Sélections simultanées de variables et de représentations pour la classification de séries temporelles

Alexis Bondu*, Dominique Gay** Vincent Lemaire*, Marc Boullé*, Eole Cervenka***

* Orange Labs, Paris, France
alexis.bondu, vincent.lemaire, marc.boulle@orange.com
** LIM-EA2525, Université de La Réunion, France
dominique.gay@univ-reunion.fr
*** Boston Consulting Group, San Francisco, United States
eolecvk@gmail.com

Résumé. Cet article présente une méthode de classification de séries temporelle qui sélectionne des représentations alternatives (telles que les dérivées, les intégrales cumulatives, le spectre de puissance) et en extrait des descripteurs informatifs. L'approche proposée est décomposée en trois étapes : i) les séries temporelles originales sont transformées en plusieurs représentations stockées sous forme de données relationnelles ; ii) ensuite, une méthode de propositionalisation est appliquée pour "aplatir" les données relationnelles et en extraire des descripteurs ; (iii) enfin, un classificateur Bayésien est appris à partir des descripteurs résultants. Les étapes précédentes sont répétées par un algorithme de sélection de type "forward / backward" pour trouver le sous-ensemble de représentations le plus informatif. L'approche proposée s'avère très compétitive par rapport aux méthodes de l'état de l'art, et extrait des descripteurs interprétables.

1 Introduction

La littérature sur les séries temporelles traite diverses tâches d'apprentissage telles que la prévision, le clustering, etc. Nous abordons le problème de la Classification de Séries Temporelles (CST). Pour une série temporelle univariée notée $\tau_i = \langle (t_1, x_1), (t_2, x_2), \ldots, (t_m, x_m) \rangle$ où x_k est la valeur de la série a l'instant t_k, l'objectif est de prédire la valeur l'appartenance de τ_i à l'une des valeurs d'une variable cible catégorielle à partir d'un ensemble de séries temporelles préalablement étiquetées. Un consensus s'est dégagé au sein de la communauté sur le fait que la transformation des séries temporelles du domaine temporel vers un espace de représentation alternatif est l'un des meilleurs moyens pour améliorer la précision des modèles. Selon les auteurs, HIVE-CoTE Lines et al. (2018) fournit des résultats quasiment optimaux en termes de précision, et désormais, l'objectif des chercheurs est de développer des approches dont la précision serait comparable, mais qui améliorent d'autres critères comme l'évolutivité, l'interprétabilité, l'automatisation etc. Cela nous a motivés à mettre au point **FEARS** : une nouvelle méthode CST rapide et performante. Cette méthode extrait des descripteurs informatifs tout en sélectionnant simultanément des représentations utiles pour la CST.

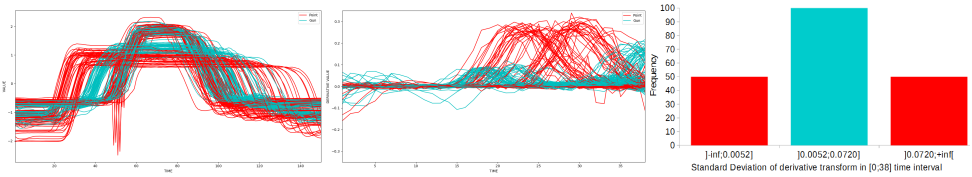

FIG. 1 – *"GunPoint" : représentation temporelle ((a) à gauche) originelle , ((b) au centre) à l'aide de la dérivée avec un focus sur l'intervalle de temps* $[0; 38]$. *((c) à droite) distribution de l'écart type des valeurs des dérivées (pour les classes GunDraw/Point) pour* $t \in [0; 38]$

La Figure 1 montre l'exemple d'un descripteur très discriminant extrait par notre approche sur le jeu de données GunPoint. La Figure 1 :a représente les séries du jeu de données dans le domaine temporel. A première vue, les deux classes semblent difficiles à discriminer, car les séries chronologiques rouge et verte sont confondues. La figure 1 :b montre la représentation dérivée de ces séries temporelles sur l'intervalle de temps $[0; 38]$. Visuellement, la dérivée semble être une représentation plus appropriée pour séparer les classes. Sur ce jeu de données, l'approche proposée a permis d'extraire un descripteur très informatif qui est l'écart-type des valeurs des séries dérivées, sélectionnées dans l'intervalle de temps $[0; 38]$. Trois intervalles permettent de séparer les classes. Toutes les séries temporelles appartenant à la classe "GunDraw" ont un écart-type entre $v_1 = 5.2 \times 10^{-3}$ et $v_2 = 7.2 \times 10^{-2}$, les autres valeurs correspondent à la classe "Point".

