Construction de descripteurs à partir du coclustering pour la classification supervisée de séries temporelles

Dominique Gay*, Marc Boullé*

*Orange Labs, Lannion, FRANCE
prenom.nom@orange.com

Résumé. Nous présentons un processus de construction de descripteurs pour la classification supervisée de séries temporelles. Ce processus est libre de tout paramétrage utilisateur et se décompose en trois étapes : *(i)* à partir des données originales, nous générons de multiples nouvelles représentations simples ; *(ii)* sur chacune de ces représentations, nous appliquons un algorithme de co-clustering ; *(iii)* à partir des résultats de co-clustering, nous construisons de nouveaux descripteurs pour les séries temporelles. Nous obtenons une nouvelle base de données objets-attributs dont les objets (identifiant les séries temporelles) sont décrits par des attributs issus des diverses représentations générées. Nous utilisons un classifieur Bayésien sur cette nouvelle base de données. Nous montrons expérimentalement que ce processus offre de très bonnes performances prédictives comparées à l'état de l'art.

1 Introduction

La classification de séries temporelles (TSC) est un sujet qui a été intensivement étudié durant les dernières années. Le but est de prédire la classe d'un objet (une série temporelle ou courbe) $\tau_i = \langle (t_1, x_1), (t_2, x_2), \ldots, (t_{m_i}, x_{m_i}) \rangle$ (où $x_k, (k = 1..m_i)$ est la valeur de la courbe au temps t_k), étant donné un ensemble de séries temporelles labellisées d'apprentissage. Les problèmes de TSC sont différents des problèmes de classification supervisée dans les bases transactionnelles puisqu'il y a une dépendance temporelle entre les attributs ; ainsi l'ordre des attributs importe. La TSC est applicable dans de nombreux domaines dont les données sont des séries temporelles : e.g., pour le diagnostic médical (par exemple la classification d'électrocardiogramme de patients) mais aussi dans d'autres domaines comme la maintenance de machines industrielles, la finance, la météo, ... Le grand nombre d'applications a succité de nombreuses approches ; toutefois la majorité de la communauté s'est attachée à suivre le processus suivant (Liao, 2005) : *(i)* choisir une nouvelle représentation des données, *(ii)* choisir une mesure de similarité (ou une distance) pour comparer deux séries temporelles et enfin *(iii)* utiliser l'algorithme (NN) du plus proche voisin (avec la mesure choisie sur la représentation choisie) comme classifieur. Ding et al. (2008) propose un état de l'art des différentes représentations et mesures ainsi qu'une étude expérimentale comparative basée sur le classifieur NN. Il en ressort que le classifieur NN couplé avec une distance Euclidienne ou Dynamic Time Warping (DTW) présente les meilleures performances prédictives pour les problèmes de TSC.

Plus récemment, Bagnall et al. (2012) démontre expérimentalement que les performances de certains classifieurs augmentent fortement en utilisant certaines représentations (par rapport au domaine temporel original) ; ainsi, pour un classifieur donné, il existe une forte variance de performance selon la transformation de données utilisée. Pour pallier ce problème, Bagnall et al. (2012) proposent une méthode ensembliste basée sur trois représentations (ainsi que sur les données originales) : les résultats expérimentaux démontrent *(i)* l'importance de la représentation dans les problèmes de TSC et *(ii)* qu'une simple combinaison ensembliste de plusieurs représentations permet d'atteindre des performances prédictives très compétitives. Nous adhérons à cette conclusion sur l'importance des représentations ; toutefois une des faiblesses de certains classifieurs ensemblistes est la perte en interprétabilité due à la combinaison (par pondération) des classifeurs.

Un exemple illustratif : les graphiques de la figure 1 confirment l'intérêt du changement de représentation : si à partir des données originales (a), il n'est pas évident de différencier les deux classes (bleu/rouge), de simples transformations (ici l'intégrale cumulative (b) et la double intégrale cumulative (c)) facilitent la discrimination des classes. En effet, après transformation par double intégrale cumulative, les courbes (séries) ayant des valeurs supérieures à 100 sont bleues et les courbes avec des valeurs inférieures à -100 sont rouges. Sur cet exemple jouet (extrait de la base TwoPatterns de la base de l'UCR (Keogh et al., 2011)), une transformation et deux descripteurs nous permettent de caractériser les deux classes de courbes.

