Pattern Analysis and Applications (2024) 27:119
https://doi.org/10.1007/s10044-024-01336-2

ORIGINAL ARTICLE

K-BEST subspace clustering: kernel-friendly block-diagonal embedded and similarity-preserving transformed subspace clustering

Jyoti Maggu[1] · Anurag Goel[2]

Received: 13 June 2024 / Accepted: 2 September 2024

Abstract

Subspace clustering methods, employing sparse and low-rank models, have demonstrated efficacy in clustering high-dimensional data. These approaches typically assume the separability of input data into distinct subspaces, a premise that does not hold true in general. Furthermore, prevalent low-rank and sparse methods relying on self-expression exhibit effectiveness primarily with linear structure data, facing limitations in processing datasets with intricate nonlinear structures. While kernel subspace clustering methods excel in handling nonlinear structures, they may compromise similarity information during the reconstruction of original data in kernel space. Additionally, these methods may fall short of attaining an affinity matrix with an optimal block-diagonal property. In response to these challenges, this paper introduces a novel subspace clustering approach named Similarity Preserving Kernel Block Diagonal Representation based Transformed Subspace Clustering (KBD-TSC). KBD-TSC contributes in three key aspects: (1) integration of a kernelized version of transform learning within a subspace clustering framework, introducing a block diagonal representation term to generate an affinity matrix with a block-diagonal structure. (2) Construction and integration of a similarity preserving regularizer into the model by minimizing the discrepancy between inner products of the original data and those of the reconstructed data in kernel space. This facilitates enhanced preservation of similarity information between the original data points. (3) Proposal of KBD-TSC by integrating the block diagonal representation term and similarity preserving regularizer into a kernel self-expressing model. The optimization of the proposed model is efficiently addressed through the alternating direction method of multipliers. This study validates the effectiveness of the proposed KBD-TSC method through experimental results obtained from nine datasets, showcasing its potential in addressing the limitations of existing subspace clustering techniques.

Keywords Block diagonal representation · Kernel representations · Non-linear subspace clustering · Similarity preserving · Transformed subspace clustering

1 Introduction

In recent research endeavors, considerable attention has been directed towards the development of subspace clustering methods, addressing the efficient processing of high-dimensional data. Subspace clustering serves as a valuable technique for grouping high-dimensional data distributed across a union of subspaces. The fundamental premise of subspace clustering posits that data samples within the same cluster should reside in a common subspace. Widely applied across various domains such as computer vision, face clustering [24, 55, 75], video analysis [66], image representation and compression [14], hyperspectral image processing [73], saliency detection [6], motion segmentation [65], and domain adaptation task [28], subspace clustering has become a versatile tool.

Existing subspace clustering methods fall into five broad categories [60]: iterative models [13], algebraic models [62], statistical models [8], spectral clustering-based models [6, 23, 30, 66], and deep learning-based models [15, 17, 18, 42, 78]. Among these, spectral clustering-based subspace clustering stands out as the most renowned due to its extensive exploration and practical applications [6, 54, 71]. The

✉ Anurag Goel
anurag@dtu.ac.in

Jyoti Maggu
jyotim@iiitd.ac.in

[1] CSED, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

[2] Department of Computer Science and Engineering, Delhi Technological University, New Delhi 110042, India

Published online: 19 September 2024

spectral clustering-based subspace clustering primarily consists of two components: (1) the construction of an affinity matrix and (2) spectral clustering [53]. Recently, diverse forms of affinity matrices have surfaced, depending on the regularization term incorporated. The most notable models include sparse subspace clustering (SSC) [4, 5, 59] and clustering based on low-rank representation (LRR) [3, 12, 26, 63, 72, 77]. These approaches, rooted in sparse or low-rank representation techniques, acquire a coefficient matrix through a self-expression model. SSC enforces an l_1 norm constraint on coefficients, while LRR employs l_* norm or nuclear norm constraints on coefficients. Several works focused on enhancing the robustness of subspace clustering algorithms by employing a multi-objective framework and subspace optimization techniques [41], integrating subspace fuzzy clustering techniques [64] or adjusting the threshold based on the distribution of data points within subspaces [22].

Various techniques for subspace clustering have been developed, each employing distinct constraints to derive an optimal affinity matrix [1, 2, 30, 44, 56, 61, 71]. However, these methods are limited to processing linear subspaces, proving inadequate for handling the prevalent nonlinear structures found in real-world data [21, 52]. Another constraint is the assumption that data can be separated into distinct subspaces. Recently, techniques based on multiview clustering [11, 45–47, 57, 58, 74, 76] have also gained attention.

To address the limitations associated with linear subspace clustering on nonlinear data, kernel self-expression methods have been introduced [16, 27, 38–40, 66]. Kernelized SSC (KSSC) [40] and kernelized LRR (KLRR) [37, 66] are two prominent methods that capture nonlinear structure information in the input space, exhibiting significant advancements in various applications. Despite their efficiency in processing nonlinear data, these kernel subspace clustering methods may lose some similarity information between samples during the reconstruction of the original data in kernel space.

Recently, transform learning-based approaches for subspace clustering [9, 10, 33–35] have garnered attention. These methods operate on data originally deemed inseparable into subspaces, transforming it into a high-dimensional feature space where linear separability into subspaces is achievable. Notably, these methods eliminate the need to manually choose a mapping function, as they autonomously learn the mapping from the data itself.

In practical scenarios, real-world data exhibiting manifold structures often entails complexities beyond mere sparsity or low-rank characteristics. Consequently, it becomes crucial to formulate a representation that can adeptly capture the intricate structural information inherent in the original data. Numerous methodologies have been devised to uncover underlying structures by delving into data relationships [33–35]. Recently introduced subspace clustering methodologies grounded in structure learning comprise Similarity Learning via Kernel-Preserving Embedding (SLKE) [20] and Structure Learning with Similarity Preserving (SLSP) [19]. SLKE constructs a model that preserves similarity information among data, resulting in improved performance. In contrast, SLSP establishes a structure learning framework that integrates the similarity information of the original data, addressing potential drawbacks associated with the SLKE algorithm, which could lead to the loss of certain low-order information. Despite these methods exhibiting commendable performance, their effectiveness is dependent on a learned similarity matrix that might lack an optimal block diagonal structure for spectral clustering.

In the literature, diverse norm regularization terms have been utilized in self-expressive models to acquire a block diagonal coefficient matrix. These terms include the 1-norm, l2-norm, and nuclear norm. However, these regularization techniques exhibit two shortcomings: an inability to control the number of blocks in the coefficient matrix and the potential suboptimality of the learned coefficient matrix due to data noise. Addressing these limitations, Block Diagonal Representation (BDR) subspace clustering algorithms [30, 70] have been introduced, aiming to attain a good block diagonal structure in the coefficient matrix. For example, Implicit Block Diagonal Low-Rank Representation (IBDLR) integrates block diagonal priors and implicit feature representation into the low-rank representation model, progressively enhancing clustering performance [67]. Notably, these BDR-based subspace clustering methods, while effective, have not been integrated into similarity-preserving mechanisms. The kernelized version of transform learning was introduced in [32].

This work proposes “Similarity Preserving Kernel Block Diagonal Representation based Transformed Subspace Clustering (KBD-TSC)” that leverages the kernel self-expressing framework. Kernelized transformed subspace clustering accounts for the data that is not originally separable into subspaces by leveraging kernel self-expression-based transform learning. The proposed method transforms the data into high-dimensional feature spaces where they are linearly separable into subspaces. It doesn’t suffer from choosing the mapping function, as it learns the mapping from the data itself. Although the kernel subspace clustering methods based on kernel self-expression can efficiently process the nonlinear structure data, some similarity information between samples may be lost when reconstructing the original data in kernel space. The integration of similarity preserving regularizer and block diagonal regularizer into the proposed model facilitates enhanced preservation of similarity information between the original data points. Experimental results on nine datasets validate the effectiveness and robustness of the proposed KBD-TSC method.

The principal contributions of this paper are as follows:

- A novel subspace clustering approach is proposed which accounts for the data that is not originally separable into subspaces by leveraging kernel self-expression-based transform learning.
- A similarity preserving regularizer is incorporated in the proposed model to facilitate enhanced preservation of similarity information between the original data points.
- The integration of a block diagonal representation into the proposed model aims to derive a similarity matrix characterized by an optimal block diagonal structure. This helps in achieving the desired optimal block diagonal matrix.
- The proposed KBD-TSC model is evaluated on nine datasets featuring different types of manifolds, including handwritten digits clustering, face image clustering, object clustering, and text clustering. The experiments involved comparing the proposed model with several state-of-the-art approaches. The results strongly support the effectiveness of our proposed model.

