The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Learning Transferable Subspace for Human Motion Segmentation

Lichen Wang,[*] Zhengming Ding,[*] Yun Fu[*][†]
[*]Department of Electrical & Computer Engineering, Northeastern University, Boston, USA
[†]College of Computer & Information Science, Northeastern University, Boston, USA
wanglichenxj@gmail.com, allanding@ece.neu.edu, yunfu@ece.neu.edu

Abstract

Temporal data clustering is a challenging task. Existing methods usually explore data self-representation strategy, which may hinder the clustering performance in insufficient or corrupted data scenarios. In real-world applications, we are easily accessible to a large amount of related labeled data. To this end, we propose a novel transferable subspace clustering approach by exploring useful information from relevant source data to enhance clustering performance in target temporal data. We manage to transform the original data into a shared low-dimensional and distinctive feature space by jointly seeking an effective domain-invariant projection. In this way, the well-labeled source knowledge can help obtain a more discriminative target representation. Moreover, a graph regularizer is designed to incorporate temporal information to preserve more sequence knowledge into the learned representation. Extensive experiments based on three human motion datasets illustrate that our approach is able to outperform state-of-the-art temporal data clustering methods.

Introduction

Temporal data segmentation is a critical technique utilized in many real-world applications as data preprocessing process, such as natural language processing, motion analysis and action recognition. The goal is to divide a long temporal data sequence into several short, non-overlapping, meaningful and reasonable segments. Temporal segmentation can be easily concatenated with other post-processing methods to further enhance task performances. Assume there is a video sequence which contains several continuous actions. Since conventional action recognition approach is designed to recognize videos which only contain a single action. Thus, a preprocessing segmentation process is essential.

Compared with independent static data, the successive information residing in data points is a unique cue to guide the clustering algorithms. And it is effective to fully utilize the data dependency to improve clustering performance. A comprehensive survey (Keogh and Kasetty 2003) reveals that temporal data clustering is difficult due to the data dimension and complex temporal connection. Based on the categorization mentioned in (Yang and Chen 2011), there are three lines of temporal clustering methods, including model-based (Smyth 1999; Xiong and Yeung 2003), temporal-proximity-based (Keogh and Kasetty 2003) and representation-based algorithms (Dimitrova and Golshani 1995; Chen and Chang 2000). Among them, representation-based approach is the most popular one, especially the methods based on subspace clustering algorithms.

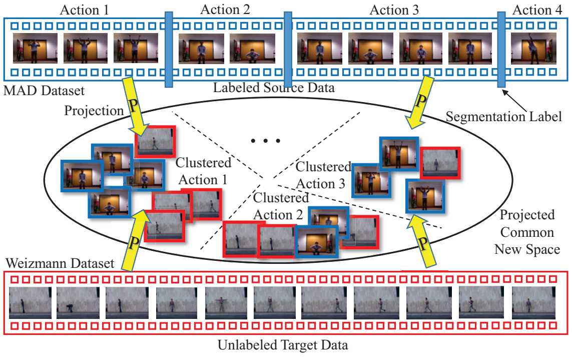


Figure 1: Framework of our approach. The information of source attempts to help target data clustering through knowledge transfer, which is used to reconstruct the target data. Furthermore, we seek a domain-invariant projection to align different distributions between the two datasets.

Subspace clustering has ideal performance in wide applications. Several representative subspace clustering methods have been proposed, including sparse subspace clustering (SSC) (Elhamifar and Vidal 2009), least-square regression (LSR) (Lu et al. 2012) as well as low-rank representation (LRR) (Liu et al. 2013), etc. The core idea of subspace clustering is learning a distinctive and low-dimensional data representations. The representations are used as input for existing clustering algorithms, which are applicable to temporal data clustering. Several methods are designed for temporal data segmentation. (Li, Li, and Fu 2015) designed a dictionary which is simultaneously updated in the learning process to obtain expressive codings. (Talwalkar et al. 2013) proposed a novel divide-and-conquer algorithm for large-scale subspace segmentation. Since these methods utilize self-representation strategy, and thus, they may hinder the clustering performance when the data are insufficient or corrupted. Furthermore, since the methods belong to unsupervised learning scenario without extra information to guide the segmentation process. Thus, it is difficult to segment data into expected groups. Supervised learning is still not an ideal solution due to high cost to generate labeled data.

Although specific labeled data are difficult to achieve, related and well-labeled data are widely available. It is reasonable to explore information from related data to help clustering process. To this end, we propose a novel transferable temporal data clustering approach, and the goal is to explore the usage of source knowledge to improve the segmentation performance in target domain. It's a challenging task due to the source and target distribution gap which would easily cause negative transfer problems. The crucial idea of our model is to learn a domain-shared subspace, where a reconstruction-based strategy is applied to guide the knowledge transfer. Therefore, the new learned representations for target domain are effective enough to do further clustering. To our best knowledge, this is the pioneer work to explore transfer learning in temporal data clustering. The contributions of this work are as follows:

- We propose a novel transfer learning based subspace clustering approach, which adapts useful information from relevant data to improve the clustering performance in target temporal data. Specifically, a reconstruction-based strategy is adopted to guide the knowledge transfer by seeking effective new representations.
- A domain-invariant projection is learned to mitigate the data distribution differences between source and target domains. Meanwhile, a graph regularizer is built to capture the temporal information of source and target for better clustering.
- Non-trivially, an optimization algorithm is designed to learn the representation and projection simultaneously and efficiently, which are used to construct a robust affinity graph for further motion segmentation.

