Rapport de stage

Formation: International Of Computer Sciences

Classification des Séries Temporelles en Présence d'Incertitudes

Rédigé par:

Michael Franklin MBOUOPDA

Année: 2018-2019

Durée: 6 mois

Responsable laboratoire: Engelbert MEPHU NGUIFO

Responsable Académique Institut d'Informatique: Violaine ANTOINE

Classification des Séries Temporelles en Présence d'Incertitudes

Rédigé par:

Michael Franklin MBOUOPDA

Sous la direction de:

Engelbert MEPHU NGUIFO

Remerciements

Je remercie mon directeur de stage Professeur Engelbert MEPHU NGUIFO pour la disponibilité, la patience et la rigueur qu'il a exprimé tout le long de mon stage.

Je remercie mes très chers parents Monsieur Jean TCHOMBEUGUIN et Madame Paulette KENMOE à qui je dois ce que je suis aujourd'hui.

Je remercie tous mes enseignants de l'institut d'informatique pour m'avoir transmis des connaissances d'excellente qualité.

Je remercie tous ceux qui ont contribué de près ou de loin, et chacun à leur manière à la réalisation de ce travail.

Abstract

La classification des séries temporelles est une tâche qui trouve ses applications dans des domaines variés tels que la météorologie, la médecine et la physique. Elle vise à classer automatiquement des objets réels modélisés sous forme de séries temporelles. Les différents algorithmes permettant d'accomplir cette tâche de classification porte majoritairement sur des séries temporelles certaines. L'incertitude est souvent traitée de manière implicite. Or nous savons qu'à toute mesure effectuée, est associée une incertitude qui souvent est exprimée de manière explicite sur la donnée mesurée. En utilisant les techniques de propagation de l'incertitude, nous proposons dans ce mémoire une adaptation de la méthode de classification basée sur la transformation des séries en shapelet, pour traiter les séries temporelles incertaines. Nous avons implémenté notre approche en JAVA et les expérimentations menées sur 21 jeux de données de UCR montrent l'efficacité de notre contribution.

Mot clés: Série temporelle, classification, transformation shapelet, incertitude.

Abstract

Time series classification is a task used in a variaty of domains such as meteorology, medecine and physics. The goal of this task is to classify real world objects modelized as time series data. The existing algorithms used to perform this task are mostly about time series that are supposed to be without uncertainty. Uncertainty is often handle implicitly. However we know that any measurement is subject to uncertainty, and this uncertainty is sometime explicitly expressed. Using error propagation techniques, we propose in this report, an adaptation of the shapelet transform algorithm for classifying uncertain time series. We have implemented our method using the Java programming language and the experiments we run over 21 datasets from UCR show the effectiveness of our approach.

Keywords: time series, classification, shapelet transform, uncertainty.

Contents

List of Figures

List of Tables

Liste des symboles

DTW Dynamic Time Warping

IG Information Gain

LSST Large Synoptic Survey Telescope

MLP Multi Layer Perceptron

RandF RandomForest

RotF Rotation Forest

SAX Symbolic Aggragate approXimation

SMO Sequential Minimal Optimization

ST Shapelet Transform

SVM Support Vector Machines

TWED Time Warped Edit Distance

UST Uncertain Shapelet Transform

UED Uncertain Euclidean Distance

Liste des algorithmes

Présentation du cadre

J'ai effectué mon stage au sein du LIMOS[1] au sein du thème *Données, Services, Intelligence* qui fait partie de l'axe de recherche *Systèmes d'Information et de Communication*. Cet axe se concentre sur l'acquisition, le transfert, le traitement et l'analyse de données. Ces données sont généralement massives, incomplètes et hétérogènes. Elles peuvent être acquises et transmises par l'intermédiaire d'un réseau de capteurs sans fil, et sont stockées dans une base de données qui peut être distribuée. Elles y sont traitées et analysées, afin d'identifier leurs propriétés.

Laboratoire: LIMOS

Le Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS) est une Unité Mixte de Recherche (UMR 6158) en informatique, et plus généralement en Sciences et Technologies de l'Information et de la Communication (STIC).

Le LIMOS est principalement rattaché à l'Institut des Sciences de l'Information et de leurs Interactions (INS2I) du CNRS et de façon secondaire à l'Institut des Sciences de l'Ingénierie et des Systèmes (INSIS). Il a pour tutelles académiques l'Université Clermont Auvergne et l'Ecole Nationale Supérieure des Mines de Saint-Etienne (EMSE), et comme établissement partenaire l'école d'ingénieur SIGMA. Le LIMOS est membre des labex IMOBS3 et ClerCVolc et de la fédération de recherche en Environnement FR 3467 (qui regroupe 17 laboratoires UCA et INRA du site de Clermont-Ferrand). Il est membre associé de la fédération MODMAD (MODélisation Mathématique et Aide à la Décision, FED 4169) portée par l'Université Jean Monnet de Saint-Etienne.

Le positionnement scientifique du LIMOS est centré autour de l'Informatique, la Modélisation et l'Optimisation des Systèmes Organisationnels et Vivants. Les principaux thèmes de recherche développés au sein du laboratoire sont :

- Optimisation Combinatoire,
- Algorithmique, Graphes, Complexité,
- Méta-Modélisation,
- Données, Services, Intelligence,

[1] https://limos.fr

- Réseaux de capteurs,
- Production de biens,
- Conception et Planification de services.

Projet TransiXplore

Débuté en Janvier 2018 pour une durée de 2 ans, le projet TransiXplore[2] est l'un des projets sur lesquels travaille le LIMOS. Le but de ce projet est de créer de nouveaux outils adaptés à l'analyse des données astronomiques mobiles. L'un de ces objectifs est la classification d'objets astronomiques filmés par le télescope LSST[3]. Ces objets sont représentés sous forme de séries temporelles incertaines.

[2] https://limos.fr/news_project/109
[3] https://lsst-tvssc.github.io/

Introduction

Les séries temporelles sont utilisées dans un ensemble riche et varié de domaines. Cet ensemble comprend la finance, la biologie, la médecine et l'ingénierie. Encore appelée série chronologique, il s'agit d'une suite ordonnées de valeurs. Par exemple les séries temporelles ont été utilisées pour modéliser les gènes par Bar-Joseph et al. [2012]; Les données financières évoluant avec les temps, les séries temporelles sont très souvent utilisées pour les modéliser afin de les analyser [Tsay, 2005]; En utilisant une représentation en séries temporelles, Siyou Fotso [2018] a fait une analyse de la locomotion en chaise roulante. Un autre exemple de domaine d'application des séries temporelles est la météorologie.

L'analyse des séries temporelles peut consister soit à faire un partitionnement(clustering), soit à associer à chaque série temporelle une classe ou un groupe (classification), dans tous les cas il est question d'identifier les tendances qu'il y a dans un ensemble de séries temporelles. Dans ce rapport, nous nous focalisons sur la tâche de classification des séries temporelles incertaines. Étant donnée un ensemble D de séries temporelles incertaines, chacune avec sa classe (son type), la tâche de classification des séries temporelles consiste à trouver une fonction permettant de passer de l'espace des séries temporelles possibles à l'espace des classes.

Il existe dans la littérature plusieurs méthodes de classification des séries temporelles. Parmi ces méthodes nous avons celles utilisant des distances élastiques telles que Dynamic Time Warping (DTW) avec ses différentes variantes, celles basées sur des sous séquences appelées shapelets, celles basées sur des caractéristiques présentes à des intervalles spécifiques et enfin celles basées sur l'analyse spectrale. Les méthodes par shapelet sont celles qui généralisent le plus. Cependant il n'y a pas de version qui traite les données incertaines. Nous proposons une nouvelle méthode shapelet appelée Uncertain Shapelet Transform pour pallier cette limitation.

La suite de ce mémoire est organisée comme suit: Le Chapitre 1 présente les concepts généraux et quelques définitions importantes. Le Chapitre 2 présente une revue de la littérature sur la classification des séries temporelles avec des approches shapelets. Le Chapitre 3 présente tout d'abord la transformation shapelet avec FOTS comme distance entre sous séquences, et ensuite **UST** (Uncertain Shapelet Transform), une adaptation de la méthode de transformation shapelet qui prend en compte l'incertitude dans les données. Le Chapitre 4 présente les expérimentations et les résultats obtenus. Pour finir, le Chapitre 5 conclut ce rapport de stage et présente les perspectives.

1

Concepts généraux

Dans ce premier chapitre nous allons définir des concepts qui sont nécessaires à la compréhension de notre travail. Nous présenterons les différentes classes d'approche de classification des séries temporelles; nous parlerons ensuite des séries temporelles incertaines; et enfin nous définirons la notion de shapelet, qui est la base de notre travail.

Commençons tout de suite par définir formellement les notions de **série temporelle** et de **classification des séries temporelles**

Définition 1.1 (Série temporelle) *Une série temporelle T est une suite ordonnée de m valeurs. m est appelée la taille de la série.*

$$T = \{t_1, t_2, ..., t_m\}$$

Définition 1.2 (Classification des séries temporelles) *Soit un jeu de données $D = \{< T_1, c_1 > , < T_2, c_2 >, ..., < T_n, c_n >>\}$, où chaque T_i est une série temporelle et c_i la classe associée. Le problème de classification de ce jeu de données consiste à trouver une fonction f telle que*

$$f(T_i) = c_i$$

1.1 Approches de classification des séries temporelles

Il existe dans la littérature plusieurs méthodes pour résoudre le problème de classification des séries temporelles. Selon les caractéristiques discriminatoires utilisées, on peut les classer en cinq groupes [Lines et al., 2018]: série brute, intervalle, shapelet, dictionnaire et spectrale. Les 4 dernières approches supposent que les séries temporelles ont des caractéristiques dont la connaissance suffit pour faire la classification.

Série brute

Les approches de ce groupe considèrent les séries brutes (c'est-à-dire qu'il n'y a pas d'extraction de caractéristiques) pour faire la classification. Ces approches comparent deux séries comme des vecteurs en utilisant ce qu'on appelle les distances élastiques. Parmi ces distances, la plus efficace est Dynamic Time Warping (DTW)[Jeong et al., 2011; Lines et al., 2018]. Batista et al. [2013] ont proposé une distance dont la complexité est invariante. Deux autres distances utilisées sont Time Warped Edit Distance (TWED) et la distance euclidienne. Ces différentes mesures sont généralement combinées avec le modèle de classification **1-NN** (1 - Plus Proche Voisin). Lines and Bagnall [2015] ont construit un modèle ensembliste qui combine plusieurs distances élastiques.

Intervalle

Contrairement aux approches qui considèrent les séries brutes, les approches par intervalles supposent qu'il existe des intervalles de temps contenant des caractéristiques suffisantes pour faire la classification. Ces approches se décomposent en trois principales phases: sélection des intervalles, extraction des caractéristiques puis classification proprement dite.

