Discriminative Embedded Clustering: A Framework for Grouping High Dimensional Data

Chenping Hou Member, IEEE, Feiping Nie, Dongyun Yi, Dacheng Tao Senior Member, IEEE

Abstract—In many real applications of machine learning and data mining, we are often confronted with high dimensional data. How to cluster high dimensional data is still a challenging problem due to the curse of dimensionality. In this paper, we try to address this problem using joint dimensionality reduction and clustering. Different from traditional approaches which conduct dimensionality reduction and clustering in sequence, we propose a novel framework referred to as Discriminative Embedded Clustering (DEC) which alternates them iteratively. Within this framework, we are able not only to view several traditional approaches and reveal their intrinsic relationships, but also to be stimulated to develop a new method. We also propose an effective approach for solving the formulated non-convex optimization problem. Comprehensive analyses, including convergence behavior, parameter determination and computational complexity, together with the relationship to other related approaches, are also presented. Plenty of experimental results on benchmark data sets illustrate that the proposed method outperforms related state-of-the-art clustering approaches and existing joint dimensionality reduction and clustering methods.

Index Terms—high dimensional data; clustering; dimensionality reduction; subspace learning; discriminative embedded clustering.

I. INTRODUCTION

Clustering is a fundamental and important topic in both machine learning and data mining fields. It has been widely used in many areas, ranging from science to engineering. The basic goal of clustering is to assign data of similar patterns into the same cluster and reveal the meaningful structure of the data. It is one of the most important tools for explorative behavior, parameter determination and computational complexity, together with the relationship to other related approaches, are also presented. Plenty of experimental results on benchmark data sets illustrate that the proposed method outperforms related state-of-the-art clustering approaches and existing joint dimensionality reduction and clustering methods.

Although we can initially reduce the dimensionality by any approach and then use clustering approaches to group high dimensional data, the performance can also be improved since these two techniques are conducted in sequence. The procedure of dimensionality reduction is not related to the subsequent clustering techniques. Intuitively, if we consider the requirement of clustering during the processing of dimensionality reduction and vice versa, the performance of clustering will be improved.

In the past years, several researchers have been dedicated to performing dimensionality reduction and clustering simultaneously. Briefly, there are two different ways: joint feature selection (or ‘feature ranking’) with clustering and joint feature learning (or ‘feature extraction’) with clustering. The first type of method, commonly called subspace clustering, maintains
the original features during the processing of feature selection. In the second category, several features are combined to formulate new representations for clustering. A native approach for joint feature selection with clustering might be to search all possible subspace and use clustering validation to determine the subspace with the best clustering performance. This is infeasible because of the subset generation problem is an NP-problem [19]. Commonly, based on the procedure for searching the optimal subspace, we can divide these methods into two categories: bottom-up subspace searching methods and top-down subspace searching methods. Typical bottom-up methods include CLIQUE [20], ENCLUS [21], MAFIA [22], CBF [23], CLTree [24] and DOC [25]. Typical top-down methods include PROCLUS [26], ORCLUS [27], FINDIT [28], δ-Clusters [29] and COSA [30]. These methods achieve noteworthy performance in many applications. Details are given in [19].

Recently, many researchers have been dedicated to transforming original features and joining feature learning with clustering. Ding et al. have combined Linear Discriminant Analysis (LDA) with K-means by learning a subspace and clustering alternately [31] [32]. They use the K-means algorithm to generate class labels and use LDA to learn the subspace alternately. Torre et al. have shown the benefits of clustering in a low dimensional discriminative space rather than in the principal components and proposed a new clustering algorithm called Discriminative Cluster Analysis (DCA) [33]. Ye et al. have also investigated the problem of alternating clustering and distance metric learning iteratively by projecting data into a low-dimensional manifold, where the separability of the data is maximized [34]. These methods all concern simultaneous subspace selection by LDA and clustering. Theoretical studies have also been provided in [35]. Niu et al. have presented a method which reduces the dimensionality of the original data for spectral clustering. It automatically learns the relevant dimensions and spectral clustering simultaneously [36]. Gu et al. have presented a Subspace Maximum Margin Clustering method (SMMC), which integrates dimensionality reduction with Maximum Margin Clustering in a joint framework [37]. Domeniconi et al. have introduced an algorithm that discovers clusters in subspaces spanned by different combinations of dimensions via local weightings of features [38]. We have also used trace ratio to learn a subspace for face image clustering [39]. Other dimensionality reduction-based clustering methods, such as [40] [41], have also been presented for the simultaneous combination of dimensionality reduction and clustering.

In this paper, we introduce a novel framework referred to as Discriminative Embedded Clustering (DEC), for high dimensional data clustering. It combines subspace learning and clustering in a unified framework. There are two main objective functions, the first being dimensionality reduction and the second being clustering. Several previous methods can be viewed as special cases when we set different values of a parameter of DEC. For example, PCAKM can be viewed as a special case of DEC when \( \lambda \to 0 \). Since the formulated problem is not joint convex with respect to two group parameters, we propose to update them in an alternate way. When we fix one group parameter, the sub-problem is joint convex to the variables, hence alternating minimization can be adopted to obtain the global optimum. We also discuss convergence behavior, together with the relationship with other methods, the parameter determination problem and computational complexity. Plenty of experimental results on different kinds of data sets have been provided for illustration.

It is worth highlighting the novelty of our proposed framework.

