JOURNAL OF DECISION MAKING AND HEALTHCARE
Volume 3 (2026), No. 2, 62–77
DOI:https://doi.org/10.69829/jdmh-026-0302-ta01
Tulipa Opera Scholarum

ROBUST SUBSPACE CLUSTERING VIA LOCAL WEIGHTED SPARSE AND GLOBAL PRIOR INFORMATION

JIAQI ZHANG[1], RENLI LIANG[1,*], AND ZAIYUN PENG[2]

[1] *Chongqing Jiaotong University, Chongqing 400074, P.R. China*
[2] *School of Mathematics, Yunnan Normal University, Kunming 650092, P.R. China*

Abstract. The task of subspace clustering is to divide high-dimensional data into several low dimensional subspaces. Most existing methods ignore the local and global prior information of the data. In response to such issues, we propose a subspace clustering model based on local weighted sparsity and global prior information. By utilizing the distance information of the data, the guidance coefficient matrix Z is represented with as few local points as possible, which can enhance the discriminative ability of our model. By introducing a matrix D to describe different prior information of data. In addition, considering that real-world data may contain multiple types of noise, we consider sparse noise and Gaussian noise priors to enhance the robustness of our model. Then, we have introduced an efficient splitting algorithm based on the augmented Lagrangian framework for the model to ensure its convergence. Finally, experiments on synthetic data, four real-world datasets, temporal action datasets, and manifold datasets demonstrate that our method outperforms existing methods in clustering accuracy, computational efficiency, and noise robustness.

Keywords. Subspace Clustering, Spectral Clustering, Weighted Sparse.

1. Introduction

Subspace clustering, as an important tool for high-dimensional data analysis, aims to assign data points to their corresponding low dimensional subspaces. Due to its ability to effectively handle high-dimensional and noisy real world data, subspace clustering has been widely applied in fields such as computer vision [16], machine learning [1], and specific tasks include motion segmentation [19], face clustering [15], video surveillance [3], and system identification [28]. Among various subspace clustering methods, the methods based on spectral clustering has attracted much attention due to its theoretical completeness and practical effectiveness. Specifically, sparse subspace clustering (SSC) [6] is based on the fact that each point in a union of subspaces has a sparse representation with respect to a dictionary formed by all other data points. Although SSC minimizes the vector l_1 norm of the representation matrix to induce sparsity, in the empirical side [18], SSC's solution is sometimes overly sparse such that the affinity graph of data from a single subspace may be disconnected. Low-rank representation (LRR) [4] minimizes the nuclear norm to capture the global low-rank structure and exhibits robustness to noise and outliers. Least squares regression (LSR) [15] utilizes the Frobenius norm to maintain the correlation between samples, while transforming the subspace clustering problem into a least squares problem with closed form solutions, giving it the advantages of stability and efficiency. However, due to its excessively smooth connection characteristics, it can easily lead to blurred subspace boundaries and damage clustering performance. To address this drawback, many joint learning methods based on sparse representation have emerged. Because the representation matrix is often simultaneously sparse and low-rank. Wang et al. [23] propose a new algorithm, termed low-rank sparse subspace

*Corresponding author.
E-mail address: zhangjiaqi53@126.com (J. Zhang), pengzaiyun@126.com (Z. Peng), and renli.liang@cqjtu.edu.cn (R. Liang)
Received: March 16, 2026, Revised: April 10, 2026, and Accepted: April 18, 2026.

clustering (LRSSC), by the combining SSC and LRR, and develop theoretical guarantees of the success of the algorithm. Yin et al. [27] a novel clustering algorithm via learning representation and affinity matrix conjointly, which can learn sparse representation and affinity matrix in a unified framework. Experimental results show the proposed method achieves better clustering results. Xin et al. [25].

proposed a new semi-supervised sparse subspace clustering method, named semi-supervised sparse subspace clustering with manifold regularization (S^4CMR), by introducing the local manifold regularization was also integrated to enhance clustering stability and local consistency. In addtion, in application scenarios such as video analysis, action recognition, and continuous signal processing, data naturally has temporal or spatial continuity. Adjacent frames or samples often exhibit strong similarity and are more likely to belong to the same subspace. Therefore, by introducing temporal information, researchers have developed various subspace clustering models for temporal data. For example, Guo et al. [8] introduced a temporal penalty term to make the representation coefficients of adjacent frames tend to be consistent. Hu et al. [10] proposed a robust temporal subspace clustering scheme based on l_1 time graph. Subsequently, Hu et al. [9] proposed a one-step temporal subspace clustering method that combines structured sparsity with temporal Indicator Graph-Laplacian (IL) regularization for clustering. The effectiveness of the above method has been validated on multiple datasets.

Inspired by the above research, we propose a subspace clustering model based on local weighted sparsity and global prior information. By considering local distance information and global prior information to guide the construction of coefficient matrix, which can enhance the discriminability of our model. Furthermore, real-world data is often contaminated by various types of noise, and we also consider sparse and dense noise in our model. Then, we introduced a segmentation algorithm based on the augmented Lagrangian framework. The experimental results on synthetic data, four real-world datasets, and one video action dataset demonstrate the effectiveness and adaptability of our method in complex scenes. Finally, our extended experiment on two moon dataset demonstrate that our method can also effectively solve manifold data problems.

2. Preliminaries

In this section, we first summarize the notations used in this paper. For any two matrices $X, Y \in \mathbb{R}^{m\times n}$, we define $\langle X, Y\rangle = \text{tr}(X^TY)$. The l_1 norm and Frobenius norm of the matrix $X \in \mathbb{R}^{m\times n}$ are respectively defined as

$$\|X\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|X_{ij}|, \quad \|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}X_{ij}^2}.$$

Lemma 2.1. [26] *For any $\lambda > 0$ and $X, Y \in R^{m\times n}$, the problem is*

$$\min_{X\in R^{m\times n}} \lambda\|X\|_1 + \frac{1}{2}\|X - Y\|_F^2.$$

The optimal solution can be obtained through the operator $X_\lambda(Y) \in R^{m\times n}$, which is defined as

$$[X_\lambda(Y)]_{ij} = \max\{|Y_{ij}| - \lambda, 0\} \cdot \text{sign}(Y_{ij}),$$

among them, the symbol function $\text{sign}(\cdot)$ *is defined as*

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0. \end{cases}$$

3. Model Formulation

Subspace clustering utilizes the property that high-dimensional data can be represented by multiple low dimensional data for clustering. Based on sparse representation theory, each point is linearly represented by a very small number of other points in the dataset. However, with the influence of noise, the sparse representation may incorrectly select samples from different subspaces, resulting in inaccurate clustering. Therefore, this motivate us to use small number of samples to represent each sample while considering local information. We propose a weighted l_1 norm $\|W \odot Z\|_1$

$$W = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{x}_i - \mathbf{x}_j\|^2}),$$

which forces each point is resprenseted by a small number of neighbor points. At the same time, we introduce a variable prior matrix D to encode different types of prior information through the $\|DZ\|_F^2$. For example, for temporal data, we set D as the first-order difference matrix to embed temporal information into the representation matrix. For mainfold data, we set $D = L^{1/2}$, which embeds manifold information into the representation matrix, i.e. manifold regularization. Then, we propose a new model considering the local weighted sparsity and global prior information as follows.

$$J_{FTV} = \min_{Z,E} \lambda_1\|W \odot Z\|_1 + \frac{\lambda_2}{2}\|DZ\|_F^2 + \lambda_3\|E\|_1 + \|G\|_F^2 \text{ s.t. } X = AZ + E + G, \tag{3.1}$$

where $Z \in^{d\times d}$ is the representation matrix. $W \in R^{d\times d}$ is a non-negative weight matrix used to impose adaptive sparsity constraints on different elements in the representation matrix. $D \in R^{(d-1)\times d}$ is a structural constraint operator that explicitly encodes the known or assumed structural relationships between data samples by acting on the coefficient matrix Z. $X \in R^{m\times n}$ is the given data matrix. $E \in R^{m\times n}$ and $G \in R^{m\times n}$ represent impulse noise and Gaussian noise, respectively. The penalty parameters λ_1, λ_2, λ_3 ¿ 0.

