Learning a subspace for clustering via pattern shrinking

Chenping Hou, Feiping Nie, Yuanyuan Jiao, Changshui Zhang, Yi Wu

Department of Mathematics and Systems Science, National University of Defense Technology, Changsha 410073, China
Department of Automation, Tsinghua University, Beijing 100084, China

Abstract

Clustering is a basic technique in information processing. Traditional clustering methods, however, are not suitable for high dimensional data. Thus, learning a subspace for clustering has emerged as an important research direction. Nevertheless, the meaningful data are often lying on a low dimensional manifold while existing subspace learning approaches cannot fully capture the nonlinear structures of hidden manifold. In this paper, we propose a novel subspace learning method that not only characterizes the linear and nonlinear structures of data, but also reflects the requirements of following clustering. Compared with other related approaches, the proposed method can derive a subspace that is more suitable for high dimensional data clustering. Promising experimental results on different kinds of data sets demonstrate the effectiveness of the proposed approach.

1. Introduction

The task of clustering is to cluster similar data into the same group and divide dissimilar data into different groups. It is an important research direction in many fields, such as statistics, pattern recognition, and information processing. The clustering problem has been addressed by many researchers in many disciplines (Jain, Murty, & Flynn, 1999; Gondek & Hofmann, 2007). This reflects its broad appeal and usefulness in exploratory data mining. It is regarded as a basic preprocessing technique and has been widely used in many real applications. For example, in web searching, if we can cluster different kinds of web texts effectively, it will be convenient for discovering the latent similarities among different web pages. The efficiency of searching for an interested web can also be improved considerably.

Due to the importance of clustering, a lot of researches have been emerged to address this problem. Existing clustering approaches can be broadly classified into four different types (Shi, Song, & Zhang, 2005): partitioning, hierarchical, grid-based and density-based algorithms (Guha, Rastogi, & Shim, 2000; Agrawal, Gehrke, Gunopulos, & Raghavan, 1998; Ankerst, Breunig, Kriegel, & Sander, 1999; Kaufman & Rousseeuw, 1990). These approaches have been widely used and achieved promising results when the samples’ dimensionality is low. In many real applications, e.g., image clustering and web mining, the dimensionality of data is very high. For example, for an image of \( d_1 \times d_2 \) pixels, its vector representation is a \((d_1 \times d_2)\)-dimensional data point (Belhumeur, Hespanha, & Kriegman, 1996). For a web page, its representation is often a high dimensional vector too (Yu, Bi, & Tresp, 2006). In the case of clustering high dimensional data, previous researches have shown that the efficiencies of traditional clustering methods are degraded with increasing dimensions (Ding & Li, 2007; Ye, Zhao, & Wu, 2007).

To cluster high dimensional data, a lot of researches are proposed to first reduce the dimensionality of original data and then to cluster the computed low dimensional embeddings. This is mainly due to the rapid increase of researches about manifold learning, whose aim is to learn a low dimensional representation for high dimensional data. Among these
dimensionality reduction approaches, subspace learning, i.e., reducing the dimensionality linearly, has been widely investigated due to its simplicity, effectiveness and inductive nature (Ding & Li, 2007; Ye et al., 2007). Thus, we will also focus on the subspace learning methods.

Among the approaches that aim to learn a subspace for clustering, the most successful one is to employ the traditional Principle component analysis (PCA) (Duda, Hart, & Stork, 2000) to reduce the dimensionality of original data at first, and then to apply Kmeans (Xu & Wunsch, 2005) for clustering. For brief, we name it PcaKm. Recently, Ding et al. extended the traditional supervised Linear discriminant analysis (LDA) method (Duda et al., 2000) for clustering. It is named as LdaKm (LdaKm) (Ding & Li, 2007). Almost at the same time, Ye et al. also proposed a new approach, which is named as DisKmeans (Ye et al., 2007). Essentially, both LdaKm and DisKmeans compute the clustering results and learn the subspace alternatively. Some other subspace learning approaches, e.g., Locality preserving embedding (LPP), have also been used before clustering (He & Niyogi, 2003). Beside, Li et al. proposed an adaptive subspace iteration algorithm for document clustering (Li, Ma, & Oghara, 2004). Gu et al. also proposed a subspace clustering method based on maximum margin criterion (SMMC) (Gu & Zhou, 2009).