The structure of the remaining paper is organized as follows: Sect. 2 provides a review of basic concepts in related work. Section 3 elaborates on the proposed algorithms and their solutions. Subsequently, Sect. 4 discusses experimental results, and Sect. 5 concludes the paper.

2 Background

2.1 Subspace clustering

The standard subspace clustering techniques are self-expression-oriented, where the aim is to express every single data point in terms of the linear combination of the rest of other data points that lie under the same subspace. The data must hold the basic pre-requisite of being separable into various subspaces. Let $\boldsymbol{X} = [x_1, x_2, \cdots, x_N] \in \Re^{d \times N}$ be the matrix of data points where every column vector $\boldsymbol{x_i}$ is drawn from a union of lower-dimensional subspaces $\{\boldsymbol{S_1} \bigcup \boldsymbol{S_2} \bigcup \cdots \boldsymbol{S_n}\}$ that have dimensions $\{d_k\}_{k=1}^{n}$ where n is the total number of manifolds. The subspace clustering technique targets to segments each set X_k of N_k points that basically belong to the same subspace S_k of dimension $d-k$.

$$\underset{\mathbf{Z}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{X} - \mathbf{XZ}\|_F^2 + \lambda(\Omega(\mathbf{Z})),\ \text{s.t.}\ \ \text{diag}(\mathbf{Z}) = 0,\ \mathbf{Z} \geq 0. \tag{1}$$

where $\Omega(\boldsymbol{Z})$ is the regularization term and $\lambda > 0$ denotes the hyperparameter. $\|\mathbf{Z}\|_*$, $\|\mathbf{Z}\|_1$, $\|\mathbf{Z}\|_F^2$ are three common regularizers. Then an affinity matrix is constructed using $\mathbf{Z}$, which applies any graph-cut technique to compute clusters.

2.2 Kernelized subspace clustering

The kernelized version for the subspace clustering is backed by mapping a kernel function as follows:

$$\begin{aligned} &\underset{\mathbf{Z}}{\text{minimize}}\ \frac{1}{2}\|\ker(\mathbf{X}) - \ker(\mathbf{X})\mathbf{Z}\|_F^2 + \lambda(\Omega(\mathbf{Z})) \\ &\equiv \underset{\mathbf{Z}}{\text{minimize}}\ Tr(\mathbf{I} - 2\mathbf{Z} + \mathbf{Z}^\top\mathbf{Z})K + \lambda(\Omega(\mathbf{Z})), \\ &\quad \text{s.t.} \quad \text{diag}(\mathbf{Z}) = 0,\ \mathbf{Z} \geq 0. \end{aligned} \tag{2}$$

where $\ker(\boldsymbol{X})$ is the kernel mapping function, $\boldsymbol{K}$ is the kernel matrix where each matrix element is calculated as $K_{i,j} = \ker(X_i)^\top \ker(X_j)$.

2.3 Kernelized transform learning

A more recent unsupervised representation learning methodology known as transform learning that functions as a dictionary learning method with analysis capabilities. It seeks to train the transform matrix and coefficient matrix from the input data in such a way that the learned transform is capable of analysing the data and ultimately generating the coefficients matrix [48–51]. Since we anticipate that the input data would be able to be divided into many groups, we typically use raw data pixels as input to the clustering method. However, transform learning can be applied to any high-dimensional data, resulting in the effective representations of the transform matrix and the coefficients' matrix being learned in latent space. The data's nonlinearity can be handled by the kernel approach in an effective and efficient manner. Kernelization can be used if the learned transform coefficients are not linearly divided into distinct subspaces. The kernel transform learning [32] formulation can be given as Eq. 3.

$$\underset{\mathbf{A},\mathbf{Z}}{\text{minimize}}\ \|\mathbf{AK} - \mathbf{Z}\|_F^2 + \epsilon(\|\mathbf{A}\|_F^2 - logdet(\mathbf{A})) + \mu\|\mathbf{Z}\|_1. \tag{3}$$

where $\boldsymbol{K} = \ker(\mathbf{X})^\top \ker(\mathbf{X})$ is the kernel, ϵ and μ are the hyperparameters. Constraints introduced in Eq. 3 assist prevent trivial solutions. The first constraint $-$ logdet[1]$(\boldsymbol{A})$

[1] logdet($\mathbf{T}$)= log(singular values). If some singular value ≤ 0, then the log takes + ∞ as output. For the case when $\boldsymbol{T}$ is not square, the algorithm solves -logdet($\boldsymbol{T'T}$) + $||\boldsymbol{T}||_F^2$.

guarantees that $\boldsymbol{A}$ is the complete matrix. To prevent it from rising to the other extreme, the second restriction $\|\boldsymbol{A}\|_F^2$ is used. The other restriction, $\|\mathbf{Z}\|_1$, is used to make the coefficients sparser.

Equation 3. is solved using an alternating minimization method. Equation 4. describes how the iterative updating procedures for $\boldsymbol{A}$ and $\boldsymbol{Z}$ are carried out alternately.

$$\begin{aligned} &\mathbf{A} \leftarrow \underset{\mathbf{A}}{\text{minimize}}\ \|\mathbf{AK}-\mathbf{Z}\|_F^2 + \epsilon(\|\mathbf{A}\|_F^2 - \log\det(\mathbf{A}));\\ &\mathbf{Z} \leftarrow \underset{\mathbf{Z}}{\text{minimize}}\ \|\mathbf{AK}-\mathbf{Z}\|_F^2 + \mu\|\mathbf{Z}\|_1. \end{aligned} \tag{4}$$

The update of $\boldsymbol{A}$ is pretty straightforward as it is directly differentiable. There are some linear algebraic tricks to solve it too which are given in [31]. Equation 5 describes how to perform one-step soft thresholding to update $\boldsymbol{Z}$.[2]

$$\mathbf{Z} \leftarrow signum(\mathbf{AK}).\max(0, abs(\mathbf{AK}) - \mu). \tag{5}$$

2.4 Transformed subspace clustering

Subspace clustering operates on the underlying assumption that the given data can be segmented into numerous sub-spaces. Yet, in the context of high-dimensional data, this assumption often does not hold true. To address this challenge, we employ clustering algorithms on the features derived from transform learning. This approach involves projecting the data onto a latent space, where we acquire separable coefficients that define distinct sub-spaces [34]. The modified subspace clustering can be articulated as follows, focusing on acquiring sparse representations based on the transformed features.

$$\begin{aligned} &\underset{\mathbf{A,Z,C}}{\text{minimize}}\ \|\mathbf{AX}-\mathbf{Z}\|_F^2 + \epsilon(\|\mathbf{A}\|_F^2 - \log\det(\mathbf{A}))\\ &+\gamma\|\mathbf{Z}-\mathbf{ZC}\|_F^2 + \lambda(\Omega(\mathbf{C})). \end{aligned} \tag{6}$$

Alternate minimization, as described in Eq. 7, is utilised to solve Eq. 6. In each cycle, we alternately update $\boldsymbol{A}$, $\boldsymbol{Z}$, and $\boldsymbol{C}$.

$$\begin{aligned} &\mathbf{A} \leftarrow \underset{\mathbf{A}}{\text{minimize}}\ \|\mathbf{AX}-\mathbf{Z}\|_F^2 + \epsilon(\|\mathbf{A}\|_F^2 - \log\det(\mathbf{A}));\\ &\mathbf{Z} \leftarrow \underset{\mathbf{Z}}{\text{minimize}}\ \|\mathbf{AX}-\mathbf{Z}\|_F^2 + \gamma(\|\mathbf{Z}-\mathbf{ZC}\|_F^2);\\ &\mathbf{C} \leftarrow \underset{\mathbf{C}}{\text{minimize}}\ \|\mathbf{Z}-\mathbf{ZC}\|_F^2 + \lambda(\Omega(\mathbf{C})). \end{aligned} \tag{7}$$

After calculating the affinity matrix and applying spectral clustering to it to discover clusters, we obtain $\boldsymbol{C}$. Additionally, to create the kernelized transform subspace clustering formulation, the subspace clustering loss is included in equation 6.

2.5 Block diagonal representation

By adding a block-diagonal regularisation term, the BDR method [29] completes the block diagonal matrix and achieves higher clustering performance. The BDR algorithm's optimization model is expressed as follows:

$$\begin{aligned} &\underset{\mathbf{Z}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{Z}\|_{\boxed{\text{m}}} + \|\ker(\mathbf{X}) - \ker(\mathbf{X})\mathbf{Z}\|_F^2,\\ &s.t.\mathbf{Z} \geq 0, \mathbf{Z}^{\top} = \mathbf{Z}, diag(\mathbf{Z}) = 0. \end{aligned} \tag{8}$$

Here, $\boldsymbol{X}$ and $\boldsymbol{Z}$ represent data matrix and coefficient matrix respectively, while $\|\mathbf{Z}\|_{\boxed{\text{m}}}$ denotes m-block diagonal regularizer.