4. Problem Solution

We adopt the splitting algorithm to optimize our proposed model. Before detailing the optimization procedure, We first convert the formulation (3.1) to the following equivalent problem (4.1) by introducing a auxiliary variable M.

$$\begin{aligned} &\min_{Z,E} \lambda_1\|W \odot M\|_1 + \frac{\lambda_2}{2}\|DZ\|_F^2 + \lambda_3\|E\|_1 + \|G\|_F^2, \\ &\text{s.t. } X = AZ + E + G, Z = M. \end{aligned} \tag{4.1}$$

the partially augmented Lagrangian multiplier problem in problem (4.1) is

$$\begin{aligned} \mathcal{L}(Z, M, E, G, Y) =& \lambda_1\|W \odot M\|_1 + \frac{\lambda_2}{2}\|DZ\|_F^2 + \lambda_3\|E\|_1 + \|G\|_F^2 \\ &+ < Y, X - AZ - E - G > + \frac{\mu}{2}\|X - AZ - E - G\|_F^2. \end{aligned} \tag{4.2}$$

where $Y \in R^{m\times n}$ is the Lagrange multiplier, and $\mu > 0$ is the penalty parameter. The iteration scheme is as follows

$$
\begin{aligned}
(Z^{k+1}, M^{k+1}) =& \underset{Z,M}{argmin} \lambda_1 \|W \odot M\|_1 + \frac{\lambda_2}{2}\|DZ\|_F^2 + \frac{\mu}{2}\|X - AZ - E^k - G^k + \frac{Y^k}{\mu}\|_F^2, \\
& \text{s.t. } Z = M, \\
E^{k+1} =& \underset{E}{argmin} \lambda_3 \|E\|_1 + \frac{\mu}{2}\|X - AZ^{k+1} - E - G^k + \frac{Y^k}{\mu}\|_F^2, \\
G^{k+1} =& \underset{G}{argmin} \|G\|_F^2 + \frac{\mu}{2}\|X - AZ^{k+1} - E^{k+1} - G + \frac{Y^k}{\mu}\|_F^2, \\
Y^{k+1} =& Y^k + \mu(X - AZ^{k+1} - E^{k+1} - G^{k+1}).
\end{aligned} \tag{4.3}
$$

4.1. **Update**(Z^{k+1}, M^{k+1})**.** As we can see in (4.3), when solving the first subproblem, since it is a convex optimization problem with constrained two variables, we use ADMM to solve the subproblem. By introducing the multiplier Λ, then we construct the following augmented Lagrangian function

$$
\begin{aligned}
\mathcal{L}(Z, M, \Lambda) =& \lambda_1 \|W \odot M\|_1 + \frac{\lambda_2}{2}\|DZ\|_F^2 + \frac{\mu}{2}\|X - AZ - E^k - G^k + \frac{Y^k}{\mu}\|_F^2 \\
& + \frac{\mu_1}{2}\|Z - M + \frac{\Lambda_i}{\mu_1}\|_F^2.
\end{aligned} \tag{4.4}
$$

(1)Fix others, update M_{i+1}^k by optimize the following funciton

$$
\begin{aligned}
M_{i+1}^k &= \underset{M^k}{argmin} \lambda_1 \|W \odot M^k\|_1 + \frac{\mu_1}{2}\|Z_i^k - M^k + \frac{\Lambda_i}{\mu_1}\|_F^2 \\
&= \mathcal{S}_{\frac{\lambda_1}{\mu_1} W}(Z_i^k + \frac{\Lambda_i}{\mu_1}),
\end{aligned}
$$

where $\mathcal{S}(\cdot)$ is a soft threshold operator, and its closed form solution can be obtained according to the lemma2.1.

(2)Fix others, update Z_{i+1}^k by optimize the following funciton

$$
Z_{i+1}^k = \underset{Z^k}{argmin} \frac{\lambda_2}{2}\|DZ^k\|_F^2 + \frac{\mu}{2}\|X - AZ^k - E^k - G^k + \frac{Y^k}{\mu}\|_F^2 + \frac{\mu_1}{2}\|Z^k - M_{i+1}^k + \frac{\Lambda_i}{\mu_1}\|_F^2.
$$

It is obvious that the objective function is quadratic function. Through a simple derivation operation, we have

$$
Z_{i+1}^k = (\lambda_2 D^T D + \mu A^T A + \mu_1 I)^{-1}(\mu A^T B + \mu_1 M_{i+1}^k - \Lambda_i),
$$

where $B = X - E^k - G^k + \frac{Y^k}{\mu}$.

(3)Fix others, update Λ_{i+1}, we use

$$
\Lambda_{i+1} = \Lambda_i + \mu_1(Z_{i+1}^k - M_{i+1}^k).
$$

After obtaining the final Z_i^k and $M_i^k (1 \le i \le t)$, let $Z^{k+1} = Z_i^k$ and $M^{k+1} = M_i^k (1 \le i \le t)$. To further update E and G.

4.2. **Update** E^{k+1}**.** Fix others, update E^{k+1} by optimize the following funciton

$$
E^{k+1} = \mathcal{S}_{\frac{\lambda_3}{\mu}}(X - AZ^{k+1} - G^k + \frac{Y^k}{\mu}). \tag{4.5}
$$

its closed form solution can be obtained according to the lemma 2.1.

4.3. **Update G^{k+1}.** Fix others, update G^{k+1} by optimize the following funciton

$$G^{k+1} = \frac{\mu+2}{\mu}(X - XZ^{k+1} - E^{k+1} - \frac{Y^k}{\mu}). \tag{4.6}$$

The whole procedure to obtain the solution of (3.1) is given in Algorithm 1

Algorithm 1: Solving Problem(4.1) by Algorithm 1

input: Data matrix X, parameter $\lambda_1, \lambda_2, \lambda_3, \lambda_4, t$;
Initialize: $Z^0 = 0, M^0 = 0, E^0 = 0, G^0 = 0, Y^0 = 0, \Lambda_1 = 0, \varepsilon = 10^{-6}, \mu = 1.2, k = 0$;
While $\|X - AZ - E - G\|_\infty > \varepsilon$ or $\|Z - M\|_\infty > \varepsilon$ **do**
  step 1:
    for $i = 1, \cdots, t$ **do**
    Fix Z_i^k and update M_{i+1}^k by Eq.(4.1);
    Fix M_{i+1}^k and update Z_{i+1}^k by Eq.(4.1);
    Fix (Z_{i+1}^k, M_{i+1}^k) and update Λ_{i+1} by Eq.(4.1) ;
    end for
  step 2: Let $Z_t^k = Z^{k+1}$, $M_t^k = M^{k+1}$. Update E^{k+1} by Eq.(4.5) ;
  step 3: Update G^{k+1} by Eq.(4.6);
  step 4: Update Y^{k+1} by Eq.(4.3);
Output: $Z^{k+1}, M^{k+1}, E^{k+1}, G^{k+1}, Y^{k+1}$.

