Integrate and Conquer: Double-Sided Two-Dimensional k -Means Via Integrating of Projection and Manifold Construction

CHONG PENG, Qingdao University

ZHAO KANG, University of Electronic Science and Technology of China

SHUTING CAI, Guangdong University of Technology

QIANG CHENG, University of Kentucky

In this article, we introduce a novel, general methodology, called integrate and conquer, for simultaneously accomplishing the tasks of feature extraction, manifold construction, and clustering, which is taken to be superior to building a clustering method as a single task. When the proposed novel methodology is used on two-dimensional (2D) data, it naturally induces a new clustering method highly effective on 2D data. Existing clustering algorithms usually need to convert 2D data to vectors in a preprocessing step, which, unfortunately, severely damages 2D spatial information and omits inherent structures and correlations in the original data. The induced new clustering method can overcome the matrix-vectorization-related issues to enhance the clustering performance on 2D matrices. More specifically, the proposed methodology mutually enhances three tasks of finding subspaces, learning manifolds, and constructing data representation in a seamlessly integrated fashion. When used on 2D data, we seek two projection matrices with optimal numbers of directions to project the data into low-rank, noise-mitigated, and the most expressive subspaces, in which manifolds are adaptively updated according to the projections, and new data representation is built with respect to the projected data by accounting for nonlinearity via adaptive manifolds. Consequently, the learned subspaces and manifolds are clean and intrinsic, and the new data representation is discriminative and robust. Extensive experiments have been conducted and the results confirm the effectiveness of the proposed methodology and algorithm.

CCS Concepts: • **Information systems** → **Clustering**; *Clustering and classification*; • **Theory of computation** → **Unsupervised learning and clustering**; • **Computing methodologies** → **Cluster analysis**; **Dimensionality reduction and manifold learning**;

This work was supported by the National Science Foundation under grant IIS-1218712; the National Natural Science Foundation of China under grant 61201392; the Science and Technology Planning Project of Guangdong Province, China, under grant 2017B090909004; a research project by Shanxi Scholarship Council of China under grant 2015-093; the Fundamental Research Fund for the Central Universities of China under grant ZYGX2017KYQD177; and the Foundation Program of Yuncheng University under grants SWSX201603 and YQ-2012020.

Authors' addresses: C. Peng, College of Computer Science and Technology, Qingdao University, 308 Ningxia Road, Qingdao, Shandong 266071, China, and Department of Computer Science, Southern Illinois University Carbondale, 1263 Lincoln Drive, Carbondale, IL 62901; email: pchong1991@163.com; Z. Kang, School of Computer Science and Engineering, University of Electronic Science and Technology of China, 4 North Jianshe Road, Chengdu, Sichuan 611731, China; email: skangz@gmail.com; S. Cai (corresponding author), School of Automation, Guangdong University of Technology, 100 West Waihuan Road, University Town, Guangzhou, Guangdong 510006, China; email: shutingcai@126.com; Q. Cheng, Institute of Biomedical Informatics and Department of Computer Science, University of Kentucky, 725 Rose Street, Lexington, KY 40536; email: Qiang.Cheng@uky.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 ACM 2157-6904/2018/05-ART57 \$15.00

<https://doi.org/10.1145/3200488>

Additional Key Words and Phrases: Clustering, unsupervised learning, two-dimensional data, feature extraction

ACM Reference format:

Chong Peng, Zhao Kang, Shuting Cai, and Qiang Cheng. 2018. Integrate and Conquer: Double-Sided Two-Dimensional k -Means Via Integrating of Projection and Manifold Construction. *ACM Trans. Intell. Syst. Technol.* 9, 5, Article 57 (May 2018), 25 pages.

<https://doi.org/10.1145/3200488>

1 INTRODUCTION

Clustering is one of the fundamental unsupervised learning techniques in machine learning and data mining with broad applications such as image segmentation (Shi and Malik 2000) and text clustering (Hotho et al. 2003). It refers to the process of partitioning a given dataset into different groups such that data points in the same group, known as a cluster, have higher similarities than those from different groups. In the past decades, a variety of clustering algorithms have been proposed such as spectral clustering (Ng et al. 2002) and nonnegative matrix factorization (NMF) (Lee and Seung 1999). Among all clustering methods, k -means is perhaps the most widely used owing to its simplicity and effectiveness (Wu et al. 2008), and many existing methods are either k -means based or its direct extensions. For example, it is usually employed as a final step for NMF-based methods (Cai et al. 2011) or spectral clustering-based methods (Liu and Yan 2011; Elhamifar and Vidal 2009; Peng et al. 2015).

k -means seeks the centroids and indicators of clusters by minimizing the total distance from the data points to the centroids in the Euclidean space. It is well known that k -means suffers from two primary drawbacks: First, it measures the distance with the squared Euclidean distance, which is known to be sensitive to outliers and gross corruptions; second, it only considers linear relationships of the data in the Euclidean space, and thus it cannot identify arbitrarily shaped clusters which often exist in real-world data. To mitigate the first drawback, more robust k -means variants have been developed. For example, the multi-view k -means (Cai et al. 2013) uses more robust $\ell_{2,1}$ norm to measure the total distances. The second drawback is due to the lack of a capability of capturing nonlinear structures of data, as a remedy kernel trick is often employed to explore potentially nonlinear structures in a high-dimensional feature space (Schiolkopf 2001; Kang et al. 2017). Two typical kernel-based clustering methods are kernel k -means (KKM) (Schölkopf et al. 1998) and spectral clustering (SC) (Ng et al. 2002). The key difference between KKM and SC is that the former maps data points into a higher-dimensional feature space where linear separators may better partition mapped data points, whereas the latter uses an affinity matrix to account for non-linearity and its eigenvectors to yield a clustering. Further study has revealed that SC is equivalent to a weighted version of kernel k -means (Dhillon et al. 2007) and both of them are particular cases of clustering with NMF under a doubly stochastic constraint (Zass and Shashua 2005).

While kernel-based methods have seen numerous successful applications, in real world data irrelevant or noisy features often exist, and thus the constructed kernel or affinity matrix may be highly noisy, leading to severely degraded learning performance with kernel-based clustering. As a mitigation, feature selection (Mitra et al. 2002) has been considered for identifying the most relevant features and reducing the dimensionality of the data. Usually, feature selection is employed as a preprocessing step, which is independent of further data processing. However, a two-step procedure might not be able to provide a satisfactory solution to many problems. For example, tandem clustering (Arabie 1994) is a two-step approach, which performs principal component analysis (PCA) (Jolliffe 2002) in the first step to obtain the principal components of the data and, subsequently, perform k -means clustering on a few top principal components in the second step to obtain the clusters. Although PCA is one of the most successful dimension reduction techniques,

its effect on clustering and this two-step procedure has been controversial (Chang 1983; Arabie 1994). Moreover, it is noted (Peng et al. 2016) that unsupervised feature selection methods are often transformed into supervised ones by the majority of existing unsupervised feature selection methods. For example, multi-class feature selection (MCFS) (Cai et al. 2010) requires clustering for preprocessing such that unsupervised feature selection is transformed into a supervised one equipped with the learned cluster label information. It may be inappropriate for highly noisy data to undergo such a two-step procedure, because the step of its feature selection heavily depends on the cluster label information which might be far from being accurate.

Recently, a clustering method with seamlessly integrated feature selection has been proposed and demonstrated highly promising results (Peng et al. 2016), where a binary feature selector is used to select important features. Besides feature selection, feature extraction or projection has also been used for clustering. For example, Patel et al. (2013) reduces the dimension of input data with an embedded projection; Allab et al. (2015) simultaneously performs semi-NMF and PCA to identify a partition of the data by seeking an optimal subspace of multi-dimensional variables, which provides simpler and more interpretable solutions with a low-dimensional representation. Since feature selection or extraction is performed simultaneously with clustering, the above-mentioned potential problems of the two-step approach are alleviated in Peng et al. (2016), Patel et al. (2013), and Allab et al. (2015), which require no clustering preprocessing.

The above-mentioned clustering methods have been used primarily for one-dimensional (1D) and two-dimensional (2D) data. 2D data (e.g., images) have become increasingly common in many real-world applications such as multi-label learning (Zhang and Zhou 2014), matrix completion (Hu et al. 2013), image denoising (Buades et al. 2005), foreground-background separation (Candès et al. 2011), and others (Alpert et al. 2012; Shi and Malik 2000). When clustering 2D data, existing methods usually convert 2D matrices into vectors with a preprocessing step, which often severely damages 2D spatial information. Throughout this article, by 2D spatial information, we refer to potential correlation between the elements of a local neighborhood in a 2D matrix. “Spatial” refers to being related to positions of matrix elements. Such information is important, and the loss of it may severely impair the learning performance of any clustering method on 2D data. Inspired by PCA and its variant of 2D principal component analysis (2DPCA) (Yang et al. 2004), in this article, we are to directly use 2D matrices for clustering, such that 2D spatial information from the original 2D data is retained to help boost learning performance.

More specifically, we introduce a new methodology that seeks projection matrices to directly operate on 2D data to capture the most expressive 2D spatial information and facilitate the recovery of the original 2D data in the projected subspaces with the most variational information retained. Our new methodology achieves the goal of clustering simultaneously with seeking projections by maximizing the total scatter of the projected 2D matrices and minimizing the total distance from original 2D matrices to their corresponding centroids in the projected subspace. The proposed methodology is robust to noise and corruptions because the reconstructed matrices in the projected subspace, which contain the most variational information and are less afflicted with corruptions, are used for clustering.

Moreover, to enhance the capability of recovering nonlinear structures of the 2D data, a new way to manifold learning is incorporated as an integral part of our new methodology. Unlike existing methods that use the original data points to construct a graph Laplacian, we use the reconstructed 2D data in the projected subspace to construct the graph Laplacian, which is adaptive to the projection and the 2D spatial information. Consequently, the constructed graph Laplacian is much cleaner and more capable of representing essential data relationships. This new way of constructing adaptive, clean manifolds is inspired by our key observation that intrinsic manifolds can be better learned in the latent space induced, for example, by NMF in our previous work (Peng et al. 2017b, 2017c).

In all, it is our key observation that the three tasks of feature extraction, clustering, and manifold construction are strongly coupled and promote each other, and thus, it is our belief that simultaneously solving these tasks is the best way to accomplish any single one of the tasks. Hence, we propose a novel, general methodology, called integrate and conquer, for simultaneous learning of projection matrices, manifolds, and representation, which naturally induces a new clustering method when used on 2D data.

The key contributions of this article are summarized as follows:

- A novel methodology, integrate and conquer, is introduced to promote simultaneous learning of optimal projections, adaptive manifolds, and data representation. The simultaneous learning is motivated by our key observation that the three tasks of feature extraction, clustering, and manifold construction are strongly coupled and boost each other, and thus, in our belief, it is the best way to achieve any individual one of the three tasks.
- A new model is developed to realize the new methodology of simultaneous learning. It enables simultaneous learning of subspaces, adaptive manifolds, and data representation, offering a strong ability to capture the intrinsic structures and information of 2D data. As a special case of 2D matrices, our model is also applicable to vectorial data.
- Projections with optimal numbers of directions are found and the optimal low-rank subspaces are obtained to retain 2D information and enhance clustering performance.
- Data representations are built in the low-dimensional, most expressive subspaces, with which the new model retains intrinsic spatial information and alleviates the adverse effects of irrelevant, corrupted, or less important information, rendering our model to possess strong robustness.
- Adaptive manifold learning is introduced into the model to enhance the capability of exploiting nonlinear structures of the data, enabling the learned graph to be less afflicted with irrelevant or grossly corrupted features.
- Extensive experiments have been performed and the results testify to the effectiveness of the proposed algorithm for clustering as well as the novel methodology.

The rest of this article is organized as follows. Some closely related work is reviewed in Section 2. We introduce the proposed methodology and the induced model on 2D data in Section 3 and analyze its connections with several well-known methods in Section 4. Then we discuss the optimization of the proposed model in Section 5. The experimental results are presented in Section 6. Finally, Section 7 concludes the article.

2 RELATED WORK

In this section, we will briefly introduce several well-known clustering and dimension reduction methods that are closely related to our work.

