Dual-Side Auto-Encoder for High-Dimensional Time Series Segmentation

Yue Bai, Lichen Wang, Yunyu Liu, Yu Yin, Yun Fu
Northeastern University, Boston, USA
{bai.yue, wang.lich, liu.yuny, yin.yu1} @northeastern.edu, yunfu@ece.neu.edu

***Abstract*—High-dimensional time series segmentation aims to segment a long temporal sequence into several short and meaningful subsequences. The high-dimensionality makes it challenging due to the complicated correlations among the sequential features. A large number of labeled data is required in existing supervised methods, and unsupervised methods mainly deploy clustering approaches, which are sensitive to outliers and hard to guarantee high performance. Also, most existing methods mainly rely on hand-craft features to deal with regular time series segmentation and achieve promising results. However, these approaches cannot effectively handle high-dimensional time series and will result in a high computational cost. In our work, we propose a novel unsupervised representation learning framework called Dual-Side Auto-Encoder (DSAE). It mainly focuses on high-dimensional time series segmentation by effectively capturing the temporal correlative patterns. Specifically, a single-to-multiple auto-encoder is designed to capture local sequential information. Besides, a long-shot distance encoding strategy is proposed. It aims to explicitly guide the learning process to obtain distinctive representations for segmentation. Furthermore, the long-short distance strategy is also executed in the decoded feature space, which implicitly directs the representation learning. Substantial experiments on six datasets illustrate the model effectiveness[1].**

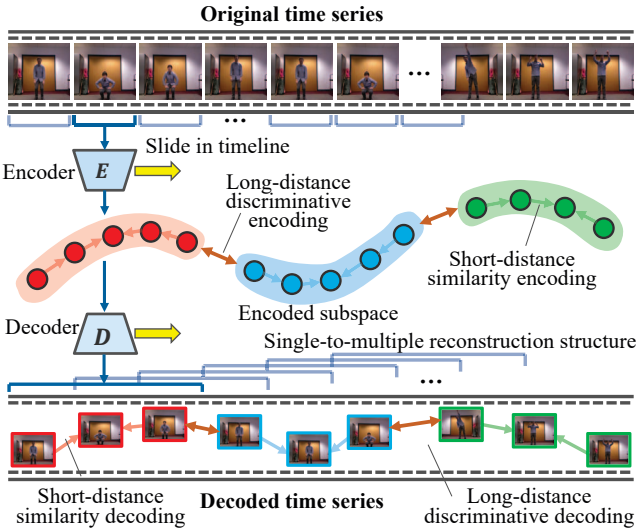

Figure 1: The framework of our DSAE. The basic framework is the single-to-multiple auto-encoder. The long-short constraint is applied on both encoder and decoder as a dual-side structure.

I. Introduction

Time series segmentation aims to segment a long time series into several short and meaningful divisions. Most real-world time series data is collected continuously without further processing. Hence, it is an indispensable preprocessing step for a wide range of downstream applications. For example, real-world videos usually consist of tens or even hundreds of actions. However, most existing video-based algorithms (e.g., action recognition [1]) are designed for handling videos which only contain a single action. Hence, the row video needs to be segmented into several short clips.

Unsupervised time series segmentation is a challenging task [2], [3]. It aims to explore efficient clustering methods to gather coherent representations into one cluster without supervision. Different from static data, time series contains more temporal correlations which is critical for clustering tasks. In addition, the high dimensionality of time series significantly increases the difficulty of segmentation [4].

As a summary, there are three main challenges for unsupervised high-dimensional time series segmentation: 1) How to extract complicated temporal and structural connections; 2) How to handle the complex correlations in the high-dimensional features space; 3) How to make accurate segmentation in an unsupervised scenario. Effectively exploring the dependencies among temporal sequence is the key factor for time series segmentation, and existing approaches can be categorized into three groups [5]: temporal-proximity-based [4], model-based [6], and representation-based [7] algorithms. Besids, leveraging the transfer learning technique, extra temporal knowledge is explored to achieve better performance in [8]. However, these methods have several drawbacks. First, most methods follow conventional optimization strategies that require computational costly algorithms (e.g., eigen decomposition). Second, some methods belong to the transductive scenario where the trained/optimized model cannot be used in new/unseen data. Third, most approaches globally consider temporal information while ignoring the trivial local details which leads to low performance.