3 Proposed method: KBD-TSC

The fundamental assumption underlying self-expression-based subspace clustering is the requirement for data to be segregable into distinct subspaces. Traditional methods also rely on the assumption of an inherent linear structure. However, when these assumptions are not met, particularly when data samples are non-separable into subspaces and linear subspace clustering methods struggle with non-linear structures, a more adaptable model is needed. In response to this challenge, we present a model designed to generalize effectively on non-linear structured data, even when they are not easily separable into subspaces.

The proposed Kernel Block Diagonal-based Transformed Subspace Clustering (KBD-TSC), is adept at preserving similarity information among samples, concurrently achieving an optimal block diagonal structure in the obtained similarity matrix. This approach involves embedding a non-linear model that integrates kernelized transformed subspace clustering with a kernel self-expression framework to achieve the desired objective. Incorporating a block diagonal regularization term into the kernel self-expression framework is pivotal for obtaining a similarity matrix characterized by a block diagonal structure. Furthermore, the preservation of similarity information is secured by minimizing the distinction between two inner products: one encompassing the inner products among original data in kernel space and the other involving the inner products of reconstructed data in

[2] The signum function returns -1 if the argument is negative, 0 if the argument is zero, and 1 if the argument is positive.

kernel space. The resolution of the entire optimization problem employs alternate minimization techniques.

3.1 Similarity preserving model

To uphold similarity information among samples, our objective is to minimize the difference between two inner products. One corresponds to the inner product among the original data in kernel space, and the other corresponds to the inner product of the reconstructed data in kernel space, drawing inspiration from the work of Kang et al. [20].

$$\underset{\mathbf{Z}}{\text{minimize}}\ \frac{1}{2}\|\mathbf{K}-\mathbf{Z}^\top\mathbf{K}\mathbf{Z}\|_F^2. \tag{9}$$

where $\boldsymbol{K}=\ker(\mathbf{X})^\top\ker(\mathbf{X})$ is a positive semi-definite matrix.

3.2 Proposed algorithm

We introduce a model designed for effective generalization on non-linear manifolds. This model integrates kernelized self-expression transformed subspace clustering with a similarity-preserving kernel block diagonal representation. The kernelized component of transform learning characterizes non-linear data as a linear combination of itself in the transform domain. The incorporation of transform learning with subspace clustering loss facilitates the separation of data into subspaces. To enhance this process, a block diagonal regularization term is introduced, aiming to achieve a similarity matrix between samples with a block diagonal structure. Consequently, our proposed model not only preserves the similarity information among non-linear samples in the transform domain but also simultaneously acquires a similarity matrix with an optimal block diagonal structure.

The complete joint formulation for the proposed model is expressed as Eq. 10.

$$\begin{aligned}\underset{\mathbf{A},\mathbf{Z}}{\text{minimize}}&\left(\underbrace{\|\mathbf{AK}\text{-}\mathbf{Z}\|_F^2+\epsilon(\|\mathbf{A}\|_F^2-\log\det(\mathbf{A}))}_{\text{Kernelized transform learning}}\right.\\ &+\underbrace{\frac{1}{2}Tr(\mathbf{Z}-2\mathbf{KZ}+\mathbf{Z}^T\mathbf{KZ})}_{\text{Kernel self expression subspace clustering}}\ \left.+\underbrace{\alpha\|\mathbf{K}-\mathbf{Z}^T\mathbf{KZ}\|_F^2+\gamma\|\mathbf{Z}\|_{\boxed{k}}}_{\text{Similarity preserving with block diagonal}}\right)\\ &s.t.\ \mathbf{Z}\geq 0, diag(\mathbf{Z})=0,\mathbf{Z}^\top=\mathbf{Z}.\end{aligned} \tag{10}$$

where α,ϵ,γ are positive hyperparameters.

To simplify and separate out the variables, let us introduce an auxiliary matrix $\boldsymbol{B}$ and a regularization term $\|\boldsymbol{Z}-\boldsymbol{B}\|_F^2$ into our proposed model. Thus, the optimization problem in equation 10 can be translated to

$$\begin{aligned}&(\|\mathbf{AK}-\mathbf{Z}\|_F^2+\epsilon\left(\|\mathbf{A}\|_F^2-\log\det(\mathbf{A})\right)+\tfrac{1}{2}Tr\left(\mathbf{K}-2\mathbf{KZ}+\mathbf{Z}^\top\mathbf{KZ}\right)\\ \underset{\mathbf{A},\mathbf{Z},\mathbf{B}}{\text{minimize}}\ &+\alpha\left\|\mathbf{K}-\mathbf{Z}^\top\mathbf{KZ}\right\|_F^2+\tfrac{\beta}{2}\|\mathbf{Z}\text{-}\mathbf{B}\|_F^2+\gamma\|\mathbf{B}\|_{\boxed{k}})\\ &s.t.\ \mathbf{Z}\geq 0, diag(\mathbf{Z})=0,\mathbf{Z}^\top=\mathbf{Z}.\end{aligned} \tag{11}$$

3.3 Optimization of the proposed KBD-TSC model

To facilitate the solution of the problem in equation 11, three new auxiliary variables are introduced that leads to the following equivalent problem:

$$\begin{aligned}&(\|\mathbf{AK}-\mathbf{Z}\|_F^2+\epsilon\left(\|\mathbf{A}\|_F^2-\log\det(\mathbf{A})\right)+\tfrac{1}{2}Tr\left(\mathbf{K}-2\mathbf{KJ}+\mathbf{Z}^\top\mathbf{K}\,\mathbf{J}\right)\\ \underset{\mathbf{A},\mathbf{Z},\mathbf{B},\mathbf{J},\mathbf{G},\mathbf{H}}{\text{minimize}}\ &+\alpha\left\|\mathbf{K}-\mathbf{G}^\top\mathbf{K}\,\mathbf{H}\right\|_F^2+\tfrac{\beta}{2}\|\mathbf{J}\text{-}\mathbf{B}\|_F^2+\gamma\|\mathbf{B}\|_{\boxed{k}})\\ &s.t.\ \mathbf{B}\geq 0, diag(\mathbf{B})=0,\mathbf{B}^\top=\mathbf{Z},\mathbf{G}=\mathbf{Z},\mathbf{H}=\mathbf{Z}.\end{aligned} \tag{12}$$

We use ADMM for solving equation 12, and its corresponding Lagrangian [25] is given as follows:

$$\begin{aligned}&\check{}\left(\mathbf{Z},\mathbf{J},\mathbf{G},\mathbf{H},\mathbf{B},\lambda_1,\lambda_2,\lambda_3\right)\\ &=\|\mathbf{AK}\text{-}\mathbf{Z}\|_F^2+\epsilon\left(\|\mathbf{A}\|_F^2-\log\det\mathbf{A}\right)+\tfrac{1}{2}Tr\left(\mathbf{K}-2\mathbf{KJ}+\mathbf{Z}^\top\mathbf{K}\,\mathbf{J}\right)\\ &\quad+\alpha\left\|\mathbf{K}-\mathbf{G}^\top\mathbf{K}\,\mathbf{H}\right\|_F^2+\tfrac{\beta}{2}\|\mathbf{J}\text{-}\mathbf{B}\|_F^2+\gamma\|\mathbf{B}\|_{\boxed{k}}\\ &\quad+\tfrac{\mu}{2}\left[\left\|\mathbf{J}\text{-}\mathbf{Z}+\tfrac{\lambda_1}{\mu}\right\|_F^2+\left\|\mathbf{G}\text{-}\mathbf{Z}+\tfrac{\lambda_2}{\mu}\right\|_F^2+\left\|\mathbf{H}\text{-}\mathbf{Z}+\tfrac{\lambda_3}{\mu}\right\|_F^2\right].\end{aligned} \tag{13}$$

where $\lambda_1,\lambda_2,\lambda_3$ are Lagrangian multipliers and $\mu>0$ is a penalty parameter. Now, these variables can be updated alternately. The updates for all variables are given as follows:

- **Update A**: After keeping other variables fixed, $\boldsymbol{A}$ can be updated as follows:

  $$\underset{\mathbf{A}}{\text{minimize}}\ \|\mathbf{AK}\text{-}\mathbf{Z}\|_F^2+\epsilon\left(\|\mathbf{A}\|_F^2-\log\det(\mathbf{A})\right). \tag{14}$$

  For updating transform, given the original data as in equation 15, we can use equation 16.

  $$\underset{\mathbf{T}}{\text{minimize}}\ \|\mathbf{TX}\text{-}\mathbf{Z}\|_F^2+\epsilon(\|\mathbf{T}\|_F^2-\log\det(\mathbf{T})). \tag{15}$$

  The update of the transform matrix $\boldsymbol{T}$ is straightforward, as each term is directly differentiable. But, there are better ways of solving the update of $\boldsymbol{T}$ by using some linear algebraic tricks [51].

  $$\begin{aligned}\mathbf{XX}^\top+\epsilon\mathbf{I}&=\mathbf{LL}^\top,\\ \mathbf{L}^{-1}\mathbf{XZ}^\top&=\mathbf{USV}^\top,\\ \mathbf{T}\leftarrow\frac{1}{2}\mathbf{U}(\mathbf{S}&+(\mathbf{S}^2+\epsilon\mathbf{I})^{1/2})\mathbf{V}^\top\mathbf{L}^{-1}.\end{aligned} \tag{16}$$

Now, the solution to equation 14 is similar to the update of $\boldsymbol{T}$ in equation 16. Here $\boldsymbol{A}$ acts as transform matrix $\boldsymbol{T}$, and instead of passing original samples $\boldsymbol{X}$, we input kernel matrix $\boldsymbol{K}$.