These approaches perform well in many applications. However, their performance can also be improved since (1) some of these approaches, such as PcaKm, separate the procedures of dimensionality reduction and clustering. In other words, the dimensionality reduction approaches are not particularly designed for clustering. (2) Some of these subspace learning approaches, such as LdaKm, adaptive approach and SMMC, use linear projection for dimensionality reduction. They have not taken fully considerations about the nonlinear structure of data points.

In this paper, we aim to find a linear subspace which can incorporate both the linear and nonlinear structures of original data. Besides, it is particular designed for clustering. Inspired by the framework of semi-supervised learning (Zhou, Bousquet, Lal, Weston, & Schölkopf, 2003), a new strategy to shrink the pattern and simultaneously learn the subspace is then proposed. Different from previous subspace learning methods that take the original data as the input, we replace the original data by their shrink patterns. This replacement not only integrates both the linear and nonlinear manifold structures, but also considers the requirements of subsequent clustering procedure. In the following, we name the proposed approach as Subspace Learning via Pattern Shrinking (SLPS).

Taking Fig. 1 as an intuitive example, we shrink the original data (Fig. 1a) to the patterns shown in Fig. 1d. Thus, it is more convenient to find a suitable subspace for clustering. More interestingly, if we change one parameter of our algorithm, i.e., \( \alpha \), the small circle could move outside of the big circle step by step. See more details in Section 3.

This paper is structured as follows. Section 2 presents brief introductions about related works. We describe the proposed subspace learning via pattern shrinking model in Section 3. Section 4 shows some experimental results, followed by the conclusions in Section 5.

2. Related work

In this section, we will review some related works. The first is about two typical subspace learning approaches and the second is mainly about a framework of semi-supervised learning.

2.1. PcaKm and LdaKm

Assume that each data is represented by a \( r \)-dimensional vector and there are totally \( n \) samples for clustering, i.e., \( \{x_1, x_2, \ldots, x_n\} \), where \( x_i \in \mathbb{R}^r \) for \( i = 1, 2, \ldots, n \). In subspace clustering, we aim to compute a transformation matrix \( P \in \mathbb{R}^{r \times d} \), where \( d \) is dimensionality of the subspace. Then, each sample is projected into a low dimensional subspace by \( z_i = P^T x_i \), and \( z_i \) is the embedding of \( x_i \). Denote \( X = [x_1, x_2, \ldots, x_n] \), \( Z = [z_1, z_2, \ldots, z_n] \). Therefore, \( Z = P^T X \). After projecting each data, we can apply some other clustering techniques, such as Kmeans, in the subspace for clustering.

We now introduce two typical subspace learning approaches for clustering, i.e., PcaKm and LdaKm (Ding & Li, 2007). Assume \( F \in \mathbb{R}^{n \times c} \) is the cluster indicator matrix. \( F_{ij} = \frac{1}{\sqrt{n_j}} \) if and only if \( x_i \) belongs to the \( j \)-th cluster. Otherwise, \( F_{ij} = 0 \). Here \( c \) is the number of clusters, \( n_j \) is the number of samples in the \( j \)-th cluster.
The regularized LdaKm algorithm has the following formulation.

number of categories (Duda et al., 2000). In order to learn a subspace of higher dimensionality, we must add a regularizer.

More interestingly, this optimization procedure is equivalent to applying LDA and Kmeans alternatively.

Recall the definition of $F$ and the equation $Z = P^T X$, we rewrite the objective function of Kmeans in Eq. (2) as follows.

In summary, PcaKm first solves the problem in Eq. (1) to compute the optimal projection matrix $P$ and then achieves the clustering results by solving the problem in Eq. (3).

Different from PcaKm, LdaKm (Ding & Li, 2007; Ye et al., 2007) first employs Kmeans approach to cluster the original data and then uses LDA to compute the projection matrix. After projecting each sample, it employs Kmeans to cluster the sample in the low dimensional space and then applies LDA again. After repeating several times, we can compute the clusters and the projection matrix $P$ simultaneously. More concretely, its objective function can be formulated in the following form.

To solve this problem, LdaKm uses the alternative optimization strategy. It fixes one parameter and optimizes the other. More interestingly, this optimization procedure is equivalent to applying LDA and Kmeans alternatively.

