International Journal of Machine Learning and Cybernetics (2023) 14:2671–2683
https://doi.org/10.1007/s13042-023-01790-0

ORIGINAL ARTICLE

Two-dimensional *k*-subspace clustering and its applications on image recognition

Yan Ru Guo[1] · Yan Qin Bai[2]

Received: 27 June 2022 / Accepted: 20 January 2023 / Published online: 23 February 2023

Abstract

Image clustering plays an important role in computer vision and machine learning. However, most of the existing clustering algorithms flatten the image into one-dimensional vector as an image representation for subsequent learning without fully considering the spatial relationship between pixels, which may lose some useful intrinsic structural information of the matrix data samples and result in high computational complexity. In this paper, we propose a novel two-dimensional *k*-subspace clustering (2D*k*SC). By projecting data samples into a discriminant low-dimensional space, 2D*k*SC maximizes the between-cluster difference and meanwhile minimizes within-cluster distance of matrix data samples in the projected space, thus dimensionality reduction and clustering can be realized simultaneously. The weight between the between-cluster and within-cluster terms is derived from a Bhattacharyya upper bound, which is determined by the involved input data samples. This weighting constant makes the proposed 2D*k*SC adaptive without setting any parameters, which improves the computational efficiency. Moreover, 2D*k*SC can be effectively solved by a standard eigenvalue decomposition problem. Experimental results on three different types of image datasets show that 2DkSC achieves the best clustering results in terms of average clustering accuracy and average normalized mutual information, which demonstrates the superiority of the proposed method.

Keywords Clustering · Subspace · Two-dimensional · Discriminant clustering

1 Introduction

Clustering is one of the most important unsupervised learning topics in machine learning, where data samples are classified into different clusters based on their similarity. It has been studied and applied in many research areas such as text mining [1–5], gene expression [6–9], and image recognition [10–12]. In particular, researchers have used many clustering algorithms for image segmentation [13–16].

Among the various clustering methods, assigning data samples to clusters based on the prototype center of a cluster is one of the most effective and well-studied methods. *k*-means [17] is the most representative and classical clustering method that clusters around center data samples, which clusters all data samples by minimizing the sum of distances from data samples to their nearest cluster prototype. *k*-means works with data samples as the cluster prototype, which often fails when the distributions of data samples are not around several central data samples. In contrast to *k*-means, *k*-plane clustering (*k*PC) [18] and *q*-flat [19] ($0 \leq q \leq m-1$) use hyperplanes or affine subspaces as the entity of the centers and assign each data sample to the nearest hyperplane or $(m-q)$-dimensional affine subspace, where m is the original feature dimension. When the value of q is 0 or $m-1$, q-flat is degraded to *k*-means or *k*PC, respectively. From the above descriptions, *k*-means, *k*PC and *q*-flat only use the structure within-clusters by minimizing some distances within-clusters. However, minimizing the distances within-clusters does not consider the discriminative information between different clusters. The *k*-proximal plane clustering (*k*PPC) [20] introduces the dissimilarity between clusters, which is a great improvement over *k*PC. Twin support vector clustering (TWSVC) [21] and least squares TWSVC (LSTWSVC) [22] also consider between-cluster separability, inspired by the twin support

✉ Yan Qin Bai
yqbai@shu.edu.cn

Yan Ru Guo
Guoyanru211@163.com

1 College of science, Zhejiang University of Science and Technology, HangZhou 310023, China

2 Department of Mathematics, Shanghai University, Shanghai 200444, China

vector machine (TWSVM) [23] and least squares twin support vector machine (LSTWSVM) [24] on classification. To improve robustness, l_1-norm-based clustering methods were also investigated, such as robust TWSVC (RTWSVC) [25], fast RTWSVC (FRTWSVC) [25], and k-subspace discriminant clustering (kSDC) [26].

However, all of the above methods are vector-based ones. If the problem has the matrix data input, a matrix must be converted to a vector before the vector-based methods can be applied. This leads to high-dimensional data and a high computational cost. In addition, some of the underlying structural information is lost. To overcome these shortcomings, a two-dimensional embedded image clustering (A2DEIC) [27] can directly work with matrices instead of flat vectors was recently proposed. However, the objective function of A2DEIC is not smooth and difficult to solve. Moreover, the A2DEIC algorithm is affected by weighting parameters, and finding the optimal parameter is time-consuming. We also notice that though much progress has been made in the field of two-dimensional dimensionality reduction [28–33], little attention has been paid to two-dimensional clustering.

Recently, Li et al. [34] proposed a matrix-based dimensionality reduction method two-dimensional Bhattacharyya bound linear discriminant analysis (2DBLDA). In 2DBLDA, the between-class distance and the within-class distance are weighted by a constant calculated from the input data. This constant helps the objective of 2DBLDA to achieve the minimum Bhattacharyya error bound. Moreover, the design of 2DBLDA avoids the small sample size problem and can be solved by a simple standard eigenvalue decomposition problem. Inspired by the spirit of 2DBLDA, in this paper, we extend 2DBLDA to the clustering problem and propose a novel two-dimensional k-subspace clustering method (2DkSC) that considers both discriminative and underlying structural information. In particular, 2DkSC succeeds in minimizing the similarity within clusters and maximizing the dissimilarity between clusters. Moreover, taking the advantage of 2DBLDA, the cluster data samples are clustered into k-subspaces. In summary, 2DkSC has the following characteristics:

• 2DkSC maximizes the matrix-based between-cluster distance which is measured by the weighted pairwise distances of cluster centers and meanwhile minimizes the matrix-based within-cluster distance, and clusters data samples into these k-subspaces directly. In this way, on the premise of preserving the original matrix data structure, 2DkSC considers both local and discriminative information during clustering by finding the most appropriate reduced dimension for lower dimensional spaces.

• The weighting constant between the between-cluster and within-cluster terms is determined by the involved data that makes the proposed 2DkSC adaptive and without setting any parameters. Inherited from 2DBLDA, the constant is meaningful in the sense that it achieves minimizing the upper bound of Bhattacharyya error.

• From the experimental results of image recognition, 2DkSC has the highest ACC and NMI in five of the six datasets. For example, 2DkSC achieves 77.4% NMI in the Coil100 dataset, which is 3.89% better than the vector-based q-flat algorithm and 6.50% better than the matrix-based A2DEIC algorithm. This phenomenon proves the superiority of our proposed algorithm for image clustering.

The rest of the paper is organized as follows. In section 2, kPC, kPPC, q-flat and A2DEIC are briefly introduced. In section 3, our method is presented. The experiments and conclusions can be found in section 4 and 5, respectively. Details of the weighting constant is provided in the appendix 6.

2 Related works

Given the dataset $T = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$, where $\mathbf{X}_l \in \mathbb{R}^{m\times n}$ for $l = 1, 2, \dots, N$. In particular, if a data sample is vector form, n equals 1. The goal of clustering is to partition T into k disjoint clusters C_i for $i = 1, 2, \dots, k$ satisfying $C_{i'} \cap_{i'\neq i} C_i = \varnothing$ and $T = \cup_{i=1}^{k} C_i$. Correspondingly, $y_l \in \{1, 2, \cdots, k\}$ can be used to indicate the cluster label of the data sample $\mathbf{X}_l$. Assume that the i-th cluster contains N_i data samples. Then $\sum_{i=1}^{k} N_i = N$. Let $\overline{\mathbf{X}}_i = \frac{1}{N_i}\sum_{s=1}^{N_i}\mathbf{X}_s^i$ be the mean of the data samples of the i-th cluster, $i = 1, 2, \dots, k$, where $\mathbf{X}_s^i$ is the s-th data sample of the i-th cluster. For a matrix $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n) \in \mathbb{R}^{m\times n}$, its Frobenius norm (F-norm) $\|\mathbf{Q}\|_F$ is defined as $\|\mathbf{Q}\|_F = \sqrt{\sum_{i=1}^{n}\|\mathbf{q}_i\|_2^2}$. The F-norm is a natural generalization of the vector l_2-norm on matrices.

