

Subspace clustering by simultaneously feature selection and similarity learning[☆]



Guo Zhong, Chi-Man Pun^{*}

Department of Computer and Information Science, University of Macau, Macau SAR, China

ARTICLE INFO

Article history:

Received 2 September 2019

Received in revised form 12 November 2019

Accepted 10 January 2020

Available online 16 January 2020

Keywords:

Subspace clustering

Feature selection

Graph learning

Similarity learning

Affinity matrix

ABSTRACT

Learning a reliable affinity matrix is the key to achieving good performance for graph-based clustering methods. However, most of the current work usually directly constructs the affinity matrix from the raw data. It may seriously affect the clustering performance since the original data usually contain noises, even redundant features. On the other hand, although integrating manifold regularization into the framework of clustering algorithms can improve clustering results, some entries of the pre-computed affinity matrix on the original data may not reflect the true similarities between data points. To address the above issues, we propose a novel subspace clustering method to simultaneously learn the similarities between data points and conduct feature selection in a unified optimization framework. Specifically, we learn a high-quality graph under the guidance of a low-dimensional space of the original data such that the obtained affinity matrix can reflect the true similarities between data points as much as possible. A new algorithm based on augmented Lagrangian multiplier is designed to find the optimal solution to the problem effectively. Extensive experiments are conducted on benchmark datasets to demonstrate that our proposed method performs better against the state-of-the-art clustering methods.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Clustering is an important data analysis method in the field of data mining and machine learning. Its goal is to assign unlabeled objects to groups, where similar objects are expected to be assigned to the same group. With the development of electronic and social media, a large number of high-dimensional data have been produced in many practical applications [1,2]. Clustering high-dimensional data is a challenge for traditional clustering methods [3]. Because of clustering data according to their underlying subspace, spectral clustering shows great promising for clustering high-dimensional data [4,5].

Spectral clustering is also one of the most popular graph-based clustering methods. Usually, it consists of two steps, i.e., learning an affinity matrix and clustering data using eigenvectors of the affinity matrix. Because of taking into account the correlation between data points, learning an affinity matrix can be essentially regarded as learning a similarity graph. Viewed from this perspective, spectral clustering solves the problem of the optimal

graph partitioning. Hence the key of spectral clustering is to learn a good graph, in which the weight of each edge reflects the similarities between data points. Maybe the most intuitive way to learning a graph is to directly compute Euclidean distance on the original data, such as k -nearest neighbor graph, where k is the neighborhood size. However, Euclidean distance is sensitive to noise and outliers. Moreover, how to theoretically determine a proper neighbor number k is still an open issue [6]. Other parameters, such as the kernel width in conventional Gaussian kernel function, also influence the quality of the similarity graph significantly. Therefore, the learned graph may not be good enough for the subsequent graph partitioning. To overcome the drawback of the Euclidean-based similarity measurement, Sparse Subspace Clustering (SSC) [5], Low-rank Representation (LRR) [7] and Least Squares Regression (LSR) [8] take advantage of the self-expressiveness property of the data, i.e., each data point in a union of subspaces can be effectively reconstructed by a combination of other points in the dataset. The representation coefficients can be used to measure the similarities between data points, i.e., the larger the coefficients are, the more similar the corresponding two data points are, while the smaller the coefficients are, the more dissimilar they are. By imposing different constraints on the representative matrix, the obtained affinity matrices can capture different information from data. For example, SSC only aims at capturing the local structure of data

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.knosys.2020.105512>.

^{*} Corresponding author.

E-mail address: cmpun@um.edu.mo (C.-M. Pun).

while LRR and LSR only capture the global representation structure. Along this line, a lot of representation-based approaches are proposed [9–12]. These methods only aim to exploit the global or the local structure of data.

In recent years, researchers propose to impose the manifold regularization on the representative matrix to exploit the local structure of data, which enforces similar data points in the original space to have similar representations in the new space [13–15]. Although these algorithms show good performance in clustering, the graph Laplacian in the manifold regularization has to be pre-computed on the raw data. Some entries of the obtained affinity matrix in the graph Laplacian may not reflect the true similarities between data points, such as data points near the intersection of subspaces because the neighborhood of a data point can contain points from different subspaces [5]. More recently, one-step spectral learning methods with superior performance were proposed [16–18]. In [17], the state-of-the-art Constrained Laplacian Rank (CLR) method learns a graph adaptively with exact c connected components, where c is the cluster number, instead of performing clustering by the learned graph beforehand. Concretely, it utilizes the rank constraint on the Laplacian matrix of a dynamic graph, and the final clustering result is this dynamic graph. Although the ideal similarity graph should surely have exact c connected components, the condition of rank constraint is overstrict since the graph rarely consists of c truly disjoint connected components in practice [19]. As a result, the learned graph with the exact c connected components cannot guarantee that it is optimal for clustering. On another hand, the learned graph of CLR is constructed on the raw data. Thus it cannot well reveal the intrinsic structure of data.

Feature selection is an important pre-processing step in data mining. Faced with a large amount of high dimensional data, we are desired to obtain a compact representation of the original feature with good generalization ability via extracting essential features and eliminate noisy ones. Recently, unsupervised feature selection algorithms have received more and more attention. Unsupervised feature selection methods can be roughly divided into three categories: filter methods [20,21], wrapper methods [22] and embedded methods [23–26]. Specifically, embedded methods show superior performance. For example, [27] proposes a general framework for feature selection based on Flexible Manifold Embedding [28] and $l_{2,1}$ -norm. [29] jointly uses Graph Embedding and $l_{2,1}$ -norm to perform feature selection. More recently, [2] proposes an efficient method for robust unsupervised feature selection via dual self-representation, manifold regularization and $l_{2,1}$ -norm. However, these methods are not specifically designed for clustering.

Recently, research [30] shows that high-dimensional data usually lie in multiple low-dimensional subspaces. Thus, spectral clustering-based subspace clustering methods learn an affinity matrix via the self-expression property of data [5,7,8]. However, the affinity matrix learned directly from raw data can severely impact clustering performance because high-dimensional data are often noisy and may contain redundant dimensions. Moreover, performing feature selection and subsequent clustering in two separate steps may not obtain the optimal clustering results [31]. To derive a better affinity matrix, this paper proposes a subspace clustering method with an adaptive transformation matrix for data clustering (SCAT), which instead of pre-computing an affinity matrix from the original data, graph learning is taken as a part of the feature selection procedure, and the graph can be dynamically learned under the guidance of an intrinsic low-dimensional space of the original data. Hence the noise in the input data space is alleviated, and the local structure of data is learned. In the meantime, SCAT can also capture the global information of data with the help of the power of self-expression of

data. Hence the quality of the learned graph should be improved. The flowchart of our SCAT method is given in Fig. 1.

We briefly summarize the main contributions of our proposed SCAT method as follows:

- Distinct from the previous graph-based clustering approaches which perform clustering on a pre-computed affinity matrix or learn a dynamic graph from the raw data, SCAT dynamically learns an affinity matrix under the guidance of an intrinsic low-dimensional space of the original data, which guarantees to obtain high quality similarity graph since noise and redundancy are eliminated in the low dimensional space.
- The proposed SCAT can capture the global and the local structure of data simultaneously. Specifically, it adaptively learns local manifold structure, and thus can greatly improve the quality of the affinity matrix.
- We derive a new Augmented Lagrange Multiplier based algorithm to efficiently and effectively solve the associated optimization problem.
- Extensive experiments conducted on nine benchmark datasets demonstrate the effectiveness of the proposed approach. Specifically, SCAT consistently outperforms the state-of-the-art one-step clustering method CLR in terms of clustering accuracy and normalized mutual information, respectively.

