Comparaison de descripteurs pour la classification de décompositions parcimonieuses invariantes par translation

Quentin BARTHÉLEMY[1,2], Maxime SANGNIER[1], Anthony LARUE[1], Jérôme I. MARS[2]

[1]CEA, LIST, Laboratoire d'Outils pour l'Analyse de Données
Gif-sur-Yvette Cedex, F-91191, France

[2]GIPSA-Lab, DIS, UMR 516 CNRS, Grenoble INP
Grenoble, F-38402, France

prénom.nom@{cea.fr,gipsa-lab.grenoble-inp.fr}

Résumé – Nous étudions les descripteurs adaptés à la classification de décompositions parcimonieuses invariantes par translation. Nous comparons les différents descripteurs de l'état de l'art sur les mêmes données et avec le même classifieur, ce qui permet d'évaluer leurs efficacités et nous testons aussi leur robustesse à la translation. Grâce à un nouveau fenêtrage, une famille de nouveaux descripteurs est proposée, dépassant l'état de l'art tout en étant robuste à la translation.

Abstract – Descriptors adapted for the classification of shift-invariant sparse decompositions are studied. Descriptors of the state-of-the-art are compared on the same data and with the same classifier, that allows to evaluate their efficiencies, and their robustness to shift is also tested. Thanks to a new window, a family of new descriptors is proposed, which surpasses state-of-the-art and which is robust to shift.

1 Introduction

Nous voulons classer différents signaux temporels, composés de structures qui peuvent être décalées/translatées dans le temps alors qu'elles renvoient au même phénomène. La classification de ces signaux doit donc être robuste à la translation. Dans [1], deux approches sont distinguées : le cas invariant par translation : si $\mathsf{f}(t) \mapsto \mathsf{g}(t)$, alors $\mathsf{f}(t+t_0) \mapsto \mathsf{g}(t+t_0)$, et le cas consistant par translation : si $\mathsf{f}(t) \mapsto \mathsf{g}(t)$, alors $\mathsf{f}(t+t_0) \mapsto \mathsf{g}(t)$. Etudiant des décompositions invariantes par translation [1, 2], la classification robuste à la translation s'obtient en appliquant une fonction qui extrait des descripteurs qui ne varient pas, *i.e.* qui est consistante par translation. Plusieurs fonctions ont été proposées mais elles n'ont jamais été comparées. Après une formalisation du problème, nous ferons la revue des descripteurs adaptés à ce problème et nous en proposerons une comparaison sur les mêmes données et avec le même classifieur. Nous introduirons enfin un nouveau fenêtrage, et nous montrerons sa robustesse et son efficacité.

2 Décompositions parcimonieuses de signaux temporels

Nous étudions la décomposition linéaire d'un signal temporel $y \in \mathbb{R}^N$ composé de N échantillons indicés par t. Dans le modèle d'invariance par translation [1], le signal y est décomposé comme la somme de L structures élémentaires $\{\psi_l\}_{l=1}^L$, appelées noyaux, caractérisées indépendamment de leurs positions. Considérant le noyau $\psi_l \in \mathbb{R}^T$, il peut être translaté à toutes les positions $\tau \in \sigma_l$, avec σ_l un sous-ensemble des N indices t. Le modèle invariant par translation s'écrit donc :

$$y(t) = \sum_{l=1}^{L} \sum_{\tau \in \sigma_l} x_{l,\tau} \psi_l(t-\tau) + \epsilon(t) \,,$$

avec x les coefficients de décomposition et ϵ le résidu. L'ensemble des noyaux translatés à tous les échantillons constitue une matrice par bloc de Toeplitz, appelé dictionnaire. Nous ajoutons une contrainte de parcimonie formalisée par la pseudo-norme ℓ_0 notée $\|x\|_0$ et définie comme le nombre d'éléments non nuls de x. L'approximation K-parcimonieuse estime les coefficients x en sélectionnant seulement K éléments non nuls :

$$\min_x \left\| y(t) - \sum_{l=1}^{L} \sum_{\tau \in \sigma_l} x_{l,\tau} \psi_l(t-\tau) \right\|_2^2 \text{ t.q. } \|x\|_0 \leq K \,.$$

