Time warp invariant dictionary learning for time series clustering: application to music data stream analysis

Saeed Varasteh Yazdi[1], Ahlame Douzal-Chouakria[1], Patrick Gallinari[2], and Manuel Moussallam[3]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, Grenoble, France
Saeed.Varasteh@imag.fr, Ahlame.Douzal@imag.fr,
[2] Université Pierre et Marie Curie, Paris, France
Patrick.Gallinari@lip6.fr
[3] Deezer, Paris, France
Manuel.Moussallam@deezer.com

Abstract. This work proposes a time warp invariant sparse coding and dictionary learning framework for time series clustering, where both input samples and atoms define time series of different lengths that involve variable delays. For that, first an l_0 sparse coding problem is formalised and a time warp invariant orthogonal matching pursuit based on a new cosine maximisation time warp operator is proposed. A dictionary learning under time warp is then formalised and a gradient descent solution is developed. Lastly, a time series clustering based on the time warp sparse coding and dictionary learning is presented. The proposed approach is evaluated and compared to major alternative methods on several public datasets, with an application to DEEZER music data stream clustering.

Keywords: Time series clustering, dictionary learning, sparse coding

1 Introduction

Sparse coding and dictionary learning become popular methods in machine learning and pattern recognition for a variety of tasks as feature extraction, reconstruction and classification. The aim of sparse coding methods is to represent input samples as a linear combination of few basis functions called *atoms* composing a given dictionary. Sparse coding problem is formalised basically as an optimisation problem that minimises the error of the reconstruction under l_0 or l_1 sparsity constraints. The l_0 constraint leads to a non convex and NP-hard problem, that can be solved efficiently by pursuit methods such as orthogonal matching pursuit OMP [15]. Relaxing the sparsity constraint from l_0 to l_1 norm yields a convex sparse coding problem, also known as LASSO problem [14]. The dictionary for the sparse representation can be selected among pre-specified family of basis functions (e.g. gabor) or be learned from the training data to sparse represent the input data. A two-step strategy is commonly used: 1) keep the

dictionary fixed and find a sparse representation by using a sparse approximation, 2) keep the representation fixed and update the dictionary, either all the atoms at once [6] or one atom at a time [2]. In the context of classification, the dictionary is generally learned from the labeled training dataset and the sparse codes of the testing samples are used for their classification [9,10,17]. In clustering setting, we distinguish two principal sparse coding and dictionary learning approaches. The first category of approaches assumes the data structured into a union of subspaces [3,5,16,18] where each sample may be represented as a linear combination of some input samples that ideally belong to the same subspace. Several of these approaches are related to sparse subspace clustering where samples are first sparse coded based on the input set as a dictionary [5,18], then a spectral clustering [11] of the sparse codes is used to cluster the data. The number of subspaces as well as their dimension may be fixed beforehand or induced from the affinity graph. In the second category of approaches, a sparse coding and dictionary learning framework is proposed to simultaneously learn a set of dictionaries, one for each cluster, to sparse represent and cluster the data [12].

For temporal data analysis, sparse coding and dictionary learning are especially effective to extract class specific latent temporal features, reveal salient primitives and sparsely represent complex temporal features. However, what makes temporal data particularly challenging is that salient events may arise with varying delays, be related to a part of observations that may appear at different time stamps. This work addresses the problem of time series clustering under sparse coding and dictionary learning framework, where both input samples and atoms define time series that may involve varying delays and be of different lengths. For that, in the first part, an l_0 sparse coding problem is formalised and a time warp invariant orthogonal matching pursuit based on a new cosine maximisation time warp operator is proposed. Subsequently, the dictionary learning under time warp is formalised and a gradient descent solution is developed. In the second part, a time series clustering approach based on the time warp invariant sparse coding and dictionary learning is proposed. The main contributions of the paper are:

1. We propose a time series clustering approach under sparse coding and dictionary learning setting.
2. We propose a tractable solution for time warp invariant orthogonal matching pursuit based on a new cosine maximisation time warp operator.
3. We provide a sparse representation of the clustered time series and learn, for each cluster, a sub-dictionary composed of the most discriminative primitives.
4. We conduct experiments on several public and real datasets to compare the proposed approach to the major alternative approaches, with an application to DEEZER music data stream clustering.

The reminder of the paper is organised as follows. Section 2 formalises the time series clustering problem under sparse coding and dictionary learning setting. Section 3 proposes a solution for sparse coding and dictionary learning under time warp, then presents the time series clustering method. Finally, Section 4

presents the experiments and discusses the results obtained.

2 Problem statement

This Section formalises the time warp invariant sparse coding and dictionary learning for time series clustering. In the following, bold lower case letters are used for vectors and upper case letters for matrices. Let $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$ be a set of N input time series $\boldsymbol{x}_i = (x_{i1}, ..., x_{iq_i})^t \in \mathbb{R}^{q_i}$ of length q_i. We formalise the problem of time series clustering under the sparse coding and dictionary learning setting as the estimation of: a) the partition $C = \{C_l\}_{l=1}^{K}$ of X into K clusters and b) the K sub-dictionaries $\{D_l\}_{l=1}^{K}$, to minimise the inertia goodness criterion (*i.e.*, the error of reconstruction) as:

$$\underset{C,D}{min} \sum_{l=1}^{K} \sum_{\boldsymbol{x}_i \in C_l} E(\boldsymbol{x}_i, D_l) \tag{1}$$

where $D_l = \{\boldsymbol{d}_j^l\}_{j=1}^{K_l}$ the sub-dictionary of C_l is composed of K_l time series atoms $\boldsymbol{d}_j^l \in \mathbb{R}^{p_j}$. Note that, both input samples $\boldsymbol{x}_i$ and atoms $\boldsymbol{d}_j^l$ define time series of different lengths that may involve varying delays. $E(\boldsymbol{x}_i, D_l)$ the error of reconstruction, under time warp, of $\boldsymbol{x}_i$ based on the sub dictionary $D_l = \{\boldsymbol{d}_j^l\}_{j=1}^{K_l}$ is formalised as:

