Combined Central and Subspace Clustering for Computer Vision Applications

Le Lu LELU@CS.JHU.EDU
Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

René Vidal RVIDAL@CIS.JHU.EDU
Center for Imaging Science, Biomedical Engineering Department, Johns Hopkins University, Baltimore, MD 21218, USA

Abstract

Central and subspace clustering methods are at the core of many segmentation problems in computer vision. However, both methods fail to give the correct segmentation in many practical scenarios, e.g., when data points are close to the intersection of two subspaces or when two cluster centers in different subspaces are spatially close. In this paper, we address these challenges by considering the problem of clustering a set of points lying in a union of subspaces and distributed around multiple cluster centers inside each subspace. We propose a generalization of Kmeans and Ksubspaces that clusters the data by minimizing a cost function that combines both central and subspace distances. Experiments on synthetic data compare our algorithm favorably against four other clustering methods. We also test our algorithm on computer vision problems such as face clustering with varying illumination and video shot segmentation of dynamic scenes.

1. Introduction

Many computer vision problems require the efficient and effective organization of huge-dimensional data for information retrieval purposes. Unsupervised learning, mostly clustering, provides a way to handle these challenges.

Central and subspace clustering are arguably the most studied clustering problems. In *central clustering*, data samples are assumed to be distributed around a collection of cluster centers, e.g., a mixture of Gaussians. This problem shows up in many vision tasks, e.g., image segmentation, and can be solved using techniques such as Kmeans (Duda et al., 2000) or Expectation Maximization (EM) (Dempster et al., 1977).

In *subspace clustering*, data samples are assumed to be distributed in a collection of subspaces. This problem shows up in various vision applications, such as motion segmentation (Vidal & Ma, 2004), face clustering with varying illumination (Ho et al., 2003), temporal video segmentation (Vidal et al., 2005), etc. Subspace clustering can also be used to obtain a piecewise linear approximation of a manifold (Weinberger & Saul, 2004), as we will show in our real data experiments. Existing subspace clustering methods include Ksubspaces (Ho et al., 2003) and Generalized Principal Component Analysis (GPCA) (Vidal et al., 2005). Such methods do not enforce a particular distribution of the data inside the subspaces. Methods such as Mixtures of Probabilistic PCA (MPPCA) (Tipping & Bishop, 1999) further assume that the distribution of the data inside each subspace is Gaussian and use EM to learn the parameters of the mixture model and the segmentation of the data.

Unfortunately, there are many cases in which neither central nor subspace clustering individually are appropriate. For example, subspace clustering fails when the data set contains points close to the intersection of two subspaces, as shown by the example in Figure 1. Similarly, central clustering fails when two clusters in different subspaces are spatially close, as shown by the example in Figure 2.

In this paper, we propose a new clustering approach that combines both central and subspace clustering. We obtain an initial solution by grouping the data into multiple subspaces using GPCA and grouping the data inside each subspace using Kmeans. This initial solution is then refined by minimizing an objective function composed of both central and subspace distances. This combined optimization leads to improved performance of our method over four different clustering approaches in terms of both clustering error and estimation accuracy. Real examples on illumination-invariant face clustering and video shot detection are also performed. Our experiments also show that combined central/subspace clustering can be effectively used to obtain a piecewise linear approximation of complex manifolds.

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

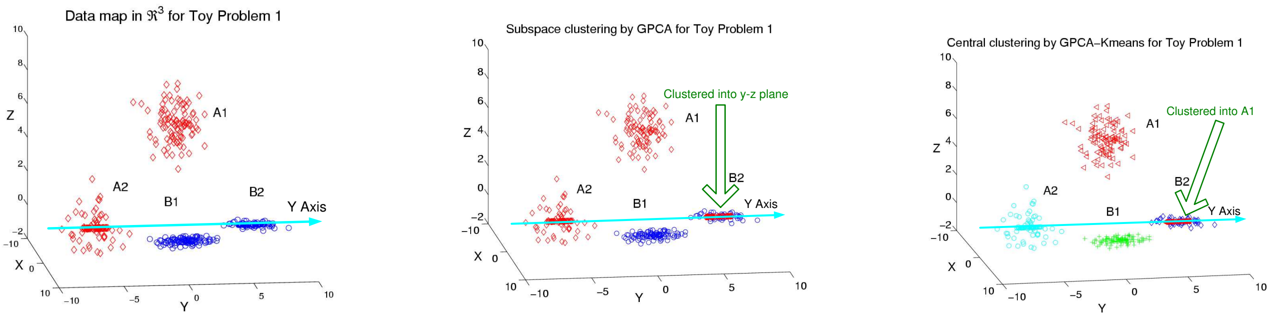


Figure 1. **Left:** A set of points in $\mathbb{R}^3$ drawn from 4 clusters labeled as A_1, A_2, B_1, B_2. Clusters B_1 and B_2 lie in the x-y plane and clusters A_1 and A_2 lie in the y-z plane. Note that some points in A_2 and B_2 are drawn from the intersection of the two planes (y-axis). **Center:** Subspace clustering by GPCA assigns all the points in the y-axis to the y-z plane, thus it misclassifies some points in B_2. **Right:** Subspace clustering using GPCA followed by central clustering inside each plane using Kmeans misclassifies some points in B_2.