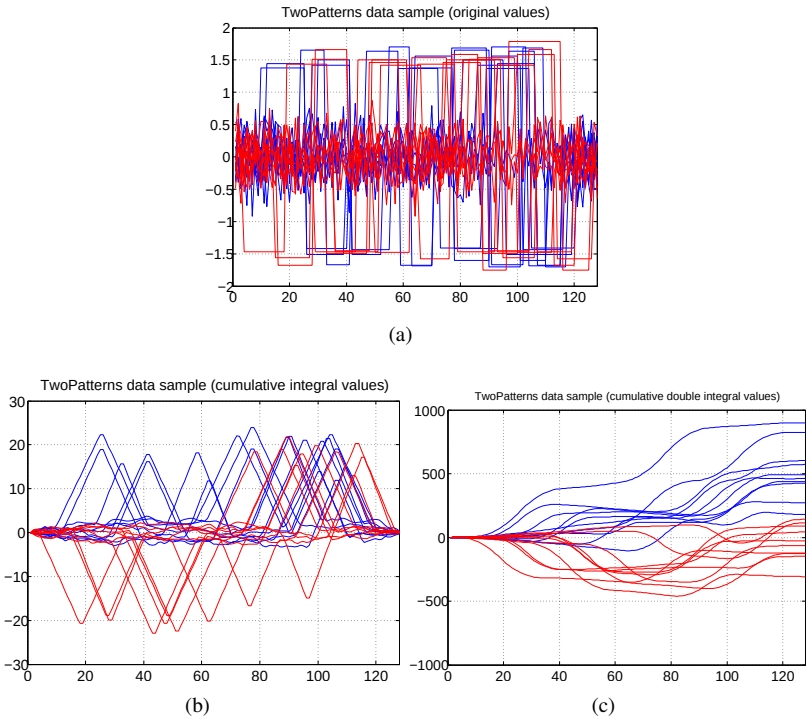


FIG. 1 – *Extrait de quelques séries temporelles pour deux classes de la base TwoPatterns : représentation originale, intégrale cumulative et double intégrale cumulative.*

Dans cet article, nous proposons un processus de construction de descripteurs interprétables pour le problème de TSC. Notre contribution est donc essentiellement méthodologique. La section suivante décrit les différentes étapes de notre processus : *(i)* une étape de transformation des données originales en de nouvelles représentations ; *(ii)* une étape de coclustering ; *(iii)* l'exploitation des résultats du coclustering pour construire de nouveaux descripteurs et ainsi une nouvelle base de données ; et enfin le classifieur utilisé. La section 3 rapporte la validation expérimentale de notre processus.

2 Processus de construction de descripteurs

Notations. Pour le problème de classification supervisée de séries temporelles (TSC), une série temporelle est définie par une paire (τ_i, y_i) où τ_i est un ensemble d'observations ordonnées $\tau_i = \langle (t_1, x_1), (t_2, x_2), \ldots, (t_{m_i}, x_{m_i}) \rangle$ de longueur m_i et y_i une valeur de classe. Une base de données de séries temporelles D est définie comme un ensemble de paires $D = \{(\tau_1, y_1), \ldots, (\tau_n, y_n)\}$, où chaque série temporelle peut avoir un nombre d'observations différent donc une longueur différente [1]. Le but est de construire un classifieur à partir de D pour prédire la classe de nouvelles séries temporelles $\tau_{n+1}, \tau_{n+2}, \ldots$

Pour ce faire, nous appliquons le processus de construction de descripteurs décrit par la figure 2 dont chaque étape est détaillée dans la suite.

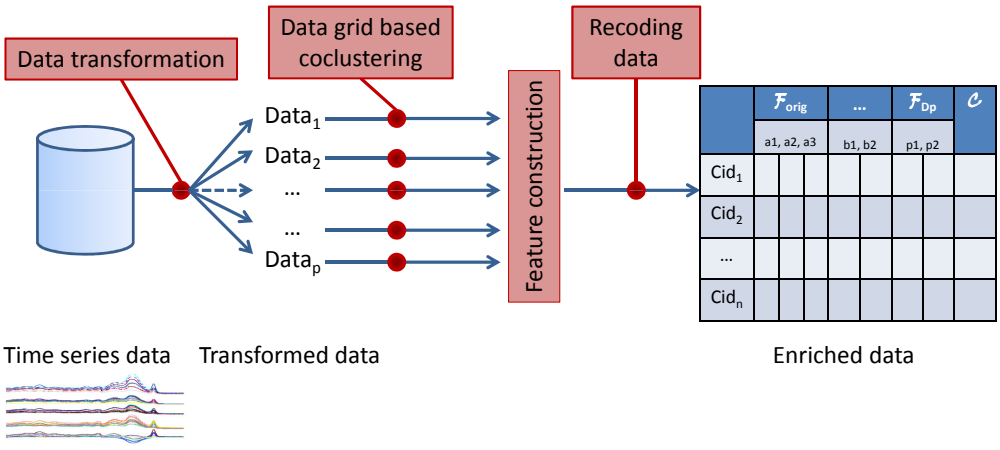


FIG. 2 – *Processus de construction de descripteurs. Etape 1 : Transformation des données en de multiples nouvelles représentations. Etape 2 : Coclustering sur chacune des représentations. Etape 3 : Construction d'un ensemble de descripteurs pour chaque résultat de coclustering pour la construction d'une nouvelle base de données réunissant les différents ensembles de descripteurs construits.*