$$E(\boldsymbol{x}_i, D_l) = \underset{\boldsymbol{\alpha}_i}{min} \|\boldsymbol{x}_i - \mathcal{F}_i(D_l)\boldsymbol{\alpha}_i^l\|_2^2 \quad \text{s.t.} \|\boldsymbol{\alpha}_i^l\|_0 \leq \tau. \tag{2}$$

where $\mathcal{F}_i(D_l) = [f_i(\boldsymbol{d}_1^l), ..., f_i(\boldsymbol{d}_{K_l}^l)] \in \mathbb{R}^{q_i \times K_l}$ is the transformation of D_l to a new dictionary composed of warped atoms $f_i(\boldsymbol{d}_j^l) \in \mathbb{R}^{q_i}$ aligned to $\boldsymbol{x}_i$ to resorb the involved delays w.r.t $\boldsymbol{x}_i$. $\boldsymbol{\alpha}_i^l = (\alpha_{1i}^l, ..., \alpha_{K_l i}^l)^t$ is the sparse codes of $\boldsymbol{x}_i$ under D_l and τ the sparsity factor under the l_0 norm.

3 Proposed solution

To resolve the clustering problem defined in Eq. 1, we use a two steps iterative refinement process, as in standard $k$means clustering. In the *cluster assignment* step, D_l's are assumed fixed and the problem remains to resolve the sparse coding based on the warped dictionary $\mathcal{F}_i(D_l)$ defined in Eq. 2. The cluster assignment is then obtained by assigning each $\boldsymbol{x}_i$ to the cluster C_l whose sub dictionary D_l minimises the reconstruction error. In the *dictionary update* step, the learned sparse codes and the clusters C_l are that time fixed and the problem in Eq. 1 defines a dictionary learning problem to minimise the clustering inertia criterion and represent sparsely samples within clusters. For the *cluster assignment*, we propose in Section 3.1, a time warp invariant orthogonal matching pursuit based on a new cosine maximisation time warp operator. In Section 3.2, a gradient descent solution for dictionary learning under time warp is developed, then the clustering algorithm for time series under sparse coding and dictionary learning setting is given in Section 3.3.

3.1 Time warp invariant sparse coding

For the sparse coding under time warp problem given in Eq. 2, we define $\mathcal{F}_i(D_l)$ as a linear transformation of D_l based on the warping function $f_i(\boldsymbol{d}_j^l) = \Delta_{ij}^l \boldsymbol{d}_j^l$, where the projector $\Delta_{ij}^l \in \{0,1\}^{q_i \times p_j}$ specifies the temporal alignment that resorbs the delays between $\boldsymbol{x}_i$ and $\boldsymbol{d}_j^l$. The problem given in Eq. 2 is then formalised as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_i,\, \Delta_i^l} & \|\boldsymbol{x}_i - \sum_{j=1}^{K_l} \Delta_{ij}^l \, \boldsymbol{d}_j^l \, \alpha_{ji}^l\|_2^2 \\ \text{s.t. } & \|\boldsymbol{\alpha}_i^l\|_0 \leq \tau,\ \Delta_{ij}^l \in \{0,1\}^{q_i \times p_j},\ \Delta_{ij}^l \mathbf{1}_{p_j} = \mathbf{1}_{q_i}. \end{aligned} \tag{3}$$

with $\Delta_i^l = \{\Delta_{ij}^l\}_{j=1}^{K_l}$. The last constraint is a row normalisation of the estimated Δ_{ij}^l to ensure for $\boldsymbol{x}_i$ equally weighted time stamps. To resolve this problem, we propose an extended variant of OMP that can be mainly summarised in the following steps:

1. For each $\boldsymbol{d}_j^l$, estimate Δ_{ij}^l by dynamic programming to maximise the cosine between $\boldsymbol{x}_i$ and $\boldsymbol{d}_j^l$.
2. Use the projector Δ_{ij}^l to align $\boldsymbol{d}_j^l$ to $\boldsymbol{x}_i$.
3. Estimate the sparse codes $\boldsymbol{\alpha}_i^l$ based on the aligned atoms.

For that and to estimate Δ_i^l, we propose a new operator COSTW to estimate the cosine between two time series under time warp. To the best of our knowledge, it is the first time that the cosine operator is generalised to time series under time warp. Then, we present a time warp invariant OMP (TWI-OMP), that extends the standard OMP approach to sparse code time series under non linear time warping transformations.

Cosine maximisation time warp (COSTW): The problem of estimating the cosine between two time series comes to find an alignment between two time series that maximises their cosine. Let $\boldsymbol{x} = (x_1, ..., x_{q_x})$, $\boldsymbol{y} = (y_1, ..., y_{q_y})$ be two time series of length q_x and q_y. An alignment $\boldsymbol{\pi}$ of length $|\boldsymbol{\pi}| = m$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as the set of m increasing couples:

$$\boldsymbol{\pi} = ((\pi_1(1), \pi_2(1)), (\pi_1(2), \pi_2(2)), ..., (\pi_1(m), \pi_2(m)))$$

where the applications π_1 and π_2 defined from $\{1, ..., m\}$ to $\{1, .., q_x\}$ and $\{1, .., q_y\}$ respectively obey the following boundary and monotonicity conditions:

$$\begin{aligned} 1 = \pi_1(1) \leq \pi_1(2) \leq ... \leq \pi_1(m) = q_x \\ 1 = \pi_2(1) \leq \pi_2(2) \leq ... \leq \pi_2(m) = q_y \end{aligned}$$

and $\forall\, l \in \{1, ..., m\}$,

$$\begin{aligned} \pi_1(l+1) \leq \pi_1(l) + 1\ ,\ \pi_2(l+1) \leq \pi_2(l) + 1, \\ (\pi_1(l+1) - \pi_1(l))\ +\ (\pi_2(l+1) - \pi_2(l)) \geq 1 \end{aligned}$$

Algorithm 1 *MaxTriplet*(u, v, z)

Input: u, v and z.
if $f(u) \geq f(v)$ and $f(u) \geq f(z)$ **then**
  return u;
else if $f(v) \geq f(u)$ and $f(v) \geq f(z)$ **then**
  return v;
else
  return z;
end if