5. Experiments

In this section, we evaluate the quality and efficiency of our method through four parts of experiments: synthetic datasets, four real world datasets, action clustering and two moon dataset. All experiments are conducted on a computer with Intel(R) Core(TM) i5-12400F (2.50 GHz) CPU and 32GB RAM in the MATLAB R2018b environment.

All experiments use the data matrix itself X as a dictionary matrix, i.e. $A = X$. In addition, to ensure the consistency of sparse data representation, unless otherwise specified, the structural constraint matrix D in the experiment is defined as follows

$$D = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}_{(n-1)\times n} \tag{5.1}$$

After obtaining the representation matrix Z^* through Algorithm 1, we calculate the affinity matrix $W = \frac{|Z^*|+|Z^*|^T}{2}$. Then, the normalized cuts(NCut)[21] is applied to affinity matrix W to produce the clustering results. Finally, we employ three universal evaluation metrics to measure the clustering quality of various methods. They are clustering Accuracy(ACC), Normalized Mutual information(NMI) and CPU time, the definition as described below.

Definition 5.1. Clustering Accuracy (ACC)[11]is formally defined as:

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}.$$

Specifically, s_i and r_i denote the predicted cluster labels and ground-truth labels; $\delta(x, y)$ is delta function, which means that if $x = y, \delta(x, y) = 1$, otherwise $\delta(x, y) = 0$; $map(\cdot)$ is the optimal mapping function, implemented using the Hungarian algorithm, to find the best correspondence between the clustering results and the actual labels.

Definition 5.2. Normalized Mutual Information (NMI)[17]is formally defined as:

$$\mathrm{NMI} = \frac{-2\sum_{i=1}^{S}\sum_{j=1}^{R} C_{ij}\log\frac{C_{ij}N}{C_iC_j}}{\sum_{i=1}^{S} C_i\log\frac{C_i}{N} + \sum_{j=1}^{R} C_j\log\frac{C_j}{N}}.$$

In the formulation above, C_i and C_j denote the number of samples assigned to the i-th cluster in A and the j-th class in B, respectively, while C_{ij} represents the number of samples that belong to both the i-th cluster in A and the j-th ground truth class in B. A higher Normalized Mutual Information value indicates superior clustering quality.

5.1. **Synthetic dataset.** Following the synthetic data generation method in literatures[14, 24], we firstly constract five independent linear subspaces $\{S_i\}_{i=1}^{5}$, where the five basis matrices $\{U_i\}_{i=1}^{5}$ are generated by $U_{i+1} = TU_i(1 \le i \le 4)$, with T representing a random rotation matrix, and U_i being a random column orthogonal matrix of dimension 150 × 4. Thus, each linear subspace is 4-dimensional, and the high-dimensional space where each sample point resides is 150-dimensional. If each subspace generates $\overline{n}$ sample points by $X_i^0 = U_iQ_i(1 \le i \le 5)$, where Q_i is a $4\times\overline{n}$ independently identically distributed standard Gaussian matrix. Finally, the data matrix is obtained as $X = [X_1^0, \cdots, X_5^0] \in \mathbb{R}^{150\times n}$, where the rank $r = 20$ and $n = 5 \times \overline{n}$. Additionally, a set of observable data point indices Ω is randomly generated, and sr denotes the proportion of observable data, where $|\Omega|/mn$. Then, data sample points are randomly selected in proportion to add Gaussian noise and impulse noise. Specifically, 20% of the data samples are randomly selected to add Gaussian noise with a mean of 0 and a variance of 0.01. Impulse noise is generated by a uniform distribution on [-1,1], and spr represents the proportion of the data matrix contaminated by impulse noise.

Based on the above synthesized data, we compare FTV with S^{4}CMR, SSC-L1TG, and SSIL. We have listed the main parameter selections of these four methods in Table 1, and the remaining parameters are set to the default settings of the original paper.

TABLE 1. Parameter selection of various methods on synthetic dataset

Synthetic dataset	S^{4}CMR (λ, μ)	SSC-L1TG (λ_1, λ_2)	SSIL (λ_1, λ_2)	FTV $(\lambda_1, \lambda_2, \lambda_3)$
	(1, 2)	(0.03, 0.001)	(0.01, 0.1)	(0.001, 0.3, 0.8)

We fix the data dimension to $n = 500$, with different ratios of sr and spr, to test the robustness of all the above methods under different levels of noise and missing values. We place the experimental results in Table 2.

From Table 2, it is clear that the FTV method consistently maintains higher clustering accuracy and shorter computation time. Especially when $sr = 0.1, spr = 0.10$, the accuracy of the comparison methods significantly decreases, while the FTV accuracy only decreases by about 6%.

Next, we analyze the clustering performance of various methods by gradually increasing the number of synthesized data samples. Specifically, we randomly generated test samples with a spacing of 300, and the number of sample points gradually increased from 300 to 900. All experiments are randomly repeated five times, and the average clustering accuracy and time of the five iterations are recorded. It should be noted that all parameter settings remain consistent with the previous description. The results are detailed in Table 3.

Table 3 shows that as the data dimension n increases, the computation time of all methods shows a upward trend. However, compared with other methods, the proposed FTV method performs better, with the shortest computation time and the highest clustering accuracy among all test dimensions.

TABLE 2. Clustering accuracy (%) and CPU time (s) of various methods on synthetic datasets with different levels of noise

spr	*sr*	S^{4}CMR		SSC-L1TG		SSIL		FTV	
		Time	ACC	Time	ACC	Time	ACC	Time	ACC
0.05	0.1	2.31	20.00	1.68	91.04	1.54	63.00	**0.47**	**95.24**
	0.3	2.53	50.88	1.68	92.48	1.25	80.08	**0.47**	**98.76**
	0.5	2.47	74.64	1.69	93.56	2.04	80.60	**0.48**	**99.12**
	0.7	2.74	92.04	1.68	95.32	1.74	80.80	**0.49**	**99.64**
	0.9	2.55	97.04	1.69	95.80	1.75	81.64	**0.52**	**99.68**
0.10	0.1	2.27	23.96	1.65	52.24	1.30	68.76	**0.47**	**89.28**
	0.3	2.33	49.24	1.67	88.60	1.50	76.04	**0.48**	**95.88**
	0.5	2.40	71.12	1.67	89.00	1.57	77.48	**0.49**	**97.92**
	0.7	2.35	87.32	1.68	89.36	2.76	78.32	**0.48**	**98.96**
	0.9	2.40	92.12	1.69	89.48	1.64	78.08	**0.50**	**99.40**