L'approche proposée [1] fonctionne comme suit : *(i)*, tout d'abord, nous transformons les séries temporelles originales en de multiples représentations qui sont stockées dans des tables secondaires, de la même manière que dans un schéma de données relationnelles ; *(ii)*, puis, des descripteurs informatifs et robustes sont extraits des données relationnelles, en utilisant une méthode régularisée de génération de descripteurs (i.e. propositionalisation Lachiche (2017); Boullé et al. (2019)) ; *(iii)*, troisièmement, un classificateur Bayésien est appris sur les données aplaties ainsi obtenues ; *(iv)*, de manière itérative, ces étapes sont répétées par un algorithme de type "feed forward / backward" afin de trouver le sous-ensemble de représentations le plus pertinent.

Travaux connexes - CST sur de multiple représentations : Un moyen efficace pour la CST est de prétraiter les séries temporelles en les transformant en représentations alternatives. Dans Bagnall et al. (2012), trois transformations sont effectuées (le spectre de puissance, l'auto-corrélation et l'analyse en composantes principales), puis un classificateur est construit sur chaque représentation, enfin les classifieurs sont combinés en utilisant un système de votes pondérés. Dans Bagnall et al. (2015), les auteurs étendent l'idée précédente en intégrant deux représentations supplémentaires basées sur l'auto-corrélation, une transformation en shapelets et ensuite en combinant 35 classifieurs. Récemment, Lines et al. (2018) a suggéré une combinaison hiérarchique probabiliste de cinq modèles ensemblistes et a obtenu les meilleurs résultats de précision connus à ce jour sur le dépôt de données UEA/UCR. Notre approche FEARS ouvre une nouvelle voie pour exploiter de multiples représentations : les données transformées sont stockées dans plusieurs tables reliées selon un schéma relationnel (i.e des données relationnelles). L'objectif est d'extraire des descripteurs pertinents et interprétables de ces tables

[1] Résumé en version française d'un article publié dans une conférence internationale en 2019.

reliées entre elles et de les aplatir dans un tableau de données unique. Ce processus d'exploration de données relationnelles peut être réalisé par une approche de propositionalisation Boullé et al. (2019)).

2 FEARS : Extraction & Sélection

Représentations - Comme dans l'article (Gay et al. (2013)), nous choisissons d'utiliser six représentations parmi les nombreuses qui existent dans la littérature : les dérivés (D), les doubles dérivés (DD), les intégrales cumulatives simples (CUMSUM) et doubles (DCUMSUM), l'auto-corrélation (ACF), le spectre de puissance (PS), et les séries représentées dans le domaine temporel (T). Le choix des représentations utilisées est guidé par deux critères : i) des travaux antérieurs en CST les utilisent ; ii) leur complexité temporelle est inférieure à $\mathcal{O}(N^2)$. **Schéma relationnel -** Les représentations précédemment calculées sont stockées dans des données relationnelles. Le "schéma" des données relationnelles définit la structure des différentes tables, leurs types *(i.e. principale ou tables secondaires)* et leurs liens. Le choix d'un schéma particulier peut avoir un impact significatif sur la qualité du classificateur appris. Nous considérons deux schémas potentiels. Le schéma **tout-en-un** est composé de : i) une table principale utilisée pour indexer les séries temporelles par leurs identifiant ; ii) une table secondaire par domaine *(temporel et fréquentiel)*. Le schéma **"un-par-un"** est composé de : i) la même table principale ; ii) une table secondaire par représentation. Les tables secondaires contiennent un index secondaire *(horodatage ou fréquence)* et stockent les valeurs de chaque point de données. Ces deux types de schémas correspondent à différentes façons de générer des descripteurs : *(i)* le schéma *"tout-en-un"* favorise la fouille de descripteurs basées conjointement sur plusieurs représentations ; *(ii)* le schéma *"un-par-un"* priorise l'extraction de descripteurs complexes indépendamment sur chaque représentation. Pendant la phase d'apprentissage, le meilleur schéma est choisi dynamiquement pour chaque jeu de données. La méthode de propositionalisation utilisée pour extraire des descripteurs pertinents à partir de ces données relationnelles est détaillée dans (Boullé et al., 2019).