2.1 *kPC*

kPC [18] divides the data samples into k clusters, so that the data samples gather around their own clustering hyperplane. For the i-th cluster, the hyperplane of kPC is determined by minimizing the sum of the distances between the data samples of the i-th cluster and the hyperplane of the i-th cluster, solving the following programming problem

$$\begin{aligned} \min_{\mathbf{w}_i,b_i} \quad & \|\mathbf{w}_i^T\mathbf{A}_i + b_i\mathbf{e}_i\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{w}_i\|_2^2 = 1, \end{aligned} \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^m$, $b_i \in \mathbb{R}$, $\mathbf{A}_i \in \mathbb{R}^{m\times N_i}$ is the matrix consisting of the data samples labeled i and $\mathbf{e}_i$ is a vector of ones of an appropriate dimension, $i = 1, 2, \dots, k$. The constraint normalizes the normal vector of the hyperplane of the cluster center.

The solution of the problem (1) can be obtained by solving k eigenvalue problems. Once k hyperplanes of the cluster center are obtained, a data sample $\mathbf{x} \in \mathbb{R}^m$ is assigned to the i-th cluster by

$$\text{Cluster}(\mathbf{x}) = \arg\min_{i=1,2,\dots,k} \|\mathbf{w}_i^T\mathbf{x} + b_i\|_2^2, \tag{2}$$

The kPC clustering starts with a random initial assignment of data samples. Each data sample is assigned a label by (2). Then k cluster center hyperplanes are updated by solving (1). The final k cluster center hyperplanes are obtained when the overall objective function does not decrease or the overall assignment of all data samples to cluster center hyperplanes is repeated.

2.2 *k*PPC

In contrast to kPC, kPPC [20] is proposed by introducing between-cluster information into each cluster to construct the cluster hyperplane. kPPC not only requires that the data samples in each cluster be as close as possible to their own center hyperplane, but also pushes the data samples in the other clusters far away from this center hyperplane, solving the following optimization problem

$$\begin{aligned} \min_{\mathbf{w}_i,b_i} \; & \|\mathbf{w}_i^T\mathbf{A}_i + b_i\mathbf{e}_i\|_2^2 - c\|\mathbf{w}_i^T\widehat{\mathbf{A}}_i + b_i\widehat{\mathbf{e}}_i\|_2^2 \\ \text{s.t.} \; & \|\mathbf{w}_i\|_2^2 = 1, \end{aligned} \tag{3}$$

where $\mathbf{A}_i \in \mathbb{R}^{m\times N_i}$ is the matrix consisting of the data samples of label i, and $\widehat{\mathbf{A}}_i \in \mathbb{R}^{m\times(N-N_i)}$ is the matrix consisting of the data samples of the other labels. c is a positive parameter, $\widehat{\mathbf{e}}_i$ is the vector of ones of an appropriate dimension as $\mathbf{e}_i$.

Different from random initialization in kPC, an initialization based on a Laplacian graph-based is constructed in kPPC, which makes kPPC more stable than kPC [20]. kPPC is also solved by an eigenvalue problem.

2.3 *q*-flat

q-flat [19] aims to partition the data samples into k clusters, each of which is well approximated by minimizing the sum of the squared distances of each data sample to the nearest flat. For the i-th cluster, q-flat minimizes the following problem to find its best fit $(m-q)$-dimensional subspace.

$$\begin{aligned} \min_{\mathbf{W}_i,\boldsymbol{\gamma}_i} \; & \|\mathbf{W}_i^T\mathbf{X}_i - \boldsymbol{\gamma}_i\mathbf{e}_i^T\|_F^2 \\ \text{s.t.} \; & \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}, \end{aligned} \tag{4}$$

where $\mathbf{W}_i \in \mathbb{R}^{m\times q}$, $q \le m$, $\mathbf{I}$ is the identity matrix of an appropriate dimension, and $\boldsymbol{\gamma}_i \in \mathbb{R}^q$, $i = 1, 2, \dots, k$, $\mathbf{e}_i$ is a vector of ones of an appropriate dimension.

In practice, q-flat also assumes a random initial assignment of the data samples and reassigns the data samples with

$$\text{Cluster}(\mathbf{x}) = \arg\min_{i=1,2,\dots,k} \|\mathbf{W}_i^T\mathbf{x} - \boldsymbol{\gamma}_i\|_2^2 \tag{5}$$

after obtaining all $\mathbf{W}_i$ and $\boldsymbol{\gamma}_i$.

Similar to kPC and kPPC, q-flats alternates between updating clusters and assigning clusters to determine k cluster flats and find k clusters.

2.4 A2DEIC

Different from kPC, kPPC, and q-flat, A2DEIC [27] proposes an image clustering algorithm that deals directly with matrix representation. It uses two projection matrices to map the original data samples into a low-dimensional subspace and achieve clustering. Given the image data set T, A2DEIC minimizes the following objective function

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} \; & \sum_{i=1}^{N}\sum_{j=1}^{k} y_{ij}\|\mathbf{U}^T(\mathbf{X}_i - \overline{\mathbf{X}}_j)\mathbf{V}\|_F^2 - \lambda\sum_{i=1}^{N}\|\mathbf{U}^T(\mathbf{X}_i - \overline{\mathbf{X}})\mathbf{V}\|_F^2 \\ \text{s.t.} \; & \mathbf{U}^T\mathbf{U} = \mathbf{I}, \mathbf{V}^T\mathbf{V} = \mathbf{I}, \end{aligned} \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{m\times q_1}$ and $\mathbf{V} \in \mathbb{R}^{n\times q_2}$ are projection matrices mapping the original data samples into a low-dimensional subspace $\mathbb{R}^{q_1\times q_2}$. $y_{ij} \in \{0, 1\}$ denotes the cluster indicator value of data samples $\mathbf{X}_i$. The value is 1 if the data samples $\mathbf{X}_i$ is partitioned into the i-th cluster, and 0 otherwise. $\overline{\mathbf{X}}$ is the mean of all data sample matrices and $\overline{\mathbf{X}}_j$ is the mean of the data samples in the j-th cluster. λ is a positive parameter. A2DEIC is solved through an iterative algorithm.

3 Two-dimensional *k*-subspace clustering

3.1 Problem formulation

When the input data is in matrix (or two-dimensional) form, such as images, vector-based algorithms must convert matrices to vectors, which limits consideration of the spatial relationship between pixels and increases computational complexity. As seen above, A2DEIC is proposed to process data input from the matrix. However, the behavior of A2DEIC is greatly affected by its tuning parameters, and its optimization problem is complicated to solve. Inspired by the spirit of 2DBLDA, we propose a new two-dimensional k-subspace clustering algorithm (2DkSC) for image matrices. Inheriting from 2DBLDA, 2DkSC automatically adapts to the given dataset and no parameters need to be adjusted, which can solve the optimization problem efficiently. Moreover, it realizes simultaneously learning the clustering results in a most discriminant subspace of an appropriate dimension by preserving the original structure information of the image matrix.

