Order-Preserving Sparse Coding for Sequence Classification

Bingbing Ni[1], Pierre Moulin[2], and Shuicheng Yan[3]

[1] Advanced Digital Sciences Center, Singapore
[2] University of Illinois at Urbana-Champaign, US
[3] National University of Singapore, Singapore
bingbing.ni@adsc.com.sg moulin@ifp.uiuc.edu eleyans@nus.edu.sg

Abstract. In this paper, we investigate order-preserving sparse coding for classifying multi-dimensional sequence data. Such a problem is often tackled by first decomposing the input sequence into individual *frames* and extracting features, then performing sparse coding or other processing for each frame based feature vector independently, and finally aggregating individual responses to classify the input sequence. However, this heuristic approach ignores the underlying temporal order of the input sequence frames, which in turn results in suboptimal discriminative capability. In this work, we introduce a temporal-order-preserving regularizer which aims to preserve the temporal order of the reconstruction coefficients. An efficient Nesterov-type smooth approximation method is developed for optimization of the new regularization criterion, with guaranteed error bounds. Extensive experiments for time series classification on a synthetic dataset, several machine learning benchmarks, and a challenging real-world RGB-D human activity dataset, show that the proposed coding scheme is discriminative and robust, and it outperforms previous art for sequence classification.

1 Introduction

Sparse coding has been successfully used in various computer vision applications [1] [2] [3]. Sparse coding compactly represents objects as a linear combination of a small number of elements of a dictionary, however there often exist *group* structures in basis data (dictionary). To handle such structures, two alternative methods, Elastic Net [4] and group Lasso [5] have been proposed. They favor the selection of few groups of the correlated dictionary samples to represent the testing data. Often we may estimate models from multiple related data sources. For example, in object recognition, we may extract K types of image representations from K different types of features associated with the same visual input. By minimizing the sum of ℓ_2 norms of the blocks of coefficients associated with each covariate group across different feature representations, similar sparsity patterns in all modalities are encouraged [6]. In [7], the objective of joint sparsity is achieved by imposing an $\ell_{2,1}$ mixed-norm penalty on the reconstruction coefficients.

In this work, we are interested in a different problem setting: the input is a sequence of feature vectors instead of a single feature vector, and there exist dependencies among the input feature vectors. An example is a multidimensional

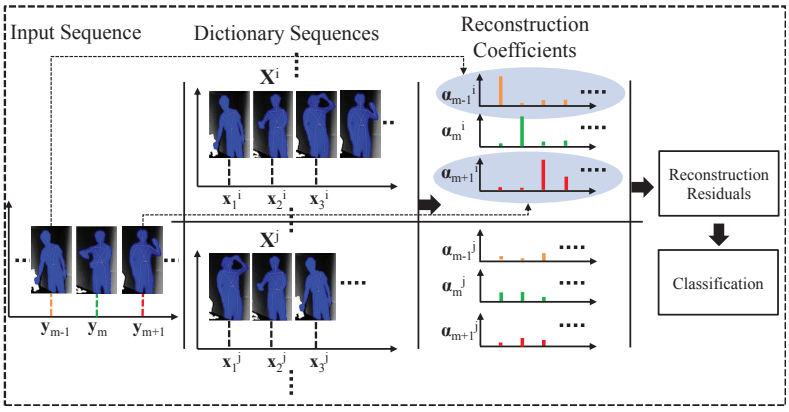

Fig. 1. Motivation of the proposed work, applied to human motion sequence classification. The input sequence receives a strong response from dictionary sequence i as their temporal ordering structures are similar. In contrast, the response from dictionary sequence j is low as its temporal structure is different, although individual frames of the input sequence are similar to those of the sequence j. Similarly to [7], the reconstruction residual is used for classification. See the color pdf for better view.

time series such as audio data, which admit a natural temporal ordering structure across instantaneous feature vectors at successive time stamps. A heuristic solution is to apply sparse coding to the feature vector for each input frame (*i.e.*, time stamp) individually, and then compute and aggregate reconstruction coefficients for individual frames over the entire audio sequence for classification [8]. The temporal structure of the sequence conveys discriminative information. Treating each frame independently would discard this important information. However, this problem has been largely ignored in current sparse coding literature.

We address this problem by developing an order-preserving sparse coding scheme for discriminatively representing multidimensional time series, as illustrated in Figure 1. Our main contribution is to introduce a temporal order preserving regularization scheme. The regularizer penalizes misfit of the temporal order of the reconstruction coefficients for individual frames with respect to the temporal order of the input sequence. To the best of our knowledge, this is the first attempt to address this temporal consistency issue for sparse coding. The resulting optimization problem is convex but nonsmooth. We therefore develop an efficient Nesterov-type smooth approximation method [9] for optimization. Extensive experiments on time series classification over a synthetic dataset, several machine learning benchmarks, and a challenging real world RGB-D human activity dataset, demonstrate that the proposed scheme is discriminative and robust, and outperforms previous art for time series classification.

2 Related Work

Time series classification is an active research topic. Hidden Markov models (HMMs) [10] is the most popular way for sequence classification. A segmental hidden Markov model (HMM) was recently used to characterize waveform

shape for recognition [11]. The limitations of HMM methods are the Markovian assumption and the complexity in training. Dynamic Time Warping (DTW) [12] is also widely used for time series classification. Rodrigues et al. [13] proposed a DTW decision tree method for time series classification. Hayashi et al. [14] used DTW distances to embed time series into a lower dimensional space by Laplacian eigenmap. Xi et al. [15] proposed numerosity reduction to accelerate nearest-neighbor DTW. However, similar to HMM based methods, the computational burden of DTW-based methods is generally high. Another tool for sequence classification is Recurrent Neural Networks (RNN) [16], which is a modification of a general Neural Network architecture that considers temporal structure. However, training the model using back propagation is also complex.

