K-Subspace Clustering

Dingding Wang[1], Chris Ding[2], and Tao Li[1]

[1] School of Computer Science, Florida International Univ., Miami, FL 33199, USA
[2] CSE Department, University of Texas, Arlington, Arlington, TX 76019, USA

Abstract. The widely used K-means clustering deals with ball-shaped (spherical Gaussian) clusters. In this paper, we extend the K-means clustering to accommodate extended clusters in subspaces, such as line-shaped clusters, plane-shaped clusters, and ball-shaped clusters. The algorithm retains much of the K-means clustering flavors: easy to implement and fast to converge. A model selection procedure is incorporated to determine the cluster shape. As a result, our algorithm can recognize a wide range of subspace clusters studied in various literatures, and also the global ball-shaped clusters (living in all dimensions). We carry extensive experiments on both synthetic and real-world datasets, and the results demonstrate the effectiveness of our algorithm.

1 Introduction

Data clustering is useful to discover patterns in a large dataset by grouping similar data points into clusters. Traditionally a cluster is viewed as a spherical (ball-shaped) data distribution and therefore is often modeled with mixture of Gaussians. For high dimensional data, sometimes an extended cluster may live in a subspace with much smaller dimension, i.e., it deviates away from a spherical cluster very significantly (Figures 1-5 in Section 5 illustrate various subspace clusters). This type of subspace clusters is difficult to discover using traditional clustering algorithms (e.g., K-means).

Here, we begin with an observation on the key difficulty of subspace clustering. We believe the main difficulty with subspace clustering is the exact definition of clusters. If a cluster lives in a subspace, but is not extended significantly, this type of clusters can be handled by traditional algorithms such as K-means. However, if a cluster has an extended size in limited dimensions, (e.g., an extended plane in 5-dimensional space), it becomes a subspace cluster. Thus the difficulty is how to model an extended cluster.

In this paper, we propose a new clustering algorithm to handle clusters with extended shapes. Our approach is a distance-minimization based approach, much like the standard K-means clustering algorithm. This approach can be cast in the framework of mixture clustering with Expectation and Maximization steps.

In the following, we discuss the related work in Section 2. In section 3 we propose several distance minimization based models to describe extended cluster objects and derive the optimal solutions to these cluster models (Sections 3.2-3.5). We discuss the model selection issue in section 3.6. Extensive illustrative examples are given in Section 4.

W. Buntine et al. (Eds.): ECML PKDD 2009, Part II, LNAI 5782, pp. 506–521, 2009.

One strong feature of our K-subspace algorithm is that it also accommodates ball-shaped clusters which live in all dimensions, in addition to the subspace clusters. In other words, our K-subspace algorithm is a comprehensive clustering algorithm, in contrast to many previous subspace clustering algorithms which are specifically tailored to find subspace clusters.

Clusters come in all shapes and sizes. Consider an elongated ball cluster which is in-between a line-shaped cluster and a ball-shaped cluster. Our algorithm tries to model it in both ways and automatically determine the best way to model it. An algorithm specifically designed for seeking subspace clusters would have to incorporate certain threshold to define whether it meets the criteria for subspace. This specific subspace clustering search algorithm goes in contradiction to the unsupervised nature of clustering.

Our K-subspace algorithm, on the other hand, retains much of the usual unsupervised learning. It is an extension of K-means algorithm for incorporating the capability to handle subspace clusters. For these reasons, we run our algorithm on an extensive list of datasets and compare with many existing clustering algorithms. Results are presented in Section 5. The new algorithm outperforms many existing algorithms, due to its capability to model a larger number of different cluster shapes. Conclusions are given in Section 6.

2 Related Work

Current study of subspace clustering mainly follows two general approaches [20].: (1) bottom-up search methods such as Clique [3], enclus [6], mafia [14], cltree [18], and DOC [21]; and (2) top-down search methods including COSA [13], Proclus [1], ORCLUS [2], Findit [25], and δ-clusters [27]. The bottom-up approaches use the monotonicity of density to prune subspaces by selecting those subspaces with densities above a given threshold, and they often cover clusters in various shapes and sizes. However, the clustering results are much dependent on the density threshold parameter, which can be very difficult to set properly. The top-down approaches integrate clustering and subspace selection by multiple iterations of selection and feature evaluation. The required iterations of clustering are very expensive and often cause slow performance. In addition to the poor efficiency, the top-down approaches are bad at finding hyper-spherical clusters due to the nature of using cluster centers to represent similar instances. Other recent works include GPCA [24], Oriented K-windows [23], Weighted K-means [30], adaptive subspace clustering [17] etc.

Adaptive dimension reduction, which attempts to adaptively find the subspace where clusters are most well-separated or well-defined, has also been proposed. Recent work in [11] combines linear discriminant analysis (LDA) and K-means clustering in a coherent way where K-means is used to generate cluster labels and LDA is used to select subspaces. Thus the adaptive subspace selection is integrated with the clustering process. Spectral clustering (e.g., Normalized Cut algorithm) and nonnegative matrix factorization (NMF) are also able to discover arbitrarily shaped clusters. It has been shown that spectral clustering can be

viewed as a weighted version of Kernel K-means, and NMF is proved to be equivalent to spectral clustering [10].

In our work, we extend K-means algorithm to K-subspace algorithm, which handles not only clusters living in all dimensions but also subspace clusters, while retaining the simple and fast implementation of K-means clustering.