Specifically, 2DkSC first initializes the cluster assignment and computes the k subspaces. Then, a new round assignment is performed according to the obtained k subspaces and the whole procedure is repeated. For the i-th cluster, $i = 1, ..., k$, we solve the following optimization problem

$$\begin{aligned} &\min_{\mathbf{W}_i} \ \Delta_i \sum_{s=1}^{N_i} \|\mathbf{W}_i^T(\mathbf{X}_s^i - \overline{\mathbf{X}}_i)\|_F^2 - \frac{1}{N}\sum_{j\neq i}\sqrt{N_iN_j}\|\mathbf{W}_i^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_F^2 \\ &\text{s.t.} \ \ \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}, \end{aligned} \tag{7}$$

where $\mathbf{W}_i \in \mathbb{R}^{m\times d}$ is the projection matrices for the i-th subspace, $d \leq m$, $\Delta_i = \frac{1}{4}\sum_{j\neq i}\frac{\sqrt{N_iN_j}}{N}\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2$ is a weighting constant.

We now give the geometric meaning of the model (7). Minimizing the first term in (7) forces the data samples of the i-th cluster around its own cluster center in its subspace. Minimizing the second term in (7) keeps the centers of two different clusters apart in the projected space, which guarantees the between-cluster separativeness. The weighting constant Δ_i in front of the first term balances the importance between clusters and the importance within clusters, which is derived by minimizing an upper bound of theoretical framework of the Bhattacharyya error bound optimality. The details can be found in the appendix 6. We can observe that 2DkSC can be adapted to different data samples since the weighting constant Δ_i is determined by the given data set. The constraint $\mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}$ ensures that the obtained discrimination directions of the i-th cluster are orthonormal to each other, which ensures minimal redundancy in the projected space.

3.2 Solving algorithm and computational complexity analysis

2DkSC can be solved by the following standard eigenvalue decomposition problem

$$\begin{aligned} &\min_{\mathbf{W}_i} \ \mathbf{tr}(\mathbf{W}_i^T\mathbf{M}_i\mathbf{W}_i) \\ &\text{s.t.} \ \ \mathbf{W}_i^T\mathbf{W}_i = \mathbf{I}, \end{aligned} \tag{8}$$

where

$$\begin{aligned} \mathbf{M}_i = {} & \Delta_i \sum_{s=1}^{N_i}\left(\mathbf{X}_s^i - \overline{\mathbf{X}}_i\right)\left(\mathbf{X}_s^i - \overline{\mathbf{X}}_i\right)^T \\ & - \frac{1}{N}\sum_{j\neq i}\sqrt{N_iN_j}\left(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\right)\left(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\right)^T. \end{aligned} \tag{9}$$

With the initial cluster labels of all data samples, 2DkSC updates the data sample labels and k clustering subspaces alternately. After finding the optimal solution of model (8) for each cluster, a data sample $\mathbf{X}_l$ is relabeled as follows

$$\text{Cluster}(\mathbf{X}_l) = \mathop{\arg\min}_{i=1,2,\dots,k} \|\mathbf{W}_i^T(\mathbf{X}_l - \overline{\mathbf{X}}_i)\|_F^2, \ l = 1, 2, \dots, N \tag{10}$$

and k clusters are updated accordingly. These updated clusters are used to determine new projection directions by model (7). The entire process continues until a repeated assignment of cluster marks is made for all data samples. The clustering process of 2DkSC can be realized by Algorithm 1.

Algorithm 1 2DkSC Algorithm.

Input: Data set $T = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$; cluster number k; maximum iteration number $Itnmax$.
Output: The final cluster labels of all data samples in T.
Process:
1. Set the iteration number $r = 0$ and random initial cluster labels of all data samples $\mathbf{X}_l (l = 1, 2, \ldots, N)$.
2. **Repeat:**
   (a) **Subspace update:**
   Calculate the projection matrix $\mathbf{W}_i$ of the i-th cluster by solving the following problem
   $$\min_{\mathbf{W}_i^T\mathbf{W}_i=\mathbf{I}} \mathbf{tr}(\mathbf{W}_i^T\mathbf{M}_i\mathbf{W}_i)$$
   where $\mathbf{M}_i (i = 1, 2, \ldots, k)$ can be solved by Eq.(9);
   (b) **Assignment update:**
   After obtaining $\mathbf{W}_i$ for each cluster, each data sample $\mathbf{X}_l$ can be assigned a cluster label based on
   $$\arg\min_{i=1,2,\ldots,k} \|\mathbf{W}_i^T(\mathbf{X}_l - \overline{\mathbf{X}}_i)\|_F^2$$
   where $\overline{\mathbf{X}}_i$ is the mean of data samples in the i-th cluster;

Until a repeated overall assignment of the cluster labels for all data samples or reaching maximum iteration number $Itnmax$.

For 2DkSC, the main computational cost is to solve the optimization problems (8). From Algorithm 1, we can see that the main computational cost of 2DkSC is to compute the matrix $\mathbf{M}_i$ and perform its standard eigenvalue decomposition. Its computational complexity is $O(m^3)$. Therefore, the computational complexity for Step (a) in Algorithm 1 is $O(rkm^3)$, where r is the number of iterations and k is the cluster number. The computational complexity for Step (b) is $O(rkmnN)$. Therefore, considering that for high-dimensional data $rknN$ is much smaller than m^2, the computational complexity for 2DkSC is $O(rkm^3)$.

To further illustrate the contribution of our method, we discuss the differences between the proposed 2DkSC and the four closely related methods, kPPC, TWSVC, q-flat and A2DEIC.

(i) *Difference From kPPC, TWSVC and q-flat:* Compared to the vector-based clustering algorithms kPPC, TWSVC and q-flat, the proposed 2DkSC is a matrix-based method. The similarity between kPPC, TWSVC and 2DkSC is that their objective functions both maximize the distance between clusters while minimizing the distance within clusters. However, q-flat minimizes only the distance within clusters. The weighting constant of 2DkSC is derived from the Bhattacharyya error bound and can be adaptively adjusted, while the weighting parameters of kPPC and TWSVC require grid search parameters. In addition, 2DkSC and q-flat can achieve clustering and dimensionality reduction simultaneously, while kPPC and TWSVC do not provide dimensionality reduction, only clustering. 2DkSC and kPPC obtain their solutions by solving eigenvalue problems, while q-flat is solved by the singular value decomposition and TWSVC by two quadratic problems.

(ii) *Difference From A2DEIC:* Although A2DEIC can also directly deal with the matrix subspace, A2DEIC is strongly influenced by its tuning parameters and the search for the optimal parameter is difficult and time-consuming, while 2DkSC does not need to tune any parameters, which can solve the optimization problem efficiently. 2DkSC can solve its optimization problem simply by a standard eigenvalue problem, while A2DEIC solves its optimization problem by an iteration technique.

