

# Classification des séries temporelles multivariées par l’usage de Mgrams

Angéline PLAUD <sup>\*1</sup>, Engelbert Mephu Nguifo<sup>2</sup>, et Jacques Charreyron<sup>3</sup>

<sup>1</sup>Université Clermont Auvergne, LIMOS, CNRS et Michelin

<sup>2</sup>Université Clermont Auvergne, LIMOS, CNRS

<sup>3</sup>Michelin, Clermont-Ferrand, France

21 juin 2019

## Résumé

La classification des séries temporelles univariées (STU) a déjà été explorée dans de multiples publications, où des modèles, nombreux et efficaces, ont été proposés. De tels modèles sont souvent inadéquats lorsqu’ils sont appliqués aux séries temporelles multivariées (STM). Ce type de données a émergé avec la multiplication de capteurs capables d’enregistrer de grandes quantités de données simultanément. Ces données sont caractérisées par de nombreuses dimensions, des longueurs variables, du bruit et des corrélations entre les dimensions. La classification des STM est un problème complexe et peu de travaux existent actuellement. Dans ce papier, nous proposons un nouveau modèle qui combine l’utilisation de M-histogrammes et l’apprentissage multi-vues via un ensemble de classifieurs. Le M-histogramme est un outil statistique, utilisé pour la visualisation de données, et qui peut révéler l’information utile contenue à travers les dimensions. Il peut aussi permettre l’encodage des STM. Quant à l’apprentissage multi-vues, cela correspond à l’intégration de données comme ensemble d’attributs, ce qui est le cas des STM. Ce concept peut également apporter des informations complémentaires sur les données. Notre méthode combine ces deux concepts et permet un encodage des STM qui peut être meilleur que d’autres tels que Symbolic Aggregate approximation (SAX), au regard des expérimentations que nous avons effectuées. Nous avons comparé notre méthode avec celles de l’état de l’art sur des données de référence, et avons discuté des propriétés de notre méthode.

**Mots-clef** : Série temporelle multivariée, Mgram, Classifieur ensembliste, Apprentissage multi-vues.

## 1 Introduction

Une série temporelle est un ensemble de valeurs indexées par rapport au temps, représentant les mesures d’un phénomène. Lorsqu’un seul capteur est utilisé pour enregistrer ces valeurs, nous obtenons une série temporelle univariée (STU). Quand de multiples capteurs enregistrent à chaque instant des mesures, nous obtenons une série temporelle multivariée (STM). Différents domaines de recherche tels que la médecine [ODF<sup>+</sup>08], l’habitat intelligent [ZH14], l’industrie automobile [LMSZ98] se sont intéressés aux STM. L’objectif est généralement la classification de ces séries. Contrairement aux STU, les STM sont caractérisées par des interactions dans le temps mais aussi entre les dimensions. De ce fait, les méthodes de classification des STU ne sont pas applicables aux STM. WEASEL MUSE [SL17] et SMTS [BR15] font partie des modèles récents dédiés aux STM. Le premier est basé sur la transformée de Fourier, la représentation symbolique et l’extraction de fenêtres de données. Le second est basé sur les forêts aléatoires permettant la transformation d’une STM en une représentation symbolique sous forme de chaîne de caractères. Ces méthodes peuvent souffrir de problèmes de scalabilité (complexité) pour les longues séries temporelles. De ce fait, la classification des STM est toujours un problème ouvert.

Dans ce papier, nous proposons un nouveau classifieur pour STM qui combine le principe d’apprentissage multi-vues [ZXXS17] avec un outil statistique appelé le M-histogramme. Celui-ci, aussi connu sous

---

\*Avec le soutien de Michelin et de l’ANRT

le nom de Mgram, est un outil permettant la visualisation de la fonction de distribution inhérente aux données. Il permet la projection d'une STM dans un tableau de  $M$  dimensions. Dans notre modèle, nous l'utilisons pour réduire nos données et extraire l'information mutuelle contenue à travers plusieurs dimensions. De plus, comme chaque dimension d'une STM correspond à une vue des données, une STM peut être considérée comme une vue multiple d'un même phénomène temporel. Cette idée, de ce que nous en savons, n'a jamais été abordée dans la littérature. Cette combinaison nous donne un classifieur ensembliste qui explore les différentes vues d'une STM. Nous avons montré expérimentalement que ce modèle est efficace comparé à plusieurs autres modèles, et qu'il est robuste sur de multiples jeux de données. De plus, le Mgram fournit une nouvelle représentation des données très compétitive pour les STM par rapport à Symbolic Aggregate approXimation (SAX) [LKWL07].

Nous avons organisé le papier tel que la Section 2, est un rappel des définitions. La Section 3 est un état de l'art. La Section 4 décrit l'approche. La Section 5 présente les résultats du modèle. Finalement, la Section 6 est une conclusion.

## 2 Définitions

### 2.1 Série temporelles multivariées

Une série temporelle est un ensemble de valeurs indexées par rapport au temps, qui représente l'évolution d'un phénomène. La différence entre une série temporelle univariée (STU) et multivariée (STM) est le nombre de valeurs à chaque instant du temps (Fig.1a). Ces valeurs multiples interagissent entre elles et donc les méthodes de classification d'STU, comme reportées dans la littérature, sont inadéquates pour la classification des STM [SYWK].

Une STM  $X^n$  possède  $M$  attributs à chaque instant  $t$  durant  $T$  observations, où  $T \in [1, \tau]$  et  $X^n$  est la  $n$ -th STM, où  $n = [1, \dots, N]$ , tel que :

$$X_m^n = [x_m^n(1), \dots, x_m^n(T)] \quad (1)$$

Nous pouvons considérer chaque dimension  $m$  comme une série temporelle univariée, et donc une STM est un ensemble de STU *liées par des interactions*. Par ailleurs, en plus des dimensions d'une STM, nous devons aussi considérer leurs longueurs. Il y a trois cas possibles :

1. *STM avec la même longueur* pour toutes les dimensions et pour toutes les STM du jeu de

données. C'est le cas le plus courant de la littérature.