- **Update J**: After keeping other variables fixed, $\boldsymbol{J}$ can be updated as follows:

$$\begin{aligned}&\underset{\mathbf{J}}{\text{minimize}}\ \tfrac{1}{2}Tr\big(\mathbf{K}-2\mathbf{KJ}+\mathbf{Z}^\top\mathbf{KJ}\big)\\&+\tfrac{\beta}{2}\|\mathbf{J}\text{-}\mathbf{B}\|_F^2+\tfrac{\mu}{2}\|\mathbf{J}\text{-}\mathbf{Z}+\tfrac{\lambda_1}{\mu}\|_F^2\\&s.t.\ \mathbf{B}\geq 0, diag(\mathbf{B})=0, \mathbf{B}^\top=\mathbf{B}.\end{aligned} \tag{17}$$

Taking its first derivative and equating it to 0 gives:

$$\mathbf{J}=(\mathbf{K}+\mathbf{B}+\mu\mathbf{I})^{-1}\big(\mathbf{K}+\mu\mathbf{Z}-\lambda_1+\beta\mathbf{B}\big). \tag{18}$$

- **Update G**: After keeping other variables fixed, $\boldsymbol{G}$ can be updated as follows:

$$\underset{\mathbf{G}}{\text{minimize}}\ \alpha\|\mathbf{K}\text{-}\mathbf{G}^\top\mathbf{KH}\|_F^2+\frac{\mu}{2}\|\mathbf{G}\text{-}\mathbf{Z}+\frac{\lambda_2}{\mu}\|_F^2. \tag{19}$$

Taking its first derivative and equating to 0 gives:

$$\mathbf{G}=\big(2\alpha\mathbf{KHH}^\top\mathbf{K}^\top+\mu\mathbf{I}\big)^{-1}\big(2\alpha\mathbf{KHK}^\top+\mu\mathbf{Z}-\lambda_2\big). \tag{20}$$

- **Update H**: After keeping other variables fixed, $\boldsymbol{H}$ can be updated as follows:

$$\underset{\mathbf{H}}{\text{minimize}}\ \alpha\|\mathbf{K}-\mathbf{G}^\top\mathbf{KH}\|_F^2+\frac{\mu}{2}\left\|\mathbf{H}\text{-}\mathbf{Z}+\frac{\lambda_3}{\mu}\right\|_F^2. \tag{21}$$

Taking its first derivative and equating to 0 gives:

$$\mathbf{H}=\big(2\alpha\mathbf{K}^\top\mathbf{GG}^\top\mathbf{K}+\mu\mathbf{I}\big)^{-1}\big(2\alpha\mathbf{K}^\top\ \mathbf{G}\ \mathbf{K}+\mu\mathbf{Z}-\lambda_3\big). \tag{22}$$

- **Update Z**: After keeping other variables fixed, the sub-problem becomes:

$$\underset{\mathbf{Z}}{\text{minimize}}\ \frac{3\mu}{2}\left\|\mathbf{Z}-\frac{\mathbf{J}+\mathbf{G}+\mathbf{H}+\mathbf{AK}+\big(\lambda_1+\lambda_2+\lambda_3\big)/\mu}{3}\right\|_F^2. \tag{23}$$

Taking its first derivative and equating to 0 gives:

$$\mathbf{Z}=\frac{\mathbf{J}+\mathbf{G}+\mathbf{H}+\mathbf{AK}+\big(\lambda_1+\lambda_2+\lambda_3\big)/\mu}{3}. \tag{24}$$

- **Update B**: After keeping other variables fixed, $\boldsymbol{B}$ can be updated as follows:

$$\underset{\mathbf{B}}{\text{minimize}}\ \frac{\beta}{2}\|\mathbf{J}-\mathbf{B}\|_F^2+\gamma\|\mathbf{B}\|_{\boxed{\text{k}}}\ s.t.\ \mathbf{B}\geq 0, diag(\mathbf{B})=0, \mathbf{B}^\top=\mathbf{B}. \tag{25}$$

Using K.Fan theorem [7], equation 25 can be rewritten as follows:

$$\begin{aligned}&\underset{\mathbf{B}}{\text{minimize}}\ \tfrac{\beta}{2}\|\mathbf{J}-\mathbf{B}\|_F^2+\gamma\langle diag(\mathbf{B})-\mathbf{B},\mathbf{S}\rangle\\&s.t.\ \mathbf{B}\geq\mathbf{0}, diag(\mathbf{B})=0, \mathbf{B}^\top=\mathbf{B}, \mathbf{0}<\mathbf{S}<\mathbf{I}, Tr(\mathbf{S})=k.\end{aligned} \tag{26}$$

where $\boldsymbol{S}=\boldsymbol{U}\boldsymbol{U}^\top$, $\boldsymbol{U}$ consists of k eigenvectors that correspond to k smallest eigenvalues of $diag(\boldsymbol{B})-\boldsymbol{B}$. Now, equation 26 can be translated to:

$$\underset{\mathbf{B}}{\text{minimize}}\ \frac{1}{2}\left\|\mathbf{B}-J+\frac{\gamma}{\beta}\big(diag(\mathbf{S})1^\top-\mathbf{S}\big)\right\|_F^2. \tag{27}$$

Let us define

$$\boldsymbol{Q}=\boldsymbol{J}-\frac{\gamma}{\beta}\big(diag(\boldsymbol{S})1^\top-\boldsymbol{S}\big),$$

$$\tilde{\boldsymbol{Q}}=\boldsymbol{Q}-Diag(diag(\boldsymbol{Q})),$$

then $\boldsymbol{B}=max(0,(\boldsymbol{Q}+\tilde{\boldsymbol{Q}})/2)$.

Once we obtain the matrix $\boldsymbol{B}$, the similarity matrix can be computed by $(\boldsymbol{B}+\boldsymbol{B}^\top)/2$. After this, the clustering results can be obtained by applying spectral clustering on the similarity matrix. The step-by-step algorithm is given as algorithm 1 to understand the entire model better.

Algorithm 1 The proposed algorithm: KBD-TSC

```
Require: $\boldsymbol{X} = [\boldsymbol{X_1}, \boldsymbol{X_2}, \cdots, \boldsymbol{X_N}], \epsilon, \mu, \alpha, \beta, \gamma$, no. of clusters $k$
  Compute kernel matrix $\boldsymbol{ker(X, X)}$.
  Initialize random matrix $\boldsymbol{Z, G, A, H, B}$, $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 0$.
  while The convergence is not reached do
    Compute transform matrix $\boldsymbol{A}$ using equation 16.
    Update matrix $\boldsymbol{J}$ using equation 18.
    Update matrix $\boldsymbol{G}$ using equation 20.
    Update matrix $\boldsymbol{H}$ using equation 22.
    Update matrix $\boldsymbol{Z}$ using equation 24.
    Update matrix $\boldsymbol{B}$ using equation 27.
    Update Lagrangian multipliers $\lambda_1, \lambda_2, \lambda_3$.
  end while
  Compute the similarity matrix by $(\boldsymbol{B} + \boldsymbol{B}^\top)/2$.
  Applying spectral clustering on the similarity matrix to obtain resulting clusters.
```

4 Experimental results and analysis

4.1 Dataset description

The proposed KBD-TSC algorithm is evaluated on nine images datasets which are as follows:

1. Yale[3]: It consists of 165 facial images of 15 individuals in grayscale mode. The images are resized to 32×32 pixels.
2. Jaffe[4]: This dataset consists of 213 facial images corresponding to 7 facial expressions. The images are resized to 26 × 26 pixels.
3. ORL[5]: It consists of 400 facial images of 40 subjects. Each image is of size 26 × 26 pixels.
4. ARFaces[6]: This dataset comprises of 4000 facial images of 126 people.
5. COIL20[7]: It consists of 1440 images of 20 objects. Each image size is 32 × 32 pixels.
6. BA[8]: This dataset contains 1404 images of handwritten digits and uppercase alphabets. Each image size is 20 ×16 pixels.
7. tr11[9]: This is a text dataset consists of 414 samples, 6429 features and 9 classes.
8. tr41[10]: This is a text dataset consists of 878 samples, 7454 features and 10 classes.
9. tr45[11]: This is a text dataset consists of 690 samples, 8261 features and 10 classes.