One point should be mentioned here. In traditional LDA, the dimensionality of subspace is at most $c - 1$, where $c$ is the number of categories (Duda et al., 2000). In order to learn a subspace of higher dimensionality, we must add a regularizer. The regularized LdaKm algorithm has the following formulation.

where $\lambda$ is a regularization parameter.

2.2. Semi-supervised learning

Since our method is inspired by a framework of some semi-supervised learning methods, we would like to provide a brief introduction about this framework. As shown in Zhou et al. (2003), some semi-supervised learning approaches employ a nonnegative weight $W_{ij}$ to measure the similarity between $x_i$ and $x_j$. The goal of these approaches can be regarded as estimating a classification function $f$ with the following constraints: (1) the computed labels on labeled data should be consistent with the given labels and (2) $f$ should be smooth with respect to the data manifold. This will result in the following objective function.

Here $g$ is a loss function to measure the consistency between computed labels and given labels. $\lambda^{(T)}$ is the labeled data. $S(f)$ is the smooth constraint, which punishes the smoothness of $f$ and usually has the form

$S(f) = f^T S f$

where $f = [f(x_1), f(x_2), \ldots, f(x_n)]^T$ and $S$ is an $n \times n$ smoothness matrix, which has been used in Belkin et al.’s manifold regularization framework (Belkin, Niyogi, & Sindhwani, 2006).

As we will show later, the formulation of pattern shrinking is similar to the function in Eq. (5). However, they are different in essence. Different from semi-supervised learning, since there is no labeled data for clustering, pattern shrinking aims to shrink the original pattern, not to predict the labels.
3. Subspace learning via pattern shrinking

In this section, we will introduce some notations and formulate the subspace learning model by our pattern shrinking technique. After that, we show how to derive the approximated solution in a quick way. Finally, some preliminary discussions are provided.

3.1. Problem formulation

Let us introduce some notations. Assume the shrunk patterns of these samples are defined by \( \{y_1, y_2, \ldots, y_n\} \), where \( y_i \in \mathbb{R}^r \) for \( i = 1, 2, \ldots, n \). In our settings, we plan to compute a projection matrix \( P \). Each sample is projected into a low dimensional subspace by \( z_i = P^r y_i \). Here \( z_i \) is the embedding of \( y_i \) (correspondingly, \( x_i \)) and \( P \in \mathbb{R}^{r \times d} \). Denote \( Y = [y_1, y_2, \ldots, y_n] \), \( Z = [z_1, z_2, \ldots, z_n] \). Therefore, \( Z = P^r Y \). In summary, we would like to list the notations in Table 1.

We begin to show our algorithm step by step. Evoked by the intuition of semi-supervised learning framework in Eq. (5), we try to optimize the following three objective functions.

(1) Recall the requirement and basic assumption of clustering, i.e., nearby points are more likely to belong to the same cluster, we should design a weight matrix whose elements can measure the similarity of any two points and the shrunk patterns should maintain these similarities. Intuitively, if two points are nearby, they should belong to the same cluster and the corresponding weight should be large. On the contrary, if two points are far away, the corresponding element should be small. This objective can be implemented by employing the second term in Eq. (5). Nevertheless, different from the label propagation strategy that can propagate labels along the manifold in semi-supervised learning, we plan to shrink nearby samples as close as possible.

(2) The shrunk pattern should keep the consistency with the original data. More concretely, the shrunk pattern and the original data should not be far away. Compared with the first objective function, which uses the local similarity for shrinking, this objective function can be regarded as keeping dissimilarity since we require that the shrunk pattern should be close to the original data.

(3) After learning the shrunk pattern, we expect to learn a subspace in which the projections of shrunk patterns should maintain their original separations. It guarantees that lose of information in deriving shrunk data is as little as possible.

We now explain the details of each step.

The first step of our approach is to characterize the manifold structure of original data points, i.e., \( \{x_1, x_2, \ldots, x_n\} \). We construct a \( k \)-nearest neighborhood graph, whose nodes are the points and whose edges are constructed by connecting every point to its \( k \) nearest neighbors. Denote that \( N(i) \) is the index set of the points, who are \( k \) nearest neighbors of \( x_i \). The weight matrix \( W \), associated with this \( k \)-nearest neighborhood graph, is computed by the following equation.

\[
W_{ij} = \begin{cases} 
\exp\left(-\frac{(x_i-x_j)^2}{2\sigma^2}\right) & x_j \in N(i) \text{ or } x_i \in N(j) \\
0 & \text{otherwise}
\end{cases}
\]

(7)

where \( \sigma \) is the width parameter of the Gaussian function. Obviously, the weight matrix \( W \) satisfies the requirement of our first objective. There are a couple of other methods which can define the weight matrix \( \text{(Hofmann & Buhmann, 1998; Robles-Kelly & Hancock, 2004)} \), we choose the above strategy since it is simple and widely used \( \text{(Belkin et al., 2006)} \).