1) We propose a unified framework for clustering high dimensional data, which combines subspace learning and clustering in a common procedure.
2) Our proposed framework provides a unified view to analyze and understand many joint subspace learning and clustering methods. Furthermore, it encourages us to develop new methods for clustering high dimensional data.
3) Our work outperforms existing methods on six benchmark data sets, which demonstrates its promising performance in real applications.

The rest of this paper is organized as follows. Section II provides notations and related works. We formulate the proposed DEC framework and provide an effective method of solving this problem in Section III. We discuss the relationship to prior works in Section IV. The performance analyses, including convergence behavior, computational complicity and parameter determination, are presented in Section V. Section VI provides promising comparison results on various kinds of data sets, followed by conclusions and future works in Section VII.

II. RELATED WORK

In this section, we will introduce two kinds of representative methods. The first group contains subspace learning methods which have close relationship to our proposed framework. In the second group, we take LDAKM as the representative method of previous investigations about simultaneous dimensionality reduction and clustering. Before going into the details, let us introduce the notation.

A. Notation

We try to cluster high dimensional data in this paper. Let \( \{ x_i \in \mathbb{R}^D | i = 1, 2, \ldots, n \} \) as the data in high dimensional space and the associated low dimensional representations be \( \{ y_i \in \mathbb{R}^d | i = 1, 2, \ldots, n \} \), where \( D \) and \( d \) are dimensionalities of high dimensional and low dimensional spaces respectively. Define \( \{ f_1, f_2, \ldots, f_n \} \subset \mathbb{R}^p \) as the cluster indicator of \( \{ x_1, x_2, \ldots, x_n \} \) respectively. Here, \( f_i = [f_{i1}, f_{i2}, \ldots, f_{ic}] \). \( f_{ij} = 1 \) if and only if \( x_i \) belongs to the \( j \)-th cluster and \( f_{ij} = 0 \) otherwise. \( c \) is the predefined number of clusters.

Define \( e = [1, 1, \ldots, 1] \in \mathbb{R}^{1 \times n} \) as a row vector of all ones and \( \lambda > 0 \) as a balance parameter. Denote \( X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{D \times n} \), \( Y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^{d \times n} \) and \( F = [f_1^T, f_2^T, \ldots, f_n^T]^T \in \mathbb{R}^{n \times c} \). Define \( Q \in \mathbb{R}^{D \times d} \) as the transformation matrix such that \( Y = Q^T X \). Other notations are summarized in Table I. We will explain the meaning of each term when it is first used.
in the given data. Assume that centroids are taken as representatives of each cluster and the vectors of the input space are transformed by an orthonormal basis of the space spanned by the centroids. OCM tries to search for a dimensionality reduction transformation matrix \( Q \) with which the cluster existing in \( X \) is preserved. Thus, the objective function of OCM is

\[
\max_Q \sum_{i} n_i \|[\tilde{y}_i - \tilde{y}]\|_F^2
\]

s.t. \( Q^TQ = I, Y = Q^TX \),

where \( \tilde{y}_i \) is the centroids for data in the \( i \)-th cluster, \( \tilde{y} \) is the centroids for all data in low dimensional space, \( n_i \) is the number of points in the \( i \)-th category, \( \| \cdot \|_F \) represents the Frobenius norm of a matrix.

After some deduction, the optimization problem of OCM can be reformulated as

\[
\max_Q \text{Tr}(Q^TS_bQ)
\]

s.t. \( Q^TQ = \mathbf{I} \).

Here \( S_b \) is referred to as the between scatter matrix in LDA. It is defined as

\[
S_b = \sum_{i=1}^c n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T,
\]

where \( \bar{x}_i \) is the mean of data in the \( i \)-th cluster.

Similarly, as shown in Eq. (5), the optimal solution to OCM can be derived by eigen-decomposition of \( S_b \).

The third is the MMC-based subspace learning approach, which tries to calculate the most discriminant vectors and to avoid the small sample size problem of LDA. Geometrically, MMC maximizes the average margin between classes. Mathematically, the objective function of MMC is

\[
\max_Q \text{Tr}(Q^T(S_b - S_w)Q)
\]

s.t. \( Q^TQ = \mathbf{I} \).

Here \( S_w \) is referred to as the within scatter matrix, whose definition is

\[
S_w = \sum_{i=1}^c \sum_{x_j \in \mathcal{X}_i} (x_j - \bar{x}_i)(x_j - \bar{x}_i)^T,
\]

where \( \mathcal{X}_i \) contains samples from the \( i \)-th cluster.

Finally, we introduce Orthogonal Least Squares Discriminant Analysis (OLSDA). It is proposed from the perspective of least squares regression. To obtain great discriminative power, the data from the same class are expected to be regressed to a single vector, and the single vector for each class can be reasonably selected as the class centroid after the transformation.

With this objective, OLSDA can be formulated as

\[
\min_Q \|Y - T\|_F^2
\]

s.t. \( Q^TQ = \mathbf{I}, Y = Q^TX \),

where \( T = [t_{i1}, t_{i2}, \ldots, t_{in}] \) is the class centroid for all data points and \( t_i = Q^T\bar{x}_i \). Here \( l_i \) is the class label of \( x_i \).