2.1 *k*-Means

Let $Y = [y_1, y_2, \dots, y_n] \in \mathcal{R}^{d \times n}$ be a data matrix, where y_i are samples, and d is their dimension. To partition the data points into k clusters, k -means minimizes the total distance from data points to the centroids, $u_j \in \mathcal{R}^d$, by

$$\min_{U, V} \left\| y_i - \sum_{j=1}^k u_j v_{ji} \right\|_F^2 \quad s.t. \quad v_{ij} \in \{0, 1\}, \sum_{i=1}^k v_{ij} = 1, \quad (1)$$

where $V = [v_{ij}]$ is the cluster indicator matrix and $U = [u_1, u_2, \dots, u_k] \in \mathcal{R}^{d \times k}$ is the latent cluster centroid matrix.

2.2 Graph Laplacian

Nonlinear structures of the data often exist and are important in real-world applications. The graph Laplacian (Chung 1997) is a widely used technique to incorporate the intrinsic geometrical structure of the data. In particular, minimizing the manifold in Equation (2) enforces the smoothness of the data in nonlinear spaces:

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \|v_i - v_j\|_2^2 w_{ij} &= \sum_{j=1}^n d_{jj} v_j^T v_j - \sum_{i=1}^r \sum_{j=1}^n w_{ij} v_i^T v_j \\ &= \text{Tr}(VDV^T) - \text{Tr}(VWV^T) = \text{Tr}(VLV^T), \end{aligned} \quad (2)$$

where $\text{Tr}(\cdot)$ is the trace operator, $W = [w_{ij}]$ is the weight matrix that measures the pairwise similarities of original data points, $D = [d_{ij}]$ is a diagonal matrix with $d_{ii} = \sum_j w_{ij}$, and $L = D - W$ is the graph Laplacian. It is seen that by minimizing Equation (2), we get a natural effect that if two data points are close in the intrinsic geometry of the data distribution, then their new representations with respect to the new basis, v_i and v_j , are also close (Cai et al. 2011).

2.3 Spectral Clustering

With the graph Laplacian being the objective function, SC solves the following optimization problem:

$$\min_{VV^T = I_k, v_{ij} \geq 0} \text{Tr}(VLV^T). \quad (3)$$

Depending on whether L is normalized, two different versions of spectral clustering have been proposed (Shi and Malik 2000; Ng et al. 2002), where a detailed discussion about them in terms of graph theory and perturbation theory can be found in Von Luxburg (2007).

2.4 PCA and 2DPCA

For centered data $Y \in \mathcal{R}^{d \times n}$, PCA finds the optimal low-dimensional subspace spanned by the principal directions that are contained in $\mathbf{P} \in \mathcal{R}^{d \times r}$, with r being the desired dimension of the subspace. P can be obtained by minimizing

$$\min_{\mathbf{P}^T \mathbf{P} = I_r} \|\mathbf{P} \mathbf{P}^T Y - Y\|_F^2. \quad (4)$$

It is seen that Equation (4) can be rewritten as

$$\max_{\mathbf{P}^T \mathbf{P} = I_r} \mathbf{P}^T G \mathbf{P}, \quad (5)$$

where $G = YY^T$ is the covariance matrix of Y . Hence, the optimal projection matrix $\mathbf{P}$ can be obtained from the eigenvalue decomposition of G . For centered two-dimensional data $\{X_i \in \mathcal{R}^{a \times b}\}_{i=1}^n$, the 2D covariance matrix can be defined with $G' = \sum_{i=1}^n X_i^T X_i$. In analogy to Equation (5), 2DPCA seeks the optimal projection matrices $\mathbf{P}' \in \mathcal{R}^{b \times b}$ by minimizing

$$\max_{\mathbf{P}'^T \mathbf{P}' = I_b} \mathbf{P}'^T G' \mathbf{P}'. \quad (6)$$

3 PROPOSED NOVEL METHODOLOGY

3.1 Motivation for Our Novel Methodology

The three tasks of feature or subspace extraction, manifold construction, and clustering are closely coupled and capable of boosting each other. For example, our previous work has shown and exploited the coupling between clustering and feature extraction or engineering (Peng et al. 2016,

2017b), between manifold construction and clustering (Peng et al. 2017b), and between subspace extraction and manifold clustering (Peng et al. 2017a). However, in the literature, there appears to lack work that effectively considers the couplings between the three tasks and solves them simultaneously. Based on our previous work and key observation of the strong coupling of these tasks, we believe that the best way to achieve any one of the three tasks is to simultaneously learn all of them. Hence, we propose a novel methodology of simultaneous learning of feature or subspace extraction, manifold construction, and clustering. In the following, for ease of composition, we will explain the use of our novel integrate and conquer methodology on 2D data, which induces a new clustering model and algorithm.

3.2 Development of New Clustering Model Using Our Methodology

Given examples $X_i \in \mathcal{R}^{a \times b}$, $i = 1, 2, \dots, n$, each of which being perhaps an above-mentioned user-item-time structure or a gray image, we form a collection of them denoted by $\mathbf{X} = \{X_i\}_{i=1}^n$. Each example, being a matrix itself, is regarded as a point in its corresponding space. In the following, we will develop a clustering algorithm that projects such 2D data points into a low-dimensional and expressive subspace, and simultaneously finds 2D centroids and indicators in that subspace. To learn the projection and new representation jointly, we propose to solve the following optimization problem:

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{P}, \mathbf{V}} \sum_{i=1}^n \left\{ \left\| X_i P - \sum_{j=1}^k C_j v_{ji} \right\|_F^2 + \left\| X_i - X_i P P^T \right\|_F^2 \right\} \\ \text{s.t. } P^T P = I_r, V V^T = I_k, v_{ij} \geq 0, \end{aligned} \quad (7)$$

where $\mathbf{C} = \{C_j \in \mathcal{R}^{a \times r}\}_{j=1}^k$ is a set of 2D centroids, $\mathbf{V} = [v_{ij}]$ is a relaxed class indicator matrix with each element relaxed to a real, instead of binary, value, $P \in \mathcal{R}^{b \times r}$ is an orthogonal matrix, and I_r is an identity matrix of size $r \times r$. It is noted that $P P^T$ is a mathematical projection matrix, as $(P P^T)^2 = P P^T$. In the summation of Equation (7), the second term ensures that the projection captures the most expressive information from the original 2D matrices, while the first term seeks a new data representation with class indicators and 2D centroids in the P -transformed space.

In Equation (7) the data representation is sought in a much lower-dimensional space than the original with $r \ll b$, and the centroids are obtained from the P -transformed matrices. For the original 2D matrices, often they have physical meanings; however, after linear transformation by P the centroids are searched for in the latent subspace, which, unfortunately, would lose physical meanings. Also, directly seeking centroids and orthogonal transformation of the examples by Equation (7) appears hard in optimization. To address both issues, we find a simple yet effective way to reformulate Equation (7) into the following model:

$$\begin{aligned} \min_{\mathbf{U}, \mathbf{P}, \mathbf{V}} \sum_{i=1}^n \left\{ \left\| X_i P P^T - \sum_{j=1}^k U_j v_{ji} \right\|_F^2 + \left\| X_i - X_i P P^T \right\|_F^2 \right\} \\ \text{s.t. } P^T P = I_r, V V^T = I_k, v_{ij} \geq 0, \end{aligned} \quad (8)$$

where $\mathbf{U} = \{U_j \in \mathcal{R}^{a \times b}\}_{j=1}^k$ contains 2D centroids in the subspace projected by $P P^T$. It is observed that, given P and V , Equation (8) is minimized with $U_j = \sum_{i=1}^n X_i P P^T v_{ij} \in \text{span}(P P^T)$, indicating that the centroids are readily in the projected space. Equation (7) naturally leads to Equation (8). To see this, we first minimize Equation (7) with respect to C_j to get $C_j = \sum_{i=1}^n X_i P v_{ij}$; with such

C_j , we let $U_j = C_j P^T$; then it is seen that U_j minimizes Equation (8). Moreover, we have

$$\left\| X_i P - \sum_{j=1}^k C_j v_{ji} \right\|_F^2 = \left\| X_i P P^T - \sum_{j=1}^k C_j P^T v_{ji} \right\|_F^2 = \left\| X_i P P^T - \sum_{j=1}^k U_j v_{ji} \right\|_F^2, \quad (9)$$

which reveals the equivalence of Equation (7) and (8). Although Equation (8) seeks centroids in a higher-dimensional space than Equation (7), its optimization is straightforward as will be shown in Section 5.

Model 8 only considers linear structures of the 2D matrices in the projected space. It is necessary to take nonlinear structures into consideration, which often exist and are important in real-world applications. To this end, we enforce the smoothness between linear and nonlinear spaces of the projected 2D matrices on manifold:

$$\min_{U, P^T P = I_r, V V^T = I_k} \sum_{i=1}^n \left\| X_i P P^T - \sum_{j=1}^k U_j v_{ji} \right\|_F^2 - \lambda_1 \text{Tr}(P^T G_P P) + \lambda_2 \text{Tr}(V L_P V^T), \quad (10)$$

where $\lambda_1, \lambda_2 \geq 0$ are two balancing parameters, $G_P = \sum_{i=1}^n X_i^T X_i$, and L_P is the graph Laplacian constructed by Equation (2) using $\mathbf{X}^P = \{X_i P P^T\}_{i=1}^n$. In Equation (10), the second term is readily obtained from Equation (8), because $\sum_{i=1}^n \|X_i - X_i P P^T\|_F^2 = \sum_{i=1}^n \text{Tr}(X_i^T X_i - 2X_i^T X_i P P^T + P P^T X_i^T X_i P P^T) = c - \text{Tr}(P^T G_P P)$ with c being constant. We add a free parameter λ_1 to elevate the flexibility of Equation (10). In the first term, we use the squared Frobenius norm to measure the reconstruction errors of the new representation. In addition to facilitating optimization, our use of such a norm is also justifiable in terms of robustness, because the new representation (i.e., centroids and indicators) is in the most expressive space where the adverse effects from noisy and less informative features are attenuated to the maximum extent. Consequently, Equation (10) is inherently robust.

It is noted that Equation (10) only exploits the column space of the matrices by seeking P from the right side of $\mathbf{X}$. To further enhance the capability of Equation (10) for revealing inherent structures of the data, similar to P -formulation, we also seek another orthogonal matrix Q to exploit the row space of the data, giving rise to the following double-sided two-dimensional k -means model (DTKM):

$$\begin{aligned} \min_{U, P, Q, V, r_1, r_2} & \left\| \vec{\mathbf{v}}(\mathbf{X}^P) - \vec{\mathbf{v}}(\mathbf{U})V \right\|_F^2 + \left\| \vec{\mathbf{v}}(\mathbf{X}^Q) - \vec{\mathbf{v}}(\mathbf{U})V \right\|_F^2 \\ & - \lambda_1 \left(\text{Tr}(P^T G_P P) + \text{Tr}(Q^T G_Q Q) \right) + \lambda_2 \text{Tr}(V^T (L_P + L_Q) V) \\ \text{s.t. } & r_1 + r_2 = 2r, P^T P = I_{r_1}, Q^T Q = I_{r_2}, V V^T = I_k, v_{ij} \geq 0. \end{aligned} \quad (11)$$

Here, we define $\vec{\mathbf{v}}(\mathbf{M}) = [\vec{\mathbf{v}}(M_1), \dots, \vec{\mathbf{v}}(M_m)]$ as a new matrix by vectorizing each M_i as a column for a set of 2D matrices $\mathbf{M} = \{M_i\}_{i=1}^m$, $\mathbf{X}^Q = \{Q Q^T X_i\}_{i=1}^n$, $G_Q = \sum_{i=1}^n X_i X_i^T$, L_Q is a graph Laplacian constructed using $\mathbf{X}^Q$, r_1 and r_2 represent the numbers of projection directions in P and Q , respectively, and $2r$ is the total number of projection directions. It is worthwhile to point out that the first term of Equation (11) is equivalent to the first term of Equation (10) due to the property of the Frobenius norm. Similarly, the second term of Equation (11) is a concise expression of $\sum_{i=1}^n \|Q Q^T X_i - \sum_{j=1}^k U_j v_{ji}\|_F^2$, which is a Q -counterpart of the first term of Equation (10). It is noted that in Equation (11) r is given while r_1 and r_2 are variables to be optimized; hence, Equation (11) is able to optimally determine the numbers of projection directions and the ranks for projected

row and column spaces of the data. We will discuss the optimization of model Equation (11) in the sequel.

Discussion 3.1. In the literature, most of existing clustering methods are designed for vectors, i.e., 1D data. To the best of our knowledge, this article is first one to directly operate on 2D matrices, without the need of vectorizing them, for clustering. In Equation (11), it is seen that we use 2D matrices explicitly, such that 2D spatial structures of the matrices are well retained for further data analysis. The suitability of our method for 2D data will be clearly demonstrated in Section 6 with experimental evaluation.