Considering the definition of \( W \), we keep the local similarity by minimizing the following equation.

\[
\arg \min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|y_i - y_j\|^2.
\]

(8)

For the second objective, we aim to keep the consistency between shrunk pattern and original data. Intuitively, the shrunk pattern should not be far way from the original data. Besides, we should also keep the data dissimilarity in

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r )</td>
<td>Dimensionality of original space</td>
</tr>
<tr>
<td>( d )</td>
<td>Dimensionality of embedding space</td>
</tr>
<tr>
<td>( x_i \in \mathbb{R}^{r \times d} )</td>
<td>The original high dimensional data</td>
</tr>
<tr>
<td>( y_i \in \mathbb{R}^{r \times r} )</td>
<td>The shrunk pattern of ( x_i )</td>
</tr>
<tr>
<td>( z_i \in \mathbb{R}^{d \times d} )</td>
<td>The projection of ( y_i ) (or ( x_i )), i.e., ( z_i = P^r y_i )</td>
</tr>
<tr>
<td>( P \in \mathbb{R}^{r \times d} )</td>
<td>The projection matrix</td>
</tr>
<tr>
<td>( X )</td>
<td>Data matrix of original samples, i.e., ( X = [x_1, x_2, \ldots, x_n] )</td>
</tr>
<tr>
<td>( Y )</td>
<td>Data matrix of shrunk patterns, i.e., ( Y = [y_1, y_2, \ldots, y_n] )</td>
</tr>
<tr>
<td>( Z )</td>
<td>Data matrix of projections, i.e., ( Z = [z_1, z_2, \ldots, z_n] )</td>
</tr>
</tbody>
</table>
minimizing Eq. (8). It also requires that the shrunk pattern consists with the original data. We minimize the following loss function directly.

$$\arg\min_{\gamma} \sum_{i=1}^{n} \|x_i - y_i\|^2.$$

(9)

For the third objective, we compute the direction in which the variance of projection is as large as possible. In other words, we plan to optimize the following objective.

$$\arg\max_{\gamma} \sum_{i=1}^{n} \left\|z_i - \frac{1}{n} \sum_{j=1}^{n} z_j\right\|^2$$

(10)

As in most cases, we constrain that the transformation matrix is orthogonal, i.e., \( P^TP = I \), and add the parameters \( \alpha \) and \( \beta \) to balance the effects of three terms. In other words, we formulate SLPS as follows.

$$\arg\min_{P^TP = I, Y \rightarrow P^T Y} (1 - \alpha) \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|y_i - y_j\|^2 + \alpha \sum_{i=1}^{n} \|x_i - y_i\|^2 - \beta \sum_{i=1}^{n} \left\|z_i - \frac{1}{n} \sum_{j=1}^{n} z_j\right\|^2.$$

(11)

Here, \( \alpha \) and \( \beta \) are two balance parameters and \( I \) is an identity matrix. We can see that the first two terms aim to derive the shrunk pattern and the last term projects the shrunk pattern to a low dimensional space.

Equivalently, the problem in Eq. (11) can be formulated, with respect to \( P \) and \( Y \), as follows.

$$\arg\min_{P^TP = I, Y \rightarrow P^T Y} (1 - \alpha) \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|y_i - y_j\|^2 + \alpha \sum_{i=1}^{n} \|x_i - y_i\|^2 - \beta \sum_{i=1}^{n} \left\|P^T y_i - \frac{1}{n} \sum_{j=1}^{n} P^T y_j\right\|^2.$$

(12)

We now use the toy experimental results in Fig. 1 to explain why the pattern could shrink along the manifold and SLPS could derive more practical low dimensional representations by minimizing Eq. (11). Intuitively, by optimizing Eq. (11), we can characterize the nonlinear structure of data and considers the requirements of following clustering by learning shrunk pattern, i.e., by minimizing the first two objectives. We can derive a subspace by the joint effects of three terms. In other words, we formulate SLPS as follows.