Related Work

We discuss three related work including subspace clustering, temporal data segmentation and transfer learning.

Subspace Clustering. Subspace clustering seeks to find clusters in different and distinctive subspaces. Sparse based clustering methods (SSC) (Elhamifar and Vidal 2009) utilizes sparse constraints to learn a sparse representation of data. Least-square regression (LSR) (Lu et al. 2012) encourages a grouping effect which tends to group highly correlated data together by using the Frobenius norm. Low-rank representation (LRR) (Liu et al. 2013) analyzes the global structure in feature space and seeks the lowest-rank representation. Discriminative subspace clustering (DSC) (Zografos, Ellis, and Mester 2013) incorporates discriminative information into the model by using a quadratic classifier trained from unlabeled data. However, these methods are not well designed for temporal data segmentation. They model the data points independently but neglect the temporal relationship in the sequential data points.

Temporal Data Clustering. The goal of temporal data clustering is to divide a long temporal data into several short, non-overlapping and meaningful groups. Semi-Markov K-means clustering (Robards and Sunehag 2009) is designed to find repeated patterns residing in temporal format data. Hierarchical aligned cluster analysis (Zhou, De la Torre, and Hodgins 2013) utilizes a dynamic time alignment kernel to cluster time series data. Maximum-margin clustering method (Hoai and De la Torre 2012) simultaneously recognizes the length and position of corresponding segments. Temporal Subspace Clustering (TSC) (Li, Li, and Fu 2015) jointly learns a dictionary and representations with a regulation to decode temporal information. Basically, these temporal clustering methods dig clustering information only from the data. These approaches are difficult to robustly cluster temporal data into a reasonable, meaningful and expected result. Thus, we propose a transfer learning based segmentation approach, which borrows information from labeled extra data to facilitate the target clustering performance.

Transfer Learning. Transfer learning is a popular technique which transfers knowledge from one task to different but related tasks. A comprehensive survey can be found in (Pan and Yang 2010). According to (Pan and Yang 2010), our approach belongs to transductive transfer scenario (Zhang, Zhang, and Ye 2012; Dai et al. 2007; Ando and Zhang 2005; Blitzer, McDonald, and Pereira 2006; Daumé III, Kumar, and Saha 2010; Argyriou, Evgeniou, and Pontil 2007; Wang and Mahadevan 2008). In transductive transfer learning, the tasks of source and target are same, however, the domains of source and target are different. Domain shifting is the key problem of transfer learning since the distributions of source and target data are inconsistent. One solution is to learn a data representation, which attempts to adjust both distributions and obtains a well-aligned feature space (Zhang, Zhang, and Ye 2012; Ding and Fu 2017). Low-rank transfer learning is proposed to use incomplete multiple source data better (Ding, Shao, and Fu 2016; 2014; 2015). Our approach is different from previous work, since the methods introduced above mainly focus on classification problems and transfer label information between domains. While our approach manages to transfer clustering information for temporal data segmentation tasks.

The Proposed Approach

We explore the transfer learning idea in temporal data clustering in a semi-supervised strategy.

Learning Transferable Representation

To transfer knowledge across different domains, we adopt a reconstruction-based scheme to seek new representations, which are lying in the similar data distribution. The reconstruction strategy is shown as follows:

$$X \approx X_S Z, \tag{1}$$

where $X = [X_S, X_T] \in \mathbb{R}^{d\times n}$ is the concatenated of source and target samples points $X_S \in \mathbb{R}^{d\times n_S}$ and $X_T \in \mathbb{R}^{d\times n_T}$, each column in X represents a sample. d is the feature dimension, n_S, n_T are the sample numbers and $n = n_S + n_T$. $Z = [z_1, z_2, ..., z_n]$ is the learned representation of X, each z_i is the representation of x_i. $Z = [Z_S, Z_T] \in \mathbb{R}^{n_S\times n}$ is also the concatenation of source and target representations Z_S and Z_T, where $Z_S \in \mathbb{R}^{n_S \times n_S}$, $Z_T \in \mathbb{R}^{n_S \times n_T}$.

As mentioned above, X_S and X_T have different data distributions. If X_S is directly used for reconstruction, the learned representation Z would contain high reconstruction errors. Therefore, we jointly seek a domain-invariant projection $P \in \mathbb{R}^{r \times d}$ which reduces the distribution gap between X_S and X_T during training process, where r is the dimension of P which regulates the dimension of projected space. Thus, we refine the reconstruction strategy as follow:

$$PX \approx PX_S Z. \tag{2}$$

To solve the constraint of Eq. (2), we design a formulation based on least-square regression, and the objective function is formulated as follow:

$$\min_{P,Z} \|PX - PX_S Z\|_{\mathrm{F}}^2 + \lambda_1(\|P\|_{\mathrm{F}}^2 + \|Z\|_{\mathrm{F}}^2), \tag{3}$$

where $\|Z\|_{\mathrm{F}}$ is the Frobenius norm, and $\|Z\|_{\mathrm{F}}^2 = \sum_{i=1}^{r}\sum_{j=1}^{n}|Z_{i,j}|^2$. λ_1 is a trade-off parameter. $\|PX - PX_S Z\|_{\mathrm{F}}^2$ captures the reconstruction error, $\|P\|_{\mathrm{F}}^2$ and $\|Z\|_{\mathrm{F}}^2$ are used to constrain the variable scale and model the global subspace structure in X. Moreover, Frobenius norm is helpful to learn a more distinctive structure in Z(Lu et al. 2012). That is to say, such reconstruction can ensure that the same cluster data reconstruct the same cluster in both source and target domain to guide knowledge transfer.