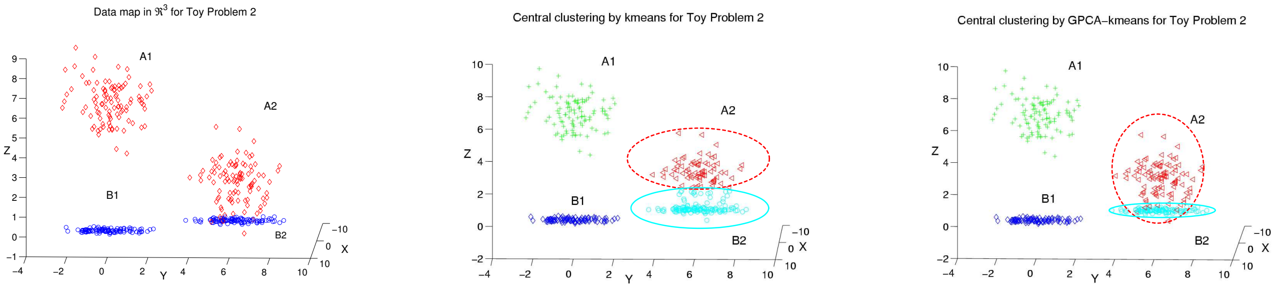


Figure 2. **Left:** A set of points in $\mathbb{R}^3$ distributed around 4 clusters labeled as A_1, A_2 B_1, B_2. Clusters B_1 and B_2 lie in the x-y plane and clusters A_1 and A_2 lie in the y-z plane. Note that cluster B_2 (in blue) is spatially close to cluster A_2 (in red). **Center:** Central clustering by Kmeans assigns some points in A_2 to B_2. **Right:** Subspace clustering using GPCA followed by central clustering inside each subspace using Kmeans gives the correct clustering into four groups.

2. Combined Central-Subspace Clustering

Let $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^P$ be a collection of P points lying approximately in n subspaces $S_j = \{\boldsymbol{x} : B_j^\top \boldsymbol{x} = 0\}$ of dimension d_j with normal bases $\{B_j \in \mathbb{R}^{(D-d_j)\times D}\}_{j=1}^n$. Assume that within each subspace S_j the data points are distributed around m_j cluster centers $\{\mu_{jk} \in \mathbb{R}^D\}_{j=1\dots n}^{k=1\dots m_j}$. In this paper, we consider the following problem:

Problem 1 (Combined central and subspace clustering) *Given $\{\boldsymbol{x}_i\}_{i=1}^P$, estimate $\{B_j\}_{j=1}^n$ and $\{\mu_{jk}\}_{j=1\dots n}^{k=1\dots m_j}$.*

When $n = 1$, Problem 1 reduces to the standard central clustering problem. A popular central clustering method is the Kmeans algorithm, which solves for the cluster centers μ_k and the membership of the ith point to the kth cluster center $w_{ik} \in \{0, 1\}$ by minimizing the within class variance

$$J_{KM} \doteq \sum_{i=1}^{P}\sum_{k=1}^{m_1} w_{ik}\|\boldsymbol{x}_i - \mu_k\|^2. \tag{1}$$

Given the cluster centers, the optimal solution for the memberships is to assign each point to the closest center. Given the memberships, the optimal solution for the cluster centers is given by the means of the points within each group. The Kmeans algorithm proceeds by alternating between these two steps until convergence to a local minimum.

When $m_j = 1$ and $n > 1$, Problem 1 reduces to the classical subspace clustering problem. This problem can be solved with an extension of Kmeans, called Ksubspaces, which solves for the subspace normal bases B_j and the membership of the ith point to the jth subspace $w_{ij} \in \{0, 1\}$ by minimizing the cost function

$$J_{KS} \doteq \sum_{i=1}^{P}\sum_{j=1}^{n} w_{ij}\|B_j^\top \boldsymbol{x}_i\|^2 \tag{2}$$

subject to the constraints $B_j^\top B_j = \mathcal{I}$, for $j = 1, \dots, n$, where $\mathcal{I}$ denotes the identity matrix. Given the normal bases, the optimal solution for the memberships is to assign each point to the closest subspace. Given the memberships, the optimal solution for the normal bases is obtained from the null space of the data matrix of each group using SVD. The Ksubspaces algorithm proceeds by alternating between these two steps until convergence to a local minimum.

In this section, we are interested in the more general problem of $n > 1$ subspaces and $m_j > 1$ centers per subspace. In principle, we could also solve this problem using Kmeans by interpreting Problem 1 as a central clustering problem with $\sum m_j$ cluster centers. However, Kmeans does not fully employ the data's structural information and can cause undesirable clustering results, as shown in Figure 2. Thus, we propose a new algorithm which combines the objective functions (1) and (2) into a single objective. The algorithm is a natural generalization of both Kmeans and Ksubspaces to simultaneous central/subspace clustering.