Ce problème peut être résolu par l'*Orthogonal Matching Pursuit* (OMP) ou par d'autres algorithmes [3]. La décomposition parcimonieuse résultante composée des K éléments actifs s'écrit alors :

$$y(t) = \sum_{k=1}^{K} x_{l^k,\tau^k} \, \psi_{l^k}(t-\tau^k) + \epsilon(t) \,.$$

L'ensemble des K coefficients actifs $\left\{x_{l^k,\tau^k}\right\}_{k=1}^K$ est affiché par une représentation temps-noyaux appelée spikegramme, illustrée en Fig. 1 sur des données d'écriture manuscrite.

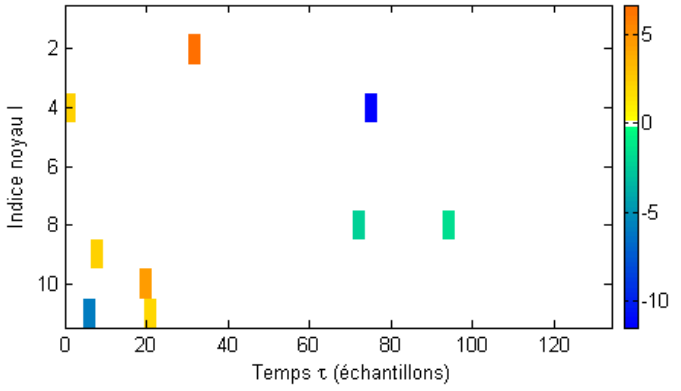

FIGURE 1 – Spikegramme de la lettre e où l'amplitude (couleur) des coefficients $x_{l,\tau}$ est représentée en fonction du temps τ (abscisse) et de l'indice du noyau l (ordonnée).

3 Descripteurs calculés par fonctions de groupement

Nous considérons maintenant un ensemble Y de signaux indicés par $p = 1..P$. Nous cherchons des fonctions qui extraient d'un spikegramme un vecteur de caractéristiques répondant aux critères évoqués ci-dessus. Une fonction de groupement, *pooling function* en anglais, est une fonction d'intégration qui calcule un vecteur de caractéristiques sur un voisinage, de telle sorte que la localisation exacte des éléments sur ce voisinage ne rentre pas en compte dans le calcul [4]. Ainsi, seule l'appartenance au voisinage importe, et non la localisation exacte ou l'ordre. Considérant le signal y_p, le vecteur de caractéristique $\mathsf{v}_p \in \mathbb{R}^L$ est calculé par la fonction de groupement suivante :

$$\mathsf{v}_p(l) = \sum_{k=1}^{K} \left| x_{l_p^k, \tau_p^k} \right|^s \quad \text{t.q.} \quad l_p^k = l \,, \tag{1}$$

pour des valeurs de $s = 0,\ 0.5,\ 1,\ 2$ [5, 6] et $+\infty$ (équivalente au *max-pooling*) [7]. Cette fonction est équivalente à une norme ℓ_s : $\mathsf{v}_p(l) = \|x_{l,.}\|_s^s$. Toutefois, les fonctions décrites en Eq. (1) engendrent une perte d'informations, notamment sur l'ordre temporel des coefficients. Grosse *et al.* suggèrent l'idée d'utiliser les fonctions de groupement sur des fenêtres décalées [5], créant ainsi plusieurs voisinages au lieu d'un seul, mais aucun détail d'implémentation n'est donné.

Dans [8] est décrit comment appliquer une fonction de groupement sur F fenêtres temporelles de Hanning, indicées par $f \in \mathbb{N}_F$. Une fenêtre de Hanning de longueur temporelle T_H est définie par :

$$\text{Hanning}(t) = \begin{cases} \frac{1}{2}\left(1 + \cos\left(\frac{2\pi}{T_H}\, t\right)\right), & \text{pour } t \in \left[-\frac{T_H}{2}, \frac{T_H}{2}\right] \\ 0 & \text{sinon} \end{cases} .$$

La taille T_H est calculée à partir du temps maximum de la base de donnée Y donné par :

$$N_{max} = \max_p \left\{ \max_k \left\{ \tau_p^k \right\}_{k=1}^{K} \right\}_{p=1}^{P} . \tag{2}$$