Dans certaines situations, les intervalles discriminatoires peuvent ne pas avoir des positions fixes; en d'autres mots, ils peuvent apparaître n'importe où dans la série temporelle. Ce genre de situation est inappropriée pour les approches par intervalle, dans lesquelles la position des intervalles est fixée une fois pour toute.

Shapelet

Ici on suppose que les caractéristiques discriminatoires peuvent être n'importe où dans la séries. Les caractéristiques discriminatoires sont appelées **Shapelets**; ce sont des sous séquences représentatives pour les classes.

Définition 1.3 (Sous séquences) *Une sous séquence S d'une série temporelle T est une suite ordonnée de l valeurs consécutives de T.*

$$S = \{t_{i+1}, \ldots, t_{i+l}\}$$

Une sous séquence est représentative pour une classe lorsqu'elle est commune aux instances de la classe.

Dictionnaire

Les approches shapelets définissent l'appartenance d'une instance à une classe par la **présence ou non** du shapelet (pattern) dans l'instance. Il arrive que l'appartenance à une classe soit déterminée par la **fréquence** d'apparition des patterns. Ainsi des instances appartenant à des classes différentes peuvent avoir les mêmes patterns, mais avec des nombres d'occurrences différents.

Spectrale

Pour certains jeu de données, les caractéristiques discriminatoires peuvent être difficiles à identifier dans le domaine du temps (approches intervalles, shapelets et dictionnaires), dans ce cas le domaine de la fréquence est plus approprié. Les approches qui traitent le domaine de la fréquence sont généralement basées sur la transformation de Fourrier ou sur l'analyse spectrale.

Dans la suite de ce mémoire nous travaillerons uniquement sur les approches shapelets. Ceci pour deux raisons: Premièrement on aimerait voir si FOTS[Siyou Fotso, 2018] fonctionne aussi bien pour la classification que pour le clustering. Deuxièmement les physiciens pensent que les approches shapelets pourraient être très efficaces sur les jeux de données du projet TransiXplore.

1.2 Séries temporelles incertaines

Contrairement à une **erreur** qui peut être évitée en faisant attention, l'incertitude est inévitable et est toujours présente[Taylor, 1996]. L'incertitude peut être réduite mais elle ne peut pas être totalement éliminée. Elle peut provenir de diverses sources aussi variées les unes des autres. Supposons qu'on veut trouver la taille d'une personne qui se trouve à 500 mètres de nous, on pourrait dire que la personne a une taille comprise entre 150 centimètres et 160 centimètres. Il s'agit là d'une mesure (estimation) incertaine de la taille de la personne. l'incertitude ici est due au fait que la personne se trouve loin; on peut donc la réduire en se rapprochant. On peut davantage réduire l'incertitude en utilisant un mètre, mais cela ne l'éliminera toujours pas car la graduation du mètre limite déjà le nombre de valeurs exactes que nous pouvons mesurer. Si la graduation est centimètre par centimètre on peut être précis au centimètre près, si c'est millimètre par millimètre on peut être précis au millimètre près. Dans le cas d'une mesure

faite par un ordinateur, l'incertitude pourrait provenir de la précision du processeur (simple ou double précision).

Il existe trois types d'incertitude [Jiao, 2015]:

- **Incertitude due à l'incomplétude**: Il s'agit ici de l'absence d'information. Par exemple, après un recensement des populations de chaque ville d'un pays, le nombre d'habitants d'une des villes est inconnu.
- **Incertitude due à l'imprécision**: C'est lorsque la valeur d'une variable est donnée mais pas avec suffisamment de précision. Toujours avec l'exemple du recensement, on pourrait avoir dans le rapport un habitant dont le sexe est inconnu; dans ce cas on sait juste que c'est soit un homme, soit une femme.
- **Incertitude due à l'absence de fiabilité**: Ici l'information est complète et précise, cependant elle peut être fausse. Ceci est le cas dans le jeu de données du challenge Plasticc[The PLAsTiCC team et al., 2018] où la valeur du flux lumineux est accompagnée d'un niveau de fiabilité.

Dans ce mémoire, nous ne traitons que l'incertitude due à l'absence de fiabilité. Pour des raisons de simplicité, nous utiliserons simplement le mot incertitude pour faire référence à l'incertitude due à l'absence de fiabilité.

Donnons maintenant une définition formelle de ce qu'est l'incertitude. La définition suivante provient du livre de Taylor [1996].

Définition 1.4 (Incertitude) *Une mesure x est dite incertaine lorsqu'on ne connaît pas avec exactitude sa valeur, mais on connaît dans quel intervalle sa valeur exacte se trouve. On note:*

$$x = x_{best} \pm \delta x$$

ce qui signifie que la valeur exacte de x est dans l'intervalle $[x_{best} - \delta x, x_{best} + \delta x]$. x_{best} est la meilleure estimation de la valeur exacte de x et δx est l'incertitude sur cette estimation.

En pratique, lorsque l'incertitude est suffisamment petite au point de ne pas avoir une influence significative, on suppose qu'elle vaut 0.

Définition 1.5 (Série temporelle incertaine) *C'est une série temporelle dont les valeurs sont incertaines.*

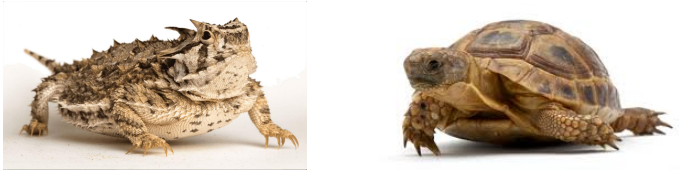
L'objectif de ce mémoire est de faire la classification des séries temporelles incertaines, en utilisant une approche basée sur les shapelets.

1.3 Notion de shapelet

La notion de **Shapelet** a été introduite par Ye and Keogh [2009]. Derrière ce gros concept se cache en réalité une idée toute simple et intuitive. Avant de revenir aux séries temporelles, prenons deux exemples simples qui nous aideront à comprendre:

Supposons que nous voulons reconnaître un proche, notre maman, notre frère ou un ami; a-t-on besoin de voir la personne entièrement pour l'identifier? Bien sûr que non; il suffit de voir sa tête par exemple, ou ses yeux, ou sa démarche, ou alors entendre sa voix.

Pour différentier un lézard à cornes (Fig 1a) d'une tortue (Fig 1b), il n'est pas nécessaire de regarder entièrement les images; une différence très visible entre les deux c'est la présence ou non des cornes.



(a) Lézard à cornes (b) Tortue

Figure 1 – Lézard à cornes vs Tortue.

De ces deux exemples, on comprend que pour reconnaître des objets (personnes, animaux, plantes, séries temporelles), il suffit d'identifier dans ces objets des caractéristiques dites discriminatoires (couleur des yeux, démarche, voix, présence ou pas de cornes). Dans le jargon des séries temporelles, ces caractéristiques discriminatoires sont appelées des **shapelets**.

Une fois les caractéristiques discriminatoires déterminées, elles sont utilisées pour classer un objet selon les shapelets qu'il contient. Une telle approche de classification a trois principaux points forts[Ye and Keogh, 2009]:

- Interprétabilité: On peut toujours dire pourquoi un objet est dans telle classe et pas dans une autre; car les objets d'une même classe auront les mêmes caractéristiques discriminatoires, et les objets de classes distinctes auront des shapelets différents;

- Robustesse: L'identification des shapelets permet de réduire l'impact des bruits sur la précision de classification. Cette étape d'identification peut être vue comme de la sélection de caractéristiques.
- Rapidité de classification: La classification par shapelets est beaucoup plus rapide que les approches qui utilisent les séries brutes. Ceci est dû au fait que les shapelets sont en général de taille beaucoup plus petite que les séries brutes.

Quelques définitions

Soit $T = \{t_1, t_2, ..., t_m\}$ une séries temporelle et soient $S = \{s_1, s_2, ..., s_l\}$ et $R = \{r_1, r_2, ..., r_l\}$ deux sous séquences de même taille, avec $l < m$.

Définition 1.6 (Distance entre deux sous séquences) *La distance entre deux sous séquences S et R de même taille est une valeur positive d donnée par l'application d'une fonction de distance. Cette fonction doit être symétrique.*

$$dist(S, R) = dist(R, S) = d, d \geq 0.$$

Plus deux sous séquences sont similaires, plus la distance entre elles est proche de 0; plus elles sont différentes, plus la distance entre elles est grande: c'est une **mesure de dissimilarité**.

Définition 1.7 (Distance entre une sous séquence et une série temporelle) *La distance entre une sous séquence S et une série temporelle T est la distance entre la sous séquence S et la sous séquence de T de même taille que S qui soit la plus proche de S.*

$$dist(S, T) = dist(T, S) = \min\{dist(S, R)\ \forall R \in T\}$$

Définition 1.8 (Entropie) *Soit D un jeu de données constitués de deux classes A et B. $p(A)$ et $p(B)$ sont les proportions respectives des objets de classe A et de classe B dans D. L'entropie de D est une mesure de l'homogénéité entre les objets d'une même classe. Elle est donnée par:*

$$H(D) = -p(A)\log p(A) - p(B)\log p(B)$$

Définition 1.9 (Séparateur) *Un séparateur sp de D est une sous séquence qui le divise en deux*

sous ensembles D_1 et D_2, avec

$$D_1 = \{T \mid \text{dist}(T, sp) \leq \varepsilon \, , \forall T \in D\}, D_2 = \{T \mid \text{dist}(T, sp) > \varepsilon \, , \forall T \in D\}$$

Définition 1.10 (Gain d'information) *Le gain d'information (IG) est une mesure de la qualité d'un séparateur.*

$$IG(D, sp) = H(D) - (\frac{|D_1|}{|D|}H(D_1) + \frac{|D_2|}{|D|}H(D_2))$$

Nous pouvons désormais définir formellement ce qu'est un shapelet

Définition 1.11 (Shapelet) *Un shapelet S pour un jeu de données D est un séparateur de D en deux parties D_1 et D_2 en maximisant le gain d'information.*

$$S = \underset{S}{\text{argmax}}(IG(D, S))$$

Dans ce chapitre, nous avons vue les concepts sur lesquels se basent les méthodes de classification à base de shapelets. Le prochain chapitre présente comment ces notions sont effectivement utilisées dans la littérature pour faire la classification des séries temporelles.

2

État de l'art

Le chapitre précédent a introduit les concepts de base utilisés par les approches shapelets de classification des séries temporelles. Parmi ces concepts nous avons le **shapelets** et le **Gain d'information**. Un shapelet est une sous séquence qui divise le jeu de données avec un gain d'information maximal. Dans ce chapitre nous ne présenterons que les travaux les plus récents sur la classification des séries temporelles avec des approches basées sur les shapelets. Cependant la revue de la littérature faite par Bagnall et al. [2017] permet d'avoir une vision plus globale de la classification des séries temporelles.

