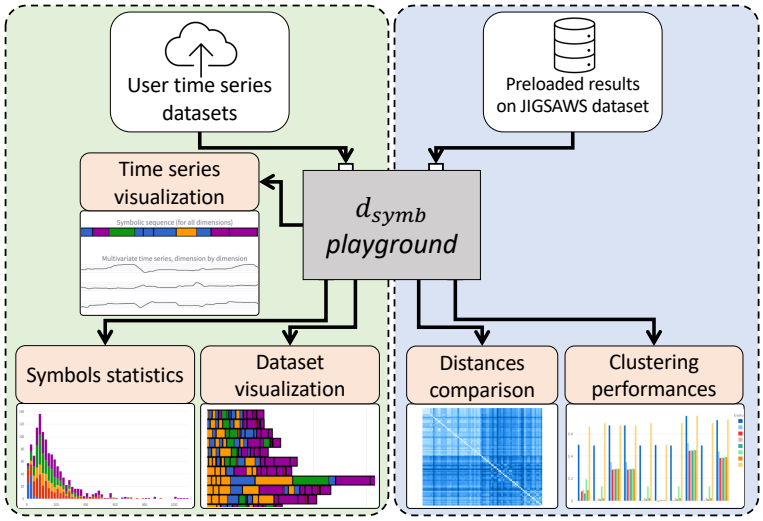
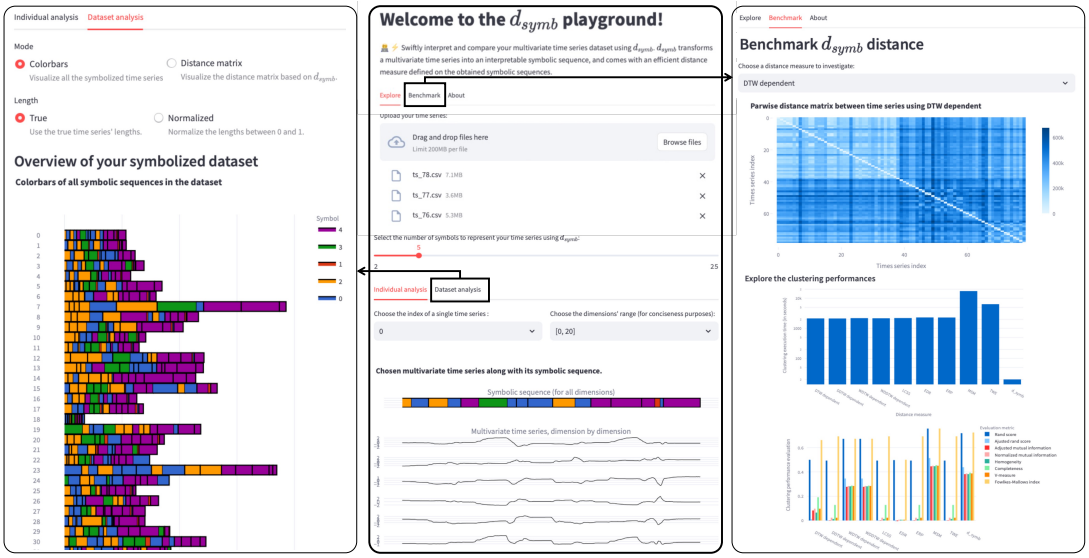


Figure V.10: Summary of the d_{symb} playground's inputs and features.

Our system is based on d_{symb} and inputs a multivariate time series data set. The GUI is composed of three main frames, shown in Figure V.11: the **Individual analysis frame**, the **data set analysis frame**, and the **Benchmark frame**. The individual and data set analysis frames enable users to explore and quickly gain insights thanks to the d_{symb} symbolization. The benchmark frame allows users to assess the performance of the d_{symb} distance compared to 9 existing distance measures on a real-world application.

As shown in Figure V.11(B), for both the individual and data set analysis frames, the user is required to upload their multivariate time series data set and then select the number of symbols to be used in the d_{symb} symbolization. Each multivariate time series must be stored in a Comma-Separated Values (CSV) file of shape `(n_timestamps,`

`n_dim`). The user can choose the number of symbols as an integer between 2 and 25. Then, the d_{symb} computation is performed: the symbolization of all time series, as well as the pairwise distance matrix between the time series, are returned. We now describe the three main frames and their corresponding available actions in more detail.



(A) Dataset analysis frame (B) Individual analysis frame (C) Benchmark frame

Figure V.11: Illustration of the three main frames of the d_{symb} playground.

V.4.1 Individual analysis frame

The d_{symb} playground enables users to select a single time series and focus on its exploration. A visualization, shown in Figure V.11(B), allows users to explore the raw multivariate time series and its corresponding symbolic sequence represented as a colorbar. Therefore, users can interpret the multivariate segmentation from d_{symb}, which is based on changes in the mean, and investigate how it deals with the potential non-stationarity of the input time series. It also allows one to understand what a symbol represents with regard to real-world events: each symbol can be interpreted as an action with a semantic meaning. For the plot of the raw multivariate time series, by default, the number of displayed dimensions on the same plot is capped at 20 for conciseness purposes. The user can investigate each group of 20 dimensions separately (while the displayed symbolic sequence is the one corresponding to all dimensions together). The user can also choose to visualize all dimensions at once.

V.4.2 Data set analysis frame

Instead of focusing on a single time series, the data set analysis frame explores the whole multivariate time series data set at once. With a quick glance, the colorbars provide a compact representation of a data set of multivariate time series, as displayed in Figure V.11(A). Each row corresponds to the symbolic representation of each time series of the data set. In a colorbar, black vertical lines illustrate change-points. Therefore, users can observe the different regimes that occur in the time series. The

colorbars can be represented in two different ways: (i) the true lengths of the time series; or (ii) the normalized lengths. In the latter, all colorbars are stretched to have the same length. Scrolling down, more visualizations are available to help users understand the meaning of the symbols: (i) the histogram of the symbols, (ii) the distribution of the lengths for each symbol, (iii) the time stamps where each symbol occurs, and (iv) two figures illustrating the similarities between each individual symbol. Finally, the users can also visualize the pairwise distance matrix between the obtained symbolic sequences. Note that the users can modify the number of symbols at any time and, thanks to the fast computation of d_{symb}, all the visualizations described above are updated in real-time.

V.4.3 Benchmark frame

The benchmark frame compares the d_{symb} distance measure to 9 existing distance measures on time series. We apply our benchmark to the real-world JIGSAWS data set [Gao+14] with the goal of identifying surgeons' gestures based on kinematic time series, as done in Section V.3.1. All results are precomputed (in order to save the users some computing time). In this data set, we consider two surgical gestures: *Knot Tying* (39 multivariate time series) and *Needle Passing* (40 multivariate time series). As shown in Figure V.11(C), we display the pairwise distance matrix for the chosen distance measure, as well as the clustering accuracy and the execution time (in seconds) for all distance measures in the benchmark.

V.5 Conclusion

We have introduced d_{symb}, a novel distance measure on multivariate and non-stationary signals, that uses symbolization as an intermediate step. Our method uses change-point detection to segment signals, K-means clustering to create symbolic representations, and the general edit distance with custom costs to compare them. The resulting algorithm is fast and produces interpretable symbols. We have applied d_{symb} to several physiological data sets. d_{symb} has achieved reasonable clustering performance while being several orders of magnitude faster than classical methods (such as DTW). In addition, the symbolic sequences allow users to understand, with a glimpse, the dynamic of the multiple signals at hand. Finally, we have implemented an online tool, called the d_{symb} playground, that allows users to upload their data set and apply d_{symb} on it.

Chapter VI

Conclusion and perspectives

In this thesis, we have proposed two novel symbolic representations and distance measures for time series: ASTRIDE for a data set of univariate time series (Chapter IV), and d_{symb} for a data set of multivariate time series (Chapter V). These distance measures transform time series into symbolic sequences which are then compared using a modified version of the Levenshtein distance. We have also conducted two surveys: one on the symbolic representations for time series (Chapter II), and one on the distance measures for time series, strings, and symbolic sequences (Chapter III). We have shown that, compared to the literature, our proposed distance can deal with physiological signals that are multivariate and non-stationary, thanks to an adaptive segmentation algorithm. The resulting symbolic sequences are interpretable, as each symbol represents a salient event such as walking or turning around in the context of gait data. ASTRIDE and d_{symb} are shown to be fast to compute. The d_{symb} playground, a web-based tool, allows a user to upload its data set of multivariate time series and gain insights into it.

Now, let us look into the perspectives of this thesis. First of all, the proposed symbolization methods could be applied to more tasks. (i) ASTRIDE or d_{symb} could be employed as an intermediate step in classifiers. They could be used in the shapelet category such as in SAX-SEQL [NGl17] and Mr-SEQL [Ngu+19], or in the dictionary category with SAX-VSM [SM13], BOSS [Sch15], TDE [Mid+20], and PETSC [FCG22]. These methods currently use SAX or SFA as the symbolization step, and involve overlapping sliding windows that increase the time and space complexities. These classifiers are described in a recent review [MSB23]. Thanks to the adaptive segmentation of ASTRIDE and d_{symb}, these sliding windows could be used more efficiently. (ii) The ASTRIDE and d_{symb} symbolic sequences could also be analyzed by methods developed in the bioinformatics community, for example in pattern discovery or anomaly detection. Indeed, research in bioinformatics revolves around the study of sequences of characters. (iii) Moreover, the obtained symbolic sequences could be modeled by Markov chains where each symbol would be a state. Examining the probability of transition between each symbol could provide meaningful information to a medical practitioner for example.

d_{symb} could be extended to adapt to physiological signals with (even) more challenging structures. (i) They could be able to take into account the multi-resolution aspect. Let us take the example of the human locomotion studied in Chapter V. It is interesting to detect each global regime (few segments are allowed): standing still, walking, and turning, and also to detect each local regime (many segments are al-

lowed): first footstep of walking, second footstep, etc. A multi-resolution approach would allow for a more comprehensive study on the phenomenon. (ii) Furthermore, d_{symb} could tackle the correlation between each dimension of physiological signals differently. With physiological signals, the dimension can be of a few hundreds, thus using a dimension selection algorithm could be beneficial [TO22; DNI23; RB23]. The segmentation and/or clustering steps could be extended to sparse features.

