This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Temporal Subspace Clustering for Human Motion Segmentation

Sheng Li[1] Kang Li[1] Yun Fu[1,2]

[1]Department of Electrical and Computer Engineering, College of Engineering
[2]College of Computer and Information Science
Northeastern University, Boston, MA, USA

shengli@ece.neu.edu, li.ka@husky.neu.edu, yunfu@ece.neu.edu

Abstract

Subspace clustering is an effective technique for segmenting data drawn from multiple subspaces. However, for time series data (e.g., human motion), exploiting temporal information is still a challenging problem. We propose a novel temporal subspace clustering (TSC) approach in this paper. We improve the subspace clustering technique from two aspects. First, a temporal Laplacian regularization is designed, which encodes the sequential relationships in time series data. Second, to obtain expressive codings, we learn a non-negative dictionary from data. An efficient optimization algorithm is presented to jointly learn the representation codings and dictionary. After constructing an affinity graph using the codings, multiple temporal segments can be grouped via spectral clustering. Experimental results on three action and gesture datasets demonstrate the effectiveness of our approach. In particular, TSC significantly improves the clustering accuracy, compared to the state-of-the-art subspace clustering methods.

1. Introduction

Subspace clustering has attracted an increasing attention in recent years, due to its impressive performance in many real-world applications, such as motion segmentation [17], face clustering [3] and digit clustering [29]. The representative subspace clustering methods include sparse subspace clustering (SSC) [3], low-rank representation (LRR) [17], least-square regression (LSR) [21], etc. The key idea in subspace clustering is to learn effective representation codings that are used to construct an affinity matrix. Many algorithms have been proposed to enhance the performance of subspace clustering, by enforcing different constraints on the coefficients [19], or developing scalable implementations [18, 23, 26, 32].

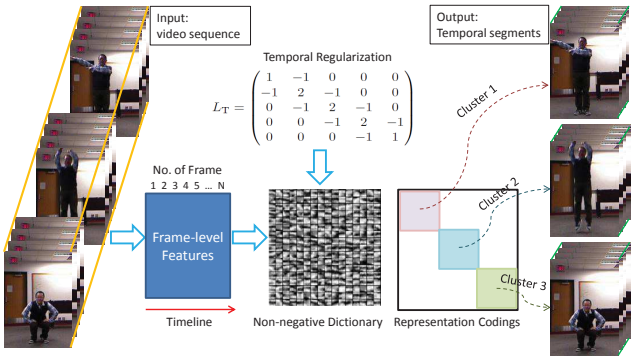

Figure 1. Framework of the proposed approach. We first extract frame-level features from video sequence, and then learn a non-negative dictionary and the representation codings, with the help of temporal regularization. Multiple temporal segments are finally generated via clustering on the codings.

However, with the notable exception of [27], there are few subspace clustering methods focusing on the data with specific properties, such as the time-series data. Generally, existing methods assume that the data points are independently drawn from multiple subspaces. They either model the data points independently [3] or implicitly consider the global structural information in data [17], but neglect the successive relationships that possibly reside in data. In reality, time-series data like videos can be found everywhere [13, 34, 12]. Labelling or manually processing a large amount of videos is expensive and time-consuming. Therefore, it is necessary to devise unsupervised visual learning algorithms to handle the time-series data.

In this paper, we propose a temporal subspace clustering (TSC) method, and apply it to the unsupervised segmentation of human motion. Figure 1 illustrates the idea of our approach. We adopt the least-square regression based formulation to learn effective codings for each data point. Motivated by the well-known Laplacian regularization technique, we design a temporal Laplacian regularization func-

tion to encode the sequential relationships in time series data. To obtain more expressive codings, we learn a non-negative dictionary from data, instead of using the data self-representation model as existing methods [17, 3]. We present an efficient optimization algorithm to jointly learn the codings and dictionary. After constructing an affinity graph using the codings, multiple temporal segments can be automatically grouped via spectral clustering. Experimental results on three action and gesture datasets demonstrate the effectiveness of our approach compared to the state-of-the-art clustering methods. In summary, the contributions of this paper include:

- We design a novel temporal Laplacian regularization function to model the sequential information in time series data. To the best of our knowledge, this paper presents the first temporal Laplacian regularization for subspace clustering.
- We develop a non-negative dictionary learning method to learn expressive codings for temporal clustering. The non-negative bases in dictionary are especially useful for the human motion data (e.g., action videos) that are usually non-negative values.
- We present an efficient optimization algorithm to jointly learn the non-negative dictionary and expressive codings, which are used for constructing a robust affinity graph.

The rest of this paper is organized as follows. In Section 2, we briefly discuss some related works. In Section 3, we present the details of our approach. Extensive experimental results on three public datasets are reported in Section 4. Section 5 concludes this paper.

2. Related Work

There are three types of works that are most related to our approach: (1) subspace clustering, (2) temporal clustering, and (3) unsupervised motion analysis.