$$\arg\min_{P^TP = I, Y \rightarrow P^T Y} (1 - \alpha) \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|y_i - y_j\|^2 + \alpha \sum_{i=1}^{n} \|x_i - y_i\|^2 - \beta \sum_{i=1}^{n} \left\|z_i - \frac{1}{n} \sum_{j=1}^{n} z_j\right\|^2.$$

(11)

Here, \( \alpha \) and \( \beta \) are two balance parameters and \( I \) is an identity matrix. We can see that the first two terms aim to derive the shrunk pattern and the last term projects the shrunk pattern to a low dimensional space.

Equivalently, the problem in Eq. (11) can be formulated, with respect to \( P \) and \( Y \), as follows.

$$\arg\min_{P^TP = I, Y \rightarrow P^T Y} (1 - \alpha) \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} \|y_i - y_j\|^2 + \alpha \sum_{i=1}^{n} \|x_i - y_i\|^2 - \beta \sum_{i=1}^{n} \left\|P^T y_i - \frac{1}{n} \sum_{j=1}^{n} P^T y_j\right\|^2.$$

(12)

3.2. Solution

In this subsection, we will solve the formulated problem. For convenience, denote \( L = D - W \), where \( D \) is a diagonal matrix with entries \( D_{ii} = \sum_{j} W_{ij} \). In fact, it is the Laplacian matrix of the above defined \( k \)-nearest graph with weight matrix \( W \) (Belkin et al., 2006). The objective function in Eq. (12), denoted by \( L(P, Y) \), becomes
\[ L(P, Y) = (1 - \alpha)tr(YLY^T) + xtr[(X - Y)^T(X - Y)] - \beta tr(P^TYHY^T)P \]
\[ = (1 - \alpha)tr(YLY^T) + xtr[(X - Y)^T(X - Y)] - \beta tr(Y^T P^T YH) \]  
(13)

where \( H = I - \frac{1}{n}11^T \) is the centralization matrix.

Since this equation is not convex with respect to \( P \) and \( Y \), it is difficult to find both optimal solutions simultaneously. We apply the alternative optimization strategy to find the approximated solutions. More concretely, we fix one parameter and optimize the other, alternatively.

### 3.2.1. Fixing \( P \) and optimizing \( Y \)

We first fix \( P \) and set the derivative of \( L(P, Y) \) with respect to \( Y \) to zero,
\[
\frac{\partial L}{\partial Y} = 2(1 - \alpha)YL + 2x(Y - X) - 2\beta MYH = 0
\]  
(14)

where \( M = PP^T \).

This problem cannot be directly solved by the spectral decomposition technique as in other subspace learning methods, for example, PCA, LDA and LPP. Nevertheless, it is a linear problem with respect to all the elements of \( Y \), i.e., \( Y_{ij} \). More concretely, we expand \( X \) and \( Y \) by connecting each row of \( X \) and \( Y \) and define the expanded vector as follows.
\[
\hat{x} = [X_{11}, X_{12}, \ldots, X_{1r}, X_{21}, \ldots, X_{2r}, \ldots, X_{nr}, \ldots, X_{nr}]
\]
\[
\hat{y} = [Y_{11}, Y_{12}, \ldots, Y_{1r}, Y_{21}, \ldots, Y_{2r}, \ldots, Y_{nr}, \ldots, Y_{nr}]
\]

By reformulating Eq. (14) in terms of \( \hat{x} \) and \( \hat{y} \), we have the following equation.
\[
[(1 - \alpha)L \otimes I]\hat{y} + \hat{x} = \hat{x} - \beta(H \otimes M)\hat{y} = 0
\]  
(15)

where \( \otimes \) represents the Kronecker product of two terms.

Considering the problem in Eq. (15), the optimal solutions can be directly derived as follows.
\[
\hat{y} = \alpha([(1 - \alpha)L \otimes I] + \alpha I - \beta(H \otimes M))^{-1}\hat{x}.
\]
(16)

### 3.2.2. Fixing \( Y \) and optimizing \( P \)

When we fix \( Y \) and optimize \( P \), Eq. (12) becomes
\[
\arg \max_{P \in \mathcal{P}} tr(P^TYHY^TP)
\]  
(17)

This is equivalent to the objective function of PCA. Thus, it can be effectively solved by spectral decomposition of \( YHY^T \).