Moreover, our distance measure on symbolic sequences, based on the general Levenshtein distance, could be investigated further. We have empirically showed that our distance measure is much faster than DTW, but due to the strong links between edit distances and DTW, it would be interesting to study the theoretical properties of our method and understand better to what extent it is an approximation of the DTW. Moreover, some other theoretical aspects could be studied, such as the possibility to have a lower-bound.

Finally, the multimodal aspect of some physiological signals could be addressed. Indeed, in some protocols developed by the Centre Borelli, several physiological functions are analyzed simultaneously (such as brain activity, respiration, movements, cardiac activity...). Constructing a symbolization for this complex multimodal data would constitute a nice challenge and extension of this thesis work. If some dimensions do not have the same sampling frequency, an adaptive segmentation on each modality could be applied, which would result in several symbolic sequences that could then constitute meta-symbols. Moreover, if the dimensions represent signals with a different physical nature, then applying change-point detection on the mean for all dimensions might not be sufficient. Indeed, some variables might need to detect changes in frequency, while others might need to detect changes in the mean or in the slope: some novel cost functions could therefore be introduced, possibly with some supervised techniques.

Bibliography

[AC01] John Aach and George M. Church. "Aligning gene expression time series with time warping algorithms ". In: *Bioinformatics* 17.6 (2001), pp. 495–508. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/17.6.495. eprint: https://academic.oup.com/bioinformatics/article-pdf/17/6/495/48837158/bioinformatics_17_6_495.pdf. URL: https://doi.org/10.1093/bioinformatics/17.6.495.

[AML19] Amaia Abanda, Usue Mori, and José Antonio Lozano. "A review on distance based time series classification". In: *Data Min Knowl Disc* 33 (2019), pp. 378–412.

[Aga+21] Surabhi Agarwal, Trang Thu Nguyen, Thach Le Nguyen, and Georgiana Ifrim. "Ranking by Aggregating Referees: Evaluating the Informativeness of Explanation Methods for Time Series Classification". In: *Advanced Analytics and Learning on Temporal Data*. Cham: Springer International Publishing, 2021, pp. 3–20. ISBN: 978-3-030-91445-5. DOI: 10.1007/978-3-030-91445-5_1.

[ASY15] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering – A decade review". In: *Information Systems* 53 (2015), pp. 16–38. ISSN: 0306-4379. DOI: https://doi.org/10.1016/j.is.2015.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0306437915000733.

[AFS93] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. "Efficient similarity search in sequence databases". In: *Foundations of Data Organization and Algorithms*. Springer Berlin Heidelberg, 1993, pp. 69–84. ISBN: 978-3-540-48047-1.

[Agr+95] Rakesh Agrawal, Giuseppe Psaila, Edward L. Wimmers, and Mohamed Zaït. "Querying Shapes of Histories". In: *Proceedings of the 21th International Conference on Very Large Data Bases*. VLDB '95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, 502–514. ISBN: 1558603794.

[AABH14] Almahdi Alshareef, Azuraliza Abu Bakar, and Abdul Hamdan. "A Harmony Search Algorithm with Multi-pitch Adjustment Rate for Symbolic Time Series Data Representation". In: *International Journal of Modern Education and Computer Science* 6 (June 2014), pp. 58–70. DOI: 10.5815/ijmecs.2014.06.08.

[AVC20] Manuel Anacleto, Susana Vinga, and Alexandra M. Carvalho. "MSAX: Multivariate Symbolic Aggregate Approximation for Time Series Classification". In: *Computational Intelligence Methods for Bioinformatics and Biostatistics*. Cham: Springer International Publishing, 2020, pp. 90–97. ISBN: 978-3-030-63061-4.

[AJB97] Henrik André-Jönsson and Dushan Z. Badal. "Using signature files for querying time-series data". In: *Principles of Data Mining and Knowledge Discovery*. Springer Berlin Heidelberg, 1997, pp. 211–220. ISBN: 978-3-540-69236-2.

[AG87] A. Apostolico and C. Guerra. "The longest common subsequence problem revisited". In: *Algorithmica* 2 (1987), 315–336. DOI: https://doi.org/10.1007/BF01840365.

[AHWM19] Oluseun Omotola Aremu, David Hyland-Wood, and Peter Ross McAree. "A Relative Entropy Weibull-SAX framework for health indices construction and health stage division in degradation modeling of multivariate time series asset data". In: *Advanced Engineering Informatics* 40 (2019), pp. 121–134. ISSN: 1474-0346. DOI: https://doi.org/10.1016/j.aei.2019.03.003. URL: https://www.sciencedirect.com/science/article/pii/S1474034618305603.

[Bag+17] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Min Knowl Disc* 31.3 (2017), pp. 606–660. DOI: https://doi.org/10.1007/s10618-016-0483-9.

[Bai+13] Xue Bai, Yun Xiong, Yangyong Zhu, and Hengshu Zhu. "Time Series Representation: A Random Shifting Perspective". In: *Web-Age Information Management*. Springer Berlin Heidelberg, 2013, pp. 37–50. ISBN: 978-3-642-38562-9. DOI: 10.1007/978-3-642-38562-9_4.

[BB21] Francisco J. Baldán and José M. Benítez. "Multivariate times series classification through an interpretable representation". In: *Information Sciences* 569 (2021), pp. 596–614. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.05.024. URL: https://www.sciencedirect.com/science/article/pii/S0020025521004825.

[BP02] Christoph Bandt and Bernd Pompe. "Permutation Entropy: A Natural Complexity Measure for Time Series". In: *Phys. Rev. Lett.* 88 (17 2002), p. 174102. DOI: 10.1103/PhysRevLett.88.174102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.174102.

[BY08] Depei Bao and Zehong Yang. "Intelligent stock trading system by turning point confirming and probabilistic reasoning". In: *Expert Systems with Applications* 34.1 (2008), pp. 620–627. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2006.09.043. URL: https://www.sciencedirect.com/science/article/pii/S0957417406003125.

[Bar+10] Cécile Barat, Christophe Ducottet, Élisa Fromont, Anne-Claire Legrand, and Marc Sebban. "Weighted Symbols-Based Edit Distance for String-Structured Image Classification". In: *ECML/PKDD*. 2010.

[Bar07] Gregory V. Bard. “Spelling-Error Tolerant, Order-Independent Pass-Phrases via the Damerau-Levenshtein String-Edit Distance Metric”. In: *Proceedings of the Fifth Australasian Symposium on ACSW Frontiers - Volume 68*. ACSW ’07. Ballarat, Australia: Australian Computer Society, Inc., 2007, 117–124. ISBN: 192068285X.

[Bar+18] Ioannis Bargiotas, Julien Audiffren, Nicolas Vayatis, Pierre-Paul Vidal, Stephane Buffat, Alain P. Yelnik, and Damien Ricard. “On the importance of local dynamics in statokinesigram: A multivariate approach for postural control evaluation in elderly”. In: *PLOS ONE* 13.2 (Feb. 2018), pp. 1–15. DOI: 10.1371/journal.pone.0192868. URL: https://doi.org/10.1371/journal.pone.0192868.

[BABO12] Peiman Mamani Barnaghi, Azuraliza Abu Bakar, and Zulaiha Ali Othman. “Enhanced symbolic aggregate approximation method for financial time series data representation”. In: *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)*. 2012, pp. 790–795. URL: https://ieeexplore.ieee.org/document/6528740.

[BM+16] R. Barrois-Müller, T. Gregory, L. Oudre, T. Moreau, C. Truong, A. Aram Pulini, A. Vienne, C. Labourdette, N. Vayatis, S. Buffat, A. Yelnik, C. de Waele, S. Laporte, P.-P. Vidal, and D. Ricard. “An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis”. In: *PLoS One* 11.10 (2016), e0164975.

[BM+17] R. Barrois-Müller, D. Ricard, L. Oudre, L. Tlili, C. Provost, A. Vienne, P.-P. Vidal, S. Buffat, and A. Yelnik. “Observational study of 180° turning strategies using inertial measurement units and fall risk in poststroke hemiparetic patients”. In: *Frontiers in Neurology* 8 (2017).

[Bat+14] Gustavo E. A. P. A. Batista, Eamonn J. Keogh, Oben Moses Tataw, and Vinícius M. A. de Souza. “CID: an efficient complexity-invariant distance for time series”. In: *Data Min Knowl Disc* 28 (2014), pp. 634–669. DOI: 10.1007/s10618-013-0312-3.

[BBC18] Seif-Eddine Benkabou, Khalid Benabdeslem, and Bruno Canitia. “Unsupervised outlier detection for time series by entropy and dynamic time warping”. In: *Knowledge and Information Systems* 54 (2018), 463–486. DOI: 10.1007/s10115-017-1067-8.

[BR02] Sèverine Bérard and Éric Rivals. “Comparison of Minisatellites”. In: *Proceedings of the Sixth Annual International Conference on Computational Biology*. RECOMB ’02. Washington, DC, USA: Association for Computing Machinery, 2002, 67–76. ISBN: 1581134983. DOI: 10.1145/565196.565205. URL: https://doi.org/10.1145/565196.565205.