4 Experiments

We compare the proposed approach with seven related clustering algorithms, including k-means [17], q-flat [19], kPPC [20], TWSVC [21], FRTWSVC [25], kSDC [26], and A2DEIC [27]. All our experiments are performed on a PC computer with an Intel 3.30 GHz CPU and 4 GB RAM memory under Matlab 2017b platform. kPPC and A2DEIC obtain their solutions by solving eigenvalue problems. q-flat is solved by singular value

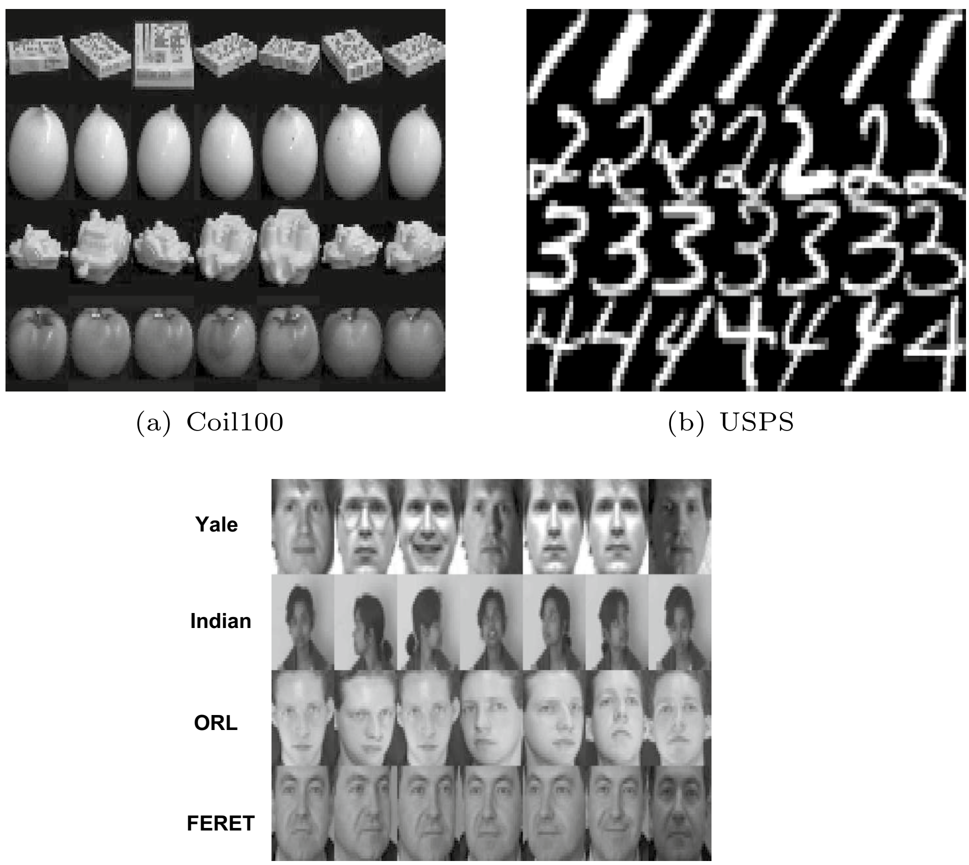
Table 1 The summary of data statistics

Data set	Sample number	Cluster number	Image Size
Coil100	900	100	32×32
USPS	11000	10	16×16
Yale	165	15	32×32
Indian	242	22	32×32
ORL	400	40	32×32
FERET	1400	200	32×32

decomposition. TWSVC is solved by two quadratic problems. FRTWSVC solves a series of linear systems of equations. *k*SDC is solved by an alternating direction method of multipliers. As for the parameter selection, the tuning parameters c in *k*PPC, TWSVC, FRTWSVC and A2DEIC are selected from the set $\{2^{-8}, 2^{-7}, \ldots, 2^{7}\}$. The optimal parameter is selected for all the investigated methods using the grid search technique. k is set to be equal to the ground truth cluster number for each dataset by default. For unknown k, one way is to use the non-parametric Bayesian method [35] to estimate it. Another approach is to run the clustering method on dataset with different number of clusters as input to find its optimum, whose quality can be measured by clustering accuracy or normalized mutual information. Once the optimal parameter is selected, it is used to learn the final clusters. For methods with random initialization, the average clustering result over ten runs are adopted.

4.1 Evaluation metrics

Following most work on clustering, we use clustering accuracy (ACC) and normalized mutual information (NMI) [36–38] as evaluation measures, which are in the range [0, 1]. A larger value indicates more accurate clustering results. Suppose, p_i represents the label predicted by a clustering algorithm and t_i represents the corresponding true label of a data sample $\mathbf{X}_l$. The ACC is defined as follows:



(a) Coil100 (b) USPS

(c) Face images

Fig. 1 Six image datasets

Table 2 ACC(%), CPU time (second) and p value for different algorithms

Data set	k-means ACC Time p value	q-flat ACC Time p value	kPPC ACC Time p value	TWSVC ACC Time p value	FRTWSVC ACC Time p value	kSDC ACC Time p value	A2DEIC ACC Time p value	2DkSC ACC Time p value
Coil100	42.33±1.03	45.00±0.71	14.89±0.63	17.22±0.31	15.11±0.06	39.77±2.17	43.55±0.15	**49.78±0.08**
	6.0122	175.8384	4247.0440	4027.3285	382.5106	372.5817	18.9154	2.5872
	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	–
USPS	40.33±2.51	50.97±4.50	21.97±0.01	22.00±0.17	20.51±0.28	40.39±0.69	57.19±0.02	**59.80±0.16**
	27.5537	299.1357	113.7399	5994.2277	2363.3316	1655.8016	40.7559	10.1032
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0007	–
Yale	50.90±4.27	54.54±9.92	24.24±0.85	23.63±0.86	23.63±1.05	60.00±1.94	**70.30±2.14**	66.28±2.45
	0.7448	26.3238	1724.4260	323.0293	198.9261	1593.941	2.0419	0.3685
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	–
Indian	57.85±3.52	56.19±2.87	18.18±1.17	19.42±0.29	18.60±0.24	57.02±0.73	66.94±1.17	**68.59±1.33**
	0.6209	23.6598	2792.6242	516.1707	280.0512	1776.0424	1.8301	0.4696
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	–
ORL	60.25±3.16	51.58±0.63	16.00±0.88	16.75±0.88	18.16±1.01	57.25±1.21	65.75±0.88	**66.00±2.65**
	1.0773	47.9476	4099.9704	1147.0584	553.7761	1304.7472	4.9167	0.9535
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005	–
FERET	26.42±0.37	23.59±0.04	11.64±0.01	16.00±0.05	16.04±0.38	25.78±0.48	28.53±0.05	**31.93±0.32**
	14.4776	303.1753	1152.8055	5865.2384	2531.8333	1803.2666	29.1946	8.5993
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	–
Average ACC	46.3466	46.9783	17.8200	19.1700	18.6750	46.7017	55.3766	**57.0633**

The bold figure in each row represents the highest ACC

$$ACC = \frac{\sum_{i=1}^{N} \delta(t_i, map(p_i))}{N}, \tag{11}$$

where $\delta(y_1, y_2) = 1$ if $y_1 = y_2$ and $\delta(y_1, y_2) = 0$ otherwise. $map(p_i)$ is the best mapping function that converts clustering labels to match true labels, using the Kuhn-Munkres algorithm [38].

Let us denote by C the set of clusters resulting from the ground truth, and by C' the set resulting from our algorithm. There is a mutual information metric $MI(C, C')$, which is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) log_2 \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)}, \tag{12}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a document arbitrarily selected from the corpus belongs to clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected document belongs to both clusters c_i and c'_j simultaneously. In our experiments, we use the normalized mutual information (NMI) as follows:

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))}, \tag{13}$$

where $H(C)$ and $H(C')$ are the entropies of C and C' respectively. It is easy to verify that $NMI(C, C')$ ranges from 0 to 1. $NMI(C, C') = 1$ if the two groups of clusters are identical, and $NMI(C, C') = 0$ if the two groups are independent.