Les F fenêtres sont recouvrantes de 50 %, soit de $T_H \times 0.5$. Ainsi, leur taille est égale à $T_H = \frac{N_{max}}{0.5 \times F}$. L'inconvénient de l'approche fenêtrée est qu'il faut choisir la position temporelle de la première fenêtre. Ici, le centre de la première fenêtre est placé sur le temps $\tilde{\tau}$ d'apparition du premier atome de chaque spikegramme : $\tilde{\tau}_p = \min_k \left\{ \tau_p^k \right\}_{k=1}^{K}$. L'information du spikegramme est intégrée sur une matrice $\mathsf{V}_p \in \mathbb{R}^{L \times F}$ définie telle que :

$$\mathsf{V}_p(l, f) = \sum_{k=1}^{K} \text{Hanning}\left(\tau_p^k - (f-1) \times \frac{T_H}{2} - \tilde{\tau}_p\right) \times x_{l_p^k, \tau_p^k}$$
$$\text{t.q.} \quad l_p^k = l \,.$$

Ainsi, chaque élément $\mathsf{V}_p(l, f)$ correspond à la projection des atomes associés au noyau l sur la fenêtre de Hanning centrée en $(f-1) \times \frac{T_H}{2} + \tilde{\tau}_p$. Cette fonction de groupement, effectuant la somme et aussi appelée *mean-pooling* en anglais, sera notée $\sum(.)$. A la fin, chaque matrice V_p est vectorisée horizontalement dans un vecteur $\mathsf{v}_p \in \mathbb{R}^{1 \times LF}$: $\mathsf{v}_p((l-1) \times F + f) \leftarrow \mathsf{V}_p(l, f)$. La consistance par dilatation s'obtient facilement en normalisant chaque vecteur v_p obtenu : $\mathsf{v}_p \leftarrow \mathsf{v}_p / \|\mathsf{v}_p\|_2$.

Par analogie, les fonctions de groupement décrites dans l'Eq. (1) sont appliquées sur les F fenêtres de Hanning, de la même manière que l'approche précédemment décrite. Seule la définition de la matrice V_p change. Ces fonctions seront abrégées par leurs normes ℓ_s associées.

Au final, tous ces vecteurs de caractéristiques normalisés peuvent être maintenant utilisés comme entrées des systèmes d'apprentissage automatique [9], comme les SVM (*Support Vector Machine*), MLP (*Multi-Layer Perceptron*), etc.

Nous venons de faire la revue des méthodes d'extraction de caractéristiques consistantes par translation, mais elles n'ont jamais été comparées entre elles avec le même classifieur et les mêmes données.

4 Comparaison des descripteurs

Les méthodes sont appliquées sur la base de données UCI *Character Trajectories* [10]. Elle est composée de $P = 1430$ signaux d'apprentissage $\boldsymbol{Y}_a$ et de $Q = 1425$ signaux de test $\boldsymbol{Y}_t$, repartis en $C = 20$ classes. Ces signaux multivariés (vitesses cartésiennes et pression) sont décomposés par le *Multivariate* OMP grâce à des noyaux appris [2], et leurs vecteurs de caractéristiques sont calculés pour les fonctions de groupement précédemment étudiées. Les vecteurs de caractéristiques sont répartis entre l'ensemble d'apprentissage $\mathbf{V}_a$ et l'ensemble de test $\mathbf{V}_t$.

Nous apprenons un SVM multiclasse à noyaux Gaussiens [9] en mode un contre tous sur l'ensemble d'apprentissage $\mathbf{V}_a$. Deux paramètres sont optimisés : le paramètre de coût ou compromis c et l'écart type du noyau σ. Nous faisons une validation croisée (CV) de type N-*fold* [9] sur une grille de paramètres de $\mathsf{C} = 18$ valeurs de c et de $\Sigma = 15$ valeurs de σ, avec $\mathsf{N} = 5$. Le taux de classification η est calculé sur l'ensemble de

TABLE 1 – Taux de classification $\bar{\eta}$ sur les vecteurs de caractéristiques extraits des signaux.