4.2 Baseline methods

We compare our proposed KBD-TSC method with several state-of-the-art methods, including spectral clustering (SC)

Table 1 Details of parameter values w.r.t. different kernel functions

Kernel	Function	Parameter values
Linear	$k(x_i, x_j) = x_i^\top x_j$	–
Polynomial	$k(x_i, x_j) = \left(x_i^\top x_j + a\right)^b$	$a \in \{0, 1\}, b \in \{2, 4\}$
Gaussian	$k(x_i, x_j) = exp\left(-\left\|x_i - x_j\right\|_2^2 / 2t\sigma^2\right)$	$t \in \{0.01, 0.05, 0.1, \}$ $\{1, 10, 50, 100\}$

[3] http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html.

[4] https://zenodo.org/record/3451524.

[5] https://cam-orl.co.uk/facedatabase.html.

[6] https://www2.ece.ohio-state.edu/~aleix/ARdatabase.html.

[7] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[8] https://cs.nyu.edu/~roweis/data/.

[9] https://trec.nist.gov/.

[10] https://trec.nist.gov/.

[11] https://trec.nist.gov/.

Table 2 Hyperparameter settings

Hyperparameter	Value
ϵ	$\{1e-2, \mathbf{1e\text{-}1}, 0.5, 1\}$
α	$\{1e-5, 1e-4, 1e-3, \mathbf{1e\text{-}2}, 0.1, 1\}$
β	$\{1e-5, \mathbf{1e\text{-}3}, 0.1, 1\}$
γ	$\{1e-2, \mathbf{1e\text{-}1}, 1, 10, 30, 50\}$

[36], kernelized sparse subspace clustering (KSSC) [40], Kernel low-rank representation (KLRR) [37], Implicit block diagonal low-rank representation (IBDLR) [67], Similarity Learning via Kernel Preserving Embedding sparse (SLKEs) [20], Similarity Learning via Kernel Preserving Embedding low rank (SLKEr) [20], Structure learning with similarity preserving sparse (SLSPs) [19], Structure learning with similarity preserving low-rank (SLSPr) [19] and Kernel block

Table 3 Comparison of clustering results based upon accuracy

Dataset	SC	KSSC	KLRR	SKLEs	SKLEr	SLSPs	SLSPr	IBDLR	KBDSP	KBD-TSC
YALE	0.4425	0.4715	0.4569	0.3404	0.4768	0.3768	0.5286	0.4727	0.5626	**0.5819**
JAFFE	0.5563	0.7691	0.7358	0.6625	0.7409	0.7514	0.9031	0.9081	0.9091	**0.9192**
ORL	0.4948	0.4573	0.4971	0.3696	0.5865	0.3858	0.6075	0.6071	0.6340	**0.6542**
AR	0.2205	0.4116	0.4888	0.2866	0.2479	0.4247	0.5056	0.5010	0.5083	**0.5383**
COIL-20	0.4205	0.5744	0.5776	0.4537	0.5236	0.5341	0.5362	0.6922	0.7119	**0.7318**
BA	0.2785	0.3711	0.3280	0.3533	0.4088	0.4103	0.4128	0.3643	0.4252	**0.4452**
TR11	0.4335	0.4443	0.3364	0.4443	0.4986	0.4016	0.5135	0.5183	0.5234	**0.5537**
TR41	0.4452	0.5179	0.5344	0.3885	0.4588	0.4192	0.5555	0.4964	0.5558	**0.5688**
TR45	0.4561	0.4771	0.4953	0.5594	0.5632	0.5319	0.6025	0.5284	**0.6492**	0.6277
Avg	0.4164	0.4994	0.4945	0.4243	0.5006	0.4706	0.5739	0.5654	0.6088	**0.6245**

Table 4 Comparison of clustering results based upon NMI score

Dataset	SC	KSSC	KLRR	SKLEs	SKLEr	SLSPs	SLSPr	IBDLR	KBDSP	KBD-TSC
YALE	0.4859	0.4761	0.4882	0.3799	0.5047	0.4030	0.5425	0.5122	0.5662	**0.5771**
JAFFE	0.6069	0.7145	0.7414	0.7847	0.7691	0.7656	0.8991	0.8971	0.9178	**0.9191**
ORL	0.6981	0.6094	0.5461	0.5614	0.7580	0.5304	0.7758	0.7696	0.7964	**0.8015**
AR	0.5797	0.6687	0.7131	0.6022	0.5616	0.6750	0.7466	**0.7484**	0.6954	0.7477
COIL-20	0.5381	0.7050	0.6888	0.6047	0.6657	0.6628	0.6649	0.7877	0.7977	**0.8015**
BA	0.4387	0.5249	0.4501	0.5278	0.5361	0.5441	0.5461	0.4997	0.5704	**0.5789**
TR11	0.3142	0.3863	0.2375	0.2831	0.4408	0.2744	0.4576	0.4372	0.4752	**0.4856**
TR41	0.3657	0.4438	0.4642	0.3635	0.3691	0.3170	0.5345	0.4462	0.5453	**0.5644**
TR45	0.3310	0.3909	0.4084	0.4773	0.4995	0.4430	0.5394	0.4570	0.6027	**0.6215**
Avg	0.4843	0.5466	0.5264	0.5094	0.5672	0.5128	0.6341	0.6172	0.6630	**0.6775**

Table 5 Comparison of clustering results based upon purity

Dataset	SC	KSSC	KLRR	SKLEs	SKLEr	SLSPs	SLSPr	IBDLR	KBDSP	KBD-TSC
YALE	0.4665	0.5379	0.5862	0.6323	0.5747	0.6313	0.6138	0.5212	0.6394	**0.6514**
JAFFE	0.5798	0.8389	0.8955	0.8108	0.8187	0.9345	0.9253	**0.9369**	0.9362	0.9368
ORL	0.5387	0.6372	0.6421	0.6302	0.6812	0.7044	0.7013	0.6765	0.7110	**0.7232**
AR	0.2558	0.5090	0.5659	0.3992	0.4545	0.5749	**0.5870**	0.5502	0.5833	**0.5870**
COIL-20	0.4559	0.6751	0.6611	0.7190	0.7420	0.7241	0.6106	0.7442	0.7578	**0.7626**
BA	0.3095	0.4219	0.4383	0.5093	0.5163	0.4573	0.4749	0.5102	0.5386	**0.5512**
TR11	0.5043	0.6417	0.5064	0.6254	0.6860	0.6409	**0.6886**	0.6749	0.6747	0.6881
TR41	0.5672	0.5934	0.6359	0.6472	0.6361	0.6412	**0.6724**	0.5966	0.6501	0.6711
TR45	0.5059	0.6221	0.6781	0.6449	0.6617	0.6989	0.7014	0.6563	0.7060	**0.7218**
Avg	0.4648	0.6086	0.6233	0.6243	0.6412	0.6675	0.6639	0.6519	0.6886	**0.6992**

diagonal representation subspace clustering with similarity preservation (KBDSP) [68].

4.3 Evaluation metrics

During experiments, it is typically presumed that the quantity of clusters is already established. Under these circumstances, various metrics, including accuracy, Normalized Mutual Information (NMI) and Purity, are commonly employed [42, 43]. The metrics are described below:

- Accuracy: The accuracy is defined as the ratio of the number of data instances that are assigned the same cluster as in the ground truth to the total number of data instances.
- Normalized Mutual Information (NMI): This metric computes the normalized measure of similarity between the labels of same data instances. The range of NMI is [0,1] where 0 signifies no correlation and 1 signifies the perfect correlation.
- Purity: Purity measures the extent to which data points within each cluster are assigned to the same true class [69]. A larger purity value indicates better clustering performance.

4.4 Kernel design

We have designed 12 different kernels in this work, including one linear kernel, four polynomial kernels, and seven Gaussian kernels (Table 1).

In the case of the Gaussian kernel, σ is the maximum distance between x_i, x_j.

4.5 Computational complexity

In the proposed algorithm, the first part is the construction of kernel matrix which is bounded by $O(n^2)$. The second part is updating step of different variables, each one of them is bounded by $O(n^3)$. Thus, the proposed algorithm has overall time complexity of $O(tn^3)$ where t and n represents the number of iterations and number of data samples respectively.