Subspace Clustering is an effective technique which can automatically group the samples into low-dimensional subspace. It has achieved impressive performance in many real-world applications, such as motion segmentation [17], face clustering [3] and digit clustering [29]. Sparse subspace clustering (SSC) [3] enforces a sparse constraint on the coefficients. Low-rank representation (LRR) [17] considers the global structure of sample space, and usually achieves better performance than LRR. Least-square regression (LSR) [21] is very efficient by using Frobenius norm. Sparse additive subspace clustering (SASC) extends SSC to the additive nonparametric setting [31]. Discriminative subspace clustering (DSC) [36] incorporates discriminative information into the model. Smooth representation (SMR) makes use of the grouping effect to further enhance the subspace clustering performance [7]. In addition, many algorithms have been devised to reduce the computational cost of subspace clustering [23, 26, 29]. However, these methods can not handle the time-series data very well. The most relevant work to ours is the ordered subspace clustering (OSC) [27]. It aims at finding the sparse representations for data, and also introduces a penalty term to take care of the sequential data. OSC also presents an "intrinsic segmentation" strategy to automatically find the segmentation boundaries. However, OSC and our approach have significant technique differences. First, OSC explicitly adds the temporal regularization via ZR, where R is a specific constant matrix. But our approach offers a more flexible way to encode temporal information through Laplacian regularization. Secondly, OSC utilizes the sample set itself as the bases for sparse representation, while our approach learns a non-negative dictionary that consists of expressive bases for temporal subspace clustering.

Temporal Clustering segments time series data into a set of non-overlapping groups. It can be applied to learning taxonomies of facial behavior, speaker diarization, discovering motion primitives and clustering human actions in videos. By far only a few temporal clustering methods have been developed, such as the extensions of dynamic Bayesian networks (DBNs) [4], k-means [24], spectral clustering [35], and maximum-margin temporal clustering [22]. Basically, these temporal clustering methods focus on the post-processing after graph construction, while the above subspace clustering methods focus on learning codings for graph construction. Our TSC approach mainly belongs to the subspace clustering category. In another word, our approach can be easily concatenated to these post-processing methods to further enhance the performance.

Unsupervised Motion Analysis is an important application of temporal clustering [2, 28, 35]. Recently, some methods based on metric learning [20], regression analysis [10] and spatio-temporal kernel [5] have achieved impressive performance. In this paper, we will evaluate the clustering performance of our approach and baselines on the real-world motion datasets.

3. Our Approach

3.1. Problem Formulation

The conventional *subspace clustering* (or subspace segmentation) problem is defined as follows.

Let $\bar{X}$ denote a set of data vectors $\bar{X} = [\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n]$ (each column is a sample) in a D-dimensional Euclidean space. These data vectors are assumed to be drawn from a union of k subspaces $\{\mathbb{S}_i\}_{i=1}^{k}$ of unknown dimensions. ***Subspace clustering*** [17] aims to cluster all data vectors into their respective subspaces.

Clearly, traditional subspace clustering problem neglects the temporal information in data, which makes it unsuitable for the time series data. Considering the temporal relationship contained in data, we define the *temporal subspace clustering* problem as follows.

Consider a sequence of time-series data $X = [x_1, x_2, \cdots, x_n]$ (the i-column is sampled at time t_i) drawn from a union of k subspaces of unknown dimensions, ***temporal subspace clustering*** groups the n samples into $m (m \geq k)$ sequential segments, and clusters the m segments into their respective subspaces.

Given a dictionary (i.e., bases) $D \in \mathbb{R}^{d\times r}$ and a coding matrix $Z \in \mathbb{R}^{r\times n}$, the time series data $X \in \mathbb{R}^{d\times n}$ can be approximately represented as:

$$X \approx DZ, \tag{1}$$

where d is the dimension of samples, r is the number of bases in dictionary, and n is the total number of samples (a.k.a., the number of time stamps).

We adopt the least-square regression based formulation for temporal subspace clustering. The objective function is:

$$\min_{Z,D} \|X - DZ\|_{\mathrm{F}}^2 + \lambda_1 \|Z\|_{\mathrm{F}}^2, \tag{2}$$

where $\|Z\|_{\mathrm{F}}$ denotes the Frobenius norm of Z, i.e., $\|Z\|_{\mathrm{F}}^2 = \sum_{i=1}^{r}\sum_{j=1}^{n} Z_{ij}^2$, and λ_1 is a trade-off parameter.

The first term $\|X - DZ\|_{\mathrm{F}}^2$ captures the reconstruction error, and the second term $\|Z\|_{\mathrm{F}}^2$ is used to model the global subspace structure in X. Moreover, it has shown that the Frobenius norm is a good choice to enforce the block diagonal structure in Z, which is the key to recovering subspace structures [21].

Inspired by the commonly-used manifold regularization technique [33], we design a temporal Laplacian regularization function $f_{\mathrm{t}}(Z)$ to incorporate the temporal information in time series X. The i-th column in coding matrix Z, z_i, can be viewed as a new representation for x_i. Our motivation is that, the sequential neighbors of z_i (e.g., z_{i-1}, z_{i+1}) could be close to z_i in the coding space.

Definition 1. (Temporal Laplacian Regularization) Given a coding matrix Z, the temporal Laplacian regularization function is defined as:

$$f(Z) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\|z_i - z_j\|_2^2 = \mathrm{tr}(ZL_{\mathrm{T}}Z^\top), \tag{3}$$

where L_{T} is a temporal Laplacian matrix, $L_{\mathrm{T}} = \tilde{D} - W$, $\tilde{D}_{ii} = \sum_{j=1}^{n} w_{ij}$, W is the weight matrix that captures the sequential relationships in X. Let s denote the number of sequential neighbors for each sample, the element in W is calculated as

$$w_{ij} = \begin{cases} 1, & \text{if } |i-j| \leq \frac{s}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Different from existing Laplacian regularization that considers the spatial closeness of all data points, our temporal regularization function $f(Z)$ mainly focuses on the temporal closeness in time series data.