Intuitively, an alignment $\boldsymbol{\pi}$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ describes a way to associate each element of $\boldsymbol{x}$ to one or more elements of $\boldsymbol{y}$ and vice-versa. Such an alignment can be conveniently represented by a path in the $q_x \times q_y$ grid, where the above monotonicity conditions ensure that the path is neither going back nor jumping. We will denote $\mathcal{A}$ as the set of all alignments between two time series. The cosine maximisation time warp can be formalised as:

$$\begin{aligned} \textsc{costw}(\boldsymbol{x}, \boldsymbol{y}) &= s(\boldsymbol{\pi}^*) \\ \boldsymbol{\pi}^* &= \arg\max_{\boldsymbol{\pi} \in \mathcal{A}} s(\boldsymbol{\pi}) \\ s(\boldsymbol{\pi}) &= \frac{\sum_{i=1}^{|\boldsymbol{\pi}|} x_{\pi_1(i)}\, y_{\pi_2(i)}}{\sqrt{\sum_{i=1}^{|\boldsymbol{\pi}|} x_{\pi_1(i)}^2}\sqrt{\sum_{i=1}^{|\boldsymbol{\pi}|} y_{\pi_2(i)}^2}} \end{aligned} \tag{4}$$

where $s(\boldsymbol{\pi})$ is the cost function of the alignment $\boldsymbol{\pi}$. The solution for COSTW is obtained by dynamic programming thanks to the recurrence relation detailed here after.

Let $\boldsymbol{x}_{q_x-1} = (x_1, ..., x_{q_x-1})$, $\boldsymbol{y}_{q_y-1} = (y_1, ..., y_{q_y-1})$ be two sub-time series composed of the $q_x - 1$ and $q_y - 1$ first elements of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. In the case of aligned time series, that do not include delays and with the same length (*i.e.*, $q_x = q_y$) the following incremental property of the standard cosine can be established:

$$\begin{aligned} \textsc{cos}(\boldsymbol{x}_{q_x-1}, \boldsymbol{y}_{q_y-1}) &= f(< \boldsymbol{x}_{q_x-1}, \boldsymbol{y}_{q_y-1} >, \|\boldsymbol{x}_{q_x-1}\|^2, \|\boldsymbol{y}_{q_y-1}\|^2) \\ \textsc{cos}(\boldsymbol{x}, \boldsymbol{y}) &= f((< \boldsymbol{x}_{q_x-1}, \boldsymbol{y}_{q_y-1} >, \|\boldsymbol{x}_{q_x-1}\|^2, \|\boldsymbol{y}_{q_y-1}\|^2) \oplus (x_{q_x}, y_{q_y})) \end{aligned} \tag{5}$$

where f is a real function defined as $f(a, b, c) = \frac{a}{\sqrt{b}\sqrt{c}}$ with $(b, c \in \mathbb{R}_+^*)$ and $\oplus$ is an operator that associates to a triplet (a, b, c) and a couple (u, v) a new triplet as:

$$(a, b, c) \oplus (u, v) = (a + uv, b + u^2, c + v^2)$$

For time series including delays and based on the incremental property given in Eq. 5, let us introduce the computation and recurrence relation that allows to estimate the alignment $\boldsymbol{\pi}^*$ that maximises $\textsc{costw}(\boldsymbol{x}, \boldsymbol{y})$ in Eq. 4.

Computation and recurrence relation: Let us define $M \in \mathbb{R}^{q_x \times q_y}$ the matrix mapping $\boldsymbol{x}$ and $\boldsymbol{y}$ of general term $M_{i,j} = (a_{i,j}, b_{i,j}, c_{i,j})$. Based on the

Algorithm 2 TWI-OMP($\boldsymbol{x}$, D, τ)

Input: $\boldsymbol{x}$, $D = \{\boldsymbol{d}_j\}_{j=1}^{K}$, τ
Output: $\boldsymbol{\alpha}$, Δ
1: $\boldsymbol{r} = \boldsymbol{x}, \Omega = \{\phi\}$
2: **while** $|\Omega| \leq \tau$ **do**
3: Select the atom $\boldsymbol{d}_j$ ($j \notin \Omega$) that maximizes $|\text{COSTW}(\boldsymbol{r}, \boldsymbol{d}_j)|$
4: Update the set of selected atoms $\Omega = \Omega \cup \{j\}$ and $S_\Omega = [\Delta_j \boldsymbol{d}_j]_{j \in \Omega}$
5: Update the coefficients: $\boldsymbol{\alpha_\Omega} = (S_\Omega^T S_\Omega)^{-1}(S_\Omega^T \boldsymbol{x})$
6: Estimate the residual: $\boldsymbol{r} = \boldsymbol{x} - S_\Omega \boldsymbol{\alpha_\Omega}$
7: **end while**

incremental property established in Eq. 5, computing recursively for $(i, j) \in \{1, ..., q_x\} \times \{1, ..., q_y\}$ the terms $M_{i,j}$ as:

$$\forall\ i \geq 2, j = 1\ M_{i,1} = (a_{i-1,1}, b_{i-1,1}, c_{i-1,1}) \oplus (x_i, y_1)$$
$$\forall\ j \geq 2, i = 1\ M_{1,j} = (a_{1,j-1}, b_{1,j-1}, c_{1,j-1}) \oplus (x_1, y_j)$$

and $\forall\ i \geq 2, j \geq 2$

$$M_{i,j} = MaxTriplet \begin{cases} (a_{i,j-1}, b_{i,j-1}, c_{i,j-1}) \oplus (x_i, y_j) \\ (a_{i-1,j}, b_{i-1,j}, c_{i-1,j}) \oplus (x_i, y_j) \\ (a_{i-1,j-1}, b_{i-1,j-1}, c_{i-1,j-1}) \oplus (x_i, y_j) \end{cases}$$

and $M_{1,1} = (x_1 y_1, x_1^2, y_1^2)$, we obtain $\text{COSTW}(\boldsymbol{x}, \boldsymbol{y}) = f(M_{q_x,q_y})$ with a quadratic complexity of $O(q_x q_y)$. The two first equations give the first row and column updates, the third equation gives the recurrence formula that ensures the cosine maximisation at each $M_{i,j}$ cell and $MaxTriplet$ function (Algorithm 1) retains the triplet that maximises the cosine at $M_{i,j}$.