Define a weighted matrix \( A \) as

\[
A_{ij} = \begin{cases} 
1/n_i, & l_i = l_j \\
0, & \text{otherwise}
\end{cases}
\]

where \( n_i \) is defined as in Eq. (4). It can be checked that \( A \) is an idempotent matrix, i.e., \( A = A^2 \) and the within scatter matrix can be rewritten as \( S_w = X(I - A)X^T \).
The objective function of OLSDA can be reformulated as
\[
\begin{align*}
\|Y - T\|_F^2 &= \|Q^T X - (Q^T x_{i_1}, Q^T x_{i_2}, \ldots, Q^T x_{i_n})\|_F^2 \\
&= \|Q^T X - Q^T X A\|_F^2 \\
&= \text{Tr}(Q^T X(I - A)X^T Q) \\
&= \text{Tr}(Q^T X(I - A)X^T Q) \\
&= \text{Tr}(Q^T S_w Q).
\end{align*}
\]

Then, the optimization problem of OLSDA is
\[
\begin{align*}
\min \text{Tr}(Q^T S_w Q) \\
\text{s.t.} \quad Q^T Q = I.
\end{align*}
\] (9)

In summary, the above-mentioned four methods can be regarded as the variants of LDA, using different kinds of criteria and having different performances. Although these methods consider the separateness of different clusters, they can not be used for dimensionality reduction directly since we do not have labels in the clustering task.

\section{C. LDAKM}

LDAKM is the representative method of joint embedding learning and clustering. It has been widely investigated from various aspects by different researchers. It employs K-means in the embedded subspace to obtain the data clusters and then uses LDA on the original data to derive the transformation. More concretely, it first optimizes the follow objective function of K-means,
\[
\begin{align*}
\min \sum_{i=1}^c \sum_{x_j \in X_i} (y_j - g_i)^T (y_j - g_i),
\end{align*}
\] (10)

where \(g_i\) is the \(i\)-th cluster centroid in the embedded subspace.

After calculating the cluster of each point, we construct \(S_b\) and \(S_w\) as in Eq. (6) and Eq. (8) in the original high dimensional space. LDAKM then uses LDA to compute the transformation matrix by optimizing the following objective function
\[
\max \text{Tr} \left( \frac{Q^T S_b Q}{Q^T S_w Q} \right). \] (11)

With several iterations between K-means and LDA, LDAKM will obtain the clustering results together with the transformation matrix \(Q\). The problem in Eq. (10) can be solved by simply employing a traditional K-means clustering approach and the problem in Eq. (11) is tackled by general eigen-decomposition of \(S_b\) and \(S_w\).

\section{III. DISCRIMINATIVE EMBEDDED CLUSTERING}

We now introduce our algorithm formally. Considering that PCA and K-means are two of the most widely-used methods in dimensionality reduction and clustering, we investigate how to unify them, which motivates our novel framework.

\subsection{A. Formulation}

We now introduce our algorithm formally. Considering that PCA and K-means are two most widely-used methods in dimensionality reduction and clustering, we would like to investigate how to unify them together, which motivates our novel framework.

As stated above, the reason why K-means and LDA can be integrated is that LDA uses the label information derived by K-means. In other words, it is difficult to combine an unsupervised dimensionality reduction approach, such as PCA, with K-means, since unsupervised methods are unable to use label information directly. To tackle this problem, we observe that if we use K-means in the low dimensional subspace, the transformation matrix dominates the performance of K-means. Thus, we propose to share the transformation matrix between two procedures, i.e., dimensionality reduction and clustering, rather than the label information as in LDAKM. To achieve this goal, we propose a framework which unifies many methods. We also present an effective solving method to guarantee convergence.

There are two main objective functions of DEC: the first one concerns dimensionality reduction and the second one concerns the loss of a clustering algorithm. Before going into the details, let us reformulate the objective function of K-means in Eq. (10).

Recalling the basic idea of K-means, we know it tries to minimize the distance of each point to its nearest centroid. Denote \(c\) centroids as \(\{g_1, g_2, \cdots, g_c\} \subset \mathbb{R}^d\) and \(G = [g_1, g_2, \cdots, g_c]\). The corresponding cluster indicators are \(\{f_1, f_2, \cdots, f_n\} \subset \mathbb{R}^c\). \(f_i = [f_{i1}, f_{i2}, \cdots, f_{ic}]\). \(f_{ij} = 1\) if and only if \(x_i\) is assigned to the \(j\)-th cluster and \(f_{ij} = 0\) otherwise. Considering the objective function of K-means in Eq. (10), we reformulate it as
\[
\begin{align*}
\sum_{i=1}^c \sum_{x_j \in X_i} (y_j - g_i)^T (y_j - g_i) \\
= \sum_{i=1}^c \sum_{j=1}^n (y_j - g_i)^T (y_j - g_i) \chi_{X_i}(x_j),
\end{align*}
\] (12)

where \(\chi_{X_i}(x_j)\) is the characteristic function whose definition is
\[
\chi_{X_i}(x_j) = \begin{cases} 
1, & x_j \in X_i \\
0, & x_j \notin X_i
\end{cases} .
\] (13)

Recalling the definition of \(G\) and \(F\), we know that
\[
\begin{align*}
\sum_{i=1}^c (y_j - g_i)^T (y_j - g_i) \\
= (y_j - Gf_j^T)^T (y_j - Gf_j^T).
\end{align*}
\] (14)

Thus, the objective function of K-means in Eq. (12) be-


\[
\sum_{i=1}^{c} \sum_{j=1}^{n} (y_j - g_i)^T (y_j - g_i) \chi_i(x_j)
\]

\[
= \sum_{j=1}^{n} (y_j - Gf_j)^T (y_j - Gf_j)
\]

\[
= \sum_{j=1}^{n} \text{Tr}((y_j - Gf_j)^T (y_j - Gf_j)^T)
\]

\[
= \text{Tr}\left(\sum_{j=1}^{n} (y_j - Gf_j)^T (y_j - Gf_j)^T\right)
\]

\[
= \text{Tr}\left((Y - GFT)^T (Y - GFT)^T\right)
\]

\[
= \|Y - GFT\|_F^2.
\]

