Graph Constrained Data Representation Learning for Human Motion Segmentation

Mariella Dimiccoli\textsuperscript{1} Lluís Garrido\textsuperscript{2} Guillem Rodríguez-Corominas\textsuperscript{1} Herwig Wendt\textsuperscript{3}
\textsuperscript{1}Institut de Robótica i Informàtica Industrial (CSIC-UPC) \textsuperscript{2}Univ. of Barcelona \textsuperscript{3}CNRS, IRIT, Univ. of Toulouse
\{mdimiccoli,grodriguez\}@iri.upc.edu luis.garrido@ub.edu herwig.wendt@irit.fr

\section*{Abstract}

Recently, transfer subspace learning based approaches have shown to be a valid alternative to unsupervised subspace clustering and temporal data clustering for human motion segmentation (HMS). These approaches leverage prior knowledge from a source domain to improve clustering performance on a target domain, and currently they represent the state of the art in HMS. Bucking this trend, in this paper, we propose a novel unsupervised model that learns a representation of the data and digs clustering information from the data itself. Our model is reminiscent of temporal subspace clustering, but presents two critical differences. First, we learn an auxiliary data matrix that can deviate from the initial data, hence confers more degrees of freedom to the coding matrix. Second, we introduce a regularization term for this auxiliary data matrix that preserves the local geometrical structure present in the high-dimensional space. The proposed model is efficiently optimized by using an original Alternating Direction Method of Multipliers (ADMM) formulation allowing to learn jointly the auxiliary data representation, a non-negative dictionary and a coding matrix. Experimental results on four benchmark datasets for HMS demonstrate that our approach achieves significantly better clustering performance than state-of-the-art methods, including both unsupervised and more recent semi-supervised transfer learning approaches\textsuperscript{3}.

\section{1. Introduction}

Human-motion segmentation (HMS) aims at breaking a continuous sequence of data depicting human actions and activities into a set of internally coherent temporal segments, and has emerged as a suitable first step in early processing of untrimmed videos for human action recognition \cite{12}. Despite the fact that HMS has been intensively investigated so far \cite{28, 15, 35}, performance have not yet reached the level of accuracy required in real world applications.

Because labelling large amounts of videos for creating a training set for HMS is very expensive and time-consuming, the problem has been traditionally addressed through unsupervised learning techniques \cite{22, 2, 21}. Several approaches cast HMS as a clustering problem by relying on the framework of subspace clustering \cite{30}. The key idea of subspace clustering is to learn an effective representation in form of a coding matrix from which it is possible to construct an affinity matrix that allows to separate data points according to their underlying low-dimensional subspaces. These subspaces are assumed to correspond with the different motions. However, these clustering approaches typically do not take into account temporal continuity and, moreover, they are sensitive to noise. Some recent subspace clustering based approaches have focused on dealing with noise and outliers naturally present in the data, as well as with its sequential nature \cite{6, 7, 28, 15, 16}. Outside the subspace clustering framework, other temporal clustering approaches work directly on the temporal data instead of on an underlying representation to estimate cluster labels \cite{27, 36, 13}. Nevertheless, the main challenge of all these unsupervised approaches remains how to cope with the lack of prior knowledge, which could easily cause unpredictable segmentation output. Recently, this problem has been partially addressed by transfer subspace techniques, that leverage prior information from source data to improve clustering accuracy on the target data \cite{31, 32, 40}. These methods have indeed reported improved performance and currently represent the state of the art in HMS.

Instead of leveraging information from source annotated data, here we propose to cope with the lack of prior knowledge in an unsupervised fashion by further exploiting the local geometric structure present in the original high-dimensional space. Our a priori on the data samples is that they are drawn from an unknown underlying similarity graph, where nodes in the graph correspond to individual frames, edges to connections between nodes, and communities (i.e., groups of nodes that are interconnected by edges with large weights) roughly correspond to different motions. Our model aims at jointly learning the subspaces and a compatible representation for the similarity graph. Since we are interested in pushing the limits of state-of-