Time warp invariant OMP (TWI-OMP): Based on the defined COSTW, let us present the time warp invariant OMP (TWI-OMP) to sparse code a given time series $\boldsymbol{x}$ based on a dictionary $D = \{\boldsymbol{d}_j\}_{j=1}^{K}$ under time warp conditions (Algorithm 2). The proposed TWI-OMP follows the three steps given in the previous section. First, perform a COSTW between $\boldsymbol{x}$ and each $\boldsymbol{d}_j$ to estimate $\Delta = \{\Delta_j\}_{j=1}^{K}$ and find the atom $\boldsymbol{d}_j$ that maximises $\text{COSTW}(\boldsymbol{x}, \boldsymbol{d}_j)$ (line 3 in Algorithm 2). Then, update the set Ω of the yet selected projected atoms and the dictionary $S_\Omega = [\Delta_j \boldsymbol{d}_j]_{j \in \Omega}$ of the yet selected warped atoms (line 4). The updated S_Ω is then used to estimate the sparse coefficients of $\boldsymbol{x}$ (line 5-6). The process is reiterated on the residuals of $\boldsymbol{x}$ until the sparsity factor τ is reached.

3.2 Time warp invariant dictionary learning

For the dictionary learning step, the problem in Eq. 1 becomes to learn the dictionary D under time warp where, that time, the sparse codes α_i^l and Δ_i^l are

assumed fixed as:

$$\min_{D} \sum_{l=1}^{K} \sum_{\boldsymbol{x}_i \in C_l} \|\boldsymbol{x}_i - \sum_{j=1}^{K_l} \Delta_{ij}^l \, \boldsymbol{d}_j^l \, \alpha_{ji}^l\|_2^2 \quad \text{s.t.} \|\boldsymbol{d}_j^l\|_2 = 1. \tag{6}$$

This problem is then resolved as K single dictionary learning problems to learn each sub-dictionary D_l that minimises the inertia of the cluster C_l:

$$\mathcal{J}_l = \min_{D_l} \sum_{\boldsymbol{x}_i \in C_l} \|\boldsymbol{x}_i - \sum_{j=1}^{K_l} \Delta_{ij}^l \, \boldsymbol{d}_j^l \, \alpha_{ji}^l\|_2^2 \tag{7}$$

which is equivalent to

$$\begin{aligned} \mathcal{J}_l = & \min_{D_l} \sum_{\boldsymbol{x}_i \in C_l} \sum_{t=1}^{q_i} (x_{it} - \sum_{j=1}^{K_l} \alpha_{ji}^l \sum_{(t,t') \in \boldsymbol{\pi}_{ij}^*} d_{jt'}^l)^2 \\ & \text{s.t.} \ \ \|\boldsymbol{d}_j^l\|_2 = 1 \end{aligned} \tag{8}$$

where x_{it} is the t^{th} time instant of $\boldsymbol{x}_i$ and $\boldsymbol{\pi}_{ij}^*$ denotes the optimal alignment path between $\boldsymbol{x}_i$ and $\boldsymbol{d}_j^l$. To resolve the Eq. 8, we propose a gradient descend method based on the following update rule at iteration m for the atom $\boldsymbol{d}_j^l$:

$$\begin{aligned} d_{jt'}^{l(m+1)} &= d_{jt'}^{l(m)} - \eta^m \frac{\partial \mathcal{J}_l}{\partial d_{jt'}^{l(m)}} \\ \boldsymbol{d}_j^{l(m+1)} &= \frac{\boldsymbol{d}_j^{l(m+1)}}{\|\boldsymbol{d}_j^{l(m+1)}\|_2} \end{aligned} \tag{9}$$

with,

$$\begin{aligned} \frac{\partial \mathcal{J}_l}{\partial d_{jt'}^l} = \sum_{\boldsymbol{x}_i \in C_l} \sum_{t=1}^{q_i} -2\alpha_{ji}^l (x_{it} - \alpha_{ji}^l d_{jt'}^l - \alpha_{ji}^l \sum_{\substack{(t,t'') \in \boldsymbol{\pi}_{ij}^* \\ (t'' \neq t')}} d_{jt''}^l \\ - \sum_{j' \neq j} \alpha_{j'i}^l \sum_{(t,t'') \in \boldsymbol{\pi}_{ij'}^*} d_{j't''}^l) \end{aligned} \tag{10}$$

where η is the learning rate. In the following section, we show how the time warp invariant OMP and dictionary learning are involved for time series clustering.

3.3 Time warp invariant dictionary learning for time series clustering

For time series clustering, the clustering criterion given in Eq. 1 is minimised by an iterative process involving, respectively, time warp invariant sparse coding (TWI-OMP) and dictionary learning for cluster assignments and dictionary

Algorithm 3 TWI-DLCLUST(X, K, τ)