In this work, we present our classification scheme and experimentally compare it with the above methods as well as with multi-resolution symbolic representation [17]. RNN is not used for comparison here because it is generally slow in training and it is less discriminative than other methods. As our method inherits the discriminative capability of conventional sparse coding scheme and further boosts this capability by explicitly encoding temporal ordering structures, it significantly outperforms previous work in classification accuracy. It requires no training and is efficient in testing. Note that *n-gram* model [18] is also used for sequence modeling for speech recognition; however, it lacks explicit representation of long range dependency, and it is very sensitive to temporal scaling. In contrast, our proposed regularization scheme does not have these limitations. The *n-gram* model is therefore not used in our comparisons. Although some works [19] [20] [21] also concern sparse coding for temporal varying signals (events, images), their regularization frameworks only use traditional temporal smoothness constraints. None of these works directly considers temporal ordering.

3 Order-Preserving Sparse Coding

3.1 Notation

We first introduce the notation used in this work. The input is a multi-dimensional time series $Y = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_t]$, where each $\mathbf{y}_i$ is a D-dimensional feature vector, and t denotes the length of the time series. We are given a basis dictionary denoted as $\mathcal{X} = \{(X_1, y_1), (X_2, y_2), \cdots, (X_S, y_S)\}$, where S is the number of dictionary sequences. Each sequence X_j is a time series represented as $X_j = [\mathbf{x}_1^j, \mathbf{x}_2^j, \cdots, \mathbf{x}_{t_j}^j]$, where t_j is the length of the j-th sequence and y_j is the corresponding class label. Each $\mathbf{x}_i^j$ is a D-dimensional feature vector normalized to unit ℓ_2 norm. Note that t_j may vary from sequence to sequence. The label y_j has K possible values in the set $\{1, 2, \cdots, K\}$. If we stack all the sequences in the dictionary one by one with the temporal order within sequence retained, we can represent the dictionary by a $D \times N$ matrix $X = [X_1, X_2, \cdots, X_S]$, where $N = \sum_{j=1}^{S} t_j$ is the total number of feature vectors. Let the N-dimensional vector $\boldsymbol{\alpha}_i$ denote the reconstruction coefficients for the input vector $\mathbf{y}_i$, which is expressed as a linear combination of all dictionary entries. Let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \cdots; \boldsymbol{\alpha}_t)$, and $\boldsymbol{\alpha}_i^j$ be the t_j-dimensional reconstruction

coefficients for input vector i from the dictionary sequence j. Using this convention, we further denote by $\boldsymbol{\alpha}^j = (\boldsymbol{\alpha}_1^j; \boldsymbol{\alpha}_2^j; \cdots; \boldsymbol{\alpha}_t^j)$ the reconstruction coefficients from the dictionary sequence j.

3.2 Temporal-Order-Preserving Regularizer

Sparse coding provides a discriminative way for encoding signals. A straightforward way to represent an input sequence within the sparse coding framework is to first decompose the input sequence into frame-based representations (feature vectors at each time stamp), then perform sparse coding for each frame based feature vector independently, and finally aggregate individual responses (reconstruction coefficients) for classification. However, this heuristic approach is suboptimal because the time series has very strong temporal correlation across individual frames. We refer to this correlation is as a temporal ordering structure. For example, in speech recognition, reordering of a given sequence of phonemes may lead to a totally different sentence and meaning. Ignoring temporal ordering information results in loss of discriminative capability in sparse coding.

Our proposed regularization scheme explicitly addresses this problem in sparsely encoding time series data. As a prerequisite, we require that the reconstruction coefficients for all individual feature vectors of the input sequence should be nonzero on only a few dictionary sequences. As the input sequence normally matches only a few dictionary sequences, spreading the reconstruction coefficients throughout all dictionary sequences degrades the representation discriminative capability. To address this problem, we adopt an $\ell_{2,1}$ norm group-sparsity regularizer as in multitask joint sparse coding [7]. Long sequence can be decomposed into shorter ones for more sparsity. Thus the first term of our regularization criterion is

$$G(\boldsymbol{\alpha}) = \sum_{j=1}^{S} \|\boldsymbol{\alpha}^j\|_2. \tag{1}$$

The second term of our regularization criterion is the temporal order preserving regularizer. The goal is to prevent the reconstruction coefficients of the input feature vector i (at the time stamp i of the input sequence) from the j-th dictionary sequence from being temporally *behind* those of the input vector $i+1$ (at the time stamp $i+1$ of the input sequence). In other words, the nonzero reconstruction coefficients for individual feature vectors obey the same temporal order as the corresponding feature vectors in the input sequence. To encourage this, we consider the expression

$$(\mathbf{w}_j^T(\boldsymbol{\alpha}_i^j - \boldsymbol{\alpha}_{i+1}^j))_+ = \max\{\mathbf{w}_j^T(\boldsymbol{\alpha}_i^j - \boldsymbol{\alpha}_{i+1}^j), 0\}, \tag{2}$$

where $\mathbf{w}_j$ is a vector that has the same length as the corresponding dictionary sequence j and it is element-wise increasing. More specifically, $\mathbf{w}_j$ satisfies the condition: $\mathbf{w}_j(a) < \mathbf{w}_j(b), \forall a < b$, where $\mathbf{w}_j(a)$ denotes the a-th element of vector $\mathbf{w}_j$. With this property, $\mathbf{w}_j^T\boldsymbol{\alpha}_i^j$ approximates the temporal position of the responses for the i-th input vector $\mathbf{y}_i$ from dictionary sequence j. When the sum of all entries in $\boldsymbol{\alpha}_i^j$ is one, the sum $\mathbf{w}_j^T\boldsymbol{\alpha}_i^j$ can be considered as the