[Ber06] P. Berkhin. “A Survey of Clustering Data Mining Techniques”. In: *Grouping Multidimensional Data: Recent Advances in Clustering*. Ed. by Jacob Kogan, Charles Nicholas, and Marc Teboulle. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 25–71. ISBN: 978-3-540-28349-2. DOI: 10.1007/3-540-28349-8_2. URL: https://doi.org/10.1007/3-540-28349-8_2.

[BC94] Donald J. Berndt and James Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series". In: *KDD Workshop*. 1994.

[BR14] Vineetha Bettaiah and Heggere S. Ranganath. "An Analysis of Time Series Representation Methods: Data Mining Applications Perspective". In: *Proceedings of the 2014 ACM Southeast Regional Conference*. ACM SE '14. Kennesaw, Georgia: Association for Computing Machinery, 2014. ISBN: 9781450329231. DOI: 10.1145/2638404.2638475. URL: https://doi.org/10.1145/2638404.2638475.

[BG+21] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. "A Review on Outlier/Anomaly Detection in Time Series Data". In: *ACM Comput. Surv.* 54.3 (2021). ISSN: 0360-0300. DOI: 10.1145/3444690. URL: https://doi.org/10.1145/3444690.

[BMV21] Mathieu Blondel, Arthur Mensch, and Jean-Philippe Vert. "Differentiable Divergences Between Time Series". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, 2021, pp. 3853–3861. URL: https://proceedings.mlr.press/v130/blondel21a.html.

[Boi+22] Alexandre Bois, Brian Tervil, Albane Moreau, Aliénor Vienne-Jumeau, Damien Ricard, and Laurent Oudre. "A topological data analysis-based method for gait signals with an application to the study of multiple sclerosis". In: *PLOS ONE* 17.5 (May 2022), pp. 1–23. DOI: 10.1371/journal.pone.0268475. URL: https://doi.org/10.1371/journal.pone.0268475.

[BBG13] A. Bondu, M. Boullé, and B. Grossin. "SAXO: An optimized data-driven symbolic representation of time series". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 2013, pp. 1–9. DOI: 10.1109/IJCNN.2013.6706816.

[BBC16] Alexis Bondu, Marc Boullé, and Antoine Cornuéjols. "Symbolic Representation of Time Series: A Hierarchical Coclustering Formalization". In: Springer International Publishing, 2016, pp. 3–16. DOI: 10.1007/978-3-319-44412-3_1.

[Bou06] Marc Boullé. "MODL: A Bayes optimal discretization method for continuous attributes". In: *Mach Learn* 65 (2006), 131–165. DOI: 10.1007/s10994-006-8364-x.

[BTT21a] Konstantinos Bountrogiannis, George Tzagkarakis, and Panagiotis Tsakalides. "Anomaly Detection for Symbolic Time Series Representations of Reduced Dimensionality". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 2398–2402. DOI: 10.23919/Eusipco47968.2020.9287474.

[BTT21b] Konstantinos Bountrogiannis, George Tzagkarakis, and Panagiotis Tsakalides. "Data-driven Kernel-based Probabilistic SAX for Time Series Dimensionality Reduction". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 2343–2347. DOI: 10.23919/Eusipco47968.2020.9287311.

[BK15] Matthew Butler and Dimitar Kazakov. "SAX Discretization Does Not Guarantee Equiprobable Symbols". In: *IEEE Transactions on Knowledge and Data Engineering* 27.4 (2015), pp. 1162–1166. DOI: 10.1109/TKDE.2014.2382882.

[Cai+98] E.G. Caiani, A. Porta, G. Baselli, M. Turiel, S. Muzzupappa, F. Pieruzzi, C. Crema, A. Malliani, and S. Cerutti. "Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume". In: *Computers in Cardiology 1998. Vol. 25 (Cat. No.98CH36292)*. 1998, pp. 73–76. DOI: 10.1109/CIC.1998.731723.

[Cam+10] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn Keogh. "iSAX 2.0: Indexing and Mining One Billion Time Series". In: *2010 IEEE International Conference on Data Mining*. 2010, pp. 58–67. DOI: 10.1109/ICDM.2010.124.

[Cam+14] Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, and Eamonn Keogh. "Beyond one billion time series: indexing and mining very large time series collections with iSAX2+". In: *Knowl Inf Syst* 39 (2014), pp. 123–151. DOI: 10.1007/s10115-012-0606-6.

[Can+12] K. Selçuk Candan, Rosaria Rossini, Xiaolan Wang, and Maria Luisa Sapino. "SDTW: Computing DTW Distances Using Locally Relevant Constraints Based on Salient Feature Alignments". In: *Proc. VLDB Endow.* 5.11 (2012), 1519–1530. ISSN: 2150-8097. DOI: 10.14778/2350229.2350266. URL: https://doi.org/10.14778/2350229.2350266.

[Cas+12] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. "Similarity measures and dimensionality reduction techniques for time series data mining". In: *Advances in data mining knowledge discovery and applications'(InTech, Rijeka, Croatia, 2012,* (2012), pp. 71–96.

[CA15] Nuno C. Castro and Paulo J. Azevedo. "Automatically Estimating ISAX Parameters". In: *Intell. Data Anal.* 19.3 (2015), 581–595. ISSN: 1088-467X. DOI: 10.3233/IDA-150733. URL: https://doi.org/10.3233/IDA-150733.

[Cha+02] Kaushik Chakrabarti, Eamonn Keogh, Sharad Mehrotra, and Michael Pazzani. "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases". In: *ACM Trans. Database Syst.* 27.2 (2002), 188–228. ISSN: 0362-5915. DOI: 10.1145/568518.568520. URL: https://doi.org/10.1145/568518.568520.

[CF99] Kin-Pong Chan and Ada Wai-Chee Fu. "Efficient time series matching by wavelets". In: *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*. 1999, pp. 126–133. DOI: 10.1109/ICDE.1999.754915.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Comput. Surv.* 41.3 (2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: https://doi.org/10.1145/1541880.1541882.

[Ch01] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Marroquín. "Searching in Metric Spaces". In: *ACM Comput. Surv.* 33.3 (2001), 273–321. ISSN: 0360-0300. DOI: 10.1145/502807.502808. URL: https://doi.org/10.1145/502807.502808.

[Che+20] Haiyan Chen, Jinghan Du, Weining Zhang, and Bohan Li. "An iterative end point fitting based trend segmentation representation of time series and its distance measure". In: *Multimed Tools Appl* 79 (2020), 13481–13499. DOI: 10.1007/s11042-019-08440-0.

[CN04] Lei Chen and Raymond Ng. "On the Marriage of Lp-Norms and Edit Distance". In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*. VLDB '04. Toronto, Canada: VLDB Endowment, 2004, 792–803. ISBN: 0120884690.

[COO05] Lei Chen, M. Tamer Özsu, and Vincent Oria. "Robust and Fast Similarity Search for Moving Object Trajectories". In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. SIGMOD '05. Baltimore, Maryland: Association for Computing Machinery, 2005, 491–502. ISBN: 1595930604. DOI: 10.1145/1066157.1066213. URL: https://doi.org/10.1145/1066157.1066213.

[CG23] Xinye Chen and Stefan Güttel. "An Efficient Aggregation Method for the Symbolic Representation of Temporal Data". In: *ACM Trans. Knowl. Discov. Data* 17.1 (2023). ISSN: 1556-4681. DOI: 10.1145/3532622. URL: https://doi.org/10.1145/3532622.

[Chu+01] Fu Lai Korris Chung, Tak-Chung Fu, Wing Pong Robert Luk, and Vincent To Yee Ng. "Flexible time series pattern matching based on perceptually important points". In: *International Joint Conference on Artificial Intelligence Workshop on Learning from Temporal and Spatial Data*. 2001.

[Com+24a] Sylvain W. Combettes, Paul Boniol, Antoine Mazarguil, Danping Wang, Diego Vaquero-Ramos, Marion Chauveau, Laurent Oudre, Nicolas Vayatis, Pierre-Paul Vidal, Alexandra Roren, and Marie-Martine Lefèvre-Colau. "Arm-CODA: A Data Set of Upper-limb Human Movement During Routine Examination". In: *Image Processing On Line* 14 (2024). https://doi.org/10.5201/ipol.2024.494, pp. 1–13.

[Com+24b] Sylvain W. Combettes, Paul Boniol, Charles Truong, and Laurent Oudre. "d_{symb} playground: an interactive tool to explore large multivariate time series datasets". In: *Proceedings of the International Conference on Data Engineering (ICDE)*. Utrecht, Netherlands, 2024.

[CTO23a] Sylvain W. Combettes, Charles Truong, and Laurent Oudre. "An Interpretable Distance Measure for Multivariate Non-Stationary Physiological Signals". In: *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. Shanghai, China, 2023, pp. 533–539. DOI: 10.1109/ICDMW60847.2023.00076.

[CTO23b] Sylvain W. Combettes, Charles Truong, and Laurent Oudre. *ASTRIDE: Adaptive Symbolization for Time Series Databases*. 2023. DOI: 10.48550/ARXIV.2302.04097. URL: https://arxiv.org/abs/2302.04097.

[CB17] Marco Cuturi and Mathieu Blondel. "Soft-DTW: A Differentiable Loss Function for Time-Series". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, 894–903.

[Dam64] Fred J. Damerau. "A Technique for Computer Detection and Correction of Spelling Errors". In: *Commun. ACM* 7.3 (1964), 171–176. ISSN: 0001-0782. DOI: 10.1145/363958.363994. URL: https://doi.org/10.1145/363958.363994.