3 K-Subspace Clustering Model

3.1 The Distance Minimization Clustering Model

The distance minimization clustering model can be represented as:

$$\min_{H,\theta} \sum_{i=1}^{n} \left[\min_{1 \le k \le K} Dist(\mathbf{x}_i, C_k) \right] \tag{1}$$

where H is the cluster indicator matrix, θ is a parameter in the optimization, $\mathbf{x}_i$ is a data point, C_k is the $k-th$ cluster, and

$$Dist(\mathbf{x}_i, C_k) = -\log Prob(\mathbf{x}_i; C_k).$$

Note $Prob(\mathbf{x}_i; C_k) \le 1$ and a common choice of the individual probability is the full Gaussian distribution using Mahalanobis distance.

This minimization can be equivalently written as

$$\min_{\{C_k\}} J = \sum_{k=1}^{K} \sum_{i \in C_k} Dist(\mathbf{x}_i, C_k) \tag{2}$$

Clearly for K-means clustering,

$$Dist(\mathbf{x}_i, C_k) = ||\mathbf{x}_i - \mathbf{c}_k||^2 \tag{3}$$

The K-subspace clustering model utilizes the distance minimization model. In this paper we consider only linear subspaces: 1D line-shaped (rod-like) subspace, 2D plane-shaped (slab-like) subspace, etc. The model also accommodates the ball-shaped clusters. Our main contribution here is to show that these subspace cluster models can be rigorously defined and efficiently computed. To our knowledge, this approach is novel and this type of analysis and results have not been previously investigated and obtained.

3.2 Line-Shaped Clusters

How to represent a line-shaped cluster C_k? A line is defined by a point in space and a unit direction. Let the point be $\mathbf{c}_k$ and the unit direction be $\mathbf{a}_k$. A data point $\mathbf{x}$ can be decomposed as a parallel component

$$\mathbf{x}^{\parallel} = \mathbf{a}_k[\mathbf{a}_k^T(\mathbf{x} - \mathbf{c}_k)]$$

and a perpendicular component

$$\mathbf{x}^{\perp} = (\mathbf{x} - \mathbf{c}_k) - \mathbf{x}^{\parallel}.$$

We define the distance between the data point $\mathbf{x}$ and cluster C_k as the perpendicular distance between the point and the line:

$$Dist(\mathbf{x}, C_k) = ||\mathbf{x}^{\perp}||^2 = ||\mathbf{x} - \mathbf{c}_k - \alpha \mathbf{a}_k||^2 \tag{4}$$

where

$$\alpha = (\mathbf{x} - \mathbf{c}_k)^T \mathbf{a}_k. \tag{5}$$

Computing model parameters. Given a set of data points in the cluster, and assuming they are described by a line, how do we compute the model parameters $(\mathbf{c}_k, \mathbf{a}_k)$?

We minimize the total distance (dispersion) of the cluster:

$$\min_{\mathbf{c}_k, \mathbf{a}_k} \sum_{i \in C_k} Dist(\mathbf{x}_i, C_k) = \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{c}_k - \alpha \mathbf{a}_k||^2 \tag{6}$$

It turns out that the model parameters can be computed in a surprisingly simple way:

Theorem 1. Let

$$X^{(k)} = [\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, ..., \mathbf{x}_{n_k}^{(k)}] \tag{7}$$

be the data points in C_k. Compute the principal component analysis (PCA) of them, and let $\mathbf{u}_1$ be the first principal direction. Let the centroid of C_k be

$$\bar{\mathbf{x}}^{(k)} = \frac{1}{|C_k|} \sum_{i \in C_k} \mathbf{x}_i. \tag{8}$$

Then the global optimal solution to the minimization problem of Eq.(6) is given by

$$\mathbf{c} = \bar{\mathbf{x}}^{(k)}, \ \mathbf{a} = \mathbf{u}_1. \tag{9}$$

Proof. We wish to minimize Eq.(6) which is

$$\min_{\mathbf{c}_k, \mathbf{a}_k} J_1 = \sum_{i \in C_k} ||(I - \mathbf{a}_k \mathbf{a}_k^T)(\mathbf{x}_i - \mathbf{c}_k)||^2 \tag{10}$$

by substituting Eq.(5) into Eq.(6).

Using $||A||^2 = \mathrm{Tr}(A^T A)$, we have

$$J_1 = \mathrm{Tr} \sum_{i \in C_k} [(\mathbf{x}_i - \mathbf{c}_k)^T (I - \mathbf{a}_k \mathbf{a}_k^T)^T (I - \mathbf{a}_k \mathbf{a}_k^T)(\mathbf{x}_i - \mathbf{c}_k)] \tag{11}$$

Because $(I - \mathbf{a}_k \mathbf{a}_k^T)^T (I - \mathbf{a}_k \mathbf{a}_k^T) \equiv M$ is a semi-positive definite matrix, J_1 is a convex function of $\mathbf{c}_k$. Its global minima is given by the zero- gradient condition:

$$0 = \frac{\partial J_1}{\partial \mathbf{c}_k} \sum_i 2(M\mathbf{c}_k - M\mathbf{x}_i)$$

Thus $M\mathbf{c}_k = M\bar{\mathbf{x}}^{(k)}$. Clearly,$\mathbf{c}_k = \bar{\mathbf{x}}^{(k)}$ is the solution. In general, M is non-deficient and$\mathbf{c}_k = \bar{\mathbf{x}}^{(k)}$ is the unique solution. When M is deficient, $\mathbf{c}_k = \bar{\mathbf{x}}^{(k)}$ is

still a correct solution, but it is not the unique solution. Because $\mathbf{a}_k$ is a unit vector,$(I - \mathbf{a}_k\mathbf{a}_k^T)^T(I - \mathbf{a}_k\mathbf{a}_k^T) = (I - \mathbf{a}_k\mathbf{a}_k^T)$. Thus

$$J_1 = \mathrm{Tr}(I - \mathbf{a}_k\mathbf{a}_k^T)\Sigma,$$

where

$$\Sigma = [\sum_{i \in C_k} (\mathbf{x}_i - \mathbf{c}_k)(\mathbf{x}_i - \mathbf{c}_k)^T]$$

is the sample covariance matrix. Now minimization for $\mathbf{a}_k$ becomes

$$\min_{\mathbf{a}_k} J_1 = \mathrm{Tr}\ (I - \mathbf{a}_k\mathbf{a}_k^T)\Sigma = const - \mathbf{a}_k^T\Sigma\mathbf{a}_k \tag{12}$$