approximated temporal position. When the sum of the entries in $\boldsymbol{\alpha}_i^j$ is not equal to one, the sum $\mathbf{w}_j^T \boldsymbol{\alpha}_i^j$ can be considered as the importance weighted version of the approximated temporal position. Therefore $\mathbf{w}_j^T \boldsymbol{\alpha}_i^j > \mathbf{w}_j^T \boldsymbol{\alpha}_{i+1}^j$ means that the approximated temporal position of the response for the $(i+1)$-th input vector on dictionary sequence j precedes that of the i-th input vector, which is the case to penalize. We call $\mathbf{w}_j$ as temporal-structure-prior-multiplier, and in this work we choose a simple form of $\mathbf{w}_j$, *i.e.*, element-wise linear, to reflect the time ordering information as:

$$\mathbf{w}_j = (\frac{1}{t_j}, \frac{2}{t_j}, \cdots, \frac{t_j - 1}{t_j}, 1)^T. \tag{3}$$

The **temporal-order-preserving regularization** term is obtained by summing (2) over all consecutive frames of the input sequence and over all dictionary sequences, namely,

$$P(\boldsymbol{\alpha}) = \sum_{i=1}^{t-1} \sum_{j=1}^{S} \max(\mathbf{w}_j^T \boldsymbol{\alpha}_i^j - \mathbf{w}_j^T \boldsymbol{\alpha}_{i+1}^j, 0). \tag{4}$$

An illustration of the effect of the regularizer (4) is given in Figure 2. Observe that *disordered* reconstruction coefficients receive large penalty, and when reconstruction coefficients are *ordered*, this term vanishes. Note that the inverse is not true, *i.e.*, zero penalty does not necessary indicate temporally-ordered reconstruction.

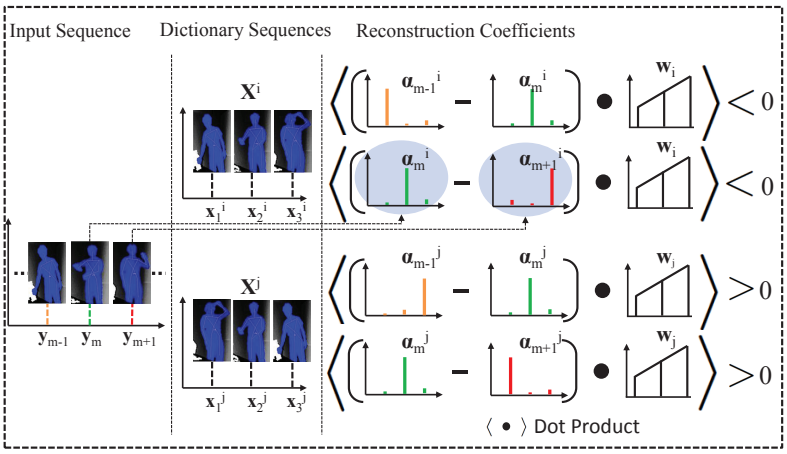

Fig. 2. Effect of applying the regularizer (4). Note that the reconstruction coefficients that follow the temporal order of the input sequence are not penalized by the expression (upper two rows). Those that do not are penalized (bottom two rows). See the color pdf for better view.

3.3 Objective Function

Here we formally state our regularization criterion. The reconstruction error term is

$$f(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{t} \|\mathbf{y}_i - X\boldsymbol{\alpha}_i\|_2^2, \tag{5}$$

which is a convex, smooth, differentiable function with Lipschitz constant $L_f = \|X^T X\|_F$, where $\|.\|_F$ denotes the Frobenius norm.

Recall that the group-sparsity regularizer given in (1) is

$$G(\boldsymbol{\alpha}) = \sum_{j=1}^{S} \|\boldsymbol{\alpha}^j\|_2.$$

This is an $\ell_{2,1}$ mixed-norm, which is convex and nonsmooth. However, $\|\boldsymbol{\alpha}^j\|_2$ can be written as

$$\|\boldsymbol{\alpha}^j\|_2 = \max_{\|\mathbf{v}_j\|_2 \leq 1} \langle \boldsymbol{\alpha}^j, \mathbf{v}_j \rangle. \tag{6}$$

We use the Nesterov smooth approximation method of [9] and approximate (6) by the smooth function:

$$q_{\mu,j}(\boldsymbol{\alpha}^j) = \max_{\|\mathbf{v}_j\|_2 \leq 1} \{\langle \boldsymbol{\alpha}^j, \mathbf{v}_j \rangle - \frac{1}{2}\mu \|\mathbf{v}_j\|_2^2\}, \tag{7}$$

where μ is a parameter that controls the approximation accuracy. The unique minimizer of (7), denoted as $\mathbf{v}_j(\boldsymbol{\alpha}^j)$, can be derived as

$$\mathbf{v}_j(\boldsymbol{\alpha}^j) = \begin{cases} \frac{\boldsymbol{\alpha}^j}{\mu}, & 0 \leq \|\boldsymbol{\alpha}^j\|_2 \leq \mu \ ; \\ \frac{\boldsymbol{\alpha}^j}{\|\boldsymbol{\alpha}^j\|_2}, & \|\boldsymbol{\alpha}^j\|_2 > \mu. \end{cases} \tag{8}$$