The structure of this paper is organized as follows. In Section 2, we briefly introduce related methods and then present our SCAT method in Section 3. We demonstrate the efficacy of our proposed SCAT method through extensive experiments on real datasets in Section 4. Finally, we conclude the paper in Section 5.

2. Related work

In this section, we give a brief introduction on SSC [5], LSR [8] and the state-of-the-art Sparse Graph with Laplacian Smoothness (SGLS) [32]. For convenience, we use the following notation throughout the paper.

Notations: Throughout this paper, we denote matrices as uppercase letters. The i th row and j th column element of a matrix $X \in R^{m \times n}$ and its trace are denoted as x_{ij} and $Tr(X)$, respectively. X^T and X^{-1} are the transpose and the inverse of X , respectively. $\|X\|_F$ stands for the Frobenius norm of X . The l_2 -norm of a vector v is denoted by $\|v\|_2$. Moreover, $\langle X, Y \rangle$ is the inner product of two matrices with identical size, which is equal to the trace of $X^T Y$. $\mathbf{1}$ denotes a column vector of ones, and x^i and x_j denote the i th row and the j th column of X , respectively. The $l_{2,1}$ -norm of X is denoted by $\|X\|_{2,1} = \sum_{i=1}^m \|x^i\|_2$.

2.1. Problem formulation

Assume that

$$X = [x_1, x_2, \dots, x_n] \in R^{m \times n}$$

is a collection of n data points drawn from a union of c linear subspaces

$$S_1 \cup S_2 \cup \dots \cup S_c.$$

Given X , we cluster the data X according to their subspaces.

2.2. Sparse subspace clustering

Motivated by Sparse Representation [33], SSC is proposed to cluster data points that lie in a union of low-dimensional subspaces by exploiting the self-expressiveness property of data. The formulation of SSC is as follows:

$$\min_{Z, E} \|Z\|_0 + \lambda \|E\|_F, \quad \text{s.t. } X = XZ + E, \quad \text{diag}(Z) = 0, \quad (1)$$

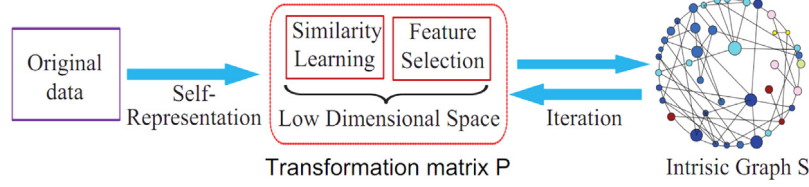


Fig. 1. The flowchart of the proposed SCAT method.

where $\lambda > 0$ is a parameter to balance the two parts, E denotes the error components and $Z = [z_1, z_2, \dots, z_n] \in R^{n \times n}$ is the self-expression coefficient matrix. Due to the property of l_0 -norm, each column z_i is the sparse representation vector corresponding to x_i . However, the l_0 -norm optimization problem is non-convex and NP-hard. Therefore, [5] solves the following graph learning model:

$$\min_{Z, E} \|Z\|_1 + \lambda \|E\|_F, \quad \text{s.t. } X = XZ + E, \quad \text{diag}(Z) = 0, \quad (2)$$

where $\|Z\|_1 = \sum_{i,j} |z_{ij}|$. Each element z_{ij} of the learned Z reflects the similarity between data pair x_i and x_j , hence Z can be viewed as an affinity matrix. Specifically, an affinity matrix can be defined as $|Z| + |Z^T|$. Finally, clustering data is performed by spectral clustering methods [4]. Note that if there are two data points having the same similarity degree as the represented data point, SSC only randomly chooses one of them for representation while ignoring another one. Moreover, the learned Z could lose a lot of global information of data if it is too sparse. As a result, SSC may not learn a high quality graph for clustering.

2.3. Least squares regression for subspace clustering

The least-squares regression (LSR) method is used to model highly correlated data by minimizing the Frobenius norm. [8] shows that LSR encourages a grouping effect, which tends to group highly correlated data together. The objective is defined as:

$$\min_{Z, E} \|Z\|_F + \frac{\lambda}{2} \|E\|_F^2, \quad \text{s.t. } X = XZ + E, \quad \text{diag}(Z) = 0, \quad (3)$$

where the Frobenius norm-based regularization may lead to a too dense coefficient matrix to be an efficient representation [34,35], which affects the clustering results. In a word, LSR captures the global representation structure of data while losing the local information of data.

2.4. Sparse graph with Laplacian smoothness

Some studies [36,37] show that exploiting the manifold structure of data is very useful for improving the performance of learning tasks. In graph embedding [38], if two data points x_i and x_j are close to each other (i.e., they are similar to each other), then the corresponding embedding in a new space should be close to each other. In mathematics, this closeness preserving can be quantified by:

$$\min_Z \sum_{i,j} \|z_i - z_j\|_2^2 w_{ij} = \frac{1}{2} \text{Tr}(Z^T L Z), \quad (4)$$

where $L = D - W$ represents Laplacian matrix, w_{ij} is the i th row and j th column entry of a pre-computed affinity matrix W , and D is a diagonal matrix with $D_{ii} = \sum_j w_{ij}$. Each w_{ij} of W is usually calculated by using a Gaussian kernel function defined as follows:

$$w_{ij} = \begin{cases} \exp(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}), & x_i \in \mathcal{N}_k(x_j) \text{ or } x_j \in \mathcal{N}_k(x_i); \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where $\mathcal{N}_k(x_i)$ denotes the set of k nearest neighbors of x_i and σ is the kernel width.

Motivated by graph embedding, [32] proposes a Sparse Graph with Laplacian Smoothness (SGLS) approach to estimate the unknown coefficient matrix Z by simultaneously integrating sparse representation and manifold regularization as follows:

$$\min_{Z, E} \|Z\|_1 + \lambda_1 \text{Tr}(Z^T L Z) + \lambda_2 \|E\|_F, \quad (6)$$

$$\text{s.t. } X = XZ + E, \quad \text{diag}(Z) = 0. \quad (7)$$

The learned Z satisfies data self-representation, graph sparsity, and Laplacian smoothness. However, SGLS still only aims at capturing the local structure of data while ignoring the global structure. Moreover, Z may be misled by L since two points lying in different subspaces could be very close to each other, and vice versa [5,39].

3. Methodology

Due to sparsity of Z , SSC may fail to reveal the important correlation structure in the data [40]. Although LSR takes advantage of data correlation, the learned Z might be dense, which degrades the clustering performance [35]. The clustering performance of SGLS is largely determined by the pre-computed affinity matrix. In addition, there are other methods directly learning a graph on the original data, such as CLR. Unfortunately, the learned graph cannot be guaranteed in various practical applications. To remedy the shortcomings of the graph-based clustering methods mentioned earlier, we first formulate the objective function of the proposed approach SCAT. Specifically, our method iteratively updates the affinity matrix and the transformation matrix, such that we can learn the intrinsic graph with the help of a low-dimensional space. Then an optimization algorithm for solving SCAT will be presented.

3.1. Model of SCAT

Given a set of clean data vectors $X = [x_1, \dots, x_n] \in R^{d \times n}$ drawn from a union of c subspaces $\{S_i\}$, $i = 1, \dots, c$. Let X_i be a collection of n_i data vectors drawn from the subspace S_i , $n = \sum_{i=1}^c n_i$. By the representation assumption [5], the data can be self-represented by a linear combination of some data vectors in X as $X = XZ$. However, there are many possible representations. To capture the global information from data, we consider the Frobenius norm of the representative matrix, i.e., $\|Z\|_F$, which encourages the learned Z to capture the correlation structure of data from the same subspace [8]. On the other hand, [41] suggests that non-negative representation often leads to better performance for various learning tasks. Then we arrive at the following objective function:

$$\min_Z \|Z\|_F^2, \quad (8)$$

$$\text{s.t. } X = XZ, Z \geq 0, \text{diag}(Z) = \mathbf{0},$$

where $Z \in R^{n \times n}$ is a coefficient matrix, and each data point x_i is represented as a linear combination of all other data points

$x_i = \sum_{j \neq i} z_{ji} x_j$. Specifically, z_{ji} can directly measure the similarity between data points x_i and x_j since $z_{ji} > 0$.