TABLE 3. Clustering accuracy (%) and CPU time (s) of various methods in different dimensions on synthetic datasets

			S^{4}CMR		SSC-L1TG		SSIL		FTV	
n	*spr*	*sr*	Time	ACC	Time	ACC	Time	ACC	Time	ACC
300	0.05	0.6	0.54	63.20	0.34	93.60	0.29	80.40	**0.11**	**99.53**
		0.9	0.52	97.20	0.34	94.13	0.28	81.80	**0.12**	**99.80**
	0.10	0.6	0.53	62.33	0.35	89.73	0.38	78.40	**0.11**	**99.33**
		0.9	0.52	93.33	0.34	89.60	0.28	78.47	**0.12**	**99.53**
600	0.05	0.6	3.88	94.20	2.55	92.87	5.16	80.97	**0.71**	**99.37**
		0.9	4.28	96.93	2.55	94.47	2.63	80.67	**0.76**	**99.53**
	0.10	0.6	3.60	80.77	2.54	88.70	3.63	77.13	**0.71**	**97.67**
		0.9	3.64	90.33	2.54	87.40	3.16	73.87	**0.72**	**98.13**
900	0.05	0.6	12.76	95.20	6.22	93.78	5.92	79.82	**1.72**	**98.49**
		0.9	12.78	97.31	6.24	93.91	9.68	80.49	**1.74**	**98.80**
	0.10	0.6	9.59	88.04	6.17	88.82	6.93	77.67	**1.73**	**97.58**
		0.9	11.13	92.67	6.19	86.51	8.23	77.07	**1.73**	**98.58**

Then, we conduct parameter sensitivity analysis using the synthetic dataset mentioned above. We set $n = 500$, $sr = 0.9$, and $spr = 0.05$. For each parameter λ_i, we vary one of them, while the others are set as section 5.1. Each experiment we randomly repeat 5 times, and finally record the average clustering accuracy of these 5 times.

Effect of λ_1, λ_2: λ_1 and λ_2 serve as trade-offs between sparse terms and structural low-rank terms. From Figure 1 and Figure 2, it is shown that compared to other methods, the clustering accuracy of our method is relatively stable in the whole selected interval. And the clustering accuracy of the entire selected interval is not less than 95%.

Effect of λ_3: λ_3 serve as trade-offs between sparse low-rank terms and noise term. From the entire interval, the clustering accuracy remains stable, and the clustering accuracy is not less than 99%.

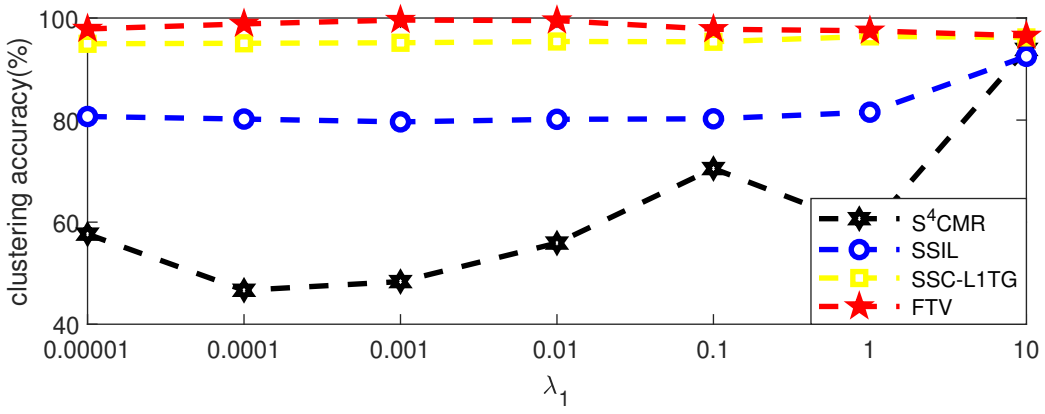


FIGURE 1. The clustering accuracy of various methods varies with λ_1

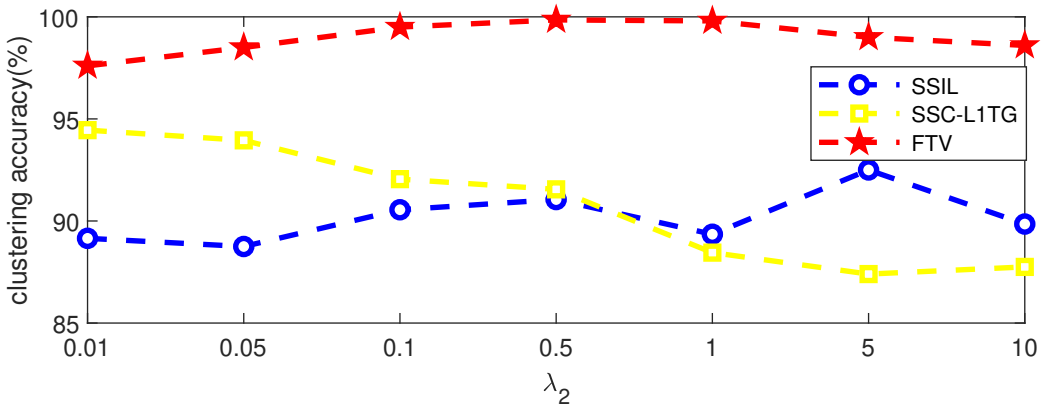


FIGURE 2. The clustering accuracy of various methods varies with λ_2

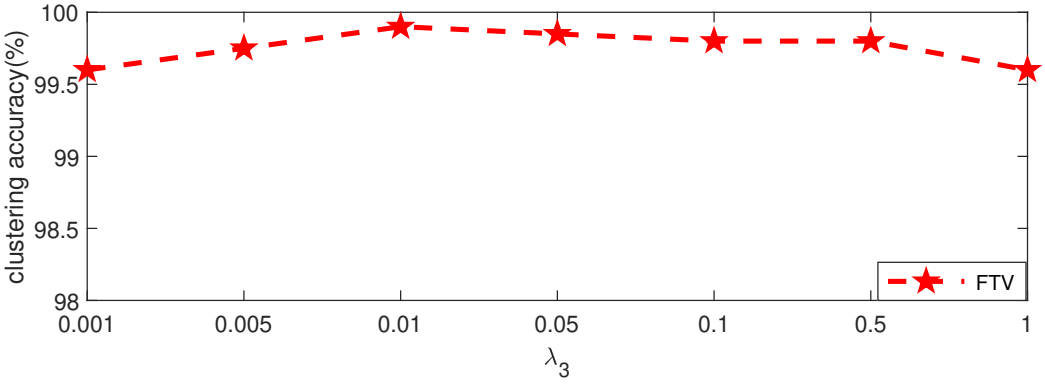


FIGURE 3. The clustering accuracy of FTV varies with λ_3

5.2. **Real-world dataset.** In this section, we test the effectiveness of our proposed FTV method through experiments on three real-world datasets. The comparison methods are also SSIL, SSC-L1TG, and S^4CMR. The details of each dataset is as follows

Extended Yale B dataset [12]: This dataset contains 2414 facial images taken by 38 subjects under 9 different poses and 64 lighting conditions.

ORL dataset [20]: This dataset contains 400 facial images of 40 subjects, each with a size of 32 × 32 pixels. Some images exhibit changes in facial expressions, posture, and lighting.