In summary, we have proposed an approach to solve the optimization problem of SLPS in an alternative way. Comparing with traditional non-iterative subspace learning methods, such as PCA, its computational complexity is a little high, since it needs several times of iterations to obtain the approximated solutions. To reduce the computational complexity, we will show a property of SLPS and propose one-shot SLPS, which can derive an approximated solution without iteration.

Assume \( \hat{Y} \) is formed by reshaping \( \hat{y} \) to a \((r \times n)\) matrix, just as the inverse process of reshaping \( Y \) to \( \hat{y} \), the following proposition shows a property when \( \beta \rightarrow 0 \).

**Proposition 1.** When \( \beta \rightarrow 0 \), \( \hat{Y} = \alpha X[(1 - \alpha)L + \alpha I]^{-1} \)

**Proof.** when \( \beta \rightarrow 0 \), \( \hat{y} = \alpha X[(1 - \alpha)L \otimes I + \alpha I]^{-1}\hat{x} \). Equivalently, \( \hat{Y} = \alpha X[(1 - \alpha)L + \alpha I]^{-1} \)

As seen from above derivation, we can get the same result by solving the problem in Eq. (14) directly. Although this proposition is simple, we can approximately solve the problem in Eq. (12) in another way and derive the one-shot SLPS by solving the following optimization problems in sequence.
\[
Y = \alpha X[(1 - \alpha)L + \alpha I]^{-1} \]
\[
\arg \max_{P \in \mathcal{P}} tr(P^TYHY^TP) \]  
(18)

Note that, the one-shot SLPS is first to compute the shrunk pattern and then apply PCA on the shrunk pattern to derive the final embedding \( Z \). Although the computing and projecting of shrunk pattern are separated in one-shot SLPS, our method could also perform well since (1) the original data are shrunk to their own clusters and it is effective to use PCA to compute the subspace directly and (2) the pattern shrunk technique can also represent the nonlinear structure of original data. Thus, it is reasonable to use PCA on the shrunk patterns. More importantly, compared with solving the problems in Eqs. (15) and (17) alternatively, in one-shot SLPS, the computational requirement is reduced since we can find an approximate solution to the problem in Eq. (12) in one time, without iteration. SLPS, however, can be regarded as solving the problem in one-shot SLPS for several iterations. Thus, in the following experiments, we all use one-shot SLPS for comparison.

In summary, the basic procedure of SLPS is summarized in Table 2.
3.3. Performance analysis

We will analyze the proposed approach in several different aspects. First, we would like to discuss the problem of parameter selection. There are four important parameters: the dimensionality of subspace, i.e., d, the balance parameters α, β and the width parameter σ. Parameter determination is an essential task in most of the learning algorithms (Zelnik-manor & Perona, 2004). The parameter σ is very important since it determines how well the constructed graph can model the low dimensional manifold. As in many related approaches, we apply the technique in Zelnik-manor and Perona (2004) to tune this parameter automatically. More concretely, instead of selecting a single parameter σ, a local scaling parameter σi is calculated for each data point xi by simply measuring the similarity of xi and its kth neighbors. See more details in Zelnik-manor and Perona (2004).

When we refer to the parameter α, it also affects the performance of SLPS, especially for toy examples, e.g., the results in Fig. 1. Among various kinds of methods, grid search is probably the simplest and most widely used one for unsupervised learning. Since we have constrained that 0 < α < 1, the grid search technique is applied in the following experiments. More concretely, we determine α by searching the grid {0.05, 0.1, 0.15, . . . , 0.95} and choose the best one. In one-shot SLPS, since β is omitted, we need not to determine this parameter.

The second concern is about the computational complexity since it is important for real applications. Considering the procedure in previous subsection, we know that the shrunk pattern Y and transformation matrix P can be directly computed by solving the problem in Eqs. (18) and (19), without iteration. The computational complexity is $O((\max(N, r))^3)$. However, this computational complexity is still a little high to solve the problem involving very high dimensional data set with very large volume. We have also used the numerical methods (Vogt, Mizaikoff, & Tacke, 2001), which have accelerated PCA in hyperspectral image processing. Another possible way is to conduct K-means for clustering at first and then use the centroid of clusters as the representative data to reduce the sample size.