[Das+97] Gautam Das, Rudolf Fleischer, Leszek Gasieniec, Dimitris Gunopulos, and Juha Kärkkäinen. "Episode matching". In: *Combinatorial Pattern Matching*. Ed. by Alberto Apostolico and Jotun Hein. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 12–27. ISBN: 978-3-540-69214-0.

[Dau+19] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. "The UCR time series archive". In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (2019), pp. 1293–1305.

[DFT03] C. S. Daw, C. E. A. Finney, and E. R. Tracy. "A review of symbolic analysis of experimental data". In: *Review of Scientific Instruments* 74.2 (Jan. 2003), pp. 915–930. ISSN: 0034-6748. DOI: 10.1063/1.1531823. URL: https://doi.org/10.1063/1.1531823.

[DNI23] Bhaskar Dhariyal, Thach Le Nguyen, and Georgiana Ifrim. "Scalable classifier-agnostic channel selection for multivariate time series classification". In: *Data Min Knowl Disc* 37 (2023), 1010–1054. DOI: 10.1007/s10618-022-00909-1.

[Din+08] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures". In: *Proc. VLDB Endow.* 1.2 (2008), 1542–1552. ISSN: 2150-8097. DOI: 10.14778/1454159.1454226. URL: https://doi.org/10.14778/1454159.1454226.

[DAM23] Lamia Djebour, Reza Akbarinia, and Florent Masseglia. "Variable-Size Segmentation for Time Series Representation". In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems LIII*. Springer Berlin Heidelberg, 2023, pp. 34–65. ISBN: 978-3-662-66863-4. DOI: 10.1007/978-3-662-66863-4_2. URL: https://doi.org/10.1007/978-3-662-66863-4_2.

[EG20a] Steve Elsworth and Stefan Güttel. "ABBA: adaptive Brownian bridge-based symbolic aggregation of time series". In: *Data Min Knowl Disc* 34 (2020), pp. 1175–1200. DOI: 10.1007/s10618-020-00689-6.

[EG20b] Steven Elsworth and Stefan Güttel. *Time series forecasting using LSTM networks: A symbolic approach*. arXiv EPrint arXiv:2003.05672v1. arXiv, 2020, p. 12. URL: https://arxiv.org/abs/2003.05672.

[EA12] Philippe Esling and Carlos Agon. "Time-Series Data Mining". In: *ACM Comput. Surv.* 45.1 (2012). ISSN: 0360-0300. DOI: 10.1145/2379776.2379788. URL: https://doi.org/10.1145/2379776.2379788.

[Esm+12] Bilal Esmael, Arghad Arnaout, Rudolf K. Fruhwirth, and Gerhard Thonhauser. "Multivariate Time Series Classification by Combining Trend-Based and Value-Based Approximations". In: *Computational Science and Its Applications – ICCSA 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 392–403. ISBN: 978-3-642-31128-4. DOI: 10.1007/978-3-642-31128-4_29.

[FRM94] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. "Fast Subsequence Matching in Time-Series Databases". In: *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*. SIGMOD '94. Minneapolis, Minnesota, USA: Association for Computing Machinery, 1994, 419–429. ISBN: 0897916395. DOI: 10.1145/191839.191925. URL: https://doi.org/10.1145/191839.191925.

[FJ20] Johann Faouzi and Hicham Janati. "pyts: A Python Package for Time Series Classification". In: *Journal of Machine Learning Research* 21.46 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/19-763.html.

[FCG22] Len Feremans, Boris Cule, and Bart Goethals. "PETSC: pattern-based embedding for time series classification". In: *Data Min Knowl Disc* 36 (2022), 1015–1061. DOI: 10.1007/s10618-022-00822-7.

[Fu11] Tak chung Fu. "A review on time series data mining". In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2010.09.007. URL: https://www.sciencedirect.com/science/article/pii/S0952197610001727.

[Fua12] Muhammad Marwan Muhammad Fuad. "Differential evolution versus genetic algorithms: towards symbolic aggregate approximation of non-normalized time series". In: *IDEAS '12*. 2012.

[Gao+14] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. "The JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling". In: *Modeling and Monitoring of Computer Assisted Interventions (M2CAI) – MICCAI Workshop*. 2014.

[GKL01] Zong Woo Geem, Joong Hoon Kim, and G.V. Loganathan. "A New Heuristic Optimization Algorithm: Harmony Search". In: *SIMULATION* 76.2 (2001), pp. 60–68. DOI: 10.1177/003754970107600201.

[Gel+19] Zoltan Geler, Vladimir Kurbalija, Mirjana Ivanović, Miloš Radovanović, and Weihui Dai. "Dynamic Time Warping: Itakura vs Sakoe-Chiba". In: *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*. 2019, pp. 1–6. DOI: 10.1109/INISTA.2019.8778300.

[Ger+22] Thibaut Germain, Charles Truong, Laurent Oudre, and Eric Krejci. "Unsupervised study of plethysmography signals through DTW clustering". In: *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 2022, pp. 3396–3400. DOI: 10.1109/EMBC48229.2022.9870907.

[GG92] Allen Gersho and Robert M. Gray. *Vector Quantization and Signal Compression*. Springer New York, NY, 1992. DOI: 10.1007/978-1-4615-3626-0.

[GD01] Dimitrios Gunopulos and Gautam Das. "Time Series Similarity Measures and Time Series Indexing (Abstract Only)". In: *SIGMOD Rec.* 30.2 (2001), p. 624. ISSN: 0163-5808. DOI: 10.1145/376284.375808. URL: https://doi.org/10.1145/376284.375808.

[HTW23] Matthieu Herrmann, Chang Wei Tan, and Geoffrey I. Webb. "Parameterizing the cost function of dynamic time warping with application to time series classification". In: *Data Min Knowl Disc* (2023). DOI: 10.1007/s10618-023-00926-8.

[HW21] Matthieu Herrmann and Geoffrey I. Webb. "Early abandoning and pruning for elastic distances including dynamic time warping". In: *Data Min Knowl Disc* 35 (2021), pp. 2577–2601. DOI: https://doi.org/10.1007/s10618-021-00782-4.

[Hir77] Daniel S. Hirschberg. "Algorithms for the Longest Common Subsequence Problem". In: *J. ACM* 24.4 (1977), 664–675. ISSN: 0004-5411. DOI: 10.1145/322033.322044. URL: https://doi.org/10.1145/322033.322044.

[HMB23] Christopher Holder, Matthew Middlehurst, and Anthony Bagnall. "A review and evaluation of elastic distance functions for time series clustering". In: *Knowl Inf Syst* (2023). DOI: 10.1007/s10115-023-01952-0.

[HY99] Yun-Wu Huang and Philip S. Yu. "Adaptive Query Processing for Time-Series Data". In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '99. San Diego, California, USA: Association for Computing Machinery, 1999, 282–286. ISBN: 1581131437. DOI: 10.1145/312129.318357. URL: https://doi.org/10.1145/312129.318357.

[Hug06] Bernard Hugueney. "Adaptive Segmentation-Based Symbolic Representations of Time Series for Better Modeling and Lower Bounding Distance Measures". In: *Knowledge Discovery in Databases: PKDD 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 545–552. ISBN: 978-3-540-46048-0. DOI: 10.1007/11871637_54.

[HA07] Nguyen Quoc Viet Hung and Duong Tuan Anh. "Combining SAX and Piecewise Linear Approximation to Improve Similarity Search on Financial Time Series". In: *2007 International Symposium on Information Technology Convergence (ISITC 2007)*. 2007, pp. 58–62. DOI: 10.1109/ISITC.2007.24.

[Inf18] InfoScout. *weighted-levenshtein*. 2018. URL: https://github.com/infoscout/weighted-levenshtein.

[Ita75] F. Itakura. "Minimum prediction residual principle applied to speech recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 23.1 (1975), pp. 67–72. DOI: 10.1109/TASSP.1975.1162641.

[IP14] Hesam Izakian and Witold Pedrycz. "Anomaly Detection and Characterization in Spatial Time Series Data: A Cluster-Centric Approach". In: *IEEE Transactions on Fuzzy Systems* 22.6 (2014), pp. 1612–1624. DOI: 10.1109/TFUZZ.2014.2302456.

[JCG20] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. "Spatio-temporal alignments: Optimal transport through space and time". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1695–1704. URL: https://proceedings.mlr.press/v108/janati20a.html.

[JJO11] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. "Weighted dynamic time warping for time series classification". In: *Pattern Recognition* 44.9 (2011). Computer Analysis of Images and Patterns, pp. 2231–2240. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2010.09.022. URL: https://www.sciencedirect.com/science/article/pii/S003132031000484X.

[Jun+21] Sylvain Jung, Laurent Oudre, Charles Truong, Eric Dorveaux, Louis Gorintin, Nicolas Vayatis, and Damien Ricard. "Adaptive Change-Point Detection for Studying Human Locomotion". In: *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 2021, pp. 2020–2024. DOI: 10.1109/EMBC46164.2021.9629775.

[KLF05] E. Keogh, J. Lin, and A. Fu. "HOT SAX: efficiently finding the most unusual time series subsequence". In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*. 2005, 8 pp.–. DOI: 10.1109/ICDM.2005.79.

[Keo+01] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases". In: *Knowl Inf Syst* 3 (2001), 263–286. DOI: 10.1007/PL00011669.

[Keo+04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. "Segmenting time series: A survey and novel approach". In: *Data Mining in Time Series Databases*. 2004, pp. 1–21. DOI: 10.1142/9789812565402_0001. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812565402_0001. URL: https://www.worldscientific.com/doi/abs/10.1142/9789812565402_0001.