Considering the formulation of K-means in Eq. (15), we propose to combine it with PCA. Assuming that the objective functions of PCA in Eq. (1) and K-means in Eq. (15) share the same Q, we induce the constraint \(Y = Q^T X\) into Eq. (15) and combine them by directly adding a balance parameter. Mathematically, the objective function of DEC is

\[
\text{arg max}_{Q,F,G} \text{Tr}(Q^T S_t Q) - \lambda \|Q^T X - GF^T\|_F^2.
\]

As in the previous method, we should add constraints on the optimization variables. First, as in PCA, the columns of Q should be orthogonal, i.e., \(Q^T Q = I\). Then, the parameter F should be an indicator matrix. Each row of F should and must have a single nonzero element. Formally, the formulation of DEC is

\[
\begin{align*}
\text{arg max}_{Q,G,F} & \text{Tr}(Q^T S_t Q) - \lambda \|Q^T X - GF^T\|_F^2 \\
\text{s.t.} & Q^T Q = I
\end{align*}
\]

where F is an indicator matrix.

As we can see from the above formulation, the derivation of Q is not only related to the first objective function of dimensionality reduction, it is also related to the objective function of clustering. In other words, when we reduce the dimensionality of high dimensional data, we also consider the requirement of the subsequent clustering. On the contrary, since we have clustered data in the embedded space, it is directly related to the transformation matrix.

Although the formulation of DEC in Eq. (17) is simple, it is meaningful because:

1. Unlike traditional approaches which share label information by combining dimensionality reduction and clustering, we propose a new way that shares the transformation matrix. This enables joint unsupervised dimensionality reduction and clustering.

2. As we will show in Section IV, the proposed DEC model is a general framework. Other methods can be unified within this framework by the selection of a suitable parameter \(\lambda\).

3. Since the optimization problem in Eq. (17) is not jointly convex with respect to Q, G and F, it is hard to find its global optimal solution. We will propose strategies for solving this problem.

B. Solution

In this section, we try to find an approximated solution to the proposed problem in Eq. (17) since it is not jointly convex with respect to all the variables. In fact, there are two groups of parameters. The first concerns dimensionality reduction, i.e., Q, and the second concerns clustering, i.e., G and F. One direct way is to optimize these two groups alternately. Nevertheless, when Q is fixed, it is hard to find the optimal solution to the formulated problem. Thus, in our solution, we alternate between Q, G and F.

1) Fixing Q, G and optimizing F

As seen from Eq. (17), when Q and G are fixed, the first term in Eq. (17) is constant and it is only necessary to minimize the second term. Since G is fixed and F is constrained to be an indicator, the optimal F is

\[
F_{ij} = \begin{cases} 
1, & j = \arg \min_k \|Q^T x_i - g_k\|_F^2, \\
0, & \text{Otherwise}
\end{cases}
\]

Certainly, when we fix Q and G, DEC is equal to K-means in assigning clusters of embedded points, given the cluster center. Also, when Q and G are fixed, the solution in Eq. (18) is the global optimization of the problem in Eq. (17), since the objective function at each point is minimized when we use Eq. (18) to derive F.

2) Fixing F and optimizing Q, G

When the cluster indicator F is fixed, the optimal Q and G to the problem in Eq. (17) can also be derived in a closed form. We derive the solution by first taking the deviation of the objective function in Eq. (17) with respect to G. For convenience, denote

\[
L(Q,G) = \text{Tr}(Q^T S_t Q) - \lambda \|Q^T X - GF^T\|_F^2.
\]

Then,

\[
\frac{\partial L(Q,G)}{\partial G} = -\lambda \frac{\partial (\text{Tr}(GF^T - Q^TX)(GF^T - Q^TX))}{\partial G}
\]

\[
= -\lambda \frac{\partial (\text{Tr}(GF^TGF^T - 2Tr(GF^TX^TQ)))}{\partial G}
\]

\[
= -2\lambda (GF^TF - Q^TXF).
\]

Let the above equations equal zero, and we have

\[
G = Q^TXF(F^TF)^{-1}.
\]

Induce the derived result in Eq. (21) into Eq. (19), and we have

\[
L(Q) = \text{Tr}(Q^T S_t Q) - \lambda \|Q^T X - Q^TXF(F^TF)^{-1}F^T\|_F^2.
\]

\[
= \text{Tr}(Q^T S_t Q) - \lambda \text{Tr}(Q^TXX^TQ - Q^TXF(F^TF)^{-1}F^TQ)
\]

\[
= \text{Tr}(Q^T(S_t - \lambda XX^T + \lambda XF(F^TF)^{-1}F^TX^T)Q).
\]
The optimization problem in Eq. (17) becomes

\[
\arg \max_Q \mathcal{L}(Q) \quad \text{s.t.} \quad Q^T Q = I. \tag{23}
\]

Based on the results of matrix theory, the optimal solution to the problem in Eq. (23) can be derived by picking up \(d\) eigenvectors corresponding to the \(d\) largest eigenvalues of \(S_t - \lambda XX^T + \lambda X(F^T F)^{-1}F^T X^T\). This solution is the global optimization to the problem in Eq. (23).

In summary, when one group of parameters is fixed, our derived solution is the global optimization to the problem in Eq. (17). We link it to the following proposition.