Example. To better illustrate why $f(Z)$ is able to encode the temporal information, we show the structure of W in a simple case. If we have $n = 5$ and $s = 2$, the weight matrix W and the temporal Laplacian matrix L_{T} are:

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad L_{\mathrm{T}} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Note that we only use the binary weights for W to show the idea. Other sophisticated graph weighting algorithms can also be applied here to attain better performance.

Then, the objective function in (2) can be rewritten as:

$$\min_{Z,D} \|X - DZ\|_{\mathrm{F}}^2 + \lambda_1 \|Z\|_{\mathrm{F}}^2 + \lambda_2 f(Z), \tag{5}$$

where λ_2 is a trade-off parameter to balance different terms.

We can observe that L_{T} is a special case of Laplacian matrix by enforcing the temporal consistency, which is more suitable for time series data. In this manner, sample x_i and its sequential neighbors $\{x_{i-s/2}, \cdots, x_{i+s/2}\}$ are encouraged to have the similar codings $\{z_{i-s/2}, \cdots, z_{i+s/2}\}$. We will show that $f(Z)$ helps us obtain continuous segments from time series data. Therefore, our model is more robust to noise and abnormal events in the temporal space.

Another key factor in subspace clustering is the choice of dictionary. Dictionary learning has attracted a lot of attention [14], but existing subspace clustering methods usually follow the data self-representation strategy, i.e., the data set X serves as the dictionary. However, when the sampling is insufficient or the data set X is heavily corrupted, employing X as dictionary may hinder the clustering performance. Thus, learning an expressive dictionary is necessary.

To address this problem, we introduce the dictionary learning procedure into problem (5). Moreover, as time series data in real-world applications (e.g., action videos, human motions) are usually non-negative values, it is reasonable to learn a non-negative dictionary for temporal subspace clustering, i.e., $D \geq 0$. Naturally, the coding matrix Z should also be non-negative, i.e., $Z \geq 0$.

After adding the dictionary learning component and two non-negative constraints, we have the **temporal subspace clustering (TSC) model** as follows:

$$\begin{aligned} \min_{Z,D} \quad & \|X - DZ\|_{\mathrm{F}}^2 + \lambda_1 \|Z\|_{\mathrm{F}}^2 + \lambda_2 f(Z), \\ \text{s.t.} \quad & Z \geq 0,\ D \geq 0,\ \|d_i\|_2^2 \leq 1,\ i = 1, \cdots, r. \end{aligned} \tag{6}$$

The non-negative constraints $Z \geq 0$ and $D \geq 0$ ensure that the learned bases and corresponding bases should be non-

negative values, and the constraint $\|d_i\|_2^2 \leq 1$ controls the model complexity.

3.2. Optimization

To solve the objective function (6), we devise an optimization algorithm based on the alternating direction method of multipliers (ADMM). To facilitate the optimization, we consider an equivalent form of (6):

$$\begin{aligned} \min_{Z,D,U,V} \quad & \|X - UV\|_{\mathrm{F}}^2 + \lambda_1 \|V\|_{\mathrm{F}}^2 + \lambda_2 f(V), \\ \text{s.t.} \quad & U = D,\ V = Z,\ Z \geq 0,\ D \geq 0, \\ & \|d_i\|_2^2 \leq 1,\ i = 1, \cdots, r, \end{aligned} \tag{7}$$

where U and V are auxiliary variables.

The augmented Lagrangian of (7) is:

$$\begin{aligned} \mathcal{L} = \quad & \tfrac{1}{2}\|X - UV\|_{\mathrm{F}}^2 + \lambda_1 \|V\|_{\mathrm{F}}^2 + \lambda_2 \mathrm{tr}(VL_{\mathrm{T}}V^\top) \\ & + \langle \Lambda, U - D\rangle + \langle \Pi, V - Z\rangle + \tfrac{\alpha}{2}(\|U - D\|_{\mathrm{F}}^2 \\ & + \|V - Z\|_{\mathrm{F}}^2) \\ \text{s.t.} \quad & Z \geq 0,\ D \geq 0,\ \|d_i\|_2^2 \leq 1,\ i = 1, \cdots, r, \end{aligned} \tag{8}$$

where Λ and Π are Lagrangian multipliers, and α is a penalty parameter.

The ADMM algorithm for (7) is derived by alternatively minimizing $\mathcal{L}$ with respect to V, U, Z and D.

Update V when fixing others. The problem (8) becomes:

$$\begin{aligned} \min_{V} \quad & \tfrac{1}{2}\|X - UV\|_{\mathrm{F}}^2 + \lambda_1 \|V\|_{\mathrm{F}}^2 + \lambda_2 \mathrm{tr}(VL_{\mathrm{T}}V^\top) \\ & + \langle \Pi, V - Z\rangle + \tfrac{\alpha}{2}\|V - Z\|_{\mathrm{F}}^2 \end{aligned} \tag{9}$$

By setting the derivative of (9) with respect to V to zero, we have the following equation:

$$(U^\top U + (\lambda_1 + \alpha)\mathrm{I})V + \lambda_2 V L_{\mathrm{T}} = U^\top X - \Pi + \alpha Z. \tag{10}$$

Eq. (10) is a standard Sylvester equation, which can be effectively solved using existing tools such as the Bartels-Stewart algorithm [1]. Alternatively, we can vectorize the linear matrix equation (10) into:

$$\begin{aligned} & [I \otimes (U^\top U + (\lambda_1 + \alpha)\mathrm{I}) + \lambda_2 L_{\mathrm{T}} \otimes I]\mathrm{vec}(V) \\ & = \mathrm{vec}(U^\top X - \Pi + \alpha Z), \end{aligned} \tag{11}$$

where $\otimes$ is the tensor product.