In our work, we propose a novel Dual-Side AutoEncoder (DSAE) (Fig. 1) to handle the above issues. A specifically designed single-to-multiple auto-encoder is firstly proposed, which reconstructs multiple neighbors of the input feature and preserves the temporal local information. Moreover, a long-short encoding strategy is proposed, which includes

[1]Code will be released at https://github.com/yueb17/HTSS

This research is supported by the U.S. Army Research Office Award W911NF-17-1-0367.

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00102

similar and distinctive constraints. It explicitly guides the model to obtain discriminative representations for clustering. Further, the long-short constraints are also deployed on the decoding side to direct the learning process implicitly. The main contributions of our work are listed as below:

- A novel single-to-multiple auto-encoder is proposed. It extracts the temporal structural knowledge from multiple neighbors of the input feature to enhance the learning process and obtain effective representations.
- A long-short constraint is designed to obtain low-variation representations of close time points while diversifying the representations of the long-distance time points. It effectively enhances the representation learning and benefits the segmentation task.
- A dual-side structure is achieved by employing the long-short constraint on both encoding and decoding to explicitly and implicitly guide the representation learning. It relaxes the strict reconstruction assumption of conventional auto-encoder and helps to obtain discriminative representations for time series segmentation.

II. Related Work

A. Temporal Clustering

Temporal clustering techniques can be widely used in many applications such as context recognition, stock data mining [9], and action recognition [10]. Hierarchical aligned cluster analysis [11] proposes a dynamic kernel for time alignment to cluster temporal data. Semi-Markov K-means clustering algorithms [12] extracts repetitive temporal patterns contained in time series data. The maximum-margin clustering method [13] identifies the length and position of each temporal segment simultaneously. A temporal subspace learning method is utilized [14] to learn a dictionary and representations jointly with a regularization term. The correlative patterns are considered in [15] to measure the time series similarity and cluster the sequence. The transfer learning techniques are utilized [8] to explore temporal knowledge from extra time series information. The low-rank constraint is included in the learning procedure for achieveing higher performance [7]. However, these methods utilize conventional algorithms with high computational cost and ignore temporal local correlations. Our DSAE framework considers both local and global temporal patterns for deriving effective representations.

B. Subspace Clustering

Most existing relevant methods are based on subspace learning. They aim to project the original data into an informative and distinctive subspace for clustering [16]–[19]. Sparse subspace clustering (SSC) [20] utilizes a sparse constraint for subspace learning. Least-square regression (LSR) [21] proposes a grouping effect for clustering data with high correlations based on the Frobenius norm. The global structure of feature space is explored in low-rank representation (LRR) [22] and further used to acquire lowest-rank representations. Robust subspace clustering (RSC) [23] aims to find a multi-subspace representation to achieve robust learning process. However, most of above methods are not specifically proposed for temporal data segmentation. Further, most of them are based on conventional computational methods such as eigen-decomposition, which result in a high computational cost.

C. Auto-encoder for Temporal Data

Many auto-encoder based frameworks have been proposed for a wide range of temporal data applications. A deep auto-encoder [24] is designed to use multi-modal fusion during feature extraction for pose recovery task. A convolutional auto-encoder is proposed for unsupervised learning temporal manifolds [25]. Action forecasting is achieved by using conditional variational auto-encoders as a generative model based on the static scene understanding [26]. A stacked denoising auto-encoder is well designed for temporal data classification based on representation learning [27]. However, most existing auto-encoder structures are supervised learning methods. In our work, we design an unsupervised auto-encoder based framework to acquire informative representations for high-dimensional time series segmentation.

III. Methodology

Let $X \in \mathbb{R}^{T\times f}$ be a time series sample, where T denotes the length and f represents the feature dimension. Our DSAE contains three major parts: 1) *single-to-multiple auto-encoder*; 2) *long-short encoding*; 3) *long-short decoding*.