2. *STM avec la même longueur pour chaque STU et une longueur variable pour chaque STM*. C'est le cas lorsque le même phénomène est enregistré via des mesures de même fréquence.
3. *STU et STM avec des longueurs variables*. C'est le cas lorsque le même phénomène est enregistré via des mesures de fréquences différentes et donc il y a plus ou moins de points par dimension.

Notre méthode peut gérer le premier et le deuxième cas. Le troisième cas n'est pas pris en considération dans ce travail.

### 2.2 Mgrams

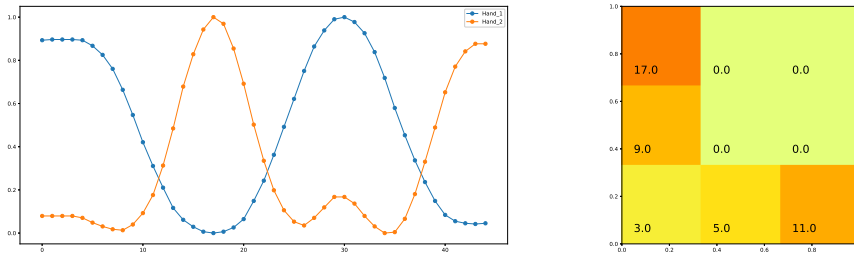
Un M-histogramme, aussi connu sous le nom de Mgram, est un outil permettant de visualiser de la fonction de densité d'un échantillon de données. Cet outil permet la projection d'une STM en un tableau de  $M$  dimensions. Les paramètres à régler pour le Mgram sont le domaine des valeurs  $[min, max]$  et le nombre d'intervalles appelés *bins*. De plus, nous avons aussi le paramètre global  $M$  du Mgram. Quand  $M = 1$  nous avons un 1-gram, plus connu sous le nom d'histogramme. Quand  $M = 2$ , nous avons un bigram. Comme le Mgram est un hypercube qui peut être difficile à représenter, la Fig.1b montre un exemple de bigram.

Un bigram indique le nombre de tuples  $[x_1^n(i), x_2^n(i)]$  qui sont contenus dans chaque intervalle  $[b_1, b_2]$ . La fréquence des tuples qui tombent dans chaque bin, permet la construction d'un tableau composé de  $b_1 \times b_2$  éléments. Plus grande est la valeur d'un élément du tableau, plus grande est la fréquence de tuples contenus.

Donc la complexité d'un bigram est égale au nombre de tuples à traiter par le nombre de bins, tel que  $O(T \times (B_1 + B_2))$ , où  $B_1$  et  $B_2$  sont les nombres maximum d'intervalles sur chaque axe. Pour l'histogramme, nous avons une complexité de  $O(TB)$ , où  $B$  est le plus grand nombre d'intervalles.

Dans notre modèle, nous utilisons le bigram pour réduire les STM et extraire l'information mutuelle contenue dans les dimensions. L'histogramme réduit  $X^n$  à un vecteur de taille  $b$ , le nombre d'intervalles, et le bigram réduit la STM à une matrice de taille  $b_1 \times b_2$ . Parce que le Mgram est un outil statistique permettant d'appréhender la fonction de densité inhérente aux données, nous avons besoin de beaucoup de points pour faire une bonne estimation. Plus nous avons de points par intervalles, plus notre Mgram est précis.

Pour éviter la perte de l'ordre des points de la STM avec cette représentation nous incluons aussi la somme



(a) Exemple d'une STM avec 2 dimensions issue du jeux de données Libras [BR15]. (b) Transformation de la STM  $a$  en un bigram avec 3 intervalles sur chaque axe.

FIGURE 1 – Exemple de transformation d'une STM en un bigram.

cumulée et la dérivée à notre modèle, en plus de la série originelle. De ce fait, le Mgram rapporte les fréquences des données de base mais aussi des dérivées et des sommes cumulées.

### 3 Etat de l'art

Il existe une pléthore de méthodes de classification de STU. Parmi elles, DTW-1NN [SC78], [KP01], SAX [LKWL07], [YAMP17], [MGQT13] and COTE [BLHB15] ont prouvé leur efficacité.

DTW-1NN est une méthode de classification basée sur une distance innovante DTW (Dynamic Time Warping) créée pour la reconnaissance de la parole [SC78]. Cette distance adaptée à la classification de STU, permet de calculer la similarité entre deux séries en prenant en compte les délais temporels. À l'heure actuelle, cette technique est l'une des plus connues et des plus efficaces, mais un temps de calcul élevé même si des versions plus rapides ont été proposées dans la littérature [DSP<sup>+</sup>18],[PFW<sup>+</sup>16].

COTE est une méthode de classification ensembliste [Bre96], permettant à des classifieurs basiques et rapides d'être combinés. Elle utilise un système de votes sur un ensemble de distances, comme DTW, et sur un ensemble de transformations, comme shapelet [YK09]. Un shapelet étant une partie extraite d'une STU. En combinant tous les meilleurs aspects de la classification d'STU, COTE donne d'excellents résultats [BLB<sup>+</sup>17].

SAX est une méthode de transformation, permettant de passer d'une STU à une chaîne de caractères. Par la même occasion, cela permet de réduire la série en gardant l'ordre des points. Cette méthode compare les nouvelles représentations grâce à une distance qui borne la distance euclidienne entre les deux séries originelles [LKWL07].

Les approches de classification de STU ne sont pas adéquates pour les STM et [SYWK] précise même que la généralisation de techniques de classification d'STU aux STM n'est pas triviale. Ceci est principalement dû à la difficulté de prendre en compte les dépendances entre les dimensions d'une STM. Cependant, il existe quand même quelques adaptations. Nous avons présenté DTW-1NN et SAX car elles font partie de ces méthodes. Par exemple, SMTS [BR15] est basé sur SAX et est une forêt aléatoire qui transforme une STM en un sac de mots. WEASEL\_MUSE [SL17] transforme une STM en un histogramme/bigram comptant la fréquence de mots Symbolic Fourier Approximation (SFA) extraits tout le long de la série. Ces méthodes donnent de bons résultats, cependant leurs complexités font qu'elles ne s'appliquent pas bien aux longues séries temporelles. En effet, les forêts aléatoires ne sont pas le meilleur outil pour les gros volumes de données, de même que les techniques d'extraction de fenêtres de données.