**Proposition 1.** When \(Q\) and \(G\) are fixed, the derived \(F\) in Eq. (18) is the global solution to the problem in Eq. (17). Similarly, when \(F\) is fixed, the derived \(G\) in Eq. (21) and the derived \(Q\) by picking up eigenvectors corresponding to the \(d\) largest eigenvalues of \(S_t - \lambda XX^T + \lambda X(F^T F)^{-1}F^T X^T\) are also the global solutions to the problem in Eq. (17).

The proof of this proposition is direct. When \(Q\) and \(G\) are fixed, the optimization problem in Eq. (17) is equal to traditional \(K\)-means on \(Q^T X\) with fixed centroid. Thus, the optimal solution is unique. When \(F\) is fixed, \(G\) is determined by \(Q\) and the optimization problem in Eq. (23) has global optimization derived by eigen-decomposition.

Additionally, in employing the above alternation, we find another problem which should be highlighted here. Although the above solving strategy can guarantee convergence to a local minimum, the solution is not satisfied. As in traditional \(K\)-means, there is considerable local optimization which is dominated by initialization [42]. Taking the above updating rule into consideration, when \(F\) is fixed, the algorithm can adjust \(Q\) and \(G\) to fit this \(F\) quickly. In other words, when we need to update \(F\) in the next step, the optimal \(F\) is the same as previously. That is to say, the algorithm converges quickly and the derived optimal solution is dominated by initialization. This phenomenon also occurs in \(K\)-means. To reduce the local optimal problem, we can use the following update rule.

- **Updating Rule 1: Comparison** In each step of updating \(F\), we randomly initialize \(F\) several times (10 in our experiments). If one of the corresponding objective functions \(\|Q_i^T X - GF_i^T\|^2_F\) is smaller than that derived by the previous \(F\), we update \(F\) by this random initialization. Otherwise, we use Eq. (18) to update \(F\). Mathematically, in the \(i\)-th iteration, we derive the optimal \(F^*_i\), \(Q^*_i\) and \(G^*_i\). In the \((i+1)\)-th iteration, the random initializations are \(\{F_{i+1}^1, F_{i+1}^2, \ldots, F_{i+1}^t\}\), where \(t\) is the number of random initializations. We update \(F\) by the following rule:

\[
F^*_{i+1} = \begin{cases} 
F_{i+1}^j, & \|\langle Q_i^T X - G_i^* (F_{i+1}^j)^T \rangle\|^2_F < \\
F_i^*, & \text{otherwise}
\end{cases}
\]

where the elements of \(F^*\) are defined as

\[
F^*_{i,j} = \begin{cases} 
1, & j = \arg \min_k \|\langle Q_i^T X - (g^*_i) k \rangle\|^2_F, \\
0, & \text{otherwise}
\end{cases}
\]

For comparison, we also list another two updating rules.

- **Updating Rule 2: Fixed** In each step of updating \(F\), we use Eq. (18) to update \(F\) directly. Mathematically,

\[
(F_{i+1})_{ij} = \begin{cases} 
1, & j = \arg \min_k \|\langle Q_i^T X - (g_i k) \rangle\|^2_F, \\
0, & \text{otherwise}
\end{cases}
\]

- **Updating Rule 3: Minimization** In each step of updating \(F\), we randomly initialize \(F\) several times (10 in our experiments) and choose \(F\) corresponding to the smallest objective function \(\|Q_i^T X - GF_i^T\|^2_F\). Mathematically,

\[
(F_{i+1})_{ij} = \arg \min_k \|\langle Q_i^T X - G_i^* (F_{i+1}^j)^T \rangle\|^2_F.
\]

As mentioned above, if we use the *Fixed* updating rule, DEC will converge to a local minimization quickly and the result will be dominated by initialization. If we use the *Minimization* updating rule, it cannot guarantee convergence since the objective function may decrease. By contrast, if we use *Comparison*, the above problems can be avoided. We will present a proposition in Section V to show the convergence behavior. In the following experiments, we will use this updating rule without specification.

In summary, the procedure of DEC is listed in Table II.

<table>
<thead>
<tr>
<th>Input:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set: ({x_i</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation matrix (Q) and cluster indicator (F).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Procedure of DEC.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Initialize (Q) by PCA, i.e., solving the problem in Eq. (3).</td>
</tr>
<tr>
<td>Initialize (F) by conducting (K)-means on (Q^T X).</td>
</tr>
<tr>
<td>2. Alternately update (Q), (G) and (F) until convergence.</td>
</tr>
<tr>
<td>a. Update (F) by <em>Comparison</em> rule in Eq. (24).</td>
</tr>
<tr>
<td>b. Update (Q) by picking up eigenvectors corresponding to the (d) largest eigenvalues of (S_t - \lambda XX^T + \lambda X(F^T F)^{-1}F^T X^T).</td>
</tr>
<tr>
<td>Update (G) by Eq. (21).</td>
</tr>
</tbody>
</table>

IV. DISCUSSION IN RELATION TO PRIOR WORK

In this section, we discuss the relation between DEC and PCA, OCM, MMC and OLSDA. Before going into the details, let us first simplify the formulation in Eq. (22).