[KK03] Eamonn Keogh and Shruti Kasetty. "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration". In: *Data Min Knowl Disc* 7 (2003), pp. 349–371. DOI: 10.1023/A:1024988512476.

[KR05] Eamonn Keogh and Chotirat Ann Ratanamahatana. “Exact indexing of dynamic time warping”. In: *Knowledge and Information Systems* 7 (2005), 358–386.

[Keo+09] Eamonn Keogh, Li Wei, Xiaopeng Xi, Michail Vlachos, Sang-Hee Lee, and Pavlos Protopapas. “Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures”. In: *The VLDB Journal* 18 (2009), pages611–630. DOI: https://doi.org/10.1007/s00778-008-0111-4.

[KP01] Eamonn J. Keogh and Michael J. Pazzani. “Derivative Dynamic Time Warping”. In: *Proceedings of the 2001 SIAM International Conference on Data Mining (SDM)*. SIAM, 2001, pp. 1–11. DOI: 10.1137/1.9781611972719.1. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972719.1. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972719.1.

[KFE12] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. “Optimal detection of changepoints with a linear computational cost”. In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1590–1598.

[KPC01] Sang-Wook Kim, Sanghyun Park, and W.W. Chu. “An index-based approach for similarity search supporting time warping in large sequence databases”. In: *Proceedings 17th International Conference on Data Engineering*. 2001, pp. 607–614. DOI: 10.1109/ICDE.2001.914875.

[KR20] Matej Kloska and Viera Rozinajova. “Distribution-Wise Symbolic Aggregate ApproXimation (dwSAX)”. In: *Intelligent Data Engineering and Automated Learning – IDEAL 2020*. Cham: Springer International Publishing, 2020, pp. 304–315. ISBN: 978-3-030-62362-3. DOI: 10.1007/978-3-030-62362-3_27.

[KR21] Matej Kloska and Viera Rozinajova. “Towards Symbolic Time Series Representation Improved by Kernel Density Estimators”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems L*. Springer Berlin Heidelberg, 2021, pp. 25–45. ISBN: 978-3-662-64553-6. DOI: 10.1007/978-3-662-64553-6_2. URL: https://doi.org/10.1007/978-3-662-64553-6_2.

[Kru83] Joseph B. Kruskal. “An Overview of Sequence Comparison: Time Warps, String Edits, and Macromolecules”. In: *SIAM Review* 25.2 (1983), pp. 201–237. DOI: 10.1137/1025045. URL: https://doi.org/10.1137/1025045.

[Kuk92] Karen Kukich. “Techniques for Automatically Correcting Words in Text”. In: *ACM Comput. Surv.* 24.4 (1992), 377–439. ISSN: 0360-0300. DOI: 10.1145/146370.146380. URL: https://doi.org/10.1145/146370.146380.

[Lar+19] James Large, Anthony Bagnall, Simon Malinowski, and Romain Tavenard. “On time series classification with dictionary-based classifiers”. In: *Intelligent Data Analysis* 23.5 (2019), pp. 1073–1089. ISSN: 1088-467X. DOI: 10.3233/ida-184333.

[LTN20] Xuan-May Thi Le, Tuan Minh Tran, and Hien T. Nguyen. "An improvement of SAX representation for time series by using complexity invariance". In: *Intell. Data Anal.* 24 (2020), pp. 625–641.

[LLP23] Zed Lee, Tony Lindgren, and Panagiotis Papapetrou. "Z-Time: efficient and effective interpretable multivariate time series classification". In: *Data Min Knowl Disc* (2023). DOI: https://doi.org/10.1007/s10618-023-00969-x.

[Lem09] Daniel Lemire. "Faster retrieval with a two-pass dynamic-time-warping lower bound". In: *Pattern Recognition* 42.9 (2009), pp. 2169–2180. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2008.11.030. URL: https://www.sciencedirect.com/science/article/pii/S0031320308004925.

[Lev+66] Vladimir I Levenshtein et al. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet Physics Doklady*. Vol. 10. 1966, pp. 707–710.

[LZY12] Guiling Li, Liping Zhang, and Linquan Yang. "TSX: A Novel Symbolic Representation for Financial Time Series". In: *PRICAI 2012: Trends in Artificial Intelligence*. Springer Berlin Heidelberg, 2012, pp. 262–273. ISBN: 978-3-642-32695-0. DOI: 10.1007/978-3-642-32695-0_25.

[LDH13] Tianyu Li, Fang-Yan Dong, and Kaoru Hirota. "Distance Measure for Symbolic Approximation Representation with Subsequence Direction for Time Series Data Mining". In: *Journal of Advanced Computational Intelligence and Intelligent Informatics* 17.2 (2013), pp. 263–271. DOI: 10.20965/jaciii.2013.p0263.

[LS22] Yucheng Li and Derong Shen. "A new symbolic representation method for time series". In: *Information Sciences* 609 (2022), pp. 276–303. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2022.07.047. URL: https://www.sciencedirect.com/science/article/pii/S0020025522007368.

[LKL05] Jessica Lin, Eamonn Keogh, and Stefano Lonardi. "Visualizing and Discovering Non-Trivial Patterns in Large Time Series Databases". In: *Information Visualization* 4.2 (2005), pp. 61–82. DOI: 10.1057/palgrave.ivs.9500089. eprint: https://doi.org/10.1057/palgrave.ivs.9500089. URL: https://doi.org/10.1057/palgrave.ivs.9500089.

[Lin+03] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms". In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. DMKD '03. Association for Computing Machinery, 2003, 2–11. ISBN: 9781450374224. DOI: 10.1145/882082.882086.

[Lin+07] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. "Experiencing SAX: a novel symbolic representation of time series". In: *Data Min Knowl Disc* 15 (2007), pp. 107–144. DOI: https://doi.org/10.1007/s10618-007-0064-z.

[LKL12] Jessica Lin, Rohan Khade, and Yuan Li. "Rotation-invariant similarity in time series using bag-of-patterns representation". In: *Journal of Intelligent Information Systems* 39 (2012), pp. 287–315.

[LB15] Jason Lines and Anthony Bagnall. "Time series classification with ensembles of elastic distance measures". In: *Data Min. Knowl. Discov.* 29 (2015), pp. 565–592.

[Liu+02] Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. "Discretization: An Enabling Technique". In: *Data Min. Knowl. Discov.* 6 (2002), 393–423. DOI: 10.1023/A:1016304305535.

[LSK06] B. Lkhagva, Yu Suzuki, and K. Kawagoe. "New Time Series Data Representation ESAX for Financial Applications". In: *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. 2006, pp. x115–x115. DOI: 10.1109/ICDEW.2006.99.

[Llo82] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[Lon+06] Stefano Lonardi, Jessica Lin, Eamonn Keogh, and Bill 'Yuan chi'Chiu. "Efficient Discovery of Unusual Patterns in Time Series". In: *New Gener. Comput* 25 (2006), pp. 61–93. DOI: 10.1007/s00354-006-0004-2.

[Low04] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision* 60 (2004), 91–110.

[Luc+19] Benjamin Lucas, Ahmed Shifaz, Charlotte Pelletier, Lachlan O'Neill, Nayyar Zaidi, Bart Goethals, François Petitjean, and Geoffrey I. Webb. "Proximity Forest: an effective and scalable distance-based classifier for time series". In: *Data Min Knowl Disc* 33 (2019), pp. 607–635.

[Mal+13] Simon Malinowski, Thomas Guyet, René Quiniou, and Romain Tavenard. "1d-SAX: A Novel Symbolic Representation for Time Series". In: *Advances in Intelligent Data Analysis XII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 273–284. ISBN: 978-3-642-41398-8. DOI: 10.1007/978-3-642-41398-8_24.

[Mar09] Pierre-François Marteau. "Time Warp Edit Distance with Stiffness Adjustment for Time Series Matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2 (2009), pp. 306–318. DOI: 10.1109/TPAMI.2008.76.

[MV93] A. Marzal and E. Vidal. "Computation of normalized edit distance and applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 926–932. DOI: 10.1109/34.232078.

[Meg+05] V. Megalooikonomou, Q. Wang, G. Li, and C. Faloutsos. "A multiresolution symbolic representation of time series". In: *21st International Conference on Data Engineering (ICDE'05)*. 2005, pp. 668–679. DOI: 10.1109/ICDE.2005.10.

[MLW04] Vasileios Megalooikonomou, Guo Li, and Qiang Wang. "A Dimensionality Reduction Technique for Efficient Similarity Analysis of Time Series Databases". In: *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. CIKM '04. Washington, D.C., USA: Association for Computing Machinery, 2004, 160–161. ISBN: 1581138741. DOI: 10.1145/1031171.1031203. URL: https://doi.org/10.1145/1031171.1031203.

[Mid+20] Matthew Middlehurst, James Large, Gavin C. Cawley, and A. Bagnall. "The Temporal Dictionary Ensemble (TDE) Classifier for Time Series Classification". In: *ArXiv* abs/2105.03841 (2020).

[MSB23] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. *Bake off redux: a review and experimental evaluation of recent time series classification algorithms*. 2023. arXiv: 2304.13029 [cs.LG].

[MVB19] Matthew Middlehurst, William Vickers, and Anthony Bagnall. "Scalable Dictionary Classifiers for Time Series Classification". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Cham: Springer International Publishing, 2019, pp. 11–19. ISBN: 978-3-030-33607-3.

[MN14] Yasser Mohammad and Toyoaki Nishida. "Robust learning from demonstrations using multidimensional SAX". In: *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*. 2014, pp. 64–71. DOI: 10.1109/ICCAS.2014.6987960.