Temporal Graph Regularizer

Since successive information is crucial to guide segmentation process, we design a graph regularization function $f_t(Z)$ to incorporate the temporal information. The purpose of $f_t(Z)$ is to make the neighbors of learned representation samples close to each other. In our approach, z_i is the ith column of Z which correlates to the ith sample x_i. $f_t(Z)$ would regulate its neighbors $[z_{i-s/2}, ..., z_{i-2}, z_{i-1}, z_{i+1}$, $z_{i+2}, ..., z_{i+s/2}]$ close to z_i, where s is the length of relevant frames. The expression of $f_t(Z)$ is shown as follow:

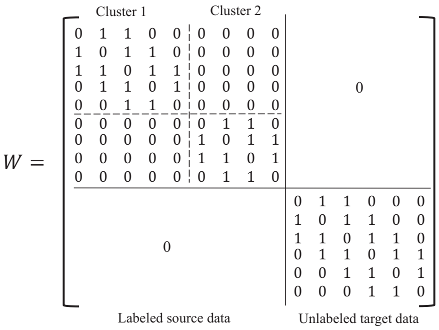
$$f_t(Z) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\|z_i - z_j\|_2^2 = \mathrm{tr}(ZL_W Z^\top), \tag{4}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace which is defined to be the elements sum on the matrix main diagonal. $L_W = D - W$ is graph Laplacian matrix (Merris 1994), $D_{ii} = \sum_{i=1}^{n} w_{ij}$, where $W \in \mathbb{R}^{n \times n}$ is the weight regularization matrix. Each element of W is shown below:

$$w_{ij} = \begin{cases} 1, & \text{if } |i-j| \leq s, l(z_i) = l(z_j), \text{for source} \\ 1, & \text{if } |i-j| \leq s, \text{for target} \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where $l(z_i)$ denotes the action label of z_i of source data.

Different from previous work (Tierney, Gao, and Guo 2014) which assumes every frame sample has temporal connections. Our approach only constrains the representation samples belonging to the same group with temporal constraint in source part. Note that there is no requirement if the class of source data would be overlapped with target data.



$$W = \left[\begin{array}{ccccc:cccc|cccccc}
0&1&1&0&0&0&0&0&0& & & & & & \\
1&0&1&1&0&0&0&0&0& & & & & & \\
1&1&0&1&1&0&0&0&0& & & & & & \\
0&1&1&0&1&0&0&0&0& & & &0& & \\
0&0&1&1&0&0&0&0&0& & & & & & \\
\hdashline
0&0&0&0&0&0&1&1&0& & & & & & \\
0&0&0&0&0&1&0&1&1& & & & & & \\
0&0&0&0&0&1&1&0&1& & & & & & \\
0&0&0&0&0&0&1&1&0& & & & & & \\
\hline
 & & & & & & & & &0&1&1&0&0&0\\
 & & & & & & & & &1&0&1&1&0&0\\
 & & & &0& & & & &1&1&0&1&1&0\\
 & & & & & & & & &0&1&1&0&1&1\\
 & & & & & & & & &0&0&1&1&0&1\\
 & & & & & & & & &0&0&0&1&1&0
\end{array}\right]$$

Figure 2: Structure of W in a simple case where $s = 3$, $n_S = 9$ and $n_T = 6$.

We assume that every target data point has temporal connections since the segmentation is unknown.

From Eq. (5), we can notice that when the distance between ith and jth sample points is less than s, the function $f_t(Z)$ would regulate the learned representation to be close. If the distance is greater than s, there is no regularization between the two samples. In order to better visualize how $f_t(Z)$ preserves label and temporal information, we illustrate a simple structure case of W in Figure 2 when $s = 3$, $n_S = 9$ and $n_T = 6$. We can see that the sequential property is preserved by the constraint weight between neighbor data points. Furthermore, we set the correlation weights between two different groups in source data as zeros. This strategy fully utilizes label information to constrain the coding result. In this work, only binary weights are used for generating W. Other graph weighting algorithms and weight values separately for temporal constraint and label constraint could be deployed to further improve the performance. Moreover, to preserve non-negative property of features, a constraint $Z \geq 0$ is added to the objective function. Putting all the terms together, our objective function is proposed as follows:

$$\begin{aligned} \min_{P,Z} \; & \|PX - PX_s Z\|_{\mathrm{F}}^2 + \lambda_1(\|Z\|_{\mathrm{F}}^2 + \|P\|_{\mathrm{F}}^2) \\ & + \lambda_2 \mathrm{tr}(ZL_w Z^\top), \\ \text{s.t.} \;\; & Z \geq 0, \;\; PXHX^\top P^\top = I, \end{aligned} \tag{6}$$

where λ_2 is utilized to balance the weights of different terms. The constraint $PXHX^\top P^\top = I$ would preserve the data variance after adaptation, which implies and introduces additional data discriminating ability into the learned model P. $X \in \mathbb{R}^{d \times n}$ is input matrix. $H = \mathrm{I} - \frac{1}{n}\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^{n \times n}$ is all-one matrix and I is an identity matrix.

After obtaining representation Z, a graph G is generated for the sequential clustering process. In order to segment temporal data accurately, the within-cluster samples are always highly correlated with each other. The definition of G is given by:

$$G(i,j) = \frac{z_i^\top z_j}{\|z_i\|_2 \|z_j\|_2}, \tag{7}$$

where z_i is an instance of representation. When G is generated, an effective conventional clustering algorithm, Normalized Cuts (Shi and Malik 2000), performs the final clustering process. The approach is shown in Algorithm 1.