A. Single-to-Multiple Auto-Encoder

Existing auto-encoder structures mainly focus on reconstructing the original input while cannot preserve the local temporal information, which is crucial for time series segmentation. To this end, we propose a single-to-multiple auto-encoder structure to preserve the local temporal correlations. It sets the reconstruction targets as multiple neighbors of the original input and consists of a single-point encoder E_e and a multi-point decoder E_d formulated as follows:

$$\begin{aligned} h_e^t &= E_e(X^t), \\ h_d^t &= E_d(h_e^t), \end{aligned} \tag{1}$$

where $X^t \in \mathbb{R}^f$ denotes the input feature at time t. h_e^t denotes the hidden representation, h_d^t represents the reconstruction output. E_e and E_d are encoder and decoder, respectively. They are achieved by dense layers with ReLU activations. We set the original input as well as its neighbors as reconstruction targets achieved by following objective:

$$L_a^t = \sum_{k=t-w}^{t+w} \|h_d^t - X^k\|_{\mathrm{F}}^2, \tag{2}$$

where $\|\cdot\|_{\mathrm{F}}^2$ denotes l_2-norms. $w = (W-1)/2$ and W denotes the number of reconstruction neighbors. In addition, for the u-th and v-th points at the beginning and end of the whole time series, where $1 \leq u \leq w$ and $T-w \leq v \leq T$, we only consider the first $[1, u+w]$ and last $[T-v-w, T]$ points as neighbors. In this way, the hidden representation of one time point is guided to preserve local temporal information from its neighbors. For the whole time series, we have

$$L_a = \sum_{t=1}^{T} L_a^t. \tag{3}$$

We achieve our single-to-multiple auto-encoder via Eqs. 1-3.

B. Long-Short Encoding

Capturing discriminative patterns is decisive to achieve higher segmentation performance. It requires the hidden representations to be similar in the same segment and be distinguishing among different segments. To this end, we design a long-short encoding strategy to enhance representation learning. It consists of two constraints: 1) short-distance similar constraint and 2) long-distance distinctive constraint. The former one is given by

$$L_l = \sum_{t=1}^{T} \sum_{k=t-s}^{t+s} \|h_e^t - h_e^k\|_{\mathrm{F}}^2, \tag{4}$$

where h_e^t is the hidden representation and h_e^k denotes the neighbor representations around h_e^t which are targets to add short-distance constraint. s is the one-side constraint length. The inner summation term denotes the constraint at time t. The outer summation indicates the constraint on the whole time series. Our short-distance constraint considers the local similarity to increase the hidden representation smoothness.

On the other hand, the long-distance distinctive constraint aims to make the representations in different segments distinctive with each other, which is formulated by

$$L_g = \sum_{t=1}^{T} \Big(\sum_{k=t-q-s}^{t-q} \|h_e^t - h_e^k\|_{\mathrm{F}}^2 + \sum_{k=t+q}^{t+q+s} \|h_e^t - h_e^k\|_{\mathrm{F}}^2 \Big), \tag{5}$$

where the two summation terms in the bracket are the two sides constraint at time t. q is the distance between the t-th representation and the targets of long-distance constraint. s is the range of constraint. This strategy makes the far away representations distinctive and prevents the dispersive temporal representations being clustered into the same segment.

The complete long-short encoding is to minimize L_l and maximize L_g simultaneously, which is formulated as follow:

$$L_h = L_l - \theta_h L_g, \tag{6}$$

where L_h is the total loss and θ_h is the trade-off parameter.

C. Long-Short Decoding

The long-short encoding strategy explicitly guides the learning process via adding constraint on the hidden representations. In the auto-encoder structure, the hidden representation will be decoded to reconstruct the input. Hence, the feature distribution of the reconstructions also affects the hidden representations. To this end, we further add this long-short strategy on decoding to implicitly direct the representation learning. Similar to the encoding constraint, the long-short decoding can be separated into two parts: the short-distance similar constraint and the long-distance distinctive constraint. The short-distance constraint can be formulated as follows:

$$L_m = \sum_{t=1}^{T} \sum_{k=t-s}^{t+s} \|h_d^t - h_d^k\|_{\mathrm{F}}^2, \tag{7}$$

where h_d^t denotes t-th decoding output and h_d^k denotes the neighbors of h_d^t. s denotes the range of constraint. On the other hand, the long-distance constraint is given by:

$$L_n = \sum_{t=1}^{T} \Big(\sum_{k=t-q-s}^{t-q} \|h_d^t - h_d^k\|_{\mathrm{F}}^2 + \sum_{k=t+q}^{t+q+s} \|h_d^t - h_d^k\|_{\mathrm{F}}^2 \Big), \tag{8}$$