Sélection de représentations - Les précédentes sections décrivent les premières étapes de notre approche qui visent à : i) transformer les séries temporelles originales en de multiples représentations ; ii) recoder ces multiples représentations sous la forme de données relationnelles ; iii) extraire des descripteurs informatifs de ces données et apprendre un classifieur. Toutes ces étapes sont répétées plusieurs fois pour sélectionner les représentations les plus informatives et le meilleur schéma relationnel. La Figure 2 présente l'algorithme de sélection utilisé qui est de type "forward / backward". Cet algorithme est répété deux fois pour sélectionner le meilleur schéma relationnel, c'est-à-dire, le schéma qui permet au classificateur d'atteindre les meilleures performances. L'étape (A) de la figure 2 consiste à choisir le meilleur point de départ pour la sélection de la représentation en fonction des performances du classifieur. L'étape (B) de la Figure 2 sélectionne les représentations les plus informatives en les ajoutant ou les supprimant selon un ordre aléatoire jusqu'à ce que la performance du classifieur ne soit plus amélioré d'au moins ϵ.

Require: X un ensemble de séries temporelles, y les étiquettes associées, $\mathcal{R}ep$ l'ensemble des représentations des séries temporelles, S l'ensemble des points de départ possibles pour la sélection de représentations ($\forall s \in S, s \subset \mathcal{R}ep$), φ un des deux schémas relationnels possibles, K le nombre maximum de variables explicatives générées à partir des données relationnelles, ϵ l'amélioration minimale du gain de compression pour ajouter ou retirer une représentation.

$s^* \leftarrow \emptyset$ */*Représentations sélectionnées*/*
$CG^* \leftarrow 0$ */*Meilleur gain de compression*/*
/* **(étape A)** *Choix des meilleures représentations.*/*
for all $s \in S$ **do**
  $classifier \leftarrow \texttt{Learn}(X, y, s, \varphi, K)$
  if $\texttt{CompressionGain}(classifier) > CG^*$ **then**
    $s^* \leftarrow s$
    $CG^* \leftarrow \texttt{CompressionGain}(classifier)$
  end if
end for
/* **(étape B)** *Sélection forward backward */*
while CG^* is $\epsilon - improved$ **do**
  / (étape B.1) Forward */*
  $\texttt{ForwardRepSelection}(Rep, CG^*, s^*)$
  / (étape B.2) Backward */*
  $\texttt{BackwardRepSelection}(Rep, CG^*, s^*)$
end while
return s^*

FIG. 2 – *Algorithme de sélection*

3 Expérimentations

Les expérimentations présentées ici ont pour but de répondre aux questions suivantes :

$\mathcal{Q}_1$ Concernant FEARS, y a-t-il un meilleur schéma relationnel ? Combien de représentations sont-elles sélectionnées ? Lesquelles ? La procédure de sélection de la représentation présente-t-elle un avantage sur le plan de la performance prédictive ? Et qu'en est-il de la complexité temporelle de l'approche ?

$\mathcal{Q}_2$ Les performances de **FEARS** sont-elles comparables aux méthodes de CST les plus récentes ? Et dans quelles conditions ?

Données/Paramétrage : Pour nos expériences, nous utilisons 85 jeux de données issus du dépôt UEA/UCR Dau et al. (2018) qui sont couramment utilisés dans la littérature. Ce dépôt présente un large panel de domaines d'application. Ces jeux de données sont fournis avec une répartition prédéfinie train / test. L'algorithme de la Figure 2 comporte quelques paramètres. Dans nos expériences nous utilisons $S = \{[TS, ACF], [TS, CUMSUM], [TS, D], [TS, CUMSUM, DCUMSUM], [TS, D, DD], [TS, D, DD, CUMSUM, DCUMSUM], [TS, D, DD, CUMSUM, DCUMSUM, ACF]\}$, et $\epsilon = 0.01$. L'approche de propositionalisation MODL génère au maximum $K = 30000$ descripteurs.