2.1 Arbre de décision shapelet

La première utilisation des shapelets fut proposée par Ye and Keogh [2009]. Ils construisent un arbre de décision permettant de faire la classification des séries temporelles à base de shapelets. À chaque noeud de l'arbre, la décision est faite en fonction de la distance entre les instances et le shapelet du noeud. La phase d'induction de l'arbre consiste donc à trouver les shapelets s_i de chaque noeud ainsi que les paramètres k_i pour la séparation. La Figure 2 est une illustration d'un arbre de décision shapelet.

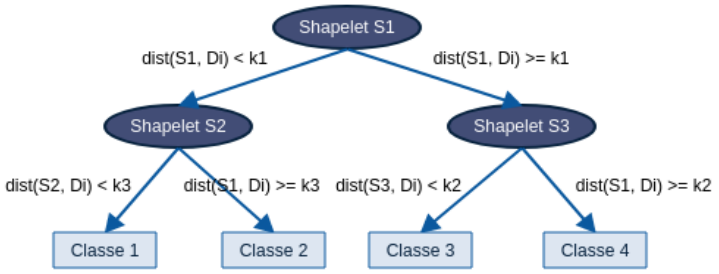


Figure 2 – Arbre de décision shapelet

Comme nous l'avons dit au début de ce chapitre, cette approche de classification des séries

temporelles a l'avantage d'être interprétable, robuste et rapide lors de l'inférence. Cependant elle a aussi des points négatifs, deux pour être précis:

- Le temps nécessaire pour l'induction peut être long: En effet pour un jeux de données de n séries temporelles dont la plus longue a une taille de m, le temps nécessaire pour trouver les shapelets est de l'ordre de $O(n^2m^4)$. C'est une complexité en temps non négligeable, surtout lorsqu'on a des séries de très grande taille.
- La dépendance au modèle d'arbre de décision: Il n'est pas possible de bénéficier des avantages des modèles de classification plus sophistiqués tels que les réseaux de neurones, les modèles ensemblistes ou encore les modèles d'apprentissage profond.

La prochaine section présente un algorithme de recherche des shapelets dont la complexité en temps est considérablement meilleure.

2.2 Fast Shapelets

Étant donné que c'est la longueur des séries qui influence majoritairement la durée nécessaire pour trouver les différents shapelets, Rakthanmanon and Keogh [2013] vont utiliser **SAX**[Lin et al., 2007] pour réduire la longueur des séries temporelles avant d'effectuer la recherche des shapelets. Ils obtiennent un algorithme de recherche des shapelets dont la complexité est $O(n^2m^3)$. Un point important à noter ici est que la précision de classification reste la même.

Cette approche a l'inconvénient d'avoir plus de paramètres à définir; de plus elle est toujours dépendante du modèle d'arbre de décision. La prochaine section présente une approche qui supprime cette dépendance.

2.3 Classification par Transformation Shapelet

Contrairement aux approches précédentes qui dépendent du modèle d'arbre de décision, la classification par transformation shapelet (**Shapelet Transform**[Hills et al., 2014]) ne dépend d'aucun modèle de classification particulier. Cette approche sépare la recherche des shapelets de la classification proprement dite.

En effet la recherche des shapelets est une étape de sélection des caractéristiques, les caractéristiques étant les shapelets. Un modèle de classification quelconque peut ensuite être utilisé pour faire la classification en fonction des caractéristiques extraites.

L'inconvénient de la transformation shapelet est qu'elle nécessite plus de mémoire pour le stockage des shapelets candidats. Tout de même, en plus d'avoir des précisions de classification très appréciables, cette approche offre plus de flexibilités et de possibilités.

2.4 Apprentissage des shapelets par optimisation

Grabocka et al. [2014] considèrent le processus de recherche des shapelets comme un problème d'optimisation. Ainsi, partant d'un ensemble de shapelets aléatoires, ils utilisent la descente stochastique du gradient pour modifier de façon itérative les shapelets afin de réduire le taux d'erreur de classification. L'approche garantit que si des shapelets optimaux existent alors elle peut les découvrir.

En procédant ainsi, l'espace des shapelets n'est plus limité au jeu de données mais devient infini. De plus les shapelets ne sont pas choisis de façon individuelle, c'est le groupe de shapelets qui minimisent au mieux le taux d'erreur de classification qui sera choisi.

Comme tout problème d'optimisation, cette approche nécessite une bonne initialisation des shapelets car cela détermine le temps nécessaire pour la convergence de l'algorithme. Par ailleurs, l'approche ne garantit pas que chacun des shapelets obtenus sera une représentation d'une caractéristique compréhensible par un expert du domaine; car il ne s'agira pas forcément de sous séquences du jeu de données.

Nous terminons cette état de l'art par une approche qui n'est pas réellement une approche shapelet, mais plutôt une combinaison d'approches de classification des séries temporelles. Parmi tous les modèles évalués dont nous avons connaissance, c'est lui qui donnes les meilleurs résultats [Lines et al., 2018].

2.5 HIVE-COTE

HIVE-COTE [Lines et al., 2018] (Hierarchical Vote Collective of Transformation-Based Ensembles) est un modèle de classification des séries temporelles basé sur un vote collaboratif de plusieurs modèles utilisant des approches différentes. L'idée de HIVE-COTE est d'être capable de s'adapter au type de jeu de données. Ainsi, selon le jeux données, HIVE-COTE veut pouvoir utiliser l'approche la plus adaptée (séries brutes, intervalle, shapelet, dictionnaire ou spectrale). HIVE-COTE est de loin le meilleur modèle[Lines et al., 2018], suivi de Flat-COTE[Bagnall et al., 2016] qui est aussi un modèle combinant plusieurs approches. Vient ensuite le modèle **ST**, qui est un modèle par transformation shapelet.

Nous concluons cet état de l'art par le tableau 1. Il présente une comparaison des différents modèles que nous avons cités. Les critères de comparaison sont:

- l'interprétabilité: Il s'agit ici de savoir si chacun des shapelets trouvé représente une caractéristiques compréhensible par un expert du domaine. Lorsque tous les shapelets trouvés sont des sous séquences du jeu de données, l'interprétabilité est garantie. Dans le cas contraire, elle peut ne pas être garantie.
- la dépendance à un modèle de classification: C'est la possibilité d'utiliser tout modèle de classification supervisée pour faire les prédictions sur le jeu de données obtenu après transformation. Ce critère est représenté par la colonne *Modèle prédéfini* qui ne contient rien si le modèle de classification est au choix, sinon elle contient le nom du modèle.
- l'espace des shapelets candidats: il s'agit de l'ensemble des shapelets candidats. Ce critère est représenté par la colonne *Espace shapelet* et vaut «Dataset »lorsque les shapelets candidats sont uniquement des sous séquences du jeu de données. Lorsque les shapelets candidats peuvent être des sous séquences quelconques, ce critère vaut «Σ »qui est l'ensemble de toutes les séries temporelles possibles.

	Interprétabilité	Modèle prédéfini	Espace shapelet
Shapelet DT (section 2.1)	Garantie	DT	Dataset
FS (section 2.2)	Garantie	DT	Dataset
ST (section 2.3)	Garantie	-	Dataset
LS (section 2.4)	Non garantie	Ridge LR	Σ
HIVE-COTE (section 2.5)	Garantie	-	Σ

Table 1 – Synthèse de l'état de l'art

Nous avons présenté dans ce chapitre les approches shapelets de classification des séries temporelles, chacune avec ses forces et ses faiblesses. Toutes ces approches supposent que les données sont certaines. Comment ces approches se comporteraient-elles si elles étaient appliquées sur des séries temporelles incertaines? La prise en compte de l'incertitude lorsque celle-ci est disponible ne permettrait-elle pas d'avoir de meilleures performances? N'ayant pas connaissance d'un modèle de classification des séries temporelles qui prend en compte l'incertitude lorsque celle-ci est disponible, nous allons en proposer une par adaptation de la méthode basée sur la transformation shapelet.

3

Classification des séries temporelles incertaines

Nous avons vu dans le chapitre précédent comment est-ce que les shapelets ont été utilisés pour faire la classification des séries temporelles. Nous avons également vu que les différentes approches proposées ne prenaient pas en compte l'incertitude. Dans ce chapitre nous allons modifier la méthode de transformation shapelet[Hills et al., 2014] afin de l'utiliser lorsque les séries temporelles sont incertaines. Pour cela nous avons identifié deux possibilités: la première consiste à absorber l'incertitude durant la phase de transformation; la seconde consiste à propager l'incertitude durant la phase de transformation, puis de faire la classification en prenant en compte l'incertitude qui a été propagée.

L'algorithme 1 est un pseudo code de la transformation shapelet. Cet algorithme que nous allons adapter afin de faire la classification des séries temporelles incertaines utilise trois procédures:

- **GenererCandidate(T, Min, Max)**: qui prend en entrée une série temporelle T, et deux nombres entiers *Min* et *Max*. Elle retourne toutes les sous séquences de T dont la taille va de *Min* à *Max*.
- **EvaluerCandidats(cands, D)**: cette procédure prend un ensemble de sous séquences *cands* et un jeu de données D de séries temporelles. Elle va calculer et retourner la qualité de chacune des sous séquences. La qualité d'un candidat est le gap d'information qu'on obtient sur D en l'utilisant comme séparateur.
- **ExtraireMeilleurs(C, Q, k)**: Cette procédure prend en paramètre un ensemble de candidats C, les qualités de chacun d'eux Q et un entier k. Elle va retourner les k candidats de plus grande qualité.

- **Transformer(D, S, k)**: Cette procédure qui est détaillée par l'algorithme 2 fait la transformation shapelet proprement dite. Elle prend en entrée un jeu de données D de séries temporelles et un ensemble S de sous séquences. Pour chaque instance de D elle va construire un vecteur de taille $|S|$, la composante i du vecteur étant la distance entre l'instance et la i-ème sous séquence. La procédure retourne un jeu de données dont chaque instance est un vecteur de distance entre les instances de D et les sous séquences dans S.

Algorithm 1: Transformation Shapelet

Input: $D[n, m]$: Dataset,
k: nombre de shapelets,
MINLEN: taille min,
MAXLEN: taille max
Result: $T[n, k]$: Le jeu de données transformé
$C \leftarrow \emptyset$; $Q \leftarrow \emptyset$;
for $i \in 1..n$ **do**
    $cands \leftarrow GenererCandidats(D_i, MINLEN, MAXLEN)$;
    $qualites \leftarrow EvaluerCandidats(cands, D)$;
    $C \leftarrow C + cands$;
    $Q \leftarrow Q + qualites$;
$S \leftarrow ExtraireMeilleurs(C, Q, k)$;
$T \leftarrow Transformer(D, S, k)$;
return T

Algorithm 2: Transformer

Input: $D[n, m]$: Dataset de n séries de longueur m,
S: les k shapelets
k: nombre de shapelets
Result: $D[n, k]$
for $i \in 1..n$ **do**
    $temp \leftarrow \emptyset$;
    for $j \in 1..k$ **do**
        $temp_j \leftarrow ED(T_i, S_j)$;
    $D_i \leftarrow temp_j$
return D

3.1 Approche par absorption de l'incertitude

Dans la transformation shapelet, les procédures *EvaluerCandidats* et *Transformer* utilisent la distance euclidienne pour calculer la distance entre les sous séquences et les séries temporelles. L'incertitude peut être prise en compte en utilisant une mesure de similarité qui prend en compte l'incertitude. Pour ce faire, il suffit de remplacer la distance euclidienne utilisée dans la méthode de transformation shapelet par une mesure de similarité qui prend en compte l'incertitude.