\textsuperscript{3}Project page at URL https://github.com/mdimiccoli/GCRL-for-HMS/.
the-art HMS, we build on the framework of Temporal Subspace Clustering (TSC) [15], that has proven to be the most effective subspace clustering technique for temporal data without transfer. The rational underlying TSC is to find a coding matrix whose affinity graph leads to good clustering with temporal coherence. This is ensured by a least squares regression formulation where the original data are approximated through a nonnegative dictionary and code under block diagonal and temporal Laplacian regularization terms that lend the model its global subspace structure and temporal smoothness, respectively. However, when the original data is noisy or corrupted, nuisances propagate also to the coding matrix. To avoid this, we allow the auxiliary data representation to vary, hence conferring to the coding matrix more degrees of freedom and robustness to subspace assumption violations and noise. Our prior for the auxiliary data matrix is that its columns act as representation vectors for the affinity matrix (graph) of the original data. Therefore, we introduce a graph regularization term, that controls the distance between the affinity graphs of the original and auxiliary data. This effectively allows the auxiliary data to cope with data nuisances while preserving the local geometrical structure of the original data. We propose an efficient optimization formulation for our model, based on the Alternating Direction Method of Multipliers (ADMM).

Our main contributions are summarized as follows: 1) we propose a novel approach for HMS, which jointly learns an auxiliary data matrix, a non-negative dictionary and a coding matrix under temporal, block-diagonal and graph constraints via an original ADMM formulation, 2) we introduce an original graph regularization term that preserves the geometrical structure of the original data in high-dimensional space, 3) we present a comprehensive analysis of our model, including ablation study and sensitivity analysis, 4) we achieve significant performance improvements (up to \( \approx 20\% \) accuracy and \( \approx 5\% \) NMI) over the state-of-the-art methods on four public benchmarks for HMS.

2. Related work

Subspace clustering. Subspace clustering assumes sampled data to be drawn from a union of multiple subspaces, hence breaking the assumption that all of the clusters in a dataset are found in the same (sub)set of dimensions. Under this assumption, the sampled data obey the so called self-expressiveness property, i.e., each data point in a union of subspaces can be represented by a linear combination of other points in the dataset. This can be formulated as

\[
X = XZ,
\]

where \( Z \in \mathbb{R}^{r \times N} \) is the representation coefficient or coding matrix. The goal of subspace clustering is to find a coding matrix \( Z \) that minimizes the reconstruction error under some regularity constraints. The final clusters are obtained by applying spectral clustering to the affinity matrix of \( Z \). Different variants of subspace clustering have been proposed that impose different constraints on \( Z \) for this purpose. Least Square Regression (LSR) [20] uses an \( \ell_2 \) norm regularizer for \( Z \). Low-rank representations (LRR) [18] impose on \( Z \) to have small rank and are therefore more robust to corrupted data. Sparse subspace clustering (SSC) algorithms [6] enforce a sparsity constraint on \( Z \) and are also applicable to noisy data that do not cluster perfectly into subspaces. Probabilistic methods usually model the data points using a mixture of probabilistic PCAs and are often more robust to noise and outliers than LSR and LRR [1]. Information theoretic learning based framework such as [16] have proved to be robust to piecewise identically distributed noise. One of the shortcomings of all of the above methods is that they neglect the temporal information that is implicitly encoded in sequential data.

Temporal data clustering. Temporal data clustering methods explore dynamic regularities underlying temporal data in an unsupervised learning fashion. Following the taxonomy introduced in [38], we can distinguish three general approaches to temporal data clustering, depending on the data dependency treatment: model-based [27, 36], temporal proximity based [14] and representation based algorithms [5, 4, 13]. Model-based and temporal proximity based approaches work directly on temporal data and deal with the temporal correlation during clustering analysis. A representative example in the context of HMS is Aligned Clustering analysis (ACA) [39], that extends kernel k-means and spectral clustering for time series clustering, formulating the problem within an optimization framework.