4.2 Datasets

The experiments are performed on six image datasets, including one object image, one handwritten image, and four face images.

Object recognition: We use the Coil100 dataset [39]. Coil100 contains 900 images with 100 different objects.

Handwritten digit recognition: We use the USPS[1] dataset to evaluate the performance of handwritten digit recognition performance. The dataset contains 11000 samples with 10 classes, where each sample corresponds to one digit.

Face recognition: Four face image datasets (Yale [40], Indian [41], ORL[2] and FERET [42]) are used. The Yale dataset contains 165 images of 15 individuals. The Indian dataset contains 242 human face images of 22 females. The ORL

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html.

[2] https://www.face-rec.org/databases/.

Table 3 NMI(%), CPU time (second) and p value for different algorithms

Data set	*k*-means NMI Time p value	*q*-flat NMI Time p value	*k*PPC NMI Time p value	TWSVC NMI Time p value	FRTWSVC NMI Time p value	*k*SDC NMI Time p value	A2DEIC NMI Time p value	2D*k*SC NMI Time p value
Coil100	65.97±0.52	73.56±0.27	45.43±0.72	54.79±0.15	42.01±1.60	67.27±0.63	70.95±1.18	**77.45±0.56**
	4.6490	300.8458	7931.6413	4007.4016	390.8559	407.6085	10.2174	4.2451
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	–
USPS	40.96±1.69	**54.09±2.67**	18.37±0.02	31.72±0.04	18.86±0.25	36.84±1.41	49.68±0.56	49.43±0.49
	25.1862	239.6224	127.8501	5798.5845	2345.62387	1675.8318	35.4348	10.0422
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3858	–
Yale	61.52±1.90	60.74±4.55	29.51±2.28	27.58±0.49	26.63±0.53	63.42±0.61	65.39±1.38	**68.31±0.22**
	0.6978	20.6587	1726.6077	341.7758	193.9132	1496.0625	1.9495	0.4231
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	–
Indian	71.53±2.04	68.73±1.72	29.93±3.53	32.73±0.59	31.04±0.56	70.48±1.26	75.27±1.09	**76.54±0.58**
	0.5577	21.1221	2582.8441	523.4336	276.5799	1687.8096	3.4018	0.3782
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0072	–
ORL	72.83±1.72	74.52±0.74	35.09±0.66	41.03±0.59	42.52±0.76	78.11±0.22	82.20±0.03	**83.43±0.42**
	1.0623	34.9908	3947.2543	1104.0668	559.0095	1442.8505	5.1351	0.7715
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0491	–
FERET	60.69±0.52	64.77±0.40	36.22±0.03	63.55±0.03	63.48±0.11	59.75±0.68	58.27±1.52	**69.64±0.22**
	12.3350	534.7257	1134.6353	5170.1034	2563.6861	1841.0086	23.3599	8.3462
	0.0000	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000	–
Average NMI	62.2500	66.0683	32.4250	41.9000	35.5100	62.6450	66.9600	**70.8000**

The bold figure in each row represents the highest NMI

dataset contains 400 images of 40 individuals. The FERET dataset contains 14126 images comprising 1199 individuals and 365 duplicate image sets. Here, we use a subset that contains 1400 images of 200 individuals.

The number of samples and categories as well as the image size are listed in Table 1. Some of the gray images are shown in Fig. 1.

4.3 Experimental results

4.3.1 Performance analysis

The results of comparing the performance of different algorithms are shown in Tables 2 and 3, and the best results are indicated in bold figures. The p-value from paired t-test in 5% significance level are adopted. From the experimental results, the following observations can be obtained:

(1) 2D*k*SC achieves the best clustering results in terms of both average clustering ACC and average NMI. Moreover, 2D*k*SC has the highest ACC and NMI in five of the six datasets, respectively. For example, 2D*k*SC achieves 77.45% NMI in the Coil100 dataset, which is 3.89% better than the vector *q*-flat-based method algorithm and 6.50% better than the matrix A2DEIC-based method algorithm. This phenomenon proves the superiority of our proposed algorithm for image clustering.

(2) As a two-dimensional embedding for image clustering, A2DEIC achieves the second best performance in terms of both average clustering ACC and average NMI. The reason is that A2DEIC can directly handle matrix representations. In this way, the spatial information can be preserved in the original data. For example, A2DEIC has the highest ACC of 70.30% on the Yale dataset, 4.02% higher than 2DkSC and 10.30% higher than the vector-based *k*SDC algorithm.

(3) We also find out that *q*-flat performs better than other vector-based algorithms, ranking third in both average accuracy and average NMI. For example, *q*-flat has the highest NMI of 54.09% on the USPS dataset, 4.66% higher than 2DkSC and 13.13% higher than the *k*-means algorithm. Similar to *q*-flat, *k*SDC is also a vector-based clustering algorithm, and its average accuracy and average