Il existe plusieurs mesures de similarités prenant en compte l'incertitude telle que FOTS, MUNICH, PROUD et DUST Siyou Fotso et al. [2018]. MUNICH prend en compte l'incertitude en utilisant plusieurs observations de l'état de la série à un instant donné [Dallachiesa et al., 2012]. PROUD et DUST quant à eux considèrent que chaque état d'une série temporelle est représenté par une variable aléatoire continue dont la loi de distribution est connue [Dallachiesa et al., 2012]. Contrairement à PROUD et DUST, FOTS n'a pas besoin de connaître la distribution de chaque observation, il n'a pas non plus besoin d'avoir plusieurs observations d'une série temporelle à un même instant comme MUNICH. En effet FOTS suppose que les séries sont incertaines mais n'a aucune connaissance précise sur cette incertitude. Soient deux séries temporelles X et Y, la dissimilarité entre elles, selon FOTS se calcule comme suit:

$$FOTS(X,Y) = ||U_X - U_Y||_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{k}(U_X - U_Y)_{ij}^2}$$

U_X et U_Y sont les décompositions en vecteurs propres des matrices d'autocorrélation de X et de Y respectivement. $||U||_F$ est la norme de Frobenius de la matrice U.

La décomposition en vecteurs propres de la matrice d'auto corrélation permet a FOTS de capturer uniquement les tendances représentatives dans la série temporelle. En supposant que l'incertitude n'est pas une tendance représentative, on dit que FOTS absorbe l'incertitude. Pour ces raisons, et surtout à cause des contraintes du stage, nous avons remplacé la distance euclidienne par FOTS. En plus de capturer les tendances oscillatoires et apériodiques dans les séries temporelles, FOTS absorbe l'incertitude; de plus Siyou Fotso [2018] montre que remplacer la distance euclidienne par FOTS permet d'avoir un meilleur clustering des séries temporelles incertaines. Nous pensons que FOTS pourrait également améliorer la précision dans une tâche de classification de séries temporelles incertaines.

Le principal inconvénient de FOTS est le temps nécessaire pour son calcul. En effet tandis que la distance euclidienne a une complexité linéaire, FOTS a une complexité quadratique.

La complexité de la transformation shapelet avec FOTS est de $O(n^2m^5)$ (au lieu de $O(n^2m^4)$ avec la distance euclidienne). Pour cette raison FOTS est envisageable lorsque les séries temporelles ne sont pas de longue taille. Par ailleurs nous ne savons pas jusqu'à quel point FOTS est capable de résister à l'incertitude.

3.2 Approche par propagation de l'incertitude

Au lieu d'absorber l'incertitude pendant la phase de transformation shapelet, nous allons la propager en utilisation les techniques de propagation de l'incertitude[Taylor, 1996]. Ces techniques permettent de calculer l'incertitude qu'on obtient en appliquant les opérateurs d'addition, de soustraction, de multiplication, de division et de puissance sur des valeurs incertaines. Nous avons appelé notre méthode **UST** pour Uncertain Shapelet Transform. Il s'agit d'une transformation shapelet dans laquelle la distance euclidienne est remplacée par une adaptation de la distance euclidienne pour les vecteurs incertains. Nous avons appelé cette distance incertaine UED pour Uncertain Euclidean Distance.

3.2.1 UED: une distance euclidienne qui propage l'incertitude

Soient $S =< s_1, s_2, ..., s_l >$ et $R =< r_1, r_2, ..., r_l >$ deux sous séquences avec leurs incertitudes respectives $\delta S =< \delta s_1, \delta s_2, ..., \delta s_l >$ et $\delta R =< \delta r_1, \delta r_2, ..., \delta r_l >$. Si S et R étaient des sous séquences certaines, alors la distance euclidienne telle qu'utilisée dans la transformation shapelet est:

$$ED(S,R) = \frac{1}{l}\sum_{i=1}^{l}(s_i - r_i)^2$$

Afin de propager l'incertitude, il est nécessaire de connaître l'incertitude qu'on a en appliquant ED sur deux sous séquences incertaines. L'utilisation des techniques de propagation de l'incertitude sur ED permet d'obtenir UED dont la formule est la suivante:

$$UED(S \pm \delta S, R \pm \delta R) = (\frac{1}{l}\sum_{i=1}^{l}(s_i - r_i)^2) \pm (\frac{2}{l}\sum_{i=1}^{l}|s_i - r_i| \times (\delta s_i + \delta r_i))$$

$$UED(S \pm \delta S, R \pm \delta R) = ED(S,R) \pm (\frac{2}{l}\sum_{i=1}^{l}|s_i - r_i| \times (\delta s_i + \delta r_i))$$

Relation d'ordre entre deux mesures incertaines

Pour pouvoir utiliser UED dans la transformation shapelet, nous devons définir une relation d'ordre entre deux valeurs incertaines. Cette relation d'ordre est utilisée pour calculer la dissimilarité incertaine entre un shapelet et une série temporelle. Soient deux mesures incertaines $x = x_{best} \pm \delta x$ et $y = y_{best} \pm \delta y$, nous définissons les propriétés suivantes:

- $x = y$ si et seulement si $x_{best} = y_{best}$ et $\delta x = \delta y$
- $x < y$ si et seulement si l'une des deux conditions suivantes est satisfaites:
  - $x_{best} < y_{best}$
  - $x_{best} = y_{best}$ et $\frac{\delta x}{x_{best}} < \frac{\delta y}{y_{best}}$

Ces propriétés sont basées sur le fait que nous accordons tout d'abord une certaine confiance à la meilleure estimation d'une valeur incertaine; ensuite nous préférons les valeurs pour lesquelles l'incertitude est minimale.

Cette relation d'ordre est utilisée pour calculer la similarité entre une sous séquence incertaine $S \pm \delta S$ et une série temporelle incertaine $T \pm \delta T$ comme suit

$$UED(S \pm \delta S, T \pm \delta T) = \min\{UED(S \pm \delta S, R \pm \delta R)\ \forall R \pm \delta R \in T \pm \delta T\}$$

3.2.2 Transformation shapelet incertain

En remplaçant *ED* par *UED* dans l'algorithme de sélection des shapelets [Hills et al., 2014], on obtient notre algorithme de sélection des shapelets avec propagation de l'incertitude. Cet algorithme nous retourne des shapelets incertains que nous utilisons ensuite pour faire la transformation shapelet incertain.

Soient un jeu de données $D = \{T_1 \pm \delta T_1, T_2 \pm \delta T_2, ..., T_n \pm \delta T_n\}$ de séries temporelles incertaines et $S =< S_1 \pm \delta S_1, S_2 \pm \delta S_2, ..., S_k \pm \delta S_k >$ l'ensemble des k meilleurs shapelets incertains sélectionnés. Le jeu de données obtenu après transformation est $X = \{X_1, X_2, ..., X_n\}$, avec $X_i = [udist_1, udist_2, ..., udist_k]$ et $udist_j = UED(S_j, T_i)$.

La complexité en temps de la transformation shapelet incertain est $O(n^2 m^4)$; Ainsi donc, la propagation de l'incertitude n'augmente pas la complexité de la transformation shapelet.

3.2.3 Classification de données incertaines

Après la transformation avec propagation de l'incertitude, la dernière étape consiste à entraîner un modèle de classification sur le jeu de données obtenu. Le modèle utilisé doit être conscient de l'incertitude pour mieux faire la classification. Pour ce faire nous avons considéré trois façon de procéder. Nous les avons appelées classification FLAT, classification GAUSS et classification FLAT-GAUSS.

Classification FLAT

La classification FLAT est présenté par la figure 3. Après la transformation shapelet avec propagation de l'incertitude, chaque instance du jeu de données obtenu est représenté par un vecteur de taille $2 \times k$, k étant le nombre de shapelets. La première moitié du vecteur contient les meilleures estimations des distances aux différents shapelets; et la seconde moitié contient les incertitudes sur ces distances. Le modèle de classification **FLatTr_Classifier** prend donc en entrée ces vecteurs de tailles $2k$ et produit en sortie des prédictions. Chaque prédiction est un vecteur contentant la probabilités d'appartenance aux différentes classes.

Par exemple, pour l'instance i dont le vecteur de distance aux différents shapelets est X_i, la représentation FLAT sera donnée par $Flat(X_i)$ comme suit:

$$X_i = [udist_1, udist_2, ..., udist_k], avec\ udist_i = d_i \pm \delta d_i$$

$$Flat(X_i) = [d_1, d_2, ..., d_k, \delta d_1, \delta d_2, ..., \delta d_k]$$

Un problème avec cette façon de procéder est le risque d'avoir des vecteurs de trop grande taille.

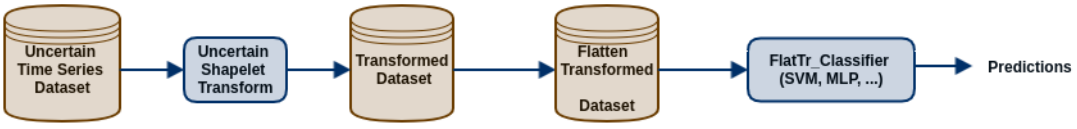


Figure 3 – Architecture de la classification FLAT

Classification GAUSS

L'idée dans cette seconde façon de faire la classification de données incertaines est d'attribuer un niveau de confiance à chaque meilleure estimation de valeur incertaine. Chacune des instances du jeux données obtenu après la transformation shapelet incertain est remplacée par un vecteur de confiance; la composante i de ce vecteur de confiance est la confiance qu'on accorde à la meilleure estimation de la distance entre l'instance et le shapelet i.

La confiance accordée à la meilleure estimation d'une distance incertaine est calculée un utilisant k lois normales $N_k(\mu_k, \sigma^2)$ de moyenne μ_k et d'écart-type σ.

$$\mu_k = \frac{max_k + min_k}{2}$$

$$\sigma = \frac{1}{\sqrt{2\pi}}$$

max_k est la valeur maximale possible de la distance au shapelet k et min_k la plus petite valeur possible de la distance au shapelet k. Ces deux valeurs sont calculées sur le jeux d'apprentissage uniquement.

La densité de probabilité de la loi N_k est la fonction $f_k(x)$

$$f_k(x) = \frac{1}{2\pi} e^{-\frac{(x-\mu_k)^2}{4\pi}}$$

La confiance accordée à la distance au shapelet k est la probabilité que $dist_k$ se réalise suivant la loi normale $N_k(\mu, \sigma_k^2)$, $dist_k$ étant la meilleure estimation de la distance au shapelet k. Les paramètres de la loi normale sont choisis de telle sorte que la confiance vaut 1 lorsque la distance est au centre de l'intervalle $[min_k, max_k]$; cette confiance diminue au fur et à mesure qu'on s'éloigne du centre, jusqu'à valoir 0 lorsqu'on sort de l'intervalle. La figure 4 illustre cette fonction de confiance.