Update U when fixing others. By ignoring the variables that are irrelevant to U, we have:

$$\min_{U} \quad \tfrac{1}{2}\|X - UV\|_{\mathrm{F}}^2 + \langle \Lambda, U - D\rangle + \tfrac{\alpha}{2}\|U - D\|_{\mathrm{F}}^2 \tag{12}$$

Setting the derivative of (12) with respect to U to zero, we have the solution:

$$U = (XV^\top - \Lambda + \alpha D)(VV^\top + \alpha \mathrm{I})^{-1}. \tag{13}$$

Update Z and D when fixing others. The update rules are:

$$Z = \mathcal{F}_+(V + \tfrac{\Pi}{\alpha}), \tag{14}$$

$$D = \mathcal{F}_+(U + \tfrac{\Lambda}{\alpha}), \tag{15}$$

where $(\mathcal{F}_+(A))_{ij} = \max\{A_{ij}, 0\}$, which meets the non-negative requirements for D and Z. We also normalize each column vector in D to unit length.

The above process is repeated until convergence. For the non-convex problems or convex problems with multiple blocks, there is no theoretical guarantee for the global convergence of ADMM. However, we can show the convergence property of ADMM under mild conditions, following the analysis in [30].

Theorem 1. *Let $\{(V_k, U_k, Z_k, D_k, \Pi_k, \Lambda_k)\}$ be a sequence generated by Algorithm 1 in the k-th iteration. If the sequences of multipliers $\{(\Pi_k, \Lambda_k)\}$ is bounded and satisfies*

$$\sum_{k=0}^{\infty}(\|\Pi_{k+1} - \Pi_k\|_{\mathrm{F}}^2 + \|\Lambda_{k+1} - \Lambda_k\|_{\mathrm{F}}^2) < \infty. \tag{16}$$

Then any accumulation point of the generated sequence $\{(V_k, U_k, Z_k, D_k, \Pi_k, \Lambda_k)\}$ satisfies the KKT condition of problem (7).

The proof will be provided in the supplementary document due to the space limit.

3.3. Clustering

The coding matrix Z can be used to construct an affinity graph G for subspace clustering. In SSC, LRR and LSR, the definition of G is $G = \frac{|Z| + |Z^{\mathrm{T}}|}{2}$. However, this graph does not well exploit the intrinsic relationships of within-cluster samples. For time series data, the within-cluster samples (i.e., sequential neighbors) are always highly correlated to each other [15, 16]. Therefore, we can take advantage of this property and devise another similarity measurement to construct G.

$$G(i,j) = \frac{z_i^{\mathrm{T}} z_j}{\|z_i\|_2 \|z_j\|_2}. \tag{17}$$

In the experiments, we evaluate both similarity measurements for every baseline, and report the better results. Finally, an effective clustering algorithm, Normalized Cuts [25], is utilized to produce the temporal clustering results. The complete temporal subspace clustering approach is summarized in Algorithm 1.

3.4. Discussion

Note that when setting $\lambda_1 = \lambda_2 = 0$ and removing the constraints $\|d_i\|_2^2 \leq 1$, our model is equivalent to the non-negative matrix factorization (NMF) [11]. However, such settings are not suitable for dealing with time-series data.

Algorithm 1. *Temporal Subspace Clustering (TSC)*

Input: Time series data X, $k = 0$, step size η, number of clusters k, parameters s, λ_1, λ_2, α

Output: Clustering index vector Y

1: Construct matrices W, $\tilde{D}$, and L according to Section 3.1;
2: **while** *not converged* **do**
3: Update $V_{(k+1)}$ using (10), given others fixed;
4: Update $U_{(k+1)}$ using (13), given others fixed;
5: Update $Z_{(k+1)}$ using (14), given others fixed;
6: Update $D_{(k+1)}$ using (15), given others fixed;
7: Update Π_{k+1}: $\Pi_{k+1} = \Pi_k + \eta\alpha(V_{k+1} - Z_{k+1})$;
8: Update Λ_{k+1}: $\Lambda_{k+1} = \Lambda_k + \eta\alpha(U_{k+1} - D_{k+1})$;
9: $k = k + 1$;
10: **end while**
11: Build an undirected graph G using (17);
12: Use NCut to generate k clusters, get index Y.

In Algorithm 1, we initialize D and Z with random values. All the other variables such as U and V are initialized with zero. To evaluate the efficiency of our algorithm, we present the analysis of time complexity. The most time-consuming step in Algorithm 1 is step 3, which costs $\mathbf{O}(r^2n)$. Let t denote the number of iterations, the overall computational complexity of our algorithm is $\mathbf{O}(tnr^2)$, which enjoys a good scalability w.r.t. the sample size n.

4. Experiments

In this section, we compare our approach with several state-of-the-art subspace clustering approach on three human action and gesture datasets.

4.1. Settings

In the experiments we utilize three public datasets, including the Keck dataset [9], Weizmann dataset [6], and Multi-Modal Action Detection (MAD) dataset [8].

Baselines. Our TSC approach is compared with the following representative clustering methods:

- Sparse Subspace Clustering (SSC) [3], which enforces a sparse constraint on the representation coefficients.
- Low-Rank Representation (LRR) [17], which incorporates a low-rank constraint on the coefficients.
- Least Square Regression (LSR) [21], which adopts a regression based formulation for subspace clustering. It's a special case of our TSC method when D is fixed, $\lambda_2 = 0$, and ignoring the non-negative constraint.
- Ordered Subspace Clustering (OSC) [27], which explicitly enforces the consecutive columns of Z to be similar. It achieves the state-of-the-art results for clustering sequential data.