Note that

\[
S_t = XX^T,
\]

\[
S_b = \sum_{i=1}^n n_i (x_i - \bar{x})(x_i - \bar{x})^T \tag{27}
\]

and the equation \(S_t = S_b + S_w\), we change Eq. (22) as follows,

\[
\text{Tr}(Q^T(S_t - \lambda XX^T + \lambda X(F^T F)^{-1}F^T X^T)Q) = \text{Tr}(Q^T S_t Q) - \lambda \text{Tr}(Q^T (S_t - S_b) Q) = (1 - \lambda) \text{Tr}(Q^T S_b Q) + \lambda \text{Tr}(Q^T S_b Q) = \lambda \text{Tr}(Q^T S_b Q) + (1 - \lambda) \text{Tr}(Q^T S_w Q). \tag{28}
\]

**Example 1.** When \(\lambda \to 0\), DEC is equivalent to performing the two popular methods PCA and \(K\)-means in sequence.
in Fig. 1.

V. PERFORMANCE ANALYSIS

In this section, we will analyze the performance of DEC in three aspects, i.e., the convergence behavior, the computational complexity analysis and the parameter determination.

A. Convergence Analysis

As mentioned above, DEC is solved in an alternate way; namely, we fix one group of variables and optimize the other. As indicated in Proposition 1, we can obtain the global optimization in each alternation. The following proposition shows that our algorithm will monotonically decrease the objective function of the problem in Eq. (17) in each iteration.

**Proposition 2.** The procedures of DEC shown in Table II will monotonically decrease the objective function of the problem in Eq. (17) in each iteration.

**Proof:** Assume that we have derived $Q_i, G_i$ as $Q(i)$ and $G(i)$ in the $i$-th iteration. In the $(i + 1)$-th iteration, we fix $Q, G$ as $Q(i), G(i)$ and optimize $F_{i+1}$ using Eq. (24). Recalling the results in Proposition 1 and the updating rule in Eq. (14), we have the following inequality:

$$\text{Tr}(Q_i^T S_i Q(i)) - \lambda\|Q_i^T X - G(i)F_{i+1}\|_2^2 \\ < \text{Tr}(Q_{i+1}^T S(i+1) Q_{i+1}) - \lambda\|Q_{i+1}^T X - G_{i+1}F_{i+1}\|_2^2.$$

Similarly, when we fix $F$ as $F_{i+1}$ and optimize $Q, G$ by maximizing the objective function in Eq. (19), the following result holds.

$$\text{Tr}(Q_i^T S_i Q(i)) - \lambda\|Q_i^T X - G(i)F_{i+1}\|_2^2 \\ < \text{Tr}(Q_{i+1}^T S Q_{i+1}) - \lambda\|Q_{i+1}^T X - G_{i+1}F_{i+1}\|_2^2.$$

Combining the formulas in Eq. (29) and (30) will result in inequality, which indicates the decrease of the objective function during iteration.

The following point should be highlighted. The final results of DEC are closely related to the initialization, since initialization seriously impacts the final performance [42]. We use two traditional methods, i.e., PCA and $K$-means, to initialize. To avoid rapid convergence to the local minimum, we also randomly initialize several times in updating $F$. Experimental results show that this kind of initialization works well. Therefore, our derived solution may be close to the global optimal solution.

B. Computational Complexity Comparison

In this section, we analyze the computational complexity of different methods. Since PCAKM, OCMKM, MMCKM and OLSDAKM can be unified within our framework, and $K$-means, LDAKM are closely related to DEC, we analyze their computational complexities. Since different implementations of these methods may have different computational complexities, we simply give common analyses.

The first method is $K$-means. It is not a convex optimization problem and its computational complexity is $O(ncD)$, where

**Example 2.** When $\lambda = 1$, DEC is equivalent to using OCM to reduce dimensionality and performing $K$-means on the reduced data alternately.

**Proof:** Considering the optimization problem in Eq. (23) and the equation in Eq. (29), we know that $Q$ is derived by solving the problem with objective function $\text{Tr}(Q^T S Q)$. Recalling the formulation of OCM in Eq. (5), it is clear that they are the same. Considering the updating rule of DEC in Table II, we know that when $\lambda = 1$, DEC is equivalent to alternating between OCM and $K$-means.

**Example 3.** When $\lambda = 2$, DEC is equivalent to using MMC to reduce dimensionality and performing $K$-means on the reduced data alternately.

The proof of this result is the same as that in Example 2. We would like to exclude it.

**Example 4.** When $\lambda \to +\infty$, DEC is equivalent to using OLSDA to reduce dimensionality and performing $K$-means on the reduced data alternately.

The proof of this result is the same as previously. Note that, unlike the scenario in which $\lambda = 0$, our framework alternates when $\lambda \to +\infty$ since the updating of $Q$ is also related to $F$.

In summary, we can view different methods within the proposed framework DEC. Different methods correspond to different selections of parameter $\lambda$. For convenience, we call these methods PCAKM, OCMKM, MMCKM and OLSDAKM respectively. We can also design new methods by choosing suitable parameters within this framework. The relationships of our DEC framework with other related methods are shown
is the number of data points, is the number of clusters and is the dimensionality of the input data. The computational complexity of -means is linear with respect to , and respectively.

The second group of methods consists of PCAKM and LDAKM. They contain two main steps, i.e., eigen-decomposition in deriving and -means in updating . Their computational complexities are and . Since PCAKM does not need iteration, its computational complexity is . Assuming that the number of iterations is , LDAKM has the computational complexity . Note that, unlike traditional -means, the -means clustering in PCAKM and LDAKM are conducted on the data in the embedding subspace.

The third group of methods contains OCMKM, MMCKM, OLSDAKM and DEC. They are all unified within our framework. Their main difference is the concrete form of . They contain two main steps, i.e., eigen-decomposition and -means, as previously. Assuming there are steps in total, their computational complexities are .