Discussion 3.2. As special cases of 2D matrices, vectorial data are suited to our method as well. For higher-order data, with each sample being a higher-order tensor, it is difficult to directly apply our method since it is not straightforward to seek projections. Two possible ways of processing such data are suggested as follows. First, for T -order data with each sample being a T -order tensor, commonly used flattening and unfolding techniques can be employed to convert each sample into a 2D matrix, and then our method can be applied to converted 2D matrices. Second, for a T -order tensor $\mathcal{T}_{i_1, i_2, \dots, i_T}$ with $i_1, \dots, i_T$ being indexes of T dimensions, by varying the $\{s, t\}$ th dimensions while fixing the other $(T - 2)$ dimensions, we obtain a 2D matrix $\mathcal{T}_{i_1, \dots, i_{s-1}, i_{s+1}, \dots, i_{t-1}, i_{t+1}, \dots, i_T}$. Repeating this procedure for $\frac{T(T-1)}{2}$ times will yield $\frac{T(T-1)}{2}$ matrices, which correspond to all possible combinations of $\{s, t\}$ dimensions. For a given collection of T -order tensors, we can perform clustering for each pair of $\{s, t\}$ dimensions and obtain soft indicator matrices $V_{(s,t)}$ from clustering results. Subsequently, two strategies can be applied to obtain the final cluster labels. First, for each sample, a standard clustering method such as k -means can be applied to obtain the final clustering result. Second, ensemble clustering methods (Zheng et al. 2010) can be applied to the soft indicator matrices for final clustering result. In this article, we focus on building a basic algorithm for directly clustering 2D matrices without a need of converting them into vectors; potential extensions to higher-order tensor data will be a future line of research.

4 CONNECTIONS WITH OTHER METHODS

The proposed model Equation (11) is closely related to some existing work. In this section, we analyze its connections with other methods such as k -means, spectral clustering, and 2DPCA.

4.1 Connection with k -Means

With $\lambda_1 = \lambda_2 = 0$, and either $r_1 = b$ or $r_2 = a$, all features of the data are used and Equation (11) is reduced to

$$\min_{VV^T = I_k, v_{ij} \geq 0} \|\vec{v}(X) - \vec{v}(U)V\|_F^2. \quad (12)$$

It is straightforward to see that because of the constraints in Equation (12), each row of V contains exactly one nonzero element which must equal to 1. Hence, Equation (12) is equivalent to k -means shown in Equation (1).

4.2 Connection with Spectral Clustering

By setting $\lambda_1 = 0$, $\lambda_2 = \infty$, and either $r_1 = b$ or $r_2 = a$, Equation (11) is reduced to

$$\min_{VV^T = I_k, v_{ij} \geq 0} \text{Tr}(VLV^T), \quad (13)$$

where L is a graph Laplacian constructed from $\vec{v}(X)$. It is apparent that Equation (13) is the spectral clustering given in Equation (3).

4.3 Connection with 2DPCA

By setting $\lambda_1 = \infty$, $\lambda_2 = 0$, and $r_2 = 0$, Equation (11) is reduced to

$$\max_{P^T P = I_r} \text{Tr}(P^T G_P P). \quad (14)$$

For centered data, G_P is the 2D covariance matrix. Therefore, Equation (11) is reduced to 2DPCA. Similarly, by setting $\lambda_1 = \infty$, $\lambda_2 = 0$, and $r_1 = 0$, Equation (11) is reduced to alternative 2DPCA (Zhang and Zhou 2005). Furthermore, if $\lambda_1 = \infty$, $\lambda_2 = 0$, and $r_1, r_2 \neq 0$, then Equation (11) is reduced to

$$\max_{P^T P = I_{r_1}, Q^T Q = I_{r_2}, r_1 + r_2 = 2r} \text{Tr}(P^T G_P P) + \text{Tr}(Q^T G_Q Q), \quad (15)$$

which can be regarded as an incremental model of 2DPCA. Equation (15) works in both row and column spaces of 2D matrices. The difference between Equation (15) and Zhang and Zhou (2005) is that r_1 and r_2 are to be optimized in Equation (15) while being fixed in the latter.

5 OPTIMIZATION OF OUR MODEL

We introduce an auxiliary variable Z and obtain the augmented Lagrange function of Equation (11) as follows:

$$\begin{aligned} \min_{U, P, Q, V, Z, r_1, r_2} & \left\| \vec{v}(X^P) - \vec{v}(U)V \right\|_F^2 + \left\| \vec{v}(X^Q) - \vec{v}(U)V \right\|_F^2 + \lambda_2 \text{Tr}(Z(L_P + L_Q)V^T) \\ & - \lambda_1 \left(\text{tr}(P^T G_P P) + \text{Tr}(Q^T G_Q Q) \right) + \frac{\rho}{2} \left\| Z - V + \Theta/\rho \right\|_F^2 \\ \text{s.t.} \quad & P^T P = I_{r_1}, Q^T Q = I_{r_2}, r_1 + r_2 = 2r, VV^T = I_k, z_{ij} \geq 0. \end{aligned} \quad (16)$$

In the sequel, we will present the steps of using an alternating optimization strategy to solve Equation (16) in detail.

5.1 Jointly Optimizing P , Q , r_1 , and r_2

It is intractable to directly optimize Equation (16) with respect to P and Q , because L_P and L_Q have no explicit analytical expressions in P or Q . Motivated by Wang et al. (2014), Peng et al. (2017a), and Peng et al. (2017b), we temporally regard the graph Laplacian terms as constant when optimizing with respect to P , Q , r_1 , and r_2 and subsequently update L_P and L_Q accordingly. With this strategy, the subproblem for $\{P, Q, r_1, r_2\}$ -optimization becomes

$$\begin{aligned} \min_{P, Q, r_1, r_2} & \left\| \vec{v}(X^P) - \vec{v}(U)V \right\|_F^2 + \left\| \vec{v}(X^Q) - \vec{v}(U)V \right\|_F^2 \\ & - \lambda_1 \text{tr}(P^T G_P P) - \lambda_1 \text{tr}(Q^T G_Q Q) \\ \text{s.t.} \quad & P^T P = I_{r_1}, Q^T Q = I_{r_2}, r_1 + r_2 = 2r. \end{aligned} \quad (17)$$

We define $F := \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix}$ and $H := \begin{bmatrix} H_{11} & 0 \\ 0 & H_{22} \end{bmatrix}$, where

$$\begin{aligned} H_{11} &:= (1 - \lambda_1)G_P - \sum_{i=1}^n (X_i^T \Theta_i + \Theta_i^T X_i), \\ H_{22} &:= (1 - \lambda_1)G_Q - \sum_{i=1}^n (X_i \Theta_i^T + \Theta_i X_i^T), \end{aligned}$$

and $\Theta_i := \sum_{j=1}^k U_j v_{ji}$. Then Equation (17) can be written succinctly as

$$\arg \min_{F^T F = I_{2r}} \text{Tr}(F^T H F). \quad (18)$$

With $\xi := \sum_{i=1}^n \|\Theta_i\|_F^2 + \lambda_1 \max\{\|G_P\|_F^2, \|G_Q\|_F^2\}$, Equation (18) is found to be equivalent to

$$\arg \min_{F^T F = I_{2r}} \text{Tr}(F^T (\mathbf{H} + \xi I) F). \quad (19)$$

It is straightforward to verify that $\mathbf{H} + \xi I$ is positive definite. Hence, immediately the solution to Equation (19) is yielded:

$$F = \text{perm}(\text{eig}_{2r}(\mathbf{H} + \xi I)) = \text{perm}(\text{eig}_{2r}(\mathbf{H})), \quad (20)$$

where $\text{eig}_{2r}(\cdot)$ returns the eigenvectors corresponding to the smallest $2r$ eigenvalues, and $\text{perm}(\cdot)$ is a permutation of the matrix columns, which ensures the block diagonal structure of F . By the definition of F , P and Q are updated as the diagonal blocks of F . Note that optimal r_1 and r_2 are naturally obtained as the numbers of columns of P and Q , respectively. Finally, L_P and L_Q are updated by using the new P and Q .

5.2 Optimizing V

The V -minimization subproblem is

$$\begin{aligned} \min_{V V^T = I} & \left\| \vec{v}(\mathbf{X}^P) - \vec{v}(\mathbf{U})V \right\|_F^2 + \left\| \vec{v}(\mathbf{X}^Q) - \vec{v}(\mathbf{U})V \right\|_F^2 \\ & + \lambda_2 \text{Tr}(Z(L_P + L_Q)V^T) + \rho/2 \left\| Z - V + \Theta/\rho \right\|_F^2. \end{aligned} \quad (21)$$

Due to the constraint of $V V^T = I$, Equation (21) is equivalent to minimizing

$$\begin{aligned} & -2\text{Tr}((\vec{v}(\mathbf{X}^P))^T \vec{v}(\mathbf{U})V) - 2\text{Tr}((\vec{v}(\mathbf{X}^Q))^T \vec{v}(\mathbf{U})V) \\ & + \lambda_2 \text{Tr}(Z(L_P + L_Q)V^T) - \text{Tr}((\rho Z + \Theta)^T V) + \zeta_1 \\ & = \text{Tr}(-2E^T V) + \zeta_1 = \|V - E\|_F^2 + \zeta_2, \end{aligned} \quad (22)$$

where ζ_1, ζ_2 are constant numbers, and

$$E := (\vec{v}(\mathbf{U}))^T (\vec{v}(\mathbf{X}^P) + \vec{v}(\mathbf{X}^Q)) - \frac{\lambda_2}{2} Z(L_P + L_Q) + \frac{\rho}{2} (Z + \Theta/\rho). \quad (23)$$

According to Lemma 3.2 in Huang et al. (2014) and Lemma 3.1 in Peng et al. (2017c), the optimal V is

$$V = AB^T, \quad (24)$$

where A and B are matrices containing the left and right singular vectors of E , respectively.

5.3 Optimizing Z

The subproblem of optimizing Z can be written as

$$\min_{z_{ij} \geq 0} \left\| Z - \left(V - \frac{\Theta}{\rho} - \frac{\lambda}{\rho} V (L_P^T + L_Q^T) \right) \right\|_F^2, \quad (25)$$

which admits the following closed-form solution

$$Z = \left(V - \frac{\Theta}{\rho} - \frac{\lambda_2}{\rho} V (L_P^T + L_Q^T) \right)_+, \quad (26)$$

where $(\cdot)_+$ is element-wisely defined by $(\theta)_+ = \frac{1}{2}(|\theta| + \theta)$.

5.4 Optimizing U

The subproblem associated with U -minimization is

$$\min_U \|\vec{v}(X^Q) - \vec{v}(U)V\|_F^2 + \|\vec{v}(X^P) - \vec{v}(U)V\|_F^2, \quad (27)$$

which admits a closed-form solution,

$$\vec{v}(U) = \frac{1}{2}(\vec{v}(X^P) + \vec{v}(X^Q))V^T. \quad (28)$$

From Equation (28), we can also obtain the explicit solution of U :

$$U = \left\{ \frac{1}{2} \sum_{i=1}^n (X_i P P^T + Q Q^T X_i) v_{ij} \right\}_{j=1}^k. \quad (29)$$

5.5 Update Θ and ρ

We update Θ and ρ in a standard way as follows:

$$\Theta = \Theta + \rho(Z - V), \quad \rho = \rho\kappa, \quad (30)$$

where $\kappa > 1$ is a parameter that controls the convergence speed. Usually, the larger κ is, the fewer iterations there are for the algorithm to converge. But for large κ , we may lose some precision of the final objective function value. Throughout this article, we set $\rho = 0.001$ and $\kappa = 1.1$.

By iterating all the above five updating steps, we can optimize the augmented Lagrangian 16 of model 11. The overall procedure for 2D k -means clustering is summarized in Algorithm 1.

Discussion 5.1. In Equation (11) we aim to simultaneously achieve clustering, feature extraction, and manifold construction. In Algorithm 1, 2D matrices are directly used as the input, and optimal projection directions are found as the output with inherent spatial structures of 2D data well preserved. With such projections, the 2D matrix samples are projected to a subspace containing the most variational information, where cluster centroids and manifolds are constructed from the most expressive features while the adverse effects from less significant or noisy features are reduced, thus leading to a robust model. Furthermore, in the projected subspaces, with the adaptively constructed manifolds the new representations of the data are yielded through incorporating the underlying nonlinear geometrical structures. Thus, the new representations and the intrinsic manifolds will also help find the optimal projection directions. Working synergistically and iteratively, the learning tasks of seeking subspaces, learning manifolds, and constructing centroids mutually enhance and yield a powerful data representation and grouping.