For those compared methods, we use the codes provided by the authors, and fine tune the parameters to achieve the best performance. Further, we will discuss how to choose parameters of our approach in the next subsections.

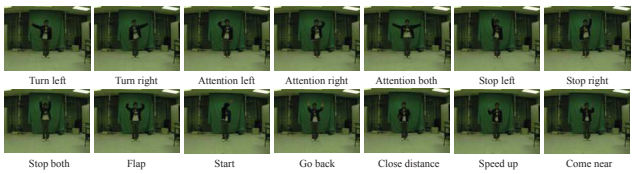

Figure 2. 14 gestures in Keck dataset.

Table 1. Clustering accuracies with standard derivation and running time of all compared methods on Keck dataset.

Methods	Accuracy (%)	NMI	Time (s)
SSC [3]	26.81±2.41	0.2861	59.05
LRR [17]	12.84±3.75	0.0617	14.82
LSR [21]	38.22±2.09	0.3244	6.89
OSC [27]	41.89±2.30	0.4933	461.18
TSC (**Ours**)	**57.12** ± 2.13	0.6695	49.05

In the experiments, we use the clustering accuracy (AC) and normalized mutual information (NMI) as the evaluation metrics.

4.2. Gesture Clustering

The Keck gesture data consists of 14 different gestures [9], which originally come from military signals. Each gesture is performed by three subjects, so there are three sequences for each gesture. In each sequence, the same gesture is repeated three times. Figure 2 shows the 14 gestures of one subject in the dataset. The original resolution of each frame is 480×640. To speed up the computation, we downsample each frame to the size of 80×106. Following [22], we extract binary masks and compute the Euclidean distance transform as frame-level features. Then we build a dictionary of temporal words with 100 clusters using the k-means clustering, and encode each frame as a 100 dimensional binary vector.

We concatenate the 14 gesture video sequences of each subject into a single long video sequence, and evaluate the performance of different methods. We also evaluate the computational cost of different methods. The machine used in our experiments installs 24 GB RAM and Intel Xeon W3350 CPU. The parameters s, λ_1 and λ_2 are empirically set to 6, 0.01 and 15, respectively. Table 1 reports the average clustering accuracy and the running time. Our TSC approach significantly outperforms the other compared methods, and improves the average accuracy by at least 15%.

Figure 3 shows the details of clustering results of one random experiment, by rendering clusters as different colors. It shows that SSC, LRR and LSR can not obtain meaningful temporal segments, as they do not consider the temporal information. OSC and our TSC methods can obtain continuous segments in most cases. Moreover, because of

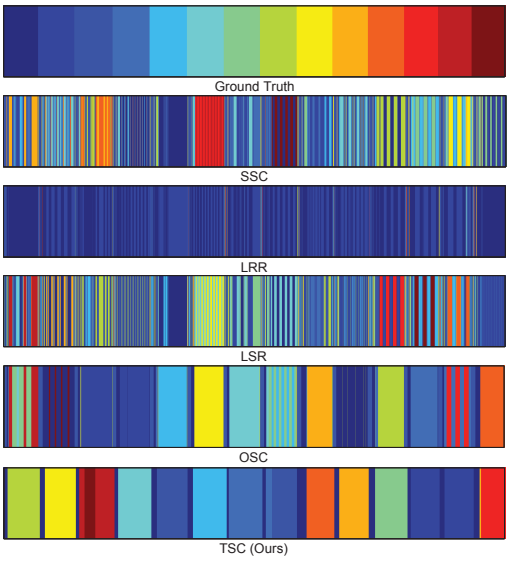

Figure 3. Clustering results on Keck dataset. 14 colors denote 14 different gestures. (Please view the color figure for better visualization)

the temporal Laplacian regularization function and the expressive dictionary, our approach is able to correctly recover the subspace structures in temporal space, and therefore achieves clearer sequential subspace structures than OSC.

In addition, we notice an interesting phenomenon from the clustering results of TSC in Figure 3. It shows that each cluster contains a short blue sequence at the beginning. After looking into the video sequences, we find that, at the beginning of each sequence, the subject walked towards the center of the room, and then performed the required gestures. It demonstrates our approach has the ability to discover undefined clusters, which might be important in some high-level vision tasks, such as video understanding.

4.3. Action Clustering

We evaluate the action clustering performance of our approach and compared methods on the Weizmann dataset [6] and the MOD dataset [8]. The action data in the Weizmann dataset are organized in isolated clips, which provides an ideal controlled evaluation platform for temporal clustering. In addition, the MAD dataset contains continuous actions, and the start and end of each action is provided. It provides a more realistic scenario for temporal clustering.

Weizmann dataset. The Weizmann dataset contains 90 video sequences (180×144 pixels, 50fps) captured from 9 subjects [6]. Each subject performs 10 different actions, including jumping-jack (or shortly jack), jump-forward-on-two-legs (jump), jump-in-place-on-two-legs (pjump), gallop-sideways (side), bend, skip, walk, run, wave-one-hand (wave1), and wave-two-hands (wave2). Figure 4 shows 10 frames of different actions in the dataset. Following the settings in [6], we extract binary masks and compute the Euclidean distance transform as frame-level features, and utilize the bag-of-words model to encode the features as binary vectors.