Instead, representation based approaches capture temporal data dependencies via a sparse representation. Within this class fall all methods that tried to adapt the subspace clustering framework [7] to sequential data. Tierney et al. [38] proposed an Ordered Subspace Clustering (OSC) method that incorporates a neighbour penalty term to enforce consecutive frames to have similar representation. In Sequential Subspace Clustering [33] a quadratic normalizer is imposed on the sparse coefficients to model the temporal correlation among the data points. Clustering accuracy is enhanced by incorporating into the model a block-diagonal prior for the spectral clustering affinity matrix. Temporal Subspace clustering (TSC) [15] learns jointly a dictionary and a representation code with a Laplacian temporal regularization. The use of a dictionary allows for more expressive coding. These temporal clustering methods cannot explicitly and efficiently deal with corrupted or partially missing data. A probabilistic approach to TSC was proposed in [8], that models temporal dependencies by using Gaussian Process (GP) priors [25] and is more robust to outliers.
Transfer subspace-learning temporal clustering. Unsupervised subspace learning methods work without any prior knowledge and achieve reasonable but relatively modest performance. Recently, transfer subspace learning-based approaches [31, 32, 40] have reported improved performance in motion segmentation. These methods adapt to the temporal data clustering problem the concept of transfer learning [23]. Transfer learning, also commonly known as domain adaptation, aims at transferring knowledge from a source domain to a target domain, typically by imposing the data distributions of two domains to be similar in a semi-supervised fashion. Contrary to [31, 32], that carry out the transfer subspace learning on the original high-dimensional feature space, [40] factorizes the original features of the source and the target data into implicit multi-layer feature spaces and uses them to fuse multi-level structural information effectively.

3. Approach

Fig. 1 provides an overview of our approach for HMS. Our method takes as input frame-level features extracted from the video sequence, and firstly computes an affinity matrix from the video sequence, and then computes an affinity matrix. Our method takes as input frame-level features extracted from the video sequence, and firstly computes an affinity matrix from the video sequence, and then computes an affinity matrix. The time series data corresponding to motion features
dimensional Euclidean space. The normalized pairwise similarity matrix is the number of atoms in the dictionary. The use of the dictionary in the above formulation instead of the self-representation model (i.e., \(X \approx XZ\)) allows to learn a more expressive coding matrix \(Z\) from the data [15]. \(Z\) can then be used to construct an affinity graph for subspace clustering, and the temporal clustering results can be obtained by applying an efficient clustering algorithm, e.g., normalized cuts. Yet, noisy or corrupted data samples contained in \(X\) as well as portions of temporal segments that deviate from the subspace assumption may nonetheless propagate to \(Z\) and \(D\) and lead the model astray.

Graph constrained data representation. Our key insight is therefore to let \(DZ\) approximate auxiliary data \(\tilde{X} \in \mathbb{R}^{n \times N}\) instead of \(X\), that we optimize jointly with \(D\) and \(Z\). This lends the model extra flexibility to find more expressive codes \(Z\). Our prior for the auxiliary data \(\tilde{X}\) is that they act as representation vectors for the affinity matrix (graph) of the original data \(X\). We additionally assume that communities in the similarity graph roughly correspond to subspaces, where nodes correspond to individual frames and edges correspond to similarity values between frames. Thus, the auxiliary data points \(\tilde{X}\) approximately reproduce the local geometry of the original data points \(X\) but can otherwise move in \(\mathbb{R}^{n \times N}\) to find a representation that is more compatible with the subspace clustering model, and to unburden \(Z\) and \(D\) from dealing with corrupted data points.

The assumptions on \(\tilde{X}\) are expressed through a regularization term that is reminiscent of graph embedding [10] and penalizes deviations of the graph of the new data points \(\tilde{X}\) from the graph of \(X\). Specifically, the graphs of \(X\) and \(\tilde{X}\) are represented by their respective \(\mathbb{R}^{n \times N}\) affinity matrices \(G_0 = S(X)\) and \(\tilde{G} = S(\tilde{X})\). These are defined element-wise via a pairwise similarity, upon normalization to unit total weight,

\[
(S(X))_{kj} = w_{kj}/W; \quad W = \sum_{(k,j) \in E} w_{kj}
\]

\[
w_{kj} = \exp \left(- (1 - d(X(j), X(k)))/h \right),
\]

where \(E\) is the set of pairwise relations, \(d(\cdot, \cdot)\) is the cosine distance and \(h\) the filtering parameter of the exponential function. The normalized pairwise similarities can be interpreted as empirical joint probability distributions between pairs of vertices \(\tilde{p}(k,j)\). Therefore, Eq. (1) defines a distribution \(p(\cdot,\cdot)\) over the space \(N \times N\). Then, to preserve first-order neighbor relations, we minimize the KL-divergence of the two joint probability distributions \(\tilde{p}_{G_0}(\cdot,\cdot), p_G(\cdot,\cdot)\) that correspond with
the original and learnt auxiliary data graphs, respectively. Upon removal of constant terms, this yields the cross entropy (CE) divergence \( CE(\hat{p}_{ij}, \ldots; \hat{p}_{ij}, \ldots) = -\sum_{(k,j)} \hat{p}_{G_{ij}}(k,j) \log(\hat{p}_{G_{ij}}(k,j)) \); we denote this additional loss term as \( \mathcal{L}_G \) and name it graph constrained loss. Our CE term is easy to minimize and strongly convex, guaranteeing convergence of our algorithm, see Sec. 4.4.