1. Notons aussi que les séries d'une base peuvent avoir des valeurs différentes pour $t_k, (k = 1..m_i)$

2.1 Transformations/Représentations

De nombreuses méthodes de transformation ont été proposées dans la littérature pour représenter les séries temporelles : par exemple les transformations polynomiales, symboliques, spectrales, en ondelettes, ...(voir (Ding et al., 2008) pour une vue synthétique sur les représentations). Pour notre processus, nous utilisons les données originales ainsi que six représentations :

Les dérivées : DV et DDV Nous utilisons les dérivées et dérivées doubles des séries temporelles (i.e. les différences et différences doubles locales entre les valeurs au temps t et $t-1$).

$$DV(\tau_i) = \langle (t_1, 0), (t_2, \frac{(x_2 - x_1)}{(t_2 - t1)}), \ldots, (t_{m_i}, \frac{x_{m_i} - x_{m_i-1}}{t_{m_i} - t_{m_i-1}}) \rangle$$

$$DDV(\tau_i) = DV(DV(\tau_i))$$

Ces transformations nous permettent de représenter l'évolution locale (croissante/décroissante, accélération/décélération) des séries.

Les intégrales cumulatives : IV et IIV Nous utilisons aussi les intégrales cumulatives (simples et doubles) des séries temporelles (calculées via l'approximation par la méthode des trapèzes).

$$IV(\tau_i) =$$
$$\langle (t_1, 0), (t_2, (t_2 - t_1) \cdot \frac{x_1 + x_2}{2}), \ldots, (t_{m_i}, IV_{m_i-1}(\tau_i) + (t_{m_i} - t_{m_i-1}) \cdot \frac{x_{m_i-1} + x_{m_i}}{2}) \rangle$$

$$IIV(\tau_i) = IV(IV(\tau_i))$$

Ces transformations nous permettent de représenter l'évolution globale (accumulée) des séries.

Le spectre de puissance : PS. Une série temporelle peut-être décomposée en une combinaison linéaire de sinusoïdes d'amplitudes p, q et de phase w. Ainsi :

$$\tau_i(t) = \sum_{k=1}^{k=m_i} p_k \cos(2\pi w_k t) + q_k \sin(2\pi w_k t)$$

On appelle transformée de Fourier la série de paires $\tau_{i,FT} = \langle (p_1, q_1), \ldots (p_{m_i}, q_{m_i}) \rangle$. Et, τ_{if} le spectre de puissance (PS) est obtenu par la somme des carrés des coefficients de Fourier : $\tau_{if} = \langle (f_1, a_1), \ldots (f_{m_i}, a_{m_i}) \rangle$ où $a_k = p_i^2 + q_i^2$ ($k = 1..m_i$). Les f_k représentent la fréquence et les a_k la puissance du signal. Cette transformation nous permet de représenter la série dans le domaine de fréquence.

La fonction d'auto-corrélation : ACF La transformation par fonction d'auto-corrélation (ACF) est $\tau_{i\rho} = \langle (t_1, \rho_1), \ldots (t_{m_i}, \rho_{m_i}) \rangle$ où

$$\rho_k = \frac{\sum_{j=1}^{j=m_i-k} (x_j - \bar{x}) \cdot (x_{j+k} - \bar{x})}{m \cdot s^2}$$

et où $\bar{x}$ et s^2 sont la moyenne et la variance de la série originale. L'ACF décrit comment les valeurs originales séparées par une certaine durée évoluent ensemble. L'ACF permet de détecter des structures d'autocorrélation dans les séries temporelles.

Ainsi pour une base de données de séries temporelles D_{orig}, nous construisons six nouvelles bases de données : D_{DV}, D_{DDV}, D_{IV}, D_{IIV}, D_{PS}, D_{ACF} suivant la transformation utilisée. Dans la suite, par souci de généralisation, un objet d'une de ces représentations sera appelé "courbe" au lieu de série temporelle puisque D_{PS} n'est plus dans le domaine temporel.