D'autres techniques de classification de STM sont basées sur la réduction de dimensionnalité comme PCA-Eros [YS04] ou MTSC [ZC15]. En particulier, PCA [JC16](Principal Component Analysis) est la technique de réduction la plus connue. Cependant, toutes ces techniques sont très coûteuses en temps. Enfin, en 2005, une autre approche utilise un ensemble d'attributs calculés à partir de la STM [KS05], comme la durée, la longueur de la série, etc.

Les meilleures méthodes, en prenant en compte l'exactitude des résultats sont WEASEL\_MUSE, SMTS et différentes versions de DTW1NN, appelées DTW1NND et DTW1NNI présentées dans [SYWK].

Concernant le M-histogramme, il a été utilisé pour améliorer le contraste d'image [SKS<sup>+</sup>15]. Cette publication décrit les différents usages de l'outil pour

déterminer le seuil permettant l'amélioration du contraste d'images. Le Mgram a aussi été utilisé pour le contraste de texte [FWL02]. Finalement, depuis 1998 Michelin utilise les Mgrams pour analyser l'usage de ses pneus [LMSZ98]. Nous avons donc choisi d'appliquer les Mgrams dans le contexte de l'apprentissage multi-vue [YWW17], [ZXXS17], comme cela permet d'extraire différentes vues des données. Le multi-vue s'applique bien aux STM comme la combinaison de Mgrams permet d'avoir différentes vues de même donnée.

## 4 Ensemble de Mgrams

La base de nos travaux sur les STM est la notion de Mgrams. Pour simplifier l'implémentation, l'explication et l'illustration, nous nous concentrerons seulement sur les bigrams et les histogrammes dans l'application de notre modèle. L'Algorithme 1 le décrit. Le modèle est un ensemble de bigrams et d'histogrammes représentant les vues multiples de même données. Nous extrayons des dimensions au sein de la STM, créons un bigram ou un histogramme via un apprentissage, puis finalement, nous réalisons un système de vote afin d'établir une prédiction finale.

### 4.1 Construction d'un bigram

Comme un bigram ne se construit qu'à partir de 2 dimensions et un histogramme avec une seule, nous devons extraire ces dimensions au sein d'une STM, s'il y en a plus. Comme nous ne pouvons pas choisir, a priori, les dimensions qui donneront la meilleure classification, nous les choisissons de manière aléatoire (Alg 1 row 3.a.).

Pour la construction d'un bigram, comme montré dans la Section 2, nous avons deux paramètres à régler qui sont les bins (intervalles)  $b_1$  et  $b_2$  pour chaque dimension. La complexité d'apprentissage est donc définie par le nombre de combinaisons possibles de  $(b_1, b_2)$  avec répétitions car  $bigram(b_1 = i, b_2 = j) \neq bigram(b_1 = j, b_2 = i)$ . Soit  $O(B_1 B_2)$ , où  $B_1$  et  $B_2$  sont les nombres maximums de bins tels que  $b_1 \in B_1, b_2 \in B_2$ .

Ce temps d'apprentissage peut être réduit en prenant  $b_1 = b_2$ . Dans ce cas la complexité devient  $O(2 \times B_1)$ . Nous obtenons alors un bigram carré. Concernant la taille des données à stocker, la transformation permet de réduire les STM à  $b_1 \times b_2$  éléments pour les bigrams et à  $b_1 + b_2$  éléments pour les histogrammes, comme nous en calculons deux. Ces transformations sont paramétrées sur le jeu de données d'entraînement

et appliquées au test. Puis elles sont stockées dans deux listes, qui sont ensuite utilisées afin de réaliser un système de vote de classifieurs, comme montré dans la Fig.2.

### 4.2 Complément d'informations

La capacité des bigrams à réduire les données est un avantage mais aussi sa principale faiblesse. Quand nous compressons les données via le bigram, nous perdons beaucoup d'informations telles que l'ordre des points. Afin de contrer ces effets, nous ajoutons deux nouvelles composantes à notre modèle qui sont, les dérivées et les sommes cumulées des données originelles.

#### 4.2.1 Dérivée

Il est facile de trouver des cas où deux STM donnent le même bigram. Par exemple, en changeant l'ordre des points d'une STM, nous en obtenons une nouvelle, où les tendances sont différentes, mais qui donnent les mêmes bigrams. Si nous considérons que ces deux STM appartiennent à des classes différentes, le bigram de ces données n'est pas suffisant pour les différencier. C'est pourquoi nous ajoutons à notre modèle, les dérivées, telles que :

$$X_m^n = [x_m^n(2) - x_m^n(1), \dots, x_m^n(T) - x_m^n(T-1)] \quad (2)$$

Quand nous appliquons le bigram aux dérivées, nous gardons l'information liée aux tendances.

#### 4.2.2 Somme cumulée

Le même raisonnement peut être tenu dans le cas de l'ordre d'évènements. Si nous changeons d'ordre des blocs de points au sein d'une STM, nous en obtenons une nouvelle qui donne le même bigram. Dans ce cas la dérivée n'est plus non plus toujours suffisante. C'est pourquoi nous ajoutons la somme cumulée à notre ensemble, afin de retrouver l'ordre des points. Nous la définissons telle que :

$$\chi_m^n = [x_m^n(1) + x_m^n(2), \dots, \sum_{t=1}^{t=T} x_m^n(t)] \quad (3)$$

Quand nous appliquons le bigram sur la somme cumulée, nous gardons l'information à propos de l'ordre des évènements. Même si, dans notre cas, nous gardons la somme cumulée en entier, l'utilisateur peut garder seulement quelques points en considérant que l'information peut être perdue quand la somme est trop longue.