Algorithm 1. *Motion Subspace Clustering*

```
Input: Source data $X_S$ and target data $X_T$, step size $\eta$,
       cluster numbers $k$, parameters $\lambda_1, \lambda_2, \alpha, s$
Output: Clustering result $Y$
1: Generate temporal constraint matrix $W$ and $L_W$
2: while not converged do
3:    Update $P_{(k+1)}$ using (13), fix other variables;
4:    Update $V_{(k+1)}$ using (12), fix other variables;
5:    Update $Z_{(k+1)}$ using (15), fix other variables;
6:    Update $\Lambda_{k+1}$, $\Lambda_{k+1} = \Lambda_k + \eta\alpha(V_{k+1} - Z_{k+1})$;
7:    $k = k + 1$
8: end while
9: Build an undirected graph $G$ based on Eq. (7)
10: Utilize NCut to cluster $k$ clusters and get index $Y$
```

Optimization

Solving Eq. (6) is challenging since it is hard to get explicit solutions. Thus, we solve the variables based on iterative strategy (ADMM) (Boyd et al. 2011). We optimize one variable by fixing others. An auxiliary variable $V \in \mathbb{R}^{n_S \times n}$ is used in the optimization process. We transform Eq. (6) as follow:

$$\begin{aligned} \min_{P,V,Z} \; & \|PX - PX_S V\|_F^2 + \lambda_1(\|V\|_F^2 + \|P\|_F^2) \\ & + \lambda_2 \text{tr}(V L_W V^\top), \\ \text{s.t.} \; & V = Z, Z \geq 0, \;\; PXHX^\top P^\top = I. \end{aligned} \tag{8}$$

Then we convert Eq. (8) to an augmented Lagrangian function (Glowinski and Le Tallec 1989)

$$\begin{aligned} \mathcal{L} = & \tfrac{1}{2}\|PX - PX_S V\|_F^2 + \lambda_1(\|V\|_F^2 + \|P\|_F^2) \\ & + \lambda_2 \text{tr}(V L_W V^\top) + \langle \Lambda, V - Z \rangle + \tfrac{\alpha}{2}\|V - Z\|_F^2, \\ \text{s.t.}, & Z \geq 0, \;\; PXHX^\top P^\top = I, \end{aligned} \tag{9}$$

where $\Lambda \in \mathbb{R}^{n_S \times n}$ is the Lagrangian multiplier, and α is a trade-off parameter. ADMM approach alternatively minimizes $\mathcal{L}$ with respect to Z, V and P. At the beginning of optimization, P and Z are initialized with random value. V and Λ are initialized with zero matrix.

Update V: By ignoring other variables, the Lagrangian equation (9) can be written as follow:

$$\begin{aligned} \min_V \;\; & \tfrac{1}{2}\|PX - PX_S V\|_F^2 + \lambda_1\|V\|_F^2 + \lambda_2 \text{tr}(V L_W V^\top) \\ & + \langle \Lambda, V - Z \rangle + \tfrac{\alpha}{2}\|V - Z\|_F^2. \end{aligned} \tag{10}$$

We assign the derivation of $\mathcal{L}$ with respect of V to zero. The equation is shown below:

$$\begin{aligned} \tfrac{\partial L}{\partial V} = \;\; & (-PX_S)^\top(-PX_S V + PX) + 2\lambda_1 V \\ & + 2\lambda_2 V L_W + \Lambda + \alpha(V - Z) = 0. \end{aligned} \tag{11}$$

Then we can get the following equation:

$$\begin{aligned} & [(PX_S)^\top(PX_S) + (\alpha + 2\lambda_1)I]V + V(2\lambda_2 L_W) \\ & \qquad = (PX_S)^\top PX - \Lambda + \alpha Z. \end{aligned} \tag{12}$$

Eq. (12) is a Sylvester equation, which is solved by Bartels-Stewart algorithm (Bartels and Stewart 1972).

Update P: By ignoring other variables we simplify the equation by converting the equation from Frobenius norm to trace format. The transformed equation is shown below:

$$\begin{aligned} P = & \mathop{\arg\min}_{PXHX^\top P^\top = I} \;\; \|PX - PX_S Z\|_F^2 + \lambda_1\|P\|_F^2 \\ = & \mathop{\arg\min}_{PXHX^\top P^\top = I} \;\; \text{tr}(P(X - X_S Z)(X - X_S Z)^\top P^\top) \\ & \qquad\qquad + \lambda_1 \text{tr}(PP^\top) \\ = & \mathop{\arg\min}_{PXHX^\top P^\top = I} \;\; \text{tr}(P[(X - X_S Z)(X - X_S Z)^\top \\ & \qquad\qquad + \lambda_1 I]P^\top). \end{aligned} \tag{13}$$

Eq. (13) can be solved by using generalized Eigen-decomposition:

$$[(X - X_S Z)(X - X_S Z)^\top + \lambda_1 I]\rho = \gamma XHX^\top \rho. \tag{14}$$

where γ is the eigenvalue of the corresponding eigenvector ρ of the generalized Eigen-decomposition formulation. $P = [\rho_0, \cdots, \rho_{p-1}]^\top$ where ρ_i is the minimum eigenvalue solutions to the eigenvalue problem

Update Z: By ignoring other variables, the function can be written as follow:

$$\min_Z \; \langle \Lambda, V - Z \rangle + \frac{\alpha}{2}\|V - Z\|_F^2. \tag{15}$$

The closed-form solution of Eq. (15) is $Z = V + \frac{\Lambda}{\alpha}$. In order to meet the non-negative constraint for Z, the update rule is shown as follow:

$$Z = F_+(V + \frac{\Lambda}{\alpha}), \tag{16}$$

where $(F_+(A))_{ij} = \max(A_{ij}, 0)$, and it regulates the non-negative value constraint and A_{ij} is an element in matrix A. The update steps are iteratively executed until the equation is convergent. The optimization process is summarized as Algorithm 1.