For the sake of simplicity, let us first assume that the subspaces are of co-dimension one, i.e. hyperplanes, so that we can represent them with a single normal vector $\boldsymbol{b}_j \in \mathbb{R}^D$. We discuss the extension to subspaces of varying dimensions in Remark 1. Our method computes the cluster centers and the subspace normals by solving the following optimization problem

$$\min \sum_{i=1}^{P} \sum_{j=1}^{n} \sum_{k=1}^{m_j} w_{ijk} \big((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2 \big) \quad (3)$$

$$\text{subject to } \boldsymbol{b}_j^\top \boldsymbol{b}_j = 1, j = 1, \dots, n, \quad (4)$$

$$\boldsymbol{b}_j^\top \mu_{jk} = 0, j = 1, \dots, n, k = 1, \dots, m_j, \quad (5)$$

$$\sum_{j=1}^{n} \sum_{k=1}^{m_j} w_{ijk} = 1, i = 1, \dots, P, \quad (6)$$

where $w_{ijk} \in \{0, 1\}$ denotes the membership of the ith point to the jkth cluster center. Equation (3) ensures that for each point $\boldsymbol{x}_i$, there is a subspace-cluster pair (j, k) such that both $|\boldsymbol{b}_j^\top \boldsymbol{x}_i|$ and $\|\boldsymbol{x}_i - \mu_{jk}\|$ are small. Equation (4) ensures that the normal vectors are of unit norm. Equation (5) ensures that each cluster center lies in its corresponding hyperplane and equation (6) ensures that each point is assigned to only one of the $\sum m_j$ cluster centers.

Using the technique of Lagrange multipliers to minimize the cost function in (3) subject to the constraints (4)–(6) leads to the new objective function

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{P} \sum_{j=1}^{n} \sum_{k=1}^{m_j} w_{ijk} ((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2) + \\ \sum_{j=1}^{n} \sum_{k=1}^{m_j} \lambda_{jk} (\boldsymbol{b}_j^\top \mu_{jk}) + \sum_{j=1}^{n} \delta_j (\boldsymbol{b}_j^\top \boldsymbol{b}_j - 1). \end{aligned} \quad (7)$$

Similarly to the Kmeans and Ksubspaces algorithms, we minimize $\mathcal{L}$ using a coordinate descent minimization technique, as shown in Algorithm 1. The following subsections describe each step of the algorithm in detail.

Initialization: Since the data points lie in a collection of hyperplanes, we can apply GPCA to obtain an estimate of the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and segment the data into n groups. Let $\boldsymbol{X}_j \in \mathbb{R}^{D \times P_j}$ be the set of points in the jth hyperplane. If we use the SVD of $\boldsymbol{X}_j$ to compute a rank $D - 1$ approximation of $\boldsymbol{X}_j \approx U_j S_j V_j$, where $U_j \in \mathbb{R}^{D \times (D-1)}$, $S_j \in \mathbb{R}^{(D-1) \times (D-1)}$ and $V_j \in \mathbb{R}^{(D-1) \times P_j}$, then the columns of $\boldsymbol{X}_j' = S_j V_j \in \mathbb{R}^{(D-1) \times P_j}$ are a set of vectors in $\mathbb{R}^{D-1}$ distributed around m_j cluster centers. We can apply Kmeans to segment the columns of $\boldsymbol{X}_j'$ into m_j groups and obtain the projected cluster centers $\{\mu_{jk}' \in \mathbb{R}^{D-1}\}_{k=1}^{m_j}$. The original cluster centers are then given by $\mu_{jk} = U_j \mu_{jk}' \in \mathbb{R}^D$.

Algorithm 1 (Combined Central and Subspace Clustering)

1. *Initialization*: Obtain an initial estimate of the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and cluster centers $\{\mu_{jk}\}_{j=1\dots n}^{k=1\dots m_j}$ using GPCA followed by Kmeans in each subspace.
2. *Computing the memberships*: Given the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$ and the cluster centers $\{\mu_{jk}\}_{j=1\dots n}^{k=1\dots m_j}$, compute the memberships $\{w_{ijk}\}$.
3. *Computing the cluster centers*: Given the memberships $\{w_{ijk}\}$ and the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$, compute the cluster centers $\{\mu_{jk}\}_{j=1\dots n}^{k=1\dots m_j}$.
4. *Computing the normal vectors*: Given the memberships $\{w_{ijk}\}$ and the cluster centers $\{\mu_{jk}\}_{j=1\dots n}^{k=1\dots m_j}$, compute the normal vectors $\{\boldsymbol{b}_j\}_{j=1}^n$.
5. *Iterate*: Repeat steps 2,3,4 until convergence of the memberships.