Par exemple, pour l'instance i dont le vecteur de distances aux différents shapelets est X_i, la représentation GAUSS sera donnée par $Gauss(X_i)$ comme suit:

$$X_i = [udist_1, udist_2, ..., udist_k], avec\ udist_i = d_i \pm \delta d_i$$

$$Gauss(X_i) = [f_1(d_1), f_2(d_2), ..., f_k(d_k)]$$

Le principal inconvénient de la classification GAUSS est que l'incertitude n'est prise en

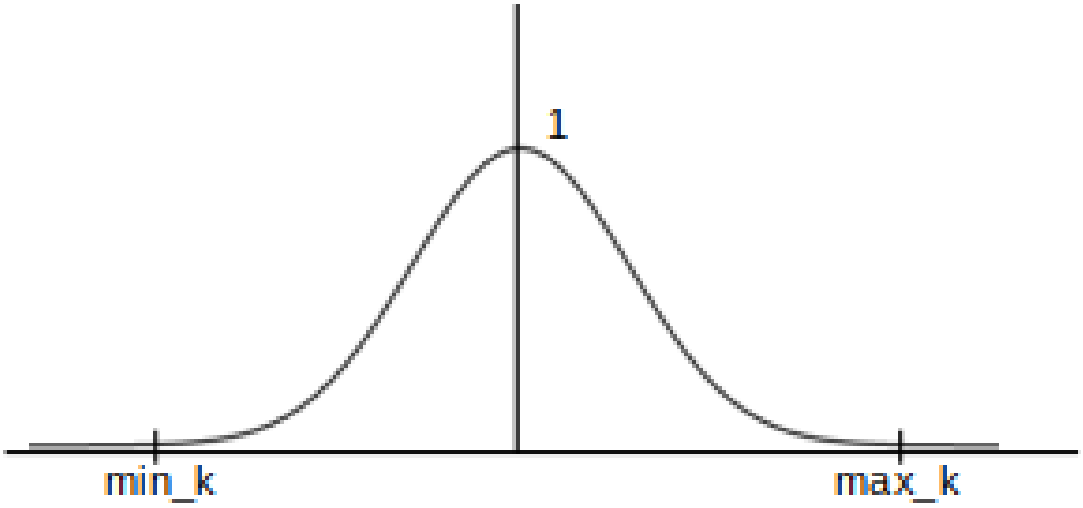


Figure 4 – Fonction de confiance

compte que pendant la phase d'apprentissage. En effet, la confiance associée à une distance incertaine est indépendante de l'incertitude sur cette valeur; elle dépend uniquement de la meilleure estimation et du shapelet. La figure 5 présente l'architecture de la classification GAUSS. Contrairement au modèle **FlatTr_Classifier** qui prend en entrée des vecteurs de taille deux fois le nombre de shapelets, le modèle **GaussTr_Classifier** prend des vecteurs dont la taille est exactement le nombre de shapelets. Pour chaque entrée, il produit un vecteur de probabilité d'appartenance aux différentes classes.

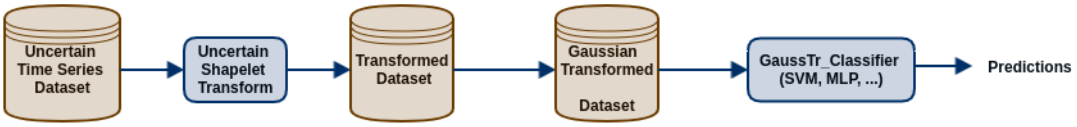


Figure 5 – Architecture de la classification GAUSS

Classification FLAT-GAUSS

Il s'agit ici d'une combinaison des deux premières approches. L'idée est d'avoir une approche qui s'adapte au jeu de données. Comme le montre la figure 6, la classe d'une instance est déterminée sur la base de la sortie de deux modèles: l'un utilisant FLAT et l'autre utilisant GAUSS. La manière de combiner ces deux modèles est déterminée automatiquement durant

la phase d'apprentissage.

Le jeux de données obtenu après la transformation shapelet incertain va subir deux transformations:

- La transformation FLAT telle que décrite dans la section 3.2.3. Cette opération va produire un nouveau jeu de données: **Flatten Transformed Dataset**. Ce jeu de données est ensuite utilisé comme entrée d'un modèle de classification supervisée (**FlatTr_Classifier**) qui va produire comme sortie un vecteur de probabilité qui représente l'appartenance aux différentes classes.

- La transformation GAUSS telle que présentée dans la section précédente (section 3.2.3. Le résultat de cette transformation est un nouveau jeu de données que nous avons appelé **Gaussian Transformed Dataset**. Un modèle de classification supervisée (**GaussTr_Classifier** va prendre ce nouveau jeu de données comme entrée et produire une distribution de probabilité d'appartenance aux différentes classes.

Les prédictions finales sont données par un troisième modèle de classification supervisée : **Joined_Classifier** . Ce dernier prend comme entrée la concaténation des sorties des modèles supervisés précédents (FlatTr_Classifier et GaussTr_Classifier) et produit en sorti les probabilités d'appartenance aux différentes classes. Avec cette architecture nous ne faisons aucune hypothèse sur la façon de combiner les deux façon de prendre en compte l'incertitude; La combinaison optimale sera déterminée automatiquement durant la phase d'apprentissage.

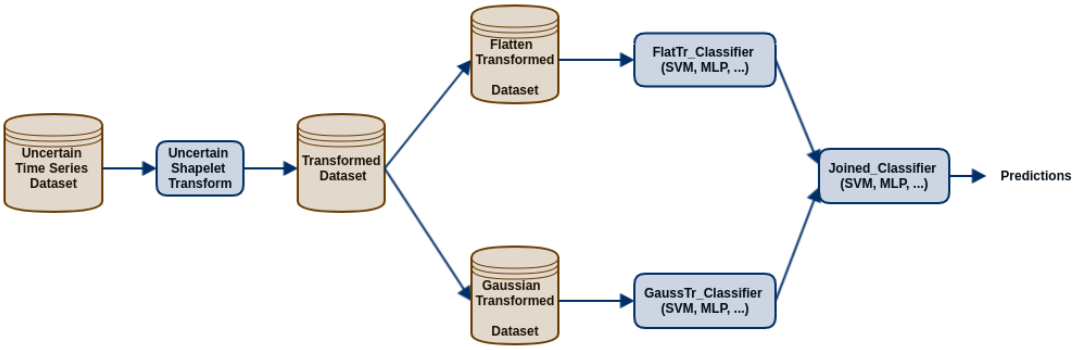


Figure 6 – Architecture de la classification Flat-Gauss

Dans ce chapitre, nous avons présentés 4 approches shapelets pour la classification des

séries temporelles incertaines. Dans la première approche, l'incertitude n'est pas prise en compte de façon explicite mais elle est absorbée. Les trois autres approches quant à elles prennent en compte l'incertitude de façon explicite en la propageant. Dans le prochain chapitre, nous présentons les expérimentations et les résultats obtenus sur différents jeux de données avec ces 4 approches.

4

Expérimentations

Dans ce chapitre, nous présentons les expérimentations et les résultats que nous avons obtenu en utilisant premièrement la prise en compte de l'incertitude par absorption (FOTS) et deuxièmement la prise en compte par propagation (UST). Durant toutes nos expérimentations, la métrique utilisée pour comparer les différents modèles est le taux de classification (accuracy); c'est-à-dire le rapport entre le nombre d'instances bien classées et le nombre total d'instances.

4.1 Jeux données

Nous avons utilisés les jeux de données de UEA/UCR[4]. Cependant ces jeux données sont supposés être sans aucune incertitude. Pour nos expérimentations, nous avons dû ajouter de l'incertitude dans les données.

Pour chaque jeu de données que nous avons utilisé nous avons généré une incertitude suivant une loi normale de moyenne 0 et d'écart type $c \times \sigma$, c étant un nombre réel positif et σ l'écart type du jeu de données. L'incertitude générée est ensuite additionnée au jeu de données afin d'obtenir un jeu de données incertain. Après l'ajout de l'incertitude, le jeu de données initial (qui est certain) n'est plus utilisé tout le long du processus d'apprentissage. La figure 7 illustre ce processus de bruitage des données.

Le figure 8 montre l'impact de l'ajout de l'incertitude sur une instance du jeu de données *Chinatown* [5]. Chaque série temporelle de ce jeu de données est exactement de longueur 24. Dans l'illustration, nous avons en abscisse l'instant l'instant d'observation et en ordonnée la valeur de l'observation. En bleu nous avons la série temporelle certaine, en orange la série temporelle obtenue avec ajout de l'incertitude pour $c = 1$ et en vert la série obtenue lorsque la constante c vaut 0.5. Plus la valeur de la constante c est élevée, plus l'incertitude est susceptible d'être élevée.

[4] http://www.timeseriesclassification.com/dataset.php
[5] http://www.timeseriesclassification.com/description.php?Dataset=Chinatown

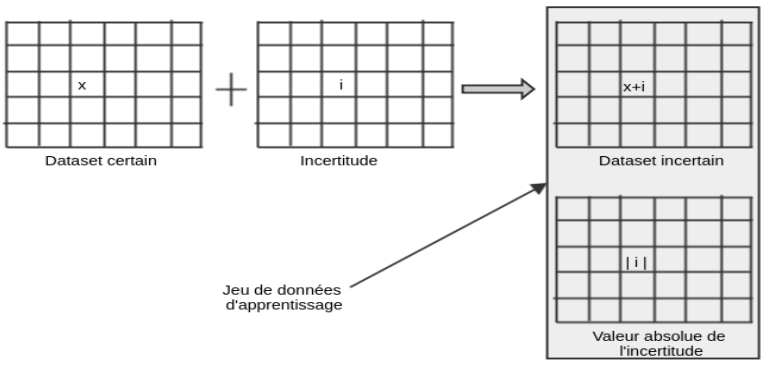


Figure 7 – Construction des jeux de données incertaines

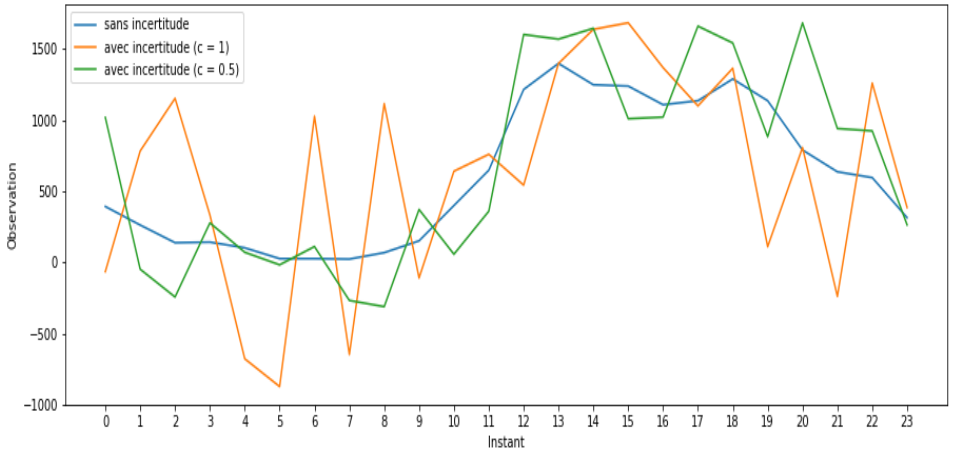


Figure 8 – Illustration de l'ajout de l'incertitude

Nous avons expérimenté sur un total de 29 jeux de données, chacun divisé en jeu d'apprentissage et en jeu de test. Ces jeux données sont présentés dans le tableau 2. La colonne *Nb d'instances* contient des valeurs sous la forme *a/b*, où *a* est le nombre d'instances d'apprentissage et *b* le nombre d'instances de test.