where q denotes the distance between constraint and t-th output and s indicates the range of constraint. The long-short decoding is achieved by minimizing L_m and maximize L_n with the trade-off parameter θ_r, which is given by

$$L_r = L_m - \theta_r L_n. \tag{9}$$

We combine the single-to-multiple auto-encoder and the pair of long-short constraints to achieve the Dual-Side Auto-Encoder (DSAE), which is achieved by minimizing the following objective:

$$L = L_a + \lambda_h L_h + \lambda_r L_r, \tag{10}$$

where L denotes the final objective. λ_h and λ_r are two trade-off parameters. Our DSAE framework aims to obtain informative representations from the original high-dimensional time series input. We follow the previous work [7], and forward the learned representations H_e to the down-stream NCuts [28] clustering algorithm for the final segmentation.

IV. Experiments

A. Datasets

We use 6 real-world human action datasets, which contain complicated high-dimensional time series data to evaluate our model. **Multi-Modal Action Detection Dataset (MAD)** [33] contains multi-modal human actions videos performed by 20 subjects and each subject performs 35 actions. We use RGB modal for evaluation. **UT-Interaction Dataset (UT)** [34] includes 20 action videos. Each one contains 6 categories of human interactive actions such as punching and pushing. **Weizmann Dataset (Weiz)** [35] contains 90

Table I: Segmentation Performance

Datasets	MAD		Keck		Weizmann		UT		ChaLearn16		ChaLearn14	
Methods	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
KMS [29]	0.3541	0.4188	0.3510	0.4553	0.4081	0.5562	0.4712	0.5677	0.4331	0.3221	0.4523	0.5968
KMD [30]	0.3226	0.3914	0.3970	0.4702	0.4441	0.5289	0.5122	0.5108	0.4160	0.2946	0.5078	0.6270
SPE [31]	0.3639	0.4369	0.3886	0.4744	0.4127	0.5435	0.4477	0.4894	0.4066	0.2721	0.4359	0.5877
LRR [22]	0.2397	0.2249	0.4297	0.4862	0.3638	0.4382	0.4162	0.4051	0.3239	0.1423	0.4137	0.5033
OSC [32]	0.4327	0.5589	0.4393	0.5931	0.5216	0.7047	0.5846	0.6877	0.4025	0.3346	0.4759	0.7189
SSC [20]	0.3817	0.4758	0.3137	0.3858	0.4576	0.6009	0.4389	0.4998	0.3867	0.2108	0.4853	0.6788
LSR [21]	0.3979	0.3667	0.4894	0.4548	0.5091	0.5093	0.5183	0.4322	0.3917	0.1973	0.5913	0.5817
TSC [14]	**0.5556**	0.7721	0.4781	0.7129	0.6111	**0.8199**	0.5340	0.7593	0.5414	0.6000	0.5373	0.7861
TSS [8]	0.4652	0.6987	0.4929	0.7342	0.6101	0.7112	0.5541	0.7114	0.5385	0.6410	0.3788	0.6602
LTS [7]	0.4833	0.7268	0.5128	0.7365	0.6155	0.7273	0.5629	0.7223	0.5359	0.5369	0.3734	0.5684
Ours	0.5548	**0.7734**	**0.5753**	**0.7407**	**0.6199**	0.7879	**0.6006**	**0.7950**	**0.5905**	**0.6673**	**0.6055**	**0.8515**

videos including 10 actions such as running and walking performed by 9 subjects. **Keck Gesture Dataset (Kect)** [36] has 14 classes of actions performed by 3 subjects. All actions are based on military signals. **ChaLearn 2014** [37] contains 14000 gesture samples . They are about 20 classes of sign gesture for Italian vocabulary performed by different users. **ChaLearn 2016** [38] contains 47933 gesture samples in 22535 RGB-D videos with one or more gestures. 249 gesture classes are performed by 21 subjects. We use RGB videos for model evaluation.