The approximation of (1) is therefore obtained as

$$G_\mu(\boldsymbol{\alpha}) = \sum_{j=1}^{S} q_{\mu,j}(\boldsymbol{\alpha}^j). \tag{9}$$

Recall that the temporal-order-preserving regularization function is given by $P(\boldsymbol{\alpha})$ in (4). By simple manipulation, we can rewrite $P(\boldsymbol{\alpha})$ more compactly as:

$$P(\boldsymbol{\alpha}) = \sum_{i=1}^{S\times(t-1)} \|(W_i\boldsymbol{\alpha})_+\|_1, \tag{10}$$

Here each W_i, $i = 1, 2, \cdots, (t-1) \times S$, is an N-dimensional row vector given by $W_{(h-1)\times S+j} = (0, \cdots, 0, \mathbf{w}_j^T, 0, \cdots, 0, -\mathbf{w}_j^T, 0, \cdots, 0)$, for $h = 1, 2, \cdots, t-1$, $j = 1, 2, \cdots, S$. The positions of $\mathbf{w}_j^T$ and $-\mathbf{w}_j^T$ correspond to those of $\boldsymbol{\alpha}_h^j$ and $\boldsymbol{\alpha}_{h+1}^j$ in $\boldsymbol{\alpha}$, respectively. As mentioned above, the regularizer (10) encourages order preserving for the reconstruction coefficients for individual feature vectors

of the input sequence. Again, $(.)_+$ denotes $\max(., 0)$. The function $\|(W_i\boldsymbol{\alpha})_+\|_1$ is convex and nonsmooth. Moreover,

$$\|(W_i\boldsymbol{\alpha})_+\|_1 = \max_{0\leq v_i\leq 1} \langle W_i\boldsymbol{\alpha}, v_i\rangle. \tag{11}$$

By using the Nesterov smooth approximation method of [9], (11) can be approximated by the following smooth function

$$p_{\mu,i}(\boldsymbol{\alpha}) = \max_{0\leq v_i\leq 1} \{\langle W_i\boldsymbol{\alpha}, v_i\rangle - \frac{1}{2}\mu\|v_i\|_2^2\}, \tag{12}$$

where μ is the parameter that controls the approximation accuracy. For fixed $\boldsymbol{\alpha}$, the unique minimizer of (12) can be derived as:

$$v_i(\boldsymbol{\alpha}) = \min\{1, \max(0, \frac{W_i\boldsymbol{\alpha}}{\mu})\}. \tag{13}$$

We then construct a smooth approximation of $P(\boldsymbol{\alpha})$ as:

$$P_\mu(\boldsymbol{\alpha}) = \sum_{i=1}^{S\times(t-1)} p_{\mu,i}(\boldsymbol{\alpha}). \tag{14}$$

The *nonsmoothed* objective function is defined as

$$F(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{t}\|\mathbf{y}_i - X\boldsymbol{\alpha}_i\|_2^2 + \lambda_1\sum_{j=1}^{S}\|\boldsymbol{\alpha}^j\|_2 + \lambda_2\sum_{i=1}^{S\times(t-1)}\|(W_i\boldsymbol{\alpha})_+\|_1. \tag{15}$$

This is a convex but nonsmooth function. According to (9) and (14), $F(\boldsymbol{\alpha})$ can be approximated by the convex and smooth function as:

$$F_\mu(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha}) + \lambda_1 G_\mu(\boldsymbol{\alpha}) + \lambda_2 P_\mu(\boldsymbol{\alpha}). \tag{16}$$

We adopt (16) as our regularization criterion. The gradient of $F_\mu(\boldsymbol{\alpha})$ is $\nabla F_\mu(\boldsymbol{\alpha}) = \nabla f(\boldsymbol{\alpha}) + \lambda_1\nabla G_\mu(\boldsymbol{\alpha}) + \lambda_2\nabla P_\mu(\boldsymbol{\alpha})$, and the corresponding Lipschitz constant is $L_{F_\mu} = L_f + \lambda_1 L_{G_\mu} + \lambda_2 L_{P_\mu}$.

To minimize the objective function $F_\mu(\boldsymbol{\alpha})$, we use the efficient accelerated proximal gradient (APG) method [22], which has the rate of convergence $O(1/n^2)$, where n is the iteration number. In terms of the desired residue ϵ, *i.e.*, $|F_\mu - \min F_\mu| \leq \epsilon$, by choosing $\mu \approx \epsilon$ we have that the rate of convergence is $O(1/\epsilon)$. Algorithm 1 gives the detailed description of the optimization procedure. The regularization parameters λ_1 and λ_2 are chosen by cross-validation on a small validation subset. In particular, we enumerate the value of λ_1 and λ_2 within the set $\{0.1, 1.0, 10, 100, 1000\}$ and select the optimal one by cross-validation.

3.4 Analysis

In this subsection, we analyze the approximations (9) and (14).

Inputs : $X \in \mathbb{R}^{D\times N}$, $\{W_i \in \mathbb{R}^{1\times N}, i = 1, 2, \cdots, S(t-1)\}$, λ_1, λ_2, μ, $\{\mathbf{y}_i, i = 1, \cdots, t\}$.
Output: $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1; \boldsymbol{\alpha}_2; \cdots; \boldsymbol{\alpha}_t) \in \mathbb{R}^{tN}$.
Initialization: Calculate $L_{F_\mu} = L_f + \lambda_1 L_{G_\mu} + \lambda_2 L_{P_\mu}$. Initialize $\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0 \in \mathbb{R}^{tN}$ to be zero vectors, and let $\gamma_0 = 0$, $k = 0$.
repeat
  $\boldsymbol{u}_k = (1-\gamma_k)\boldsymbol{\alpha}_k + \gamma_k \boldsymbol{\beta}_k$,
  Calculate the gradient $\nabla F_\mu(\boldsymbol{u}_k)$.
  $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - \frac{1}{\gamma_k L_{F_\mu}} \nabla F_\mu(\boldsymbol{u}_k)$,
  $\boldsymbol{\alpha}_{k+1} = (1-\gamma_k)\boldsymbol{\alpha}_k + \gamma_k \boldsymbol{\beta}_{k+1}$,
  $\gamma_{k+1} = \frac{2}{k+1}$, $k \leftarrow k+1$.
until *Converged*;