Input: $X = \{\boldsymbol{x}_i\}_{i=1}^{N}$, K, τ.
Output: $\{C_1, ..., C_K\}$, $\{D_1, ..., D_K\}$
1: {*Clustering Initialisation:*}
2: Define the affinity matrix $S \in \mathbb{R}^{N \times N}$ of general term:
3: $s_{ii'} =$COSTW$(\boldsymbol{x}_i, \boldsymbol{x}_{i'})$
4: Apply the affinity propagation (or spectral clustering) to cluster S into
5: K clusters: $C_1, ..., C_K$
6: {*Sub-dictionary initialisation:*}
7: **for** $l = 1, ..., K$ **do**
8: Initialise D_l randomly from C_l
9: **repeat**
10: Sparse code each $\boldsymbol{x}_i \in C_l$: $[\boldsymbol{\alpha}_i^l, \Delta_i^l] =$TWI-OMP$(\boldsymbol{x}_i, D_l, \tau)$
11: Update each $\boldsymbol{d}_j^l \in D_l$ by using Eq. 9 and 10.
12: **until** Convergence (stopping rule)
13: **end for**
14: **repeat**
15: {*Cluster assignment:*}
16: Sparse code each $\boldsymbol{x}_i \in X$ based on each D_l $(l = 1, ..., K)$:
17: $[\boldsymbol{\alpha}_i^l, \Delta_i^l] =$TWI-OMP$(\boldsymbol{x}_i, D_l, \tau)$
18: Assign $\boldsymbol{x}_i$ to the cluster C_l whose D_l minimises $E(\boldsymbol{x}_i, D_l)$:
19: $C_l = \{\boldsymbol{x}_i \,/\, l = \underset{l'}{min} \|\boldsymbol{x}_i - \sum_{l'=1}^{K_{l'}} \Delta_{ij}^{l'} \boldsymbol{d}_j^{l'} \alpha_{ji}^{l'}\|_2^2\}$
20: {*Dictionaries update:*}
21: **for** $l = 1, ..., K$ **do**
22: Update each $\boldsymbol{d}_j^l \in D_l$ by using Eq. 9 and 10.
23: **end for**
24: **until** Convergence (no changes in cluster assignments)

update steps (Algorithm 3). In the initialisation step, a clustering (e.g., spectral clustering, affinity propagation) is performed on the COSTW matrix S to determine an initial partition $\{C_l\}_{l=1}^{K}$ of X (line 1-5). A sparse coding and a dictionary learning are then performed on the samples of each cluster to initialise the sub-dictionaries $\{D_l\}_{l=1}^{K}$ (line 6-13). Based on the initial partition $\{C_l\}_{l=1}^{K}$ and sub-dictionaries $\{D_l\}_{l=1}^{K}$, the cluster assignment step consists to perform a sparse coding of each input sample based on each dictionary D_l, then to assign it the cluster whose dictionary minimises its reconstruction error (line 15-19). Subsequently, in the dictionary update step, the atoms $\boldsymbol{d}_j^l$ of each dictionary are updated by using the formula given in Eq. 9 and 10 (line 20-23).

4 Experimental study

In this section, we evaluate the proposed time series clustering under dictionary learning setting (TWI-DLCLUST) on several synthetic and real datasets, including multivariate and univariate time series, that may involve varying delays and be of different or equal lengths. The proposed TWI-DLCLUST clustering method is

compared to two major alternative approaches, the subspace sparse clustering (SSC) [5] and the Dictionary Learning with Structured Incoherence (DLSI) [12]. For SSC, two variants SSC-BP [5] and SSC-OMP [18] are studied for a sparse coding under l_0 and l_1 norms, where an orthogonal matching pursuit and a basis pursuit methods are used respectively. For DLSI, both sample-based and atom-based affinity matrix initialisations proposed in [12] are studied. The Matlab codes of these methods are available online [4].

4.1 Data description

We have considered in Table 1 two groups of datasets. The first group is composed of the top 12 datasets for which the ground truth clustering is given. The four first datasets are composed of public multivariate time series that have different lengths and involve varying delays. In particular, DIGITS, LOWER, and UPPER datasets give the description of 2-D air-handwritten motion gesture of digits, upper and lower case letters performed on a Nintendo (R) Wii device by several writers [4]. The CHAR-TRAJ dataset gives the 2-dimensional handwritten character trajectory performed on a Wacom tablet by the same user [1]. The ECG-MIT dataset was obtained from the MIT-BIH Arrhythmia [7] database where the heartbeats represented by QRS complexes. The 7 remaining datasets are composed of univariate time series of the same lengths that involve significant delays [8]. The last two datasets are provided by DEEZER [5], the online music streaming service that offers access to the music content of nearly 40 million licensed tracks. DEEZER data, for which we have no ground truth, give the description of streaming data of music albums, randomly selected among 10^5 French user streams and recorded from October 2016 to September 2017. They are composed of univariate time series that give the daily total number of streams per album from its release date to September 2017; this study consider only the streams of a duration ≥ 30 seconds. In particular, DEEZER15 and DEEZER30 are provided for the streams analysis over the crucial early period after the album release date. They give the description of the prefix time series on the early period covering a cumulative number of 10^3 streams (in red in Figure 1). In addition, for the pertinence of the analysis, the prefix time series of length < 7 days are extended to 15 days in DEEZER15 and to 30 days in DEEZER30. Table 1 gives some characteristics of the studied datasets: the size of the clusters when the ground truth is known, the size of the validation and evaluation sets and the length of the time series that may be variable or fixed.

4.2 Validation protocol

For the top 12 datasets in Table 1, for which the ground truth partition is known, the proposed method TWI-DLCLUST as well as the alternative clustering

[4] SSC-OMP: https://goo.gl/E6khsq, SSC-BP: https://goo.gl/719pvx and DLSI: https://goo.gl/X5nZgE.

[5] https://www.deezer.com/fr/

Table 1. Data description

Dataset	Nb. class	Valid. set size	Eval. set size	Length range
DIGITS	10	100	100	29∼218
LOWER	26	130	260	27∼163
UPPER	26	130	260	27∼412
CHAR-TRAJ	20	100	200	109∼205
ECG-MIT	4	40	160	541
CBF	3	30	900	128
FACEFOUR	4	24	88	350
LIGHTNING2	2	60	61	637
LIGHTNING7	7	70	73	319
CC	6	300	300	60
TRACE	4	100	100	275
ECG200	2	100	100	96
DEEZER15	-	-	281	15∼301
DEEZER30	-	-	278	30∼301

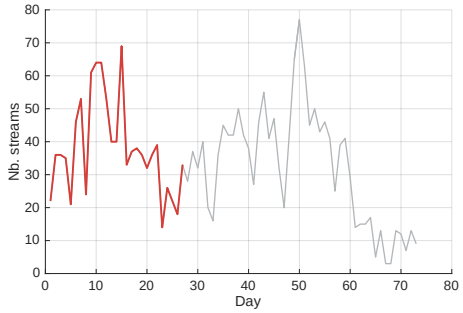

Fig. 1. An album streaming time series, in red the prefix time series covering a cumulative number of 10^3 streams.