ALGORITHM 1: DTKM: Clustering Method of 2D k -Means

- 1: **Input:** data X , parameters $\{\lambda_1, \lambda_2, r\}$, maximal iteration number t_{max} , number of clusters k
 - 2: **Initialize:** $U^0, V^0, Z^0, P^0, t = 0$.
 - 3: **repeat**
 - 4: Update P, Q by (20);
 - 5: Update L_P and L_Q by (2) using X^P and X^Q , respectively;
 - 6: Update $\vec{v}(U)$ by (28), or update U by (29);
 - 7: $t = t + 1$.
 - 8: **until** $t \geq t_{max}$ or convergence
 - 9: Apply standard K-means to rows of V
 - 10: **Output:** Predicted cluster indicators
-

Table 1. Clustering Performance on Yale

N	Accuracy (%)								
	HCA	<i>k</i> -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	51.36±02.20	71.36±22.78	70.00±10.97	71.36±19.05	73.64±24.97	73.18±223.61	53.64±04.18	86.36±13.03	92.27±07.44
3	40.00±08.43	61.82±15.20	62.73±11.52	52.12±13.08	63.33±21.40	65.45±18.75	52.12±04.24	72.12±12.10	84.55±06.62
4	30.91±05.79	46.14±11.54	58.18±12.69	44.09±06.96	49.55±11.33	48.41±10.28	40.45±07.93	60.45±10.45	78.41±12.96
5	28.18±06.43	50.00±08.45	52.73±12.03	50.73±16.47	53.27±09.35	54.36±19.59	39.82±05.65	58.73±11.37	70.91±14.97
6	23.64±05.40	50.61±04.91	43.94±05.39	37.58±07.92	53.18±03.53	51.97±04.04	33.64±03.90	51.36±04.07	65.15±09.56
7	22.73±06.13	47.27±06.57	47.53±07.75	39.61±11.19	54.55±04.37	51.95±05.09	34.81±06.12	52.73±04.02	67.01±06.40
8	21.36±04.54	48.30±05.55	46.25±08.02	37.73±10.67	53.98±04.92	52.61±06.16	32.95±05.59	49.89±04.68	63.86±07.42
9	19.29±03.31	45.45±07.68	45.15±06.95	32.53±06.28	54.44±02.67	48.28±05.92	30.00±03.20	50.30±06.85	58.59±04.97
10	17.64±03.30	43.36±04.29	41.82±06.23	28.91±05.71	50.00±04.56	46.45±04.91	28.73±02.36	46.73±03.81	57.64±05.47
12	16.97±01.72	40.83±04.23	43.03±04.31	30.30±03.50	49.92±05.74	46.44±03.03	27.80±01.43	45.08±06.04	52.05±04.07
14	15.39±00.31	41.75±02.82	37.92±04.10	25.52±03.02	45.65±02.87	46.04±02.09	25.06±00.98	44.61±03.33	51.62±02.96
15	14.55	39.39	43.03	21.82	44.24	43.03	21.82	44.85	52.12
Average	25.17	48.88	49.36	39.36	53.81	52.35	35.07	55.27	66.13
N	Normalized Mutual Information (%)								
	HCA	<i>k</i> -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	01.41±02.27	35.14±42.14	17.22±15.69	30.43±29.47	43.23±46.31	39.63±42.56	01.15±01.90	52.87±29.87	67.87±27.05
3	10.49±16.72	33.94±22.01	33.19±16.80	25.24±20.95	43.87±31.44	41.45±28.65	18.91±10.05	46.94±18.56	62.74±12.59
4	09.91±11.06	23.75±16.61	37.56±14.27	21.25±11.23	33.70±15.98	28.47±14.65	14.95±10.54	39.72±12.17	63.39±16.14
5	14.66±12.12	35.68±14.72	36.96±15.16	33.17±22.40	44.13±12.21	39.87±26.15	23.49±09.30	44.78±15.32	61.59±17.07
6	13.48±09.35	40.74±07.93	33.76±07.42	21.77±12.82	44.67±05.95	42.47±06.28	20.54±07.17	38.37±06.02	57.72±09.79
7	16.59±08.73	39.44±07.78	40.62±07.07	29.02±16.15	48.63±06.08	46.31±05.75	27.41±08.44	43.89±06.09	63.12±07.54
8	16.60±06.88	43.51±05.72	43.17±09.20	33.19±14.25	49.65±06.68	47.97±08.00	26.32±06.87	44.85±05.35	61.17±06.62
9	15.94±05.53	42.94±08.06	41.48±06.11	25.19±08.62	51.76±03.55	46.80±06.43	26.60±05.05	45.97±06.09	57.43±05.23
10	15.70±04.55	43.73±03.17	41.53±05.41	23.94±06.23	49.02±05.39	46.39±05.34	27.50±03.24	44.59±04.10	58.00±05.78
12	16.82±02.40	42.83±04.26	44.42±04.20	31.53±05.45	51.76±02.82	50.24±02.37	30.29±01.43	46.56±04.94	55.99±03.03
14	16.82±00.13	46.20±02.59	44.88±03.26	28.78±04.08	50.86±02.13	51.36±01.95	30.63±00.76	47.77±02.04	55.53±02.72
15	16.40	43.84	46.95	26.86	49.64	52.56	29.25	48.20	52.59
Average	13.74	39.31	38.48	27.53	46.74	44.46	23.09	45.38	59.76
N	Purity (%)								
	HCA	<i>k</i> -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	51.30±02.20	71.36±22.78	70.00±10.97	71.36±19.05	73.64±24.97	73.18±23.61	53.64±04.18	86.36±13.03	92.27±07.44
3	40.61±09.28	61.82±15.20	63.64±10.69	53.94±13.76	63.33±21.40	65.45±18.75	54.24±04.62	72.12±12.10	84.55±6.62
4	32.50±06.61	46.14±11.54	60.45±11.98	45.68±07.83	49.55±11.33	48.64±10.18	42.95±08.73	61.14±09.55	79.09±11.53
5	30.91±07.22	50.00±08.45	54.18±10.90	50.91±16.46	53.27±09.35	54.73±19.93	42.73±06.87	59.09±11.48	71.64±14.35
6	27.27±05.62	50.76±04.80	45.15±05.16	38.18±08.62	53.18±03.53	51.97±04.04	36.82±05.49	51.52±04.04	65.76±09.18
7	27.01±05.77	47.53±06.16	49.48±07.36	40.78±11.74	54.55±04.37	52.34±04.37	38.57±06.71	53.77±04.21	68.44±06.06
8	25.34±04.55	48.75±05.45	47.61±08.27	39.77±10.44	53.98±04.92	52.95±06.23	35.45±06.49	50.45±04.65	65.11±06.82
9	23.74±03.54	46.67±07.25	45.86±06.53	33.74±07.95	54.44±02.67	49.39±06.03	33.23±03.35	50.61±06.89	60.30±04.72
10	22.55±03.02	45.09±03.81	43.36±05.54	29.91±07.79	50.73±04.24	47.82±04.35	32.18±03.07	47.55±04.02	59.45±04.11
12	21.89±01.61	43.03±03.81	43.64±04.56	34.55±04.38	51.29±03.75	48.18±02.48	31.97±01.92	46.52±05.44	54.17±03.72
14	20.78±00.00	43.12±02.67	40.45±03.61	28.31±04.38	47.01±02.62	47.60±01.23	29.42±01.06	46.04±03.08	53.64±02.79
15	20.00	40.61	43.03	21.82	45.45	46.06	26.06	44.85	52.73
Average	28.66	49.57	50.57	40.75	54.20	53.19	38.11	55.84	62.26

6 EXPERIMENTS

In this section, we conduct extensive experiments on several benchmark image datasets to assess the effectiveness of DTKM in comparison with several state-of-the-art clustering algorithms. To quantitatively evaluate the performance, we adopt three metrics that are widely used in clustering, including clustering accuracy, normalized mutual information (NMI), and purity. Clustering accuracy measures the extent to which each cluster contains data points from the same class:

$$\text{Accuracy} = \frac{\sum_{i=1}^n \delta(\text{map}(s_i), s'_i)}{n}, \quad (31)$$

Table 2. Clustering Performance on PIX

N	Accuracy (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	94.50±10.39	94.50±10.39	99.50±01.58	94.50±10.39	99.50±01.58	99.50±01.58	73.00±11.11	96.50±07.84	100.0±00.00
3	85.33±14.42	96.00±05.84	96.67±03.85	95.00±06.89	96.00±05.84	97.67±06.30	60.67±09.27	97.33±03.06	99.67±01.05
4	87.00±22.26	96.25±04.60	77.00±13.43	90.50±13.58	97.25±03.81	99.25±01.69	59.25±12.47	96.50±04.44	99.75±00.79
5	72.80±24.37	87.20±11.00	75.40±11.85	71.80±13.62	90.80±09.34	95.40±08.17	55.00±10.38	90.80±07.50	99.40±00.97
6	63.33±24.47	84.83±12.38	74.17±12.65	72.83±12.74	90.17±09.51	90.00±11.92	47.33±05.89	89.00±08.72	98.17±04.19
7	68.14±23.00	84.14±05.89	71.86±05.68	65.57±08.31	90.27±07.64	94.86±05.64	51.29±06.48	87.14±07.85	94.43±05.53
8	61.25±08.27	84.12±05.65	78.13±04.18	58.00±06.07	87.25±07.14	95.25±02.27	51.38±03.79	82.37±05.38	88.50±05.92
9	72.11±09.83	83.22±07.45	88.89±04.89	57.11±04.03	92.89±00.57	96.44±00.70	45.33±05.02	87.00±06.83	94.33±05.53
10	73.00	80.00	70.00	62.00	80.00	87.00	11.00	81.00	95.00
Average	75.27	87.81	81.29	74.15	91.57	95.04	50.47	89.74	96.58
N	Normalized Mutual Information (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	83.81±28.77	83.81±28.77	97.58±07.64	83.81±28.77	97.58±07.64	97.58±07.64	25.16±21.49	88.28±22.45	100.0±00.00
3	77.98±11.64	89.87±11.64	90.84±08.94	88.11±14.96	89.87±11.64	94.95±12.80	34.04±13.10	92.32±08.18	98.98±03.22
4	86.63±07.13	93.39±07.13	74.99±12.76	86.82±16.72	94.67±05.78	98.42±03.44	42.33±16.08	93.45±07.08	98.75±02.01
5	75.40±07.04	87.42±08.35	77.38±10.89	67.15±14.69	90.13±07.54	93.60±10.71	47.52±10.15	88.04±07.35	98.04±03.78
6	70.44±05.09	86.71±08.47	73.30±08.81	69.75±11.39	90.64±05.68	91.12±08.83	45.19±07.04	87.05±07.23	93.71±05.88
7	75.50±04.18	86.74±03.63	77.56±06.43	64.08±09.35	91.54±04.57	93.92±04.67	49.54±08.12	87.06±07.05	92.46±02.09
8	72.33±01.28	85.70±03.87	81.73±04.35	60.00±05.12	90.56±03.07	93.29±02.72	49.33±03.90	83.54±04.15	93.50±03.19
9	80.16±01.76	86.31±04.83	87.91±06.31	60.67±03.91	92.78±00.82	94.76±00.94	45.26±03.24	87.89±04.59	94.31±03.27
10	81.42	88.09	69.29	62.45	87.01	92.52	11.73	86.02	95.85
Average	78.19	87.56	81.18	71.43	91.64	94.46	38.90	88.18	96.83
N	Purity (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	94.50±10.39	94.50±10.39	99.50±01.58	94.50±10.39	99.50±01.58	99.50±01.58	73.00±11.11	96.50±07.84	100.0±00.00
3	85.33±05.84	96.00±05.84	96.67±03.85	95.00±06.89	96.00±05.84	97.67±06.30	62.67±08.13	97.33±03.06	99.67±01.05
4	89.25±04.60	96.25±04.60	80.00±10.61	91.50±11.74	97.25±03.81	99.25±01.69	62.00±11.77	96.50±04.44	99.75±00.79
5	76.80±07.21	89.20±07.61	78.60±09.89	75.00±11.16	91.80±06.83	95.40±08.17	60.20±09.16	91.40±06.11	99.40±00.97
6	69.00±06.45	87.50±09.24	77.67±09.17	74.33±12.20	91.83±06.36	92.17±08.57	53.83±05.09	89.67±07.06	98.17±04.19
7	72.71±04.88	86.43±03.94	76.00±04.80	68.86±07.12	91.71±05.08	95.29±04.57	55.57±07.51	88.57±06.02	94.43±05.53
8	67.00±01.94	85.25±04.52	80.00±03.95	61.75±04.83	89.50±04.53	95.25±02.27	55.13±02.91	85.00±03.82	90.38±03.91
9	76.11±01.95	85.00±06.05	88.89±04.83	59.56±03.93	92.89±00.57	96.44±00.70	49.33±03.89	88.22±05.14	94.78±03.48
10	77.00	84.00	70.00	63.00	84.00	89.00	19.00	85.00	95.00
Average	78.63	89.35	83.04	75.94	92.72	95.55	54.53	90.91	96.84