B. Baseline Methods

We first use three conventional clustering methods for comparisions: 1) K-means (KMS) [29], 2) K-medoids (KMD) [30], and 3) Spectral Clustering (SPE) [31]. Next, we utilize four recently published representation learning methods: 1) Sparse Subspace Clustering (SSC) [20], 2) Least Square Regression (LSR) [21], 3) Low-Rank Representation (LRR) [22], and 4) Ordered Subspace Clustering (OSC) [32]. We also introduce three state-of-the-art approaches: 1) Temporal Subspace Clustering (TSC) [14] utilizes a Laplacian regularization on temporal domain and a learned dictionary simultaneously to acquire distinctive temporal representations, 2) Transfer Subspace Segmentation (TSS) [8] proposes a transferable temporal segmentation algorithm based on source and target datasets. It fully utilizes the auxiliary data information to boost the segmentation performance, and 3) Low-Rank Transfer Segmentation (LTS) [7] proposes a sequential graph model and adds a weighted low-rank constraint to improve the temporal data segmentation performance using a transfer learning fashion.

C. Implementation

We apply the HoG encoding [39] to obtain 324-dimensional frame-level features as time series input. Further, we standardize the data samples from different datasets into the same format. Concretely, we follow [13] to obtain long video containing 10 actions for Weizmann and Keck datasets. Samples in MAD dataset contain more than 10 actions, we cut them to 10 actions per video. UT dataset contains 6 actions per video, we directly use it for evaluation. For ChaLearn2016 dataset, we pick samples with more than 5 actions and remove the excrescent actions to make all videos contain 5 actions. For ChaLearn2014 dataset, we pick out videos containing more than 10 actions and rearrange them to 10 actions using the same strategy.

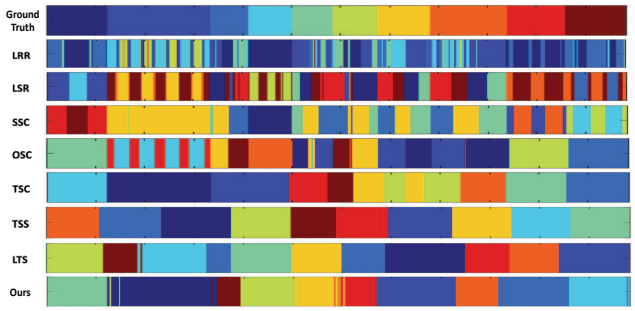

Figure 2: Visualization of segmentation results. Different colors denote different temporal clips.

As illustrated in Fig. 1, the HoG features are set as input for our DSAE framework to obtain the effective representations for clustering tasks. The single-to-multiple auto-encoder, achieved by two linear mappings with ReLU activations, and dual-side constraints synergistically coordinate to obtain the representations. Then, the NCuts cluster algorithm is employed to make final segmentation.

D. Performance Analysis

We use Normalized Mutual Information (NMI) [40] and accuracy (ACC) for model evaluation and results are shown in Table I. The first three traditional clustering algorithms obtain low clustering performances. The next four recently proposed baselines focus on subspace learning and obtain better performances. However, these methods cannot achieve state-of-the-art performances, since they are not designed for temporal data. TSC is specifically designed for temporal data segmentation and obtains competitive results. The last two methods use the transfer learning technique, which utilize auxiliary information from source data to improve target data performance. However, after adopting their experimental settings for fair comparisons, they only obtain comparable performances like other baseline methods. Different from all the baseline approaches, our proposed model takes advantage of deep neural networks to extract complicated

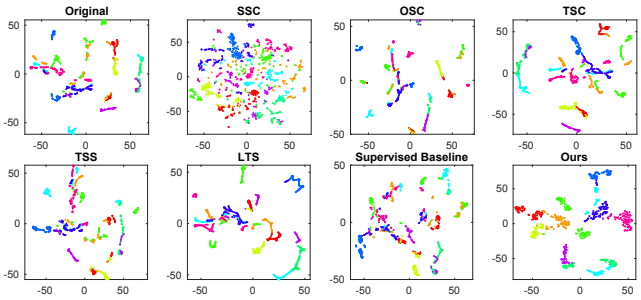

Figure 3: t-SNE visualizations for several baselines and our model

and informative patterns for clustering. It fully considers the temporal relationship residing in time series data and outperforms other methods.

The segmentation results of one Weizmann dataset sample are visualized in Fig. 2. Different colors denote different segments. The results of LSR and LRR are disordered with many fragments. SSC and OSC obtain better results, however, for many temporal sequences with rhythmed patterns (e.g. repetitive motion in human action), their results are still divided into many fragments. TSC has much better performance. However, it still generates multiple fragments in one cluster occasionally, and the boundaries of several segments are not accurate. TSS and LTS achieve more reasonable results without many redundant fragments in each segment. However, they always make mistakes on boundaries for each cluster. Our method segments the whole time series more accurately without many unreasonable fragments.