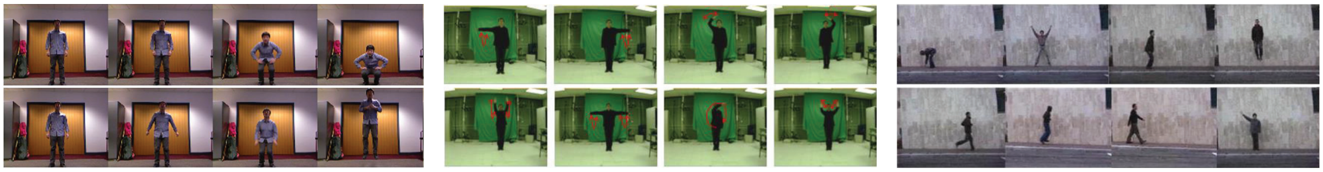
Complexity Analysis

There are two key time-consuming parts in our model during optimization. The first one is the Step 3 (Eigen-decomposition) and Step 4 (V Updating). Specifically, for step 3, it contains Eigen-decomposition, which costs $\mathbf{O}(d^3)$ for the matrix with size of $d \times d$. Step 4 updates V by using Bartels Stewart algorithm, and its complexity is $\mathbf{O}(n_S^2 n)$. Denote t as the iterations number, the total computational complexity of our approach is $\mathbf{O}(td^3 + tn_S^2 n)$. Step 3 can be reduced to $O(d^{2.376})$ using the Coppersmith-Winograd algorithm (Coppersmith and Winograd 1987). Step 4 mainly suffers from the size of source data. Generally, we could seek a more effective basis from source to reduce its size. To this end, our approach is more applicable in real-world applications.

Experiment

Temporal Datasets

Multi-modal Action Detection (MAD) Dataset (Huang et al. 2014) contains multi-modal actions of 20 subjects recorded by Microsoft Kinect in three formats. First is regular RGB image in resolution of 640 × 480. Second is 3D depth image. The third is human skeleton information. Each subject performs 35 actions in two different indoor environments. Figure 3(a) shows frame samples of MAD dataset.



(a) MAD Dataset

(b) Keck Dataset

(c) Weizmann Dataset

Figure 3: Example frames of three human motion datasets.

Table 1: Clustering accuracies and NMI of compared methods on three human motion datasets. Names in brackets denote the respective source dataset. The best and second best clustering results are denoted by bold and italic respectively.

(a) Results on MAD Dataset

Method	Acc	NMI
LRR	0.2397	0.2249
OSC	0.4327	0.5589
SSC	0.3817	0.4758
LSR	0.3979	0.3667
TSC	0.5556	0.7721
TSC (Weiz)	0.5418	0.7684
TSC (Keck)	0.5473	0.7691
Ours (Weiz)	*0.5736*	*0.8202*
Ours (Keck)	**0.5792**	**0.8286**

(b) Results on Keck Dataset

Method	Acc	NMI
LRR	0.4297	0.4862
OSC	0.4393	0.5931
SSC	0.3137	0.3858
LSR	0.4894	0.4548
TSC	0.4781	0.7129
TSC (MAD)	0.4653	0.6935
TSC (Weiz)	0.4548	0.6862
Ours (MAD)	*0.5395*	**0.8049**
Ours (Wei)	**0.5485**	*0.7928*

(c) Results on Weizmann Dataset

Method	Acc	NMI
LRR	0.3638	0.4382
OSC	0.5216	0.7047
SSC	0.4576	0.6009
LSR	0.5091	0.5093
TSC	*0.6111*	0.8199
TSC (MAD)	0.5961	0.8032
TSC (Keck)	0.5931	0.7971
Ours (MAD)	**0.6208**	**0.8509**
Ours (Keck)	0.6030	*0.8326*

Keck Gesture Dataset (Jiang, Lin, and Davis 2012) consists of 14 different actions from military signals. The resolution is 640×480. Three subjects preforms the 14 gestures and each action is repeatedly preformed 3 times by each subject. Thus, $3 \times 3 \times 14 = 126$ human action videos are obtained. The actions are captured by a fixed position camera with static and simple background. The frame samples are shown in Figure 3(b).

Weizmann Dataset (Gorelick et al. 2007) contains 90 video sequences include 10 different actions performed by nine subjects in outdoor environments. The resolution is 180×144 with the frame rate of 50 fps. The subjects preform ten regular actions including run, walk, skip and so on. The frame samples are shown in Figure 3(c).