USPS dataset [5]: This dataset contains 9298 images of different handwritten digits and letters, each with a size of 16 × 16 pixels.

COIL20 dataset [2]: This dataset consists of 1440 images of 20 different objects. each target includes 72 images captured from different perspectives (every 5 degrees of rotation). Each image has a size of 32 × 32 pixels.

Partial sample images are shown in Figure 4. Please refer to Table 4 for specific main parameters for following experiments, and the remaining parameters are still set as default. To reduce the computational cost and memory requirements of all methods, we partially sampled the images. For four real datasets, we take even classes from the top 2 to 10, that is $n \in \{2, 4, 6, 8, 10\}$. The clustering experiment results are placed in Tables 5, 6, 7, and 8, respectively.

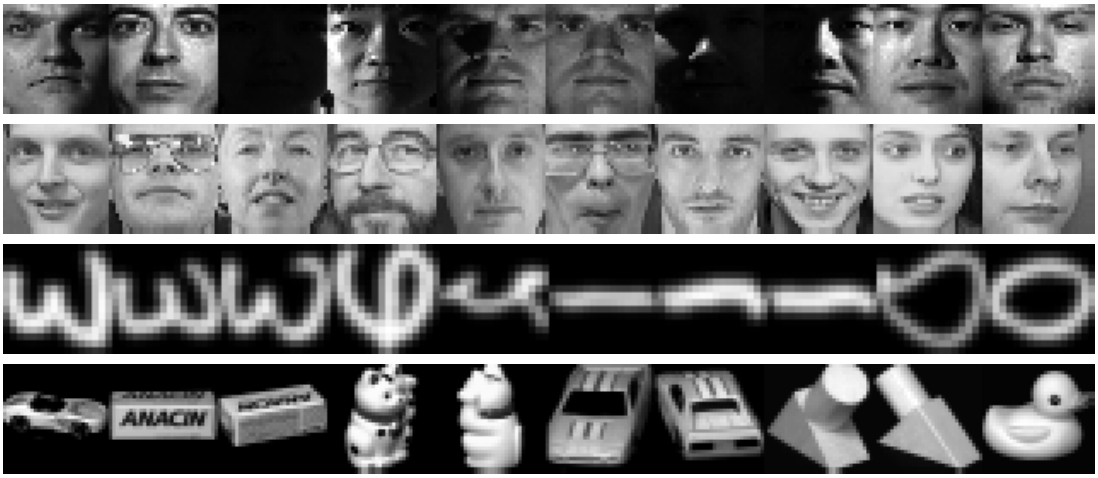


FIGURE 4. Partial images from Extended YaleB, ORL, USPS, COIL20 datasets (From top to bottom)

From Tables 5, 6, 7, and 8, we can see that, FTV takes the shortest time. Especially in the large-scale datasets USPS and COIL20, the comparison method takes almost two to three times longer than ours. In addtion, in most cases, FTV preforms better than compared methods in clustering accuracy and NMI.

Specifically, In Table 5, FTV achieved comparable and even higher accuracy at all points in a shorter time. Meanwhile, the performance degradation rate of FTV is slower, demonstrating stronger robustness.

In Table 6, FTV achieves 100% clustering accuracy and NMI for $ncluster = 2$ and $ncluster = 4$. In addition, within the entire selected category, FTV takes the shortest time, the ACC of FTV is not less than 98%, and the NMI is not less than 90%.

TABLE 4. Parameter values of various methods on different datasets

Dataset	S^4CMR (λ, μ)	SSC-L1TG (λ_1, λ_2)	SSIL (λ_1, λ_2)	FTV (λ_1, λ_2, λ_3)
Yale B	(0.1, 0.5)	(0.5, 0.2)	(0.01, 10)	(0.001, 1.1, 13)
ORL	(3, 0.3)	(0.5, 0.3)	(0.01, 14)	(0.01, 2.2, 8)
USPS	(0.1, 0.5)	(0.99, 0.01)	(0.3, 10)	(0.001, 0.9, 0.5)
COIL20	(0.1, 0.5)	(0.8, 0.001)	(0.5, 10)	(0.6, 0.25, 0.21)

TABLE 5. Clustering accuracy (%) and CPU time (s) of various methods on the Extended Yale B dataset

ncluster	S^4CMR Time—ACC—NMI	SSC-L1TG Time—ACC—NMI	SSIL Time—ACC—NMI	FTV Time—ACC—NMI
2	0.07—96.09—89.93	0.05—97.66—86.26	0.05—**99.22—100.00**	**0.01—99.22**—94.18
4	0.27—94.14—90.11	0.17—85.94—78.20	0.24—**98.44—95.00**	**0.08**—97.66—92.60
6	1.13—85.68—74.13	0.95—63.54—71.96	0.54—86.61—71.80	**0.31—95.31—88.31**
8	2.08—81.05—74.48	1.79—58.40—72.19	1.30—79.09—76.81	**0.75—81.45—78.77**
10	3.78—70.63—75.01	3.12—52.81—73.32	4.95—69.84—72.11	**1.36—76.25—78.22**

TABLE 6. Clustering accuracy (%) and CPU time (s) of various methods on the ORL dataset

ncluster	S^4CMR Time—ACC—NMI	SSC-L1TG Time—ACC—NMI	SSIL Time—ACC—NMI	FTV Time—ACC—NMI
2	0.02—**100.00**—92.19	0.03—**100.00—100.00**	0.07—96.53—81.60	**0.01—100.00—100.00**
4	0.07—66.67—80.20	0.07—64.00—65.35	0.14—97.57—91.05	**0.06—100.00—100.00**
6	0.17—75.00—87.26	0.14—58.33—61.27	0.20—96.99—82.19	**0.11—98.33—90.88**
8	0.21—78.75—87.68	0.24—67.50—69.67	0.31—97.92—90.38	**0.20—98.75—92.15**
10	0.68—68.00—79.57	0.36—67.00—66.53	0.36—97.78—94.33	**0.28—99.00—97.15**

TABLE 7. Clustering accuracy (%) and CPU time (s) of various methods on the USPS dataset

ncluster	S^4CMR Time—ACC—NMI	SSC-L1TG Time—ACC—NMI	SSIL Time—ACC—NMI	FTV Time—ACC—NMI
2	0.16—99.50—95.96	0.12—87.00—76.10	0.16—**100.00—100.00**	**0.05—100.00—100.00**
4	1.31—95.75—89.78	1.04—70.25—72.57	1.62—99.50—97.22	**0.33—99.75—98.23**
6	3.21—91.50—90.38	2.72—53.33—64.97	1.46—83.00—88.59	**0.91—99.50—96.24**
8	6.44—90.25—82.96	5.26—49.25—76.61	3.62—85.50—82.17	**1.84—98.50—95.27**
10	11.18—84.50—83.17	9.02—47.40—71.65	9.00—86.70—76.03	**3.11—98.30—95.77**

TABLE 8. Clustering accuracy (%) and CPU time (s) of various methods on the COIL20 dataset

ncluster	S^4CMR Time—ACC—NMI	SSC-L1TG Time—ACC—NMI	SSIL Time—ACC—NMI	FTV Time—ACC—NMI
2	1.26—95.00—72.30	0.06—**100.00—100.00**	0.16—96.53—81.60	**0.06—100.00—100.00**
4	2.47—96.25—77.81	0.24—97.92—84.41	1.62—97.57—91.05	**0.22—100.00—100.00**
6	6.94—81.17—84.61	1.22—79.86—72.59	2.08—96.99—82.19	**1.12—100.00—100.00**
8	11.63—81.50—78.08	2.39—81.94—84.00	3.55—97.92—90.38	**2.25—100.00—100.00**
10	24.88—75.60—81.04	4.69—75.42—82.36	9.27—90.00—94.33	**4.65—100.00—100.00**

In Table 7, FTV achieves 100% clustering accuracy and NMI for $ncluster = 2$. In addition, within the entire selected category, FTV takes the shortest time, the ACC of FTV is not less than 98%, and the NMI is not less than 95%.