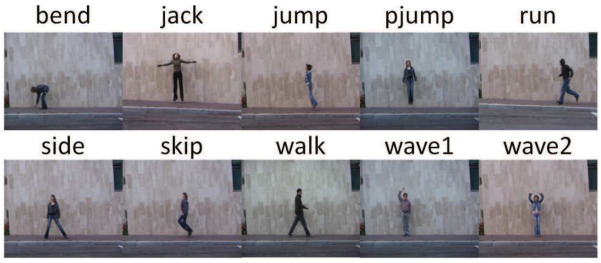

Figure 4. 10 actions in Weizmann dataset.

Table 2. Clustering accuracies (with standard derivation) and running time of all compared methods on Weizmann dataset.

Methods	Accuracy (%)	NMI	Time (s)
SSC [3]	38.81±3.28	0.1214	289.53
LRR [17]	43.55±3.75	0.1365	10.26
LSR [21]	40.11±2.94	0.1164	3.61
OSC [27]	65.89±3.27	0.4655	692.14
TSC (**Ours**)	**76.15**±2.88	**0.6844**	34.16

In this dataset, each video sequence only contains a single action. To evaluate the clustering performance of our approach and related methods, we follow the experimental protocol in [22], and concatenate multiple single-action sequences into a longer video sequence. In particular, we randomly select 5 action sequences from each subject, and concatenate these sequences into a long sequence. We repeat this procedure with 10 runs. The parameters s, λ_1 and λ_2 are empirically set to 6, 0.001 and 15, respectively. Table 2 lists the average clustering accuracy (with standard derivation) and running time of each method. We can observe that our approach obtains much better results than the compared methods. The average clustering accuracy is improved by at least 10%, comparing with the state-of-the-art method OSC.

Multi-modal Action Detection (MAD) Dataset. The sequences in the Keck dataset and Weizmann dataset are isolated clips. However, manually concatenating the isolated clips results in discontinuous time series that are not valid in realistic scenario. The recently published MAD dataset contains multiple continuous actions, which is more challenging than the Weizmann dataset. The MAD dataset contains 40 sequences captured from 20 subjects (2 sequences per subject). Each subject performs all the 35 activities continuously, and the segments between two actions are considered the null class (i.e., the subject is standing) [8]. The 35 actions include full-body motion (e.g., Running, Crouching, jumping), lower-body motion (e.g., kicking), and upper-body motion(e.g., Throw, Basketball

Table 3. Clustering accuracies (%) with standard derivation of all compared methods on MAD dataset.

Methods	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Average±Std
SSC [3]	34.78	39.25	38.71	38.24	30.05	36.21±3.86
LRR [17]	35.30	39.82	40.15	39.71	31.23	37.24±3.91
LSR [21]	36.12	40.73	38.54	40.10	33.15	37.73±3.11
OSC [27]	38.55	41.98	40.12	42.25	38.22	40.22±1.87
TSC (**Ours**)	**45.92**	**50.61**	**48.14**	**49.17**	**44.52**	**47.67**±2.45

Table 4. Normalized mutual information (NMI) of all compared methods on MAD dataset.

Methods	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Average±Std
SSC [3]	0.2241	0.2385	0.2301	0.2297	0.2154	0.2276±0.0085
LRR [17]	0.2016	0.2187	0.2248	0.2045	0.1985	0.2096±0.0115
LSR [21]	0.2401	0.2398	0.2515	0.2349	0.2207	0.2374±0.0111
OSC [27]	0.2618	0.2714	0.2703	0.2925	0.2544	0.2701±0.0143
TSC (**Ours**)	**0.3435**	**0.3677**	**0.3520**	**0.3312**	**0.3287**	**0.3446**±0.0160

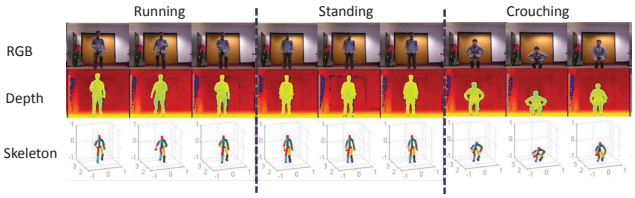

Figure 5. Example frames of MAD dataset.

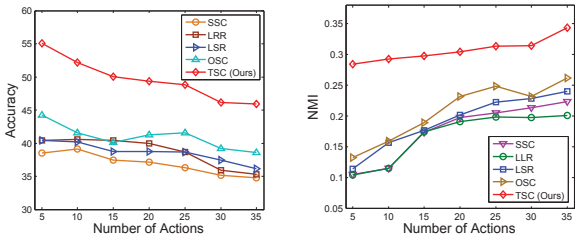

Figure 6. Clustering results on MAD dataset (Subject 1) with different number of actions. (Left: Accuracy. Right: NMI)

Dribble, Baseball swing). The length of each sequence is around 2-4 minutes (4000-7000 frames). Each sequence has three different modalities: RGB video (240×320), 3D depth (240×320), and a body-joint sequence (3D coordinates of 20 joints per frame). Figure 5 shows some frames in the MAD dataset.

We use depth sequences to generate binary masks of human, and compute the Euclidean distance transform as frame-level features. Then we build a dictionary of temporal words with 100 clusters using the k-means clustering, and encode each frame as a 100 dimensional binary vector. We randomly choose 5 subjects, and evaluate the clustering performance of each compared method. The parameters are fine tuned to achieve the best result of each method. Figure 6 shows the clustering results on MAD dataset (Subject 1) with different number of actions. It shows that our TSC approach consistently achieves much better results than other methods. Table 3 and Table 4 list the clustering accuracy and NMI for each subject, which demonstrates the effectiveness of our approach.