Fonctions	ℓ_0	$\ell_{0.5}$	ℓ_1	ℓ_2	ℓ_∞	$\sum(.)$
Taux $\bar{\eta}$ (%)	38.37 ± 0.60	59.76 ± 0.74	65.00 ± 0.46	64.18 ± 0.19	66.42 ± 0.40	76.40 ± 0.94
Taux $\bar{\eta}$ (%) - fenêtres	60.19 ± 0.38	79.96 ± 0.13	82.53 ± 0.36	78.44 ± 0.83	83.54 ± 0.29	90.18 ± 0.21
Taux $\bar{\eta}$ (%) - fenêtres $'$	60.23 ± 0.38	79.98 ± 0.06	84.81 ± 0.30	80.28 ± 0.94	85.05 ± 0.23	90.88 ± 0.06

test $\mathbf{V}_t$. Cet apprentissage est réalisé pour les différentes fonction de groupement décrites précédemment, mais avec la même permutation aléatoire de l'ensemble d'apprentissage $\mathbf{V}_a$. Pour la validation croisée, $\mathbf{V}_a$ est divisé en un ensemble d'entrainement (de taille 4/5) et un ensemble de validation (de taille 1/5) qui tournent successivement (plus de détails dans [11]). Ce processus est moyenné 10 fois et nous relevons le taux de classification moyen $\bar{\eta}$.

La Table 1 (1ère et 2ième ligne) résume les taux de classification moyens obtenus par les différentes fonctions de groupement [1]. Nous observons que les fonctions issues de normes fenêtrées ont de meilleurs résultats que celles non fenêtrées. Les normes intègrent temporellement les coefficients sur tout le spikegramme, perdant ainsi toute information liée à l'ordre temporel, alors que les fenêtres préservent la localisation en temps des coefficients. D'autre part, les fonctions de groupement issues de normes (fenêtrées ou pas) perdent le signe des coefficients, alors que ce n'est pas le cas de la fonction $\sum(.)$, d'où le meilleur taux de classification.

5 Test de robustesse à la translation

Nous effectuons deux tests pour vérifier les consistances par dilatation et par translation temporelle des fonctions de groupement. Dans un premier temps, nous testons la consistance par dilatation. Pour chaque signal de l'ensemble de test $\boldsymbol{Y}_t$, nous appliquons un coefficient multiplicatif tiré aléatoirement. Les vecteurs de caractéristiques $\mathbf{V}_t$ sont calculés, et le taux de classification est calculé avec les paramètres de SVM appris précédemment. Nous n'affichons pas les résultats, car ils sont tous strictement similaires à la Table 1. Ces résultats prouvent effectivement la consistance par dilatation obtenue par la normalisation des vecteurs de caractéristiques.

Dans un deuxième temps, nous testons si les fonctions de groupement génèrent bien la consistance par translation. Pour chaque signal de $\boldsymbol{Y}_t$, nous appliquons un décalage temporel (complétion de zéros) aléatoire, tiré uniformément sur un intervalle de taille $N_Z = 10, 20, 30, 40$ et 50. Les signaux étant de longueur moyenne $N = 250$, un décalage de 50 échantillons représente à peu près $1/5$ du signal. Les vecteurs de caractéristiques $\mathbf{V}_t$ sont calculés, et le taux de classification $\eta_{consist}$ est calculé avec les paramètres de SVM appris précédemment.

Les résultats pour les fonctions de groupement non fenêtrées sont identiques à ceux de la Table 1 quelle que soit la valeur de N_Z, ce qui est dû au fait que le temps n'intervient pas Eq. (1).

1. La fonction de groupement définie comme : $\log \sum_k \exp\left(x_{l_p^k, \tau_p^k} \right)$ et appelée *log-mixture* [4] n'est pas comparée ici à cause de ses mauvais résultats.

Pour les fonctions fenêtrées, la Fig 2 résume les taux de classification $\bar{\eta}_{consist}$ de ce deuxième test. Nous constatons que les taux se dégradent d'autant plus que les signaux sont décalés. La disposition des fenêtres introduite dans [8] n'est donc pas strictement consistante par translation.