Algorithm 1: Minimization algorithm for (16)

Proposition 1. *$G_\mu(\boldsymbol{\alpha})$ is a μ-accurate approximation to $G(\boldsymbol{\alpha})$, that is*

$$G_\mu(\boldsymbol{\alpha}) \leq G(\boldsymbol{\alpha}) \leq G_\mu(\boldsymbol{\alpha}) + \frac{1}{2}\mu S. \tag{17}$$

Proof. By (7) we have

$$0 \leq q_{\mu,j}(\boldsymbol{\alpha}^j) \leq \max_{0\leq \|\mathbf{v}_j\|_2 \leq 1} \langle \boldsymbol{\alpha}^j, \mathbf{v}_j \rangle = \|\boldsymbol{\alpha}^j\|_2. \tag{18}$$

Summing (18) over j, we obtain the upper bound

$$G_\mu(\boldsymbol{\alpha}) = \sum_{j=1}^{S} q_{\mu,j}(\boldsymbol{\alpha}^j) \leq \sum_{j=1}^{S} \|\boldsymbol{\alpha}^j\|_2 = G(\boldsymbol{\alpha}). \tag{19}$$

Since $0 \leq \|\mathbf{v}_j\|_2 \leq 1$, we also obtain the lower bound

$$q_{\mu,j}(\boldsymbol{\alpha}) \geq \max_{0\leq \|\mathbf{v}_j\|_2 \leq 1} \langle \boldsymbol{\alpha}^j, \mathbf{v}_j \rangle - \frac{1}{2}\mu = \|\boldsymbol{\alpha}^j\|_2 - \frac{1}{2}\mu. \tag{20}$$

Summing (20) over j we obtain

$$G_\mu(\boldsymbol{\alpha}) \geq G(\boldsymbol{\alpha}) - \frac{1}{2}\mu S. \tag{21}$$

Combining (19) and (21) yields (17).

Theorem 1. *The function $G_\mu(\boldsymbol{\alpha})$ is convex and continuously differentiable. Moreover, its gradient $\nabla G_\mu(\boldsymbol{\alpha}) = \sum_{j=1}^{S} \mathbf{v}_j(\boldsymbol{\alpha}^j)$ is Lipschitz continuous with constant $L_{G_\mu} = \frac{tN}{\mu}$.*

Proof. It follows directly from [9] that for $1 \leq j \leq S$, $\nabla q_{\mu,j}(\boldsymbol{\alpha}^j) = \mathbf{v}_j(\boldsymbol{\alpha}^j)$. The function $q_{\mu,j}(\boldsymbol{\alpha}^j)$ is Lipschitz continuous with constant $L_{G_{\mu,j}} = \frac{t\times t_j}{\mu}$. We thus have $L_{G_\mu} = \sum_{j=1}^{S} \frac{t\times t_j}{\mu} = \frac{tN}{\mu}$.

Proposition 2. *$P_\mu(\boldsymbol{\alpha})$ is a μ-accurate approximation to $P(\boldsymbol{\alpha})$, that is*

$$P_\mu(\boldsymbol{\alpha}) \leq P(\boldsymbol{\alpha}) \leq P_\mu(\boldsymbol{\alpha}) + \frac{1}{2}\mu S(t-1). \tag{22}$$

Proof. By definition we have

$$0 \leq p_{\mu,i}(\boldsymbol{\alpha}) \leq \max_{0\leq v_i\leq 1} \langle W_i\boldsymbol{\alpha}, v_i\rangle = \|(W_i\boldsymbol{\alpha})_+\|_1. \tag{23}$$

Summing (23) over i, we obtain

$$P_\mu(\boldsymbol{\alpha}) = \sum_{i=1}^{S(t-1)} p_{\mu,i}(\boldsymbol{\alpha}) \leq \sum_{i=1}^{S(t-1)} \|(W_i\boldsymbol{\alpha})_+\|_1 = P(\boldsymbol{\alpha}). \tag{24}$$

Since $0 \leq v_i \leq 1$, we have

$$p_{\mu,i}(\boldsymbol{\alpha}) \geq \max_{0\leq v_i\leq 1} \langle W_i\boldsymbol{\alpha}, v_i\rangle - \frac{1}{2}\mu = \|(W_i\boldsymbol{\alpha})_+\|_1 - \frac{1}{2}\mu. \tag{25}$$

Summing both sides of (25) over i, we obtain

$$P_\mu(\boldsymbol{\alpha}) \geq P(\boldsymbol{\alpha}) - \frac{1}{2}\mu S(t-1). \tag{26}$$

Combining (24) and (26) yields (22).