4.4. Discussions

Graph Visualization. Constructing an effective graph is the key in clustering methods. Indeed, existing subspace clustering methods and our approach mainly focus on estimating the coding matrix for graph construction. To illustrate why our approach performs much better than its competitors, we visualize the graphs learned by SSC, LRR, LSR, OSC and our approach in Figure 7. By considering the sequential relationships in time-series data, we can observe the much denser block diagonals in the graphs of OSC and our approach compared to other graphs. It implies that the with-cluster structures are enhanced in OSC and our graph. Moreover, as our approach is more flexible to control the sequential neighbors, the graph structure of our approach is clearer than OSC.

Comparisons with Motion Segmentation Methods. We also compare the clustering performance between our method and the motion segmentation methods. MMTC is a maximum margin clustering method. ACA is based on spectral clustering, and HACA is an improved version of ACA. To the best of our knowledge, HACA is the state-of-the-art unsupervised motion analysis method. Table 5 shows the clustering results on three datasets. It shows that HACA usually performs better than ACA and MMTC, while our TSC method consistently outperforms others.

Parameter Sensitivity. There are two major parameters in our model, λ_1 and λ_2. Figure 8(Left) shows the clustering accuracy of our TSC approach with different values of λ_1 and λ_2. We can observe clustering results are not very sensitive to λ_1 in the range $[0, 0.002]$. Meanwhile, $\lambda_2 = 10$ can lead to the best clustering result. We can also validate the effectiveness of the temporal Laplacian regularization func-

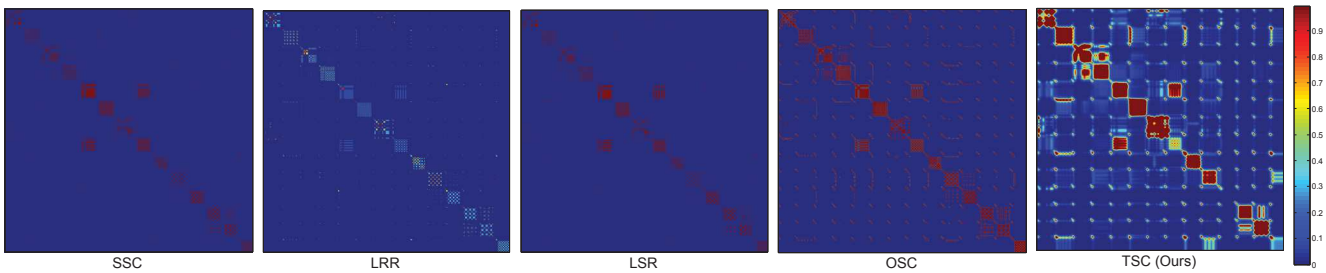

Figure 7. Visualization of graphs learned by SSC, LRR, LSR, OSC and our TSC approach on Keck gesture dataset. The red color denotes large graph weights, while the blue color indicates small weights.

Table 5. Clustering accuracies of TSC and the state-of-the-art motion segmentation methods on three datasets.

Methods	Keck	Weizmann	MAD
MMTC [22]	N/A	68.05	N/A
ACA [35]	51.76	75.06	42.05
HACA [35]	53.42	75.80	45.31
TSC (**Ours**)	**57.12**	**76.15**	**47.67**

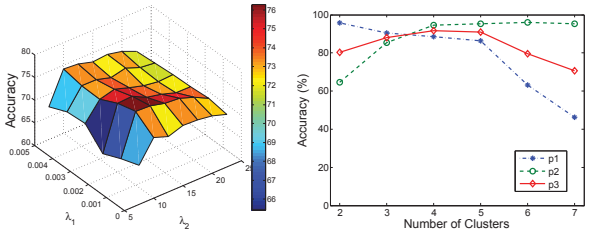

Figure 8. Sensitivity analysis on Weizmann dataset. Left: accuracy with different values of parameters; Right: accuracy with different number of clusters.

tion from Figure 8 (Left). In practice, we may not know the true number of clusters. For sensitivity analysis, we vary the desired number of clusters but fix the number of true classes. In this case, the evaluation metrics like accuracy and NMI cannot be directly applied, since there was no one-to-one mapping between the generated clusters and ground truth. Instead, we use a *pair-counting* measurement designed in [22]. Consider all pairs of same class video frames, p_1 is defined as the percentage of pairs of which both frames were assigned to the same cluster. Consider all pairs of different-class video frames, p_2 is defined as the percentage of pairs of which two frames were assigned to different clusters. Moreover, p_3 is the average of p_1 and p_2, which shows the clustering performance. Figure 8 (Right) shows the values of p_1, p_2 and p_3 by varying the desired number of clusters from 2 to 7. We observe that the summarized value p_3 is relatively stable in a wide range.

In addition, Figure 9 shows the clustering results with different size of dictionary and different number of sequential neighbors on the Keck dataset. It shows that our approach is not very sensitive to the size of dictionary in a wide range. Clustering accuracy would decrease slightly when the number of sequential number increases.

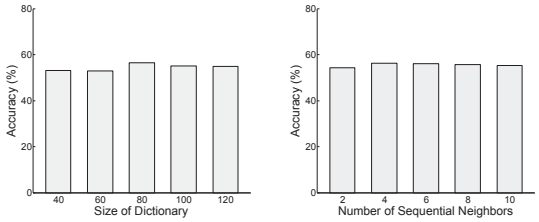

Figure 9. Clustering results with different dictionary size (Left) and different number of sequential neighbors (Right) on Keck dataset.