4.2 Code Source

Nous avons réutilisé le code open source mise à disposition par Bagnall et al. [2017] à l'adresse `https://github.com/uea-machine-learning/tsml`. Ce dépôt github contient une implémentation en langage Java de plusieurs algorithmes de classification des séries temporelles parmi lesquels l'algorithme de transformation shapelet.

Dataset	Nb d'instances	longueur des séries	Nb classes
Chinatown	20/345	25	2
SmoothSubspace	150/150	16	3
ECGFiveDays	23/861	137	2
SonyAIBORobotSurface	120/601	71	2
MoteStrain	20/1252	85	2
Fungi	18/186	202	18
DiatomSizeReduction	16/306	346	4
ArrowHead	36/175	252	3
TwoLeadECG	23/1139	83	2
BirdChicken	20/20	513	2
PowerCons	180/180	145	2
UMD	36/144	151	3
CBF	30/900	129	3
BME	30/150	129	3
GunPoint	50/150	151	2
MoteStrain	20/1252	85	2
ArrowHead	36/175	252	3
Plane	105/105	145	7
DistalPhalanxTW	400/139	81	6
GunPointOldVersusYoung	136/315	151	2
SyntheticControl	300/300	61	6
DistalPhalanxTW	400/139	81	6
MelbournePedestrian	1200/2450	25	10
MiddlePhalanxOutlineAgeGroup	400/154	81	3
MiddlePhalanxOutlineCorrect	600/291	81	2
MiddlePhalanxTW	399/154	81	6
PowerCons	180/180	145	2
ProximalPhalanxOutlineAgeGroup	400/205	81	3
ProximalPhalanxTW	400/205	81	6

Table 2 – Liste des jeux de données utilisés

Nous avons étendu le code en y implémentant FOTS en tant que distance entre sous séquences de telle sorte que l'algorithme de transformation shapelet puisse l'utiliser en lieu et place de la distance euclidienne.

Nous y avons également implémenté UED et UST en suivant le même principe d'implémentation que ST. Nous avons réutilisé autant que possible le code de ST et avons exposé la même interface.

Nous avons mis le code source étendu à la disposition du grand public à l'adresse https:

//github.com/frankl1/Uncertain-Shapelet-Transform.

En ce qui concerne l'ajout de l'incertitude dans les données, nous avons utilisé un notebook Jupyter. Le notebook est écrit en langage Python (Version 3) et est disponible à l'adresse https://github.com/frankl1/Uncertain-Shapelet-Transform/blob/master/add-noise.ipynb sur le même dépôt que UST.

4.3 Résultat des expérimentations avec FOTS

Nous avons effectué deux expérimentations avec FOTS, chacune avec un but bien précis. La première a pour objectif de voir si FOTS se comporte au moins aussi bien que que ED lorsque les séries temporelles sont certaines. La seconde expérimentation quant à elle porte sur des séries temporelles incertaines et a pour but de confirmer que FOTS absorbe bel et bien l'incertitude.

Nous avons deux implémentations de l'algorithme shapelet transform, l'une utilisant la distance euclidienne et l'autre utilisant FOTS. L'algorithme de classification utilisé dans les deux implémentations est l'algorithme Rotation Forest tel qu'implémenté dans le framework Weka[Frank et al., 2016].

Étant donné la complexité relativement élevée de la transformation shapelet avec FOTS, nous avons limité le nombre de shapelets candidats à explorer lors de la phase de recherche des k meilleurs shapelets. En effet l'algorithme va explorer les sous séquences de taille allant de 3 jusqu'à la taille de la série, avec un incrément de 10.

La figure 9 présente les taux de classification obtenus sur 25 jeux de données lorsqu'il n'y a pas d'incertitude. Sur l'ensemble des 25 jeux de données, le taux de classification moyen pour le modèle avec FOTS est de 64%. Ce taux est de 81% pour le modèle avec ED. Ainsi nous pouvons conclure que ED est de loin meilleur que FOTS lorsqu'il y a pas d'incertitude dans les données; cette conclusion se confirme par le diagramme de différence critique de la figure 10 qui ne présente aucune clique couvrant les deux modèles.

En présence d'incertitude, l'écart entre FOTS et ED diminue. Le taux de classification obtenu par les deux modèles sur les 25 jeux de données après ajout de l'incertitude est présenté par la figure 11. Bien que les deux modèles se dégradent en présence d'incertitude, le modèle avec FOTS se dégrade beaucoup moins. En effet le taux de classification moyen sur l'ensemble des jeux de données est de 45% pour le modèle avec FOTS et de 57% pour celui avec ED. Cependant le modèle avec ED reste de loin le meilleur des deux (voir figure 12). Cependant cette conclusion pourrait être différente si tous les shapelets candidats sont pris en compte.

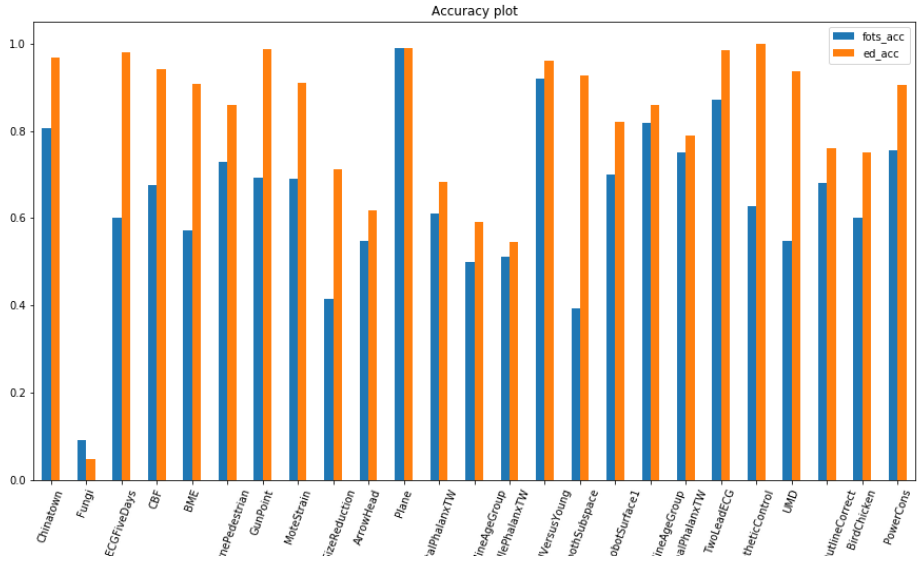


Figure 9 – Taux de classification entre FOTS et ED en absence d'incertitude

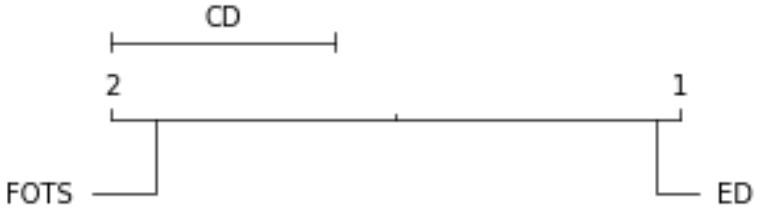


Figure 10 – Diagramme de différence critique entre FOTS et ED en absence d'incertitude

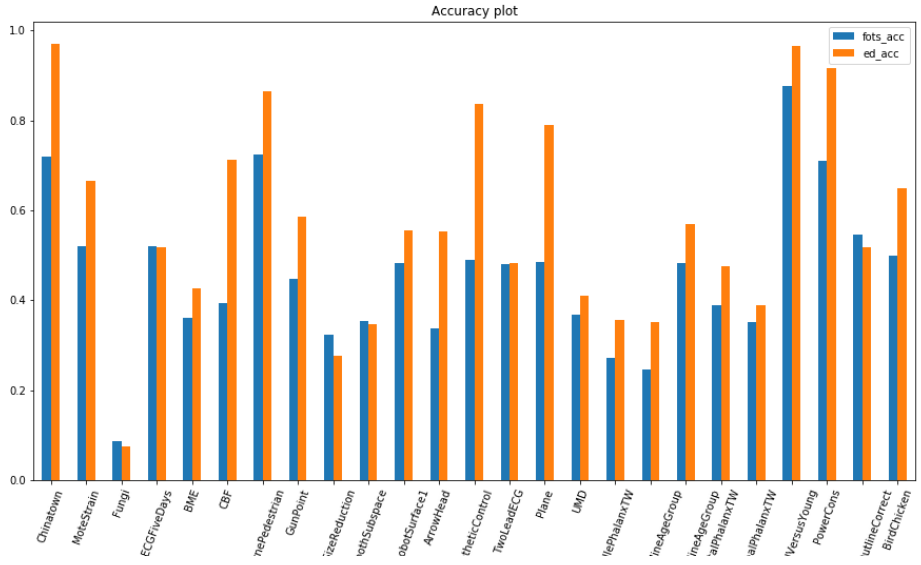


Figure 11 – Taux de classification entre FOTS et ED en présence d'incertitude

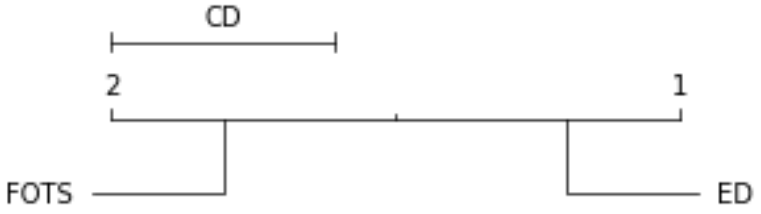


Figure 12 – Diagramme de différence critique entre FOTS et ED en présence d'incertitude