where s and s' are the predicted and true labels of the i th data point, $map(s_i)$ is a mapping function such that each cluster label s'_i is mapped to an equivalent label from the dataset by permutation with Equation (31) being maximized, $\delta(x, y)$ is the Kronecker delta function that returns 1 when $x = y$ and otherwise 0, and n represents the total number of data points. Normalized mutual information is to measure the quality of clusters, which is defined as

$$NMI = \frac{\sum_{i=1}^N \sum_{j=1}^N n_{i,j} \log \frac{n_{i,j}}{n_i \hat{n}_j}}{\sqrt{(\sum_{i=1}^N n_i \log \frac{n_i}{n})(\sum_{j=1}^N \hat{n}_j \log \frac{\hat{n}_j}{n})}}, \quad (32)$$

where $n_{i,j}$ is the number of data points in the intersection between the i th cluster and j th class, whose sizes are n_i and $\hat{n}_j$ respectively, and N is the number of clusters. Purity measures the extent to which each cluster contains data points from primarily one class. It is defined as

$$\text{Purity} = \frac{1}{n} \sum_{i=1}^N \max(n_i^j), \quad (33)$$

Table 3. Clustering Performance on EYaleB

N	Accuracy (%)									
	HCA	k-Means	KKM	SC	PCA	RPCA	2DPCA	NMF	RMNMF	DTKM
5	21.06±00.24	23.01±00.80	28.82±02.30	————	23.38±00.68	23.01±00.99	23.32±01.06	23.79±01.61	28.82±03.98	53.74±08.83
10	11.04±00.13	13.84±01.19	16.72±00.93	————	13.46±00.67	13.93±01.03	13.89±00.81	14.66±00.39	22.67±02.02	35.76±05.19
15	07.56±00.09	11.46±01.01	14.06±01.31	————	10.60±00.47	10.97±00.77	11.13±00.94	11.38±00.40	20.98±01.98	28.58±03.30
20	05.94±00.09	10.69±01.08	11.33±01.00	————	09.65±00.51	09.85±00.80	10.48±00.79	09.77±00.43	19.64±01.19	25.72±01.76
25	04.94±00.04	09.35±01.05	10.16±00.67	————	07.70±00.47	08.37±00.38	08.78±00.72	08.54±00.22	17.75±01.30	22.74±01.78
30	04.25±00.07	08.48±00.71	09.57±00.71	————	07.33±00.24	08.14±00.66	08.61±01.02	07.88±00.17	17.21±01.10	21.04±01.74
35	03.77±00.02	08.85±00.75	08.86±00.63	————	06.63±00.28	08.67±00.80	08.31±00.48	07.12±00.16	16.48±01.23	19.56±01.27
38	03.56	08.53	08.37	————	06.59	08.99	08.33	07.08	16.86	19.30
Average	07.76	11.78	13.49	————	10.67	11.49	11.61	11.28	20.05	28.31
N	Normalized Mutual Information (%)									
	HCA	k-Means	KKM	SC	PCA	RPCA	2DPCA	NMF	RMNMF	DTKM
5	01.30±00.11	00.94±00.73	03.76±01.23	————	00.78±00.44	00.74±00.35	00.95±00.87	01.25±01.61	06.49±04.14	35.29±10.44
10	01.52±00.09	02.39±02.21	03.62±00.67	————	01.56±00.44	02.17±01.15	02.24±00.96	03.07±00.61	15.34±01.93	31.41±06.69
15	01.55±00.07	03.96±01.48	05.53±00.86	————	02.77±00.68	03.31±01.35	03.70±01.63	04.05±00.63	19.89±03.74	29.24±03.15
20	01.56±00.06	06.91±01.59	05.82±00.60	————	05.02±01.17	05.50±01.99	06.61±01.49	05.54±00.56	23.11±01.25	28.41±01.67
25	01.60±00.04	07.00±01.42	07.08±00.41	————	04.18±01.05	05.41±00.88	05.93±01.49	06.16±00.39	24.04±01.15	27.01±01.84
30	01.61±00.02	07.79±01.19	08.35±00.45	————	05.51±00.41	07.49±01.02	07.71±01.01	07.34±00.28	26.28±01.09	26.46±01.31
35	01.62±00.02	10.04±01.44	09.10±00.37	————	05.81±00.55	09.41±01.48	09.56±00.74	08.08±00.24	27.67±01.58	26.46±01.35
38	01.62	10.51	10.02	————	06.13	10.42	10.26	08.79	28.46	27.11
Average	01.55	06.19	06.66	————	03.97	05.56	05.87	05.54	21.41	28.93
N	Purity (%)									
	HCA	k-Means	KKM	SC	PCA	RPCA	2DPCA	NMF	RMNMF	DTKM
5	21.37±00.22	23.73±01.31	29.66±02.28	————	23.95±01.08	23.80±01.22	23.98±01.61	24.26±01.70	29.45±03.99	54.72±08.16
10	11.57±00.12	14.58±01.31	17.82±01.23	————	14.20±00.90	14.64±01.31	14.88±01.04	15.54±00.86	23.73±01.88	37.56±05.53
15	08.25±00.09	12.26±01.02	14.84±01.22	————	11.19±00.63	11.73±00.90	11.83±00.95	11.99±00.48	22.16±02.24	30.78±03.25
20	06.59±00.07	11.27±01.10	12.06±00.99	————	10.28±00.58	10.39±00.82	11.20±00.66	10.53±00.55	20.48±01.25	27.57±01.88
25	05.62±00.04	10.00±01.00	10.57±00.59	————	08.29±00.50	08.95±00.37	09.36±00.75	09.07±00.29	18.77±01.29	24.37±01.84
30	04.95±00.03	09.23±00.60	09.95±00.74	————	07.91±00.20	08.79±00.68	09.22±00.99	08.40±00.25	18.13±01.04	22.62±01.96
35	04.49±00.01	09.66±00.71	09.25±00.57	————	07.22±00.36	09.36±00.86	09.09±00.49	07.62±00.22	17.42±01.18	21.07±01.34
38	04.27	08.86	08.70	————	07.08	09.57	09.03	07.62	17.94	21.67
Average	08.39	12.46	14.11	————	11.27	12.15	12.32	11.88	20.01	30.05

where n_j^i is the number of data points in the j th cluster that actually belong to the i th class. It should be noted that all these measures have values ranging from 0 to 1, where higher value suggests better performance. More details about these measures can be found in Han et al. (2011) and Peng et al. (2017b). In the following, we will present the state-of-the-art algorithms in comparison, benchmark data, and experimental results.

6.1 Algorithms in Comparison

We compare DTKM with nine competing methods which are as follows. First, we have Hierarchical Cluster Analysis (HCA). HCA is a clustering method which seeks to build a hierarchy of clusters. In the experiment, we use the complete-linkage approach. Second, we have k -means, which is one of the most widely used clustering algorithms. A fast implementation¹ of it is used as a standalone method as well as the final step of other methods. Third, we have KKM (Schölkopf et al. 1998). Radial basis function (RBF) kernel is used, with radius selected from the set $\mathcal{S} = \{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$. Fourth, we have SC (Ng et al. 2002). RBF kernel is used to construct the graph Laplacian with radius selected from $\mathcal{S}$. Fifth, we have Robust PCA (RPCA)

¹<http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html>.

(Candès et al. 2011), which is a robust extension of PCA. We use the inexact augmented Lagrange multiplier method (Lin et al. 2010) for its optimization, where the theoretically optimal parameter given in Candès et al. (2011) is used. Sixth, we have Two-Dimensional PCA (2DPCA) (Yang et al. 2004). We choose different r values ranging from 1 to R as the number of projection directions, where R is fixed to 5. Seventh, we have NMF (Lee and Seung 1999). Standard multiplicative updating rules are used. Eighth, we have Robust Manifold NMF (RMNMF) (Huang et al. 2014), which relaxes the basis to have mixed signs and incorporates the graph Laplacian to measure nonlinear structures of the data. It can be regarded as a more robust version of graph NMF (Cai et al. 2011) or a nonlinear version of Semi-NMF (Ding et al. 2010). Ninth, we have DTKM. Because DTKM seeks projections on two sides of the data, for fair comparison with 2DPCA, r is chosen in the range of 1 and $2R$, such that the average number of projections on each side can be up to R . In the same way as RMNMF, the regularization parameter is chosen from $\mathcal{S}$, and the graph Laplacian is constructed using binary weighting with five neighbors.

Estimating the number of clusters of a given dataset is an important problem in cluster analysis both theoretically and practically (Kolesnikov et al. 2015). In this article, we focus on proposing a new clustering method, through simultaneous learning of projections, data representations, and nonlinear relationships. How to determine k is out of the scope of this article, and we regard k as an input as commonly done by many clustering methods (Huang et al. 2014; Peng et al. 2017b, 2015; Patel et al. 2013; Cai et al. 2013). In practice, when k is unknown, it is suggested to estimate k in a preprocessing step, where existing approaches such as Kolesnikov et al. (2015) can be adopted for a reasonable estimation. In this article, for fair comparison, we specify the number of clusters for all algorithms.

6.2 Benchmark Datasets

We briefly describe the datasets used in our experiment as follows. First, Yale (Belhumeur et al. 1997) contains 165 grayscale images of 15 persons with 11 images of size 32×32 per person. Second, the Extended Yale B (EYaleB) (Georghiades et al. 2001) dataset has 38 persons and around 64 face images under different illuminations per each person. The images were cropped to 192×168 and were resized to 32×32 in our experiments. This dataset is heavily noisy due to shadows on faces. Third, PIX (Hond and Spacek 1997) has 100 grayscale images of 100×100 pixels from 10 objects that were collected. Fourth, ORL (Samaria and Harter 1994) data has 40 individuals and 10 images of size 32×32 pixels. Fifth, the Faces96 (Hond and Spacek 1997) dataset contains images of size 196×196 from 152 individuals. We resize each image to 28×28 pixels.

6.3 Clustering Performance on Original Data

The total number of clusters, denoted by $\bar{N}$, varies across datasets. To reduce contingency and for more detailed comparison, we follow the approach in Cai et al. (2011). For each dataset, we conduct experiments not only on the whole data, but also its subsets. By doing so, we can see how the performance varies with different data sizes. For this purpose, for each dataset, we randomly choose N out of $\bar{N}$ clusters as a subset. Out of all possible subsets, we randomly choose 10 subsets so that contingency effect can be reduced. Then for each method, we tune its parameters with all but selected ones and test it on the 10 selected subsets, and report the highest average performance. For a given dataset, by varying N from 1 to $\bar{N}$ we are able to conduct experiments with subsets of different sizes, which are specified in the first columns of Tables 1 to 3. The same strategy is applied throughout this section.

The clustering performance is presented in Tables 1 to 3. It is observed that DTKM has significantly improved the clustering performance on Yale, EYaleB, and PIX data. On Yale and EYaleB

data, DTKM has the best performance in all cases. The average improvements on these datasets are around 10% and 9%, respectively. It is observed that on EYaleB data, when N is large, the improvement by DTKM is only about 4%. However, it should be noted that all methods have low performance on this dataset. For example, except for RMNMF and DTKM, the performance of other methods is below 10%. The (absolute) improvement of DTKM against all competing methods but RMNMF is more than 10%, sometimes as high as about 20%; therefore, DTKM has significant improvement over these methods. When compared to RMNMF, although the improvement is around 4% with large N values, considering the difficulty of clustering on this dataset and the relative improvement of about 25%, it is also deemed significant. Because EYaleB contains heavy noise or corruptions, the improvement on EYaleB demonstrates that DTKM is more robust to noise or corruptions than other methods. On PIX, 2DPCA has competitive performance, whereas DTKM has the best results in almost all cases; even in cases when it is not the best, DTKM still has highly competitive performance. It is worth pointing out that, although 2DPCA has good performance on PIX data with some N values, its performance on other datasets is not as promising as DTKM whose performance is always among the best ones.