Theorem 2. *The function $P_\mu(\boldsymbol{\alpha})$ is convex and continuously differentiable. Moreover, its gradient $\nabla P_\mu(\boldsymbol{\alpha}) = \sum_{i=1}^{S(t-1)} W_i^T v_i(\boldsymbol{\alpha})$ is Lipschitz continuous with constant $L_{P_\mu} = \frac{1}{\mu}\sum_{i=1}^{S(t-1)} \|W_i\|_2^2$.*

Proof. It follows directly from [9] that for $1 \leq i \leq S(t-1)$, $\nabla p_{\mu,i}(\boldsymbol{\alpha}) = W_i^T v_i(\boldsymbol{\alpha})$, and it is Lipschitz continuous with constant $L_{P_{\mu,i}} = \frac{1}{\mu}\|W_i\|_2^2$. We thus obtain $L_{P_\mu} = \frac{1}{\mu}\sum_{i=1}^{S(t-1)} \|W_i\|_2^2$.

3.5 Time Series Classification Rule

We denote by $X^{(j)} = [X_{j_1}, X_{j_2}, \cdots]$ the set of dictionary sequences from the j-th class. Here X_{j_i} denotes the i-th sequence in $X^{(j)}$ that belongs to the j-th class. Let $\boldsymbol{\alpha}^{(j)}$ denote the corresponding reconstruction coefficients for $X^{(j)}$. One can approximate the input sequence Y by using only the optimal coefficients associated with the j-th class as $X^{(j)}\boldsymbol{\alpha}^{(j)}$. According to the classification rule defined in [7], the predicted class label is the one with the lowest total reconstruction error:

$$j_{opt} = \arg\min_j \|Y - X^{(j)}\boldsymbol{\alpha}^{(j)}\|_F^2. \tag{27}$$

4 Experiments

To evaluate the effectiveness of our proposed method, we conduct experiments on multidimensional time series classification on a synthetic dataset, three machine learning benchmarks, and a real-world RGB-D human activity dataset. We also compare the proposed method with other state-of-the-art time series classification algorithms.

4.1 Synthetic Dataset

We first generate eight two-dimensional dictionary sequences $(X_1, X_2, \cdots, X_8)$. They are generated from eight polynomial functions: $f^1(i) = \frac{\pi}{4}$; $f^2(i) = \frac{\pi}{5}$; $f^3(i) = \frac{\pi}{10}$; $f^4(i) = \frac{\pi}{10}i$; $f^5(i) = \frac{\pi}{10}(4-i)$; $f^6(i) = \frac{\pi}{10}(i-2)^2$;$f^7(i) = \frac{2\pi}{5} - \frac{\pi}{10}(i-2)^2$; and $f^8(i) = \frac{\pi}{5}|\sin(\frac{\pi}{2}i)| + \frac{\pi}{10}$. From these functions, we generate eight length-5 two-dimensional time series as

$$X_j = \begin{bmatrix} \sin(f^j(0)) & \sin(f^j(1)) & \cdots & \sin(f^j(4)) \\ \cos(f^j(0)) & \cos(f^j(1)) & \cdots & \cos(f^j(4)) \end{bmatrix}, \quad j = 1, 2, \cdots, 8. \tag{28}$$

For all sequences, we add independent Gaussian noise samples with zero mean and variance 0.2 to both dimensions at each time stamp. We select the fourth sequence from the dictionary as the test time series input, *i.e.*, $Y = X_4$. Each dictionary sequence is labeled as different class. For comparison, we use a) sparse coding (SC) which computes reconstruction coefficients for each frame of the input sequence individually; b) multitask group sparse coding (Group-SC) [7]; and c) the proposed order-preserving sparse coding method (MTO-SC). Reconstruction coefficients are shown in Figure 3. We can observe that 1) reconstruction

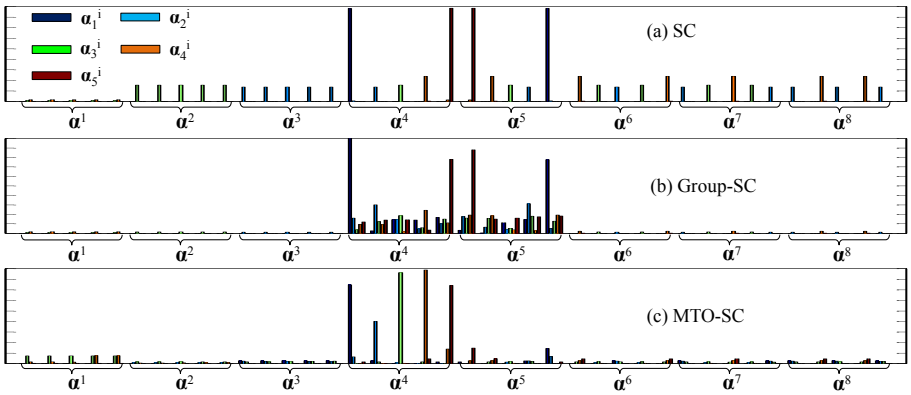

Fig. 3. Reconstruction coefficients of the input time series on the synthetic dataset using three algorithms: (a) sparse coding; (b) group sparse coding, and (c) Order-preserving sparse coding. See the color pdf for better view.

coefficients from sparse coding are similar among different dictionary sequences

and therefore the representation is less discriminative. This is because sparse coding does not utilize any structure information of either the dictionary or the input sequence; 2) reconstruction coefficients from multitask group sparse coding [7] are nonzero on few sequences; however, without enforcing the temporal order constraint, dictionary sequences with similar individual features but different ordering structures receive similar reconstruction coefficients, which results in ambiguities; and 3) as our method explicitly encourages temporal order preservation, the reconstruction coefficients well follow the temporal ordering of the input sequence.