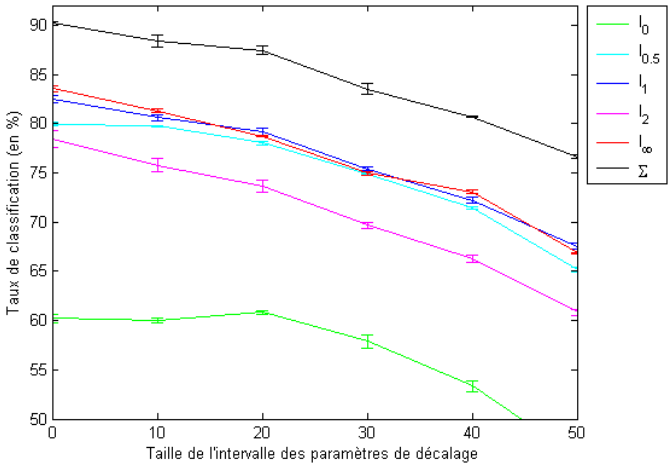

FIGURE 2 – Taux de classification $\bar{\eta}_{consist}$ sur les vecteurs de caractéristiques extraits des signaux translatés aléatoirement pour les fonctions de groupement fenêtrées.

6 Nouveau fenêtrage

Le problème constaté vient de la définition de la taille T_H des fenêtres de Hanning qui dépend de N_{max} défini dans l'Eq. (2). La valeur N_{max} est sensible à une translation temporelle des signaux car elle ne prend pas en compte le début du support du spikegramme. Dans cet article, nous proposons de calculer la taille de cette fenêtre à partir de l'amplitude temporelle maximale :

$$A_{max} = \max_p \left\{ \max_k \left\{\tau_p^k\right\}_{k=1}^K - \min_k \left\{\tau_p^k\right\}_{k=1}^K + 1 \right\}_{p=1}^P .$$

La taille T_H de la fenêtre de Hanning, définie comme $T_H = \frac{A_{max}}{0.5 \times F}$, est donc maintenant indépendante du découpage temporel des signaux.

Les nouvelles fonctions de groupement sont utilisées pour l'apprentissage de SVM comme décrit en Section 4, et avec les mêmes permutations. Les taux de classification $\bar{\eta}$ sont résumés dans la Table 1 (3ième ligne). Nous remarquons que ces taux de classification sont meilleurs que ceux de la Table 1 (1ère et 2ième ligne). Cette légère différence est due au fait que les signaux originaux de la base UCI *Character Trajectories* ne sont

pas tous recalés exactement de la même façon, *i.e.* découpés au même échantillon. Ce léger problème de recalage est aussi constaté dans [12]. Ainsi, une bonne disposition des fenêtres permet de s'affranchir de cet inconvénient lié à l'acquisition de données temporelles. Nous effectuons aussi le test de consistance par translation temporelle, et les taux de classification $\bar{\eta}_{consist}$ sont tous identiques à ceux de la Table 1, quelle que soit la valeur de N_Z. Les taux de classification sont inchangés quand les signaux sont translatés, ce qui prouve bien la consistance par translation du nouveau fenêtrage ayant engendré ces fonctions de groupement.

Ce test est aussi réalisé avec un réseau de neurones convolutifs [13], *Convolutional Neural Network* (CNN) en anglais, qui est la méthode de référence pour la classification de signaux temporels. C'est un classifieur discriminatif optimisé entièrement grâce à une fonctionnelle de classification. Les résultats obtenus sont tracés en Fig. 3 et le taux de classification est de $98.85 \pm 0.33\%$ pour $N_Z = 0$, ce qui est l'état de l'art en matière de classification de signaux temporels. Cependant, nous observons que les performances se dégradent très vite quand les signaux sont translatés. Ce phénomène vient du fait que l'apprentissage du CNN est fait seulement sur des signaux non translatés. Dans ce test de consistance, les signaux sont translatés. Si les fonctions de groupement sont effectivement consistantes par translation, il n'y a pas de modification de la distribution statistique des vecteurs de caractéristiques : les performances du classifieur restent donc identiques. Par contre, quand le classifieur est appliqué sur les signaux bruts comme pour le CNN, la distribution des signaux est modifiée par les translations, et les performances du classifieur se dégradent. Ainsi, le CNN réclame une segmentation des signaux qui soit toujours identique. Un décalage trop important dans la découpe initiale d'un signal met en défaut le CNN.