Our model is designed in an unsupervised fashion, which fully explores the latent patterns from time series itself without any other supervisions. To further illustrate the model effectiveness, we make a comparison between our model and a supervised method. We regard the HoG feature of each temporal input as a sample with the label of its corresponding segment. Next, a linear classifier is trained and the pre-trained classifier is used to extract features for both training and test sets. The extracted features are clustered via NCuts algorithm for evaluation. We use the leave-one-subject-out evaluation. The performances on three datasets are shown in Table II. We observe the considerable performance drop in supervised scenario compared with our proposed method. Our framework fully considers the temporal correlative information to obtain the effective representations which significantly improve the clustering performance. Further, our model is trained in an unsupervised scenario without a high labeling cost.

Finally, we visualize representation learning results using t-SNE [41] in Fig. 3. We pick the first video sample from the Keck dataset. We visualize the original data and the learned representations from five competitive baselines, the supervised learning classifier, and from our DSAE framework. Different colors represent different clusters. The t-SNE visualizations of these five baselines are consistent with the clustering visualization in Fig. 2. The SSC and

Table II: Supervised method v.s. our model

Methods	MAD		Keck		Weiz	
	ACC	NMI	ACC	NMI	ACC	NMI
Supervised	0.3976	0.4835	0.3855	0.4349	0.5081	0.5942
Ours	**0.5548**	**0.7734**	**0.5753**	**0.7407**	**0.6199**	**0.7879**

Table III: Ablation study on Keck dataset

S-to-M AE	Short-En	Long-En	Short-De	Long-De	ACC	NMI
×	✓	✓	✓	✓	0.5233	0.7287
✓	×	✓	✓	✓	0.2161	0.2170
✓	✓	×	✓	✓	0.4718	0.7170
✓	✓	✓	×	✓	0.5542	0.7324
✓	✓	✓	✓	×	0.5376	0.7280
✓	×	×	✓	✓	0.3603	0.4891
✓	✓	✓	×	×	0.5263	0.7364
✓	×	✓	×	✓	0.2219	0.2279
✓	✓	×	✓	×	0.4849	0.7288
✓	✓	✓	✓	✓	**0.5753**	**0.7407**

OSC cannot effectively learn the discriminative features. They have many fragments in clusters. The three most competitive methods, TSC, TSS and LTS, obtain much better results. They have fewer fragments but the cluster boundaries are not accurate compared with the ground truth. Also, we notice that some representations belonging to different segments are still gathered into one segment, which reduces the clustering performance. The representation derived from the supervised classifier is also not discriminative enough for clustering. Our model achieves the most reasonable and promising results.

E. Ablation Study

The ablations on the Keck dataset are shown in Table III. "S-to-M AE", "En" and "De" represent simple-to-multiple auto-encoder, encoding and decoding, respectively. "Short" and "Long" denote the short-distance and long-distance constraints, respectively. We conclude that S-to-M AE is an effective basic framework. The short-distance constraints play a dominant role during the learning process, while the long-distance constraints further improve the performance significantly which are also indispensable components. On the other view, the encoding constraints guide the learning process directly and provide dramatic performance improvement; while the decoding constraints implicitly direct the model to make further improvement. Our complete model achieves the best performance via all the proposed modules collaboratively working together.

V. Conclusions

We propose a novel Dual-Side Auto-Encoder (DSAE) framework for unsupervised high-dimensional time series segmentation. A single-to-multiple auto-encoder is applied for capturing temporal structural information. A long-short constraint is deployed on the encoder side to explicitly guide the learning process. Moreover, the long-short constraint is also utilized on the decoder side to implicitly direct

the representation learning. Experiments on six real-world datasets illustrate the model effectiveness. An extensive ablation proves the indispensability of each model component.

REFERENCES

[1] L. Wang, Z. Ding, Z. Tao, Y. Liu, and Y. Fu, "Generative multi-view human action recognition," in *ICCV*, 2019, pp. 6212–6221.

[2] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *KDD*, 2010, pp. 333–342.