4.6 Parameter sensitivity analysis

There are four hyper-parameters in the proposed KBD-TSC algorithm 11, i.e., $\epsilon, \alpha, \beta, \gamma$. The parameter ϵ controls the good conditioning of the transform. The parameter α balances the similarity-preserving term $\left\|\boldsymbol{K} - \boldsymbol{Z}^T\boldsymbol{K}\boldsymbol{Z}\right\|_F^2$, the parameter β is used to control the term $\|\boldsymbol{Z} - \boldsymbol{B}\|_F^2$, the parameter γ is used to control the block-diagonal structure term $\|\boldsymbol{B}\|_{\boxed{k}}$.The YALE and JAFFE datasets are used for parameter evaluation using NMI. The parameters $\epsilon, \alpha, \beta, \gamma$ take values from the sets $\{1e-2, 1e-1, 0.5, 1\}$, $\{1e-5, 1e-4, 1e-3, 1e-2, 0.1, 1\}$, $\{1e-5, 1e-3, 0.1, 1\}$, and $\{1e-2, 1e-1, 1, 10, 30, 50\}$ respectively. The best values of the parameters for the optimal clustering performance is 0.1, 0.01, 0.001, 0.1 for $\epsilon, \alpha, \beta, \gamma$, respectively. Therefore, we keep $\epsilon = 0.1, \alpha = 0.01, \beta = 0.001, \gamma = 0.1$ in all the experiments of this paper. The parameter tuning is done using grid searching. The parameter settings of all experiments have been given in the Table 2, in which recommended parameters are indicated in bold. We performed parameter sensitivity analysis on JAFFE Dataset for wide range of α, β and γ values against NMI score. It is observed that the proposed technique is not very sensitive to these hyperparameters. Moreover, when these parameters are set to $\alpha = 1e-2, \beta = 1e-3, and\gamma = 1e-1$, the best clustering performance is achieved. Hence, we fixed these values for all the experiments.

4.7 Results and discussion

The experimental results for all the nine datasets are shown in terms of Accuracy, NMI, and Purity in Tables 3, 4, and 5 respectively. We average out the results of all experiments on 12 kernels. The experimental results are reported by averaging the results of ten iterations in the Tables 3, 4 and 5. From the results, it can be observed that the proposed KBD-TSC approach outperforms the state-of-the-art methods.

To be more specific in analysis, the results from Tables 3, 4, 5 are discussed below:

1. Compared to SC algorithm, the proposed KBD-TSC method obtains better results regarding all evaluation metrics: accuracy, NMI, and purity. From Tables 3, 4, 5, it can be easily observed that the average value of accuracy, NMI, and purity of the proposed method are 20.81, 19.32, and 23.44 % higher than SC, respectively. The reason for the same is that the input to spectral clustering is the learned $\boldsymbol{Z}$ instead of the raw kernel matrix.
2. Our proposed KBD-TSC method also outperforms the kernel-based methods KSSC and KLLR. This is because of the similarity-preserving trick in the transform domain.
3. In comparison to SLKEs and SLKEr, our proposed algorithm exhibits superior performance. This improvement can be attributed to two key aspects: firstly, the proposed framework for kernel self-expression has the capability to retain specific low-order details from the input data; secondly, the introduction of the term representing block diagonal structures in our model, within the latent transform space, facilitates the acquisition of a similarity matrix characterized by a block diagonal arrangement.

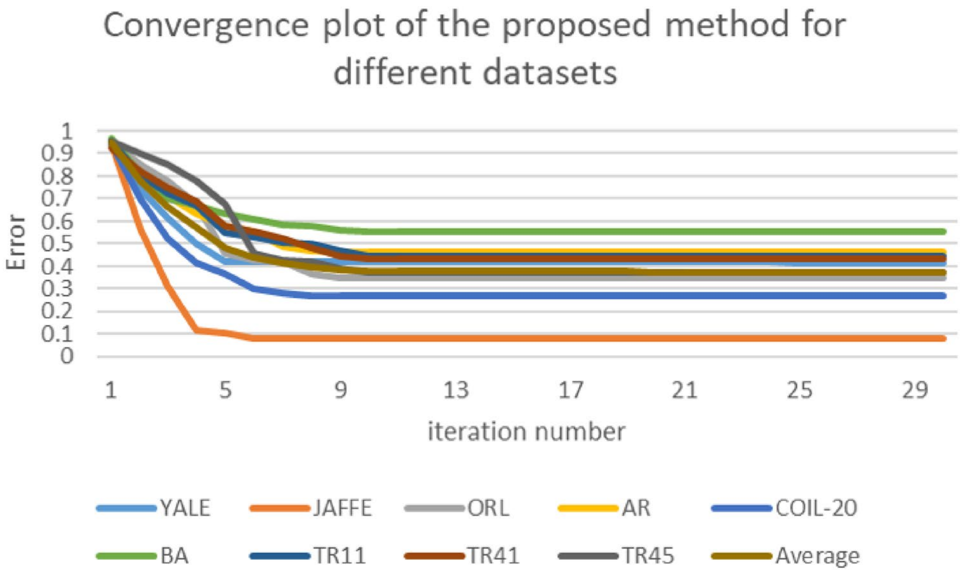

Fig. 1 Convergence graph of the proposed method with nine different datasets in 30 iterations

Table 6 Runtime comparison (in seconds)

Dataset	SSC	KSSC	KLRR	SKLEs	SKLEr	SLSPs	SLSPr	IBDLR	KBD-TSC
YALE	77	88	93	92	94	80	82	79	**68**
JAFFE	62	84	89	87	90	71	74	70	**55**
ORL	86	97	101	99	103	89	93	88	**78**
AR	91	113	119	117	122	95	97	94	**83**
COIL-20	98	117	125	121	128	102	105	100	**88**
BA	77	89	93	90	94	82	85	80	**67**
TR11	110	124	130	125	131	115	118	110	**96**
TR41	121	137	141	139	145	127	131	124	**105**
TR45	117	129	136	133	137	123	128	120	**101**

Table 7 Comparison of NMI score of the proposed objective function against the objective function without similarity preserving term and the objective function without block diagonal term for all the datasets

Dataset	Similarity-	Block-	Proposed
YALE	0.4665	0.5542	**0.5771**
JAFFE	0.8678	0.8832	**0.9191**
ORL	0.7798	0.7814	**0.8015**
AR	0.6872	0.7185	**0.7477**
COIL-20	0.7112	0.7665	**0.8015**
BA	0.4887	0.5443	**0.5789**
TR11	0.4661	0.4727	**0.4856**
TR41	0.5311	0.5392	**0.5644**
TR45	0.5991	0.6003	**0.6215**

4. SLSPs and SLSPr, which are capable of handling nonlinear datasets and preserving similarity information, gives better performance as compared to SC, KSSC, KLRR, SLKEs, and SLKEr. However, the proposed KBD-TSC algorithm consistently outperforms them in most instances. Specifically, average values of accuracy, NMI, and purity in Tables 3, 4, and 5 indicate that the proposed method surpasses SLSPs by 15.39%, 16.47%, and 3.17%, respectively. These findings confirm that the introduced term representing block diagonal structures significantly contributes to improving performance.
5. Both IBDLR and the proposed KBD-TSC mthod facilitate the acquisition of a desired affinity matrix with an optimal block diagonal structure by integrating the block diagonal representation term. Tables 3, 4, and 5 demonstrate that the proposed KBD-TSC method and IBDLR outperform other compared algorithms on all datasets. This underscores the effectiveness of methods incorporating the block diagonal representation term, particularly for datasets with multiple classes. Remarkably, in the case of COIL20 and BA datasets characterized by a larger number of instances, the proposed method demonstrates a performance improvement of almost 15% compared to alternative methods, with the exception of IBDLR on the COIL20 dataset. Additionally, the suggested KBD-TSC method surpasses the performance of

IBDLR specifically on the COIL20 dataset, underscoring the advantages of incorporating a similarity-preserving strategy in the transform domain.

6. In datasets with high-dimensional features such as TR11, TR41, and TR45, SLSPr demonstrates superior performance to IBDLR, attributed to its integration of a similarity-preserving mechanism. Capitalizing on both the similarity-preserving strategy and the block diagonal representation term, the suggested KBD-TSC consistently outshines IBDLR and even surpasses SLSPr in most instances across TR11, TR41, and TR45 datasets. These outcomes underscore the effectiveness of the proposed KBD-TSC method in effectively managing datasets with intricate features, enabling the extraction of inherent data structures.

In a nutshell, the experimental results demonstrate the effectiveness of our proposed KBD-TSC method combined with similarity preserving regularizer, transform learning-based kernel self-expressing model, and block diagonal representation term.

4.8 Convergence analysis

The convergence plots of the proposed method is shown in Fig. 1. For all the datasets, the proposed method converges within 10 iterations.