In summary, these results testify to the effectiveness of DTKM in clustering. It is noted that datasets used in these experiments have less than 40 clusters. Such data are common in many areas such as earthquake engineering or bioinformatics. Hence, the effectiveness of DTKM on such data implies meaningful applications to these areas. We will also test DTKM on dataset with larger numbers of clusters later on.

6.4 Clustering Performance on Corrupted Data

As mentioned above, Table 3 demonstrates the robustness of DTKM to noise or corruptions. To more extensively confirm the robustness, we compare all methods on corrupted ORL data. We randomly chose 40%, 60%, and 80% pixels from the images and set the pixel values to zero. In Figure 1, we illustrate several original and corrupted images from the ORL dataset for visualization. The detailed experimental results are reported in Tables 4 to 6. It is observed that DTKM has the best performance in all cases. Compared to the second-best performances, DTKM has average improvements in three measures by about 15%, 10%, and 5% on 40%, 60%, and 80% corrupted data, respectively. When the corruption level is increased, the performance improvement is reduced. This is reasonable, because it is more difficult to cluster the data with heavier corruption and all methods naturally fail if all elements from the data are missing. SC fails to work with corrupted data because the corresponding graph Laplacian is too noisy to give any meaningful results. Consequently, no results of SC are reported on corrupted data. In contrast, while also using the graph Laplacian, the proposed DTKM can work well thanks to the intrinsic robustness of our model.

In summary, with heavy corruptions, the experimental results demonstrate that DTKM has significant improvements over and thus is more robust than other methods in comparison.

6.5 Clustering Performance on a Larger Dataset

In the above, we have evaluated DTKM on some datasets with cluster numbers smaller than 40. To better assess DTKM, we compare it with other methods on a larger dataset, Faces96. For this dataset, we select 100 persons, i.e., $\bar{N} = 100$, and in a way similar to Huang et al. (2014) we select the first 10 images for each person. To further investigate the robustness of DTKM, we also add 60% corruptions to this dataset as done in Section 6.4. We vary N value over $\{10, 20, \dots, 100\}$ and report the performance of all methods in Table 7. It is observed that DTKM has the best performance in accuracy and purity, with improvements of around 10% with small N values and 3% with large N values compared with the second-best method, i.e., 2DPCA. In NMI, it is observed that DTKM has

Fig. 1. Sample images from ORL dataset with 0% (original image), 40%, 60%, and 80% corruptions.

improvements of around 10% with small N values and comparable results with large N values. In summary, DTKM is highly effective on Faces96 with or without corruption.

6.6 Parameter Sensitivity

In this section, we empirically analyze how the parameters affect the performance of DTKM. Due to space limit, we only show results on part of the data.

We first show the performance of DTKM with respect to different r values. For better illustration, we compare DTKM with 2DPCA with r values ranging over $\{1, 2, 3, 4, 5\}$. Without loss of generality, for each data, we use the same subsets as Section 6.3 with the smallest number of clusters. For each r value, we tune $\{\lambda_1, \lambda_2\} \in \mathcal{S} \times \mathcal{S}$ and report the best average performance in Figure 2. With small r , DTKM already demonstrates promising performance, indicating that DTKM captures the most expressive features with the top few projection directions. Moreover, the performance curves of DTKM are much higher and more stable than 2DPCA.

Then we assess the performance of DTKM with respect to different combinations of λ_1 and λ_2 . For each combination, we report the performance with r tuned such that the best results are observed. From Figure 3, it is observed that with relatively larger values of λ_1 and λ_2 , over a wide range of combinations DTKM has favorable performance.

Fig. 2. Performance variations in accuracy, NMI, and purity, with respect to different r values on Yale, EYaleB, PIX, and ORL with 40%, 60%, and 80% corruptions.

In summary, the experimental results empirically demonstrate insensitivity of DTKM to parameters. Such insensitivity suggests its ease of use in real-world applications.

6.7 Projections

In this section, we will investigate the projections by visualizing the extracted 2D features from the data. Figure 4 presents two sample images from the Yale dataset, along with the i th features extracted by P and Q projections, and the images recovered by the top i projection directions of P and Q , for $i = 1, 2, 3, 4$, respectively. It is observed that with the top few projection directions, the major information of the original matrix can be well retained, suggesting that seeking projections from both directions is indeed effective.

6.8 Potential Applications

As a piece of fundamental research, in this work we have proposed a new clustering method called DTKM and evaluated it on some widely used benchmark datasets. As experimentally demonstrated, DTKM has strong performance and thus high potential in real-world applications. Some prospective applications include, but are not limited to, the following:

- *Biomedical images*: On positron emission tomography scans, clustering is often used, which is to distinguish the types of tissues for various purposes (Filipovych et al. 2011).
- *Image segmentation*: In computer vision, there is often a need to partition a scene into multiple segments, which makes it easier to analyze with a new representation of the image.

Fig. 3. Performance variations in accuracy, NMI, and purity with respect to different combinations of λ_1 and λ_2 values on Yale, EYaleB, and PIX datasets.

Fig. 4. Extracted 2D features from two sample images of the Yale dataset. In each block of (a) and (b), the top left is the original image; in the right, from top to bottom are $Xp_i p_i^T$, $\sum_{j=1}^i Xp_j p_j^T$, $q_i q_i^T X$, and $\sum_{j=1}^i q_j q_j^T X$, i.e., the i th feature extracted by P , the recovered image by the top i projection directions of P , the i th feature extracted by Q , and the recovered image by the top i projection directions of Q , respectively. From left to right, i equals 1, 2, 3, and 4, respectively.

Table 4. Clustering Performance on 40% Corrupted ORL

N	Accuracy (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	60.00±14.14	78.50±15.82	55.00±00.00	————	65.50±09.85	77.50±20.03	61.00±06.58	76.50±16.67	93.50±09.73
4	31.00±03.16	49.75±09.68	37.50±00.00	————	49.50±06.65	59.00±10.08	38.75±04.45	53.50±10.01	68.75±12.09
6	23.83±01.37	36.33±07.89	31.67±00.00	————	43.67±07.28	48.17±12.36	31.67±02.48	45.50±10.45	68.67±17.12
8	18.50±01.84	30.38±04.72	31.25±00.00	————	41.25±05.56	42.75±06.42	29.75±04.16	34.25±04.87	59.12±06.92
10	15.90±00.88	29.27±03.58	27.00±00.00	————	34.50±04.22	39.40±05.13	11.00±00.00	33.10±03.60	53.20±07.39
12	13.67±01.43	27.42±02.40	22.50±00.00	————	33.50±04.02	34.50±04.93	23.33±01.80	30.75±02.65	47.17±04.16
14	12.14±00.95	25.50±02.78	21.43±00.00	————	32.64±03.63	36.21±04.73	23.29±01.31	28.50±02.98	44.00±02.84
16	11.37±00.77	23.81±02.80	21.88±00.00	————	29.25±02.55	33.88±03.06	21.69±00.84	25.94±01.57	43.19±04.73
18	10.61±00.81	23.06±02.03	19.44±00.00	————	30.72±03.74	32.06±02.80	21.39±01.29	25.83±01.58	44.00±04.44
20	10.20±00.89	22.10±02.00	20.00±00.00	————	27.70±03.10	31.95±03.07	19.85±01.36	26.35±01.43	39.45±03.64
Average	20.72	34.61	28.77	————	35.57	43.54	25.84	38.02	55.70
N	Normalized Mutual Information (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	14.62±30.00	43.76±35.61	00.79±00.00	————	12.74±12.09	43.50±44.67	07.33±09.71	34.46±33.07	76.60±30.23
4	09.63±03.12	26.82±12.41	08.93±00.00	————	24.67±06.12	38.37±12.68	11.68±04.82	32.00±12.80	54.17±14.51
6	10.45±02.09	20.64±09.85	17.02±00.00	————	30.53±09.59	35.82±13.95	15.68±03.16	31.24±10.86	61.41±20.70
8	10.38±01.55	22.45±06.22	21.29±00.00	————	36.09±06.28	36.88±07.35	20.69±03.54	26.83±06.68	56.09±06.98
10	09.72±00.86	25.37±04.16	23.72±00.00	————	33.73±04.38	38.11±05.97	11.73±00.00	29.77±04.75	54.80±05.89
12	10.16±00.78	26.16±03.84	24.40±00.00	————	35.80±03.90	36.13±06.41	24.13±01.96	30.12±02.11	51.80±04.46
14	10.00±00.42	26.10±02.79	26.36±00.00	————	38.26±03.47	41.14±05.66	27.07±02.20	30.89±03.02	50.78±02.45
16	10.08±00.36	27.18±02.80	29.41±00.00	————	35.31±02.43	41.68±03.10	27.75±00.68	31.74±02.58	51.45±04.41
18	10.35±00.65	27.78±01.76	28.80±00.00	————	38.96±03.73	42.12±02.52	29.92±01.17	33.37±01.92	49.46±04.50
20	10.17±00.43	28.41±02.30	30.99±00.00	————	37.79±03.30	43.96±03.97	30.09±01.33	36.87±01.33	49.75±03.64
Average	10.56	27.47	21.17	————	32.39	39.77	20.61	31.74	55.63
N	Purity (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	60.00±14.14	78.50±15.82	55.00±00.00	————	65.50±09.85	77.50±20.03	61.00±06.58	76.50±16.67	93.50±09.73
4	34.00±02.69	52.00±09.04	40.00±00.00	————	51.00±05.43	60.25±09.46	39.75±03.62	55.00±09.72	70.75±09.58
6	26.67±01.76	37.83±07.41	33.33±00.00	————	45.17±07.76	49.83±11.56	33.67±03.12	46.50±09.64	70.00±16.22
8	22.25±01.29	32.63±05.79	32.50±00.00	————	43.25±06.02	45.62±05.69	31.25±03.58	36.25±04.37	61.50±07.52
10	19.40±00.70	31.50±03.41	30.33±00.00	————	37.00±04.69	41.60±05.15	19.00±00.00	35.60±03.44	56.70±07.42
12	18.00±00.70	29.92±03.13	23.33±00.00	————	35.50±04.67	37.33±05.04	24.50±01.63	33.33±02.29	50.42±04.53
14	16.71±00.37	27.71±02.35	22.86±00.00	————	34.93±03.29	38.64±05.49	24.57±01.27	30.79±03.20	46.64±02.50
16	15.88±00.32	26.13±02.74	23.75±00.00	————	31.81±02.80	36.44±03.09	22.50±01.06	27.75±01.70	46.50±05.23
18	15.39±00.46	24.89±01.88	20.56±00.00	————	32.67±03.74	35.06±02.94	22.56±01.34	28.11±01.78	42.72±04.08
20	14.80±00.35	23.65±01.86	21.50±00.00	————	29.85±02.83	34.20±03.36	20.80±01.42	28.10±01.76	42.45±04.39
Average	24.31	36.48	30.28	————	41.57	45.65	29.97	39.80	58.12

For example, this technique can be widely used for border detection and object detection (Bewley and Upcroft 2013).