Experimental Setup

We extract low-level HoG features (Dalal and Triggs 2005) and obtain a 324-dimensional feature vector from each frame. Since different datasets contain different number of actions, to make segmentation results comparable across different datasets, we randomly choose ten actions performed by the same subject from each dataset. In evaluation, we utilize 5 randomly selected sequences in source datasets, and report the average performance. Moreover, since Weizmann and Keck datasets contain only a single action per sequence, we follow the setting of (Hoai and De la Torre 2012) which concatenates single action from the same subject to generate a long sequential data, each data contains 10 actions with 1000-3000 frame samples. The parameter values λ_1 and λ_2 are set to be 0.015 and 12 as the default, the correlated frame distance $s = 9$ and the projection size $r = 80$. Parameter sensitivity is analyzed later in this section. Our approach is compared with five state-of-the-art methods listed below:

- **Low-Rank Representation (LRR)** (Liu et al. 2013) seeks the representations of the lowest rank among all candidates. LRR obtains the global structure from the data. And it is a more effective approach for subspace segmentation.
- **Ordered Subspace Clustering (OSC)** (Tierney, Gao, and Guo 2014) proposes to explicitly enforce the temporal data representation to be close and achieves the best performance for clustering data.
- **Sparse Subspace Clustering (SSC)** (Elhamifar and Vidal 2009) assumes that each point has a sparse representation, and enforces a sparse constraint to learn a sparse representation.
- **Least Square Regression (LSR)** (Lu et al. 2012) is an efficient clustering approach. By using the Frobenius norm, LSR lets the highly correlated data to be effectively grouped together.
- **Temporal Subspace Clustering (TSC)** (Li, Li, and Fu 2015) proposes a temporal Laplacian regularization as well as jointly learns a dictionary to obtain expressive and distinctive codings for time series data.

We evaluate the compared methods by running the codes provided by authors. All parameters are tuned to achieve the best performance. Both Normalized Mutual Information (NMI) and Accuracy (ACC) are utilized to test the clustering performance of the methods.

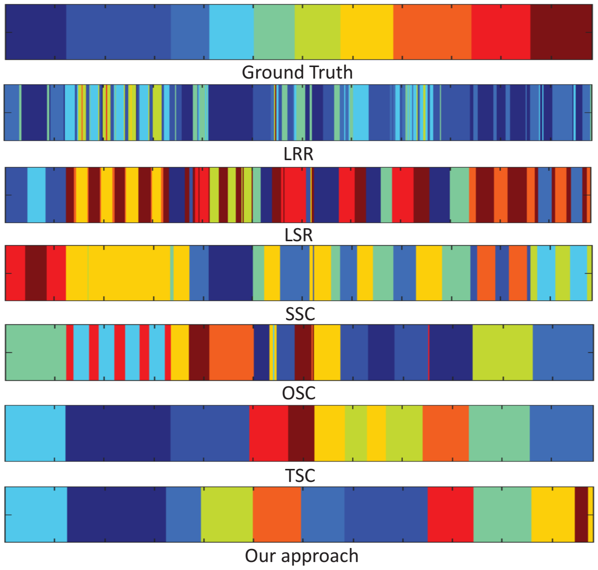


Figure 4: Visualization of clustering result. Ten colors denote ten different actions. The results illustrate that LRR, LSR and SSC cannot segment temporal data accurately since no temporal information preserved in the model. The results of OSC and TSC are better but still contain fragments and inaccurate part. While our approach contains less fragments and sensitive to similar actions.

Performance Comparison

In our approach, we set one dataset as source and another one as target. Since there are three datasets for evaluation, each target dataset is segmented based on other two datasets as source data respectively. Results are shown in Table 1. For other methods, since they cannot utilize source information, we only input the target data and get the segmentation results. From Table 1 we can see that our approach outperforms other methods. Compared with the second best approach, TSC, our approach has averagely 5.3% and 7.1% improvement in accuracy and NMI.

We visualize one MAD video segmentation results of our approach and other methods in Figure 4. Different colors indicate different segments (i.e., clusters). We observe that SSC, LRR and LSR segment a lot of fragments in one action sequence which is not reasonable. One reason is that these methods do not preserve temporal information between adjacent points. OSC and TSC have better performance since both methods are designed for temporal data segmentation. However, they still suffer from data variations. OSC still exists fragments situation and TSC is not sensitive in segmenting similar temporal data. However, since our approach utilizes extra source data and temporal graph regularization, it is able to obtain continuous segments and distinguish similar but different actions. Therefore, our approach achieves better and more accurate results than other methods.

Our approach fails to achieve significant improvement in Weizmann dataset. We find out that there are various differences in Weizmann Dataset. First, since the action sequences are directly concatenated, the subject positions and motion patterns changed suddenly between two sequences in some cases, such as from moving to left switch to moving to right. Second, since the video captured in different time, the illumination situations in videos are inconsistent. Furthermore, the subject visual appearance also changes such as clothes and hair styles. However, in other two datasets the subject performs the action without any appearance changing, and the illumination is consistent in all actions which are similar to real-world applications. These differences give compared approaches distinctive cues while our method can only obtain limited benefits from source data. Thus the improvement of our approach is not significant compared with other methods due to the data is distinctive already.