As seen from above analyses, the most computational step of these methods is the eigen-decomposition. If the input data is of very high dimensionality, on one hand, we need to use fast eigen-decomposition algorithms, such as Lanczos algorithm[43]. On the other hand, when the dimensionality is very high, we should reduce the dimensionality at first by using algorithm with low computational cost to avoid the curse of dimensionality.

Commonly, the computational complexities of six methods have the following relationships. \( \text{PCAKM} < \text{LDAKM} = \text{OCMKM} = \text{MMCKM} = \text{OLSDAKM} = \text{DEC} \). Nevertheless, different implementations of the same method may have different time costs in real computing. We will provide numerical results in the next section.

C. Parameter Determination

There are two important parameters in our DEC algorithm. The first is the dimensionality of the embedding subspace and the second is the balance parameter . Since parameter determination is still an open problem in the related fields, we determine parameters heuristically and empirically.

The first parameter is used to reveal the intrinsic dimensionality of the original data. When is large, the representation of the original data is still redundant and the curse of dimensionality still occurs. When is too small, there will be overlap of different clusters. We determine this empirically by grid search in this paper by varying this parameter and choosing the one with the best clustering accuracy. The experiments in Section VI(C) will provide numerical results with different .

The second parameter is the balance parameter . Evidently, this parameter balances the effects of dimensionality reduction and clustering in subspace. The larger is, the more attention we pay to clustering. As shown in Fig. 1, several traditional methods can be unified within DEC by selecting suitable parameters. In our methods, we determine heuristically at two levels. We vary this parameter within a large range by large step size and determine the approximate optimal parameter. Subsequently, we vary this parameter near the approximate optimal value with small range and small length of interval. This two-level strategy reduces the number of experiments and finds a suitable . Experimental results show that when we find suitable , DEC outperforms other methods in most cases. This kind of method can help us to find the optimal parameter. See more details in Section VI(F).

VI. Experiments

There are five kinds of experiment in this section. The first group contains experimental result on a toy data. The second group mainly focuses on the evaluation of clustering accuracy. To demonstrate the convergence property, we provide objective function values with different numbers of iterations in the third group. We also show results with different reduced dimensionality by different updating rules. The forth group contains the computational time comparison results. Finally, clustering accuracies with different parameters are presented. We first introduce the data sets and evaluation method.

To show the effectiveness of joint dimensionality reduction and clustering, we have presented the projection directions of DEC with different numbers of iterations in Fig. 2.

A. Data Description and Evaluation Metric

There are six different kinds of matrix data sets, i.e., Umist, Coil, Isolet, Corel, Pollen and Orl. They can be classified into three types. Umist, Coil, Pollen and Orl are used by converting matrix representations of images into vectors. Isolet is voice data collected from 26 individuals. Corel contains images from 30 different topics over all the Corel data sets. They contain the computational time comparison results. Finally, clustering accuracies with different parameters are presented. We first introduce the data sets and evaluation method.

As mentioned above, we compare the performance of DEC with PCAKM, OCMKM, MMCKM and OLSDAKM since they are closely related to DEC. We also provide the results of -means to show whether the dimensionality reduction is

<table>
<thead>
<tr>
<th>Data</th>
<th>Size (n)</th>
<th>Scale (D)</th>
<th>Clusters(c)</th>
<th># of Red Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Umist</td>
<td>575</td>
<td>644</td>
<td>20</td>
<td>5, 6, 14</td>
</tr>
<tr>
<td>Coil</td>
<td>1440</td>
<td>1024</td>
<td>20</td>
<td>5, 6, 14</td>
</tr>
<tr>
<td>Isolet</td>
<td>1559</td>
<td>617</td>
<td>20</td>
<td>5, 10, 50</td>
</tr>
<tr>
<td>Corel</td>
<td>3000</td>
<td>36</td>
<td>30</td>
<td>1, 2, 10</td>
</tr>
<tr>
<td>Pollen</td>
<td>625</td>
<td>625</td>
<td>7</td>
<td>2, 4, 20</td>
</tr>
<tr>
<td>Orl</td>
<td>400</td>
<td>1024</td>
<td>40</td>
<td>20, 21, 29</td>
</tr>
</tbody>
</table>

1http://images.ue.ac.uk/danny/database.html
3http://archive.ics.uci.edu/ml/datasets/ISOLET
4http://kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.data.html
6http://www.uk.research.att.com/facedatabase.html
effective. Since LDAKM is the most famous method which combines dimensionality reduction and clustering, we also show its results.

The performance of different methods is evaluated by Clustering Accuracy. Its is defined as follows

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^{n} \delta(l_i, \text{map}(r_i))$$  \hspace{1cm} (31)

where $\text{map}(\cdot)$ is a function that maps each cluster index to a class label. It can be found by the Hungarian algorithm[45]. $l_i$ is the true class label of $x_i$, $r_i$ is the cluster index derived from $f_i$. $\delta(a, b)$ is the function that equals to 1 when $a$ equals $b$ and 0 otherwise. This metric discovers one-to-one relationships between clusters and the true classes. It measures the extent to which a cluster contains examples from the corresponding category. We compute it by summing the number of matches between all pair clusters. Higher clustering accuracy means better clustering performance.

B. Toy example

In this section, we will give a toy experiments. The data set can be derived into two parts. Every part contains 500 samples. Data points of the left part are generated by Matlab from a two-dimensional normal distribution with mean $[−3, 0]$ and covariance matrix $[1, 0; 0, 20]$. Data points of the right part are generated by a two-dimensional normal distribution with mean $[3, 0]$ and the same covariance matrix.