Approches concurrentes : L'approche proposée dans cet article est comparée à l'état-de-l'art : (1-NN-DTW) classifieur basé sur le plus proche voisin, utilisant la métrique Dynamic Time Warping (DTW) Sakoe et Chiba (1978) ; (1-N-DTW-CV) une variante qui règle la taille de la fenêtre de DTW par validation croisée ; (COTE) Bagnall et al. (2015) et son successeur HIVE-COTE Lines et al. (2018) ; (BOSS) pour Bag-of-SFA-Symbols Schäfer (2015) ; (WEASEL) pour Word ExtrAction for time SEries cLassification Schäfer et Leser (2017) ; Shapelet Transform (ST) Hills et al. (2014) ; Elastic Ensemble (EE) Lines et Bagnall (2015) qui est une méthode ensembliste combinant onze classificateurs ; Time Series Bag of Features (TSBF) Baydogan et al. (2013) ; l'apprentissage de shapelet (LS) Grabocka et al. (2014). H. I. Fawaz et al. Fawaz et al. (2019) ont réalisé un benchmark étendu des principales architectures Deep Learning utilisant les mêmes 85 jeux de données. Nous retenons les deux architectures de Deep Learning les plus performantes : (RESNET) et (FCN) Wang et al. (2017).

Effet du schéma relationnel : Le test de Wilcoxon a été appliqué aux 85 jeux de données et le résultat indique que le schéma "tout-en-un " donne de meilleurs résultats($z = -3,47$). Cela souligne la pertinence de générer des descripteurs issus de "représentation croisée" (voir Section 2). Comme décrit ci-dessus, l'approche proposée est capable de choisir le meilleur schéma relationnel, basé sur la performance du classificateur. Le choix dynamique du meilleur schéma est retenu dans notre approche, car cela améliore les performances (mais pas significativement selon le test de Wilcoxon).

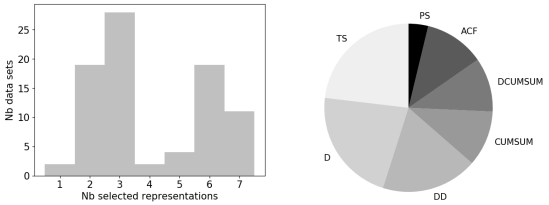

FIG. 3 – *((a) à gauche) Distribution du nombre de représentations sélectionnées ; ((b) à droite) Répartition des représentations.*

Sélection des représentations : Le temps de calcul de notre approche augmente linéairement avec le nombre de boucles de l'algorithme de sélection (voir Figure 2). Le meilleur sous-ensemble de représentations est trouvé en utilisant seulement deux boucles pour 71 des jeux de données, et trois boucles pour les autres. Le nombre de représentations sélectionnées varie également selon les cas. La Figure 3 :a montre la distribution du nombre de représentations sélectionnées sur tous les jeux de données La Figure 3 :b montre la distribution du nombre d'occurrences de sélection de chaque représentation. Dans notre benchmark, les représentations les plus fréquemment utilisées sont le domaine temporel (TS) et les dérivés (D et DD). Les intégrales cumulatives (CUMSUM, DCUMSUM) et la fonction d'auto corrélation (ACF) sont moins fréquemment utilisées. Enfin, le spectre de puissance (PS) est la représentation la moins utilisée.

Comparaison des résultats : Les expériences présentées dans cette section évaluent et comparent les performances de FEARS avec douze approches concurrentes sur les 85 jeux de données. Le test de Nemenyi Nemenyi (1962) est utilisé pour classer et regrouper les diffé-

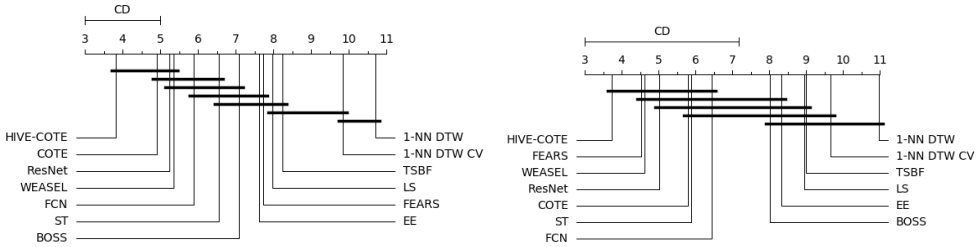

FIG. 4 – *Test de Nemenyi : ((a) à gauche) appliqué au 85 jeux de données ; ((b) à droite) appliqué aux jeux de données tel que* $N > 500$.