Finally, the pattern shrinking technique can be regarded as a preprocessing technique. Other clustering methods are required to compute the final results. As shown in the toy example, since SLPS can effectively shrink the pattern, it will be greatly beneficial to the next clustering. For illustration, we employ two different kinds of clustering approaches, i.e., Kmeans and Spectral Clustering (Normalized Cut (Ncut) (Shi & Malik, 2000) as an example), to cluster the samples in the subspace. For brief, they are named as PsKm and PsNcut respectively.

One point should also be highlighted here is that Shi et al. have also proposed another “shrinking” based approach (Shi et al., 2005). It uses Newtons Universal Law of Gravitation to compute the similarity of any two points and then clusters all data. Different from this approach, our research aims to learn a subspace for clustering and it can inherit the advantages of subspace learning.

In addition, there are several subspace segmentation approaches which are also related to our research. Vidal et al. have proposed an approach, named as Generalized Principal Component Analysis (GPCA) (Vidal, Ma, & Sastry, 2005), to segment an unknown number of subspaces of unknown and varying dimensions. It is mainly used to solve computer vision problems. Ho et al have presented the Ksubspace algorithm (Ho, Yang, Lim, Lee, & Kriegman, 2003) to learn several subspaces for clustering a set of images of 3-D objects. Moreover, Tipping and Bishop have also formulated PCA in a probabilistic framework and computed several subspaces by EM algorithms (Tipping & Bishop, 1999). Comparing with our approach, these methods mainly focus on segmenting subspaces. Our research, however, learns only one subspace. They have different goals, although they all focus on subspace learning.

4. Experiments

In the following experiments, we have employed five different kinds of data sets, including the face image set: Orl,1 the handwritten digit set: Mnist,2 the object image set: Coil20,3 the voice data set: Isolet4 and the web text set: Newsgroup.5

---

2 http://yann.lecun.com/exdb/mnist/.
4 http://archive.ics.uci.edu/ml/datasets/ISOLET.
5 http://people.csail.mit.edu/jrennie/20Newsgroups/.
The character of each data is as follows. We select 100 samples per class from the Mnist data set. Thus, there are totally 1000 images. For Isolet, we select the fifth subset. The Newsgroup data set is preprocessed by Yu et al. (2006). It contains 8014-dimensional TFIDF features and there are totally four different categories, covering autos, motorcycles, baseball and hockey. Since the dimensionality is very high, we only select 200 samples per category. Some typical images from the first three data are shown in Fig. 2.

For illustration, we compare our algorithm (We employ One-shot SLPS and take 'SLPS' for brief) with state of the art subspace learning approaches: PCA, LdaKm and LPP. Since Kmeans is commonly used and Spectral Clustering also employs the similar graph to characterize data structure, we compare PsKm and PsNcut with these approaches (Kmeans and Ncut). Although SLPS is evoked by semi-supervised learning, we have not compared with them since they focus on classification but we try to cluster data. Besides, the parameter $\alpha$ is determined by grid search in the following experiments. Since

![Fig. 2. Sample figures from three image data sets. From top to bottom: Orl, Mnist, Coil20.](image)

![Fig. 3. Toy Examples. The blue line shows the projection direction. (a) Original data and ideal projection direction; (b) original data and PCA projection direction; (c) original data and LdaKm projection direction; (d) original data and LPP projection direction; and (e) shunk pattern and SLPS projecting direction.](image)

![Fig. 4. Toy examples with different $\beta$. The blue line shows the projection direction derived by SLPS with different $\beta$. From (a) to (j), $\beta$ are chosen from 0, 0.01, 0.06, 0.11, ..., 0.41.](image)
parameter determination is still an open problem, other parameters, such as $\sigma$ and $k$ in Eq. (7), are determined by the same way as in other related literatures (Zelnik-manor & Perona, 2004; van der Maaten, Postma, & van den Herik, 2009). More concretely, $\sigma$ is computed by local scaling and $k = 7$ in all experiments.

There are mainly three groups of experiments. The first group contains experimental results on toy examples. We also vary the parameter $\beta$ and compare SLPS with one-shot SLPS intuitively. In the second group, we compare our methods with other approaches quantitatively. Finally, we vary the parameter $\alpha$ and show its influence on the clustering performance.