### 4.3 Construction finale

Dans le modèle, nous avons donc les données originales mais aussi leurs dérivées et leurs sommes cumulées. Comme précisé précédemment, nous avons implémenté la version bigram/histogramme et nous devons donc extraire les deux dimensions de manière aléatoire. Nous réitérons ce processus plusieurs fois afin d’extraire plusieurs paires de dimensions (Alg 1 row 3). Finalement, nous réalisons un système de vote sur classifieur afin d’obtenir une prédiction finale (Alg 1 row 4). De plus, nous avons ajouté une étape d’étude de la corrélation pour supprimer les informations redondantes (Alg 1 row 2). Cela nous permet d’avoir, un vote où toute information a le même poids, et un calcul plus rapide.

---

**Algorithm 1** Ensemble de bigrams et de histogrammes

---

**Require:** : STM, bins min max,  $R$  nombre de ré-itérations

1. Normalisation des données
  2. Suppression des corrélations par rapport à un seuil défini par un utilisateur
  3.  $k$  de 1 à  $R$  nombre maximum de ré-itérations
    - (a) Choix aléatoire de 2 dimensions (pas déjà analysés) parmi les  $M$  attributs de la STM
    - (b) Choix aléatoire de la transformation en bigram ou en histogramme
    - (c) Choix aléatoire des données originelles, dérivées, sommes cumulées
    - (d) Apprentissage afin de trouver le meilleur paramétrage de bins et transformation des données
  4. Vote sur classifieurs multiples afin de réaliser une prédiction finale.
- 

Le nombre de ré-itérations  $k$  est un paramètre du modèle et est défini par l’utilisateur où le nombre maximum de ré-itérations est appelé  $R$  dans le schéma Fig.2. Cette figure représente la construction du modèle final avec le concept de multi-vues. Nous pouvons voir que le paramétrage des histogrammes et bigrams est établi sur le jeu d’entraînement. Chaque bigram / histogramme permet d’établir une prédiction propre via un classifieur choisi par l’utilisateur. Finalement, nous réalisons un système de vote pour avoir une prédiction finale.

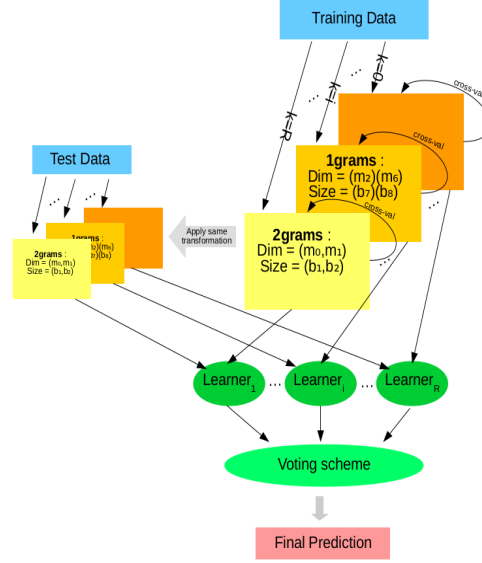


FIGURE 2 – Schéma de l’approche multi-vues constituée de bigrams et d’histogrammes.

## 5 Expérimentations

### 5.1 Conditions expérimentales

#### 5.1.1 Les données

Le site *UEA classification* fournit 30 jeux de données à propos des STM[BDL+18]. C’est l’archive la plus fournie à l’heure actuelle, cependant la taille des séries qui sont généralement petites, ne correspond pas parfaitement à l’application des Mgrams. Par ailleurs, comme ces jeux de données ont des longueurs fixes, nous ajoutons aussi 20 jeux de données provenant du site *Baydogan* [BR15], où la taille des STM varie. Les jeux de données provenant de ce site ont déjà servi de référence aux algorithmes WEASEL\_MUSE et SMTS. Par ailleurs, les deux archives présentent dans certains cas les mêmes jeux de données. Pour conclure, de tous les jeux de données, seulement *EigenWorms* correspond réellement au cas d’application des Mgrams, en terme de longueur de séries. Nous mettons à disposition une description complète des jeux de données sur notre site [PNMC19].

#### 5.1.2 Classifieurs

Dans ce papier, l’objectif est de montrer l’intérêt des Mgrams, nous n’avons utilisé qu’un seul type de classifieur, celui du plus proche voisin (1NN).

### 5.1.3 Paramètres

Les paramètres à régler sont le nombre de répétitions  $k$ , le nombre de bins de chaque dimension pour le bigram  $b_1, b_2$  et l’histogramme  $b$ .  $k$  est, dans notre implémentation, un pourcentage de combinaisons possibles lié au choix des dimensions, de la transformation et de la représentation :

$$k = \text{pourcentage} \times \binom{2}{M} \times 3 \times 2 \quad (4)$$

Pour rappel, le modèle choisit deux dimensions parmi  $M$ , puis peut effectuer deux transformations des données dérivé, somme cumulée) ou garder les données initiales, et enfin deux représentations (bigram, histogramme). Nous avons testé plusieurs pourcentages : 25%, 50%, 75% and 95%.

Pour le nombre d’intervalles du bigram, nous avons  $b_1, b_2 \in [2, 11]$  et  $[b \in [2, 50]$  pour l’histogramme. Les limites sont basées sur le fait qu’ajouter plus de choix, ajoute plus d’éléments potentiellement vides dans le tableau.

Par ailleurs, dans notre cas, nous ne prenons pas en compte le paramètre de l’évolution des données, présenté dans la Section 2, car nous avons réalisé une normalisation min-max. Cela conduit les données à évoluer seulement entre  $[0, 1]$ . Sinon, l’utilisateur peut choisir la normalisation de son choix.

## 5.2 Propriétés principales

Premièrement, nous allons exposer les propriétés principales de notre modèle, montrer l’influence de ses paramètres, ainsi que démontrer sa robustesse, et sa capacité à traiter les STM de longueurs variables. De plus, nous allons décrire dans cette partie, des variantes de notre modèle telles que la version du bigram carré ou du Mgram.

### 5.2.1 Robustesse

Comme expliqué dans la section 5.1.3, nous avons le paramètre  $k$  fonction d’un pourcentage de combinaisons. Nous avons essayé les pourcentages  $P \in [25\%, 50\%, 75\%, 95\%]$ . Nous montrons ici que même si nous n’exécutons qu’une petite partie du nombre de combinaisons possibles, *i.e.* 25%, nous avons des résultats robustes.