Computing the memberships: Since the cost function $\mathcal{L}$ is positive and linear in w_{ijk}, the minimum is attained at w_{ijk}=0. However, since $\sum_{jk} w_{ijk}$=1, the w_{ijk} multiplying the smallest $\big((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2\big)$ must be 1. Thus,

$$w_{ijk} = \begin{cases} 1 & \text{if } (j,k) = \arg\min \big((\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2\big) \\ 0 & \text{otherwise} \end{cases}.$$

Computing the cluster centers: From the first order condition for a minimum we have

$$\frac{\partial \mathcal{L}}{\partial \mu_{jk}} = -2 \sum_{i=1}^{P} w_{ijk} (\boldsymbol{x}_i - \mu_{jk}) + \lambda_{jk} \boldsymbol{b}_j = 0. \quad (8)$$

Left-multiplying (8) by $\boldsymbol{b}_j^\top$ and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$ and $\boldsymbol{b}_j^\top \boldsymbol{b}_j = 1$ yields

$$\lambda_{jk} = 2 \sum_{i=1}^{P} w_{ijk} (\boldsymbol{b}_j^\top \boldsymbol{x}_i). \quad (9)$$

Substituting (9) into (8) and dividing by two yields

$$-\sum_{i=1}^{P} w_{ijk} (\boldsymbol{x}_i - \mu_{jk}) + \sum_{i=1}^{P} w_{ijk} \boldsymbol{b}_j \boldsymbol{b}_j^\top \boldsymbol{x}_i = 0$$

$$\Longrightarrow \boxed{\mu_{jk} = (\mathcal{I} - \boldsymbol{b}_j \boldsymbol{b}_j^\top) \frac{\sum_{i=1}^{P} w_{ijk} \boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}}}$$

where $\mathcal{I}$ is the identity matrix in $\mathbb{R}^D$. Note that the optimal μ_{jk} has a simple geometric interpretation: it is the projection of the mean of the points associated with the jkth cluster onto the jth hyperplane.

Computing the normal vectors: From the first order condition for a minimum we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}_j} = 2\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)\boldsymbol{x}_i + \sum_{k=1}^{m_j}\lambda_{jk}\mu_{jk} + 2\delta_j \boldsymbol{b}_j = 0. \tag{10}$$

After left-multiplying (10) by $\boldsymbol{b}_j^\top$ to eliminate λ_{jk} and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$, we obtain

$$\delta_j = -\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2. \tag{11}$$

After substituting (9) into equation (10) and recalling that $\boldsymbol{b}_j^\top \mu_{jk} = 0$, we obtain

$$\boxed{(\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j \mathcal{I})\boldsymbol{b}_j = 0.} \tag{12}$$

Therefore, the optimal normal vector $\boldsymbol{b}_j$ is the eigenvector of $(\sum_{i=1}^{P}\sum_{k=1}^{m} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})$ associated with its smallest eigenvalue, which can be computed via SVD.

Remark 1 (Extension from hyperplanes to subspaces) *In the case of subspaces of co-dimension larger than one, each normal vector $\boldsymbol{b}_j$ should be replaced by a matrix of normal vectors $B_j \in \mathbb{R}^{D\times(D-d_j)}$, where d_j is the dimension of the jth subspace. Since the normal bases and the means must satisfy $B_j^\top \mu_{jk} = 0$ and $B_j^\top B_j = \mathcal{I}$, the objective function* (3) *should be changed to*

$$\mathcal{L} = \sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m_j} w_{ijk}(\|B_j^\top \boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_i - \mu_{jk}\|^2) + \sum_{j=1}^{n}\sum_{k=1}^{m_j}\lambda_{jk}^\top(B_j^\top \mu_{jk}) + \sum_{j=1}^{n} \textit{trace}\big(\Delta_j(B_j^\top B_j - \mathcal{I})\big).$$

where $\lambda_{jk} \in \mathbb{R}^{(D-d_j)}$ and $\Delta_j \in \mathbb{R}^{(D-d_j)\times(D-d_j)}$ are, respectively, vectors and matrices of Lagrange multipliers. Given the normal basis B_j, the optimal solution for the means is given by

$$\mu_{jk} = (\mathcal{I} - B_j B_j^\top)\frac{\sum_{i=1}^{P} w_{ijk}\boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}}.$$

One can show that the optimal solution for Δ_j is a scaled identity matrix whose jth diagonal entry is $\delta_j = -\sum_{i=1}^{P}\sum_{j=1}^{m_j} w_{ijk}\|B_j^\top \boldsymbol{x}_i\|^2$. Given δ_j and μ_{jk}, one can still solve for B_j from the null space of $(\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}(\boldsymbol{x}_i + \mu_{jk})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})$, which now has dimension $D - d_j$.

Remark 2 (Maximum Likelihood Solution) *Notice that in the combined objective function* (7) *the term $|\boldsymbol{b}_j^\top \boldsymbol{x}_i|$ is the distance to the jth hyperplane, while $\|\boldsymbol{x}_i - \mu_{jk}\|$ is the distance to the jkth cluster center. Since the former is mostly related to the variance of the noise in the orthogonal direction to the hyperplane, σ_b^2, while the latter is mostly related to the within class variance, σ_μ^2, the relative magnitudes of these two distances need to be taken into account. One way of doing so is to assume that the data is generated by a mixture of $\sum m_j$ Gaussians with means μ_{jk} and covariances $\Sigma_{jk} = \sigma_b^2\boldsymbol{b}_j\boldsymbol{b}_j^\top + \sigma_u^2(\mathcal{I} - \boldsymbol{b}_j\boldsymbol{b}_j^\top)$. This automatically allows the variances inside and orthogonal to the hyperplanes to be different. Application of the EM algorithm to this mixture model leads to the minimization of the following normalized objective function*