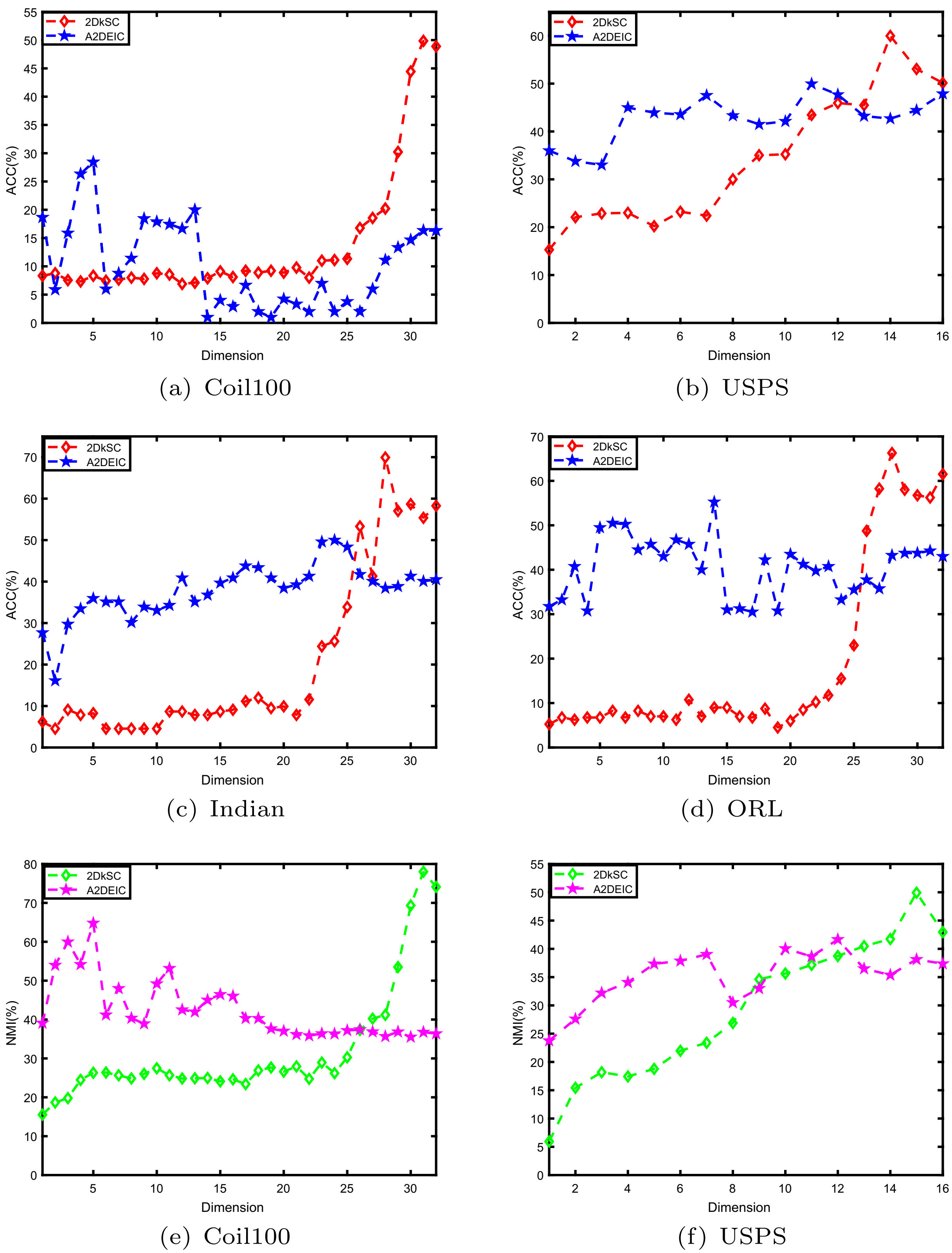


Fig. 2 Clustering results of A2DEIC and 2D*k*SC along different dimensions

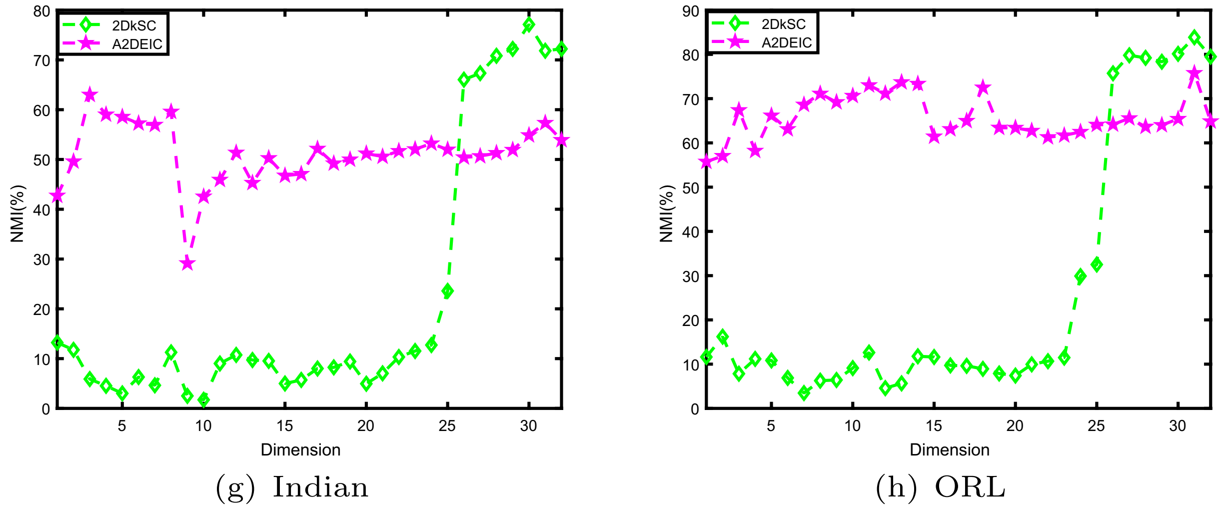


Fig. 2 (continued)

NMI are ranked fourth. The result supports the fact that q-flat and kSDC are able to capture the intrinsic structure in the low-dimensional subspace.

(4) k-means has better performance than the plane-based clustering algorithms kPPC, TWSVC, and FRTWSVC in terms of both average accuracy and average NMI. The performance of TWSVC is better than that of kPPC and FRTWSVC. kPPC has the worst performance.

(5) In terms of CPU time, kPPC, TWSVC and kSDC are slower than other methods. In contrast, 2DkSC costs the least CPU time compared to the seven similar clustering algorithms. This is because 2DkSC does not require any adjusting parameters and the solution can be achieved quickly, which shows the efficiency of our proposed method.

(6) p values bettween 2DkSC and other methods show that on most of the datasets, 2DkSC is statistically different from other methods.

4.3.2 The influence of the dimension

To observe the discriminative ability, the clustering results of 2DkSC along different dimensions are shown in Fig. 2. In Fig. 2, the clustering results of A2DEIC and our 2DkSC are shown when the reduced dimension is set to $d = 1, 2, ..., m$. The results show the following: (i) Although the curve of A2DEIC algorithm is below the optimal parameters, its highest ACC and NMI are not as good as our method. (ii) With the increase of the number of reduced dimensions, ACC and NMI of our 2DkSC vary relatively. (iii) The 2DkSC has the highest results under the optimal reduced dimension on all datasets. (iv) The A2DEIC and 2DkSC are strongly affected by the reduced dimension, and it is necessary to choose an optimal reduced dimension.

5 Conclusion

In this paper, a novel two-dimensional k-subspace clustering method named 2DkSC is investigated. Both discriminative and underlying structural information are embedded in 2DkSC. Therefore, 2DkSC realizes dimensionality and clustering simultaneously. The 2DkSC algorithm has no parameters, its weighting constant can be adaptively adjusted according to the involved data, and the optimization problem has a closed form solution. Experimental results on image recognition have shown the superiority of the proposed method. However, a drawback of 2DkSC is that it may not be very robust to noise since it is based on the squared F-norm. Therefore, we will investigate the robust two-dimensional subspace clustering algorithm in the future. Our MATLAB code can be downloaded from http://www.optimalgroup.org/Resources/Code/2DkSC.html.

Appendix

In the appendix, we present the proof procedure of the relevant Bhattacharyya error bound. It is further explained that the weighting constant Δ_i balances the importance between clusters and the importance within clusters, which is derived by minimizing an upper bound of theoretical framework of the Bhattacharyya error bound optimality.

The Bhattacharyya error [43] is a close upper bound to the Bayes error, which is given by

$$\epsilon_B = \sum_{i<j}^{k} \sqrt{P_i P_j} \int \sqrt{p_i(\mathbf{X}) p_j(\mathbf{X})} d\mathbf{X}, \tag{1}$$

where $\mathbf{X}$ is a data sample, P_i is the prior probability, and $p_i(\mathbf{X})$ is the probability density function of the i-th class of the data.

Proposition 1 *Assume P_i and $p_i(\mathbf{X})$ are the prior probability and the probability density function of the i-th class for the training data set T, respectively, and the data samples in each class are independent and identically normally distributed. Let $p_1(\mathbf{X}), p_2(\mathbf{X}), \dots, p_k(\mathbf{X})$ be the Gaussian functions given by $p_i(\mathbf{X}) = \mathcal{N}(\mathbf{X} \mid \overline{\mathbf{X}}_i, \mathbf{\Sigma}_i)$, where $\overline{\mathbf{X}}_i$ and $\mathbf{\Sigma}_i$ are the class mean and the class covariance matrix, respectively. We further suppose $\mathbf{\Sigma}_i = \mathbf{\Sigma}$, $i = 1, 2, \dots, k$, where $\mathbf{\Sigma}$ is the covariance matrix of the data set T, and $\overline{\mathbf{X}}_i$ and $\mathbf{\Sigma}$ can be estimated accurately from T. Then for arbitrary projection vector $\mathbf{w} \in \mathbb{R}^m$, the Bhattacharyya error bound ϵ_B defined by (1) on the data set $\widetilde{T} = \{\widetilde{\mathbf{X}}_i \mid \widetilde{\mathbf{X}}_i = \mathbf{w}^T\mathbf{X}_i \in \mathbb{R}^{1\times n}\}$ satisfies the following [34]:*

$$\epsilon_B \leq -\frac{a}{8}\sum_{i<j}^{k} \sqrt{P_iP_j}\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 + \frac{a}{8}\Delta \sum_{i=1}^{k}\sum_{s=1}^{N_i} \|\mathbf{w}^T(\mathbf{X}_{is} - \overline{\mathbf{X}}_i)\|_2^2 + \sum_{i<j}^{k}\sqrt{P_iP_j}, \tag{2}$$

where $\Delta = \frac{1}{4}\sum_{i<j}^{k} \frac{\sqrt{N_iN_j}}{N}\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2$, $P_i = \frac{N_i}{N}$, $P_j = \frac{N_j}{N}$, and $a > 0$ is some constant.