approaches are applied to cluster the data. For alternative methods, time series of different lengths are zero padded beforehand. The adjusted Rand index [13] is then used to evaluate the goodness of the obtained clusterings. The Rand index lies between 0 and 1, it measures the agreement between the obtained clusters and the ground truth ones. The higher the index, the better the agreement is. In particular, the maximum value "1" of the Rand index is reached when the obtained partition and the ground truth one are identical. For DEEZER datasets, the ground truth being unknown, a DTW-based within-class W_r ratio [6] is used. The lower the within-class ratio W_r, the better the clustering is. W_r is as well used to select the optimal number of clusters. Finally, the parameters related to each studied method, indicated in Table 2, are learned by line/grid search on the validation set, the best parameters are then used to perform the clustering

[6] $W_r = \frac{\sum_{l=1}^{K} \sum_{\boldsymbol{x},\boldsymbol{y} \in C_l} \mathrm{DTW}(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{x},\boldsymbol{y} \in X} \mathrm{DTW}(\boldsymbol{x},\boldsymbol{y})}$

Table 2. Parameter Line/Grid values

Method	Line / Grid values	Desc.
SSC-OMP	$\tau \in \{1, 2, 3, 4, 5\}$	l_0 sparsity threshold
SSC-BP	$\lambda \in \{0.001, 0.01\}$, lag of 0.01	l_1 sparsity regularisation
DLSI	$\lambda \in \{0.001, 0.01\}$, lag of 0.01	l_1 sparsity regularisation
	$\eta \in \{0, 0.1, 0.01\}$	dictionary incoherence regularisation
	$K_l = 5$, $\forall\, l \in \{1, ..., K\}$	Sub-dictionary D_l size
TWI-DLCLUST	$sc \in [0, 100]$, lag of 10	Sakoe-Chiba band width
	$\tau \in \{1, 2, 3, 4, 5\}$	l_0 sparsity threshold
	$K_l = 5$, $\forall\, l \in \{1, ..., K\}$	Sub-dictionary D_l size

on the evaluation set. The process is iterated over 10 runs and the averaged performances are reported in Tables 3 and 4.

Table 3. Adjusted Rand index

Dataset	SSC-OMP (τ)	SSC-BP	DLSI-S	DLSI-A	TWI-DLCLUST (τ)
DIGITS	0.839 (2)	0.856	0.854	0.841	**0.940** (1)
LOWER	0.935 (3)	0.943	0.937	0.934	**0.970** (1)
UPPER	*0.940* (2)	**0.942**	*0.940*	*0.938*	**0.942** (1)
CHAR-TRAJ	0.947 (5)	*0.977*	**0.978**	*0.971*	*0.965* (3)
ECG-MIT	0.327 (2)	*0.789*	*0.772*	*0.773*	**0.792** (2)
CBF	0.558 (2)	0.668	0.599	0.601	**0.770** (2)
FACEFOUR	**0.810** (5)	0.722	0.767	0.769	0.776 (3)
LIGHTNING2	**0.559** (2)	**0.559**	**0.559**	0.519	**0.559** (2)
LIGHTNING7	*0.793* (2)	*0.808*	0.724	0.747	**0.814** (3)
CC	0.736 (5)	0.630	0.813	0.791	**0.910** (1)
TRACE	0.680 (5)	0.752	0.755	0.753	**0.805** (1)
ECG200	0.547 (4)	0.631	**0.689**	*0.664*	*0.653* (3)
Nb. Best	2	2	3	0	**9**
Avg. Rank	4.00	2.83	2.92	3.58	**1.67**

4.3 Results and discussion

Table 3 gives for the top 12 datasets the obtained adjusted Rand index values. The best values are indicated in bold, the non significantly different ones from the best (t-test at 5% risk) are in italic and the remaining results are significantly different from the bold values. For the two l_0 sparse coding methods SSC-OMP and TWI-DLCLUST, the learned sparsity coefficient τ is given between brackets. The two last rows give, over all the datasets, the number of times a method reaches the best value as well as its average ranking. From Table 3, we can see that the proposed TWI-DLCLUST reaches the best clustering results

Table 4. Within-class ratio W_r per number of clusters K.

Dataset	K	SSC-OMP ($\tau = 5$)	SSC-BP	DLSI-S	TWI-DLCLUST ($\tau = 2$)
DEEZER15	2	*0.266*	0.310	**0.245**	*0.262*
	3	0.201	0.253	0.177	**0.145**
	4	0.212	0.146	*0.114*	**0.112**
	5	0.188	*0.118*	*0.112*	**0.096**
	6	0.133	0.106	*0.074*	**0.069**
DEEZER30	2	0.339	0.348	0.322	**0.303**
	3	0.226	0.292	0.273	**0.173**
	4	0.175	0.241	*0.133*	**0.127**
	5	0.154	*0.119*	*0.110*	**0.096**
	6	*0.100*	**0.080**	*0.085*	*0.085*
Nb. Best		0	1	1	**8**
Avg. Rank		3.40	3.30	2.05	**1.25**

with 9 times (9 out of 12) as the best values, 2 times as significantly non different from the best and obtained the lowest average ranking. The second best results are obtained by SSC-BP and DLSI-S, followed by SSC-OMP. Although the l_1 sparse coding models (here SSC-BP and DLSI-S) are known to be more efficient than the l_0 models, TWI-DLCLUST even involving an l_0 sparse coding leads to the best results. While TWI-DLCLUST and DLSI-S involve smaller size sub-dictionaries ($K_l = 5$), SSC-OMP and SSC-BP are based on larger dictionary of the size of the evaluation set (Table 1). Finally, by comparing the two l_0 sparse coding methods SSC-OMP and TWI-DLCLUST, we can see that TWI-DLCLUST leads for all datasets to sparser solutions with a lower or equal sparsity coefficient τ than SSC-OMP. For DEEZER data we have performed each clustering method for several number of clusters and the within-class ratio of the obtained partitions reported in Table 4. For simplicity, the DLSI approach is conducted only with DLSI-S variant, DLSI-A being highly equivalent in Table 3. We can see easily that, for both datasets and almost all the number of clusters, the best values are reached by TWI-DLCLUST, followed by the l_1 sparse code approaches SSC-BP and DLSI-S, then by SSC-OMP. Finally, note that from both Tables 3 and 4, SSC-OMP and SSC-BP lead to the lowest performances with a slightly better results for SSC-BP as using an l_1 norm sparse coding. These results may be partly explained by the fact that both SSC-OMP and SSC-BP are purely sparse coding methods based on one global dictionary fixed beforehand, unlike DLSI and TWI-DLCLUST that learn one sub-dictionary per cluster.