This is equivalent to $\max_{\mathbf{a}_k} \mathbf{a}_k^T\Sigma\mathbf{a}_k$, whose solution is the first principle direction, because Σ is the covariance matrix. □

3.3 Plane-Shaped Clusters

A 2D plane is defined by a point in space and two unit directions. Let $\mathbf{c}_k$ be the point, and $\mathbf{a}_k, \mathbf{b}_k$ be the two unit directions. A data point $\mathbf{x}$ can be decomposed as a parallel component (lying inside the plane)

$$\mathbf{x}^{\|} = \mathbf{a}_k[\mathbf{a}_k^T(\mathbf{x} - \mathbf{c}_k)] + \mathbf{b}_k[\mathbf{b}_k^T(\mathbf{x} - \mathbf{c}_k)]$$

assuming the two unit directions are orthogonal: $\mathbf{a}_k^T\mathbf{b}_k = 0$. And the perpendicular component

$$\mathbf{x}^{\perp} = (\mathbf{x} - \mathbf{c}_k) - \mathbf{x}^{\|}.$$

We define the distance between the data point $\mathbf{x}$ and cluster C_k as the perpendicular distance between the point and the plane:

$$Dist(\mathbf{x}, C_k) = ||\mathbf{x}^{\perp}||^2 = ||\mathbf{x} - \mathbf{c}_k - \alpha\mathbf{a}_k - \beta\mathbf{b}_k||^2 \tag{13}$$

where

$$\alpha = (\mathbf{x} - \mathbf{c}_k)^T\mathbf{a}_k, \quad \beta = (\mathbf{x} - \mathbf{c}_k)^T\mathbf{b}_k. \tag{14}$$

Clearly, we can extend this to 3D and higher dimensional subspace clusters.

Computing model parameters. Given a set of data points in the cluster. Assuming they are described by a plane, how do we compute the model parameters $(\mathbf{c}_k, \mathbf{a}_k, \mathbf{b}_k)$?

We minimize the dispersion (total distance) of the cluster:

$$\min_{\mathbf{c}_k, \mathbf{a}_k, \mathbf{b}_k} \sum_{i \in C_k} Dist(\mathbf{x}_i, C_k) = \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{c}_k - \alpha\mathbf{a}_k - \beta\mathbf{b}_k||^2 \tag{15}$$

As one might have expected, the solution is given by PCA.

Theorem 2. Compute the (PCA) of $X^{(k)}$ of Eq.(7) to obtain the first two principal directions $(\mathbf{u}_1, \mathbf{u}_1)$. Compute the centroid as in Eq.(8). The optimal solution to Eq.(15) is given by

$$\mathbf{c} = \bar{\mathbf{x}}^{(k)},\ \mathbf{a} = \mathbf{u}_1,\ \mathbf{b} = \mathbf{u}_2. \tag{16}$$

Proof. We outline the proof here by pointing out the main difference from the proof for Theorem 1.

We wish to minimize Eq.(15) which is

$$\min_{\mathbf{c}_k,\mathbf{a}_k,\mathbf{b}_k} J_2 = \sum_{i \in C_k} ||(I - \mathbf{a}_k\mathbf{a}_k^T - \mathbf{b}_k\mathbf{b}_k^T)(\mathbf{x}_i - \mathbf{c}_k)||^2 \tag{17}$$

by substituting Eq.(5) into Eq.(17). J_2 can be written as

$$J_2 = \sum_{i \in C_k} \text{Tr } [(\mathbf{x}_i - \mathbf{c}_k)^T B^T B(\mathbf{x}_i - \mathbf{c}_k)] \tag{18}$$

where

$$B = (I - \mathbf{a}_k\mathbf{a}_k^T - \mathbf{b}_k\mathbf{b}_k^T)$$

Because $B^T B$ is a semi-positive definite matrix, J_2 is a convex function of $\mathbf{c}_k$. Its global minima is given by$\mathbf{c}_k = \bar{\mathbf{x}}^{(k)}$.

Now focusing on $(\mathbf{a}_k, \mathbf{b}_k)$. Since we restrict $\mathbf{a}_k^T\mathbf{b}_k = 0$,and $B^T B = B = (I - \mathbf{a}_k\mathbf{a}_k^T - \mathbf{b}_k\mathbf{b}_k^T)$, the minimization over $(\mathbf{a}_k, \mathbf{b}_k)$ becomes

$$\min_{\mathbf{a}_k,\mathbf{b}_k} J_2 = \text{Tr } (I - \mathbf{a}_k\mathbf{a}_k^T - \mathbf{a}_k\mathbf{b}_k^T)\Sigma = const - \mathbf{a}_k^T\Sigma\mathbf{a}_k - \mathbf{b}_k^T\Sigma\mathbf{b}_k \tag{19}$$

The solution for $\mathbf{a}_k$ is given by the first principal direction $\mathbf{u}_1$. Since $\mathbf{a}_k^T\mathbf{b}_k = 0$, the solution for $\mathbf{b}_k$ is given by the second principal direction $\mathbf{u}_2$. □

3.4 3D and Higher Dimensional Planes

Clearly, we can model a subspace cluster using 3D and higher dimensional planes. The same definitions using the perpendicular distances can be adopted and the same computational algorithms using PCA can be easily generalized from 1D and 2D cases.