The development of manifold learning theory [42] has demonstrated the importance of the local geometric structure of data for learning tasks [38,43,44]. Instead of using a pre-computed Laplacian matrix to preserve locality, we add an adaptive distance regularization into the objective (8) to capture the local structure of data due to the nonnegativeness of Z . Specifically, if x_i and x_j are close to each other in the original space, then z_{ji} should have large probability and vice versa. Thus, we have the following objective function:

$$\min_Z \sum_{i,j} \|x_i - x_j\|_2^2 z_{ij} + \frac{\lambda_1}{2} \|Z\|_F^2, \quad (9)$$

s.t. $X = XZ, Z \geq 0, \text{diag}(Z) = \mathbf{0}, Z^T \mathbf{1} = \mathbf{1},$

where $\lambda_1 > 0$ is a penalty parameter, and $\mathbf{1}$ represents an all-ones vector. Sometimes data points lie in a union of affine subspaces rather than linear subspaces, thus we add the affine constraint $Z^T \mathbf{1} = \mathbf{1}$ [5] to deal with affine subspaces.

In real world applications, fully noise-free is impossible. We extend the problem to be robust to noises by introducing a noise term E as follows:

$$\min_{Z,E} \sum_{i,j} \|x_i - x_j\|_2^2 z_{ij} + \frac{\lambda_1}{2} \|Z\|_F^2 + \lambda_2 \Theta(E), \quad (10)$$

s.t. $X = XZ + E, Z \geq 0, \text{diag}(Z) = \mathbf{0}, Z^T \mathbf{1} = \mathbf{1},$

where $\lambda_2 > 0$ is a penalty parameter. $\Theta(E)$ is the model of E , which can model the noise depending on the characteristic of data. The choice of the model only depends on which kind of error is assumed in the dataset. For instance $\|E\|_F$ is optimal for Gaussian residuals while $\|E\|_1$ for Laplacian. Considering the simplicity we simply adopt $\|E\|_F$.

Note that the distance regularization term in the objective (10) may misguide Z due to the existence of irrelevant, redundant and noisy dimensions in the original data. To resolve this issue, we propose to learn the affinity matrix from a low-dimensional space of the original data by simultaneously conducting feature selection and similarity learning. The objective function (10) can be rewritten as:

$$\min_{Z,E,P} \sum_{i,j} \|P^T x_i - P^T x_j\|_2^2 z_{ij} + \frac{\lambda_1}{2} \|Z\|_F^2 + \frac{\lambda_2}{2} \|E\|_F^2 + \lambda_3 \|P\|_{2,1}, \quad (11)$$

s.t. $X = XZ + E, Z \geq 0, \text{diag}(Z) = \mathbf{0}, Z^T \mathbf{1} = \mathbf{1}, P^T P = I,$

where $\lambda_3 > 0$ is the regularization parameter. We adopt $l_{2,1}$ -norm constraint on P to make it row sparse, hence it is suitable for selecting valuable features [45]. The orthogonal constraint on P make the selected feature distinctive. The objective (11) integrates feature selection and similarity learning into a unified framework such that P can be iteratively updated while fixing Z , vice versa. As the assumption that the high-dimensional data reside on a low-dimensional intrinsic space [42], more discriminative features should be captured if data points are transformed into a proper subspace. Hence learning Z benefits from the low-dimensional space via the transformation matrix P . It is worth noting that the first regularization term, i.e., $\sum_{i,j} \|P^T x_i - P^T x_j\|_2^2 z_{ij}$ can be considered as the weighted sparse regularization since Z is nonnegative, which can simultaneously guarantee the learned Z captures the locality, and has sparsity. Thus we can learn a high-quality similarity graph.

3.2. Optimization algorithm

In our objective function (11), there are three variables to be optimized, and it is not convex for Z, E , and P , but we can still solve this problem by breaking it down into several convex subproblems. In particular, these variables can be optimized alternately. Note that the problem (11) is not separable for optimizing, so we introduce two auxiliary variables $Z = S$ and $Z = U$ as follows:

$$\min_{Z,E,P,S,U} \sum_{i,j} \|P^T x_i - P^T x_j\|_2^2 s_{ij} + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|E\|_F^2 + \lambda_3 \|P\|_{2,1}, \quad (12)$$

s.t. $X = XZ + E, Z = U, Z = S, S \geq 0,$
 $\text{diag}(S) = \mathbf{0}, S^T \mathbf{1} = \mathbf{1}, P^T P = I.$

The augmented Lagrangian function of the problem (12) can be rewritten into the following equivalent problem:

$$\min_{Z,E,F,S,U} \sum_{i,j} \|P^T x_i - P^T x_j\|_2^2 s_{ij} + \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\lambda_2}{2} \|E\|_F^2 + \lambda_3 \|P\|_{2,1} + \langle \Lambda_1, X - XZ - E \rangle + \langle \Lambda_2, Z - U \rangle + \langle \Lambda_3, Z - S \rangle + \frac{\mu}{2} (\|X - XZ - E\|_F^2 + \|Z - U\|_F^2 + \|Z - S\|_F^2), \quad (13)$$

where Λ_1, Λ_2 and Λ_3 are Lagrangian multipliers, and μ is a regularity coefficient to control the penalty for the three violation of equality constraints in problem (12). Now we alternately solve each variable of the formidable problem (13) while fixing the other variables. As we will see, the problem (13) is reduced to five manageable subproblems. The detail solution steps are as follows.

Update Z

Fix E, P, S and U except Z , and update Z by minimizing the following problem:

$$\mathcal{L}(Z) = \|X - XZ - E\|_F^2 + \frac{\Lambda_1}{\mu} \|X - XZ - E\|_F^2 + \|Z - U\|_F^2 + \frac{\Lambda_2}{\mu} \|Z - U\|_F^2 + \|Z - S\|_F^2 + \frac{\Lambda_3}{\mu} \|Z - S\|_F^2. \quad (14)$$

For computing Z , we take derivative of $\mathcal{L}(Z)$ with respect to Z and set it to zero, then obtain Z as follows:

$$Z = (X^T X + 2I)^{-1} (X^T Q_1 + Q_2 + Q_3), \quad (15)$$

where $Q_1 = X - E + \frac{\Lambda_1}{\mu}$, $Q_2 = U - \frac{\Lambda_2}{\mu}$ and $Q_3 = S - \frac{\Lambda_3}{\mu}$.

Update S

When Z, E, P and U are fixed, the problem turns into the following objective function:

$$\min_S \sum_{i,j} \|x_i - x_j\|_2^2 s_{ij} + \frac{\mu}{2} \|Z - S\|_F^2 + \frac{\Lambda_3}{\mu} \|Z - S\|_F^2, \quad (16)$$

s.t. $S \geq 0, \text{diag}(S) = \mathbf{0}, S^T \mathbf{1} = \mathbf{1}.$

Note that the objective (16) is independent between different j , thus we can tackle the following objective function individually for every j :

$$\min_{s_j} \sum_{i=1}^n (\|x_i - x_j\|_2^2 s_{ij} + \frac{\mu}{2} s_{ij}^2 - \mu s_{ij} h_{ij}), \quad (17)$$

s.t. $s_j \geq 0, s_{jj} = 0, s_j^T \mathbf{1} = 1,$

where h_{ij} is the (i, j) th element of $H = Z + \frac{\Lambda_3}{\mu}$. Let

$$d_{ij} = \|x_i - x_j\|_2^2 - \mu h_{ij}$$

be the i th element of $d_j \in R^{n \times 1}$. With a little algebra, the above problem arrive at:

$$\min_{s_j \geq 0, s_{ij}=0, s_j^T \mathbf{1}=1} \|s_j + \frac{d_j}{\mu}\|_2^2. \quad (18)$$

The Lagrangian function of the problem (18) is

$$\mathcal{L}(s_j, \eta, \beta) = \|s_j + \frac{d_j}{\mu}\|_2^2 + \eta(1 - s_j^T \mathbf{1}) + \beta^T(-s_j), \quad (19)$$

where η and $\beta \geq 0$ are the Lagrangian multipliers. From the Karush-Kuhn-Tucker condition [46], it is easy to verify that the optimal solution s_j is $(-\frac{1}{\mu}d_j + \eta\mathbf{1})_+$.