**Model.** With this graph constrained loss and the new data fidelity term on \( \hat{X} \), we obtain our final model:

\[
\begin{aligned}
\min_{Z,D,\hat{X}} \quad & \mathcal{L}_G(S(\hat{X}), G_0) + \lambda_0 \| \hat{X} - DZ \|_F^2 \\
\text{s.t.} & \quad Z \geq 0, \quad D \geq 0, \quad \|d_i\|_2^2 \leq 1, \quad i = 1, \ldots, r. \quad (2)
\end{aligned}
\]

The Frobenius norm on \( Z \) is used to enforce block diagonal structure in \( Z \), see e.g. [19]. Moreover, \( f(Z) = \text{tr}(Z^T L Z) \) is a weighted Laplacian \( \ell^2 \) regularization for temporal consistency defined as in [15]. The unit norm constraints on \( d_i \) resolve the scale ambiguity between \( D \) and \( Z \).

### 3.2. Optimization

To solve the objective function in (2), we devise an optimization algorithm based on ADMM. Upon introduction of a set of auxiliary variables \( Y, U \) and \( V \), (2) is formulated equivalently as

\[
\begin{aligned}
\min_{\hat{X},U,Y,V,D,Z} \quad & \mathcal{L}_G(S(\hat{X}), G_0) + \lambda_0 \| Y - UV \|_F^2 + \lambda_1 \| V \|_F^2 + \lambda_2 f(V) \\
\text{s.t.} & \quad Z = \hat{X}, \quad U = D, \quad V = Z, \\
& \quad Z \geq 0, \quad D \geq 0, \quad \|d_i\|_2^2 \leq 1, \quad i = 1, \ldots, r.
\end{aligned}
\]

The augmented Lagrangian for this problem is

\[
\begin{aligned}
\mathcal{L}_G(S(\hat{X}), G_0) + \lambda_0 \| Y - UV \|_F^2 + \lambda_1 \| V \|_F^2 + \lambda_2 \text{tr}(VLTV) \\
+ \langle \hat{X}, Y - \hat{X} \rangle + \langle U, U - D \rangle + \langle V, V - Z \rangle \\
+ \frac{\rho}{2} \| Y - \hat{X} \|_F^2 + \rac{\rho}{2} \| U - D \|_F^2 + \frac{\rho}{2} \| V - Z \|_F^2 \\
\text{s.t.} & \quad Z \geq 0, \quad D \geq 0, \quad \|d_i\|_2^2 \leq 1, \quad i = 1, \ldots, r. \quad (3)
\end{aligned}
\]

The ADMM algorithm that solves this optimization problem (3) is given by alternately minimizing \( \mathcal{L}_G \) w.r.t. each variable \( Y, V, U, Z, D \) individually [3].

**Update for \( V, U \).** The updates for \( V, U \) are similar to those of the TSC algorithm of Li et al. [15], but need to be slightly modified to account for the new term \( \mathcal{L}_G \) and the additional regularization parameter \( \lambda_0 \). Specifically, the update for \( V \) is obtained as the solution to the Sylvester equation

\[
(2\lambda_0 UT + \lambda_1 I + \rho I)V + \lambda_2 VLTV = 2\lambda_0 UTY - AU + \rho Z \quad (4)
\]

and the update for \( U \) has the closed-form solution

\[
U = (2\lambda_0 YV^T - AU + \rho D)(2\lambda_0 YV^T + \rho I)^{-1}.
\]

**Update for \( Z, D \).** The updates for \( V, U \) are identical to those of the TSC algorithm of Li et al. [15] and are given by the expressions

\[
Z = \mathcal{F}_+(V + \Lambda V / \rho),
D = \mathcal{F}_+(U + \Lambda U / \rho),
\]

where \( \mathcal{F}_+(A)_{ij} = \max(A_{ij}, 0) \).