Proof We first note that $p_i(\widetilde{\mathbf{X}}) = \mathcal{N}(\widetilde{\mathbf{X}} \mid \widetilde{\overline{\mathbf{X}}}_i, \widetilde{\mathbf{\Sigma}})$, where $\widetilde{\mathbf{X}}_i = \mathbf{w}^T\mathbf{X}_i$, $\widetilde{\overline{\mathbf{X}}}_i = \mathbf{w}^T\overline{\mathbf{X}}_i \in \mathbb{R}^{1\times n}$ is the i-class mean, and $\widetilde{\mathbf{\Sigma}}$ is the covariance matrix in the $1 \times n$ projected space. Denote

$$\mathbf{D} = \begin{pmatrix} \mathbf{w}^T\mathbf{X}_1 \\ \vdots \\ \mathbf{w}^T\mathbf{X}_N \end{pmatrix}^T \in \mathbb{R}^{n\times N} \text{ and } \widetilde{\overline{\mathbf{X}}}_{\mathbf{I}} = \begin{pmatrix} \mathbf{w}^T\overline{\mathbf{X}}_{t_1} \\ \vdots \\ \mathbf{w}^T\overline{\mathbf{X}}_{t_N} \end{pmatrix}^T \in \mathbb{R}^{n\times N}. \tag{3}$$

Then $\widetilde{\mathbf{\Sigma}} = (\mathbf{D} - \widetilde{\overline{\mathbf{X}}}_{\mathbf{I}})(\mathbf{D} - \widetilde{\overline{\mathbf{X}}}_{\mathbf{I}})^T$.

According to [44], we have

$$\int \sqrt{p_i(\widetilde{\mathbf{X}}) p_j(\widetilde{\mathbf{X}})} = e^{-\frac{1}{8}(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)\widetilde{\mathbf{\Sigma}}^{-1}(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)^T}. \tag{4}$$

The upper bound of the error ϵ_B can be estimated as

$$\begin{aligned}
\epsilon_B &= \sum_{i<j}^{k} \sqrt{P_iP_j e^{-\frac{1}{8}(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)\widetilde{\mathbf{\Sigma}}^{-1}(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)^T}} \\
&= \sum_{i<j}^{k} \sqrt{P_iP_j e^{-\frac{1}{8}\|(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)\widetilde{\mathbf{\Sigma}}^{-\frac{1}{2}}\|_2^2}} \\
&\leq \sum_{i<j}^{k} \sqrt{P_iP_j}\left(1 - \frac{a}{8}\|(\widetilde{\overline{\mathbf{X}}}_i - \widetilde{\overline{\mathbf{X}}}_j)\widetilde{\mathbf{\Sigma}}^{-\frac{1}{2}}\|_2^2\right) \\
&= \sum_{i<j}^{k} \sqrt{P_iP_j} - \frac{a}{8}\sum_{i<j}^{k} \sqrt{P_iP_j} \cdot \|(\mathbf{w}^T\overline{\mathbf{X}}_i - \mathbf{w}^T\overline{\mathbf{X}}_j)\widetilde{\mathbf{\Sigma}}^{-\frac{1}{2}}\|_2^2 \\
&\leq \sum_{i<j}^{k} \sqrt{P_iP_j} - \frac{a}{8}\sum_{i<j}^{k} \sqrt{P_iP_j} \cdot \frac{\|(\mathbf{w}^T\overline{\mathbf{X}}_i - \mathbf{w}^T\overline{\mathbf{X}}_j)\|_2^2}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2} \\
&\leq \sum_{i<j}^{k} \sqrt{P_iP_j} - \frac{a}{8}\sum_{i<j}^{k} \sqrt{P_iP_j} \cdot \|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 \\
&\quad + \frac{a}{8}\sum_{i<j}^{k} \sqrt{P_iP_j} \cdot \Delta'_{ij}\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2,
\end{aligned} \tag{5}$$

where $\Delta'_{ij} = \frac{1}{4}\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2$, $a > 0$ is some constant.

For the first inequality of (5), note that the real value function $f(z) = e^{-z}$ is concave when $z \in [0, b]$, $b > 0$; therefore, $e^{-z} \leq 1 - \frac{1-e^{-b}}{b}z$. By taking $a = \frac{1-e^{-b}}{b}$ and noting $\widetilde{\overline{\mathbf{X}}}_i = \mathbf{w}^T\overline{\mathbf{X}}_i$, the first inequality is obtained. For the second inequality, we first note that for any $\mathbf{z} \in \mathbb{R}^{1\times n}$ and an invertible $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\|\mathbf{z}\|_2 = \|(\mathbf{zA})\mathbf{A}^{-1}\|_2 \leq \|\mathbf{zA}\|_2 \cdot \|\mathbf{A}^{-1}\|_F$, which implies $\|\mathbf{zA}\|_2 \geq \frac{\|\mathbf{z}\|_2}{\|\mathbf{A}^{-1}\|_F}$. By taking $\mathbf{z} = \mathbf{w}^T\overline{\mathbf{X}}_i - \mathbf{w}^T\overline{\mathbf{X}}_j$ and $\mathbf{A} = \widetilde{\mathbf{\Sigma}}^{-\frac{1}{2}}$, we get the second inequality. For the last inequality, since $\|\mathbf{w}\|_2 = 1$, $\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 \leq \|\mathbf{w}\|_2^2 \cdot \|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2 = \|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2$ and $\frac{1}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2}\left(1 - \frac{1}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2}\right) \leq \frac{1}{4}$, we have

$$\begin{aligned}
&\left(\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 - \frac{\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2}\right) \cdot \frac{1}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2} \\
&= \|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 \cdot \frac{1}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2}\left(1 - \frac{1}{\|\widetilde{\mathbf{\Sigma}}^{\frac{1}{2}}\|_F^2}\right) \\
&\leq \frac{1}{4}\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 \\
&\leq \frac{1}{4}\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2 \\
&= \Delta'_{ij}.
\end{aligned} \tag{6}$$

which implies

$$-\frac{\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2}{\|\widetilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}\|_F^2} \le -\|\mathbf{w}^T(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j)\|_2^2 + \Delta'_{ij} \cdot \|\widetilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}\|_F^2. \quad (7)$$

By multiplying $\frac{a}{8}\sqrt{P_iP_j}$ to both sides of (7) and summing it over all $1 \le i < j \le k$, we obtain the last inequality of (5).