4.2 Machine Learning Benchmarks

We apply the proposed algorithm on three benchmark time series datasets:

- **UCI Australian Sign Language signs (High Quality) Dataset** [23]: it consists of 2565 samples of Auslan signs captured from 9 native signers using high-quality position trackers. It contains 95 different signs, with 27 samples per sign. The average length of each sign is about 60 frames. Each frame is represented as a 15 dimensional feature vector consisting of hand position (X, Y, Z), roll, yaw, pitch, bend measurements of different fingers. For the ease of experiment, we randomly selected 4 subsets of the whole dataset with each subset containing 20 categories, denoted by AusLan1, AusLan2, AusLan3 and AusLan4, respectively.
- **UCI Spoken Arabic Digits Dataset** [24]: it contains times series of mel-frequency cepstrum coefficients (MFCCs) corresponding to spoken Arabic digits. It includes data from 44 male and 44 female native Arabic speakers, capturing 8800 (10 digits $\times$10 repetitions $\times$88 speakers) time series of 13 Frequency Cepstral Coefficients (MFCCs). The average length of each sample is about 40 frames.
- **CMU Motion Capture Dataset (CMU MoCap)** [25]: we use the same subset as in [26] which includes 5 actions, *i.e.*, jumping, golf swing, running, climbing and walking. The dataset contains 225 sequences with average length of 300 frames. Each frame is represented by rotation angles of 11 joints and end points including head, shoulders, elbows, hands, knees and feet.

For all above datasets, we randomly split them into training and testing sets of equal size. The random split is performed 10 times and all the reported testing results are averaged over the 10 random choices of the training and testing partition. For each frame, we normalize the feature vector to be of unit ℓ_2 norm. We compare our method (MTO-SC) with the following state-of-the-art time series classification methods: 1) Segmental Hidden Markov Model [11] (S-HMM); 2) DTW-based decision tree method [13] (DTW-DT); 3) DTW-based distance embedding [14] (DTW-DE); 4) numerosity reduction based DTW [15] (NR-DTW); 5) multi-resolution symbolic representation [17] (MSR); 6) sparse coding which is performed for every individual input frame (SC) followed by majority voting; and 7) multitask group sparse coding [7] (Group-SC). For all

Table 1. Classification accuracy [mean(std. dev.)] for different algorithms on different datasets.

Dataset	AusLan1	AusLan2	AusLan3	AusLan4	ArabSpokenDigit	CMU MoCap
S-HMM	75.46(1.39)	80.07(2.04)	82.30(1.77)	80.02(1.74)	54.30(1.37)	79.53(1.93)
DTW-DT	79.70(2.01)	82.73(1.98)	84.98(1.49)	83.76(1.91)	58.73(1.76)	82.97(1.64)
DTW-DE	79.93(1.49)	84.96(2.34)	85.41(1.84)	84.11(2.40)	59.60(2.40)	83.21(1.84)
NR-DTW	79.63(1.95)	85.43(1.91)	85.03(2.30)	83.09(2.21)	57.32(1.98)	83.60(1.28)
MSR	80.33(2.20)	89.77(1.35)	87.96(1.90)	85.98(1.90)	64.90(2.45)	85.36(1.92)
SC	83.09(1.89)	90.46(1.09)	90.00(1.66)	86.78(1.72)	45.98(2.10)	84.00(1.57)
Group-SC	84.19(1.76)	89.97(1.12)	91.50(1.89)	87.18(1.92)	46.04(1.99)	84.56(1.73)
MTO-SC	**91.40**(1.53)	**96.73**(1.80)	**95.53**(1.79)	**92.45**(1.81)	**75.80**(2.25)	**92.65**(1.09)

competing algorithms, their corresponding parameters (*e.g.*, λ_1 and λ_2 in MTO-SC, number of hidden states for HMM and number of neighborhood samples for DTW, etc.) are set by cross-validation on a validation subset.

We implemented these algorithms using Matlab 2010 on a 2.63 GHz machine with 8GB of memory. The average testing time per sample on AusLan1 dataset for S-HMM, NR-DTW (the fastest DTW-based method among the three), MSR, SC, Group-SC and MTO-SC are 3.5, 11.5, 2.8, 2.0, 2.2 and 2.4 seconds, respectively. We see that MTO-SC is among the most efficient methods in testing. For training, S-HMM, DTW-DT, DTW-DE, NR-DTW and MSR take about 4, 13, 3, 2.5 and 1.5 minutes, respectively, and the rest do not require training.

In Table 1, we report the mean classification accuracies averaged over 10 random splits with standard deviations. Observe that MTO-SC achieves the highest classification accuracy, owing to its capability in encoding temporal ordering structure for time series classification. To evaluate algorithmic robustness, we also add Gaussian noise to the AusLan1 dataset with zero mean and standard deviation in $\{0.1, 0.2, 0.3\}$ in three different experiments. The classification results are shown in Figure 4. Observe that our MTO-SC method is significantly more robust than its competitors.

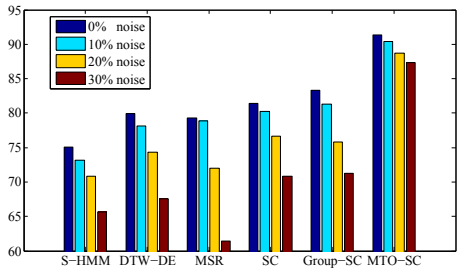

Fig. 4. Classification accuracies for different algorithms on the AusLan1 dataset under different noise conditions.