We can use this in the reverse direction. Suppose we are given a set of data points, and the question is which dimensions of the subspace cluster we should choose?

According to Theorem 1 and Theorem 2, we compute PCA of the dispersion and obtain eigenvalues and eigenvectors. If there is only one significant eigenvalue, this implies that the cluster is a 1D (line-shaped) subspace cluster, then we set $\mathbf{a}_k = \mathbf{u}_1$. If there are only two significant eigenvalues, it implies that the cluster is a 2D (plane-shaped) subspace cluster, then we set $\mathbf{a}_k = \mathbf{u}_1$ and $b_k = \mathbf{u}_2$.

In general, if there are k significant eigenvalues, then the cluster is likely a k-dimensional (kD plane) subspace cluster. For this cluster, we need k vectors (kp parameters, where $p =$ dimension of data space) to describe it.

However, to keep the simplicity of cluster modeling to prevent overfitting with too many parameters, we restrict the cluster dimensions to 2. For higher dimensions, we use a spherical cluster to describe the cluster, instead of using the high-dimensional planes which have much more parameters.

3.5 Spherical Clusters

Although Gaussian mixtures using the EM algorithm can describe more complicated clusters, in practice very often the much simpler K-means clustering algorithm provides results as good as the mixture models.

This implies two fundamental facts in statistics: (S1) complex functions do not necessarily perform better (mainly due to overfitting) and (S2) most clusters are reasonably modeled by the spherical cluster model. In fact, clusters of any shape can be modeled as spherical clusters if they are reasonably separated.

From (S1), we should not use too many parameters in describing a cluster. From (S2), we should make use of spherical clusters as much as possible. These two facts motivate us to consider spherical clusters more carefully, rather than using 3D or higher dimensional plane-shaped clusters.

A spherical (ball-shaped) data cluster is a global cluster in the sense that it exists in all dimensions. We adopt it as our K-subspace clustering because not all clusters in a dataset are subspace clusters.

Then a natural question is: what is the distance between a data point and a spherical cluster? We could use the distance to the cluster centroid

$$Dist(\mathbf{x}_i, C_k) = ||\mathbf{x}_i - \mathbf{c}_k||^2 \tag{20}$$

as in K-means clustering. If all clusters in the dataset are spherical clusters, the problem is reduced to the usual K-means clustering.

However, we wish to measure the distance between a data point to the *cluster*, not to the cluster *center*. In the case of the line-shaped cluster, the point-cluster distance is the perpendicular distance, i.e., data point $\mathbf{x}$ to the closest data points on the line. In the case of the plane-shaped cluster, the point-cluster distance is the perpendicular distance, i.e., data point $\mathbf{x}$ to the closest data points on the plane.

For these reasons, we define the point-cluster distance for spherical clusters to be the distance between the data point and the closest data point on the sphere. For this we introduce

$$Dist(\mathbf{x}_i, C_k) = \max\left(0, ||\mathbf{x}_i - \mathbf{c}_k||^2 - \eta\sigma^2\right) \tag{21}$$

where $\eta \geq 0$ is an input parameter. This distance assumes that the data points inside the cluster are located in the spherical surface of radius $\sqrt{\eta}\sigma_k$. Thus $Dist(\mathbf{x}_i, C_k)$ is the distance of a outside data point to this spherical surface. Setting $\eta = 1$, assuming Gaussian distribution, there will be about 69% data points inside this surface. Setting $\eta = 0$, the distance metric is reduced to the distance to the cluster center. In practice, we set $\eta \simeq 0.2 \sim 0.5$.

Computing model parameters. To compute the model parameter $\mathbf{c}_k$, we optimize the dispersion of C_k

$$\min_{\mathbf{c}_k} \sum_{i \in C_k} \max\left(0, ||\mathbf{x}_i - \mathbf{c}_k||^2 - \eta\sigma_k^2\right) \tag{22}$$

which, by substituting the variance, is

$$\min_{\mathbf{c}_k} \sum_{i \in C_k} \max \left(0, ||\mathbf{x}_i - \mathbf{c}_k||^2 - \eta \sum_{j \in C_k} ||\mathbf{x}_j - \mathbf{c}_k||^2 \right) \tag{23}$$

The $\max(\cdot)$ is not differentiable and thus difficult to handle analytically. We use the following continuous function to handle it:

$$\max(u, v) = \frac{1}{\tau} \lim_{\tau \to \infty} \log(e^{\tau u} + e^{\tau v}) \tag{24}$$

Thus we minimize

$$J_3(\mathbf{c}_k) = \sum_{i \in C_k} \frac{1}{\tau} \lim_{\tau \to \infty} \log f(\tau, \mathbf{x}_i, \mathbf{c}_k) \tag{25}$$

where

$$f(\tau, \mathbf{x}_i, \mathbf{c}_k) = 1 + \exp \left(\tau ||\mathbf{x}_i - \mathbf{c}_k||^2 - \tau \frac{\eta}{n} \sum_{j \in C_k} ||\mathbf{x}_j - \mathbf{c}_k||^2 \right) \tag{26}$$

Taking the derivative w.r.t. $\mathbf{c}_k$, we have

$$\frac{\partial J_3(\mathbf{c}_k)}{\partial \mathbf{c}_k} = \sum_{i \in C_k} \lim_{\tau \to \infty} \frac{2e^f}{1 + e^f} \left(\mathbf{c}_k - \mathbf{x}_i - \frac{\eta}{n} \sum_{j \in C_k} (\mathbf{c}_k - \mathbf{x}_j) \right) \tag{27}$$