[3] J. M. Kleinberg, "An impossibility theorem for clustering," in *NeurIPS*, 2003, pp. 463–470.

[4] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: a survey and empirical demonstration," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.

[5] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *TKDE*, vol. 23, no. 2, pp. 307–320, 2010.

[6] P. Smyth, "Probabilistic model-based clustering of multivariate and sequential data," in *International Workshop on AI and Statistics*. San Francisco, CA: Morgan Kaufman, 1999, pp. 299–304.

[7] L. Wang, Z. Ding, and Y. Fu, "Low-rank transfer human motion segmentation," *TIP*, vol. 28, no. 2, pp. 1023–1034, 2019.

[8] L. Wang, Z. Ding, and Y. FU, "Learning transferable subspace for human motion segmentation," in *AAAI*, 2018.

[9] F.-l. Chung, T.-c. Fu, R. Luk, and V. Ng, "Evolutionary time series segmentation for stock data mining," in *ICDM*, 2002, pp. 83–90.

[10] L. Wang, B. Sun, J. Robinson, T. Jing, and Y. Fu, "EV-Action: Electromyography-vision multi-modal action dataset," in *FG*, 2020.

[11] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *TPAMI*, vol. 35, no. 3, pp. 582–596, 2012.

[12] M. W. Robards and P. Sunehag, "Semi-markov kmeans clustering and activity recognition from body-worn sensors," in *ICDM*, 2009, pp. 438–446.

[13] M. Hoai and F. De la Torre, "Maximum margin temporal clustering," in *Artificial Intelligence and Statistics*, 2012, pp. 520–528.

[14] S. Li, K. Li, and Y. Fu, "Temporal subspace clustering for human motion segmentation," in *ICCV*, 2015, pp. 4453–4461.

[15] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz inverse covariance-based clustering of multivariate time series data," in *KDD*, 2017, pp. 215–223.

[16] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[17] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *TKDD*, vol. 3, no. 1, pp. 1–58, 2009.

[18] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[19] L. Wang, Z. Ding, and Y. Fu, "Adaptive graph guided embedding for multi-label annotation." in *IJCAI*, 2018, pp. 2798–2804.

[20] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *CVPR*, 2009, pp. 2790–2797.

[21] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *ECCV*. Springer, 2012, pp. 347–360.

[22] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *TPAMI*, vol. 35, no. 1, pp. 171–184, 2012.

[23] M. Soltanolkotabi, E. Elhamifar, E. J. Candes *et al.*, "Robust subspace clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.

[24] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *TIP*, vol. 24, no. 12, pp. 5659–5670, 2015.

[25] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *SIGGRAPH*. ACM, 2015, p. 18.

[26] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *ECCV*. Springer, 2016, pp. 835–851.

[27] A. Budiman, M. I. Fanany, and C. Basaruddin, "Stacked denoising autoencoder for feature representation learning in pose-based action recognition," in *IEEE Global Conference on Consumer Electronics*, 2014, pp. 684–688.

[28] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers*, p. 107, 2000.

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] L. K. P. J. RDUSSEEUN, "Clustering by means of medoids," 1987.

[31] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NeurIPS*, 2002, pp. 849–856.

[32] S. Tierney, J. Gao, and Y. Guo, "Subspace clustering for sequential data," in *CVPR*, 2014, pp. 1019–1026.

[33] D. Huang, S. Yao, Y. Wang, and F. De La Torre, "Sequential max-margin event detectors," in *ECCV*. Springer, 2014, pp. 410–424.

[34] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: video structure comparison for recognition of complex human activities." in *ICCV*, vol. 1, 2009, p. 2.

[35] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *TPAMI*, vol. 29, no. 12, 2007.

[36] Z. Jiang, Z. Lin, and L. Davis, "Recognizing human actions by learning and matching shape-motion prototype trees," *TPAMI*, vol. 34, no. 3, pp. 533–547, 2012.

[37] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *ECCV*. Springer, 2014, pp. 459–473.

[38] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li, "Chalearn looking at people RGB-D isolated and continuous datasets for gesture recognition," in *CVPR*, 2016, pp. 56–64.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[40] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *KDD*, 2009, pp. 877–886.

[41] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *JMLR*, vol. 9, no. Nov, pp. 2579–2605, 2008.