- *Recommender systems*: Hybrid filtering (Burke 2002) is usually based on bio-inspired or probabilistic models and clustering (Shinde and Kulkarni 2012). As mentioned previously, user-item data may change over time, where features of a specific user turn out to be a matrix in an item-time form. For such 2D matrices, DTKM can be a suitable clustering technique.
- *Social networks*: Social networks are ubiquitous and it is pivotal to discover close-knit clusters in them (Mishra et al. 2007). By monitoring the change of a community over time, higher-order data is frequently encountered, for which clustering of 2D matrices is a promising technique.
- *Intelligent transportation systems*: It is a long-standing topic to reduce traffic congestions in intelligent transportation systems. 2D matrices are also used in traffic and congestion

Table 5. Clustering Performance on 60% Corrupted ORL

N	Accuracy (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	55.00±00.00	72.50±13.18	55.00±00.00	————	55.00±00.00	67.00±12.74	59.00±06.58	66.50±11.56	87.50±11.84
4	30.50±01.05	42.25±05.83	37.50±00.00	————	27.50±00.00	44.00±08.51	38.00±03.07	44.75±06.06	61.50±05.55
6	23.50±01.46	33.00±04.50	31.67±00.00	————	18.33±00.00	37.00±09.26	31.50±03.80	35.50±04.91	51.67±16.10
8	18.75±01.95	29.25±01.79	31.25±00.00	————	13.75±00.00	32.25±03.27	28.50±02.75	29.13±02.95	42.50±05.30
10	16.70±01.42	26.20±01.69	27.00±00.00	————	11.00±00.00	28.30±03.56	11.00±00.00	26.30±02.36	37.50±03.72
12	14.58±01.06	23.92±02.39	22.50±00.00	————	10.00±00.00	27.17±02.49	24.33±01.88	24.33±02.63	34.58±01.58
14	12.93±01.09	23.36±01.69	21.43±00.00	————	08.57±00.00	25.21±02.36	22.14±00.89	24.14±02.23	34.86±04.61
16	12.06±00.84	22.00±01.21	21.88±00.00	————	07.50±00.00	24.69±02.29	21.25±01.41	21.94±01.65	30.69±02.53
18	11.44±00.65	20.94±01.17	19.44±00.00	————	06.67±06.00	24.22±01.05	21.06±01.21	21.33±01.18	29.28±03.78
20	10.60±00.57	21.10±01.76	20.00±00.00	————	06.00±00.00	23.65±02.59	20.75±01.32	20.85±01.93	28.25±03.63
Average	20.61	31.45	28.77	————	16.43	33.35	27.75	31.48	43.83
N	Normalized Mutual Information (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	05.19±00.00	26.61±23.96	00.81±00.05	————	05.19±00.00	17.01±16.16	03.86±04.01	14.33±15.97	56.93±36.99
4	07.97±00.86	15.79±08.02	08.93±00.00	————	08.20±00.00	15.80±08.36	07.87±02.82	16.33±08.77	38.29±08.99
6	09.32±00.94	16.06±04.27	17.02±00.00	————	09.56±00.00	23.44±11.20	14.05±03.90	20.78±05.58	41.03±19.23
8	09.57±00.66	20.66±02.71	21.29±00.00	————	10.60±00.00	23.65±04.99	18.06±02.48	18.89±02.97	35.96±04.88
10	09.84±00.74	21.07±02.14	23.72±00.00	————	11.73±00.00	23.88±04.12	11.73±00.00	20.44±03.48	36.60±04.20
12	09.86±00.47	23.22±03.14	24.40±00.00	————	12.36±00.00	28.30±02.50	24.64±02.97	22.14±04.06	36.21±02.61
14	09.77±00.36	25.32±02.41	26.36±00.00	————	11.96±00.00	29.14±02.51	25.36±01.62	24.18±04.22	39.28±04.05
16	09.66±00.15	26.09±01.70	29.41±00.00	————	11.81±00.00	30.63±02.90	26.75±02.27	25.45±01.67	37.04±02.36
18	09.99±00.32	27.10±01.42	28.80±00.00	————	11.86±00.00	31.93±01.43	29.55±01.23	27.93±01.71	38.29±03.80
20	09.87±00.27	28.84±02.03	30.99±00.00	————	12.13±00.00	33.40±03.26	31.41±01.48	28.52±03.42	38.04±03.52
Average	09.10	23.08	21.17	————	10.54	25.72	19.33	21.90	39.77
N	Purity (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	55.00±00.00	72.50±13.18	55.00±00.00	————	55.00±00.00	67.00±12.74	59.00±06.58	66.50±11.56	87.50±11.84
4	32.25±01.42	42.75±05.71	40.00±00.00	————	32.50±00.00	44.75±08.20	38.50±03.16	45.75±05.66	62.25±05.20
6	25.67±00.86	34.33±04.46	33.33±00.00	————	25.00±00.00	39.00±09.07	32.17±03.52	38.17±04.68	52.67±15.40
8	21.75±00.65	30.75±01.88	32.50±00.00	————	21.25±00.00	34.00±04.48	29.38±01.79	30.75±03.02	44.37±04.61
10	19.50±00.71	27.80±02.10	30.00±00.00	————	19.00±00.00	29.80±03.71	19.00±00.00	28.10±02.69	39.60±04.33
12	17.83±00.43	25.92±02.24	23.33±00.00	————	17.50±00.00	29.00±02.45	25.75±01.98	26.33±02.97	36.50±02.07
14	16.64±020.35	25.36±01.62	22.86±00.00	————	16.43±00.00	27.29±02.20	23.43±01.34	25.71±02.40	36.79±04.25
16	15.63±00.00	23.25±01.34	23.75±00.00	————	15.62±00.00	25.94±02.09	22.19±01.77	23.81±01.75	32.31±02.55
18	15.17±00.27	22.33±01.28	20.56±00.00	————	15.00±00.00	25.72±01.26	21.94±01.15	23.11±01.44	31.22±03.48
20	14.60±00.21	22.55±01.72	21.50±00.00	————	14.50±00.00	24.85±02.40	21.90±01.81	22.75±01.44	29.75±03.53
Average	23.40	32.75	30.28	————	23.18	34.73	29.33	33.10	45.30

analysis, with spatial-temporal matrices being the input and clustering a critical technique (Wang et al. 2016).

7 CONCLUSION AND FUTURE WORK

In this article, we propose a novel methodology for integrating and conquering three tasks of finding subspaces, learning manifolds, and building group representations to mutually enhance. The seamlessly integrated learning framework endows the solutions to these tasks a powerful capability of capturing essential information of data and combating noise. More specifically, our novel integrate and conquer methodology works in the following way: The most expressive subspaces with optimal ranks are sought, in which data representations and manifolds are constructed such that the adverse effects of irrelevant features, noise, or corruptions are alleviated, leading to clean, intrinsic manifolds and robust, discriminative representations. Meanwhile, feature extraction in the optimally projected subspaces is also leveraged by finding discriminative data representations on

Table 6. Clustering Performance on 80% Corrupted ORL

N	Accuracy (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	55.00±00.00	60.50±04.38	55.50±01.58	————	55.00±00.00	58.00±03.50	58.00±06.32	62.00±06.75	74.00±10.75
4	30.25±00.79	38.50±03.76	37.50±00.00	————	29.50±03.69	37.75±02.49	37.00±02.30	40.50±04.68	48.75±07.19
6	22.67±00.86	30.50±03.52	31.67±00.00	————	21.83±07.47	31.17±04.45	29.83±02.28	33.17±04.19	40.33±09.68
8	18.75±00.83	27.75±02.55	31.25±00.00	————	15.00±03.95	27.13±02.29	27.00±01.79	29.25±02.96	34.00±02.87
10	15.80±00.63	25.20±02.25	27.00±00.00	————	19.30±07.32	24.30±01.95	11.00±00.00	25.70±02.11	29.80±02.70
12	14.33±00.95	23.50±02.03	22.50±00.00	————	18.25±07.22	22.33±01.79	22.83±01.19	23.67±01.97	27.42±02.79
14	13.07±00.83	22.93±01.76	21.43±00.00	————	17.21±07.54	21.64±01.54	21.86±01.36	22.29±00.74	26.07±02.21
16	12.06±00.59	22.06±01.59	21.88±00.00	————	15.69±07.07	20.75±01.53	20.06±00.95	21.00±01.42	23.94±01.84
18	11.39±00.80	21.17±01.58	19.44±00.00	————	13.11±06.83	19.61±01.85	20.22±01.74	20.11±01.90	23.22±01.92
20	10.65±00.71	20.55±01.12	20.00±00.00	————	13.95±06.87	19.50±00.91	17.95±01.06	20.00±01.03	22.35±01.29
Average	20.40	29.27	28.82	————	21.88	28.22	26.58	29.77	34.99
N	Normalized Mutual Information (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	05.19±00.00	04.04±02.70	01.02±00.71	————	05.19±00.00	05.21±02.82	03.20±03.93	05.66±05.09	25.08±20.82
4	07.75±00.06	09.66±04.10	08.93±00.00	————	07.47±02.06	10.19±02.90	09.27±03.03	09.66±06.27	21.09±10.97
6	08.57±00.11	13.83±03.41	17.02±00.00	————	11.00±03.27	14.32±04.88	13.97±03.14	15.80±05.09	25.99±11.29
8	08.92±00.06	17.29±03.27	21.29±00.00	————	11.36±02.40	17.35±03.25	18.35±02.62	18.74±02.81	27.41±03.53
10	09.26±00.19	21.09±01.59	23.72±00.00	————	17.60±05.30	19.13±03.14	11.73±00.00	22.73±02.40	26.87±02.84
12	09.38±00.08	22.45±01.94	24.40±00.00	————	18.69±05.94	20.83±02.40	23.04±01.68	23.42±02.52	28.17±02.25
14	09.49±00.07	25.95±01.35	26.36±00.00	————	20.45±07.40	23.19±01.65	25.54±01.62	24.91±02.08	29.21±01.62
16	09.59±00.05	27.84±01.45	29.41±00.00	————	21.69±08.55	25.14±02.56	26.26±01.14	27.24±02.37	30.44±02.18
18	09.64±00.06	29.56±01.31	28.80±00.00	————	20.33±08.95	25.88±01.90	28.55±01.50	24.75±03.59	31.26±01.17
20	09.71±00.05	30.68±01.89	30.99±00.00	————	22.74±17.90	27.86±02.21	30.27±01.74	29.47±01.37	32.61±02.02
Average	08.75	20.34	21.19	————	15.65	18.91	19.02	20.34	27.81
N	Purity (%)								
	HCA	k-Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
2	55.00±00.00	60.50±04.38	55.50±01.58	————	55.00±00.00	58.00±03.50	58.00±06.32	62.00±06.75	74.00±10.75
4	32.50±00.00	38.75±03.95	40.00±00.00	————	33.00±01.97	39.00±02.93	38.25±02.65	40.75±04.87	50.00±07.36
6	25.00±00.00	31.83±03.55	33.33±00.00	————	27.17±04.72	32.33±04.10	31.33±02.33	34.00±04.92	42.00±09.19
8	21.25±00.00	28.63±02.24	32.50±00.00	————	21.75±01.58	29.00±02.69	28.00±01.69	30.63±03.45	35.87±02.57
10	19.00±00.00	26.60±01.90	30.00±00.00	————	23.30±03.92	25.60±02.12	19.00±00.00	26.60±02.12	31.40±02.95
12	17.50±00.00	24.83±01.92	23.33±00.00	————	21.75±04.00	23.75±01.81	24.25±01.14	24.75±02.12	29.08±02.53
14	16.43±00.00	24.14±01.87	22.86±00.00	————	21.12±04.34	22.64±01.65	22.50±01.36	23.57±00.95	27.14±02.26
16	15.63±00.00	23.62±01.58	23.75±00.00	————	19.56±03.50	22.19±01.48	21.12±01.17	22.06±01.41	25.31±02.00
18	15.00±00.00	22.39±01.64	20.56±00.00	————	17.83±03.07	20.78±02.02	21.17±01.45	21.72±01.40	24.78±01.84
20	14.50±00.00	21.65±01.18	21.50±00.00	————	17.90±03.00	20.70±00.95	20.60±01.17	20.60±01.07	23.85±01.20
Average	23.18	30.30	30.00	————	25.84	29.40	28.42	30.67	36.34

the intrinsic manifolds, which enables the extracted features and subspaces to maximally preserve 2D spatial information and enhance clustering. To overcome the drawbacks of existing clustering methods on 2D data, we illustrate the use of the proposed integrate and conquer methodology on 2D data to yield a new clustering algorithm, called DTKM. Extensive experiments confirm that DTKM has strong performance on datasets with small or large cluster numbers and fortified robustness to noise. As special matrices, our methodology and algorithm can also be directly applied on vectorial data.