Take $\Delta = \sum_{i<j}^{k} \sqrt{P_iP_j}\Delta'_{ij} = \frac{1}{4}\sum_{i<j}^{k} \frac{\sqrt{N_iN_j}}{N}\|\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j\|_F^2$, and note that $\|\widetilde{\boldsymbol{\Sigma}}^{\frac{1}{2}}\|_F^2 = \sum_{i=1}^{k}\sum_{s=1}^{N_i}\|\mathbf{w}^T(\mathbf{X}_s^i - \overline{\mathbf{X}}_i)\|_2^2$, we then obtain (2). □

Acknowledgements This work is supported by the National Natural Science Foundation of China (No.12171307) and Zhejiang Soft Science Research Project (No.2021C35003).

References

1. Tan PN, Steinbach M, Kumar V (2005) Introduction to Data Mining. Addison Wesley, Boston
2. Zheng CT (2018) C, Liu, H. San Wong, Corpus based topic diffusion for short text clustering, Neurocomputing 275:2444–2458
3. Abasi AK, Khader AT, Al-Betar MA et al (2020) Link based multi verse optimizer for text documents clustering. Appl Soft Comput 87:106002
4. Costa G, Ortale R (2021) Jointly modeling and simultaneously discovering topics and clusters in text corpora using word vectors. Inf Sci 563:226–240
5. Thirumoorthy K, Muneeswaran K (2021) A hybrid approach for text document clustering using jaya optimization algorithm. Expert Syst Appl 178:115040
6. Jiang Z, Li T, Min W et al (2017) Fuzzy c-means clustering based on weights and gene expression programming. Pattern Recogn Lett 90:1–7
7. Shukla AK, Muhuri PK (2019) Big data clustering with interval type 2 fuzzy uncertainty modeling in gene expression datasets. Eng Appl Artif Intell 77:268–282
8. Zeng YP, Xu ZS, He Y et al (2020) Fuzzy entropy clustering by searching local border points for the analysis of gene expression data. Knowledge Based Systems 190:105309
9. Rahman MA, Ang LM, Seng KP (2020) Clustering biomedical and gene expression datasets with kernel density and unique neighborhood set based vein detection. Inf Syst 91:101490
10. Wang M, Deng WH (2020) Deep face recognition with clustering based domain adaptation. Neurocomputing 393:1–14
11. Liu N, Guo B, Li XJ et al (2021) Gradient clustering algorithm based on deep learning aerial image detection. Pattern Recogn Lett 141:37–44
12. Fang U, Li JX, Lu XQ et al (2021) Self-supervised cross-iterative clustering for unlabeled plant disease images. Neurocomputing 456:36–48
13. Pham TX, Siarry P, Oulhadj H (2018) Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation. Appl Soft Comput 65:230–242
14. Mahata N, Kahali S, Adhikari SK et al (2018) Local contextual information and Gaussian function induced fuzzy clustering algorithm for brain MR image segmentation and intensity inhomogeneity estimation. Appl Soft Comput 68:586–596
15. Lei T, Jia X, Zhang Y et al (2019) Superpixel-based fast fuzzy C-means clustering for color image segmentation. IEEE Trans Fuzzy Syst 27(9):1753–1766
16. Wei D, Wang ZB, Si L et al (2021) An image segmentation method based on a modified local information weighted intuitionistic fuzzy C-means clustering and gold panning algorithm. Eng Appl Artif Intell 101:104209
17. Wu J, Liu H, Xiong H et al (2015) k-means based consensus clustering: a unified view. IEEE Trans Knowl Data Eng 27(1):155–169
18. Bradley PS, Mangasarian OL (2000) k-plane clustering. J Global Optim 16(1):23–32
19. Tseng P (2000) Nearest q-Flat to m Points. J Optim Theory Appl 105:249–252
20. Liu LM, Guo YR, Wang Z et al (2017) k-proximal plane clustering. Int J Mach Learn Cybern 8(5):1537–1554
21. Wang Z, Shao YH, Bai L et al (2015) Twin support vector machine for clustering. IEEE Trans Neural Netw Learn Sys 26(10):2583–2588
22. Khemchandani R, Pal A, Chandra S (2018) Fuzzy least squares twin support vector clustering. Neural Comput Appl 29(2):553–563
23. Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
24. Arun Kumar M, Gopal M (2009) Least squares twin support vector machines for pattern classification. mExpert Sys With Appl 36(4):7535–7543
25. Ye Q, Zhao H, Li Z et al (2017) L1-norm distance minimization-based fast robust twin support vector *k*-plane clustering. IEEE Trans Neural Netw Learn Sys 29(9):4494–4503
26. Li CN, Shao YH, Guo YR et al (2019) Robust k-subspace discriminant clustering. Appl Soft Comput 85:105858
27. Li Z, Yao L, Wang S et al (2020) Adaptive two-dimensional embedded image clustering, Proceedings of the AAAI conference on. Artif Intell 34(04):4796–4803
28. Lu Y, Yuan C, Lai Z et al (2019) Horizontal and vertical nuclear norm based 2DLDA for image representation. IEEE Trans Circuits Syst Video Technol 29(4):941–955
29. Li CN, Shao YH, Deng NY (2015) Robust L1-norm two-dimensional linear discriminant analysis. Neural Netw 65:92–104
30. Li CN, Shang MQ, Shao YH et al (2019) Sparse L1-norm two dimensional linear discriminant analysis via the generalized elastic net regularization. Neurocomputing 337:80–96
31. Lu Y, Yuan C, Lai Z et al (2018) Horizontal and vertical nuclear norm-based 2DLDA for image representation. IEEE Trans Circuits Syst Video Technol 29(4):941–955
32. Li CN, Shao YH, Chen WJ et al (2021) Generalized two-dimensional linear discriminant analysis with regularization. Neural Netw 142:73–91
33. Li CN, Shao YH, Wang Z et al (2019) Robust bilateral Lp-norm two-dimensional linear discriminant analysis. Inf Sci 500:274–297
34. Guo YR, Bai YQ, Li CN et al (2021) Two dimensional Bhattacharyya bound linear discriminant analysis with its applications. Appl Intell 1-17
35. Ma Z, Lai Y, Kleijn WB et al (2019) Variational bayesian learning for dirichlet process mixture of inverted dirichlet distributions in non-gaussian image feature modeling. IEEE Trans Neural Netw Learn Sys 30(2):449–463
36. Cai D, He X, Han J (2005) Document clustering using locality preserving indexing. IEEE Trans Knowl Data Eng 17(12):1624–1637
37. Yang J, Parikh D, Batra D (2016) Joint unsupervised learning of deep representations and image clusters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

(CVPR), Las Vegas, pp 5147–5156. https://doi.org/10.1109/CVPR.2016.556
38. Xie Y, Lin B, Qu Y et al (2020) Joint deep multi-view learning for image clustering. IEEE Trans Knowledge Data Eng 33(11):3594–3606
39. Nene SA, Nayar SK, Murase H (1996) Columbia object image library: Coil-100. Technical Report CUCS-006-96, Department of Computer Science, Columbia University, New York
40. Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
41. Jain V (2002) The Indian face database, http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/
42. Phillips PJ, Moon H, Rizvi SA et al (2000) The FERET evaluation methodology for face-recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22(10):1090–1104
43. Nielsen F (2014) Generalized bhattacharyya and chernoff upper bounds on bayes error using quasi-arithmetic means. Pattern Recogn Lett 42:25–34
44. Fukunaga K (2013) Introduction to statistical pattern recognition. Academic Press, New York