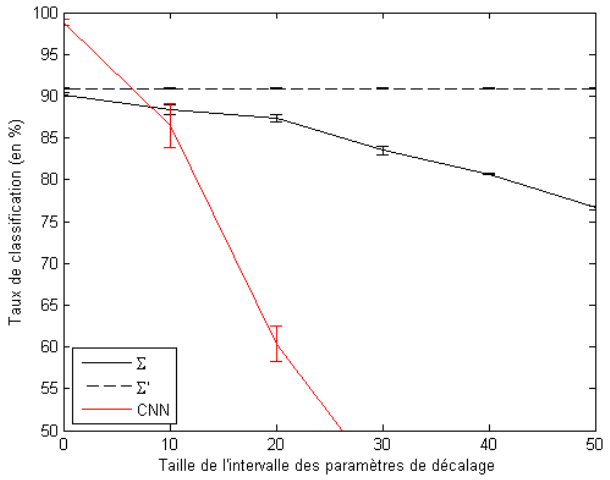

FIGURE 3 – Taux de classification $\bar{\eta}_{consist}$ sur les vecteurs de caractéristiques extraits des signaux translatés aléatoirement par les fonctions de groupement $\sum(.)$ et $\sum'(.)$, et pour le CNN.

7 Conclusion

Après une revue des différents descripteurs adaptés aux décompositions invariantes par translation, nous les avons comparés entre eux. Le nouveau fenêtrage introduit dépasse l'état de l'art et est pleinement robuste à la translation des signaux. Par ailleurs, certaines décompositions parcimonieuses incluent des contraintes de discrimination pour améliorer la classification. Nos perspectives sont d'étendre cette approche au cas invariant par translation en utilisant des fonctions de groupement adéquates. Une autre perspective est de regarder l'influence de la fenêtre et de tester des fenêtres de type triangle, Hamming, etc.

Références

[1] T. Blumensath. *Bayesian Modelling of Music : Algorithmic Advances and Experimental Studies of Shift-Invariant Sparse Coding*. PhD thesis, University of London, 2005.

[2] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J.I. Mars. Shift & 2D rotation invariant sparse coding for multivariate signals. *IEEE Trans. on Signal Processing*, 60 :1597–1611, 2012.

[3] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proc. IEEE*, 98 :948–958, 2010.

[4] Y. LeCun. Learning invariant feature hierarchies. In *Computer Vision – ECCV 2012*, volume 7583 of *LNCS*, pages 496–505, 2012.

[5] H. Grosse, R. Raina, R. Kwong, and A.Y. Ng. Shift-invariant sparse coding for audio classification. In *Proc. Conf. Uncertainty in Artificial Intelligence UAI*, pages 149–158, 2007.

[6] S. Scholler and H. Purwins. Sparse approximations for drum sound classification. *IEEE Journal of Selected Topics in Signal Processing*, 5 :933–940, 2011.

[7] P.-S. Huang, J. Yang, M. Hasegawa-Johnson, F. Liang, and T.S. Huang. Pooling robust shift-invariant sparse representations of acoustic signals. In *Conf. Int. Speech Communication Association, Interspeech 2012*, 2012.

[8] A. Mayoue, Q. Barthélemy, S. Onis, and A. Larue. Preprocessing for classification of sparse data : Application to trajectory recognition. In *Proc. IEEE Workshop SSP '12*, pages 37–40, 2012.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer New York Inc., 2009.

[10] A. Frank and A. Asuncion. UCI machine learning repository. http ://archive.ics.uci.edu/ml, 2010.

[11] Q. Barthélemy. *Représentations parcimonieuses pour les signaux multivariés*. PhD thesis, Université de Grenoble, 2013.

[12] C. Vollmer, J.P. Eggert, and H.-M. Gross. Modeling human motion trajectories by sparse activation of motion primitives learned from unpartitioned data. In *KI 2012 : Advances in Artificial Intelligence*, LNCS, pages 168–179, 2012.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86 :2278–2324, 1998.