rentes approches. La Figure 4 :a montre le test de Nemenyi appliqué aux 85 jeux de données. FEARS appartient au quatrième groupe et se classe 9ème. La précision de l'approche FEARS est significativement supérieure à celle de la méthode de base (DTW 1-NN). FEARS étant une approche régularisée, elle favorise les modèles simples lorsque la taille du jeu de données d'apprentissage est insuffisante. En revanche, des modèles plus précis sont appris lorsque la taille du jeux de données augmente. En résumé, FEARS est une approche conservatrice sur de petits jeux de données, même si cela signifie sacrifier de la précision. Ce phénomène peut être observé dans nos expérimentations. Plus la taille des jeux de données augmente, plus la performance de FEARS croit par rapport aux approches concurrentes. La figure 4 :b montre le test Nemenyi appliqué aux jeux de données qui contiennent plus de 500 exemples d'apprentissage. Dans ce cas, le test de Nemenyi est appliqué aux 23 plus grands jeux de données. Cette fois-ci, FEARS est dans le premier groupe, se place second derrière HIVE-COTE et devant les approches WEASEL et ResNet. Ici, le test de Nemenyi indique qu'il n'y a pas de différence significative en termes de performance prédictive entre FEARS et les approches les plus performantes de la littérature CST. Ces résultats expérimentaux montrent que l'approche FEARS offre un très bon compromis entre la performance des modèles et le coût calculatoire pour les apprendre.

Efficacité en terme de scalabilité : La complexité temporelle de FEARS est en $\mathcal{O}(|\mathcal{R}ep|.K.N\ log(K.N))$, où N est le nombre d'exemples d'apprentissage, K est le nombre de descripteurs générés et $|\mathcal{R}ep|$ le nombre de représentations. Le code source de nos expérimentation utilise une bibliothèque de Machine Learning optimisée, qui implémente les approches MODL [2]. Nos expérimentations ont nécessité des ressources matérielles limitées et ont été réalisées en environ 90 heures en utilisant un simple poste de travail équipé d'un processeur 2Ghz Xeon E5-2650 et de 32 Go de RAM. La Figure 5 :a montre la distribution des temps de calcul lorsque les jeux de données sont traités sur un seul cœur. La plupart des jeux de données sont traités en moins de 5 heures. Et les trois plus gros jeux de données ont des temps de calcul plus importants *(ElectricDevices - 49h, NonInvasiveFatalECGThorax1 - 45h, NonInvasiveFatalECGThorax2 - 43h)*. La Figure 5 :b présente le temps de calcul en fonction de la taille N des jeux de données. Cette figure montre une tendance linéaire (en échelle logarithmique) du temps de calcul *vs.* N, ce qui est cohérent avec la complexité temporelle de l'approche FEARS.

[2]disponible sur `http://www.khiops.com`

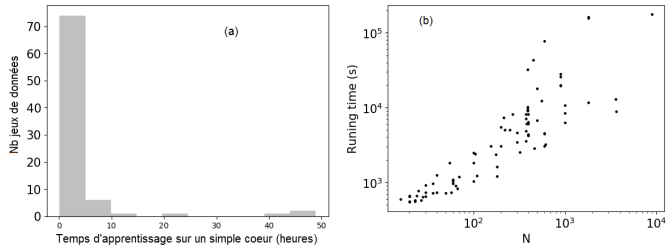

FIG. 5 – *(a) Distribution des temps d'apprentissage (en heures) ; (b) temps d'apprentissage (en secondes) pour chaque jeu de données en fonction de la taille de l'ensemble d'apprentissage, N (échelle logarithme).*

4 Conclusion

Au carrefour des communautés de classification des séries temporelles et d'exploration de données relationnelles, notre contribution, FEARS est une nouvelle méthode qui exploite de multiples représentations des séries temporelles pour une classification efficace et efficiente. L'originalité est d'apporter un point de vue différent - une vision relationnelle. Les concepts clés de FEARS sont le stockage de représentations multiples de séries temporelles dans un schéma de données relationnel, la construction/sélection de caractéristiques interprétables et la sélection de représentations. L'ensemble du processus permet d'obtenir des résultats très compétitifs en termes de précision par rapport aux récents concurrents à la pointe de la technologie sur les jeud de données de référence UCR/UEA, à condition qu'il y ait suffisamment de séries en apprentissage. De plus, l'approche suggérée permet d'extraire des caractéristiques interprétables des représentations sélectionnées, ce qui donne un compromis très avantageux entre (i) le temps de calcul, (ii) les résultats de précision et (iii) l'interprétabilité des variables construites. Ainsi, l'approche proposée peut être facilement utilisée dans un contexte industriel, en raison de son haut niveau d'automatisation, de performance et de facilité d'utilisation.