In the second study, we analyse more closely the obtained clusterings. For instance, based on Figure 2 that displays the progression of the within-class ratio w.r.t the number of clusters, a partitioning into four clusters is performed on DEEZER30. Accordingly, Figure 3, shows for each of the four clusters (each row), the profile of the medoids (in the first column), the closest albums to the medoid in the second column and at the third column, the atom that most contributes to sparse represent the cluster's samples.

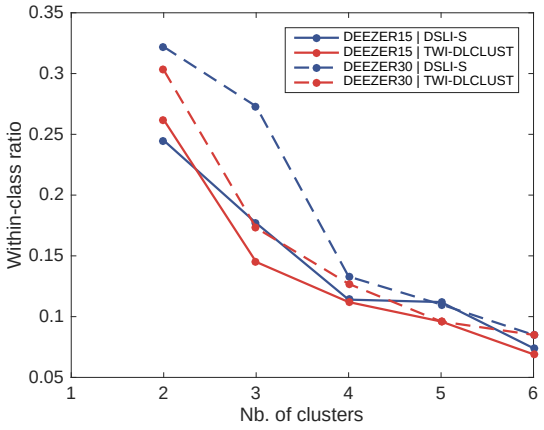

Fig. 2. Number of clusters K vs. Within-class ratio W_r.

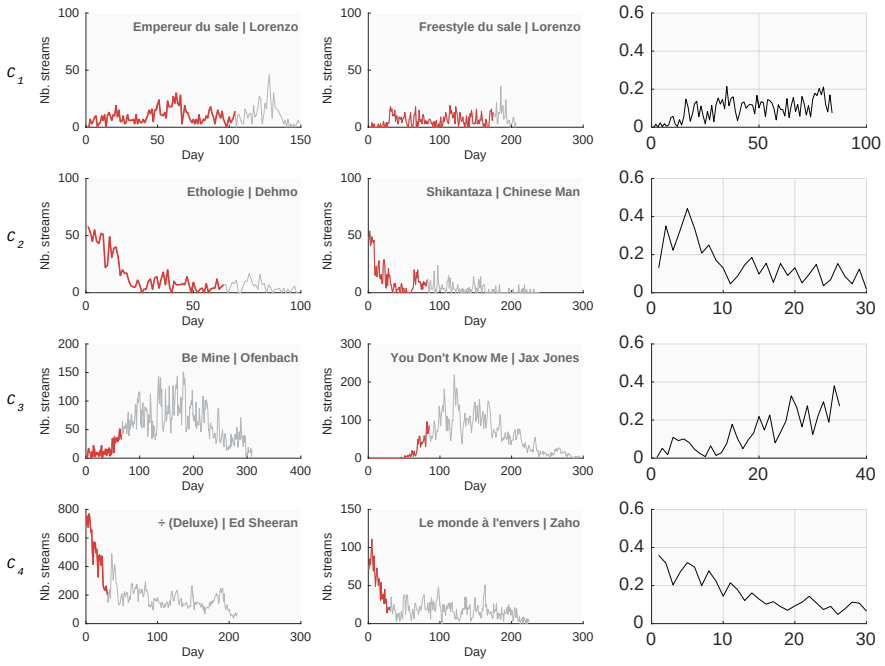

Fig. 3. Four clusters partitioning of DEEZER30: Medoid profile (left column), Nearest album to the medoid (middle column), the most contributing atom to the cluster (right column).

DEEZER data provide additional album descriptive features as "Full" (composed of several tracks) or "Single" (composed of one track), if it is a "Deluxe" edition, namely a re-edition of the album featuring extra contents related to the album, as well as the artist popularity before and after the release. The analysis of some album characteristics brings meaningful interpretation of the extracted clusters (Figure 3).

The first cluster is composed of 71% of "Full" albums and 15% of "Deluxe" editions. It corresponds to album releases with flat stream profiles. Such behaviour usually occurs when the content has already been published ("Deluxe" versions) or for lesser-known artists, as assessed by the cluster medoid "Empereur du Sale" album of the rapper "Lorenzo" that released several singles a few weeks before the album release date and although not highly popular has still a steady fan base.

In the second cluster, 75% of the albums are "Single". The fast decrease stream profile just after the release date is not surprising for short albums (composed of 1-4 tracks). Indeed, a "Full" album is generally released shortly after the "Single" release, inducing a decrease of streams for the "Single" few weeks after its release. The cluster medoid "Ethologie" is produced by the rapper "Dehmo" that has not released albums since a long period, that may explain the burst of streams for the new content just after its release.

The cluster 3 is composed of 69% of "Full" albums mainly produced by artists that became popular after their album release. This is reflected by the stream profiles that initially evolve at low level then increase significantly several days/weeks after the album release. This is confirmed by the medoid album "Be Mine" a single produced by "Ofenbach" that was in fact revealed to the public with that album.

Finally, the cluster 4 comprises a majority of "Single" albums (84%) produced by very popular artists with a huge fan base and immediate success. The medoid album "Divide" produced by "Ed Sheeran" was one of the biggest hits of 2017. Although the stream profiles of the clusters 4 and 2 seem similar, albums of cluster 4 concern more established artists in their second/third albums while cluster 2 is more related to emerging works and first successes.