$$\mathcal{L} = \sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m_j} w_{ijk}\Big(\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{2\sigma^2} + \frac{\|\boldsymbol{x}_i - \mu_{jk}\|^2}{2\sigma_\mu^2} + \log(\sigma_b) + (D-1)\log(\sigma_u)\Big) + \sum_{j=1}^{n}\sum_{k=1}^{m_j}\lambda_{jk}(\boldsymbol{b}_j^\top \mu_{jk}) + \sum_{j=1}^{n}\delta_j(\boldsymbol{b}_j^\top \boldsymbol{b}_j - 1)$$

where $w_{ijk} \propto \exp(-\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{2\sigma^2} - \frac{\|\boldsymbol{x}_i - \mu_{jk}\|^2}{2\sigma_\mu^2})$ is now the probability that the ith point belongs to the jkth cluster center, and $\sigma^{-2} = \sigma_b^{-2} - \sigma_\mu^{-2}$. The optimal solution can be obtained using coordinate descent, similarly to Algorithm 1, with the following formulae for updating the parameters

$$\lambda_{jk} = 2\sum_{i=1}^{P} w_{ijk}\frac{\boldsymbol{b}_j^\top \boldsymbol{x}_i}{\sigma_\mu^2}, \quad \delta_j = -\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}\frac{(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{\sigma^2}$$

$$\mu_{jk} = (\mathcal{I} - \boldsymbol{b}_j\boldsymbol{b}_j^\top)\frac{\sum_{i=1}^{P} w_{ijk}\boldsymbol{x}_i}{\sum_{i=1}^{P} w_{ijk}}$$

$$0 = (\sum_{i=1}^{P}\sum_{k=1}^{m_j} w_{ijk}(\frac{\boldsymbol{x}_i}{\sigma^2} + \frac{\mu_{jk}}{\sigma_\mu^2})\boldsymbol{x}_i^\top + \delta_j\mathcal{I})\boldsymbol{b}_j$$

$$\sigma_b^2 = \frac{\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m_j} w_{ijk}(\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2}{\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m_j} w_{ijk}}$$

$$\sigma_u^2 = \frac{\sum_{ijk} w_{ijk}\big(\|\boldsymbol{x}_i - \mu_{jk}\|^2 - (\boldsymbol{b}_j^\top \boldsymbol{x}_i)^2\big)}{(D-1)\sum_{i=1}^{P}\sum_{j=1}^{n}\sum_{k=1}^{m_j} w_{ijk}}.$$

3. Experiments

3.1. Clustering performance on simulated data

We randomly generate $P = 600$ data points in $\mathbb{R}^3$ lying in 2 intersecting planes $\{S_j\}_{j=1}^2$ with 3 clusters in each plane $\{\mu_{jk}\}_{j=1,2}^{k=1,2,3}$. 100 points are drawn around each one of the six cluster centers according to a zero-mean Gaussian distribution with standard deviation $\sigma_\mu = 1.5$ within each plane. The angle between the two planes is randomly chosen from $20^o \sim 90^o$, and the distance among the three cluster centers is randomly selected in the range $2.5\sigma_\mu \sim 5\sigma_\mu$.

Zero-mean Gaussian noise with standard deviation σ_b is added in the direction orthogonal to each plane. Using simulated data, we compare 5 different clustering methods:

- Kmeans clustering in $\mathbb{R}^3$ using 6 cluster centers, then merging them into 2 planes[1] (**KM**),
- MPPCA[2] clustering in $\mathbb{R}^3$ using 6 cluster centers, then merging them into 2 planes[1] (**MP**),
- Ksubspaces clustering in $\mathbb{R}^3$ using 2 planes, then Kmeans using 3 clusters within each plane (**KK**),
- GPCA clustering in $\mathbb{R}^3$ using 2 planes, then Kmeans using 3 clusters within each plane (**GK**),
- GPCA-Kmeans clustering for initialization followed by combined central and subspace clustering (**JC**) as described in Section 2 (Algorithm 1).

Figure 3 shows a comparison of the performance of these five methods in terms of clustering error ratios and the error in the estimation of the subspace normals in degrees. The results are the mean of the errors over 100 trials. It can be seen in Figure 3 that the errors in clustering and normal vectors of all five algorithms increase as a function of noise. **MP** performs better than **KM** and **KK** for large levels of noise, because of its probabilistic formulation. The two stage algorithms, **KK**, **GK** and **JC**, in general perform better than **KM** and **MP** in terms of clustering error. The random initialization based methods, **KM**, **MP** and **KK**, have non-zero clustering error even with noise-free data. Within the two stage algorithms, **KK** begins to experience subspace clustering failures more frequently with more severe noises, due to its random initialization, while GPCA in **GK** and **JC** employ an algebraic solution of one-shot subspace clustering, thus avoiding the initialization problem. The subspace clustering errors of **KK** can cause the estimate of the normals to be very inaccurate, which explains why **KK** has worse errors in the normal vectors than **KM** and **MP**. In summary, **GK** and **JC** have smaller average errors in clustering and normal vectors than **KM**, **MP** and **KK** . The combined optimization procedure of **JC** converges within $2 \sim 5$ iterations according to our experiments, which further advocates **JC**'s clustering performance.