Note

$$\lim_{\tau \to \infty} \frac{2e^{f(\cdot)}}{1 + e^{f(\cdot)}} = 2 \quad \text{if} \quad ||\mathbf{x}_i - \mathbf{c}_k||^2 > \frac{\eta}{n} \sum_{j \in C_k} ||\mathbf{x}_j - \mathbf{c}_k||^2 \tag{28}$$

$$\lim_{\tau \to \infty} \frac{2e^{f(\cdot)}}{1 + e^{f(\cdot)}} = 0 \quad \text{if} \quad ||\mathbf{x}_i - \mathbf{c}_k||^2 < \frac{\eta}{n} \sum_{j \in C_k} ||\mathbf{x}_j - \mathbf{c}_k||^2 \tag{29}$$

Thus we obtain

$$\frac{\partial J_3(\mathbf{c}_k)}{\partial \mathbf{c}_k} = \sum_{\mathbf{x}_i^>} \left(\mathbf{c}_k - \mathbf{x}_i - \frac{\eta}{n} \sum_{j \in C_k} (\mathbf{c}_k - \mathbf{x}_j) \right) \tag{30}$$

where $\mathbf{x}_i^>$ denotes the points outside the sphere, i.e., they satisfy Eq.(28). Let the center for averaging outside points be

$$\bar{\mathbf{x}}_>^k = \left(\sum_{\mathbf{x}_i^>} \mathbf{x}_i \right) \Big/ \left(\sum_{\mathbf{x}_i^>} 1 \right). \tag{31}$$

The final solution to the optimization problem is

$$\mathbf{c}_k = \frac{\bar{\mathbf{x}}_>^k - \eta\, \bar{\mathbf{x}}^k}{1 - \eta} \tag{32}$$

where $\bar{\mathbf{x}}^k$ is the centroid of C_k. Clearly, as $\eta \to 0$, $\bar{\mathbf{x}}_>^k \to \bar{\mathbf{x}}^k$. Thus this result is consistent.

3.6 Model Selection and Computation

In our distance-minimization based clustering model, because we allow each cluster to be modeled by different models, there is a model selection issue. Given the data points in C_k as in Eq.(7), which model describes the data best?

We define the model dispersion as

$$Dispersion(ball) = \sum_{i \in C_k} \max(0, ||\mathbf{x}_i - \mathbf{c}_k||^2 - \eta\sigma_k^2),$$

$$Dispersion(line) = \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{c}_k - \alpha_i \mathbf{a}_k||^2,$$

$$Dispersion(plane) = \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{c}_k - \alpha_i \mathbf{a}_k - \beta_i b_k||^2,$$

$$\alpha_i = (\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{a}_k, \beta_i = (\mathbf{x}_i - \mathbf{c}_k)^T b_k;$$

We choose the model with the smallest dispersion.

Then we minimize the total distance via the fundamental EM algorithmic approach. The two steps are the Cluster Assignment step (E-step) and Model Estimation step (M-step). The *Cluster Assignment* step is estimating the posterior probability for all data points belonging to different clusters. In our distance minimization model, for every $\mathbf{x}_i$, we assign $\mathbf{x}_i$ to the closest cluster C_k:

$$k = \arg\min_{1 \le k \le K} Dist(\mathbf{x}_i, C_k) \tag{33}$$

With new assignments of $\{\mathbf{x}_i\}$ to each cluster, in the model estimation step we re-estimate the model parameters using the computational procedure described in §2.1, §2.2 and §2.4.

4 Illustrative Examples

To demonstrate the effectiveness of the K-subspace clustering in extended clusters, we give the following examples. In the following figures, points with different colors belong to different clusters.

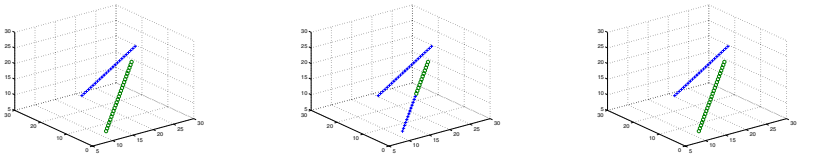
Two lines in 3-D space: 50 points along two straight lines are generated randomly in 3-D space as shown in Figure 1(a).

A line and a 2-D plane in 3-D space: 50 points along a straight line and 300 points in a 2-D plane are generated randomly in the 3-D space as shown in Figure 2(a).

Two 2-D planes in 3-D space: 300 points in each 2-D plane are generated randomly in the 3-D space as shown in Figure 3(a).

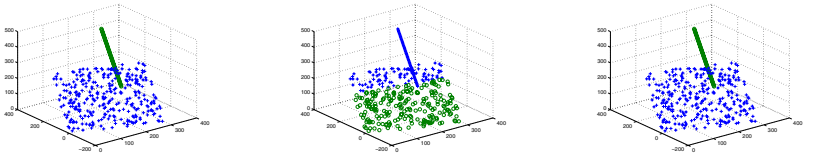
A Line, a 2-D plane and a sphere in 3-D space: 50 points along a straight line, 300 points in a 2-D plane, and 400 points in the surface of a sphere are generated randomly in the 3-D space as shown in Figure 4a.