La vision relationnelle sur le domaine de la classification de séries temporelles offre une perspective naturelle pour les travaux futurs : les séries temporelles multivariées. Le stockage des différentes dimensions des séries multivariées (ou de plusieurs représentations alternatives de ces dimensions) dans des tables secondaires est naturellement "absorbable" par FEARS.

Les expériences sur la sélection de la représentation ont confirmé la prévalence du choix de la représentation dans divers domaines d'application de la classification de séries temporelles. Comme, a priori, aucune représentation ne se démarque des autres, il appartient aux experts du domaine d'application de préétablir un ensemble de représentations potentiellement pertinentes au départ. De plus, l'ensemble des fonctions des règles de construction ($\min, \max$, etc.) peut être alimenté par des fonctions dédiées supplémentaires. Heureusement, FEARS est assez "générique" pour permettre une personnalisation pilotée par le domaine.

Références

Bagnall, A., J. Lines, J. Hills, et A. Bostrom (2015). Time-series classification with cote : The collective of transformation-based ensembles. *IEEE Transactions on Knowledge & Data Engineering 27*(9), 2522–2535.

Bagnall, A. J., L. M. Davis, J. Hills, et J. Lines (2012). Transformation based ensembles for time series classification. In *Proceedings of the Twelfth SIAM International Conference on Data Mining, (SDM'12), Anaheim, California, USA, April 26-28, 2012.*, pp. 307–318.

Baydogan, M. G., G. Runger, et E. Tuv (2013). A bag-of-features framework to classify time series. *IEEE TPAMI 35*(11), 2796–2802.

Boullé, M., C. Charnay, et N. Lachiche (2019). A scalable robust and automatic propositionalization approach for bayesian classification of large mixed numerical and categorical data. *Machine Learning 108*(2), 229–266.

Dau, H. A., A. J. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, et E. J. Keogh (2018). The UCR time series archive. *CoRR abs/1810.07758*. `http://www.timeseriesclassification.com`.

Fawaz, H. I., G. Forestier, J. Weber, L. Idoumghar, et P. Muller (2019). Deep learning for time series classification : a review. *Data Mining & Knowledge Discovery 33*(4), 917–963.

Gay, D., R. Guigourès, M. Boullé, et F. Clérot (2013). Feature extraction over multiple representations for time series classification. In *New Frontiers in Mining Complex Patterns - Workshop NFMCP 2013, ECML-PKDD 2013, Revised Selected Papers*, pp. 18–34.

Grabocka, J., N. Schilling, M. Wistuba, et L. Schmidt-Thieme (2014). Learning time-series shapelets. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 392–401.

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

Lachiche, N. (2017). Propositionalization. In *Encyclopedia of Machine Learning and Data Mining*, pp. 1025–1031. Springer.

Lines, J. et A. Bagnall (2015). Time series classification with ensembles of elastic distance measures. *Data Mining & Knowledge Discovery 29*(3), 565–592.

Lines, J., S. Taylor, et A. J. Bagnall (2018). Time series classification with HIVE-COTE : the hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data 12*(5), 52 :1–52 :35.

Nemenyi, P. (1962). Distribution-free multiple comparisons. *Biometrics 18*(2), 263.

Sakoe, H. et S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26*(1), 43–49.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining & Knowledge Discovery 29*(6), 1505–1530.

Schäfer, P. et U. Leser (2017). Fast and accurate time series classification with WEASEL. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*, pp. 637–646.

Wang, Z., W. Yan, et T. Oates (2017). Time series classification from scratch with deep neural

networks : A strong baseline. In *2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14-19, 2017*, pp. 1578–1585.

Summary

This paper presents a method which extracts informative features while selecting simultaneously adequate representations for Time Series Classification (such as derivatives, cumulative integrals, power spectrum ...). The suggested approach is decomposed in three steps: (i) the original time series are transformed into several representations which are stored as relational data; (ii) then, a regularized propositionalisation method is applied in order to generate informative aggregate features; (iii) finally, a selective Naive Bayes classifier is learned from the outcoming feature-value data table. The previous steps are repeated by a forward backward selection algorithm in order to select the most informative subset of representations. The suggested approach proves to be highly competitive when compared with state-of-the-art methods while extracting interpretable features. Furthermore, the suggested approach is almost parameter free and only requires few hardware resources.