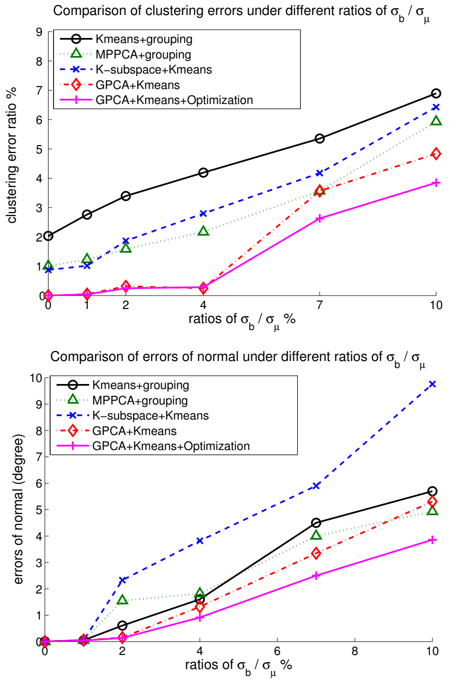


Figure 3. **Top:** Clustering error as a function of noise in the data. **Bottom:** Error in the estimation of the normal vectors (degrees) as a function of the level of noise in the data.

3.2. Applications with real data

3.2.1. Illumination-invariant face clustering

The Yale face database B (see *http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html*) contains a collection of face images $I_j \in \mathbb{R}^K$ of 10 subjects taken under 576 viewing conditions (9 poses × 64 illumination conditions). Here we only consider the illumination variation for face clustering in the case of frontal face images. Thus our task is to sort the images taken for the same person by using our combined central/subspace clustering algorithm. As shown in (Ho et al., 2003), the set of all images of a (Lambertian) human face with fixed pose taken under all lighting conditions forms a cone in the image space which can be well approximated by a low dimensional subspace. Thus images of different subjects live in different subspaces. Since the number of pixels K of each image is in general much larger than the dimension of the underlying subspace, PCA (Duda et al., 2000) is first employed for dimensionality reduction. Successful GPCA clustering results have been reported by (Vidal et al., 2005) for a subset of 3x64 images of subjects 5, 8 and 10. The images in (Vidal et al., 2005) are cropped to 30x40 pixels and 3 PCA components are used as image features in homogeneous coordinates.

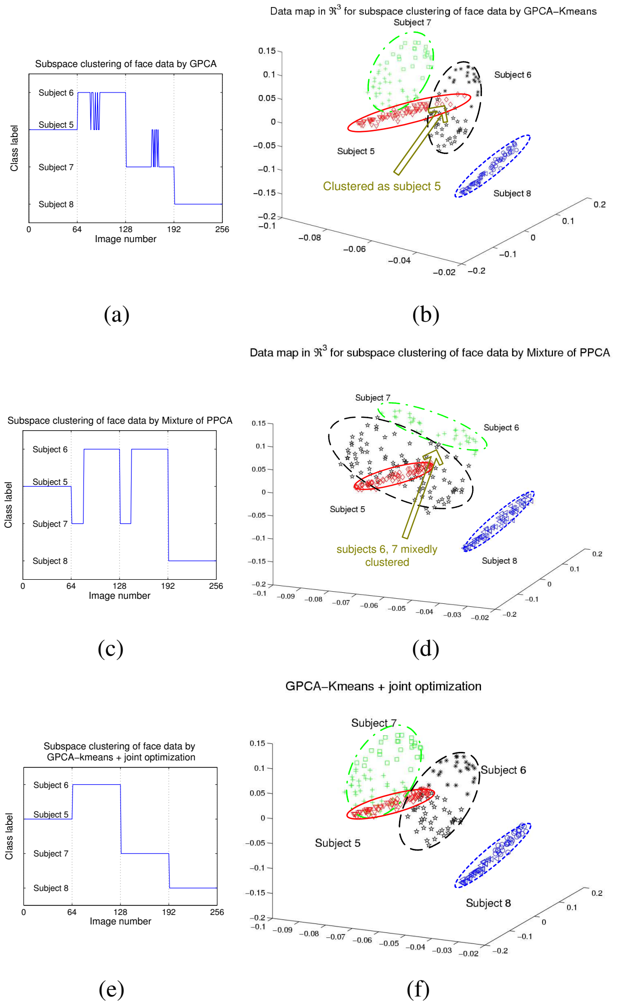
In this subsection, we further explore the performance of combined central/subspace face clustering under more complex imaging conditions. We keep 3 PCA components for 4x64 (240x320 pixels) images of subjects 5, 6, 7, and

[1] In order to estimate the plane normals, we group the 6 clusters returned by **KM** or **MP** into 2 planes. The idea is that 3 clusters which lie in the same plane have the dimensionality of 2 instead of 3. A brute-force search with $\binom{6}{3}/2$ selections is employed to find the 2 best fitting planes, by considering the minimal strength of the data distributed in the third dimension via Singular Value Decomposition (Duda et al., 2000).