5. Conclusions

We propose a temporal subspace clustering (TSC) approach in this paper. TSC considers the sequential information in time-series data by virtue of a temporal Laplacian regularization term. In addition, a non-negative dictionary is learned to form an expressive encoding space. We design an efficient ADMM optimization algorithm to solve the problem. Experimental results on human action and gesture datasets show that TSC significantly outperforms the state-of-the-art subspace clustering methods. Specifically, our TSC approach improves the average clustering accuracy by at least 10%. In our future work, we will design algorithms to automatically find the number of clusters for time-series data. We would also apply the proposed temporal Laplacian regularization function to other temporal analysis tasks.

Acknowledgment

This research is supported in part by the National Science Foundation (NSF) CNS award 1314484, Office of Naval Research (ONR) award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, Naval Postgraduate School (NPS) award N00244-15-1-0041 and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

[1] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c. *Comm. of the ACM*, 15(9), 1972. 4

[2] F. De la Torre, J. Campoy, Z. Ambadar, and J. F. Cohn. Temporal segmentation of facial behavior. In *ICCV*, pages 1–8. IEEE, 2007. 2

[3] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797, 2009. 1, 2, 5, 6, 7

[4] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Nonparametric bayesian learning of switching linear dynamical systems. In *NIPS*, pages 457–464, 2008. 2

[5] D. Gong, G. Medioni, S. Zhu, and X. Zhao. Kernelized temporal cut for online temporal segmentation and recognition. In *ECCV*, pages 229–243. Springer, 2012. 2

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2247–2253, 2007. 5, 6

[7] H. Hu, Z. Lin, J. Feng, and J. Zhou. Smooth representation clustering. In *CVPR*, 2014. 2

[8] D. Huang, S. Yao, Y. Wang, and F. D. la Torre. Sequential max-margin event detectors. In *ECCV*, pages 410–424, 2014. 5, 6

[9] Z. Jiang, Z. Lin, and L. S. Davis. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):533–547, 2012. 5

[10] S. Jones and L. Shao. Linear regression motion analysis for unsupervised temporal segmentation of human actions. In *WACV*, pages 816–822. IEEE, 2014. 2

[11] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2001. 4

[12] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1644–1657, 2014. 1

[13] K. Li, S. Li, and Y. Fu. Early classification of ongoing observation. In *ICDM*, pages 310–319. IEEE, 2014. 1

[14] L. Li, S. Li, and Y. Fu. Learning low-rank and discriminative dictionary for image classification. *Image and Vision Computing*, 32(10):814–823, 2014. 3

[15] S. Li and Y. Fu. Low-rank coding with b-matching constraint for semi-supervised classification. In *IJCAI*, pages 1472–1478, 2013. 4

[16] S. Li and Y. Fu. Learning balanced and unbalanced graphs via low-rank coding. *IEEE Trans. Knowledge and Data Engineering*, 27(5):1274–1287, 2015. 4

[17] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010. 1, 2, 5, 6, 7

[18] G. Liu and S. C. Yan. Active subspace: toward scalable low-rank learning. *Neural Computation*, 24(12):3371–3394, 2012. 1

[19] R. S. Liu, Z. C. Lin, F. D. Torre, and Z. X. Su. Fixed-rank representation for unsupervised visual learning. In *CVPR*, pages 598–605, 2012. 1

[20] A. López-Méndez, J. Gall, J. R. Casas, and L. J. Van Gool. Metric learning from poses for temporal clustering of human motion. In *BMVC*, pages 1–12, 2012. 2

[21] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360, 2012. 1, 2, 3, 5, 6, 7

[22] M. H. Nguyen and F. D. la Torre. Maximum margin temporal clustering. In *AISTATS*, pages 520–528, 2012. 2, 5, 6, 8

[23] X. Peng, L. Zhang, and Z. Yi. Scalable sparse subspace clustering. In *CVPR*, pages 430–437, 2013. 1, 2

[24] M. W. Robards and P. Sunehag. Semi-markov kmeans clustering and activity recognition from body-worn sensors. In *ICDM*, pages 438–446, 2009. 2

[25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 4

[26] A. Talwalkar, L. W. Mackey, Y. Mu, S. Chang, and M. I. Jordan. Distributed low-rank subspace segmentation. In *ICCV*, pages 3543–3550, 2013. 1, 2

[27] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *CVPR*, 2014. 1, 2, 5, 6, 7

[28] A. Vögele, B. Krüger, and R. Klein. Efficient unsupervised temporal segmentation of human motion. *ACM SCA*, 2014. 2

[29] S. Wang, B. Tu, C. Xu, and Z. Zhang. Exact subspace clustering in linear time. In *AAAI*, pages 2113–2120, 2014. 1, 2

[30] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2):365–384, 2012. 4

[31] X. Yuan and P. Li. Sparse additive subspace clustering. In *ECCV*, pages 644–659, 2014. 2

[32] X. Zhang, F. Sun, G. Liu, and Y. Ma. Fast low-rank subspace segmentation. *IEEE Trans. Knowl. Data Eng.*, 26(5):1293–1297, 2014. 1

[33] Z. Zhang and K. Zhao. Low-rank matrix approximation with manifold regularization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1717–1729, 2013. 3

[34] X. Zhao, Y. Fu, and Y. Liu. Human motion tracking by temporal-spatial local gaussian process experts. *IEEE Trans. Image Processing*, 20(4):1141–1151, 2011. 1

[35] F. Zhou, F. D. la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(3):582–596, 2013. 2, 8

[36] V. Zografos, L. Ellis, and R. Mester. Discriminative subspace clustering. In *CVPR*, pages 2107–2114, 2013. 2