In Table 8, the performance of FTV is perfect. The clustering accuracy and the NMI are all 100% for all ncluster values. In contrast, other methods show a significant decline in accuracy as the cluser increases.

5.3. **Action clustering.** The Weizmann dataset[7] contains 90 video sequences captured in outdoor environments, recording 10 different actions performed by 9 subjects. Each video has a resolution of 144 × 180 pixels and is recorded at a frame rate of 50 frames per second. Partial samples are shown in Figure 5. Due to each sequence in the Weizmann dataset containing only a single action, we generate longer continuous data by connecting individual sequences. Specifically, we constructed three long motion sequences: Weizmann3 (including bending, jumping, and placing jumps), Weizmann6 (including walking, jumping forward, waving 2, placing jumps, jumping, and bending), and Weizmann9 (including walking, jumping foreground, running, jumping, waving 1, bending, jumping, waving 2, and placing jumps). Before starting the experiment, we adjusted each image to 72 × 90 pixels and vectorized it into 6480 dimensional data points. Subsequently, we concatenate the vectorized image with consecutive frames.

The parameters of each method have been adjusted to the optimal level. We place the main parameter selections for each method in Table 9, other parameters are default settings. We present the experimental results of the three datasets in Table 10. In addition, For better visualization of results, we present the corresponding segmentation results in Figure 6, Figure 7, and Figure 8, respectively.

TABLE 9. Parameter selection of various methods on Weizmann dataset

OSC[22] (λ_1, λ_2)	**TSC**[13] (λ_1, λ_2)	**SSC-L1TG** (λ_1, λ_2)	**SSIL** (λ_1, λ_2)	**FTV** $(\lambda_1, \lambda_2, \lambda_3)$
(0.6, 0.1)	(0.01, 15)	(3, 0.001)	(0.5, 0.1)	(0.4, 0.2, 0.2)

Table 10 demonstrate that our FTV method outperforms comparing methods in all cases on Weizmann dataset. Especially when $k = 3$, the ACC is 100.00%. Forthermore, from Figure 6, Figure 7, and Figure 8, we can see that the experimental values of FTV generated segmentation labels have a higher alignment with the real values, further demonstrating the effectiveness of our method in temporal data.

5.4. **Manifold data clustering.** As mentioned before, our model can handle different data according to the different D values. So we set $D = L^{1/2}$ to handle the manifold data, that our proposed model

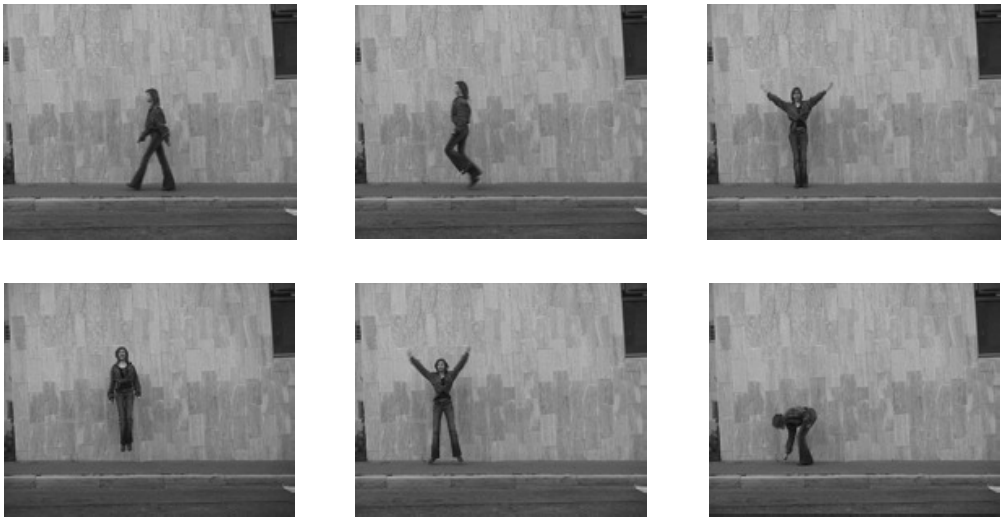


FIGURE 5. Partial actions in Weizmann dataset

TABLE 10. ACC (%) and NMI (%) of different methods on Weizmann dataset

	$k=3$		$k=6$		$k=9$	
method	ACC	NMI	ACC	NMI	ACC	NMI
OSC	81.64	77.42	78.34	72.86	62.69	73.65
TSC	88.94	86.45	74.59	80.06	66.36	74.37
SSC-L1TG	89.36	80.59	74.18	79.17	53.24	70.15
SSIL	98.30	91.62	84.25	76.71	71.91	75.84
FTV	**100.00**	**92.46**	**87.53**	**87.70**	**74.38**	**78.11**

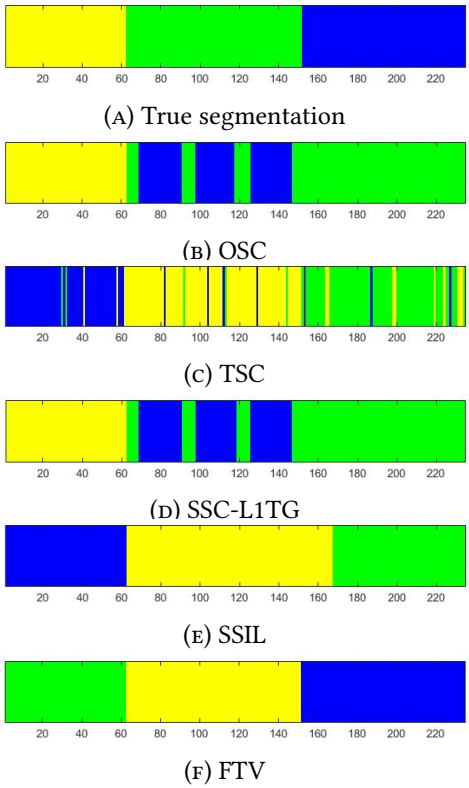
(3.1) is transformed into the following model-FTV-Laplace

$$\min_{Z,E} \lambda_1\|W \odot Z\|_1 + \lambda_2 tr(ZLZ^T) + \lambda_3\|E\|_1 + \lambda_4\|G\|_F^2 \text{ s.t. } X = AZ + E + G. \tag{5.2}$$

Following the generation method described in reference [29], the two moon data consists of two types of data with a total of 400 data points, shaped like two crescent moons. Then, we test the clustering performance of the FTV-origin, FTV-Laplace, S^{4}CMR, SSC-L1TG, and SSIL on this dataset. We present the experimental results in Figure 9. For each method, the regularization parameters have been optimized to achieve optimal performance, as shown in Table 11.