2.2 Coclustering

Une courbe peut être vue comme un ensemble de points (X, Y) décrits par ses valeurs en abscisses et en ordonnées. Un ensemble de courbes peut être vu comme un ensemble de points (C_{id}, X, Y) où C_{id} est l'identifiant de courbe. Cette représentation tridimensionnelle (une variable catégorielle et deux variables numériques) d'une base de courbes est nécessaire à l'application de méthodes de coclustering. En effet, le but est de partitionner la variable catégorielle et discrétiser les variables numériques afin d'obtenir des clusters de courbes et des intervalles pour X et Y. Le résultat final est une grille tridimensionnelle dont chaque cellule est définie par un groupe de courbes, un intervalle de X et un intervalle de Y.
Pour ce faire, nous utilisons la méthode de coclustering KHC de Boullé (2012) utilisable via le logiciel KHIOPS [2]. Originalement développée pour le cas général des données fonctionnelles (Ramsay et Silverman, 2005), elle s'adapte bien pour le cas particulier des courbes comme définies ci-dessus. KHC est libre de tout paramétrage utilisateur, robuste (évite le sur-apprentissage), supporte des bases de données de courbes de plusieurs millions de points et sa complexité en temps est de $\Theta(N\sqrt{N}\log N)$ où N est le nombre de points de la base : c'est donc une méthode adaptée à notre problématique.

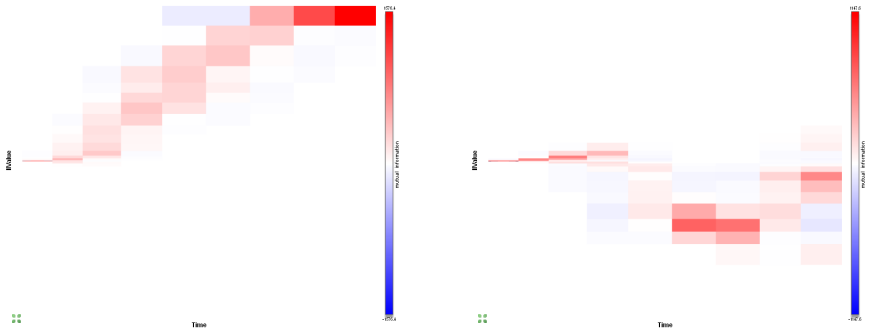
KHC est une méthode basée sur l'estimation de densité constante par morceaux et suit l'approche MODL (Boullé, 2006) (similaire à une approche Bayésienne (MAP) Maximum A Posteriori). Le modèle optimal M, i.e., la grille optimale est obtenue par optimisation (gloutonne bottom-up) d'un critère Bayésien qui mise sur un compromis entre précision et robustesse du modèle :

$$cost(M) = -\log(\underbrace{p(M \mid D)}_{\text{posterior}}) = -\log(\underbrace{p(M)}_{\text{prior}} \times \underbrace{p(D \mid M)}_{\text{vraisemblance}})$$

La grille obtenue constitue un estimateur non-paramétrique de la densité jointe des courbes et dimensions des points. Du point de vue de la théorie de l'information, selon Shannon (1948), les logarithmes négatifs de probabilités s'interprètent comme des longueurs de codage. Ainsi, le critère $cost$ peut être interprété comme la longueur de codage du modèle (la grille) plus la longueur des données D connaissant le modèle M, selon le principe de Minimum Description Length (MDL (Rissanen, 1978)).

Un exemple de visualisation de résultat de coclustering. La figure 3 présente un exemple de visualisation de deux clusters de courbes de la grille optimale obtenue pour la base de données TwoPatterns (transformée par IIV). Le graphique (a) (resp. (b)) présente un cluster dont les courbes sont majoritairement de classe c_1 (bleu dans l'exemple introductif de la figure 1),

2. http://www.khiops.com



(a) (b)

FIG. 3 – *Représentation de l'information mutuelle des cellules pour deux clusters de courbes obtenus par la méthode* KHC *sur la base TwoPatterns entière (transformée par IIV) : (a) cluster dont les courbes sont majoritairement de classe c_1 ; (b) cluster dont les courbes sont majoritairement de classe c_2. Plus les couleurs sont vives, plus la différence de distribution de points entre la cellule courante (donc du cluster) et le reste des données est significative.*

(resp. c_2, rouge dans l'exemple introductif). La grille optimale obtenue par KHC est composée de 133 clusters de courbes, 7 intervalles pour X et 22 intervalles pour IIV. L'estimateur de densité jointe obtenue (i.e. la grille optimale) est plus fin que nécessite le problème de départ : en effet, la base TwoPatterns est un problème de classification à 4 classes or nous obtenons 133 clusters de courbes ; ce qui nous donne un potentiel de caractérisation fine des classes du problème, lorsque la représentation s'y prête.