The first experiment is to show the effectiveness of our SLPS approach on a toy example shown in Fig 3. It contains 80 examples that is sampled from two distinct Gaussian distribution and can be separated linearly. We employ PCA, LdaKm, LPP and SLPS to determine the projecting directions (the blue dashed line).

As seen from this figure, it is obvious that SLPS could shrink the pattern and project the data in a separate way. Other methods fail to discover the right projection directions.

Fig. 5. Clustering results of different methods on the Orl data set with different embedding dimensionality, i.e., $d$; left: The Acc results; right: The NMI results.

Fig. 6. Clustering results of different methods on the Mnist data set with different embedding dimensionality, i.e., $d$; left: The Acc results; right: The NMI results.

For interpretation of color in Figs. 3 and 4, the reader is referred to the web version of this article.
Additionally, since one-shot SLPS is obtained when $\beta = 0$, we choose this parameter from 0, 0.01, 0.06, 0.11, ..., 0.41, and draw the corresponding projection in Fig. 4 intuitively. The other setting are the same as in previous experiments.

As seen from above result, we can draw following conclusion. (1) One-shot SLPS ($\beta = 0$) could approximate the solution of SLPS in a proper way. (2) With different $\beta$, the effects of pattern shrinking and subspace learning are balanced. If it is too large, it will tend to derive the similar projection of PCA.

The second group of experiments is to compare our methods, i.e., PsKm and PsNcut, with other approaches quantitatively. For brief, we name PCA + Kmeans as PcaKm, LPP + Kmeans as LppKm, PCA + Ncut as PcaNcut. We also combine original SLPS with Ncut as SlpsNcut and report its results. We employ two popular metrics: Accuracy (Acc) (Papadimitriou & Steiglitz, 1982) and Normalized Mutual Information (NMI) (Strehl & Ghosh, 2003) to evaluate the performance of clustering. With different $d$, the clustering results, which minimize the loss function of Kmeans for 100 runs, are employed for evaluation. The clustering results on five data sets are shown in Figs. 5–9 respectively.

The main observations from these figures include: (1) Since our method could fully characterize the low dimensional manifold structure and reflects the requirement for clustering by pattern shrinking, the results of PsKm, PsNcut and SlpsNcut
in Figs. 5–9 are better than their corresponding clustering methods and other subspace learning methods in most cases. (2) When the data have distinct manifold structures, for example, the Coil20 data, the improvements are more significant. It means that our method could characterize the manifold structure effectively. (3) By comparing the results of PsKm and PsNcut with Kmeans and Ncut, we know that pattern shrinking is an effective preprocessing technique. For example, on the Newsgroup data, the clustering accuracy is much higher when we reduce the dimensionality. (4) For different dimensionality d, since Kmeans and Ncuts do not involve dimensionality, their results are not changed. Besides, the subspace learning methods also have different performances. Nevertheless, dimensionality reduction is helpful in most cases.

In the third group, we aim to show the influence of parameter a. We have fixed d and employed PsKm on these sets. With different a, changing from 0.1 to 0.9, each run is also repeated for 100 times. Other parameters are determined as in previous experiments. The Acc and NMI results are shown in Figs. 10 and 11.

There are mainly three observations from Figs. 10 and 11. (1) The performance of PsKm depends on the selection of a. When a is suitably selected, PsKm could achieve very high accuracy. (2) With different a, the performance of PsKm is also related to the characters of data sets. For example, the variance between the largest and smallest values for Coil20 is much smaller than that for Newsgroup. (3) For different data, we should select different a to achieve the highest accuracy. Take Coil20 as an example, we should choose the small a. For some data, such as Newsgroup, the larger a is better.
5. Conclusion

In this paper, we proposed a new subspace learning model for clustering. It is mainly based on our proposed pattern shrinking technique. The main idea is to project the original data into a subspace, which is more convenient for clustering. Compared with the state-of-the-art subspace clustering methods, it performs better. Moreover, it can also be regarded as an effective preprocessing technique. In other words, we can improve the performance of other clustering algorithms by simply adding our proposed pattern shrinking technique. Future researches mainly include the accelerating issue of our approach.

Acknowledgements

This work is supported by the 973 Program of China (No. 2013CB329503), NSFC China (No. 61005003, 60975038) and Zhejiang Provincial Natural Science Foundation of China (No. Y1110661).

References