TABLE 11. Parameter settings of different methods on two moons dataset

S^{4}CMR (λ_1, λ_2)	**SSC-L1TG** (λ_1, λ_2)	**SSIL** (λ_1, λ_2)	**FTV-origin** (λ_1, λ_2, λ_3)	**FTV-Laplace** (λ_1, λ_2, λ_3)
(0.01, 2)	(0.06, 0.1)	(2, 0.01)	(1, 5, 1)	(1, 5, 1)



(A) True segmentation

(B) OSC

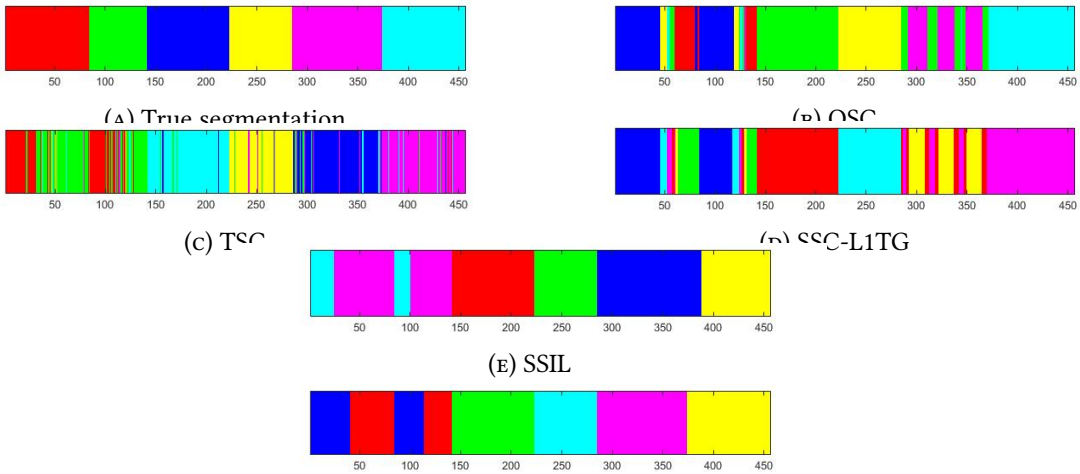
(C) TSC

(D) SSC-L1TG

(E) SSIL

(F) FTV

FIGURE 6. Weizmann3



(A) True segmentation

(B) OSC

(C) TSC

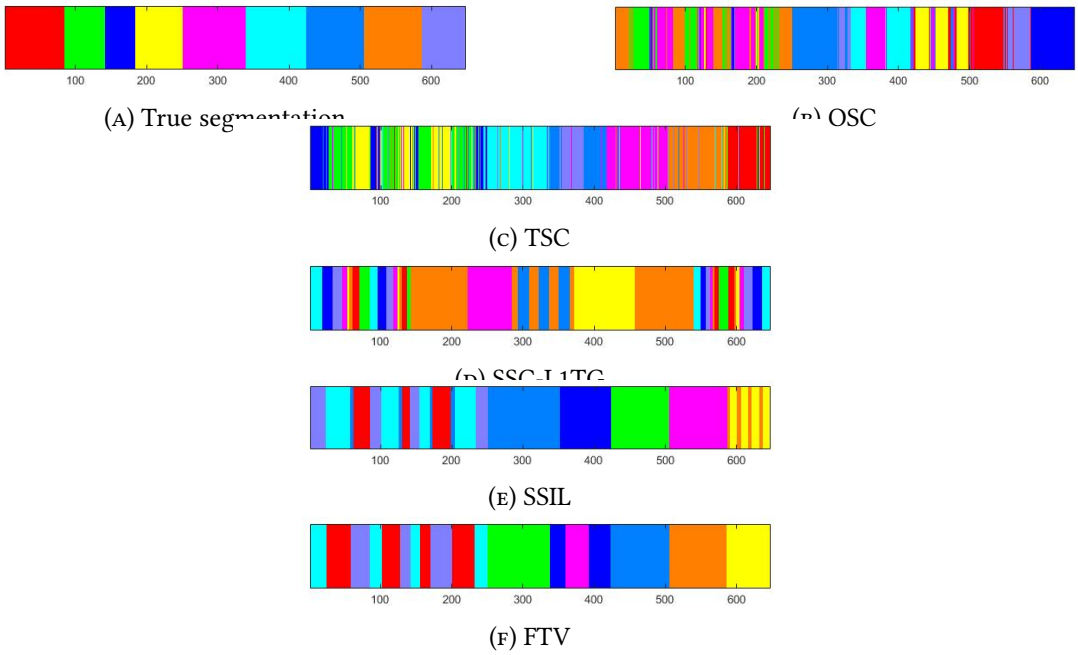
(D) SSC-L1TG

(E) SSIL

(F) FTV

FIGURE 7. Segmentation results on Weizmann6

Figure 9 demonstrates the clustering results of different methods on a two moons dataset. It can be seen from Figure 9(b) and Figure 9(c) that the clustering effect of SSC-L1TG on manifold structures is limited, with clustering accuracies of 67.25% and 48.50%, respectively. Figure 9(d) shows that although both S^4CMR and SSIL methods consider manifold regularization, their clustering accuracy on this dataset is only 51.13%. In contrast, the clustering results of Figure 9(e) and Figure 9(f) clearly



(A) True segmentation (B) OSC

(C) TSC

(D) SSC-L1TG

(E) SSIL

(F) FTV

FIGURE 8. Segmentation results on Weizmann9

demonstrate that FTV-Origin and the FTV-Laplace method proposed in this chapter can effectively partition two moon dataset, achieving clustering accuracies of 91.00% and 100.00%, respectively, indicating that our method can also address the problem of manifold data structure.

6. CONCLUSION

In this paper, we propose a subspace clustering model based on local weighted sparsity and global prior information, utilizing the distance information of the data itself to guide the construction of coefficient matrix, enhancing the discriminative ability. Simultaneously selecting appropriate structural prior information to flexibly adapt to the prior structures of different data. In addition, by integrating sparse noise and Gaussian noise, the model has stronger robustness for the real-world data. To efficiently solve the model, we introduced a splitting algorithm based on the augmented Lagrangian framework, which theoretically ensures convergence. A large number of experimental results on synthetic data, four real-world datasets, temporal action datasets, and manifold datasets show that our method outperforms existing methods in clustering accuracy, computational efficiency, and noise robustness.

STATEMENTS AND DECLARATIONS

The authors declare that they have no conflict of interest, and the manuscript has no associated data.

ACKNOWLEDGMENTS

The first author was supported by the Postgraduate Scientific Research and Innovation Project of Chongqing Jiaotong University(2025ST005). The second author was supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202500718). The third author was supported by the National Natural Science Foundation of China (12271067), the Chongqing Natural Science Foundation (CSTB2024NSCQ-MSX0973) and the Science and Technology Research Key Program of Chongqing Municipal Education Commission (KJZD-K202200704).