2.3 Construction de descripteurs

A partir de chaque résultat de coclustering obtenus sur chacune des représentations utilisées (D_{orig}, D_{DV}, D_{DDV}, D_{IV}, D_{IIV}, D_{PS}, D_{ACF}), nous créons un ensemble de descripteurs, i.e., $\mathcal{F}_{orig}$, $\mathcal{F}_{DV}$, $\mathcal{F}_{DDV}$, $\mathcal{F}_{IV}$, $\mathcal{F}_{IIV}$, $\mathcal{F}_{PS}$, $\mathcal{F}_{ACF}$. Ces descripteurs sont les attributs de notre nouvelle base de données dont les objets sont les courbes. Pour chaque représentation, nous créons trois types de descripteurs définis comme suit :
Soit D_{rep} une représentation parmi celles décrites plus haut. Soit $M_{rep} = KHC(D_{rep})$ la grille tridimensionnelle optimale obtenue par coclustering KHC sur D_{rep}. On note k_C le nombre de clusters de M_{rep} et k_Y le nombre d'intervalles de M_{rep} pour la dimension Y. Nous créons les attributs suivants :

– k_C attributs numériques (un pour chaque cluster C de courbes issu de M_{rep}) dont la valeur pour une courbe c_{id} est la distance définie par $d(c_{id}, C) = cost(M_{rep,c_{id} \cup C}) - cost(M_{rep})$, i.e., la différence de coût entre le modèle optimal M_{rep} et $M_{rep,c_{id} \cup C}$, la grille optimale dans laquelle on a intégré la courbe c_{id} au cluster de courbes C. Intuitivement, la distance d mesure la perturbation qu'apporte l'intégration d'une courbe à un cluster de la grille optimale.

- Un attribut catégoriel indiquant l'index du cluster de courbes i_C le plus proche d'un objet courbe c_{id} selon la distance définie ci-dessus (i.e. au sens du critère $cost$ utilisé pour l'optimisation de la grille).
- k_Y attributs numériques (un pour chaque intervalle i_Y de Y issu de M_{rep}) dont la valeur pour une courbe c_{id} est le nombre de points de c_{id} dans l'intervalle i_Y.

Ainsi pour une courbe donnée c_{id}, nous avons les informations suivantes fournies par les descripteurs (pour chaque représentation) : *(i)* la distance de c_{id} à tous les clusters de courbes, *(ii)* l'index du cluster de courbes le plus proche et *(iii)* le nombre de points de c_{id} dans chaque intervalle de Y.

2.4 Classification supervisée

Nous avons vu que notre processus de construction de descripteurs peut générer des centaines de descripteurs par représentation. Ainsi, l'ensemble total d'attributs générés $\mathcal{F}_{tot}$ peut contenir plusieurs milliers d'attributs. Le classifieur en fin de processus doit pouvoir supporter un grand nombre d'attributs et doit être capable de sélectionner les attributs pertinents pour la tâche de classification supervisée. Nous choisissons le classifieur sélectif Naive Bayes (SNB (Boullé, 2007)) qui répond à ces attentes. Notons aussi que le prédicteur SNB exploite des prétraitements (de type MODL) de variables numériques par discrétisation et de variables catégorielles par groupement de valeurs en utilisant des estimateurs de densité conditionnelle robustes. Ainsi les varibles construites profitent de ces prétraitements et offrent un potentiel d'interprétabilité (voir section 3). De plus, le SNB est libre de tout paramétrage utilisateur, ce qui facilite l'utilisation de l'ensemble du processus.

3 Validation expérimentale

L'implémentation de notre processus utilise des outils déjà existants (KHC pour le coclustering et SNB pour la classification supervisée, disponibles sur `http://www.khiops.com`). Le branchement entre ces outils est encore au stade de prototype et a été réalisé avec MATLAB.

Protocole. Pour valider notre processus, nous utilisons 26 bases de données de classification de séries temporelles : 17 bases de l'UCR (Keogh et al., 2011) et 9 nouvelles bases introduites dans (Bagnall et al., 2012). Une description succinte des caractéristiques de ces données est présentée dans la table 1. Cet ensemble de données présente une grande variété de bases tant en terme d'applications, qu'en terme de nombre d'instances, de classes et en longueur de série. Les expériences sont menées en suivant un protocole train-test prédéfini pour chaque base. Nous comparons notre processus de classification, qu'on appellera ici MODL-TSC, avec : *(i)* DTW-NN un classifieur basé sur le plus proche voisin et la distance Dynamic Time Warping, considéré par la littérature comme difficile à battre ; *(ii)* TSC-ENSEMBLE (Bagnall et al., 2012) qui exploite de multiples représentations via une méthode ensembliste et l'agorithme du plus proche voisin (NN).