The aim of the last study is to analyse the pertinence of the learned sub-dictionaries $\{D_1, ..., D_K\}$ for both DLSI and TWI-DLCLUST; the dictionary for the other methods SSC-OMP and SSC-BP is not learned but fixed beforehand. For that, for each method DLSI and TWI-DLCLUST, the atoms of the learned sub-dictionaries are gathered together to built one global dictionary $\cup_{l=1}^{K} D_l$. Let us denote D_{G1} and D_{G2} the global dictionaries obtained for DLSI and TWI-DLCLUST, respectively. Subsequently, the samples in X are sparse coded, by using first an l_1 norm regularisation based on D_{G1}, then a TWI-OMP based on D_{G2}.
For instance, for DEEZER30, Figure 4 shows for the 278 albums the learned sparse codes based on D_{G1} (on left) and on D_{G2} (on right), organised for interpretation purpose per cluster $\{C_1, ..., C_4\}$ and per sub-dictionary $\{D_1, ..., D_4\}$. It emerges from Figure 4, that sparse codes based on D_{G2} highlight clearly a block

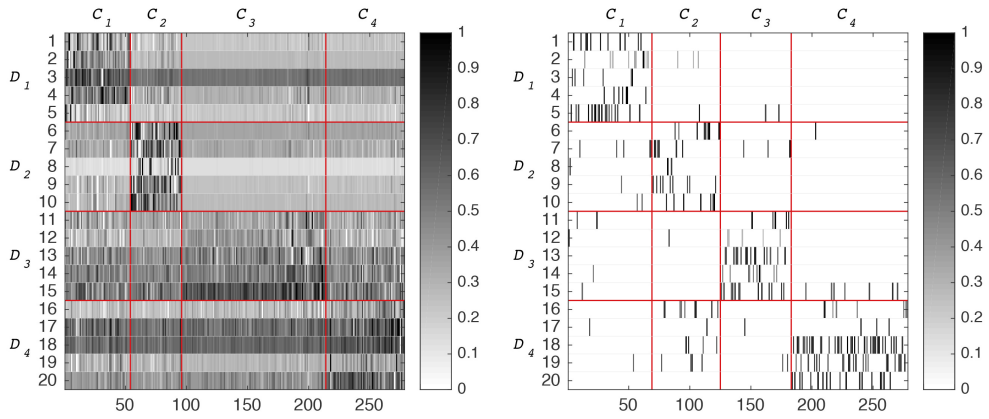

Fig. 4. Sparse representations based on: D_{G1} learned by DLSI (left) and D_{G2} learned by TWI-DLCLUST (right).

structure that reflects the discriminative performance of the sub-dictionaries composing D_{G2} (learned by TWI-DLCLUST). Indeed, sparse codes show that each sub-dictionary D_l is mainly involved to reconstruct samples of C_l. On the other hand, the structure of the sparse codes based on D_{G1} (learned by DLSI) seems much less sparser and less discriminative. We can note, in particular, that the atoms $d_3^1, d_{15}^3, d_{17}^4$ and d_{18}^4 define common primitives involved to reconstruct the samples of all the clusters.

5 Conclusion

This work proposes a time warp invariant sparse coding and dictionary learning for time series clustering where both input samples and atoms define time series that may have different lengths and involve varying delays. For that, first a time warp invariant orthogonal matching pursuit based on a new cosine maximisation time warp operator is proposed. Then, a dictionary learning approach under time warp is formalised and a gradient descent solution is developed. The proposed time series clustering allows to sparse represent the clustered time series and learn, for each cluster, a sub-dictionary composed of the most discriminative primitives. The conducted experiments show that although TWI-DLCLUST involves an l_0 sparse coding approach based on a very small size sub-dictionaries, it leads to the sparser and the best clustering results, while revealing atoms with a good discriminative capacity to represent the time series of each cluster.

Acknowledgment

This work is supported by the French National Research Agency (ANR-Locust project) and Bpifrance funds in the frame of the French National PIA Program.

References

1. A. Frank, A.A.: Uci machine learning repository. http://archive.ics.uci.edu/ml/ (2010), [Online access]
2. Aharon, M., Elad, M., Bruckstein, A.: k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. Signal Processing, IEEE Transactions on 54(11), 4311–4322 (2006)
3. Bradley, P.S., Mangasarian, O.L.: K-plane clustering. Journal of Global Optimization 16(1), 23–32 (2000)
4. Chen, M., AlRegib, G., Juang, B.H.: 6dmg: A new 6d motion gesture database. In: Proceedings of the 3rd Multimedia Systems Conference. pp. 83–88. ACM (2012)
5. Elhamifar, E., Vidal, R.: Sparse subspace clustering: Algorithm, theory, and applications. IEEE transactions on pattern analysis and machine intelligence 35(11), 2765–2781 (2013)
6. Engan, K., Aase, S.O., Husoy, J.H.: Method of optimal directions for frame design. In: Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on. vol. 5, pp. 2443–2446. IEEE (1999)
7. Goldberger, A.L., Amaral, L.A., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: Physiobank, physiotoolkit, and physionet. Circulation 101(23), e215–e220 (2000)
8. Keogh, E.: The ucr time series data mining archive. http://www. cs. ucr. edu/~eamonn/ (2006), [Online access]
9. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in neural information processing systems. pp. 801–808 (2007)
10. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Discriminative learned dictionaries for local image analysis. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. pp. 1–8. IEEE (2008)
11. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems. pp. 849–856 (2002)
12. Ramirez, I., Sprechmann, P., Sapiro, G.: Classification and clustering via dictionary learning with structured incoherence and shared features. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3501–3508. IEEE (2010)
13. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66(336), 846–850 (1971)
14. Tibshirani, R.J., et al.: The lasso problem and uniqueness. Electronic Journal of Statistics 7, 1456–1490 (2013)
15. Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. Information Theory, IEEE Transactions on 50(10), 2231–2242 (2004)
16. Tseng, P.: Nearest q-flat to m points. Journal of Optimization Theory and Applications 105(1), 249–252 (2000)
17. Yazdi, S.V., Douzal-Chouakria, A.: Time warp invariant ksvd: Sparse coding and dictionary learning for time series under time warp. Pattern Recognition Letters 112, 1–8 (2018), https://doi.org/10.1016/j.patrec.2018.05.017
18. You, C., Robinson, D., Vidal, R.: Scalable sparse subspace clustering by orthogonal matching pursuit. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3918–3927 (2016)