Four clusters in 3-D space: Points with four clusters, each of which exists in just two dimensions with the third dimension being noise [20]. Points from two clusters can be very close together, which confuses many traditional clustering



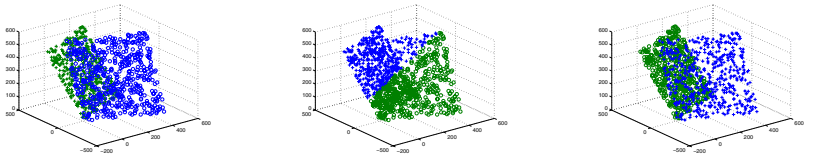
(a) Original Data (b) K-means Results (c) K-subspace Results

Fig. 1. Example (I): Two Lines in 3-D Space



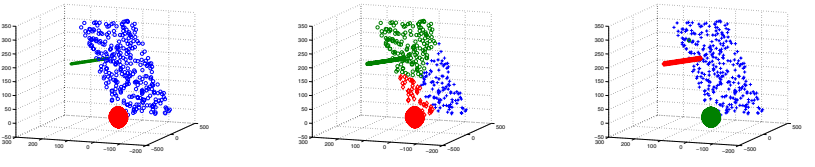
(a) Original Data (b) K-means Results (c) K-subspace Results

Fig. 2. Example (II): Line-Plane in 3-D Space



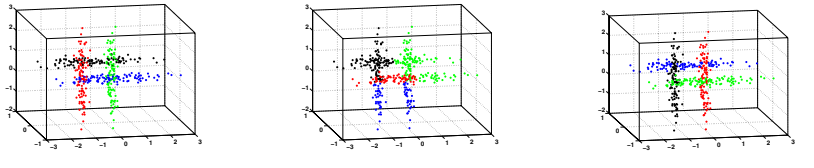
(a) Original Data (b) K-means Results (c) K-subspace Results

Fig. 3. Example (III): Plane-Plane in 3-D Space



(a) Original Data (b) K-means Results (c) K-subspace Results

Fig. 4. Example (IV): Line-Plane-Ball in 3-D Space



(a) Original Data (b) K-means Results (c) K-subspace Results

Fig. 5. Example (V): Four clusters in 3-D Space

algorithms. The first two clusters exist in dimensions x and y. The data forms a normal distribution with means 0.6 and -0.6 in dimension x and 0.5 in dimension y, and standard deviations of 0.1. In dimension z, these clusters have $\mu = 0$ and $\sigma = 1$. The second two clusters are in dimensions y and z and are generated in the same manner. The data is illustrated in Figure 5a.

As you can observe from these figures, K-means algorithm has difficulties in dealing with subspace clusters. However, our K-subspace algorithm is a comprehensive clustering algorithm, which can recognize a wide range of subspace clusters and is also able to discover the global sphere clusters.

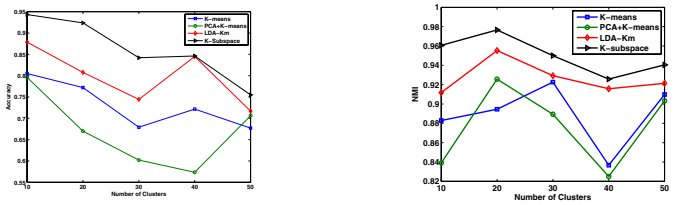
5 Experimental Results

In this section, we conduct comprehensive experiments to evaluate the effectiveness of the K-subspace clustering method. We use both synthetic datasets and real-world datasets. The purpose of using synthetic data is that the detailed cluster structures are known, hence we can evaluate K-subspace with different factors such as cluster structures and sizes systematically. And we believe that those cluster structures do exist in real world applications. We use accuracy [12] and normalized mutual information (NMI) [22] as performance measures.

5.1 Experiments on Synthetic Datasets

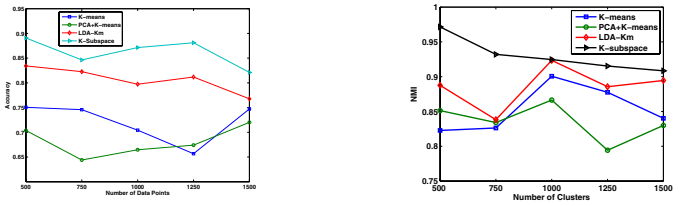
Dataset Generation. Synthetic datasets are generated by using the algorithm proposed by Milligan [15]. The clusters are non-overlapping and truncated multivariate normal mixtures. Subspace structures are embedded in the synthetic datasets. Basically the dimension with clustering structures (called cluster dimensions) is created based on a cluster length chosen from a uniform distribution. The mean and standard deviation of the cluster on this dimension can be derived from this cluster length. The dimension without clustering structures (called noise dimensions) is created from a uniform distribution in some generated range. Thus cluster dimensions and noise dimensions are independent from each other. The data points are assigned to the cluster if they are within 1.5 standard deviation of every cluster dimension of this cluster. The number of points that are assigned to each cluster are determined as following: From the range $[a, a+d]$ where $a, d > 0$, we choose k numbers $\{P_1, \cdots, P_k\}$ uniformly, and k is the number of clusters. Then, the number of points belonging to the cluster i is $n\frac{P_i}{\sum_j P_j}$, where n is the total number of data points.

Results on Synthetic Datasets. Three sets of experiments are conducted on synthetic datasets: (1) We fix the number of data points and dimensionality and investigate the clustering performance of k-subspace as a function of the number clusters. (2) We fix the number of clusters and dimensionality and investigate the clustering performance of K-subspace as a function of the number of points. (3) We fix the number of clusters and points and investigate the clustering performance of K-subspace as a function of the number of dimensionality.