Update P

With fixed Z, S, U and E , the problem is transformed into the following:

$$\min_{P^T P=I} \text{Tr}(P^T X L_S X^T P) + \lambda_3 \|P\|_{2,1}, \quad (20)$$

where $L_S = D - \frac{S^T + S}{2}$ is a Laplacian matrix, and the diagonal matrix D is its degree matrix [47], i.e., the i th entry on the diagonal of D is $\sum_{j=1}^n \frac{s_{ij} + s_{ji}}{2}$.

By [45], we have the following:

$$\min_{P^T P=I} \text{Tr}(P^T X L_S X^T P) + \lambda_3 \text{Tr}(P^T Q P), \quad (21)$$

where Q is a diagonal matrix, the i th entry on the diagonal of Q is defined as $Q_{ii} = \frac{1}{2\sqrt{p_i^T p_i + \epsilon}}$, and ϵ is a small enough constant. We rewrite the problem (21) as:

$$\min_{P^T P=I} \text{Tr}(P^T (X L_S X^T + \lambda_3 Q) P). \quad (22)$$

In the light of the Algorithm 1 in [45], we can solve this problem iteratively, i.e., (1) With fixed Q , the optimal solution $P \in R^{d \times m}$ for problem (22) is formed by the m eigenvectors of $(X L_S X^T + \lambda_3 Q)$ with respect to the m smallest eigenvalues; (2) With fixed P , Q is obtained by $Q_{ii} = \frac{1}{2\sqrt{p_i^T p_i + \epsilon}}$.

Update U

Optimizing the problem (13) with respect to U becomes the following formula:

$$\mathcal{L}(U) = \frac{\lambda_1}{2} \|U\|_F^2 + \frac{\mu}{2} \|Z - U + \frac{\Lambda_2}{\mu}\|_F^2. \quad (23)$$

We take derivative of $\mathcal{L}(U)$ with respect to U and set it to zero, then obtain:

$$U = \frac{\Lambda_2 + \mu Z}{\lambda_1 + \mu}. \quad (24)$$

Update E

Similar to U , we fix other variables except E , we obtain the following formula:

$$\mathcal{L}(E) = \frac{\lambda_2}{2} \|E\|_F^2 + \frac{\mu}{2} \|X - XZ - E + \frac{\Lambda_1}{\mu}\|_F^2. \quad (25)$$

By setting the derivative of $\mathcal{L}(E)$ with respect to E to zero, it is obvious that

$$E = \frac{\Lambda_1 + \mu(X - XZ)}{\lambda_2 + \mu}. \quad (26)$$

Update $\Lambda_1, \Lambda_2, \Lambda_3$ and μ

From [48], we update $\Lambda_1, \Lambda_2, \Lambda_3$ and μ as follows:

$$\Lambda_1 = \Lambda_1 + \mu(X - XZ - E), \quad (27)$$

$$\Lambda_2 = \Lambda_2 + \mu(Z - U), \quad (28)$$

$$\Lambda_3 = \Lambda_3 + \mu(Z - S), \quad (29)$$

$$\mu = \min(\rho\mu, \mu_{\max}), \quad (30)$$

where ρ and $\mu_{\max}$ are pre-fixed constants, in which ρ controls the convergence speed. The proposed optimization approach is summarized in Algorithm 1.

Algorithm 1 Framework for solving Eq. (13)

Input: Data X ; parameters λ_1, λ_2 , and λ_3 ; the projection dimensionality m .

Output: Z

- 1: Constructing the k nearest neighbor graph as the initial matrix of Z ;
- 2: **repeat**
- 3: Update Z with Eq. (15);
- 4: Update S by solving Eq. (18);
- 5: Update P by solving problem (20);
- 6: Update U with Eq. (24);
- 7: Update E with Eq. (26);
- 8: Update $\Lambda_1, \Lambda_2, \Lambda_3$ and μ .
- 9: **until** Converge

Similar to the LSR, SCC, and SGLS methods, once we obtain the coefficient matrix Z , we feed $Z + Z^T$ into the framework of spectral clustering to segment the data. The proposed SCAT method is summarized in Algorithm 2.

Algorithm 2 SCAT for Data Clustering

Input: Data X ; parameters λ_1, λ_2 , and λ_3 ; the projection dimensionality m ; the number of clusters.

Output: Cluster labels for all data points;

- 1: Apply Algorithm 1 to find the coefficient matrix Z ;
- 2: Normalize the columns of Z as $z_i \leftarrow (\frac{z_i}{\|z_i\|_\infty})$;
- 3: Form a similarity graph with n nodes and set the weights on the edges between nodes by $Z + Z^T$;
- 4: Apply spectral clustering to the affinity matrix $Z + Z^T$.

3.3. Computational complexity analysis

We will provide empirical evidence of the convergence of this algorithm in the experimental part. The mainly computational costs of Algorithm 1 are the inverse operation and eigen-decomposition in steps 1 and 3, respectively. Since only partial eigen-decomposition is used in step 3, it takes $O(ndm)$ time to compute m approximate eigenvectors. We can pre-compute the inverse of $(X^T X + 2I)$ taking $n^{2.373}$ operations [49], and use it at each iteration. Step 2 takes $O(n^2)$. Therefore, the total computational complexity of the algorithm is $O(t(nmd + n^2))$, where t is the number of iterations.

4. Experimental results

In this section, we conducted experiments on real-world datasets to evaluate the performance of the proposed SCAT. Following [43] and [50], we use clustering accuracy (ACC) and normalized mutual information (NMI) as evaluation measurements.

4.1. Evaluation metrics

Assume that we have N samples, then ACC can be defined as follows:

$$\text{ACC} = \frac{\sum_{i=1}^N f(o_i, \text{map}(r_i))}{N},$$

where $f(\cdot, \cdot)$ is a function of two variables satisfying $f(x, y) = 1$ if $x = y$ and $f(x, y) = 0$ otherwise, $\text{map}(r_i)$ is the permutation

Table 1
Statistics of the datasets.

Dataset	#dimensionality	#instances	#clusters
JAFFE	1024	213	10
AR	768	840	120
Yale	1024	165	15
Dermatology	34	366	6
Segment	19	2310	7
ORL	1024	400	40
Mpeg7	6000	1400	70
YaleB	1024	2414	38
Yeast	1470	1484	10

mapping function that maps each ground truth label r_i to the equivalent label from the dataset, and o_i and r_i are the cluster label and the ground truth label, respectively.

Generally, mutual information (MI) is used to measure how similar two sets of clusters are. Assume that C is the ground truth set of clusters for the original dataset, and C' is the set of clusters generated from the conducted algorithm. Then the mutual information of C and C' is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)},$$

where $p(c_i, c'_j)$ denotes the joint probability that this arbitrarily selected image belongs to the cluster c_i as well as c'_j simultaneously, and $p(c_i)$ and $p(c'_j)$ are the probabilities that a sample arbitrarily selected from the dataset belongs to the clusters c_i and c'_j , respectively. In the experiment, we use the following NMI:

$$NMI = \frac{MI(C, C')}{\max(H(C), H(C'))},$$

where $H(C)$ and $H(C')$ denote the entropies of C and C' , respectively. The range of NMI is in $[0, 1]$. The bigger the NMI is, the more similar C and C' are.