Notons que depuis 2012, il existe d'autres bases de données répertoriées par l'UCR pour la TSC. Toutefois, nous nous limitons à ces 26 bases pour nos expérimentations comparatives car les performances prédictives de nos concurrents (rapportées de (Bagnall et al., 2012)) ne

Bases	#Train	#Test	Longueur	Classes
Ligthing2	60	61	637	2
Lighting7	70	73	319	7
ECG200	100	100	96	2
Adiac	390	391	176	37
FaceFour	24	88	350	4
50words	450	455	270	50
CBF	30	900	128	3
Fish	175	175	463	7
GunPoint	50	150	150	2
OSULeaf	200	242	427	6
SwedishLeaf	500	625	128	15
SyntheticControl	300	300	60	6
Trace	100	100	275	6
TwoPatterns	1000	4000	128	4
Wafer	1000	6174	152	2
Yoga	300	3000	426	2
FaceAll	560	1690	131	14
Beef	30	30	470	5
Coffee	28	28	286	2
OliveOil	30	30	570	4
Earthquakes	322	139	512	2
HandOutlines	1000	300	2709	2
FordA	3571	1320	500	2
FordB	3601	810	500	2
ElectricDevices	8953	7745	96	7
ARSim	2000	2000	500	2

TAB. 1 – *Description des bases de données de séries temporelles.*

sont accessibles que pour ces bases. D'autre part, les bases de séries de l'UCR sont un cas particulier du cadre général dans lequel nous nous plaçons, puisque pour une base donnée, toutes les séries sont de la même longueur et utilisent le même domaine temporel (i.e., les t_k sont identiques).

Résultats. Les résultats en terme de taux d'erreur sont reportés dans la table 2. Le meilleur résultat pour chaque base est mis en gras.
Premièrement, les résultats globaux (Taux d'erreur moyen, nombre de victoires et rang moyen) indiquent que MODL-TSC est très compétitif par rapport aux deux méthodes de l'état de l'art. Même si nous avons l'avantage numérique, cet ensemble de données ne nous permet pas de montrer qu'il y a une différence significative de performance entre les trois méthodes. En effet, nous avons procédé au test de Friedman (Demsar, 2006) et ne pouvons rejeter l'hypothèse nulle.

Notons les performances remarquables de MODL-TSC sur les bases OSULeaf, FordA, ElectricDevices et ARSim. Sur ces bases, la différence de performance est d'au moins 0,1 (i.e. 10%) par rapport à ses concurrents. Ici, l'apport des représentations (via les nouveaux descripteurs) est certainement à l'oeuvre dans notre processus, alors que TSC-ENSEMBLE n'exploite que 3 représentations et que DTW-NN se base sur les données originales. A l'inverse, les performances de MODL-TSC sont dramatiques pour les bases ECG200, Coffee et OliveOil. La différence de performance tourne en notre défaveur (au moins 0,1 par rapport aux deux concurrents). Nous pensons que cette différence peut être dûe à deux raisons : *(i)* les bases d'apprentissage de ECG200, OliveOil et Coffee sont très petites (quelques dizaines de courbes), ce qui rend l'apprentissage difficile ; *(ii)* nous n'avons pas encore trouvé la bonne représentation qui

Bases	DTW-NN	TSC-ENSEMBLE	MODL-TSC	#Attributs V/DV/DDV/IV/IIV/PS/ACF
Ligthing2	**0,1311**	0,2295	0,2623	74/21/20/49/45/24/52
Lighting7	0,2877	0,3014	**0,2603**	56/21/21/38/36/22/39
ECG200	0,12	**0,11**	0,21	17/14/14/21/28/15/22
Adiac	0,3887	**0,3555**	0,3581	27/30/29/32/34/31/63
FaceFour	0,1818	0,1364	**0,1023**	16/12/29/22/34/11/18
50words	0,3297	0,3516	**0,3143**	194/179/77/118/89/55/129
CBF	**0,0111**	0,1722	0,0344	15/3/3/18/22/7/15
Fish	**0,2171**	**0,2171**	0,2286	25/36/26/31/44/48/47
GunPoint	0,0867	0,0467	**0,02**	29/15/13/20/25/23/20
OSULeaf	0,3843	0,4215	**0,2562**	104/107/30/105/102/31/64
SwedishLeaf	0,1776	0,152	**0,0944**	26/29/32/33/46/19/30
SyntheticControl	**0,0233**	0,09	**0,0233**	19/14/14/25/36/13/22
Trace	0,01	0,19	**0,0**	32/13/5/35/39/22
TwoPatterns	**0,0007**	0,1	0,0118	765/42/40/234/156/17/38
Wafer	0,0045	0,0044	**0,0**	149/183/178/52/74/23/47
Yoga	0,1653	**0,1633**	0,2667	50/54/54/72/84/38/41
FaceAll	**0,1923**	0,2941	0,2953	33/29/21/42/65/24/24
Beef	**0,3667**	0,4	0,4667	25/14/8/29/34/18/25
Coffee	**0,0714**	0,1786	0,3571	15/11/13/21/22/15/22
OliveOil	**0,1667**	0,2	0,6	45/29/16/40/56/26/46
Earthquakes	0,2734	**0,2662**	0,2878	8/17/21/64/54/16/27
HandOutlines	0,16	**0,1243**	0,1271	76/365/303/94/118/126/83
FordA	0,267	0,1515	**0,0136**	185/87/49/233/259/156/78
FordB	0,3812	**0,2654**	0,3272	107/63/64/226/272/147/70
ElectricDevices	0,35	0,3779	**0,2608**	124/62/80/163/109/126/137
ARSim	0,4426	0,3215	**0,0005**	645/23/56/676/293/30/830
Moyenne ER	0,19965	0,21620	0,19918	-
#Victoires	9	7	12	-
Moyenne Rang	2	2,03846	1,88461	-