**Update for \( Y \).** We cancel the gradient of (3) w.r.t. \( Y \)

\[
\nabla_Y L_\rho = 2\lambda_0 Y - 2\lambda_0 UV + \Lambda \hat{X} + \rho(Y - \hat{X})
\]

and find the update for \( Y \)

\[
Y = (2\lambda_0 + \rho)^{-1}(2\lambda_0 UV - \Lambda \hat{X} + \rho \hat{X}).
\]

**Update for \( \hat{X} \).** The update for \( \hat{X} \) does not have a closed form solution since one has to solve

\[
\hat{X} = \arg \min_{\hat{X}} \mathcal{L}_G(S(\hat{X}), G_0) + \langle \Lambda \hat{X}, X - \hat{X} \rangle + \frac{\rho}{2} \| Y - \hat{X} \|_F^2.
\]

This expression is easy to solve using gradient descent. The expression for the gradient is given by

\[
\nabla_{\hat{X}} L_\rho = \nabla_{\hat{X}} \mathcal{L}_G(S(\hat{X}), G_0) - \Lambda \hat{X} - \rho(Y - \hat{X}),
\]

and thus essentially requires the calculation of the gradient of the cross-entropy loss \( \mathcal{L}_G \) composed with the similarity function \( S \). Both \( Y \) and \( \hat{X} \) are normalized to \([0, 1] \), as \( X \) is assumed to be.

**Update for the Lagrange multipliers \( \Lambda \).** Finally, the Lagrange multipliers are updated as usual as

\[
\Lambda_U \leftarrow \Lambda_U + \rho(U - D),
\Lambda_V \leftarrow \Lambda_V + \rho(V - Z),
\Lambda \hat{X} \leftarrow \Lambda \hat{X} + \rho(Y - \hat{X}).
\]

The details for solving Eq. (3) via the above ADMM formulation are summarized in Algorithm 1. Note that the normalization of \( X \) and \( \hat{X} \), in addition to the strong convexity property of the CE function, allows to ensure mild convergence conditions for the graph regularization term, see Sec. 4.4.

### 3.3. Clustering

To produce the temporal segmentation results, we applied the Normalized Cut algorithm [26] to the affinity matrix \( A \) of the learnt coding matrix \( Z \). We proceed as in the TSC algorithm [13] and define each element of \( A \) as the cosine distance between the respective pair of the learnt code vectors

\[
(A(Z))_{ij} = \frac{z_i^T z_k}{\| z_i \|_2 \| z_k \|_2}. \quad (5)
\]
consists of 90 video sequences, each with a different person
posed model, which we detail next.