[2] Software available at www.ncrg.aston.ac.uk/netlab/

8, which gives more background details (as shown in Figure 4). Figures 5 (a,b) show the imperfect clustering result of GPCA due to the intersection of the subspace of subject 5 with the subspaces of subjects 6 and 7. GPCA assigns all the images on the intersection to subject 5. Mixtures of PPCA is implemented in Netlab as a probabilistic variation of subspace clustering with one spatial center per subspace. It can be initialized with Kmeans (originally in Netlab) or GPCA, both of which result in imperfect clustering. We show one example of the subspaces of subjects 6 and 7 mixed (Kmeans initialization) in Figure 5 (c,d). Our combined subspace-central optimization process successfully corrects the wrong labels for some images of subjects 6 and 7, as demonstrated in Figure 5 (e,f). In the optimization, the local clusters in the subspaces of subjects 6 and 7 contribute with smaller central distances to their misclassified images, which re-classifies them to the correct subspaces using our combined subspace-central clustering algorithm. In this experiment, 4 subspaces with 2 clusters per subspace are used. Compared with the results in (Vidal et al., 2005), we obtain perfect illumination-invariant face clustering for a more complex data distribution.

Figure 4. Sample images of subjects 5, 6, 7 and 8.



(a) (b)

(c) (d)

(e) (f)

Figure 5. Illumination-invariant face clustering by GPCA (a-b), Mixtures of PPCA (c-d), and our method (e-f). Plots on the right show 3 principal components with proper labels and color-shapes. The colors match the colors of subjects 5, 6, 7 and 8 in Figure 4.

3.2.2. Video Shot Segmentation

Unlike face images under different illumination conditions, video data provides continuous visual signals. Video structure parsing and analysis applications need to segment the whole video sequence into several video shots. Each video shot may contain hundreds of image frames which are either captured with a similar background or have a similar semantical meaning.

Figure 6 shows 2 sample videos, *mountain.avi* and *drama.avi*, containing 4 shots each. Archives are publicly available from *http://www.open-video.org*. For the mountain sequence, 4 shots are captured. The shots display different backgrounds and show either multiple dynamic objects and/or severe camera motions. In this video, the frames between each pair of successive shots are gradually blended from one to another. Because of this, the correct video shot segmentation is considered to split every two successive shots at their blending frames. In order to explore how the video frames are distributed in feature space, we plot the first 3 PCA components for each frame in Figure 7 (b, d, f). Note that a manifold structure can be observed in Figure 7 (f), where we manually label each portion of the data as shots 1 through 4 (starting from red dots to green, black and ending in blue) according to the result of our clustering method. The video shot segmentation results of the mountain sequence by Kmeans, GPCA and GPCA-Kmeans followed by combined optimization are shown in Figure 7 (a,b), (c,d) and (e,f), respectively. Because Kmeans is based on the central distances among data, it segments the data into spatially close blobs. There is no guarantee that these spatial blobs will correspond to correct video shots. Comparing Figure 7 (b) with the correct segmentation in (f), the Kmeans algorithm splits shot 2 into clusters 2 and 3, while it groups shots 1 and 4 into cluster 1. By considering the data's manifold nature, GPCA provides a more effective approximation with multiple planes to the manifold in $\mathbb{R}^3$ than the spatial blobs given by central clustering. The essential problem for GPCA is that it only deploys the co-planar condition in $\mathbb{R}^3$, without any constraint relying on their spatial locations. In the structural approximation of the data's manifold, there are many intersecting data points among 4 planes. These data points represent video frames with the clustering ambiguity solely based on the subspace constraint. Fortunately this limitation can be well tackled by GPCA-Kmeans with combined optimization. Combining central and subspace distances provides correct video shot clustering results for the mountain sequence, as demonstrated in Figure 7 (e,f).

The second video sequence shows a drama scenario which is captured with the same background. The video shots should be segmented by the semantic meaning of the performance of the actor and actress. In Figure 6 **Right**, we show 2 sample images for each shot. This drama video sequence contains very complex actor and actress' motions in front of a common background, which results in a more complex manifold data structure[3] than that of the mountain video. For better visualization, the normal vectors of data samples recovered by GPCA or the combined central/subspace optimization, are drawn originating from each data point in $\mathbb{R}^3$ with different colors for each cluster. For this video, the combined optimization process shows a smoother clustering result in Figure 8 (c,d), compared with (a,b). In summary, GPCA can be considered as an effective way to group data in a manifold into multiple subspaces or planes in $\mathbb{R}^3$ which normally better represent video shots than central clustering. GPCA-Kmeans with combined optimization can then associate the data at the intersection of planes into the correct clusters by optimizing combined distances. Subspace clustering seems to be a better method to group the data on a manifold by somehow preserving their geometric structure. Central clustering, such as Kmeans[4], provides a piecewise constant approximation; while subspace clustering shows a piecewise linear approximation. On the other hand, subspace clustering can meet severe clustering ambiguity problems when the shape of the manifold is complex, as shown in Figure 8 (b,d). In this case, there are many intersections of subspaces so that subspace clustering results can be very sparse, without considering the spatial coherence. Combined optimization of central and subspace distances demonstrates superior clustering performance with real video sequences.

3.2.3. Discussion on Model Selection

Throughout the paper we have assumed that the number of subspaces n, their dimensions d_j and the number of clusters within each subspace m_j are known. In practice, these quantities may not be known beforehand.