To show the effectiveness of DEC in combining dimensionality reduction and clustering, we show the projection directions of DEC with different numbers of iterations. The results with iterations number $T = 0$ (Fig.2 (a)), $T = 1$ (Fig.2 (b)), $T = 10$ (Fig.2 (c)) and $T = 20$ (Fig.2 (d)) are shown in Fig. 2. The blue dashed lines are the projection directions derived by DEC. Besides, the corresponding clustering accuracies of DEC are 0.5050, 0.5140, 0.8510 and 0.9980 respectively. In this situation, the mean clustering accuracy of $K$-means with 50 independent random initializations is 0.5984.

As seen from the results shown in Fig. 2, we have the following observations. (1) With the help of clustering, DEC improves the performance of finding suitable projection directions step by step. (2) With the help of dimensionality reduction, the clustering accuracy of DEC also increases steadily.

C. Clustering Accuracy Comparison

In this subsection, we provide the clustering results of different methods on different data sets. By repeating each method for 50 independent runs, we set the reduced dimensionality varying from 1 to 30 as shown in Table III. The balance parameter $\lambda$ is determined heuristically by grid search, as mentioned in Section V(C).

With different data sets and different reduced dimensionality, the mean clustering accuracies of 50 independent runs on all data sets are shown in Fig.3 (a) (Umist data), Fig.3 (b) (Coil data), Fig.3 (c) (Isolet data), Fig.3 (d) (Corel data), Fig.3 (e) (Pollen data) and Fig.3 (f) (Orl data), respectively.

There are several observations from the performance comparisons as follows.

(1) Of the different methods and different data sets, DEC performs best. It achieves the highest clustering accuracy in most cases, especially on the data sets Isolet and Pollen. This is mainly due to the fact that DEC balances the influence of dimensionality reduction and subspace learning in a suitable way.

(2) With the increase of reduced dimensionality, not all methods achieve higher clustering accuracy. This is consistent with intuition, since different data sets have different intrinsic dimensionality.

(3) If we choose suitable parameters in DEC for clustering, joint dimensionality reduction and clustering perform better than sequential methods. Taking the results in Fig. 3 as an example, PCAKM often performs worse than the methods in our framework, which may be because we consider the requirement of clustering in deriving low dimensional embeddings.

(4) When we represent the original data by its low dimensional embedding, it is not always helpful for the next clustering. Taking Fig.3 (a) as an example, we achieve higher clustering accuracy when $K$-means is used to cluster the original data directly compared to the results achieved in using $K$-means to cluster the embedding derived by PCA.
Comparison function values are smaller than those of also seem to be convergent. Nevertheless, their objective algorithm. The other two rules, i.e., Fixed updating rule guarantees the convergence of our rules.

Together with these results, we show the clustering accuracies same as previously given in the experiments in Section VI(C).

We conduct experiments on three representative data sets, Coil, Corel and Pollen. For illustration, we compare DEC with K-means, PCAKM, LDAKM, OCMKM, MMCKM, OLSDA. For impartiality, these methods are all implemented in their original formulation, without the use of other accelerating strategies. With each fixed number of reduced dimensionality, we randomly repeat each experiment for 50 runs. With a naive MATLAB implementation, the calculations are made on a 3.2-GHz Windows machine. The computational time of the different methods is listed in Tables IV, V and VI. The balance parameter \( \lambda \) is also heuristically determined.

We make the following observations from the results in these tables:

(1) Of the different methods on the different data sets, K-means and PCAKM consume the least time. The time cost for DEC is similar to the other iterative methods, such as OCMKM, and the reason for this is that they need to iterate several times.

(2) Although some methods have the same computational complexities, as mentioned in Section V(B), the time cost in
real applications is different. This may be caused by the fact that different methods need different numbers of iterations to guarantee convergence, and the eigen-decomposition of different matrices has different time costs.

F. Clustering with Different Parameters

In this section, we will provide results with different $\lambda$, since it is the most important parameter in our framework.

As mentioned in Section V(C), we determine this parameter in two levels. At the top level, we determine it in a wider range with larger step length and choose a suitable interval. At the lower level, we vary $\lambda$ within this interval and choose the optimal value. This strategy decreases the cost of determining a suitable parameter.

To show the influence of $\lambda$, we conduct experiments on three data sets, i.e., Umist, Coil and Isolet. With the same
setting as previously given in Section VI(C) and different \( \lambda \), we report the mean clustering accuracy in Fig. 5. On the top plane, the range of \( \lambda \) and the length of the step is large. On the lower plane, we vary \( \lambda \) within a smaller range and provided the corresponding clustering accuracy.

As can be seen from the results in Fig. 5, we see that \( \lambda \) dominates the performance of our approach. If it is not suitably selected, performance will be degraded. The efficiency of DEC is also demonstrated, since DEC includes many well-known methods as special cases. We can also see from the bottom plane that when \( \lambda \) varies within a small range, performance does not change drastically. This is helpful for determining this parameter.

VII. CONCLUSION

In this paper, we have proposed an efficient framework, Discriminative Embedded Clustering (DEC), for clustering high dimensional data. In contrast to traditional approaches, we join dimensionality reduction and clustering into a unified framework. Several traditional methods are viewed within this framework and are considered as special cases of our model. The convergence behavior, computational complexity and parameter determination problem have also been analyzed. A significant number of experimental results have been proposed to show the efficiency of DEC. Further research will include the extension of DEC to supervised cases.

REFERENCES