TAB. 2 – *Comparaisons des performances prédictives en terme de taux d'erreur (ER) pour les méthodes* DTW-NN, TSC-ENSEMBLE *et notre processus* MODL-TSC *sur 26 bases. La dernière colonne rapporte le nombre de descripteurs construits par* MODL-TSC *pour chaque représentation.*

permet de construire de bons descripteurs. C'est le cas pour OliveOil où l'étape de coclustering de chacune des six représentations ne génère qu'*un seul* cluster de courbes. Pour les autres (ECG200 et Coffee), même si l'étape de coclustering produit des clusters, donc fournit des descripteurs, la majorité d'entre eux ne sont pas considérés comme pertinents par le SNB.

Ces expériences rappellent l'importance des représentations pour la TSC d'une manière générale, et en particulier dans notre processus. Nous pouvons donc espérer une amélioration des performances prédictives en rajoutant des représentations de la littérature dans notre processus.

Interprétation : un exemple. Pour la représentation IV de la base TwoPatterns, la grille optimale obtenue par KHC est composée de 224 clusters de courbes, 11 intervalles pour X et 9 intervalles pour Y_{IV}. Les deux variables les plus pertinentes (parmi toutes les variables générées à partir de toutes les représentations) selon les prétraitements MODL du SNB sont issues de la représentation IV et sont :

1. v_1, le nombre de points dans l'intervalle $I_{Y_{IV}} =]-\infty; -3,9082]$
2. v_2, l'index du cluster le plus proche

Le groupement de valeurs pour v_2 et la discrétisation pour v_1 fournissent les tables de contingence suivantes en apprentissage (cf tables 3 et 4) :

$p =$ #points $\in I_{Y_{IV}}$	c_1	c_2	c_3	c_4
$0 \leq p \leq 7$	**100%**	0,00%	0,00%	0,00%
$7 < p \leq 12$	3,17%	17,46%	**79,37%**	0,00%
$12 < p \leq 26$	0,97%	51,21%	45,17%	2,66%
$26 < p \leq 29$	0,00%	25,58%	25,58%	48,84%
$29 < p$	0,46%	1,39%	0,93%	**97,22%**

TAB. 3 – *Table de contingence de la variable v_1, nombre de points dans l'intervalle $I_{Y_{IV}} =]-\infty; -3,9082]$ issue de la représentation IV.*

Groupes d'index de clusters	c_1	c_2	c_3	c_4
G_1	0,39%	3,53%	3,53%	**92,55%**
G_2	1,26%	0,42%	**97,91%**	0,42%
G_3	3,42%	**94,44%**	0,00%	2,14%
G_4	**95,22%**	2,21%	2,57%	0,00%

TAB. 4 – *Table de contingence de la variable v_2, index du cluster le plus proche, issue de la représentation IV.*

Nous observons (table 3) que le nombre de points p d'une courbe dans $I_{Y_{IV}}$ (i.e. le nombre de points dont la valeur est inférieure à -3,9082) est pertinent pour caractériser sa classe. En effet, en apprentissage, les courbes telles que $p \leq 7$ sont de classe c_1 ; lorsque $p > 29$ elles sont très majoritairement de classe c_4 et lorsque $7 < p \leq 12$ elles sont majoritairement de classe c_3.