4.9 Computational time

The experiments are conducted on a 64-bit Windows system with Intel i7 processor and 32GB RAM. The running time of the proposed method and the various state-of-the-art methods for all the datasets are shown in Table 6. From Table 6, it can be observed that the proposed KBD-TSC method is the fastest among all kernel-based techniques.

4.10 Ablation experiments

For the ablation experiments, we have compared the NMI score of the proposed technique against the objective function without the similarity preserving term and the objective function without the block diagonal term. It is observed that these terms are important for making clustering efficient. The NMI score gives the best value for all the datasets when these terms are included in the objective function. The ablation experiment results are discussed in Table 7. The objective function without the similarity preserving term is labelled as "similarity-" and the objective function without the block diagonal term is labelled as "block-."

5 Conclusion

This paper presents a novel subspace clustering approach that integrates transform learning-based kernel block diagonal representation and a similarity-preserving strategy. The method exhibits effective performance even when the raw data lacks inherent separability into subspaces and demonstrates robust generalization capabilities for non-linear manifolds. The proposed KBD-TSC operates through a three-step process. Initially, it captures the non-linear structure of the input data by incorporating the kernel self-expressing framework into the transform learning-based framework. The second step introduces the block diagonal representation term to create a similarity matrix with a block diagonal structure. In the final step, the similarity-preserving term is introduced to capture pairwise similarity information between various data points. The effectiveness of the proposed approach is evaluated on nine benchmark datasets, showcasing its superiority over several state-of-the-art methods. In future work, we aim to extend the proposed method to multiple kernel learning.

References

1. Bai L, Liang J (2020) Sparse subspace clustering with entropy-norm. ICML'20. JMLR.org
2. Chen H, Wang W, Feng X, He R (2018) Discriminative and coherent subspace clustering. Neurocomputing 284:177–186. https://doi.org/10.1016/j.neucom.2018.01.006
3. Chen J, Zhang H, Mao H, Sang Y, Yi Z (2016) Symmetric low-rank representation for subspace clustering. Neurocomputing 173:1192–1202. https://doi.org/10.1016/j.neucom.2015.08.077
4. Chen Y, Gu Y (2017) Active orthogonal matching pursuit for sparse subspace clustering. IEEE Signal Process Lett. https://doi.org/10.1109/LSP.2017.2741509
5. Cheng W, Chow TW, Zhao M (2016) Locality constrained-lp sparse subspace clustering for image clustering. Neurocomputing 205:22–31. https://doi.org/10.1016/j.neucom.2016.04.010
6. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans Pattern Anal Mach Intell 35:2765–2781. https://doi.org/10.1109/TPAMI.2013.57
7. Fan K (1949) On a theorem of Weyl concerning eigenvalues of linear transformations I. Proc Nat Acad Sci 35:652–655. https://doi.org/10.1073/pnas.35.11.652
8. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24:381–395. https://doi.org/10.1145/358669.358692
9. Goel A, Majumdar A (2021) Sparse subspace clustering friendly deep dictionary learning for hyperspectral image classification. IEEE Geosci Remote Sens Lett 19:1–5. https://doi.org/10.1109/LGRS.2021.3112603
10. Goel A, Majumdar A (2024) Sparse subspace clustering incorporated deep convolutional transform learning for hyperspectral band selection. Earth Sci Inf 17:2727–2735. https://doi.org/10.1007/s12145-024-01312-8
11. Gupta P, Goel A, Majumdar A, Chouzenoux E, Chierchia G (2024) Deconfcluster: Deep convolutional transform learning based

multiview clustering fusion framework. Sig Process 224:109597. https://doi.org/10.1016/j.sigpro.2024.109597

12. He R, Wang L, Sun Z, Zhang Y, Li B (2015) Information theoretic subspace clustering. IEEE Trans Neural Netw Learn Syst 27:2643–2655. https://doi.org/10.1109/TNNLS.2015.2500600
13. Ho J, Yang M-H, Lim J, Lee K-C, Kriegman D (2003) Clustering appearances of objects under varying illumination conditions. In: Proceedings of the 2003 IEEE computer society conference on computer vision and pattern recognition, 2003. IEEE, vol 1, p I. https://doi.org/10.1109/CVPR.2003.1211332
14. Hong W, Wright J, Huang K, Ma Y (2006) Multiscale hybrid linear models for lossy image representation. IEEE Trans Image Process 15:3655–3671. https://doi.org/10.1109/TIP.2006.882016
15. Huang Q, Zhang Y, Peng H, Dan T, Weng W, Cai H (2020) Deep subspace clustering to achieve jointly latent feature extraction and discriminative learning. Neurocomputing 404:340–350. https://doi.org/10.1016/j.neucom.2020.04.120
16. Ji P, Reid I, Garg R, Li H, Salzmann M (2017) Adaptive low-rank kernel subspace clustering. arXiv preprint arXiv:1707.04974. https://doi.org/10.48550/arXiv.1707.04974
17. Ji P, Zhang T, Li H, Salzmann M, Reid I (2017) Deep subspace clustering networks. In: Proceedings of the 31st international conference on neural information processing systems NIPS'17. Curran Associates Inc, Red Hook, pp 23–32
18. Ji P, Zhang T, Li H, Salzmann M, Reid I (2017) Deep subspace clustering networks. Adv Neural Inf Process Syst 30:5509–5521
19. Kang Z, Lu X, Lu Y, Peng C, Chen W, Xu Z (2020) Structure learning with similarity preserving. Neural Netw 129:138–148. https://doi.org/10.1016/j.neunet.2020.05.030
20. Kang Z, Lu Y, Su Y, Li C, Xu Z (2019) Similarity learning via kernel preserving embedding. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 4057–4064. https://doi.org/10.1609/aaai.v33i01.33014057
21. Kang Z, Peng C, Cheng Q (2017) Kernel-driven similarity learning. Neurocomputing 267:210–219. https://doi.org/10.1016/j.neucom.2017.06.005
22. Kelkar BA, Rodd SF, Kulkarni UP (2019) Estimating distance threshold for greedy subspace clustering. Expert Syst Appl 135:219–236. https://doi.org/10.1016/j.eswa.2019.06.011
23. Li C-G, You C, Vidal R (2017) Structured sparse subspace clustering: a joint affinity learning and subspace clustering framework. IEEE Trans Image Process 26:2988–3001. https://doi.org/10.1109/TIP.2017.2691557
24. Liao M, Li Y, Gao M (2022) Graph-based adaptive and discriminative subspace learning for face image clustering. Expert Syst Appl 192:116359. https://doi.org/10.1016/j.eswa.2021.116359
25. Lin Z, Chen M, Ma Y (2010) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055. https://doi.org/10.48550/arXiv.1009.5055
26. Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2012) Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 35:171–184. https://doi.org/10.1109/TPAMI.2012.88
27. Liu M, Wang Y, Sun J, Ji Z (2022) Adaptive low-rank kernel block diagonal representation subspace clustering. Appl Intell 52:2301–2316. https://doi.org/10.1007/s10489-021-02396-1
28. Liu Z, Ou W, Zhang K, Xiong H (2024) Robust manifold discriminative distribution adaptation for transfer subspace learning. Expert Syst Appl 238:122117. https://doi.org/10.1016/j.eswa.2023.122117
29. Lu C, Feng J, Lin Z, Mei T, Yan S (2018) Subspace clustering by block diagonal representation. IEEE Trans Pattern Anal Mach Intell 41:487–501. https://doi.org/10.1109/TPAMI.2018.2794348
30. Lu C, Feng J, Lin Z, Mei T, Yan S (2019) Subspace clustering by block diagonal representation. IEEE Trans Pattern Anal Mach Intell 41:487–501. https://doi.org/10.1109/TPAMI.2018.2794348
31. Maggu J, Majumdar A (2016) Alternate formulation for transform learning. In: Proceedings of the tenth Indian conference on computer vision, graphics and image processing, pp 1–8. https://doi.org/10.1145/3009977.3010069
32. Maggu J, Majumdar A (2017) Kernel transform learning. Pattern Recogn Lett 98:117–122. https://doi.org/10.1016/j.patrec.2017.09.002
33. Maggu J, Majumdar A, Chouzenoux E (2018) Transformed locally linear manifold clustering. In: 2018 26th European signal processing conference (EUSIPCO), pp 1057–1061. https://doi.org/10.23919/EUSIPCO.2018.8553061
34. Maggu J, Majumdar A, Chouzenoux E (2021) Transformed subspace clustering. IEEE Trans Knowl Data Eng 33:1796–1801. https://doi.org/10.1109/TKDE.2020.2969354
35. Maggu J, Majumdar A, Chouzenoux E, Chierchia G (2020) Deeply transformed subspace clustering. Signal Process 174:107628. https://doi.org/10.1016/j.sigpro.2020.107628
36. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 14:256
37. Nguyen H, Yang W, Shen F, Sun C (2015) Kernel low-rank representation for face recognition. Neurocomputing 155:32–42. https://doi.org/10.1016/j.neucom.2014.12.051
38. Patel VM, Van Nguyen H, Vidal R (2013) Latent space sparse subspace clustering. In: Proceedings of the IEEE international conference on computer vision, pp 225–232
39. Patel VM, Van Nguyen H, Vidal R (2015) Latent space sparse and low-rank subspace clustering. IEEE J Sel Top Signal Process 9:691–701. https://doi.org/10.1109/JSTSP.2015.2402643
40. Patel VM, Vidal R (2014) Kernel sparse subspace clustering. In: 2014 IEEE international conference on image processing (ICIP), pp 2849–2853. IEEE. https://doi.org/10.1109/ICIP.2014.7025576
41. Paul D, Saha S, Mathew J (2020) Improved subspace clustering algorithm using multi-objective framework and subspace optimization. Expert Syst Appl 158:113487. https://doi.org/10.1016/j.eswa.2020.113487
42. Peng X, Feng J, Zhou JT, Lei Y, Yan S (2020) Deep subspace clustering. IEEE Trans Neural Netw Learn Syst 31:5509–5521. https://doi.org/10.1109/TNNLS.2020.2968848
43. Peng X, Xiao S, Feng J, Yau W-Y, Yi Z (2016) Deep subspace clustering with sparsity prior. In: IJCAI, pp 1925–1931
44. Pham D-S, Budhaditya S, Phung D, Venkatesh S (2012) Improved subspace clustering via exploitation of spatial constraints. In: 2012 IEEE conference on computer vision and pattern recognition, pp 550–557. IEEE. https://doi.org/10.1109/CVPR.2012.6247720
45. Qin Y, Pu N, Wu H (2023) EDMC: efficient multi-view clustering via cluster and instance space learning. IEEE Trans Multimedia. https://doi.org/10.1109/TMM.2023.3331197
46. Qin Y, Pu N, Wu H (2023) Elastic multi-view subspace clustering with pairwise and high-order correlations. IEEE Trans Knowl Data Eng
47. Qin Y, Tang Z, Wu H, Feng G (2023) Flexible tensor learning for multi-view clustering with Markov chain. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2023.3305624
48. Ravishankar S, Bresler Y (2013) Closed-form solutions within sparsifying transform learning. In: 2013 IEEE international conference on acoustics, speech and signal processing, pp 5378–5382. IEEE. https://doi.org/10.1109/ICASSP.2013.6638690
49. Ravishankar S, Bresler Y (2013) Learning sparsifying transforms. IEEE Trans Signal Process 61:1072–1086. https://doi.org/10.1109/TSP.2012.2226449
50. Ravishankar S, Bresler Y (2015) Online sparsifying transform learning—Part II. IEEE J Sel Top Signal Process 9:637–646. https://doi.org/10.1109/JSTSP.2015.2407860