Nous pouvons voir sur la Fig.3 que l’écart-type diverge pour assez peu de jeux de données. Dans la plupart des autres cas, les variations sont faibles. Nous pouvons donc conclure qu’une majorité des dimensions

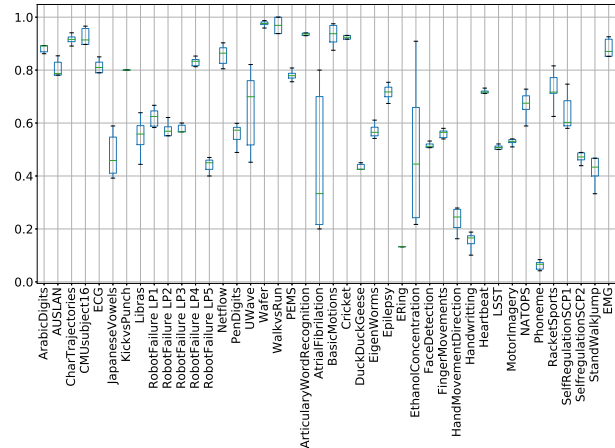


FIGURE 3 – Boîte à moustache des moyennes et des écart-types pour 25% des combinaisons.

d’une STM contiennent assez d’informations pour classifier les données. Et donc notre modèle avec 25% de combinaisons est suffisant pour émettre une classification robuste. Les autres jeux de données *AtrialFibrillation*, *EthanolConcentration* sont des cas spéciaux où seule une infime partie des dimensions contient réellement les informations nécessaires à la classification. De ce fait, lorsque nous extrayons les bigrams/histogrammes des mauvaises dimensions, ceux-ci ne permettent pas d’établir une bonne prédiction.

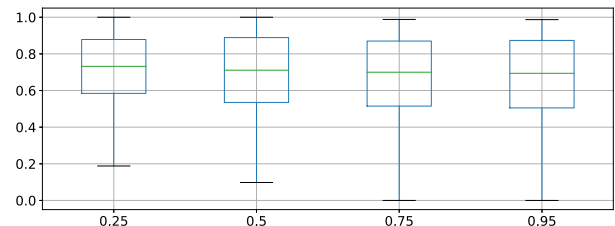


FIGURE 4 – Boîte à moustache des moyennes et écart-types pour 25%, 50%, 75%, 95% des combinaisons.

Nous montrons aussi dans la Fig.4 que le modèle ne nécessite pas le calcul de tous les bigrams/histogrammes pour avoir de bonnes performances. En effet, les moyennes sont meilleures et les écart-types sont plus petits pour 25% que pour 95%. Cela signifie qu’ajouter trop d’informations au modèle revient à ajouter du bruit dans le processus de prédiction. Dans tous les cas, il est mieux, en temps de calcul et en précision du modèle, de choisir de manière aléatoire de petits sous-ensembles de combi-

naisons plutôt que de tout calculer.

### 5.2.2 Longueur de séries

Dans la section 2, nous avons expliqué que les performances du Mgrams sont liées au nombre de points, et donc à la longueur, des STM. Comme le bigram est un outil statistique, le nombre de points peut impacter l'efficacité de l'outil.

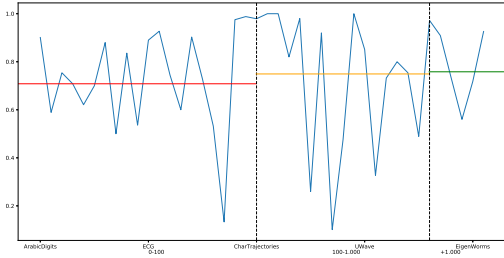


FIGURE 5 – Prédications du modèle par taille des STM avec moyennes par intervalle de longueurs : De 0 à 100 points, de 100 à 1000 points et plus de 1000 points.

Nous pouvons voir sur la Fig.5 qu'il y a une corrélation entre la taille des STM et les performances du modèle.

### 5.2.3 Activités

Nous avons aussi voulu voir si notre modèle obtient de meilleures performances pour des données au sein d'un certain domaine d'activité. Nous avons donc regroupé les jeux en groupes d'activités définis par [SL17] : Ecriture, Mouvement, Capteur et reconnaissance de paroles.

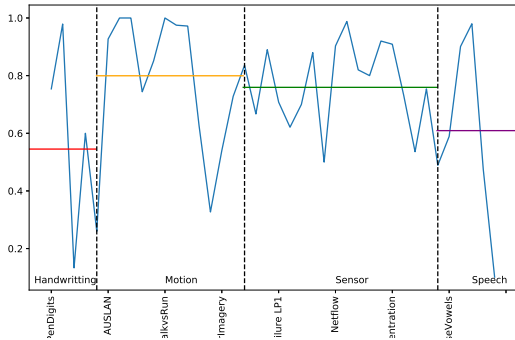


FIGURE 6 – Prédiction par domaines d'activité avec moyennes par domaines.

Nous ne pouvons pas tirer de conclusions à partir de la Fig. 6, à propos d'activités pour lesquelles notre modèle performerait mieux.

Donc nous pouvons conclure que le bigram peut être utile pour extraire, réduire, classifier les données et qu'il n'y a pas d'impact concernant les domaines d'activité.

### 5.3 Ensemble de Mgrams

Nous présentons ici un autre cas d'application de notre modèle.

Un Mgram est le résultat de la combinaison de toutes les dimensions  $M$  d'une STM. Nous créons alors un hypercube à  $M$  dimensions, qui compile les fréquences de tous les tuples  $[x_1^n(i), x_M^n(i)]$  par bins. Nous présentons les résultats ici de ces hypercubes dans le même modèle qu'appliqué précédemment. Premièrement, il doit être précisé que cet outil ne s'adapte pas bien, sur des données possédant beaucoup de dimensions, que ce soit en temps de calcul ou en espace de stockage. Par ailleurs, les hypercubes sont difficiles à représenter dans le cadre de l'interprétation des résultats. Nous n'avons donc testé que les jeux de données où  $M \leq 8$ . Par exemple, si nous avons une STM avec 8 dimensions et que nous la représentons dans un tableau où chaque axe n'a que 2 bins, nous obtenons un objet qui possède 256 éléments. La taille du tableau est définie par  $\prod_{i=1}^M b_i$  et la complexité par  $O(T(\sum_{i=1}^M b_i))$ .