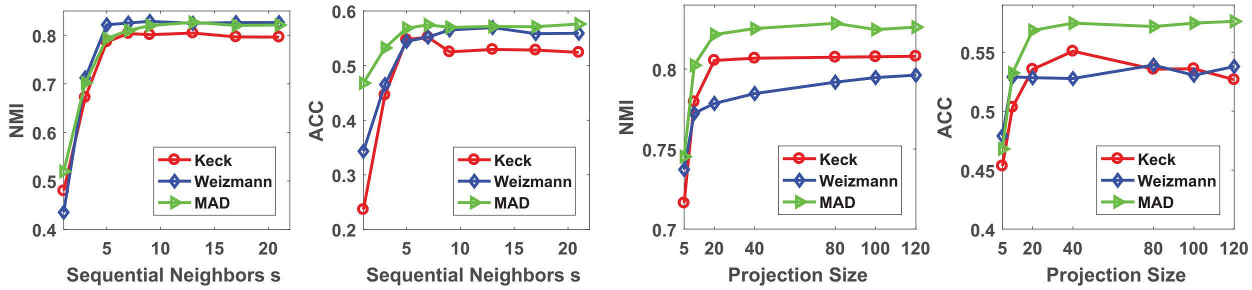
Table 2: Result of our model in manipulated dictionaries. **Ours-1**: Original dictionary. **Ours-2**: Shuffle dictionary sequence. **Ours-3**: Add noise in dictionary.

Dictionary	ACC	NMI
Ours-1	**0.5736**	**0.8202**
Ours-2	0.5710	0.8188
Ours-3	0.5251	0.7598

Source Data Analysis

To fairly compare with different methods, we concatenate source and target data together as input to the second best approach, TSC. This aims to demonstrate that our improvement is not directly from data augmentation. Table 1 shows the result. TSC performance drops slightly (about 1%) and the result denotes two facts. First, simply increasing data samples cannot improve clustering performance. Second, the performance would reduce if the model cannot align the distribution gap of source and target appropriately. It demonstrates that our approach is able to well align different distributions and transfer useful information to improve segmentation performance.

To prove that source data are crucial to improve segmentation performance, we manipulate the dictionary and test the model. We set Weizmann as source and MAD as target. First, original source is set as baseline (Ours-1). Second, we randomly shuffle the sequence of dictionary (Ours-2). Third, a Gaussian distribution noise $x_n \sim N(0,1)$ is added to dictionary (Ours-3). The result is shown in table 2. We observe that shuffling the dictionary sequence has less than 0.2% negative influence on segmentation performance. However, the result drops more than 7 % in accuracy and NMI when adding Gaussian noise. Figure 6 further shows the performance with different scales of noise added to dictionary. We observe that the model achieves the best performance without noise, and as the noise increases, the performance drops significantly and then gradually becomes stable. The results denote that clean and good structure of source data can help to learn more distinctive representations and improve segmentation performance. Adding noise would weaken or destroy the structure information and weaken the reconstruction performance, thus it reduces the clustering performance.



(a) (b)

Figure 5: (a) Performance in different number of sequential neighbors s. (b) Performance in different project size r.

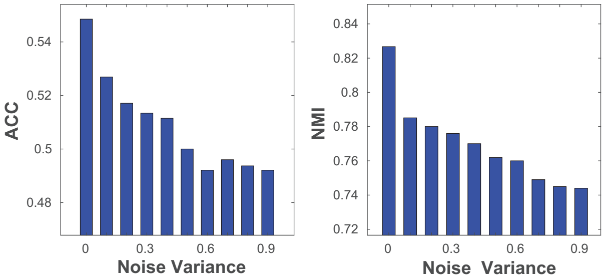


Figure 6: Segmentation result when different variances of Gaussian noise is added to dictionary.

Table 3: Results based on modifications of our model by changing the reconstruction term.

Modified Model	ACC	NMI
Model-1	**0.5415**	**0.8111**
Model-2	0.5224	0.8066
Model-3	0.5268	0.7736
Model-4	0.4530	0.6866

Model Variant Analysis

To verify the effectiveness of each term in our model, we change the first reconstruction term in Eq. (3), and the results are shown in Table 3. We set MAD as source and Keck as target. First, we test the original model as baseline in Model-1. Second, we remove P and obtain Model-2: $\|X - X_S Z\|_F^2$. Third, we remove P and concatenate X_S and X_T as dictionary in Model-3: $\|X - XZ\|_F^2$. Fourth, we set both source and target data as dictionary in Model-4: $\|PX - PXZ\|_F^2$. We observe that when projection P is removed, the segmentation accuracy drops more than 3% which indicates that P is essential to connect both source and target data, and to improve the performance. When we concatenate X_T as dictionary, the segmentation performance drops significantly. We assume the reason for this situation is that since X_T exists in the dictionary, so target data could be represented by X_T instead of X_S, and X_S would have negative influence for the segmentation performance. From the experimental results, we conclude that every term in our approach is required and contributes to improve the segmentation performance.

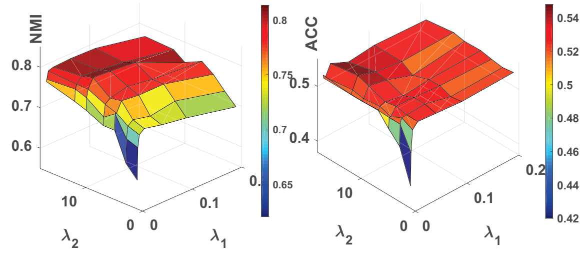


Figure 7: Parameter sensitivity tested on Keck dataset. Left and right are visualization result of NMI and ACC metrics with different value of λ_1 and λ_2.

Parameter Analysis

We use different values to test the parameter sensitivity of our model on Keck and MAD as source. Figure 5(a) shows that when $s \geqslant 5$, the performance would be accurate and stable. λ_1 constraints P and Z scale, and λ_2 regulates the temporal data constraint. Figure 5(b) shows the parameter sensitivity of r. It indicates that if $r \geqslant 20$, the result is relatively stable even though there is a little fluctuation as r increases. s is also a major parameter in our model. The result in Figure 7 demonstrates that our approach can get an accurate result when λ_1 in the range of $[0.02, 0.1]$ and λ_2 in the range of $[9, 20]$. The ranges of λ_1 and λ_2 are wide.