Although the effectiveness of DTKM has been well verified with extensive experimental results, there are still several directions to expand the model. First, it would be interesting to develop theoretical convergence analysis of the proposed algorithm. Second, it has been observed that promising results can be obtained with a small r , which empirically guarantees the efficiency of DTKM. It would be instructional to theoretically investigate how r affects the final clustering performance. Third, we have considered seeking projection directions both horizontally and vertically; as a line

Table 7. Clustering Performance on Faces96 with 60% Corruptions

N	Accuracy (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
10	21.30±06.85	52.80±09.09	27.00±00.00	————	52.80±09.09	66.00±14.27	28.60±01.26	62.60±10.99	72.60±08.49
20	14.10±03.89	47.15±03.67	20.05±00.16	————	47.15±03.67	53.35±07.29	19.55±00.16	47.95±05.96	63.95±03.79
30	11.67±02.26	42.63±04.03	17.33±00.00	————	42.63±04.03	43.97±07.49	18.47±00.63	41.60±04.58	57.00±05.33
40	11.27±01.84	35.28±02.36	16.28±00.08	————	35.28±02.36	42.25±05.13	16.35±00.47	37.45±03.17	47.68±04.17
50	10.94±01.59	35.72±01.92	15.42±00.06	————	35.72±01.92	39.90±04.65	14.76±00.13	34.14±02.76	44.88±03.09
60	10.38±01.66	34.47±01.58	14.53±00.07	————	34.47±01.58	40.07±05.17	14.27±00.32	33.57±02.20	42.43±02.86
70	08.73±01.01	31.67±01.44	14.63±00.10	————	31.67±01.44	35.61±02.90	14.04±00.14	30.53±01.92	38.34±03.51
80	08.14±00.85	32.21±02.10	14.15±00.05	————	32.21±02.10	34.98±02.19	14.30±00.24	29.51±01.00	37.49±02.09
90	08.26±00.49	31.71±01.88	13.67±00.00	————	31.71±01.88	33.80±01.98	13.94±00.18	28.08±01.71	36.61±01.71
100	08.00	30.80	14.00	————	30.80	32.60	13.55	26.00	36.70
Average	11.28	37.44	16.71	————	37.44	42.28	16.78	37.14	47.77
N	Normalized Mutual Information (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
10	17.06±09.39	51.74±10.04	23.84±00.37	————	51.74±10.04	69.44±14.23	22.25±00.94	65.15±13.84	78.56±08.85
20	16.16±04.30	56.00±02.72	31.05±00.13	————	56.00±02.72	64.41±06.02	29.88±00.74	58.24±05.22	73.93±04.36
30	14.45±03.28	54.85±03.09	36.64±00.05	————	54.85±03.09	60.28±06.91	36.22±00.74	57.15±04.10	71.03±03.65
40	14.95±02.82	52.89±01.67	39.94±00.07	————	52.89±01.67	61.87±04.47	38.18±00.16	56.58±03.58	63.27±03.56
50	15.35±02.14	55.31±01.70	43.07±00.03	————	55.31±01.70	62.10±03.42	38.81±00.66	56.68±01.63	61.65±02.30
60	15.46±02.17	56.30±00.98	44.90±00.05	————	56.30±00.98	64.20±03.95	39.50±00.81	58.47±01.36	59.96±01.82
70	13.62±01.43	54.57±01.81	46.68±00.05	————	54.57±01.81	61.67±02.28	41.97±00.32	57.83±01.16	57.87±03.99
80	13.24±01.23	56.52±01.45	48.25±00.02	————	56.52±01.45	61.96±01.51	44.30±00.78	58.82±00.81	61.54±01.60
90	13.49±00.78	56.91±01.40	49.31±00.00	————	56.91±01.40	61.66±01.65	44.35±00.22	58.66±01.10	61.63±01.17
100	13.21	58.29	50.48	————	58.29	61.99	46.47	58.52	60.50
Average	14.70	55.34	41.42	————	55.34	62.88	38.19	58.61	65.00
N	Purity (%)								
	HCA	k -Means	KKM	SC	RPCA	2DPCA	NMF	RMNMF	DTKM
10	24.70±06.96	55.70±08.88	30.00±00.00	————	55.70±08.88	68.40±14.20	29.50±01.58	64.70±11.15	75.70±08.65
20	19.30±03.43	51.15±02.86	21.55±00.16	————	51.15±02.86	56.70±07.06	20.50±00.00	51.10±06.29	67.75±04.16
30	16.27±02.46	45.73±03.46	17.67±00.00	————	45.73±03.46	46.97±07.29	19.37±00.95	44.27±04.35	60.73±04.73
40	15.85±02.04	39.57±02.01	16.53±00.08	————	39.57±02.01	45.27±05.17	16.68±00.24	40.35±03.42	51.10±04.39
50	15.88±01.64	39.70±02.15	15.42±00.06	————	39.70±02.15	42.62±04.45	15.16±00.13	36.58±02.85	48.46±03.32
60	15.48±01.64	38.20±01.27	14.70±00.07	————	38.20±01.27	43.15±05.08	14.48±00.47	35.92±02.08	45.85±02.72
70	14.00±01.07	35.00±01.51	14.77±00.10	————	35.00±01.51	38.64±02.97	14.20±00.18	32.57±01.90	41.21±03.26
80	13.55±00.95	35.98±02.21	14.26±00.04	————	35.98±02.21	37.91±02.11	14.36±00.04	31.50±01.21	40.41±03.26
90	13.53±00.61	35.03±01.57	13.67±00.00	————	35.03±01.57	36.49±01.96	14.08±00.11	29.84±01.74	39.19±01.51
100	13.20	34.50	14.10	————	34.50	34.70	14.20±00.32	29.70	39.40
Average	16.18	41.06	17.27	————	41.06	45.09	17.25	39.47	50.98

of further research, it would be interesting to seek projection directions from other possible directions, e.g., the diagonal directions. Fourth, both P and Q project global 2D features of the data in DTKM. It would be interesting to extend the model to seek projections of local 2D features. Fifth, we have used binary weighting in the formulation; it would be interesting to extend the model to multiple Laplacians, such that the manifolds can be more general. Sixth, we have discussed possible ways to extend our model to using our methodology for higher-order data, which is out of the scope of this article and would be a proper topic for future research.

REFERENCES

- Kais Allab, Lazhar Labiod, and Mohamed Nadif. 2015. Simultaneous semi-NMF and PCA for clustering. In *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM'15)*. IEEE, Los Alamitos, CA, 679–684.
- Sharon Alpert, Meirav Galun, Achi Brandt, and Ronen Basri. 2012. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 2, 315–327.

- Phipps Arabie. 1994. Cluster analysis in marketing research. In *Advanced Methods in Marketing Research*, R. P. Bagozzi (Ed.). Blackwell & Company, Oxford, England, 160–189.
- Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 7, 711–720.
- Alex Bewley and Ben Upcroft. 2013. Advantages of exploiting projection structure for segmenting dense 3D point clouds. In *Proceedings of the Australian Conference on Robotics and Automation*.
- Antoni Buades, Bartomeu Coll, and J.-M. Morel. 2005. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2. IEEE, Los Alamitos, CA, 60–65.
- Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction* 12, 4, 331–370.
- Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8, 1548–1560.
- Deng Cai, Chiyuan Zhang, and Xiaofei He. 2010. Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 333–342.
- Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-view k -means clustering on big data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 2598–2604.
- Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *Journal of the ACM* 58, 3, 11.
- Wei-Chien Chang. 1983. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics* 32, 3, 267–275.
- Fan R. K. Chung. 1997. *Spectral Graph Theory*. Vol. 92. American Mathematical Society.
- Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. 2007. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 11, 1944–1957.
- Chris H. Q. Ding, Tao Li, and Michael I. Jordan. 2010. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 1, 45–55.
- Ehsan Elhamifar and René Vidal. 2009. Sparse subspace clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, Los Alamitos, CA, 2790–2797.
- Roman Filipovych, Susan M. Resnick, and Christos Davatzikos. 2011. Semi-supervised cluster analysis of imaging data. *NeuroImage* 54, 3, 2185–2197.
- Athinodoros S. Georgiades, Peter N. Belhumeur, and David J. Kriegman. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 6, 643–660.
- Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data Mining: Concepts and Techniques*. Elsevier.
- Darryl Hond and Libor Spacek. 1997. Distinctive descriptions for face processing. In *Proceedings of the British Machine Vision Conference (BMVC'97)*. 1–4.
- Andreas Hotho, Steffen Staab, and Gerd Stumme. 2003. Ontologies improve text document clustering. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*. IEEE, Los Alamitos, CA, 541–544.
- Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. 2013. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 9, 2117–2130.
- Jin Huang, Feiping Nie, Heng Huang, and Chris Ding. 2014. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data* 8, 3, 11.
- Ian Jolliffe. 2002. *Principal Component Analysis*. Wiley Online Library.
- Zhao Kang, Chong Peng, and Qiang Cheng. 2017. Kernel-driven similarity learning. *Neurocomputing* 267, C, 210–219.
- Alexander Kolesnikov, Elena Trichina, and Tuomo Kauranne. 2015. Estimating the number of clusters in a numerical data set via quantization error modeling. *Pattern Recognition* 48, 3, 941–952.
- Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755, 788–791.
- Zhouchen Lin, Minming Chen, and Yi Ma. 2010. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055.
- Guangan Liu and Shuicheng Yan. 2011. Latent low-rank representation for subspace segmentation and feature extraction. In *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, Los Alamitos, CA, 1615–1622.
- Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E. Tarjan. 2007. Clustering social networks. In *Proceedings of the International Workshop on Algorithms and Models for the Web-Graph*. 56–67.
- Pabitra Mitra, C. A. Murthy, and Sankar K. Pal. 2002. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 3, 301–312.

- Andrew Y. Ng, Michael I. Jordan, Yair Weiss, and others. 2002. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems* 2, 849–856.
- Vishal M. Patel, Hien Van Nguyen, and René Vidal. 2013. Latent space sparse subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*. 225–232.
- Chong Peng, Zhao Kang, and Qiang Cheng. 2017a. Integrating feature and graph learning with low-rank representation. *Neurocomputing* 249, 106–116.
- Chong Peng, Zhao Kang, Yunhong Hu, Jie Cheng, and Qiang Cheng. 2017b. Nonnegative matrix factorization with integrated graph and feature learning. *ACM Transactions on Intelligent Systems and Technology* 8, 3, 42.
- Chong Peng, Zhao Kang, Yunhong Hu, Jie Cheng, and Qiang Cheng. 2017c. Robust graph regularized nonnegative matrix factorization for clustering. *ACM Transactions on Knowledge Discovery from Data* 11, 3, 33.
- Chong Peng, Zhao Kang, Huiqing Li, and Qiang Cheng. 2015. Subspace clustering using log-determinant rank approximation. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 925–934.
- Chong Peng, Zhao Kang, Ming Yang, and Qiang Cheng. 2016. Feature selection embedded subspace clustering. *IEEE Signal Processing Letters* 23, 7, 1018–1022. DOI: <http://dx.doi.org/10.1109/LSP.2016.2573159>
- Ferdinando S. Samaria and Andy C. Harter. 1994. Parameterisation of a stochastic model for human face identification. In *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*. IEEE, Los Alamitos, CA, 138–142.
- Bernhard Schölkopf. 2001. The kernel trick for distances. In *Proceedings of the 2000 Conference on Advances in Neural Information Processing Systems*, Vol. 13. 301.
- Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10, 5, 1299–1319.
- Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8, 888–905.
- Subhash K. Shinde and Uday Kulkarni. 2012. Hybrid personalized recommender system using centering-bunching based clustering algorithm. *Expert Systems with Applications* 39, 1, 1381–1387.
- Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17, 4, 395–416.
- Jingyuan Wang, Qian Gu, Junjie Wu, Guannan Liu, and Zhang Xiong. 2016. Traffic speed prediction and congestion source exploration: A deep learning method. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, Los Alamitos, CA, 499–508.
- Jim Jing-Yan Wang, Halima Bensmail, and Xin Gao. 2014. Feature selection and multi-kernel learning for sparse representation on a manifold. *Neural Networks* 51, 9–16.
- Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, S. Yu Philip, et al. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems* 14, 1, 1–37.
- Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-Yu Yang. 2004. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1, 131–137.
- Ron Zass and Amnon Shashua. 2005. A unifying approach to hard and probabilistic clustering. In *Proceedings of the 10th International Conference on Computer Vision (ICCV'05)*, Vol. 1. IEEE, Los Alamitos, CA, 294–301.
- Daoqiang Zhang and Zhi-Hua Zhou. 2005. (2D) 2PCA: Two-directional two-dimensional PCA for efficient face representation and recognition. *Neurocomputing* 69, 1, 224–231.
- Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8, 1819–1837.
- Li Zheng, Tao Li, and Chris Ding. 2010. Hierarchical ensemble clustering. In *Proceedings of the 2010 IEEE 10th International Conference on Data Mining (ICDM'10)*. IEEE, Los Alamitos, CA, 1199–1204.

Received May 2017; revised December 2017; accepted February 2018