Dans la table 4, nous observons tout d'abord que le prétraitement supervisé par groupement de valeurs MODL sur la variable v_2 ("index du cluster de courbes le plus proche") produit 4

groupes : G_1, (resp. G_2, G_3 et G_4) constitués de 56, (resp. 53, 53 et 62) index de clusters qui sont majoritairement de classe c_4 (resp. c_3, c_2, c_1). La forme diagonale de la table de contingence (table 4) indique la pertinence de l'attribut v_2 pour caractériser la classe d'une courbe. En effet, par exemple, si i_C l'index du cluster de courbes le plus proche d'une courbe c_{id} appartient au groupe G_2 (i.e. $i_C \in G_2$), alors c_{id} est considéré comme très similaire aux courbes de classe c_3. De plus, la variable *"index du cluster de courbes le plus proche"* est un indicateur de la pertinence de la représentation pour notre processus dans le problème courant de TSC. Dans cet exemple, la variable v_2 à elle seule permet de caractériser environ 95% de la base, donc la représentation IV est très pertinente pour caractériser les classes de la base TwoPatterns. A l'inverse, pour la représentation originale (D_V), la grille optimale générée par KHC est composée de 255 clusters de courbes mais le prétraitement indique que la variable *"index du cluster de courbes le plus proche"* n'est pas pertinente pour caractériser les classes de la base TwoPatterns.

4 Conclusion & Perspectives

Nous avons proposé MODL-TSC, un processus générique de construction de descripteurs pour le problème de la classification supervisée de séries temporelles (TSC). Ce processus est libre de tout paramétrage utilisateur et donc simple d'utilisation. Il se décompose en trois étapes : *(i)* génération de multiples représentations des données par le biais de transformations ; *(ii)* application d'une technique de coclustering sur chacune des représentations ; *iii* construction de descripteurs à partir des résultats du coclustering. La nouvelle base de données objets-attributs – dont les objets (identifiant les séries temporelles) sont décrits par des attributs issus des diverses représentations générées– est notre base d'apprentissage. Pour classer de nouvelles séries nous utilisons un classifieur naïf Bayésien sélectif. Les résultats expérimentaux ont montré que les performances prédictives de MODL-TSC sont très compétitives et comparables aux meilleures approches de la littérature.

Les premiers résultats expérimentaux sont prometteurs et confirment l'importance des transformations dans la TSC. En effet, selon les applications, certaines transformations faciliteront la découverte de motifs caractérisant les classes de séries temporelles. De plus, la combinaison de plusieurs représentations par le biais de notre processus MODL-TSC permet d'atteindre des performances prédictives très compétitives. Nous avons utilisé quelques représentations simples dans ces travaux préliminaires pour démontrer le bien fondé de ce processus de construction de descripteurs – ce qui nous laisse un potentiel d'amélioration des performances pour les données où MODL-TSC est moins performant que ses concurrents, pour peu qu'on trouve la bonne représentation. "Chercher la ou les bonnes représentations via une transformation" est certainement la principale perspective à ce travail et ce que nous pouvons recommander à ceux qui s'intéressent à la TSC. La littérature sur la TSC regorge de représentations (voir (Wang et al., 2012) pour une vue d'ensemble) et trouver une bonne représentation pour la TSC est toujours un sujet d'actualité (e.g. (Lines et al., 2012)).
D'autre part, une perspective pratique sera d'identifier la ou les bonnes représentations pour un domaine d'application spécifique (e.g., ECG, consommation électrique, ...).

Références

Bagnall, A., L. M. Davis, J. Hills, et J. Lines (2012). Transformation based ensembles for time series classification. In *SIAM DM'12*, pp. 307–318.

Boullé, M. (2006). MODL : A bayes optimal discretization method for continuous attributes. *Machine Learning 65*(1), 131–165.

Boullé, M. (2007). Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research 8*, 1659–1685.

Boullé, M. (2012). Functional data clustering via piecewise constant nonparametric density estimation. *Pattern Recognition 45*(12), 4389–4401.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7*, 1–30.

Ding, H., G. Trajcevski, P. Scheuermann, X. Wang, et E. J. Keogh (2008). Querying and mining of time series data : experimental comparison of representations and distance measures. *PVLDB 1*(2), 1542–1552.

Keogh, E., Q. Zhu, B. Hu, H. Y., X. Xi, L. Wei, et C. A. Ratanamahatana (2011). The UCR time series classification/clustering. `http://www.cs.ucr.edu/~eamonn/time_series_data/`.

Liao, T. W. (2005). Clustering of time series data - a survey. *Pattern Recognition 38*(11), 1857–1874.

Lines, J., L. M. Davis, J. Hills, et A. Bagnall (2012). A shapelet transform for time series classification. In *KDD'12*, pp. 289–297.

Ramsay, J. et B. Silverman (2005). *Functional data analysis*. Springer.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica 14*, 465–471.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*.

Wang, X., A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, et E. Keogh (2012). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 1–35.

Summary

We suggest a parameter-free process for feature construction for time series classification. Our process is decomposed in three steps: *(i)* we transform original data into several simple representations; *(ii)* on each representation, we apply a coclustering method; *(iii)* we use coclustering results to build new features for time series. It results in a new transactional data set, made of time series identifiers described by features related to the various generated representations. We show that a Selective Naive Bayes classifier on this new data set is highly competitive when compared with state-of-the-art times series classification methods.