4.2. Datasets

We collect nine real-world benchmark datasets. JAFFE [51], AR [52], Yale [53] and ORL [54] are facial image datasets. Mpeg7 [55] is an object dataset, and other datasets are from UCI Machine Learning Repository [56]. All these datasets have been frequently used to evaluate the performance of different clustering methods. Their statistics are summarized in Table 1. Specifically, Fig. 2 shows some example images from JAFFE, AR, Yale, and Mpeg7 database, respectively.

4.3. Compared algorithms

In our experiments, we compare SCAT with the following seven algorithms:

- K -means performs clustering in the original feature space of data.
- NCut [4] is a classic spectral clustering method, which treats the grouping problem as a graph partitioning problem. In particular, a generalized eigenvalue system provides a real-valued solution to the graph partitioning problem.
- LSR [8] is a robust subspace clustering method. It tends to group highly correlated data together.
- RMKKM [57] is a robust multiple kernel K -means algorithm. It simultaneously finds the best clustering label, the cluster membership, and the optimal combination of multiple kernels.

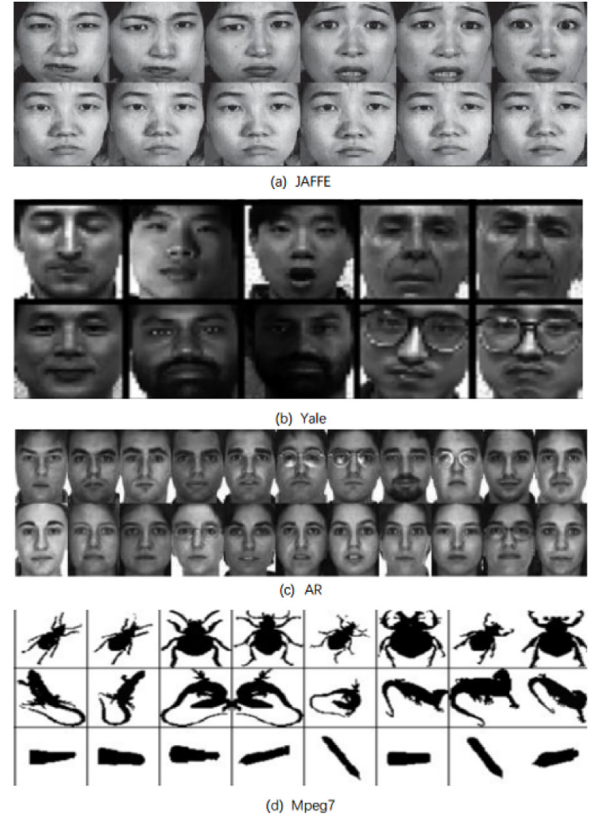


Fig. 2. Sample images of JAFFE, AR, YALE and Mpeg7.

- APMF [50] is a state-of-the-art matrix factorization method for data clustering. It inherits the merits of both spectral clustering and NMF [58].
- CLR [17] is a state-of-the-art one step spectral clustering method. It learns a graph with exact c connected components (where c is the number of clusters).
- SGLS [32] is a new sparse graph construction method that integrates manifold constraint on the unknown sparse codes as a graph regularizer.

4.4. Setup

K -means clustering is performed on the original features. For NCut, we build the adjacency graph by setting the number k of nearest neighbors from a candidate domain $\{3, 5, 7, 11, 13, 15\}$. The Gaussian kernel $K(x, y) = \exp(-\frac{\|x-y\|_2^2}{td_{max}^2})$ is adopted, where d_{max} is the maximal distance between data points and the parameter t is in the range $\{0.01, 0.1, 1, 10, 50, 100\}$. About other algorithms, we follow the setting of the released codes. Specifically, to obtain the best performance, we tune the regularization parameters in a wide range. The projection dimensionality m of $P \in R^{d \times m}$ is empirically set to the number of clusters. For LSR, SGLS, and SCAT, K -means clustering is conducted on spectral embedding to obtain the final clustering results. To reduce statistical variability resulted by K -means, we repeat K -means clustering 20 times and record the results with the best objective values. Specifically, to guarantee the same experimental settings between all the compared methods and our method on each benchmark, we fix the detailed parameters of K -means, e.g., initialization, distance measure, and the number of repeats. All the methods are performed for each dataset on the same testbed. Because different datasets are affected by noise to varying degrees, λ_2 is selected

Table 2
Clustering ACC (%) of different methods.

Dataset	K-means	NCut	LSR	RMKMM	CLR	SGLS	APMF	SCAT
JAFFE	74.16	80.89	98.17	87.22	96.71	100.00	92.23	100.00
AR	32.50	27.65	69.97	34.98	46.43	63.33	60.95	80.95
Yale	46.18	49.62	38.78	52.01	57.58	56.97	57.90	64.63
Dermatology	77.53	92.89	91.23	90.68	95.36	95.36	96.12	96.45
Segment	58.66	61.91	66.88	62.96	34.37	70.82	70.76	71.78
ORL	49.60	57.84	73.00	55.06	64.50	67.75	65.75	75.68
Mpeg7	47.64	50.65	47.57	49.36	47.71	52.79	52.71	56.43
YaleB	9.86	56.32	59.82	10.83	28.63	77.25	25.49	80.78
Yeast	30.99	34.56	42.39	31.41	31.60	35.78	45.33	52.56

Table 3
Clustering NMI (%) of different methods.

Dataset	K-means	NCut	LSR	RMKMM	CLR	SGLS	APMF	SCAT
JAFFE	82.37	85.89	97.06	89.22	96.24	100.00	90.12	100.00
AR	64.50	58.65	85.15	65.98	67.54	85.32	83.95	91.76
Yale	52.18	56.62	42.78	55.01	57.81	57.35	61.90	63.75
Dermatology	83.53	88.89	79.98	88.58	91.30	90.93	92.36	93.13
Segment	56.66	61.00	57.72	69.81	30.94	58.43	61.76	62.98
ORL	74.60	72.84	85.28	74.06	80.05	82.77	80.75	86.60
Mpeg7	67.05	70.25	64.45	67.45	62.16	70.41	70.79	73.52
YaleB	13.33	72.67	60.99	13.04	32.06	79.41	38.86	81.17
Yeast	0.59	21.38	26.11	20.53	7.10	6.36	23.11	32.15

from the range of $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. And the range of λ_1 and λ_3 is slightly wider than that of λ_2 . In addition, we run 20 times for each experiment and record their values to compute the p -value for a statistical significance test. The significance level is typically set to 0.05, which means that if the significance evaluation is lower than this level, the performance difference between the evaluated methods is statistically significant [59].

4.5. Clustering results

Tables 2 and 3 show the clustering results in terms of ACC and NMI on all datasets, respectively. A higher value means a better clustering quality. The best results are highlighted in boldface. As we can see, the proposed SCAT performs consistently better than other methods. Besides, we have other observations listed in the following.