Table 1: Clustering comparison results in terms of ACC and NMI on four HMS datasets. For the transfer subspaces learning
person is treated as a motion segmentation task.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRR</td>
<td>0.4382</td>
<td>0.3638</td>
<td>LRR</td>
<td>0.4862</td>
<td>0.4297</td>
<td>LRR</td>
<td>0.4051</td>
<td>0.4162</td>
<td>LRR</td>
<td>0.2249</td>
<td>0.2397</td>
</tr>
<tr>
<td>RSC</td>
<td>0.4112</td>
<td>0.4894</td>
<td>RSC</td>
<td>0.3485</td>
<td>0.3252</td>
<td>RSC</td>
<td>0.3664</td>
<td>0.1881</td>
<td>RSC</td>
<td>0.3730</td>
<td>0.3418</td>
</tr>
<tr>
<td>OSC</td>
<td>0.7047</td>
<td>0.5216</td>
<td>OSC</td>
<td>0.5931</td>
<td>0.4393</td>
<td>OSC</td>
<td>0.6877</td>
<td>0.5846</td>
<td>OSC</td>
<td>0.5589</td>
<td>0.4327</td>
</tr>
<tr>
<td>SSC</td>
<td>0.6009</td>
<td>0.4576</td>
<td>SSC</td>
<td>0.3858</td>
<td>0.3137</td>
<td>SSC</td>
<td>0.4998</td>
<td>0.4389</td>
<td>SSC</td>
<td>0.4758</td>
<td>0.3817</td>
</tr>
<tr>
<td>LSR</td>
<td>0.5093</td>
<td>0.5091</td>
<td>LSR</td>
<td>0.4548</td>
<td>0.4894</td>
<td>LSR</td>
<td>0.4322</td>
<td>0.5183</td>
<td>LSR</td>
<td>0.3667</td>
<td>0.3979</td>
</tr>
<tr>
<td>TSC</td>
<td>0.6111</td>
<td>0.8199</td>
<td>TSC</td>
<td>0.4781</td>
<td>0.7129</td>
<td>TSC</td>
<td>0.5340</td>
<td>0.7593</td>
<td>TSC</td>
<td>0.5556</td>
<td>0.7721</td>
</tr>
<tr>
<td>TSS(M)</td>
<td>0.6208</td>
<td>0.8509</td>
<td>TSS(M)</td>
<td>0.5395</td>
<td>0.8049</td>
<td>TSS(W)</td>
<td>0.5944</td>
<td>0.7878</td>
<td>TSS(K)</td>
<td>0.5792</td>
<td>0.8286</td>
</tr>
<tr>
<td>LTS(K)</td>
<td>0.6391</td>
<td>0.8599</td>
<td>LTS(M)</td>
<td>0.5509</td>
<td>0.8226</td>
<td>LTS(M)</td>
<td>0.6299</td>
<td>0.8128</td>
<td>LTS(U)</td>
<td>0.5980</td>
<td>0.8211</td>
</tr>
<tr>
<td>MTS(K)</td>
<td>0.6436</td>
<td>0.8371</td>
<td>MTS(M)</td>
<td>0.6010</td>
<td>0.8270</td>
<td>MTS(M)</td>
<td>0.6433</td>
<td>0.8239</td>
<td>MTS(U)</td>
<td>0.6163</td>
<td>0.8314</td>
</tr>
<tr>
<td>Ours</td>
<td>0.8501</td>
<td>0.9053</td>
<td>Ours</td>
<td>0.7864</td>
<td>0.8325</td>
<td>Ours</td>
<td>0.8700</td>
<td>0.8256</td>
<td>Ours</td>
<td>0.8297</td>
<td>0.8471</td>
</tr>
<tr>
<td>±0.0048</td>
<td>±0.0017</td>
<td>±0.0069</td>
<td>±0.0041</td>
<td>±0.0022</td>
<td>±0.0018</td>
<td>±0.0042</td>
<td>±0.0028</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Results on Weiz dataset (b) Results on Keck dataset (c) Results on UT dataset (d) Results on MAD dataset

4. Experimental results

4.1. Datasets

We used four human motion datasets to evaluate our proposed model, which we detail next.

Weizmann action dataset (Weiz). The Weiz dataset [9] consists of 90 video sequences, each with a different person
performing 10 different types of actions in an outdoor environment.

Keck gesture dataset (Keck). The Keck dataset [17] consists of 168 video sequences with 4 people performing 14 military gestures with perturbations in the background. The same gesture is repeated three times by each person. Each person is treated as a motion segmentation task.

UT interaction dataset (UT). The UT dataset [24] consists of 20 videos, each containing 6 classes of human-human interactions. Each video sequence is around 60 seconds long.

Multi-model Action Detection dataset (MAD). The MAD dataset [11] consists of 40 sequences captured from 20 subjects. Each subject performs 35 activities (e.g., jumping, kicking) continuously in two different indoor environments, each corresponding to a different sequence. Each person is treated as a motion segmentation task.

4.2. Experimental setup

Dataset. We used the n = 324 dimensional HOG features, extracted frame by frame and concatenated together, publicly available for these benchmark datasets\(^2\), and used the same setting as in [32].

Evaluation metrics and parameters settings. In all the experiments, we use the clustering accuracy (ACC) and normalized mutual information (NMI) as the evaluation metrics [34]. The clustering accuracy is defined as

\[
\text{ACC} = \sum_{i=1}^{N} \delta(g_{i}, \text{map}(c_{i}))/N,
\]

where \(\delta(\cdot, \cdot)\) is the Kronecker delta function and \(\text{map}(c_{i})\) is the best permutation mapping function which maps each

---

\(^2\)https://github.com/wanglichenxj/Low-Rank-Transfer-Human-Motion-Segmentation
cluster label \( c \) to a ground truth label \( g \). The NMI is defined as

\[
\text{NMI}(\Omega, C) = \frac{I(\Omega, C)}{\left( \frac{H(\Omega)}{2} + \frac{H(C)}{2} \right)}
\]

where \( I \) denotes the mutual information and \( H \) the entropy. \( \Omega \) is the set of clusters and \( C \) the set of classes. Higher values indicate better performance for the two metrics.