4.3 Human Activity Recognition

In this experiment, we use the RGB-D human activity dataset [27]. The video dataset is captured using the Kinect sensor, which produces 640 × 480 color-depth image sequences with human 3D motion sequences, namely, each activity sample can be represented as a sequence of 3D joint positions (or angles), similar to those in the CMU MoCap dataset. The dataset consists of five scenarios: office, kitchen, bedroom, bathroom, and living room. Three to four common activities were identified for each location, giving a total of twelve unique activities collected from 4 subjects (with an additional *neutral activity* category). We use similar feature representation as in [27], where each frame is represented as a combination of body pose, hand position, motion information and object contextual information. We use the leave-one-subject-out scheme, hence subjects in the testing samples do not occur in the training samples. We compare the multiclass classification accuracies for various algorithms including the proposed MTO-SC method, the time series classification methods compared in the previous experiment and SVM, One-level MEMM and hierarchical maximum entropy Markov model, which are also evaluated in [27]. The classification accuracies are summarized in Table 2 and the class confusion matrix for our method is illustrated in Figure 5. We observe that our method outperforms the other methods.

Table 2. Classification accuracy (%) on RGB-D human activity dataset.

Method	S-HMM	DTW-DT	DTW-DE	NR-DTW	MSR	SVM	MEMM	HMEMM	SC	Group-SC	MTO-SC
Accuracy	55.78	58.71	57.66	57.02	60.34	50.67	61.98	63.75	59.60	58.73	**65.32**

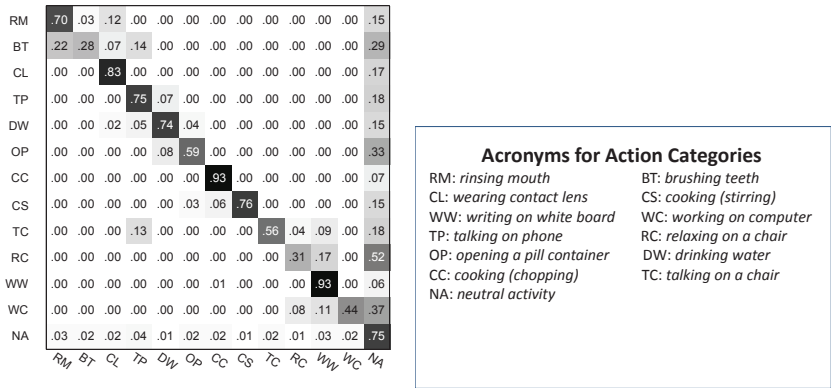

Fig. 5. Class confusion matrix for the proposed MTO-SC method on RGB-D human activity dataset.

5 Conclusions and Future Work

We have proposed an order-preserving sparse coding scheme for time series classification. Extensive experiments demonstrate that this scheme is highly discrim-

inative and robust. We will further consider learning a weighted reconstruction for different frames of the input sequence with the proposed order-preserving regularization scheme.

Acknowledgments. This study is supported by the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

References

1. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 210–227
2. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing **15** (2006) 3736—3745
3. Rao, S., Tron, R., Vidal, R., Ma, Y.: Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: CVPR. (2008)
4. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society B **67** (2005) 301–320
5. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society B **68** (2006) 49–67
6. Zhang, J.: A probabilistic framework for multi-task learning. Technical report, CMU-LTI-06-006 (2006)
7. Yuan, X., Yan, S.: Visual classification with multi-task joint sparse representation. In: CVPR. (2010)
8. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: Transfer learning from unlabeled data. In: ICML. (2007)
9. Nesterov, Y.: Smooth minimization of non-smooth functions. Mathematical Programming (2005) 127–152
10. Rabiner, L.R., Juang, B.H.: An introduction to hidden markov models. IEEE Magazine on Accoustics, Speech and Signal Processing **3** (1986) 4–16
11. Kim, S., Smyth, P.: Segmental hidden markov models with random effects for waveform modeling. Journal of Machine Learning Research **7** (2006) 945–969
12. Myers, C.S., Rabiner, L.R.: A comparative study of several dynamic time-warping algorithms for connected word recognition. The Bell System Technical Journal (**60**)
13. Rodríguez, J.J., Alonso, C.J.: Interval and dynamic time warping-based decision trees. In: ACM Symposium on Applied Computing (2004) 548–552
14. Hayashi, A., Mizuhara, Y., Suematsu, N.: Embedding time series data for classification. Machine Learning and Data Mining in Pattern Recognition (2005) 356–365
15. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: ICML. (2006) 1033–1040
16. Nanopoulos, A., Alcock, R., Manolopoulos, Y.: Feature-based classification of time series data. Information Processing and Technology (2001) 49–61
17. Megalooikonomou, V., Wang, Q., Li, G., Faloutsos, C.: A multiresolution symbolic representation of time series. In: ICDE. (2005) 668–679
18. Manning, C.D., Schtze, H.: Foundations of statistical natural language processing, MIT press. (1999)
19. Cadieu, C., Olshausen, B.: Learning transformational invariants from natural movies. In: NIPS. (2008) 209–216
20. Kim, T., Shakhnarovich, G., Urtasun, R.: Sparse coding for learning interpretable spatio-temporal primitives. In: NIPS. (2010) 1117–1125
21. Zhao, B., Fei-Fei, L., Xing, E.P.: Online detection of unusual events in videos via dynamic sparse coding. In: CVPR. (2011)
22. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Submitted to SIAM Journal on Optimization (2008)
23. Kadous, M.W.: Temporal classification: Extending the classification paradigm to multivariate time series. PhD Thesis, School of Computer Science and Engineering, University of New South Wales (2002)
24. Hammami, N., Bedda, M.: Improved tree model for arabic speech recognition. In: Internationl Conference on Computer Science and Information Technology (2010) 521—526
25. : (http://mocap.cs.cmu.edu/)
26. Shen, Y., Ashraf, N., Foroosh, H.: Action recognition based on homography constraints. In: ICPR. (2008)
27. Sung, J., Ponce, C., Selman, B., Saxena, A.: Human activity detection from rgbd images. CoRR **abs/1107.0169** (2011)