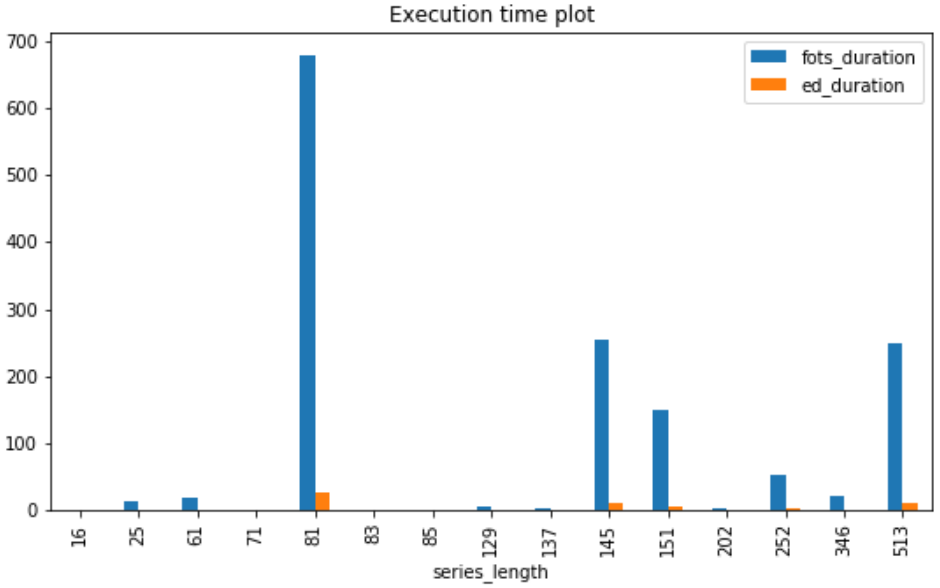


Figure 13 – Temps d'exécution entre FOTS et ED en fonction de la taille des séries

Le modèle avec FOTS semble bien adapté pour les jeux de données qui ont des tendances apériodiques (tels que Fungi, DiatomSizeReduction et InsectEPGSmallTrain) ou signisoidales (tel que ECGFiveDays).

Durant les expérimentations, nous avons mesuré le temps d'exécution des deux approches. La figure 13 montre en bleu les durées d'exécution de la transformation shapelet avec FOTS et en orange les durées d'exécution de la transformation shapelet avec ED. En abscisses, nous avons la longueur des séries et en ordonnées la durée d'exécution en heures. Comme on s'y attendait, le temps d'exécution avec FOTS est très élevés.

4.4 Résultat des expérimentations avec UST

Les objectifs de ces expérimentations sont au nombre de deux: premièrement il est question d'évaluer UST, et deuxièmement nous souhaitons confirmer que la prise en compte de l'incertitude lorsqu'elle est disponible permet de faire une meilleure classification des séries temporelles avec la méthode des shapelets.

Afin d'éviter que l'algorithme de classification supervisée ne biaise nos conclusions, nous avons fait plusieurs expérimentations en utilisant différents algorithmes. En effet nous avons utilises les arbres de décision (DT), le perceptron multicouche (MLP), les forêts de rota-

tion (RotF), les machines à vecteur de support (SVM) et les forêts aléatoires (RandF). Nous avons utilisé ces modèles tels qu'il sont été implémentés dans le framework Weka[Frank et al., 2016]. Aussi nous n'avons pas tuné les paramètres de ces modèles, nous nous sommes contentés d'utiliser les paramètres par défaut. En ce qui concerne les arbres de décision, nous avons utilisé l'algorithme *J48* qui est une implémentation open source de l'algorithme *C4.5*[Sharma and Kumar, 2016]. SVM est implémenté dans Weka sous l'appellation SMO.

Dans le modèle FLAT-GAUSS, les trois modèles des classifications utilisent à chaque fois le même algorithme. Ansi UST-FLAT-GAUSS(MLP) est un modèle de transormation shapelet incertain qui combine la classification FLAT et la classification GAUSS, en utilisant trois perceptrons multicouches au niveau des trois blocs *Classifier 1*, *Classifier 2* et *Classifier 3* de la figure 6; de même ST(SVM) est une implémentation de ST qui utilise SVM pour la classification après la transformation shapelet. Au total nous avons comparé 20 modèles dont 5 sont des algorithmes ST et les 15 autres sont des algorithmes UST.

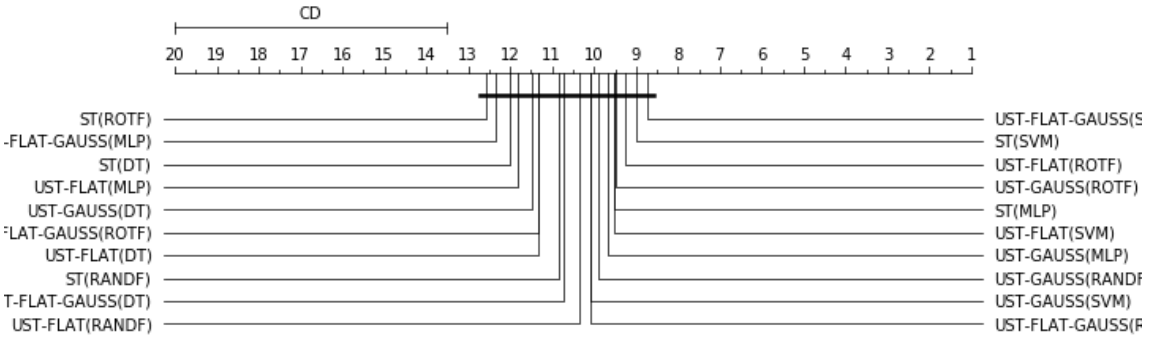
Contrairement aux expérimentations avec FOTS où nous avons limité l'espace des shapelets candidats, ici nous n'avons pas limité. Durant toute l'expérimentation avec UST, les algorithmes évaluent les sous séquences de taille allant de 3 à la taille des séries avec un incrément de 1.

Nous avons considéré deux cas selon l'intervalle dans lequel l'incertitude est prise. Dans le **Cas** 1σ, où la constance c vaut 1, l'incertitude est prise dans l'intervalle $[-\sigma; \sigma]$. Dans le **Cas** 2σ nous augmentons l'espace de l'incertitude en mettant la constante c à $\frac{3}{2}$; l'incertitude est donc prise dans l'intervalle $[-\frac{3}{2}\sigma; \frac{3}{2}\sigma]$.

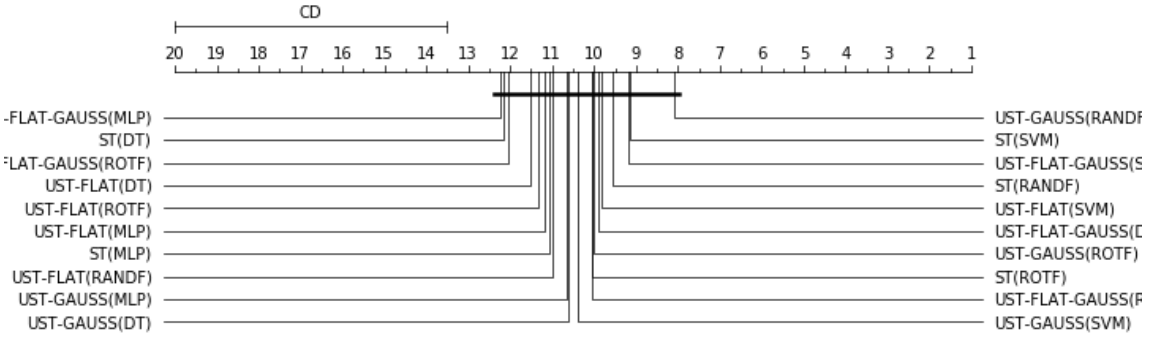
Nous avons expérimenté UST sur les 21 premiers jeux de données du tableau 2. Commençons par une analyse globale des résultats en utilisant la figure 14. La première remarque flagrante est qu'il y a une clique qui recouvre tous les modèles et ceci quelque soit l'espace de l'incertitude. Cela signifie que tous ces 20 modèles s'équivalent sur l'ensemble des jeux de données utilisés. Cependant nous pensons que cette équivalence apparente est due au fait que les 20 modèles sont comparés sur uniquement 21 jeux de données. La seconde remarque est que le modèle le plus performant est un modèle UST: UST-FLAT-GAUS(SVM) l'emporte quand $c = 1$ et UST-GAUSS(RANDF) l'emporte lorsque $c = \frac{3}{2}$.

Nous allons à présent faire un zoom sur la figure 14. Nous allons faire une analyse en considérant à tour de rôle les différents algorithmes de classification supervisée utilisés.

Si nous considérons les modèles utilisant les machines à vecteur de support, on obtient les diagrammes de différence critique de la figure 15. Nous avons toujours une clique qui couvre les modèles; de plus l'augmentation de l'espace d'incertitude ne change pas le classement.



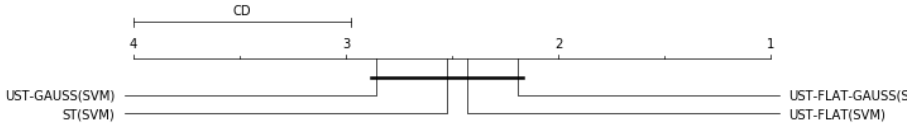
(a) Diagramme de différence avec $c = 1$



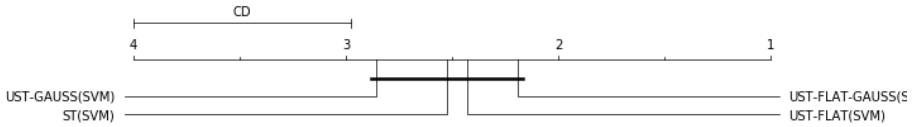
(b) Diagramme de différence critique avec $c = \frac{3}{2}$

Figure 14 – Diagramme de différence critique des modèles UST et ST.

Nous pouvons tout de même noter que les deux meilleurs modèles sont des modèles UST.



(a) Avec $c = 1$

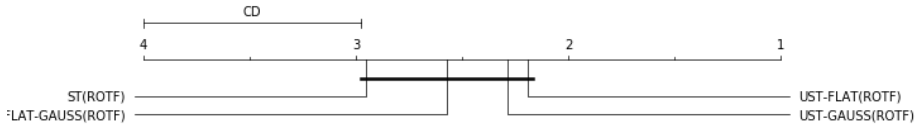


(b) Avec $c = \frac{3}{2}$

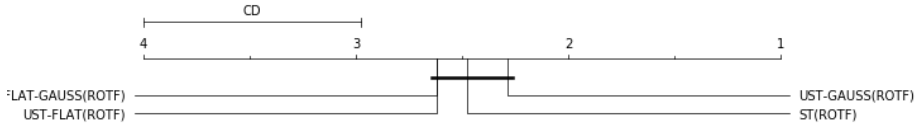
Figure 15 – Diagramme de différence critique en utilisant SVM.

En se focalisant uniquement sur les modèles utilisant l'algorithme des forêts de rotation on obtient la figure 16. Nous n'avons toujours pas de différence significative entre les modèles.