(1) For AR, ORL, Mpeg7, YaleB, and Yeast datasets, which have high dimension of features, the proposed method outperforms the compared algorithms a lot in terms of ACC and NMI. For example, from Tables 2 and 3, we can see that our SCAT achieves nearly 7% higher than those of compared methods in terms of ACC and NMI on the Yeast dataset. This indicates that the proposed method is beneficial from datasets that have more features since those features provide much information such that our method can capture the intrinsic structure of data. Hence our method improves the clustering performance. In Table 2, we can see that SCAT achieves more than 10% scores of ACC in comparison with the second-best method LSR on the AR dataset. Although AR is a noisy dataset, our method simultaneously conducts feature selection and similarity learning such that the effect of noise and redundancy on clustering performance can be alleviated. While on the remaining two non-high dimensional datasets, i.e., Dermatology and Segment datasets, our method still achieves comparative clustering ACC with other methods. In Table 3, RMKMM outperforms our method in terms of NMI on the Segment dataset. The reason is that the Segment dataset has very low dimension, and redundancy of its features may also be low, thus performing feature selection is easy to cause information loss and degrade clustering performance.

(2) K -Means is probably the most well-known clustering method in the communities of data mining and pattern recognition, and many algorithms have to invoke it to output the final

clustering results, such as NCut. However, Tables 2 and 3 show that K -means cannot achieve satisfactory results in most cases. K -means directly run on the original data space, which usually contains many redundant features, and it does not exploit the global structure of data. By contrast, NCut shows better performance than K -means since it exploits the manifold structure of data. Benefiting from $l_{2,1}$ -norm and multiple kernel learning, RMKMM consistently outperforms K -means even NCut in most cases. Consequently, it is not a right choice to conduct clustering by directly using K -means. Furthermore, in order to improve clustering performance, we must consider more factors, such as the redundancy of features in data.

(3) CLR is a state-of-the-art, adaptive graph learning method for data clustering. From Tables 2 and 3, we can find that CLR obtains better performance than K -means, NCut, and RMKMM on most datasets. CLR conducts clustering on a graph that was learned in advance. This graph was learned on the original data, and it only considers the local structure of data. As a result, the performance of CLR is sensitive to noise and redundancy. For example, CLR achieves unsatisfactory clustering performance on AR, YaleB, and Yeast datasets. For APMF, it integrates adaptive graph learning and matrix factorization into a framework. Specifically, it not only seeks low dimension representations of data but also simultaneously conducts clustering on the low dimensional representations. By doing this, the influence of noise and redundancy on clustering performance can be alleviated. Consequently, APMF usually outperforms CLR in most cases. We also note that SCAT achieves more than 20% ACC score in comparison with CLR and APMF on the AR dataset. On the YaleB dataset, our method achieves more than 40% NMI score in comparison with CLR and APMF. From these observations, we can know that, for improving clustering performance, it is not enough to only consider the local structure of data or the low-dimensional representation of data.

(4) The subspace clustering methods LSR and SGLS show good performance on some datasets, but LSR only captures the global structure of data, while SGLS only captures the local structure of data. LSR lags behind NCut on the Yale, Dermatology, Mpeg7 datasets in terms of ACC and NMI. SGLS, APMF, and CLR are comparable since all of them consider the local structure of data. However, they ignore the global structure of data such that the learned graphs are not optimal for clustering. LSR outperforms SGLS on the AR, ORL, and Yeast datasets in terms of

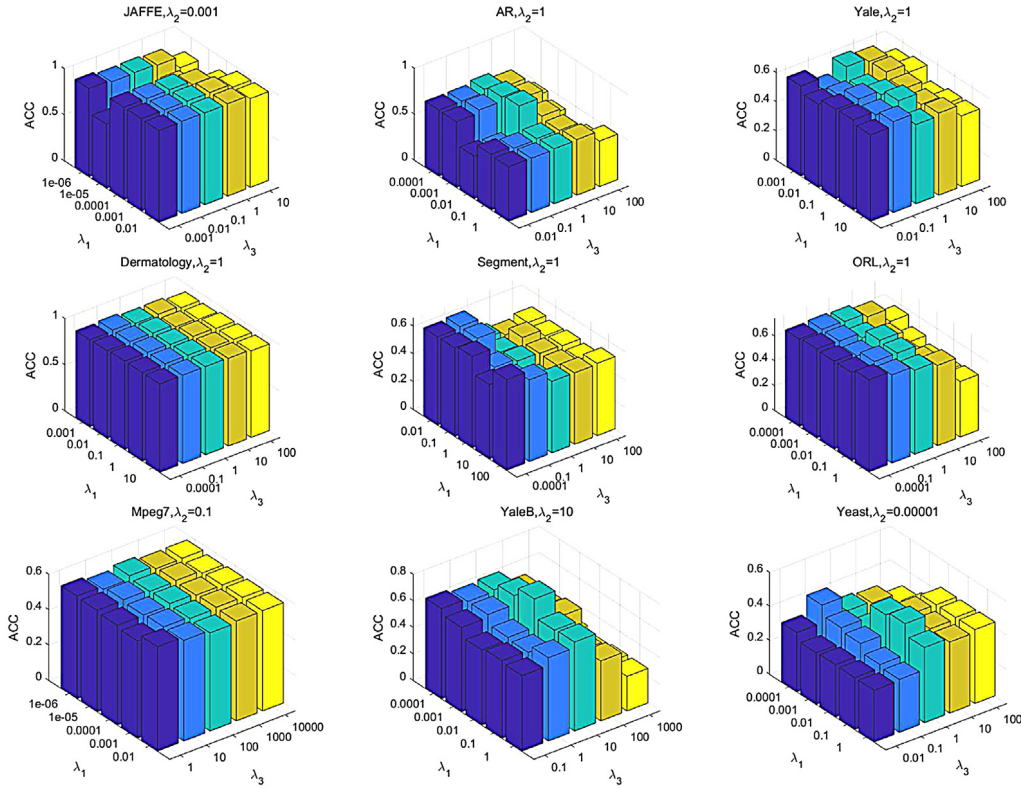


Fig. 3. Clustering ACC versus different values of parameters λ_1 and λ_3 on nine benchmark datasets.

Table 4

p -values of clustering ACC between SCAT and other methods on the nine datasets.

Dataset	K-means	NCut	LSR	RMKMM	CLR	SGLS	APMF
JAFFE	1.5510e-11	1.1502e-20	7.5279e-04	2.3057e-20	1.1028e-32	1.7039e-10	7.2637e-19
AR	7.6429e-30	7.1433e-24	3.7926e-22	1.2472e-30	1.2603e-34	1.4637e-33	8.9501e-24
Yale	1.4347e-08	4.8190e-04	7.0697e-18	1.5057e-07	0.0311	2.9606e-08	0.0077
Dermatology	8.3735e-08	3.4472e-25	6.2418e-31	8.8352e-14	1.1037e-14	2.6466e-05	3.2361e-06
Segment	8.4510e-10	2.9309e-19	8.4927e-16	2.1710e-07	3.5523e-43	0.00113	0.0141
ORL	6.0198e-13	4.7647e-33	0.0039	4.8712e-16	5.2592e-04	2.9863e-18	7.8058e-07
Mpeg7	3.0333e-14	3.7478e-35	1.5943e-18	3.4430e-05	3.6020e-24	4.9096e-20	7.8046e-14
YaleB	1.8546e-34	1.7715e-35	5.7569e-19	6.0629e-32	2.5334e-32	1.4571e-25	7.8057e-31
Yeast	5.2050e-16	1.0715e-28	2.6105e-30	4.9087e-24	1.4730e-49	8.4314e-34	2.3626e-14

ACC and NMI. For JAFFE, Yale, Dermatology, Segment, Mpeg7, and YaleB datasets, SGLS outperforms LSR in terms of ACC and NMI. These observations show that considering the global and local structures of data can bring improvement of clustering performance. Consequently, the proposed SCAT simultaneously exploits local distance information and global representation of data. Specifically, during the process of optimization, the adaptive transformation matrix eliminates the adverse impact of noisy or redundant features, such that the distance regularization can help the affinity matrix capture the true local structure of data. As a result, the SCAT method achieves the best performance in all the compared algorithms.