Les résultats de ce modèle sont présentés dans le Tableau 1 sous l'appellation *M1gr1NN* et mènent à deux conclusions. La première est que ce n'est pas nécessaire de calculer un Mgrams plutôt qu'un bigram. Ce modèle a une performance inférieure, en générale, à celles des bigrams comme montré sur la Fig.7. De fait, cet objet demande plus de temps de calcul et est moins pertinent qu'un bigram. La deuxième conclusion est que certaines dimensions contiennent du bruit qui délaye les performances du modèle. Ce résultat confirme ceux obtenus sur la robustesse 5.2.1 ci-avant.

### 5.4 Bigram carré

Durant le paramétrage des bins du bigram/histogramme, un moyen de réduire le nombre de calcul est de prendre seulement des bins identiques pour les deux dimensions, comme expliqué dans la section 4.1. Ceci représente un bon compromis dans le cas de jeux de données très larges. Nous pouvons voir dans la Fig.7 que ce modèle est toujours préférable au Mgram.



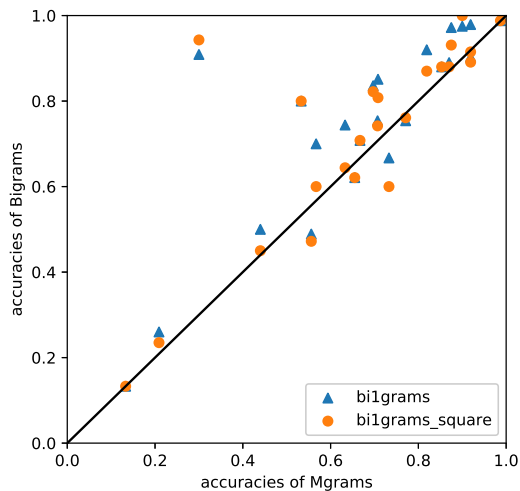


FIGURE 7 – Projection des prédictions (accuracies) des modèles bigram et bigram carré en fonction du modèle Mgram. Les points au-dessus de la droite  $x = y$  signifient que les bigram et bigram carré performant mieux que le Mgram.

## 5.5 SAX et bigrams

Il semblait aussi important de tester si, combiner les bigrams avec d'autres méthodes de classification STU, pouvait donner des résultats intéressants. C'est pourquoi, nous avons combiné les bigrams avec SAX. Nous avons gardé notre modèle comme dans Fig.2, seulement nous avons remplacé les histogrammes par SAX. Le but est de montrer que les bigrams peuvent être combinés facilement avec d'autres méthodes de classification de STU.

### 5.5.1 Utilisation de SAX

L'algorithme SAX a été créé en 2007 par *Lin et al.* [LKW07]. Pendant nos expérimentations, nous avons réglé le nombre de segments en apprentissage et nous avons fixé l'alphabet à dix chiffres de 0 à 9. Cela nous a permis de remplacer facilement les histogrammes dans le modèle initial. Nous avons choisi d'appliquer dessus une distance euclidienne.

### 5.5.2 Combinaison

Comme nous pouvons le voir sur le diagramme de différence critique Fig.8, le modèle Bigram-SAX performe mieux que le modèle de SAX seul. Par ailleurs,

il n'y a pas de différence entre le modèle abordé précédemment et ce nouveau modèle.

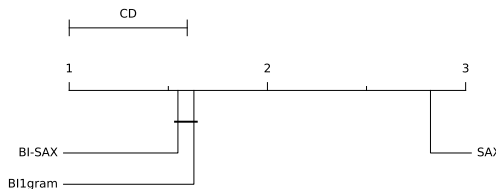


FIGURE 8 – Diagramme de différence critique pour le bigram, le modèle bigram-SAX et SAX.

Nous pouvons conclure que la combinaison de SAX avec les bigrams augmente les performances de l'algorithme seul. De ce fait, les bigrams peuvent se combiner facilement à d'autres méthodes de classification et en améliorer les performances.

## 5.6 Prédictions sur les jeux de données de référence

Finalement, nous avons réalisé une comparaison globale avec des méthodes de la littérature.

Nos méthodes, appelés *B1gr1NN* et *BSAX1NN* dans la table 1, sont donc comparés à *SMTS* [BR15], *WEASEL\_MUSE* [SL17], *DTW1NN<sub>I</sub>*, *DTW1NN<sub>D</sub>* [BDL<sup>+</sup>18] comme décrits dans la Section 3. Nous avons aussi le modèle *ED1NN* présenté dans [BDL<sup>+</sup>18], qui correspond au classifieur 1NN appliqué avec la distance euclidienne. Pour tous les algorithmes de la littérature, nous avons pris les résultats issus de leurs publications, car nous n'avons pas les paramètres nécessaires à leurs exécutions. Les résultats pour nos modèles sont issus du meilleur paramétrage réglé par validation croisée sur le jeu d'apprentissage. Les modèles *SAX1NN* et *BSAX1NN* correspondent aux applications de SAX et de SAX-bigrams. Le modèle *SquarB1gr1NN* est l'application du bigram carré quant à *M1gr1NN* c'est le modèle Mgram.