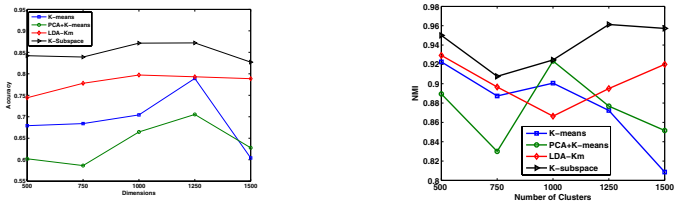
(a) Accuracy (b) NMI

Fig. 6. Clustering performance comparison on synthetic dataset (I). Remark: the number of data points is 1000; the dimensionality of the data is 500; and the number of clusters changes from 10 to 50.



(a) Accuracy (b) NMI

Fig. 7. Clustering performance comparison on synthetic dataset (II). Remark: the number of clusters is 30; the dimensionality of the data is 1000; and the number of data points changes from 500 to 1500.



(a) Accuracy (b) NMI

Fig. 8. Clustering performance comparison on synthetic dataset (III). Remark: the number of clusters is 30; the number of data points is 1000; and the dimensionality of the data changes from 500 to 1500.

We compare the K-subspace clustering algorithm with three other algorithms: (1) Standard K-means algorithm; (2) LDA-Km algorithm [11]: an adaptive subspace clustering algorithm by integrating linear discriminant analysis (LDA) and K-means clustering into a coherent process. (3) PCA+K-means clustering algorithm: PCA is firstly applied to reduce the data dimension followed by K-means clustering. The results are shown in Figure 6, Figure 7, and Figure 8, respectively. We observe that: (1) LDA-Km outperforms K-means and PCA+K-means since it integrates adaptive subspace selection with clustering process. (2) K-subspace outperforms LDA-Km and achieves the best results among all the methods. This

is due to the fact that K-subspace clustering can recognize a wide range of subspace clusters and also global spherical clusters (living in all dimensions). (3) The clustering performance of K-subspace clustering decreases gradually as the number of clusters increases (as in Figure 6), and its performance does not vary much as the number of points and dimensions increase (shown in Figure 7 and Figure 8).

5.2 Experiments on Real-World Datasets

Dataset Description. We use a wide range of real-world datasets in our experiments as summarized in Table 1. The number of classes ranges from 4 to 20, the number of samples ranges from 47 to 7494, and the number of dimensions ranges from 9 to 1000. These datasets represent applications from different domains such as information retrieval, gene expression data and pattern recognition. The summary of the datasets is as follows: (1) Five datasets including Digits, Glass, Protein, Soybean, and Zoo are from UCI data repository [5]. (2) Five datasets including CSTR, Log, Reuters, WebACE, and WebKB are standard text datasets that have been frequently used in document clustering. The documents are represented as the term vectors using vector space model. These document datasets are pre-processed (removing the stop words and unnecessary tags and headers) using rainbow package [19]. The dataset descriptions can be found in [12].

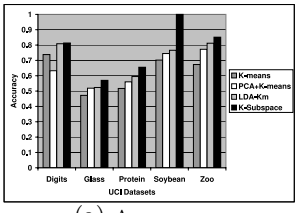
Table 1. Dataset Descriptions

Datasets	# Samples	# Dimensions	# Class
CSTR	475	1000	4
Digits	7494	16	10
Glass	214	9	7
Protein	116	20	6
Log	1367	200	8
Reuters	2900	1000	10
Soybean	47	35	4
WebACE	2340	1000	20
WebKB	4199	1000	4
Zoo	101	18	7

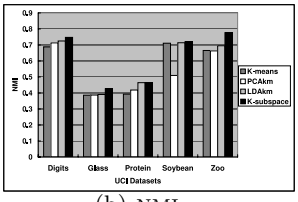
Results on Real Datasets. On the five datasets from UCI data repository, we compare the K-subspace clustering algorithm with standard K-means algorithm, LDA-Km algorithm, PCA+K-means clustering algorithm, and Spectral Clustering with Normalized Cuts (Ncut) [28]. We use normalized cut since it has been shown that weighted Kernel K-means is equivalent to the normalized cut [8]. The implementation of Ncut is based on [28] and the variance of the Gaussian similarity is determined using local scaling as in [29]. The results are shown in Figure 9(a) and Figure 9(b). On the text datasets, besides the above three alternative clustering methods, we also compare K-subspace algorithm with the following algorithms: (i) Non-negative Matrix Factorization (NMF) method [16]; (ii) Tri-Factorization Method (TNMF) [12]; (iii) Euclidean co-clustering (ECC) and minimum squared residue co-clustering (MSRC) algorithms [7]. Note that NMF

Table 2. Clustering accuracy comparison on text datasets

	K-means	PCA+K-means	LDA-Km	K-Subspace	ECC	MSRC	NMF	TNMF	Ncut
WebACE	0.4081	0.4432	0.4774	**0.5158**	0.4081	0.5021	0.4803	0.4996	0.4513
Log	0.6979	0.6562	0.7198	**0.7696**	0.7228	0.5655	0.7608	0.7527	0.7574
Reuters	0.436	0.3925	0.5142	**0.6426**	0.4968	0.4516	0.4047	0.4682	0.4890
CSTR	0.5210	0.5630	0.5630	**0.6555**	0.5945	0.6513	0.5945	0.6008	0.5435
WebKB	0.5951	0.6354	0.6468	**0.8583**	0.5210	0.5165	0.4568	0.6094	0.6040



(a) Accuracy



(b) NMI

Fig. 9. Clustering comparison on UCI data

has been shown to be effective in document clustering [26] and Tri-Factorization (TNMF) is an extension of NMF [12]. Both ECC and MSRC are document co-clustering algorithms that are able to find blocks in a rectangle document-term matrix. Co-clustering algorithms generally perform implicit dimension reduction during clustering process [9]. The comparison results on text datasets are shown in Table 2.