To demonstrate the statistical significance of our method in comparison with the compared methods, we conducted a statistical significance test, as shown in Table 4. Accurately, Table 4 records p -values of clustering ACC between SCAT and the other algorithms on the nine datasets, respectively. Following [59,60], we set the significance level to 0.05, i.e., when the estimated p -value is lower than 0.05, the performance difference between the two compared algorithms is statistically significant. From Table 4, we observe that all p -values are less than 0.05. This phenomenon indicates that the clustering performance difference between our

method and the other compared methods are statistically significant in all circumstances. Moreover, the effectiveness of our method is further verified.

4.6. Parameter sensitivity

From the objective function (11), there are three parameters in the proposed method, i.e., λ_1 , λ_2 and λ_3 . So far as we know, it is still an open problem to adaptively choose these optimal parameters for different datasets [60]. Hence we have to set the values of λ_1 , λ_2 and λ_3 in advance. Usually, the greater the parameter value is, the greater the importance or influence of the corresponding item is. The parameter λ_2 balances the importance of error term and can usually be taken in a small range [5]. We mainly investigate the effects of parameters λ_1 and λ_3 for data clustering when parameter λ_2 is fixed. Parameters λ_1 balances the importance of the global structure of data, while parameters λ_3 balances the importance of the feature selection term. The candidate parameter range set for λ_1 and λ_3 is $\{10^{-4}, 10^{-3}, \dots, 10^4\}$ and then the proposed method is conducted with different combinations of parameters for data clustering. With λ_2 fixed, Fig. 3 shows that clustering ACC versus different parameter values of

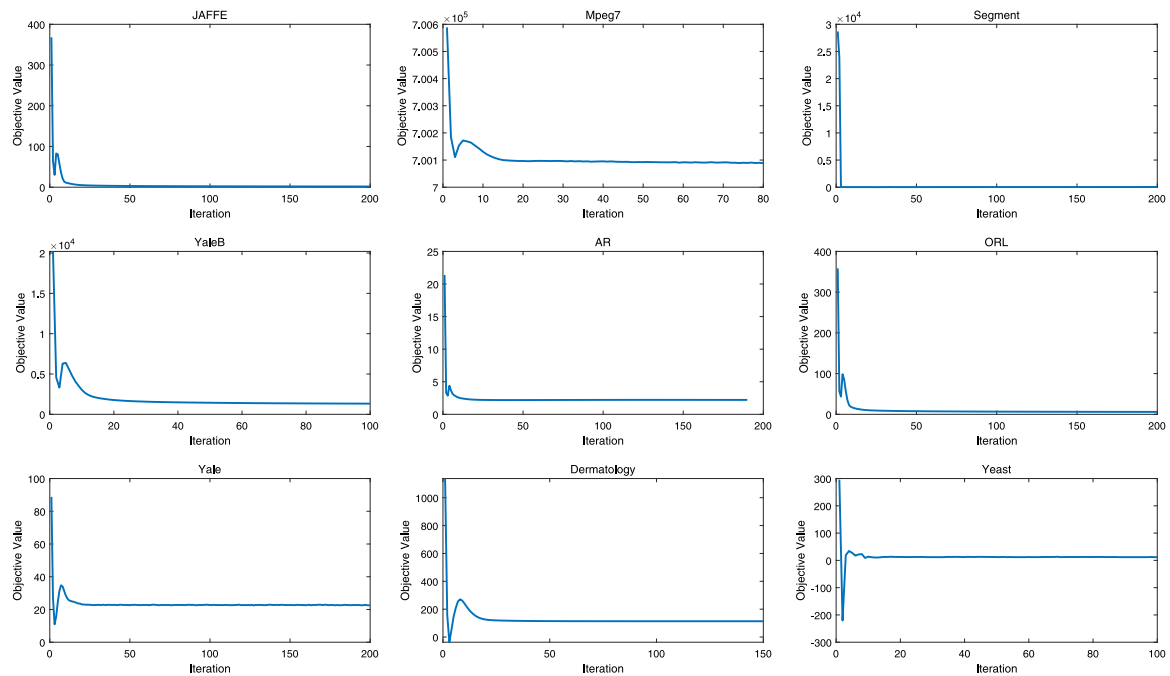


Fig. 4. The convergence curve of our proposed algorithm on eight datasets. Note that in early iterations, variables are not feasible.

λ_1 and λ_3 . According to Fig. 3, we can know that the clustering ACC is sensitive to the regularization parameters. In particular, there always is a feasible range for parameters λ_1 and λ_3 such that the best clustering result can be achieved. Different datasets should have different combinations of parameters since the global structure, and the redundancy of data are normally different for different datasets. A too-large parameter λ_1 makes $\|Z\|_F$ play the dominant role in the graph learning while ignoring the local structure of data, and a too-small parameter λ_3 results in the low level of the sparsity of P while cannot remove noise features from the original features. For example, the YaleB dataset achieves the best result when $\lambda_1 = 0.1$ and $\lambda_3 = 10$, while the AR dataset cannot achieve the best result for these parameter values. The high-dimensional dataset Mpeg7 requires a relatively large λ_3 , while the low-dimensional dataset Segment requires a relatively small λ_3 . The above results show that the proposed SCAT is data-driven.

4.7. Convergence

From theoretical results in [61], the following are sufficient for SCAT to convergence: (1) The parameter μ in the last step is needed to be upper bounded. (2) The dictionary X is of full column rank. (3) The optimization error of each iteration is monotonically decreasing. In SCAT, (1) can be guaranteed by the last step of Algorithm 1. Liu et al. [61] showed that X could be replaced by the orthogonal basis of X , so (2) is satisfied. (3) can be proved empirically from the aspect of experiments. Specifically, Fig. 3 shows the relationships of the objective function value of SCAT and the iteration step, i.e., the curves of the cost (i.e., objective function) vs. iterations on nine datasets, respectively. It is clear that SCAT satisfies (3), and the optimization algorithm converges at a stable value (see Fig. 4).

5. Conclusions

This paper proposes a novel clustering method via simultaneously conducting feature selection and similarity learning. Specifically, we integrate the learning of the affinity matrix and

the projection matrix into a framework to iteratively update them, so that a good graph can be obtained. Extensive experimental results on nine real datasets show that the proposed SCAT method consistently outperforms its competitors on the clustering task in terms of ACC and NMI, respectively. In the next step of our research, we may extend our method to multi-task clustering, which is another interesting research direction.

CRediT authorship contribution statement

Guo Zhong: Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Chi-Man Pun:** Conceptualization, Funding acquisition, Investigation, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - review & editing.

Acknowledgment

This work was partly supported by the University of Macau under Grants: MYRG2018-00035-FST and MYRG2019-00086-FST, and the Science and Technology Development Fund, Macau SAR (File no. 041/2017/A1, 0019/2019/A).

References

- [1] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, Vol. 27, ACM, 1998.
- [2] C. Tang, X. Liu, M. Li, P. Wang, J. Chen, L. Wang, W. Li, Robust unsupervised feature selection via dual self-representation and manifold regularization, *Knowl.-Based Syst.* 145 (2018) 109–120.
- [3] R. Li, X. Yang, X. Qin, W. Zhu, Local gap density for clustering high-dimensional data with varying densities, *Knowl.-Based Syst.* (2019) 104905.
- [4] Jianbo Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.* (ISSN: 0162-8828) 22 (8) (2000) 888–905, <http://dx.doi.org/10.1109/34.868688>.
- [5] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.