We set the similarity bandwidth parameter \( h \) in (2) empirically to \( h = 0.0015 \), which leads to visually satisfactory results for the graph affinity matrices \( G \). The hyperparameters \( \lambda_0, \lambda_1, \lambda_2 \) of our method are tuned independently for each dataset by grid search. Sensitivity to hyperparameter tuning is reported in Section 4.4 below.

### 4.3. Comparative results

We compare to several state-of-the-art subspace clustering based methods such as LRR [18], SSC [7], LSR [20], temporal data clustering approaches such as TSC [15], OSC [28], and more recent Transfer Subspace Clustering based methods including Transfer Subspace Segmentation (TSS) [31], Low-rank Transfer Subspace (LTS) [32], Multi-mutual transfer subspace learning (MTSL) [40]. For the latter class of methods, we show in Tab. 1 the best results obtained independently of the source data as reported in [32] and [40]. We computed the best performance on the four benchmark datasets using the method [16] that we named Robust Subspace Clustering (RSC), and report average values over five random seeds obtained by using the code made available by the authors. Bold indicates the best performance. Our method consistently obtains considerably better performance, also when compared to the transfer subspace clustering-based methods (TSS, LTS, MTS). Qualitative results are shown in Fig. 2, where we show the estimated clusters for the first sequences of the four datasets, respectively. It can be appreciated that the segmentation obtained with the proposed approach is highly consistent and well aligned with the ground truth. Results obtained with TSC and LTS are less satisfactory and prone to produce spurious erroneous short segments, in particular close to segment transitions (e.g., at the center of the Weizmann dataset sequence), as well as merge segments into one cluster (e.g., at the center of the MAD dataset sequence).

### 4.4. Model study

**Graph constrained representation.** We illustrate how our method leverages on the local geometric structure of the original data \( \tilde{X} \) to produce an improved auxiliary data representation \( \tilde{X} \). Specifically, we compare, for the first subject of the Weizmann dataset, the data \( X \) and the graph constrained data representation \( \tilde{X} \) for different relative weight

3https://github.com/YuanmanLi/github-MWEE
4https://github.com/wanglichenxj/Low-Rank-Transfer-Human-Motion-Segmentation
of the graph penalty $L_G$ in (3) (realized by multiplying/dividing $\lambda_i, i = 0, 1, 2$, by a common factor). In Fig. 3 we plot a subset of 4 arbitrarily chosen features of $X$ and $\tilde{X}$ (plotted as time series, left plots), the 50 first features of $\tilde{X}$ (center plots) and a visualization of $\tilde{X}$ by t-SNE [29] (right plots); the vertical bars indicate true motion boundaries. It can be observed that the features $X$ resemble de-noised versions of the original data $X$, that become smoother and smoother as the relative weight of $L_G$ in (3) is decreased and thus $\tilde{X}$ is more free to move from $X$, while at the same time retaining the global motion feature evolution and, in particular, sharp transitions at motion boundaries (cf., e.g., motion segment 4). The t-SNE visualization corroborates such observations and reveals that the clusters get more and more clearly separated in the high dimensional space.

**Temporal versus Graph regularization.** Fig. 4 provides a more quantitative illustration of the effect of the graph penalty $L_G$ (results obtained for the first subject of the Weizmann dataset). Specifically, we plot the norm of the temporal second derivative of $\tilde{X}$ as a measure of smoothness (left plot) and of the norm of $X - \tilde{X}$ as a measure of data fidelity (right plot) as functions of temporal regularization parameter $\lambda_2$, and for different relative weight of $L_G$ in (3) (blue, red and yellow color). The results for this experiment indicate that reducing the relative weight of $L_G$ leads to auxiliary representations $\tilde{X}$ that are smoother and differ more from $X$, as expected. The same holds true for increasing $\lambda_2$. However, it can be appreciated that reducing the relative weight of the graph penalty term (i.e., letting $\tilde{X}$ move more freely) is more effective in controlling the temporal smoothness of $\tilde{X}$ than tuning $\lambda_2$, while at the same time retaining a significantly tighter fit with the original data $X$. For instance, decreasing the weight of $L_G$ by a factor 10 increases smoothness by a factor $10^{0.7} \approx 5 - 8$ (left plot) at only $10^{0.05} \approx 10 - 15\%$ increased deviation from the original data, while increasing $\lambda_2$ by two orders of magnitude leads to modest smoothness increase of $10^{0.3} \approx 2$ at more than $10^{0.08} \approx 20\%$ larger deviation of $\tilde{X}$ from $X$.