Conclusion

We introduced a novel transfer learning based temporal data clustering approach in this paper. This approach adapted useful information from relevant source data, and transferred knowledge for target temporal data segmentation tasks. Specifically, a reconstruction-based strategy was adopted to guide the knowledge transfer. A domain-invariant projection was learned to mitigate the data distribution differences, and a graph regularizer was built to capture the temporal information. Our approach outperformed state-of-the-art temporal subspace clustering methods on three human motion datasets. The results also demonstrated that our approach is robust, accurate and parameter insensitive.

Acknowledgments

This research is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.

References

Ando, R. K., and Zhang, T. 2005. A high-performance semi-supervised learning method for text chunking. In *ACL*, 1–9.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2007. Multi-task feature learning. *NIPS* 19:41.

Bartels, R. H., and Stewart, G. 1972. Solution of the matrix equation ax+ xb= c [f4]. *Communications of the ACM* 15(9):820–826.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *ACL EMNLP*, 120–128.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Chen, W., and Chang, S.-F. 2000. Motion trajectory matching of video objects. In *Storage and Retrieval for Media Databases*, volume 3972, 544–553.

Coppersmith, D., and Winograd, S. 1987. Matrix multiplication via arithmetic progressions. In *ACM Symposium on Theory of Computing*, 1–6.

Dai, W.; Xue, G.-R.; Yang, Q.; and Yu, Y. 2007. Co-clustering based classification for out-of-domain documents. In *ACM SIGKDD*, 210–219.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *IEEE CVPR*, volume 1, 886–893.

Daumé III, H.; Kumar, A.; and Saha, A. 2010. Frustratingly easy semi-supervised domain adaptation. In *ACL NLP*, 53–59.

Dimitrova, N., and Golshani, F. 1995. Motion recovery for video content classification. *ACM TOIS* 13(4):408–439.

Ding, Z., and Fu, Y. 2017. Robust transfer metric learning for image classification. *IEEE TIP* 26(2):660–670.

Ding, Z.; Shao, M.; and Fu, Y. 2014. Latent low-rank transfer subspace learning for missing modality recognition. In *AAAI*, 1192–1198.

Ding, Z.; Shao, M.; and Fu, Y. 2015. Deep low-rank coding for transfer learning. In *IJCAI*, 3453–3459.

Ding, Z.; Shao, M.; and Fu, Y. 2016. Transfer learning for image classification with incomplete multiple sources. In *IEEE IJCNN*, 2188–2195.

Elhamifar, E., and Vidal, R. 2009. Sparse subspace clustering. In *IEEE CVPR*, 2790–2797.

Glowinski, R., and Le Tallec, P. 1989. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. SIAM.

Gorelick, L.; Blank, M.; Shechtman, E.; Irani, M.; and Basri, R. 2007. Actions as space-time shapes. *IEEE TPAMI* 29(12):2247–2253.

Hoai, M., and De la Torre, F. 2012. Maximum margin temporal clustering. In *AI and Statistics*, 1–9.

Huang, D.; Yao, S.; Wang, Y.; and De La Torre, F. 2014. Sequential max-margin event detectors. In *ECCV*, 410–424.

Jiang, Z.; Lin, Z.; and Davis, L. 2012. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE TPAMI* 34(3):533–547.

Keogh, E., and Kasetty, S. 2003. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery* 7(4):349–371.

Li, S.; Li, K.; and Fu, Y. 2015. Temporal subspace clustering for human motion segmentation. In *IEEE ICCV*, 4453–4461.

Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; and Ma, Y. 2013. Robust recovery of subspace structures by low-rank representation. *IEEE TPAMI* 35(1):171–184.

Lu, C.-Y.; Min, H.; Zhao, Z.-Q.; Zhu, L.; Huang, D.-S.; and Yan, S. 2012. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, 347–360.

Merris, R. 1994. Laplacian matrices of graphs: a survey. *Linear algebra and its applications* 197:143–176.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE TKDE* 22(10):1345–1359.

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

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE TPAMI* 22(8):888–905.

Smyth, P. 1999. Probabilistic model-based clustering of multivariate and sequential data. In *AI and Statistics*, 299–304.

Talwalkar, A.; Mackey, L.; Mu, Y.; Chang, S.-F.; and Jordan, M. I. 2013. Distributed low-rank subspace segmentation. In *IEEE ICCV*, 3543–3550.

Tierney, S.; Gao, J.; and Guo, Y. 2014. Subspace clustering for sequential data. In *IEEE CVPR*, 1019–1026.

Wang, C., and Mahadevan, S. 2008. Manifold alignment using procrustes analysis. In *ACM ICML*, 1120–1127.

Xiong, Y., and Yeung, D.-Y. 2003. Mixtures of arma models for model-based time series clustering. In *IEEE ICDM*, 717–720.

Yang, Y., and Chen, K. 2011. Temporal data clustering via weighted clustering ensemble with different representations. *IEEE TKDE* 23(2):307–320.

Zhang, C.; Zhang, L.; and Ye, J. 2012. Generalization bounds for domain adaptation. In *NIPS*, 3320–3328.

Zhou, F.; De la Torre, F.; and Hodgins, J. K. 2013. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE TPAMI* 35(3):582–596.

Zografos, V.; Ellis, L.; and Mester, R. 2013. Discriminative subspace clustering. In *IEEE CVPR*, 2107–2114.