- [6] Z. Kang, C. Peng, Q. Cheng, Twin learning for similarity and clustering: A unified kernel approach, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
- [7] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2012) 171–184.
- [8] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: European Conference on Computer Vision, Springer, 2012, pp. 347–360.
- [9] X. Fang, Y. Xu, X. Li, Z. Lai, W.K. Wong, Robust semi-supervised subspace clustering via non-negative low-rank representation, *IEEE Trans. Cybern.* 46 (8) (2015) 1828–1838.
- [10] J. Zhang, C.-G. Li, H. Zhang, J. Guo, Low-rank and structured sparse subspace clustering, in: 2016 Visual Communications and Image Processing (VCIP), IEEE, 2016, pp. 1–4.
- [11] J. Wang, X. Wang, F. Tian, C.H. Liu, H. Yu, Constrained low-rank representation for robust subspace clustering, *IEEE Trans. Cybern.* 47 (12) (2016) 4534–4546.
- [12] Y. Wang, Y.Y. Tang, L. Li, H. Chen, J. Pan, Atomic representation-based classification: theory, algorithm, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 41 (1) (2017) 6–19.
- [13] M. Yin, J. Gao, Z. Lin, Laplacian regularized low-rank representation and its applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (3) (2015) 504–517.
- [14] M. Yin, J. Gao, Z. Lin, Q. Shi, Y. Guo, Dual graph regularized latent low-rank representation for subspace clustering, *IEEE Trans. Image Process.* 24 (12) (2015) 4918–4933.
- [15] J. Liu, Y. Chen, J. Zhang, Z. Xu, Enhancing low-rank subspace clustering by manifold regularization, *IEEE Trans. Image Process.* 23 (9) (2014) 4022–4030.
- [16] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 977–986.
- [17] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained laplacian rank algorithm for graph-based clustering, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016.
- [18] Z. Kang, C. Peng, Q. Cheng, Clustering with adaptive manifold structure learning, in: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), IEEE, 2017, pp. 79–82.
- [19] Z. Li, F. Nie, X. Chang, L. Nie, H. Zhang, Y. Yang, Rank-constrained spectral clustering with flexible embedding, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (12) (2018) 6073–6082.
- [20] Z. Zhao, H. Liu, Spectral feature selection for supervised and unsupervised learning, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 1151–1157.
- [21] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection, in: Advances in Neural Information Processing Systems, 2006, pp. 507–514.
- [22] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, *Eng. Appl. Artif. Intell.* 32 (2014) 112–123.
- [23] S. Wang, J. Tang, H. Liu, Embedded unsupervised feature selection, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
- [24] D. Cai, C. Zhang, X. He, Unsupervised feature selection for multi-cluster data, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 333–342.
- [25] Y. Liu, K. Liu, C. Zhang, J. Wang, X. Wang, Unsupervised feature selection via diversity-induced self-representation, *Neurocomputing* 219 (2017) 350–363.
- [26] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, S. Zhang, Graph self-representation method for unsupervised feature selection, *Neurocomputing* 220 (2017) 130–137.
- [27] C. Hou, F. Nie, D. Yi, Y. Wu, Feature selection via joint embedding learning and sparse regression, in: Twenty-Second International Joint Conference on Artificial Intelligence, 2011.
- [28] F. Nie, D. Xu, L.W.-H. Tsang, C. Zhang, Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction, *IEEE Trans. Image Process.* 19 (7) (2010) 1921–1932.
- [29] R. Shang, W. Wang, R. Stolk, L. Jiao, Subspace learning-based graph regularized feature selection, *Knowl.-Based Syst.* 112 (2016) 152–165.
- [30] R. Vidal, Subspace clustering, *IEEE Signal Process. Mag.* 28 (2) (2011) 52–68.
- [31] P. Zhou, Z. Lin, C. Zhang, Integrated low-rank-based discriminative feature learning for recognition, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (5) (2015) 1080–1093.
- [32] F. Dornaika, L. Weng, Sparse graphs with smoothness constraints: Application to dimensionality reduction and semi-supervised classification, *Pattern Recognit.* 95 (2019) 285–295.
- [33] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (2) (2008) 210–227.
- [34] C. You, C.-G. Li, D.P. Robinson, R. Vidal, Oracle based active set algorithm for scalable elastic net subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3928–3937.
- [35] J. Yang, J. Liang, K. Wang, P. Rosin, M. Yang, Subspace clustering via good neighbors, *IEEE Trans. Pattern Anal. Mach. Intell.* (ISSN: 0162-8828) (2019) 1, <http://dx.doi.org/10.1109/TPAMI.2019.2913863>.
- [36] D. Cai, X. He, X. Wu, J. Han, Non-negative matrix factorization on manifold, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 63–72.
- [37] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 33 (8) (2010) 1548–1560.
- [38] P. Goyal, E. Ferrara, Graph embedding techniques, applications, and performance: A survey, *Knowl.-Based Syst.* 151 (2018) 78–94.
- [39] M. Yin, S. Xie, Z. Wu, Y. Zhang, J. Gao, Subspace clustering via learning an adaptive low-rank graph, *IEEE Trans. Image Process.* 27 (8) (2018) 3716–3728.
- [40] Z. Wu, M. Yin, Y. Zhou, X. Fang, S. Xie, Robust spectral subspace clustering based on least square regression, *Neural Process. Lett.* 48 (3) (2018) 1359–1372.
- [41] J. Xu, W. An, L. Zhang, D. Zhang, Sparse, collaborative, or nonnegative representation: Which helps pattern classification?, *Pattern Recognit.* 88 (2019) 679–688.
- [42] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Advances in Neural Information Processing Systems, 2002, pp. 585–591.
- [43] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, *IEEE Trans. Knowl. Data Eng.* 23 (6) (2010) 902–913.
- [44] M. Zheng, J. Bu, C. Chen, C. Wang, L. Zhang, G. Qiu, D. Cai, Graph regularized sparse coding for image representation, *IEEE Trans. Image Process.* 20 (5) (2010) 1327–1336.
- [45] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint ℓ_2 , ℓ_1 -norms minimization, in: Advances in Neural Information Processing Systems, 2010, pp. 1813–1821.
- [46] S. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
- [47] U. Von Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (4) (2007) 395–416.
- [48] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [49] F. Le Gall, Powers of tensors and fast matrix multiplication, in: Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ACM, 2014, pp. 296–303.
- [50] M. Chen, Q. Wang, X. Li, Adaptive projected matrix factorization method for data clustering, *Neurocomputing* 306 (2018) 182–188.
- [51] M.J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (12) (1999) 1357–1362.
- [52] A.M. Martinez, The AR Face Database, CVC Technical Report 24, 1998.
- [53] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using laplacianfaces, *IEEE Trans. Pattern Anal. Mach. Intell.* (3) (2005) 328–340.
- [54] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, IEEE, 1994, pp. 138–142.
- [55] X. Bai, X. Yang, L.J. Latecki, W. Liu, Z. Tu, Learning context-sensitive shape similarity by graph transduction, *IEEE Trans. Pattern Anal. Mach. Intell.* (ISSN: 0162-8828) 32 (5) (2010) 861–874, <http://dx.doi.org/10.1109/TPAMI.2009.85>.
- [56] K. Bache, M. Lichman, UCI Machine Learning Repository, University of California, School of Information and Computer Science, Irvine, CA, 2013, p. 28, <http://archive.ics.uci.edu/ml>.
- [57] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, Y.-D. Shen, Robust multiple kernel k-means using ℓ_{21} -norm, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.
- [58] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562.
- [59] Z. Zhang, Z. Lai, Y. Xu, L. Shao, J. Wu, G. Xie, Discriminative elastic-net regularized linear regression, *IEEE Trans. Image Process.* 26 (3) (2017) 1466–1481, <http://dx.doi.org/10.1109/TIP.2017.2651396>.
- [60] J. Wen, X. Fang, Y. Xu, C. Tian, L. Fei, Low-rank representation with adaptive graph regularization, *Neural Netw.* 108 (2018) 83–96.
- [61] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 171–184, <http://dx.doi.org/10.1109/TPAMI.2012.88>.