On remarque ici que le modèle ST est le moins bon lorsque $c = 1$ et curieusement il passe deuxième lorsque $c = \frac{3}{2}$.



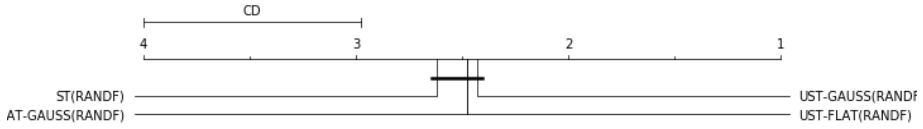
(a) Avec $c = 1$



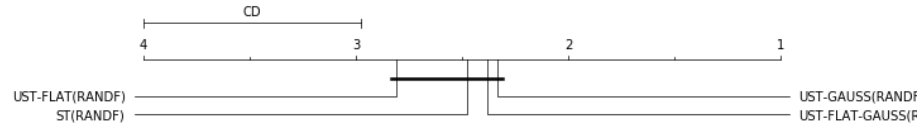
(b) Avec $c = \frac{3}{2}$

Figure 16 – Diagramme de différence critique en utilisant RotF.

Regardons maintenant les modèles utilisant les forêts aléatoires (figure 17). Nous avons toujours la clique d'équivalence, et on observe une dégradation de UST-FLAT(RandF) et de ST(RandF) lorsque l'espace d'incertitude augmente (c passe de 1 à $\frac{3}{2}$). UST-GAUSS(RandF) est assez stable et garde sa place de meilleur modèle.



(a) Avec $c = 1$

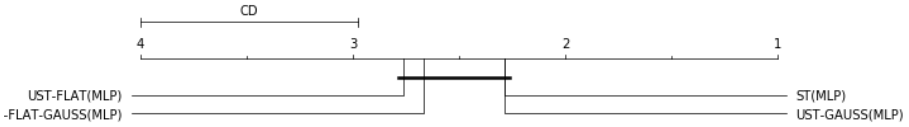


(b) Avec $c = \frac{3}{2}$

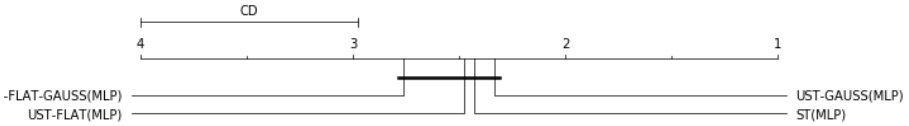
Figure 17 – Diagramme de différence critique en utilisant RandF.

Avec le perceptron multicouches, ST(MLP) se dégrade au profit de UST-GAUSS(MLP) qui reste stable lorsque l'incertitude augmente (voir figure 18).

Pour finir concentrons nous sur les modèles à base d'arbre de décision uniquement. La figure 19 montre cette fois une différence significative entre les modèles. Cette différence est d'autant plus grande lorsque l'espace d'incertitude augmente. Le modèle ST(DT) est le moins bon, et ses performances se sont terriblement dégradées lorsque l'espace d'incertitude



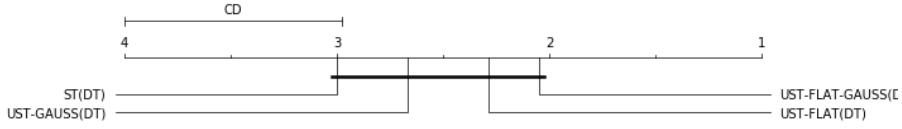
(a) Avec $c = 1$



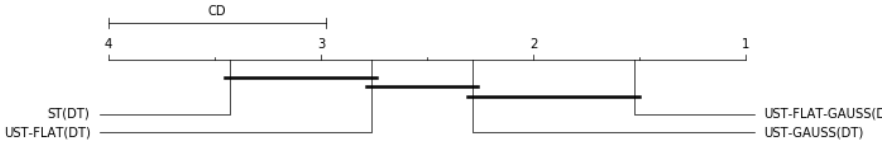
(b) Avec $c = \frac{3}{2}$

Figure 18 – Diagramme de différence critique en utilisant MLP.

a augmenté. UST-FLAT(DT) s'est aussi dégradé au profit de UST-GAUSS(DT) qui passe deuxième. UST-FLAT-GAUSS(DT) quant à lui a très bien réagi fasse à l'augmentation de l'incertitude.



(a) Avec $c = 1$



(b) Avec $c = \frac{3}{2}$

Figure 19 – Diagramme de différence critique en utilisant un arbres de décision.

Quelque soit l'angle sur lequel nous regardons nos résultats, c'est toujours un modèle UST qui est en tête; Nous pouvons conclure cette expérimentation en disant que lorsque l'incertitude sur les séries temporelles est connue, il est plus judicieux d'utiliser la transformation shapelet incertain.

4.5 Limites de notre approche

La première limite de ce travail est que nous n'avons pas utilisé des jeu de données réelles pour évaluer notre approche. Nous avons pris des jeu de données réelles et nous y avons ajouté une

incertitude qui ne représente probablement aucune réalité. Nous avons procédé ainsi parce que nous n'avons trouvé aucun jeu de données sur les séries temporelles avec des incertitudes connues. Nous aurons pu utiliser le jeu de données du Challenge Plasticc[The PLAsTiCC team et al., 2018], mais il contient beaucoup de données manquantes et leur traitement est un véritable challenge.

La seconde limite que nous avons identifié est que nous faisons la classification en prenant en compte l'incertitude de façon Ad hoc. En effet aucun des modèles de classification supervisée que nous avons utilisé n'est conscient de l'incertitude dans les données. Nous pensons que si on utilise un modèle conscient de l'incertitude sur les données qu'il prend en entrée, on obtiendrait très probablement de bien meilleurs résultats.

5

Conclusion et Perspectives

Il était question durant ce stage de 6 mois de traiter la classification des séries temporelles en présence d'incertitude. Pour cela il fallait utiliser la transformation shapelet, un algorithme qui continu de faire ses preuves pour la classification des séries temporelles certaines. Après une phase de compréhension du problème et d'exploration de la littérature, nous n'avons trouvé aucun algorithme permettant d'accomplir cette tâche; cependant nous avons trouvé des éléments pouvant contribuer à l'atteinte de cet objectif.

Premièrement, nous avons modifié l'algorithme de transformation shapelet en remplaçant la distance euclidienne par FOTS, une mesure de dissimilarité qui absorbe l'incertitude. Les résultats obtenus avec cette approche n'ont pas été satisfaisants. Par ailleurs la complexité en temps de cette approche est élevée, soit $O(n^2m^5)$.

Deuxièmement nous avons décidé de conserver la distance euclidienne, mais en propageant l'incertitude. Nous avons appelé cette approche UST pour Uncertain Shapelet Transform. Les résultats des expérimentations ont été très encourageants et valident la pertinence de UST en tant que modèle de classification des séries temporelles incertaines.

Nous sommes conscient des limites de notre travail. En effet, la prise en compte de l'incertitude dans UST est faite de façon ad hoc; un des futurs travaux consistera à utiliser un modèle de classification supervisée qui soit conscient de l'incertitude sur les données.

Les performances de UST ont été évaluées sur seulement 21 jeux de données de UCR/UEA, pourtant il y en a 128, tous aussi variés les uns que les autres. Afin d'identifier les jeux de données sur lesquels UST est approprié ou pas du tout approprié, nous comptons faire des expérimentations sur les jeux de données restants.

Une autre perspective, toute aussi importante que les précédentes est d'utiliser UST sur un jeu de données réelles. En effet nous avons ajouté nous même l'incertitude dans les jeux de données, et cette incertitude ne représente probablement pas une réalité.

References

A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote: The collective of transformation-based ensembles. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 1548–1549, May 2016. doi: 10.1109/ICDE.2016.7498418.

A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

Z. Bar-Joseph, A. Gitter, and I. Simon. Studying and modeling dynamic biological processes using time-series gene expression data. *Nature reviews. Genetics*, 13:552–64, 07 2012. doi: 10.1038/nrg3244.

G. Batista, E. J. Keogh, O. Moses Tataw, and V. Alves de Souza. Cid: An efficient complexity-invariant distance for time series. *Data Mining and Knowledge Discovery*, 28, 04 2013. doi: 10.1007/s10618-013-0312-3.

M. Dallachiesa, B. Nushi, K. Mirylenka, and T. Palpanas. Uncertain Time-Series Similarity: Return to the Basics. *arXiv e-prints*, art. arXiv:1208.1931, Aug 2012.

E. Frank, M. A. Hall, and I. H. Witten. *The WEKA workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 392–401. ACM, 2014.

J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.

Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu. Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240, 2011.

L. Jiao. *Classification of uncertain data in the framework of belief functions : nearest-neighbor-based and rule-based approaches*. PhD thesis, 10 2015.

J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, Oct 2007. ISSN 1573-756X. doi: 10.1007/s10618-007-0064-z. URL https://doi.org/10.1007/s10618-007-0064-z.

J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, May 2015. ISSN 1384-5810. doi: 10.1007/s10618-014-0361-2. URL http://dx.doi.org/10.1007/s10618-014-0361-2.

J. Lines, S. Taylor, and A. Bagnall. Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Trans. Knowl. Discov. Data*, 12 (5):52:1–52:35, July 2018. ISSN 1556-4681. doi: 10.1145/3182382. URL http://doi.acm.org/10.1145/3182382.

T. Rakthanmanon and E. Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*, pages 668–676. SIAM, 2013.

H. Sharma and S. Kumar. A survey on decision tree algorithms of classification in data mining. *International Journal of Science and Research (IJSR)*, 5, 04 2016.

V. S. Siyou Fotso. *Extraction des connaissances dans les séries temporelles incertaines et cyclique: Application à l'analyse de la locomotion en chaise roulante*. PhD thesis, University Clermont Auvergne, 2018.

V. S. Siyou Fotso, E. Mephu Nguifo, and P. Vaslin. Frobenius correlation based u-shapelets discovery for time series clustering. Apr. 2018. URL https://hal.archives-ouvertes.fr/hal-01771003. working paper or preprint.

J. R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, 2 sub edition, 1996. ISBN 093570275X. URL http://www.amazon.com/Introduction-Error-Analysis-Uncertainties-Measurements/dp/093570275X%3FSubscriptionId%3D13CT5CVB80YFWJEPWS02%26tag%3Dws%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D093570275X.

The PLAsTiCC team, J. Allam, Tarek, A. Bahmanyar, R. Biswas, M. Dai, L. Galbany, R. Hložek, E. E. O. Ishida, S. W. Jha, and D. O. Jones. The Photometric LSST Astronomical Time-series Classification Challenge (PLAsTiCC): Data set. *arXiv e-prints*, art. arXiv:1810.00001, Sep 2018.

R. Tsay. *Analysis of financial time series*. Wiley series in probability and statistics. Wiley-Interscience, Hoboken, NJ, 2. ed. edition, 2005. ISBN 978-0-471-69074-0.

L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM, 2009.