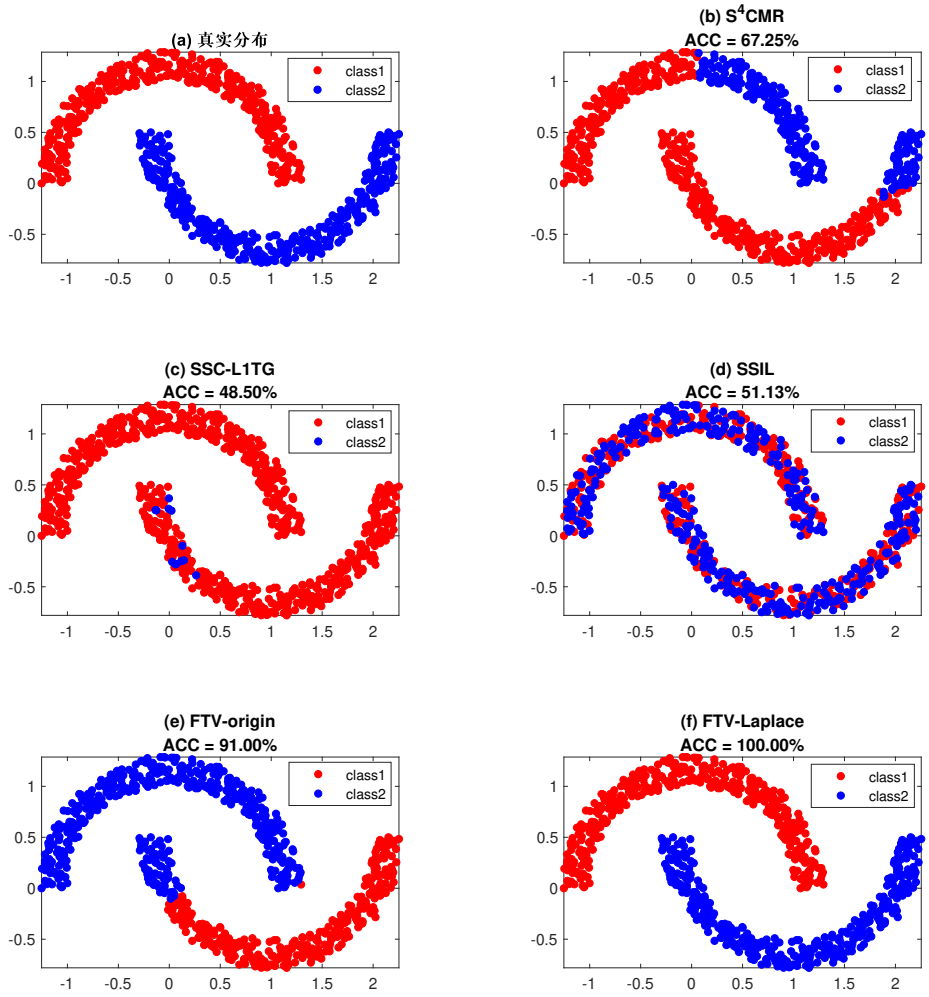


FIGURE 9. clustering results on two moon dataset

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 94–105, USA, 1998. Association for Computing Machinery, New York.

[2] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, 58(3):1–37, 2011.

[4] J. Chen and J. Yang. Robust subspace segmentation via low-rank representation. *IEEE Transactions on Cybernetics*, 44(8):1432–1445, 2014.

[5] D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 46(1):161–190, 2002.

[6] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, Miami, USA, 2009. IEEE, New york.

[7] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[8] Y. Guo, J. Gao, and F. Li. Spatial subspace learning for hyperspectral data segmentation. In *International Conference on Digital Information Processing and Communications (ICDIP)*, pages 180–190. Society of Digital Information and Wireless Communications (SDIWC), 2013.

[9] W. Hu, H. Huang, and T. Wang. One-step temporal subspace clustering with structured sparsity and indicator graph-laplacian regularization. *Expert Systems with Applications*, 296:Article ID 128889, 2026.

[10] W. Hu, S. Li, W. Zheng, Y. Lu, and G. Yu. Robust sequential subspace clustering via ℓ_1-norm temporal graph. *Neurocomputing*, 383:380–395, 2020.

[11] X. Ji, S. Liu, L. Yang, W. Ye, and P. Zhao. Clustering ensemble based on approximate accuracy of the equivalence granularity. *Applied Soft Computing*, 129:Article ID 109492, 2022.

[12] K.-C. Lee, J. Ho, and D. J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[13] S. Li, K. Li, and Y. Fu. Temporal subspace clustering for human motion segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4453–4461, Santiago, Chile, 2015. IEEE, New York.

[14] Y. Liu, L. Jiao, and F. Shang. A fast tri-factorization method for low-rank matrix recovery and completion. *Pattern Recognition*, 46(1):163–173, 2013.

[15] 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 A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *European conference on Computer Vision*, volume 7578, pages 347–360, ECCV 2012, 2012. Springer, Berlin, Heidelberg.

[16] L. Lu and R. Vidal. Combined central and subspace clustering for computer vision applications. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 593–600, Pennsylvania, USA, 2006. Association for Computing Machinery, New York.

[17] A. Mahmoudi and D. Jemielniak. Proof of biased behavior of normalized mutual information. *Scientific Reports*, 14(1):Article ID 9021, 2024.

[18] B. Nasihatkon and R. Hartley. Graph connectivity in sparse subspace clustering. In *CVPR 2011*, pages 2137–2144, Colorado, USA, 2011. IEEE, New York.

[19] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

[20] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, Sarasota, USA, 1994. IEEE, New York.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[22] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1026, Columbus, USA, 2014. IEEE, New York.

[23] Y.-X. Wang, H. Xu, and C. Leng. Provable subspace clustering: when LRR meets SSC. In C. J. C. Burges, L. Bottou, and M. Welling, editors, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 64–72, Lake Tahoe, USA, 2013. Curran Associates Inc., New York.

[24] Y. Xiao, S.-Y. Wu, and D.-H. Li. Splitting and linearizing augmented lagrangian algorithm for subspace recovery from corrupted observations. *Advances in Computational Mathematics*, 38(4):837–858, 2013.

[25] Z. Xing, J. Peng, X. He, and M. Tian. Semi-supervised sparse subspace clustering with manifold regularization. *Applied Intelligence*, 54(9-10):6836–6845, 2024.

[26] J. Yang, W. Yin, Y. Zhang, and Y. Wang. A fast algorithm for edge-preserving variational multichannel image restoration. *SIAM Journal on Imaging Sciences*, 2(2):569–592, 2009.

[27] M. Yin, Z. Wu, D. Zeng, P. Li, and S. Xie. Sparse subspace clustering with jointly learning representation and affinity matrix. *Journal of the Franklin Institute*, 355(8):3795–3811, 2018.

[28] C. Zhang and R. R. Bitmead. Subspace system identification for training-based mimo channel estimation. *Automatica*, 41(9):1623–1632, 2005.

[29] Z. Zheng, J. Zhang, S. Zhu, C. Tang, F. Lin, H. Lan, Z. Chen, and J. Yang. Cupid: consistent unlabeled probability of identical distribution for image classification. *Knowledge-Based Systems*, 137:115–122, 2017.