From the comparisons, we observe that: (1) On most datasets, PCA+K-means outperforms K-means due to the pre-processing by PCA. (2) Co-clustering algorithms (ECC and MSRC) generally outperform K-means since they are performing implicit dimension reduction during the clustering process. (3) NMF outperforms K-means on most datasets since NMF can model widely varying data distributions due to the flexibility of matrix factorization as compared to the rigid spherical clusters that the K-means clustering objective function attempts to capture [10]. When the data distribution is far from a spherical clustering, NMF has advantages over K-means. (4) TNMF provides a good framework for simultaneously clustering the rows and columns of the input documents. Hence TNMF generally outperforms NMF on text datasets. (5) The results of spectral clustering (Ncut) is better than K-means. Note that spectral clustering can be viewed as a weighted version of Kernel K-means and hence it is able to discover arbitrarily shaped clusters. The experimental results of Ncut is similar to those of NMF and TNMF. Note that it has also been shown that NMF is equivalent to spectral clustering [10]. (6) K-subspace clustering outperforms the other methods on all datasets. The improvements on Soybean dataset, Reuters dataset, and WebKB dataset are evident. The comparisons demonstrate the effectiveness of K-subspace algorithm in modeling different subspaces in real-life datasets.

Besides the above experimental comparisons, we also test the sensitivity of the parameter η introduced in Eq.(21). Figure 10 shows the clustering performance

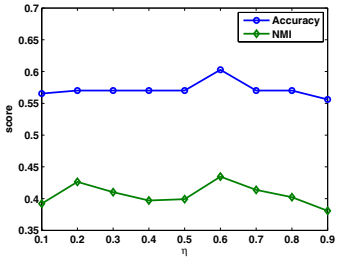


Fig. 10. Effects of η on glass dataset

as a function of η on glass dataset. Unlike many existing subspace clustering methods as discussed in Section 2, whose performance largely depends on the parameter tuning, and proper parameters setting in which sometimes are particularly difficult, we observe that the clustering performance of our K-subspace algorithm is not very sensitive to the choice of the parameters.

6 Conclusion

We extend the K-means clustering algorithm to accommodate subspace clusters in addition to the usual ball-shaped clusters. The optimal solutions to subspace models can be efficiently computed using PCA. We demonstrate that the model can capture most (if not all) of subspace clusters investigated so far. Running on synthetic datasets and 10 real life datasets, the K-subspace algorithm outperforms existing methods due to its increased capability of modeling clusters of different shapes.

Acknowledgement. The work of D. Wang and T. Li is partially supported by NSF grants IIS-0546280, CCF-0830659, and DMS-0844513. The work of C. Ding is partially supported by NSF grants DMS-0844497 and CCF-0830780.

References

1. Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C., Park, J.S.: Fast algorithms for projected clustering. In: SIGMOD 1999 (1999)
2. Aggarwal, C.C., Yu, P.S.: Finding generalized projected clusters in high dimensional spaces. In: SIGMOD 2000 (2000)
3. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: SIGMOD 1998 (1998)
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
5. Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
6. Cheng, C.-H., Fu, A.W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: SIGKDD 1999 (1999)

7. Cho, H., Dhillon, I., Guan, Y., Sra, S.: Minimum sum squared residue co-clustering of gene expression data. In: SDM 2004 (2004)
8. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: SIGKDD 2004 (2004)
9. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretical co-clustering. In: SIGKDD 2003 (2003)
10. Ding, C., He, X., Simon, H.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: SDM 2005 (2005)
11. Ding, C., Li, T.: Adaptive dimension reduction using discriminant analysis and k-means c lustering. In: ICML 2007 (2007)
12. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: SIGKDD 2006 (2006)
13. Friedman, J.H., Meulman, J.J.: Clustering objects on subsets of attributes. Journal of the Royal Statistical Society: Series B (Statistical Methodology) (2004)
14. Goil, S., Nagesh, H., Choudhary, A.: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern Univ. (1999)
15. Milligan, G.W.: An algorithm for generating artificial test clusters. Psychometrika 50, 123–127 (1985)
16. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. In: NIPS 2001 (2001)
17. Li, T., Ma, S., Ogihara, M.: Document Clustering via Adaptive Subspace Clsutering. In: SIGIR 2004 (2004)
18. Liu, B., Xia, Y., Yu, P.S.: Clustering through decision tree construction. In: CIKM 2000 (2000)
19. McCallum, A.K.: Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (1996), `http://www.cs.cmu.edu/~mccallum/bow`
20. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: A review. In: SIGKDD Explorations 2004 (2004)
21. Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.M.: A monte carlo algorithm for fast projective clustering. In: SIGMOD 2002 (2002)
22. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research (2003)
23. Tasoulis, D.K., Zeimpekis, D., Gallopoulos, E., Vrahatis, M.N.: Oriented -windows: A pca driven clustering method. In: Advances in Web Intelligence and Data Mining (2006)
24. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (GPCA). In: CVPR 2003 (2003)
25. Woo, K.-G., Lee, J.-H., Kim, M.-H., Lee, Y.-J.: Findit: a fast and intelligent subspace clustering algorithm using dimension voting. Information and Software Technology (2004)
26. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: SIGIR 2003 (2003)
27. Yang, J., Wang, W., Wang, H., Yu, P.: d-clusters: Capturing subspace correlation in a large data set. In: ICDE 2002 (2002)
28. Yu, S.X., Shi, J.: Multiclass spectral clustering. In: ICCV 2003 (2003)
29. Zelnik-manor, L., Perona, P.: Self-tuning spectral clustering. In: NIPS 2005 (2005)
30. Zhang, Q., Liu, J., Wang, W.: Incremental subspace clustering over multiple data streams. In: ICDM 2007 (2007)