Nous pouvons voir que nos modèles sont parfaitement compétitifs par rapport aux modèles *DTW1NN<sub>I</sub>*, *DTW1NN<sub>D</sub>* dans la Fig.9a. Nous avons réalisé deux diagrammes de différence critique car les prédictions des différents modèles de la littérature ne sont pas disponibles sur les mêmes jeux de données. De plus, sur la Fig.9b nous pouvons aussi voir que bien que *SMTS* et *WEASEL\_MUSE* performant mieux que nos modèles il n'y a pas de différence critique. En effet, excepté le jeu de données *Japanese Vowels*, où nous avons une mauvaise performance, nos

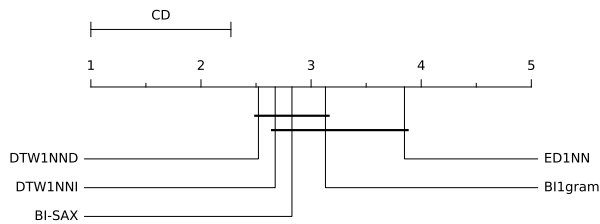


	EDINN	DTWINN <sub>I</sub>	DTWINN <sub>D</sub>	SMITS	WEASELMUSE	BIgrINN	SquareBIgrINN	M1grINN	SAXINN	BSAXINN
AUSLAN				0.947	<b>0.97</b>	0.927	0.902			
CharTrajectories	0.364	0.969	0.989	<b>0.992</b>	0.973	0.979	0.915	0.919		
GMUSubject16				<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>			
ECC				0.818	0.88	<b>0.89</b>	0.88	0.87		
JapaneseVowels	0.924	0.959	0.949	0.969	<b>0.976</b>	0.989	0.535			
KickvsPunch				0.82	<b>1.0</b>	<b>1.0</b>	0.8			
Libras	0.833	0.894	0.87	<b>0.909</b>	0.894	0.744	0.644	0.633	0.656	0.794
RobotFailureP1				0.856	0.94	0.708	0.708	0.667	0.50	0.688
RobotFailureP2				<b>0.76</b>	0.733	0.621	0.621	0.655	0.517	0.69
RobotFailureP3				0.76	<b>0.9</b>	0.7	0.60	0.567	0.5	0.6
RobotFailureP4				0.895	<b>0.96</b>	0.88	0.88	0.853	0.453	0.8
RobotFailureP5				0.65	<b>0.69</b>	0.45	0.45	0.44	0.44	0.52
NetFlow				<b>0.977</b>	0.961	0.903	0.891	0.919		
PenDigits	0.973	0.939	<b>0.977</b>	0.917	0.912	0.754	0.742	0.707	0.785	0.818
UWave	0.881	0.868	0.903	<b>0.941</b>	0.916	0.851	0.808	0.708	0.791	0.878
Wafer				0.965	<b>0.997</b>	0.988	0.988	0.987		
WalkvsRun				<b>1.0</b>	<b>1.0</b>	<b>1.0</b>	<b>1.0</b>			
ArabicDigits	0.367	0.959	0.963	0.964	<b>0.992</b>	0.901	0.90		0.677	0.98
ArticularyWordRecognition	0.97	0.98	<b>0.987</b>			0.98	0.96		0.467	<b>0.8</b>
AtrialFibrillation	0.267	0.267	0.22			<b>0.8</b>	<b>0.8</b>	0.533	0.425	0.95
BasicMovtions	0.676	<b>1.0</b>	0.975	<b>1.0</b>	<b>1.0</b>	0.972	0.931	0.875	0.472	<b>1.0</b>
Cricket	0.944	0.986	<b>1.0</b>	0.986	<b>1.0</b>	0.475	<b>0.45</b>		0.2	0.2
DrexelDrumGeese	0.275	0.55	<b>0.6</b>			0.626	0	<b>0.718</b>	0.291	0.58
EigenWorms	0.549	-	0.618			0.92	0.87	0.623	0.285	0.90
Epilepsy	0.666	<b>0.978</b>	0.964			<b>0.943</b>	<b>0.943</b>	0.304	0.285	0.654
EthanolConcentration	0.293	0.304	0.323			0.133	0.133	0.133	<b>0.3</b>	0.133
Ering	0.133	0.133	0.133			<b>0.532</b>	0.511		0.503	0.517
FaceDetection	0.519	-	0.529			0.6	0.59		0.49	<b>0.6</b>
FingerMovements	0.55	0.52	0.53			0.33	0.33		0.365	<b>0.381</b>
HandMovmentDirection	0.278	0.306	0.231			0.26	0.235	0.209	0.126	0.228
HandWriting	0.2	<b>0.316</b>	0.286			0.732	0.727		0.722	0.727
Heartbeat	0.619	0.658	0.717			0.536	0.535		0.441	0.539
LSSST	0.456	0.575	0.551			0.56	0.56	0.440	0.5	<b>0.56</b>
MotorImagery	0.51		0.5			0.728	0.733		0.594	0.767
NATOPS	0.85	0.85	0.883			0.82	0.792		0.792	<b>0.827</b>
PEMS	0.705	0.734	0.711			0.101	0.96		0.094	0.094
Phoneme	0.104	0.151	0.151			0.836	0.822	0.697	0.487	0.783
RacketSports	<b>0.868</b>	0.842	0.803			0.761	0.761	0.771	0.744	0.812
SelfRegulationSCP1	0.771	0.765	0.775			0.489	0.472	0.556	0.54	<b>0.583</b>
SelfRegulationSCP2	0.483	0.533	0.5339			<b>0.667</b>	0.6	0.733	0.467	<b>0.667</b>
StandWalkJump	0.2	0.333	0.2							

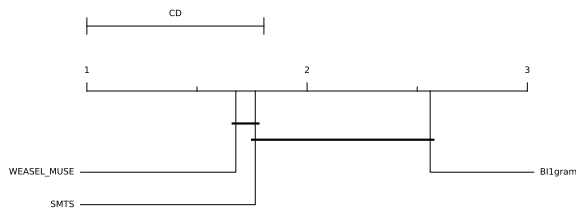
TABLE 1 – Prédications sur les jeux de données de référence.

La couleur *gris clair* correspond aux résultats que nous n'avons pas encore été publiés. Tandis que la couleur *gris foncé* correspond aux résultats que nous n'avons pas pu les compiler. Pour les modèles Bi-Sax et SAX, c'est car le modèle Sax est incompatible avec les variations de longueurs entre STM. Pour le modèle Mgram, c'est à cause du nombre de dimensions des STM incompatible avec le calcul de l'hypercube.

résultats sont très proches de ceux de *SMTS* et de *WEASEL\_MUSE*. De plus, comme expliqué dans la section 4, la complexité de notre modèle est meilleure. Pour ce qui est du jeu de données *Japanese Vowels*, nous performons mal à cause de la petite taille des séries et de la multitude des dimensions. Nous pouvons conclure, que comme supposé, nous performons mieux avec des STM longues, ce qui correspond à l’usage voulu de notre modèle.