[MABH11] Almahdi Mohammed Ahmed, Azuraliza Abu Bakar, and Abdul Razak Hamdan. "Harmony Search algorithm for optimal word size in symbolic time series representation". In: *2011 3rd Conference on Data Mining and Optimization (DMO)*. 2011, pp. 57–62. DOI: 10.1109/DMO.2011.5976505.

[Moh+20] Marisa Mohr, Florian Wilhelm, Mattis Hartwig, Ralf Möller, and Karsten Keller. "New Approaches in Ordinal Pattern Representations for Multivariate Time Series". In: *Proceedings of the Thirty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2020)*. 2020.

[MU05] Fabian Mörchen and Alfred Ultsch. "Optimizing Time Series Discretization for Knowledge Discovery". In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. KDD '05. Chicago, Illinois, USA: Association for Computing Machinery, 2005, 660–665. ISBN: 159593135X. DOI: 10.1145/1081870.1081953.

[MF12] Muhammad Marwan Muhammad Fuad. "Genetic Algorithms-Based Symbolic Aggregate Approximation". In: *Data Warehousing and Knowledge Discovery*. Ed. by Alfredo Cuzzocrea and Umeshwar Dayal. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 105–116. ISBN: 978-3-642-32584-7.

[MFM10] Muhammad Marwan Muhammad Fuad and Pierre-François Marteau. "Enhancing the Symbolic Aggregate Approximation Method Using Updated Lookup Tables". In: *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer Berlin Heidelberg, 2010, pp. 420–431. ISBN: 978-3-642-15387-7. DOI: 10.1007/978-3-642-15387-7_46.

[MMK06] Meinard Müller, Henning Mattes, and Frank Kurth. "An efficient multiscale approach to audio synchronization." In: *ISMIR*. Vol. 546. Citeseer. 2006, pp. 192–197.

[MGAMMM17] Aldo Márquez-Grajales, Héctor-Gabriel Acosta-Mesa, and Efrén Mezura-Montes. "An adaptive symbolic discretization scheme for the classification of temporal datasets using NSGA-II". In: *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. 2017, pp. 1–8. DOI: 10.1109/ROPEC.2017.8261674.

[MG+20] Aldo Márquez-Grajales, Héctor-Gabriel Acosta-Mesa, Efrén Mezura-Montes, and Mario Graff. "A multi-breakpoints approach for symbolic discretization of time series". In: *Knowl Inf Syst* 62 (2020), pp. 2795–2834. DOI: 10.1007/s10115-020-01437-4.

[Nav01] Gonzalo Navarro. "A Guided Tour to Approximate String Matching". In: *ACM Comput. Surv.* 33.1 (2001), 31–88. ISSN: 0360-0300. DOI: 10.1145/375360.375365. URL: https://doi.org/10.1145/375360.375365.

[NBY99] Gonzalo Navarro and Ricardo Baeza-Yates. "Fast Multi-dimensional Approximate Pattern Matching". In: *Lecture Notes in Computer Science* (Jan. 1999), pp. 243–257. ISSN: 0302-9743. DOI: 10.1007/3-540-48452-3_18.

[NW70] Saul B. Needleman and Christian D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Journal of Molecular Biology* 48.3 (1970), pp. 443–453. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(70)90057-4.

[NGI17] Thach Le Nguyen, Severin Gsponer, and Georgiana Ifrim. "Time Series Classification by Sequence Learning in All-Subsequence Space". In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)* (2017), pp. 947–958.

[Ngu+19] Thach Le Nguyen, Severin Gsponer, Iulia Ilie, Martin O'Reilly, and Georgiana Ifrim. "Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations". In: *Data Min Knowl Disc* 33 (2019), pp. 1183–1222. DOI: 10.1007/s10618-019-00633-3.

[NI22] Thach Le Nguyen and Georgiana Ifrim. *MrSQM: Fast Time Series Classification with Symbolic Representations*. 2022. arXiv: 2109.01036 [cs.LG].

[Oud+18] L. Oudre, R. Barrois-Müller, T. Moreau, C. Truong, A. Vienne-Jumeau, D. Ricard, N. Vayatis, and P.-P. Vidal. "Template-based step detection with inertial measurement units". In: *Sensors* 18.11 (2018).

[Pap+22] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. "TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection". In: *Proc. VLDB Endow.* 15.8 (2022), 1697–1711. ISSN: 2150-8097. DOI: 10.14778/3529337.3529354. URL: https://doi.org/10.14778/3529337.3529354.

[PJ20] Hoonseok Park and Jae-Yoon Jung. "SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining". In: *Expert Systems with Applications* 141 (2020), p. 112950. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.112950. URL: https://www.sciencedirect.com/science/article/pii/S0957417419306682.

[PLD10a] Ninh D. Pham, Quang Loc Le, and Tran Khanh Dang. "HOT aSAX: A Novel Adaptive Symbolic Representation for Time Series Discords Discovery". In: *Intelligent Information and Database Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 113–121. ISBN: 978-3-642-12145-6. DOI: https://doi.org/10.1007/978-3-642-12145-6_12.

[PLD10b] Ninh D. Pham, Quang Loc Le, and Tran Khanh Dang. "Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases". In: *2010 12th International Asia-Pacific Web Conference*. 2010, pp. 181–187. DOI: 10.1109/APWeb.2010.23.

[PSS08] Chaliaw Phetking, Mohd Noor Md. Sap, and Ali Selamat. "A multiresolution important point retrieval method for financial time series representation". In: *2008 International Conference on Computer and Communication Engineering*. 2008, pp. 510–515. DOI: 10.1109/ICCCE.2008.4580656.

[Pin+13] Tamar Pinhas, Shay Zakov, Dekel Tsur, and Michal Ziv-Ukelson. "Efficient edit distance with duplications and contractions". In: *Algorithms for Molecular Biology* 8.1 (2013), pp. 1–28.

[Qy09] Yan Qiu-yan. "A novel SAX based time streams similarity approach". In: *2009 International Conference on Future BioMedical Information Engineering (FBIE)* (2009), pp. 233–236.

[Rak+12] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. "Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping". In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. Beijing, China: Association for Computing Machinery, 2012, 262–270. ISBN: 9781450314626. DOI: 10.1145/2339530.2339576. URL: https://doi.org/10.1145/2339530.2339576.

[RK13] Thanawin Rakthanmanon and Eamonn Keogh. "Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets". In: *Proceedings of the 2013 SIAM International Conference on Data Mining (SDM)*. 2013, pp. 668–676. DOI: 10.1137/1.9781611972832.74. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972832.74. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972832.74.

[Ram72] Urs Ramer. "An iterative procedure for the polygonal approximation of plane curves". In: *Computer Graphics and Image Processing* 1.3 (1972), pp. 244–256. ISSN: 0146-664X. DOI: https://doi.org/10.1016/S0146-664X(72)80017-0. URL: https://www.sciencedirect.com/science/article/pii/S0146664X72800170.

[RKAJB05] Chotirat Ratanamahatana, Eamonn Keogh, and Stefano Lonardi Anthony J. Bagnall. "A Novel Bit Level Time Series Representation with Implication of Similarity Search and Clustering". In: *Advances in Knowledge Discovery and Data Mining*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 771–777. ISBN: 978-3-540-31935-1.

[Rat+10] Chotirat Ratanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn J. Keogh, Michail Vlachos, and Gautam Das. "Mining Time Series Data". In: *Data Mining and Knowledge Discovery Handbook*. 2010. URL: https://link.springer.com/chapter/10.1007/978-0-387-09823-4_56.

[RK04] Chotirat Ann Ratanamahatana and Eamonn Keogh. "Making Time-series Classification More Accurate Using Learned Constraints". In: *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*. 2004, pp. 11–22. DOI: 10.1137/1.9781611972740.2. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972740.2. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972740.2.

[RK05] Chotirat Ann Ratanamahatana and Eamonn Keogh. "Three Myths about Dynamic Time Warping Data Mining". In: *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM)*. 2005, pp. 506–510. DOI: 10.1137/1.9781611972757.50. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972757.50. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972757.50.

[Rua+20] Hui Ruan, Xiaoguang Hu, Jin Xiao, and Guofeng Zhang. "TrSAX—An improved time series symbolic representation for classification". In: *ISA Transactions* 100 (2020), pp. 387–395. ISSN: 0019-0578. DOI: https://doi.org/10.1016/j.isatra.2019.11.018. URL: https://www.sciencedirect.com/science/article/pii/S0019057819304987.

[RB23] Alejandro Pasos Ruiz and Anthony Bagnall. "Dimension Selection Strategies for Multivariate Time Series Classification with HIVE-COTEv2.0". In: *Advanced Analytics and Learning on Temporal Data: 7th ECML PKDD Workshop, AALTD 2022, Grenoble, France, September 19–23, 2022, Revised Selected Papers*. Springer. 2023, pp. 133–147.

[Rui+21] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and A. Bagnall. "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances". In: *Data Min Knowl Disc* 35 (2021), pp. 401 –449.

[Rut+19] Nicholas Ruta, Naoko Sawada, Katy McKeough, Michael Behrisch, and Johanna Beyer. "SAX Navigator: Time Series Exploration through Hierarchical Clustering". In: *Proceedings of 2019 IEEE Visualization Conference, Short Papers*. 2019, pp. 236–240. DOI: 10.1109/VISUAL.2019.8933618.

[SC78] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (1978), pp. 43–49. DOI: 10.1109/TASSP.1978.1163055.

[SC71] Hiroaki Sakoe and Seibi Chiba. "A Dynamic Programming Approach to Continuous Speech Recognition". In: *Proceedings of the Seventh International Congress on Acoustics*. Vol. 3. 1971, pp. 65–69.