51. Ravishankar S, Wen B, Bresler Y (2015) Online sparsifying transform learning—Part I. IEEE J Sel Top Signal Process 9:625–636. https://doi.org/10.1109/JSTSP.2015.2417131
52. Seung HS, Lee DD (2000) The manifold ways of perception. Science 290:2268–2269. https://doi.org/10.1126/science.290.5500.2268
53. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22:888–905. https://doi.org/10.1109/34.868688
54. Soltanolkotabi M, Candés EJ (2012) A geometric analysis of subspace clustering with outliers. Ann Stat 40:2195–2238. https://doi.org/10.1214/12-AOS1034
55. Somandepalli K, Narayanan S (2019) Reinforcing self-expressive representation with constraint propagation for face clustering in movies. In: ICASSP 2019—2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 4065–4069. https://doi.org/10.1109/ICASSP.2019.8682314
56. Song J, Yoon G, Hahn K, Yoon SM (2019) Subspace clustering via structure-enforced dictionary learning. Neurocomputing 362:1–10. https://doi.org/10.1016/j.neucom.2019.07.025
57. Tang K, Xu K, Jiang W, Su Z, Sun X, Luo X (2022) Selecting the best part from multiple Laplacian autoencoders for multi-view subspace clustering. IEEE Trans Knowl Data Eng 35:7457–7469. https://doi.org/10.1109/TKDE.2022.3178145
58. Tang K, Xu K, Su Z, Zhang N (2023) Multi-view subspace clustering via consistent and diverse deep latent representations. Inf Sci 651:119719. https://doi.org/10.1016/J.INS.2023.119719
59. Tschannen M, Bölcskei H (2018) Noisy subspace clustering via matching pursuits. IEEE Trans Inf Theory 64:4081–4104. https://doi.org/10.1109/TIT.2018.2812824
60. Vidal R (2011) Subspace clustering. Sig Process Mag IEEE 28:52–68. https://doi.org/10.1109/MSP.2010.939739
61. Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). Pattern Recogn Lett 43:47–61. https://doi.org/10.1016/j.patrec.2013.08.006
62. Vidal R, Ma Y, Sastry S (2005) Generalized principal component analysis (GPCA). IEEE Trans Pattern Anal Mach Intell 27:1945–1959. https://doi.org/10.1109/TPAMI.2005.244
63. Wang J, Shi D, Cheng D, Zhang Y, Gao J (2016) LRSR: low-rank-sparse representation for subspace clustering. Neurocomputing 214:1026–1037. https://doi.org/10.1016/j.neucom.2016.07.015
64. Waqas M, Tahir MA, Khan SA (2023) Robust bag classification approach for multi-instance learning via subspace fuzzy clustering. Expert Syst Appl 214:119113. https://doi.org/10.1016/j.eswa.2022.119113
65. Xia G, Sun H, Feng L, Zhang G, Liu Y (2017) Human motion segmentation via robust kernel sparse subspace clustering. IEEE Trans Image Process 27:135–150. https://doi.org/10.1109/TIP.2017.2738562
66. Xiao S, Tan M, Xu D, Dong ZY (2015) Robust kernel low-rank representation. IEEE Trans Neural Netw Learn Syst 27:2268–2281. https://doi.org/10.1109/TNNLS.2015.2472284
67. Xie X, Guo X, Liu G, Wang J (2017) Implicit block diagonal low-rank representation. IEEE Trans Image Process 27:477–489. https://doi.org/10.1109/TIP.2017.2764262
68. Yang Y, Li F (2023) Kernel block diagonal representation subspace clustering with similarity preservation. Appl Sci 13:9345. https://doi.org/10.3390/app13169345
69. Yang Z, Oja E (2010) Linear and nonlinear projective nonnegative matrix factorization. IEEE Trans Neural Netw 21:734–749. https://doi.org/10.1109/TNN.2010.2041361
70. Yin M, Liu W, Li M, Jin T, Ji R (2021) Cauchy loss induced block diagonal representation for robust multi-view subspace clustering. Neurocomputing 427:84–95. https://doi.org/10.1016/j.neucom.2020.11.017
71. You C, Robinson D, Vidal R (2016) Scalable sparse subspace clustering by orthogonal matching pursuit, pp 3918–3927. https://doi.org/10.1109/CVPR.2016.425
72. Yu S, Yiquan W (2018) Subspace clustering based on latent low rank representation with frobenius norm minimization. Neurocomputing 275:2479–2489. https://doi.org/10.1016/j.neucom.2017.11.021
73. Zhai H, Zhang H, Zhang L, Li P, Plaza A (2016) A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery. IEEE Geosci Remote Sens Lett 14:43–47. https://doi.org/10.1109/LGRS.2016.2625200
74. Zhang C, Li H, Chen C, Jia X, Chen C (2022) Low-rank tensor regularized views recovery for incomplete multiview clustering. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3232538
75. Zhang C, Li H, Chen C, Qian Y, Zhou X (2020) Enhanced group sparse regularized nonconvex regression for face recognition. IEEE Trans Pattern Anal Mach Intell 44:2438–2452. https://doi.org/10.1109/TPAMI.2020.3033994
76. Zhang C, Li H, Lv W, Huang Z, Gao Y, Chen C (2023) Enhanced tensor low-rank and sparse representation recovery for incomplete multi-view clustering. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, pp 11174–11182. https://doi.org/10.1609/aaai.v37i9.26323
77. Zhang H, Lin Z, Zhang C, Gao J (2014) Robust latent low rank representation for subspace clustering. Neurocomputing 145:369–373. https://doi.org/10.1016/j.neucom.2014.05.022
78. Zhou L, Xiao B, Liu X, Zhou J, Hancock ER et al (2019) Latent distribution preserving deep subspace clustering. In: 28th international joint conference on artificial intelligence. York, pp 4440–4446. https://doi.org/10.24963/ijcai.2019/617

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.