**Noise modeling.** In Fig. 5 we illustrate on the first subject of the Weizmann dataset the effectiveness of our model in dealing with noise. Specifically, we add centred piece-wise iid Gaussian noise to each frame, as a function of noise level: fixed piece-wise geometry (left) and randomly varying piece-wise geometry (right).

![Figure 4: Frobenius norm of the temporal second derivative](image)

**Figure 4:** Frobenius norm of the temporal second derivative (left) and difference with data $X$ (right) of auxiliary representation $\tilde{X}$ for different values of the regularization term $L_G$ and temporal regularization $\lambda_2$.

![Figure 5: Clustering performance](image)

**Figure 5:** Clustering performance (Weiz, subject 1) upon addition of centred piece-wise iid Gaussian noise to each frame, as a function of noise level: fixed piece-wise geometry (left) and randomly varying piece-wise geometry (right).

Parameter sensitivity analysis. In our approach, three key regularization parameters, i.e., $\lambda_0$, $\lambda_1$ and $\lambda_2$, need to be manually tuned. To study sensitivity to their variation on the model output, we fix the value of one parameter and vary the other two parameters over 2 to 3 orders of magnitude, i.e., $\lambda_0$, $\lambda_1$, $\lambda_2$. Segmentation performance are manually tuned. To study sensitivity to their variation on the model output, we fix the value of one parameter and vary the other two parameters over 2 to 3 orders of magnitude. Results for the Weizmann dataset are shown in Fig. 6 (left). Clearly, very small values for the temporal regularization weight $\lambda_2$ worsen the performance. This is to be expected because data temporal coherence is not modeled any longer when $\lambda_2 \to 0$. Otherwise, the results are consistent in a large range of parameter values. In particular, our proposed method obtains better NMI performance when $\lambda_0 \in [0.1, 1.2]$ and when $\lambda_2 \geq 5$. Moreover, similarly to the observations in [18], the performance of our model is found to vary little with the parameter $\lambda_1$ that regularizes the block-diagonal constraint. Most importantly, these results demonstrate that every term in our model is useful for improving performance.

Moreover, Fig. 6 (right) shows the performance of our method for different values of the filtering parameter $h$ of the similarity kernel $S$ that tunes the data graph representations and hence the coupling of the original and the auxiliary data graphs $G_0$ and $G$. Segmentation performance are found to remain nearly constant over more than two orders of magnitude for $h$, which demonstrates that our approach is...
the mean has complexity $O(Kr^2N)$, thus equal to TSC.

**Ablation study.** To validate the effectiveness of the proposed model, we compare the performance of our method with that obtained when the graph based regularization term is removed and $X \approx X$ is fixed in the fidelity term. This corresponds to the TSC model of [15], but with different values for the parameters $\lambda_i, i = 0, 1, 2$, individually optimized here for each dataset. Quantitative results are reported in Tab. 2 for all four datasets and confirm that our approach is highly effective and leads to consistent and significant improvements (NMI increase of up to +9%).

5. **Conclusions**

We proposed a novel subspace clustering based approach for HMS which jointly learns an auxiliary data representation, a non-negative dictionary and a coding matrix under temporal, block-diagonal and graph constraints via an original ADMM formulation. The rationale underlying our approach is to operate on auxiliary data $\tilde{X}$ instead of the original time-series data $X$, lending the model extra flexibility for finding more expressive codes $Z$, while preserving the local geometrical structure of the original data in high-dimensional space. We present a comprehensive analysis of our model, including several illustrations of the model behaviour, model design justification, ablation study, parameter sensitivity and convergence analysis. Furthermore, we achieve significant performance improvements of up to $\approx 20\%$ (accuracy) and $\approx 5\%$ (NMI) over the state-of-the-art methods on four public benchmarks for HMS.

6. **Acknowledgements**

Work partially funded by projects MINECO/ERDF Ryc, PID2019-110977GA-I00, JAEINT19-EX-0014, RED2018-102511-T.

**References**


[2] Results obtained using 5 random seeds and grid search to achieve best performance in terms of NMI, which explains quantitative differences with results from [12, 40] shown in Tab. 1.


[38] Yun Yang and Ke Chen. Temporal data clustering via weighted clustering ensemble with different representations. IEEE transactions on knowledge and data engineering, 23(2):307–320, 2010. 2