(a) Diagramme des différences critiques pour les STM avec des longueurs fixes, comparés avec *ED1NN*, *DTW1NN<sub>D</sub>* et *DTW1NN<sub>I</sub>*



(b) Diagramme des différences critiques pour les STM avec des longueurs variables, comparés avec *SMTS* et *WEASEL\_MUSE*

FIGURE 9 – Diagramme des différences critiques sur les jeux de données de référence.

## 6 Conclusion

L’exploitation rapide de STM de longue taille, ainsi que la représentation compacte de ces séries, sont les challenges que nous avons relevés ici. Nous avons aussi montré que les bigrams donnent de bon résultats même sans être exhaustif dans l’exploration des dimensions. Ce sont aussi des objets visuels qui permettent une compréhension et une interprétation rapide des résultats. Finalement, nous avons démontré que notre méthode a une meilleure performance sur les STM longues. Enfin, les bigrams peuvent être combinés avec d’autres méthodes STU pour généraliser leur application aux STM. Par ailleurs, notre approche étant

basée sur plusieurs choix aléatoires des dimensions, il peut être envisager de rechercher d’abord des liens de dépendances entre dimensions en prélude à notre méthode.

## 7 Remerciements

Nous remercions l’Association Nationale de la Recherche et de la Technologie (ANRT) pour leur soutien financier, ainsi que les relecteurs pour leurs avis et corrections.

## Références

- [BDL<sup>+</sup>18] A. Bagnall, H. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The UEA multivariate time series classification archive, 2018. *CoRR*, abs/1811.00075, 2018.
- [BLB<sup>+</sup>17] 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. *DMKD*, 31(3) :606–660, 2017.
- [BLHB15] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with cote : The collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9) :2522–2535, 2015.
- [BR15] M. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *DMKD*, 29(2) :400–422, 2015.
- [Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2) :123–140, 1996.
- [DSP<sup>+</sup>18] H. Dau, D. Silva, F. Petitjean, G. Forestier, A. Bagnall, A. Mueen, and E. Keogh. Optimizing dynamic time warping’s window width for time series data mining applications. *Data Mining and Knowledge Discovery*, 32(4) :1074–1120, 2018.
- [JC16] I. Jolliffe and J. Cadima. Principal component analysis : a review and recent developments. *Philosophical Transactions of the Royal Society of London Series A*, 374 :20150202, 2016.
- [KP01] E. Keogh and M. Pazzani. *Derivative Dynamic Time Warping*, pages 1–11. 2001.

- [KS05] M. Kadous and C. Sammut. Classification of multivariate time series and structured data using constructive induction. *Machine Learning*, 58(2) :179–216, 2005.
- [LKWL07] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax : a novel symbolic representation of time series. *DMKD*, 15(2) :107–144, 2007.
- [LMSZ98] O. Le Maître, M. Süßner, and C. Zarak. Evaluation of tire wear performance. In *SAE Technical Paper*. SAE International, 1998.
- [MGQT13] S. Malinowski, T. Guyet, R. Quiniou, and R. Tavenard. 1d-sax : A novel symbolic representation for time series. In *Advances in Intelligent Data Analysis XII*, 2013.
- [ODF<sup>+</sup>08] P. Ordóñez, M. DesJardins, C. Feltes, C. U. Lehmann, and J. Fackler. Visualizing multivariate time series data to detect specific medical conditions. *Annual Symposium proceedings. AMIA Symposium*, page 530–534, 2008.
- [PFW<sup>+</sup>16] F. Petitjean, G. Forestier, G. Webb, A. Nicholson, Y. Chen, and E. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems*, 47(1) :1–26, 2016.
- [PNMC19] A. Plaud, E. NGuifo Mephu, and J. Charreyron. *MTS Mgrams*, April 2019. Available at <https://sites.google.com/view/mts-bihistograms/accueil>.
- [SC78] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1) :43–49, 1978.
- [SKS<sup>+</sup>15] G. Shah, A. Khan, A. Shah, M. Raza, and M. Sharif. A review on image contrast enhancement techniques using histogram equalization. *Science International*, 27 :1297–1302, 2015.
- [SL17] P. Schäfer and U. Leser. Multivariate time series classification with WEASEL+MUSE. *CoRR*, abs/1711.11343, 2017.
- [SYWK] M. Shokoohi-Yekta, J. Wang, and E. Keogh. *On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case*, pages 289–297.
- [TWL02] C. Tan, Y. Wang, and C. Lee. The use of bigrams to enhance text categorization. *Information Processing & Management*, 38(4) :529 – 546, 2002.
- [YAMP17] D. Yagoubi, R. Akbarinia, F. Masegla, and T. Palpanas. Dpisax : Massively distributed partitioned isax. In *ICDM : International Conference on Data Mining*, pages 1–6, 2017.
- [YK09] 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.
- [YS04] K. Yang and C. Shahabi. A pca-based similarity measure for multivariate time series. In *2Nd ACM International Workshop on Multimedia Databases*, pages 65–74. ACM, 2004.
- [YWW17] Q. Yin, S. Wu, and L. Wang. Unified subspace learning for incomplete and unlabeled multi-view data. *Pattern Recognition*, 67 :313 – 327, 2017.
- [ZC15] P-Y. Zhou and K. Chan. A feature extraction method for multivariate time series classification using temporal patterns. In *Advances in Knowledge Discovery and Data Mining*, pages 409–421. Springer International Publishing, 2015.
- [ZH14] J. Zbigniew and Z. Holger. The debts 2014 grand challenge. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS ’14, pages 266–269. ACM, 2014.
- [ZXXS17] J. Zhao, X. Xie, X. Xu, and S. Sun. Multi-view learning overview : Recent progress and new challenges. *Information Fusion*, 38 :43 – 54, 2017.