[SYF05] Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. "FTW: Fast Similarity Search under the Time Warping Distance". In: *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS '05. Baltimore, Maryland: Association for Computing Machinery, 2005, 326–337. ISBN: 1595930620. DOI: 10.1145/1065167.1065210. URL: https://doi.org/10.1145/1065167.1065210.

[SC04] Stan Salvador and Philip Chan. "FastDTW: Toward accurate dynamic time warping in linear time and space". In: *KDD workshop on mining temporal and sequential data*. Vol. 6. Seattle, Washington. 2004, pp. 70–80.

[SM83] David Sankoff and Sylvie Mainville. "Common subsequences and monotone subsequences". In: *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison* (1983), pp. 363–365.

[SW10] Anita Sant'Anna and Nicholas Wickström. "A Symbol-Based Approach to Gait Analysis From Acceleration Signals: Identification and Detection of Gait Events and a New Measure of Gait Symmetry". In: *IEEE Transactions on Information Technology in Biomedicine* 14.5 (2010), pp. 1180–1187. DOI: 10.1109/TITB.2010.2047402.

[SW11] Anita Sant'Anna and Nicholas Wickstrom. "Symbolization of time-series: An evaluation of SAX, Persist, and ACA". In: *2011 4th International Congress on Image and Signal Processing* 4 (2011), pp. 2223–2228.

[SP+17] Henrique dos Santos Passos, Felipe Gustavo Silva Teodoro, Bruno Matarazzo Duru, Edenilton Lima de Oliveira, Sarajane M. Peres, and Clodoaldo A. M. Lima. "Symbolic representations of time series applied to biometric recognition based on ECG signals". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 3199–3207. DOI: 10.1109/IJCNN.2017.7966255.

[Sch15] Patrick Schäfer. "The BOSS is concerned with time series classification in the presence of noise". In: *Data Min Knowl Disc* 29 (2015), pp. 1505–1530. DOI: 10.1007/s10618-014-0377-7.

[Sch16] Patrick Schäfer. "Scalable time series classification". In: *Data Min Knowl Disc* 30 (2016), 1273–1298. DOI: 10.1007/s10618-015-0441-y.

[SH12] Patrick Schäfer and Mikael Högqvist. "SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets". In: *Proceedings of the 15th International Conference on Extending Database Technology*. EDBT '12. Berlin, Germany: Association for Computing Machinery, 2012, 516–527. ISBN: 9781450307901. DOI: 10.1145/2247596.2247656.

[SL17] Patrick Schäfer and Ulf Leser. "Fast and Accurate Time Series Classification with WEASEL". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: Association for Computing Machinery, 2017, 637–646. ISBN: 9781450349185. DOI: 10.1145/3132847.3132980.

[SL23] Patrick Schäfer and Ulf Leser. *WEASEL 2.0 – A Random Dilated Dictionary Transform for Fast, Accurate and Memory Constrained Time Series Classification*. 2023. arXiv: 2301.10194 [cs.LG].

[Sen+18] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P. Boedihardjo, Crystal Chen, and Susan Frankenstein. "GrammarViz 3.0: Interactive Discovery of Variable-Length Time Series Patterns". In: *ACM Trans. Knowl. Discov. Data* 12.1 (2018). ISSN: 1556-4681. DOI: 10.1145/3051126. URL: https://doi.org/10.1145/3051126.

[SM13] Pavel Senin and Sergey Malinchik. "SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model". In: *2013 IEEE 13th International Conference on Data Mining*. 2013, pp. 1175–1180. DOI: 10.1109/ICDM.2013.52.

[SCF06] Ari Shapiro, Yong Cao, and Petros Faloutsos. "Style components." In: *Graphics Interface*. 2006, pp. 33–39.

[SZ96] H. Shatkay and S.B. Zdonik. "Approximate queries and representations for large data sequences". In: *Proceedings of the Twelfth International Conference on Data Engineering*. 1996, pp. 536–545. DOI: 10.1109/ICDE.1996.492204.

[SK08] Jin Shieh and Eamonn Keogh. "ISAX: Indexing and Mining Terabyte Sized Time Series". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. Las Vegas, Nevada, USA: Association for Computing Machinery, 2008, 623–631. ISBN: 9781605581934. DOI: 10.1145/1401890.1401966. URL: https://doi.org/10.1145/1401890.1401966.

[Shi+23] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I. Webb. "Elastic similarity and distance measures for multivariate time series". In: *Knowl Inf Syst* (2023). DOI: 0.1007/s10115-023-01835-4.

[SY+17] Mohammad Shokoohi-Yekta, Bing Hu, Hongxia Jin, Jun Wang, and Eamonn Keogh. "Generalizing DTW to the multi-dimensional case requires an adaptive approach". In: *Data Min Knowl Disc* 31 (2017), pp. 1–31. DOI: https://doi.org/10.1007/s10618-016-0455-0.

[SB16] Diego F. Silva and Gustavo E. A. P. A. Batista. "Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation". In: *Proceedings of the 2016 SIAM International Conference on Data Mining (SDM)*. 2016, pp. 837–845. DOI: 10.1137/1.9781611974348.94. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611974348.94. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611974348.94.

[Sil+18] Diego F. Silva, Rafael Giusti, Eamonn Keogh, and Gustavo E. A. P. A. Batista. "Speeding up similarity search under dynamic time warping by pruning unpromising alignments". In: *Data Min Knowl Disc* 32 (2018), 988–1016. DOI: https://doi.org/10.1007/s10618-018-0557-y.

[SW81] T.F. Smith and M.S. Waterman. "Identification of common molecular subsequences". In: *Journal of Molecular Biology* 147.1 (1981), pp. 195–197. ISSN: 0022-2836. DOI: https://doi.org/10.1016/0022-2836(81)90087-5.

[Son+20] Wei Song, Lu Liu, Minghao Liu, Wenxiang Wang, Xiao Wang, and Yu Song. "Representation Learning with Deconvolution for Multivariate Time Series Classification and Visualization". In: *Data Science*. Springer Singapore, 2020, pp. 310–326. ISBN: 978-981-15-7981-3.

[SJA14] Stephan Spiegel, Brijnesh-Johannes Jain, and Sahin Albayrak. "Fast Time Series Classification under Lucky Time Warping Distance". In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. SAC '14. Gyeongju, Republic of Korea: Association for Computing Machinery, 2014, 71–78. ISBN: 9781450324694. DOI: 10.1145/2554850.2554885. URL: https://doi.org/10.1145/2554850.2554885.

[SAD13] Alexandra Stefan, Vassilis Athitsos, and Gautam Das. "The Move-Split-Merge Metric for Time Series". In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2013), pp. 1425–1438. DOI: 10.1109/TKDE.2012.88.

[Str] *Streamlit documentation*.

[Sun+12] Chao Sun, David Stirling, Christian Ritz, and Claude Sammut. "Variance-Wise Segmentation for a Temporal-Adaptive SAX". In: *Proceedings of the Tenth Australasian Data Mining Conference - Volume 134*. AusDM '12. Sydney, Australia: Australian Computer Society, Inc., 2012, 71–77. ISBN: 9781921770142. DOI: 10.5555/2525373.2525382.

[Sun+14] Youqiang Sun, Jiuyong Li, Jixue Liu, Bing-Yu Sun, and Christopher Chow. "An improvement of symbolic aggregate approximation distance measure for time series". In: *Neurocomputing* 138 (2014), pp. 189–198.

[TTK17] Mariem Taktak, Slim Triki, and Anas Kamoun. "SAX-Based Representation with Longest Common Subsequence Dissimilarity Measure for Time Series Data Classification". In: *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. 2017, pp. 821–828. DOI: 10.1109/AICCSA.2017.29.

[Tan+19] Chang Wei Tan, Francois Petitjean, Eamonn Keogh, and Geoffrey I. Webb. *Time series classification for varying length series*. 2019. DOI: 10.48550/arxiv.1910.04341.

[TWP17] Chang Wei Tan, Geoffrey I. Webb, and François Petitjean. "Indexing and classifying gigabytes of time series under time warping". In: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*. 2017, pp. 282–290. DOI: 10.1137/1.9781611974973.32. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611974973.32. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611974973.32.

[Tav21] Romain Tavenard. *An introduction to Dynamic Time Warping*. https://rtavenar.github.io/blog/dtw.html. 2021.

[Tav+20] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. "Tslearn, A Machine Learning Toolkit for Time Series Data". In: *Journal of Machine Learning Research* 21.118 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-091.html.

[Tru+19] Charles Truong, Rémi Barrois-Müller, Thomas Moreau, Clément Provost, Aliénor Vienne-Jumeau, Albane Moreau, Pierre-Paul Vidal, Nicolas Vayatis, Stéphane Buffat, Alain Yelnik, Damien Ricard, and Laurent Oudre. "A Data Set for the Study of Human Locomotion with Inertial Measurements Units". In: *Image Processing On Line* 9 (2019). https://doi.org/10.5201/ipol.2019.265, pp. 381–390.

[TO22] Charles Truong and Laurent Oudre. "Supervised change-point detection with dimension reduction, applied to physiological signals". In: *NeurIPS 2022 Workshop on Learning from Time Series for Health*. 2022. URL: https://openreview.net/forum?id=RmJNi2ieCiC.

[TOV20] Charles Truong, Laurent Oudre, and Nicolas Vayatis. "Selective review of offline change point detection methods". In: *Signal Processing* 167 (2020), p. 107299. ISSN: 0165-1684. DOI: 10.1016/j.sigpro.2019.107299. URL: http://dx.doi.org/10.1016/j.sigpro.2019.107299.

[Vay+22] Titouan Vayer, Romain Tavenard, Laetitia Chapel, Nicolas Courty, Rémi Flamary, and Yann Soullard. "Time Series Alignment with Global Invariances". In: *Transactions on Machine Learning Research Journal* (Oct. 2022). URL: https://hal.science/hal-02473959.

[VJ+19] Aliénor Vienne-Jumeau, Laurent Oudre, Albane Moreau, Flavien Quijoux, Pierre-Paul Vidal, and Damien Ricard. "Comparing Gait Trials with Greedy Template Matching". In: *Sensors* 19.14 (2019). ISSN: 1424-8220. DOI: 10.3390/s19143089. URL: https://www.mdpi.com/1424-8220/19/14/3089.

[VKG02] M. Vlachos, G. Kollios, and D. Gunopulos. "Discovering similar multidimensional trajectories". In: *Proceedings 18th International Conference on Data Engineering*. 2002, pp. 673–684. DOI: 10.1109/ICDE.2002.994784.

[WL75] Robert A. Wagner and Roy Lowrance. “An Extension of the String-to-String Correction Problem”. In: *J. ACM* 22.2 (1975), 177–183. ISSN: 0004-5411. DOI: 10.1145/321879.321880. URL: https://doi.org/10.1145/321879.321880.

[Wan+19] Lin Wang, Faming Lu, Minghao Cui, and Yunxia Bao. “Survey of Methods for Time Series Symbolic Aggregate Approximation”. In: *Data Science*. Springer Singapore, 2019, pp. 645–657. ISBN: 978-981-15-0118-0. DOI: 10.1007/978-981-15-0118-0_50.

[WML05] Qiang Wang, V. Megalooikonomou, and Guo Li. “A symbolic representation of time series”. In: *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005.* Vol. 2. 2005, pp. 655–658. DOI: 10.1109/ISSPA.2005.1581023.

[Wan+13] Xiaoyue Wang, Abdullah Al Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn J. Keogh. “Experimental comparison of representation methods and distance measures for time series data”. In: *Data Min Knowl Disc* 26 (2013), pp. 275–309.

[War05] T. Warren Liao. “Clustering of time series data—a survey”. In: *Pattern Recognition* 38.11 (2005), pp. 1857–1874. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2005.01.025. URL: https://www.sciencedirect.com/science/article/pii/S0031320305001305.

[WKX06] Li Wei, Eamonn Keogh, and Xiaopeng Xi. “SAXually Explicit Images: Finding Unusual Shapes”. In: *Sixth International Conference on Data Mining (ICDM’06)*. 2006, pp. 711–720. DOI: 10.1109/ICDM.2006.138.

[WF94] A. Weigel and F. Fein. “Normalizing the weighted edit distance”. In: *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5)*. Vol. 2. 1994, 399–402 vol.2. DOI: 10.1109/ICPR.1994.576958.

[Wei04] Gary M. Weiss. “Mining with Rarity: A Unifying Framework”. In: *SIGKDD Explor. Newsl.* 6.1 (2004), 7–19. ISSN: 1931-0145. DOI: 10.1145/1007730.1007734. URL: https://doi.org/10.1145/1007730.1007734.

[Win90] William E Winkler. “String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage.” In: (1990).

[WM92] Sun Wu and Udi Manber. “Fast Text Searching: Allowing Errors”. In: *Commun. ACM* 35.10 (1992), 83–91. ISSN: 0001-0782. DOI: 10.1145/135239.135244. URL: https://doi.org/10.1145/135239.135244.

[Yag+17] Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. “DPiSAX: Massively Distributed Partitioned iSAX”. In: *2017 IEEE International Conference on Data Mining (ICDM)*. 2017, pp. 1135–1140. DOI: 10.1109/ICDM.2017.151.

[YAD19] Hamdi Yahyaoui and Reem Al-Daihani. “A novel trend based SAX reduction technique for time series”. In: *Expert Systems with Applications* 130 (2019), pp. 113–123. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.04.026. URL: https://www.sciencedirect.com/science/article/pii/S0957417419302568.

[Yan+03] Albert C.-C. Yang, Shu-Shya Hseu, Huey-Wen Yien, Ary L. Goldberger, and C.-K. Peng. "Linguistic Analysis of the Human Heartbeat Using Frequency and Rank Order Statistics". In: *Phys. Rev. Lett.* 90 (10 2003), p. 108103. DOI: 10.1103/PhysRevLett.90.108103. URL: https://link.aps.org/doi/10.1103/PhysRevLett.90.108103.

[Yan+07] Dragomir Yankov, Eamonn Keogh, Jose Medina, Bill Chiu, and Victor Zordan. "Detecting Time Series Motifs under Uniform Scaling". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. San Jose, California, USA: Association for Computing Machinery, 2007, 844–853. ISBN: 9781595936097. DOI: 10.1145/1281192.1281282. URL: https://doi.org/10.1145/1281192.1281282.

[Yao88] Y.-C. Yao. "Estimating the number of change-points via Schwarz' criterion". In: *Statistics and Probability Letters* 6.3 (1988), pp. 181–189.

[YLV04] Arthur B. Yeh, Dennis K.J. Lin, and Chandramouliswaran Venkataramani. "Unified CUSUM Charts for Monitoring Process Mean and Variability". In: *Quality Technology & Quantitative Management* 1.1 (2004), pp. 65–86. DOI: 10.1080/16843703.2004.11673065.

[YF00] Byoung-Kee Yi and Christos Faloutsos. "Fast Time Sequence Indexing for Arbitrary Lp Norms". In: *Proceedings of the 26th International Conference on Very Large Data Bases*. VLDB '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, 385–394. ISBN: 1558607153.

[Yin+15] Hong Yin, Shuqiang Yang, Xiao-Qian Zhu, Shao-Dong Ma, and Lumin Zhang. "Symbolic representation based on trend features for knowledge discovery in long time series". In: *Frontiers of Information Technology & Electronic Engineering* 16 (2015), pp. 744–758.

[YB07] Li Yujian and Liu Bo. "A Normalized Levenshtein Distance Metric". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1091–1095. DOI: 10.1109/TPAMI.2007.1078.

[Zal+12a] Willian Zalewski, Feng Chung Wu Fabiano Silva, Huei Diana Lee, and André Gustavo Maletzke. "A Symbolic Representation Method to Preserve the Characteristic Slope of Time Series". In: *Advances in Artificial Intelligence - SBIA 2012*. Springer Berlin Heidelberg, 2012, pp. 132–141. ISBN: 978-3-642-34459-6. DOI: 0.1007/978-3-642-34459-6_14.

[Zal+12b] Willian Zalewski, Fabiano Silva, Huei Diana Lee, Andre Gustavo Maletzke, and Feng Chung Wu. "Time Series Discretization Based on the Approximation of the Local Slope Information". In: *Advances in Artificial Intelligence – IBERAMIA 2012*. Springer Berlin Heidelberg, 2012, pp. 91–100. ISBN: 978-3-642-34654-5. DOI: 10.1007/978-3-642-34654-5_10.

[ZY16] Chaw Thet Zan and Hayato Yamana. "An Improved Symbolic Aggregate Approximation Distance Measure Based on Its Statistical Features". In: *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*. iiWAS '16. Singapore, Singapore: Association for Computing Machinery, 2016,

72–80. ISBN: 9781450348072. DOI: 10.1145/3011141.3011146. URL: https://doi.org/10.1145/3011141.3011146.

[ZY17] Chaw Thet Zan and Hayato Yamana. "Dynamic SAX parameter estimation for time series". In: *International Journal of Web Information Systems* 13 (2017), pp. 387–404. DOI: 10.1108/IJWIS-04-2017-0035.

[ZDX20] Haowen Zhang, Yabo Dong, and Duanqing Xu. "Entropy-based Symbolic Aggregate Approximation Representation Method for Time Series". In: *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*. Vol. 9. 2020, pp. 905–909. DOI: 10.1109/ITAIC49862.2020.9339021.

[Zha+18] Ke Zhang, Yuan Li, Yi Chai, and Lei Huang. "Trend-based symbolic aggregate approximation for time series representation". In: *2018 Chinese Control And Decision Conference (CCDC)*. 2018, pp. 2234–2240. DOI: 10.1109/CCDC.2018.8407498.

[Zha+17] Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang, and Thomas Corpetti. "Dynamic Time Warping under limited warping path length". In: *Information Sciences* 393 (2017), pp. 91–107. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2017.02.018. URL: https://www.sciencedirect.com/science/article/pii/S0020025517304176.

[ZT09] Feng Zhou and Fernando Torre. "Canonical Time Warping for Alignment of Human Behavior". In: *Advances in Neural Information Processing Systems*. Ed. by Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta. Vol. 22. Curran Associates, Inc., 2009. URL: https://proceedings.neurips.cc/paper_files/paper/2009/file/2ca65f58e35d9ad45bf7f3ae5cfd08f1-Paper.pdf.

[ZS03] Yunyue Zhu and Dennis Shasha. "Warping Indexes with Envelope Transforms for Query by Humming". In: *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. SIGMOD '03. San Diego, California: Association for Computing Machinery, 2003, 181–192. ISBN: 158113634X. DOI: 10.1145/872757.872780. URL: https://doi.org/10.1145/872757.872780.

[Zun+22] Luciano Zunino, Felipe Olivares, Haroldo V. Ribeiro, and Osvaldo A. Rosso. "Permutation Jensen-Shannon distance: A versatile and fast symbolic tool for complex time-series analysis". In: *Phys. Rev. E* 105 (4 2022), p. 045310. DOI: 10.1103/PhysRevE.105.045310. URL: https://link.aps.org/doi/10.1103/PhysRevE.105.045310.