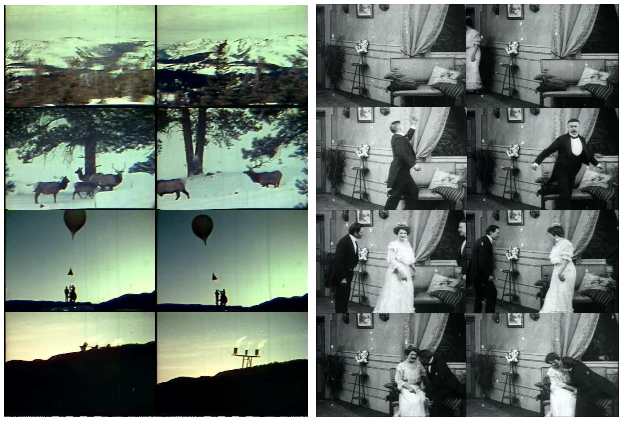
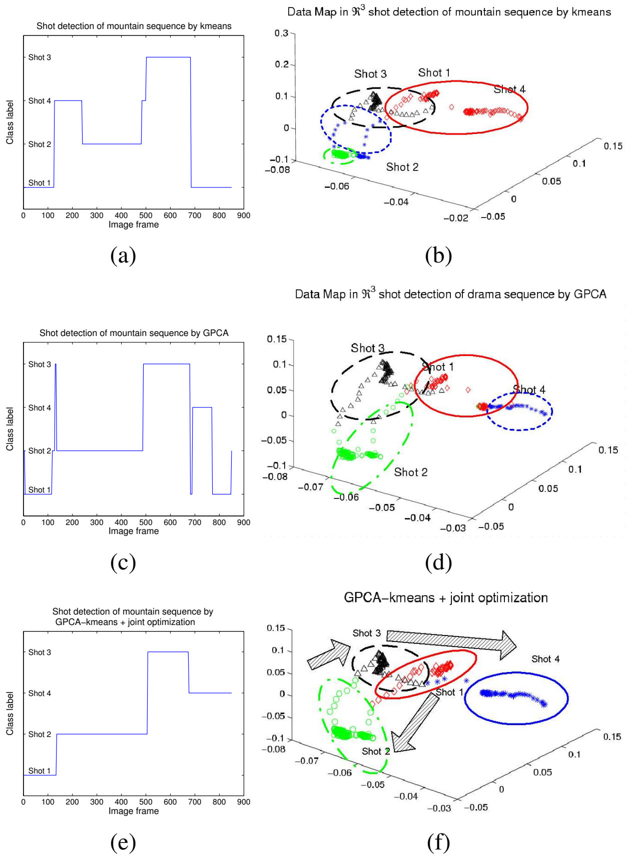


Figure 6. Sample images used for video shot segmentation. **Left:** mountain sequence. **Right:** drama sequence.



(a) (b)

(c) (d)

(e) (f)

Figure 7. Video shot segmentation of mountain sequence by Kmeans (a-b), GPCA (c-d) and our algorithm (e-f). Plots on the right show 3 principal components of the data grouped in 4 clusters shown by ellipses with proper color-shapes. In (f), three arrows show the topology of the video manifold.

[3] Because there are image frames of transiting subject motions from one shot to another, the correct video shot segmentation is considered to split successive shots at their transiting frames.

[4] Due to space limitation, we do not provide the clustering result using Kmeans for this sequence which is similar with Figure 7 (a,b).

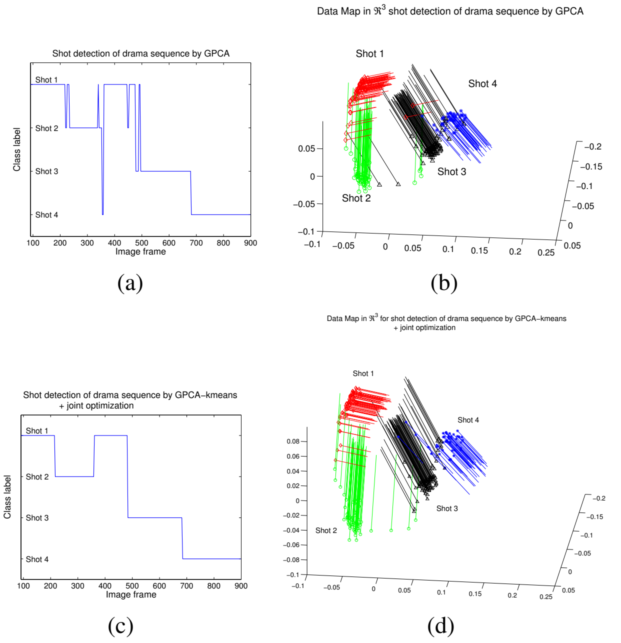


Figure 8. Video shot segmentation of drama sequence by GPCA (a-b), and our algorithm (c-d). Plots on the right show 3 principal components of the data with the normal to the plane at each point. Different normal directions illustrate different shots.

When the number of subspaces is $n = 1$, the estimation of the dimension of the subspace d_1 is essentially equivalent to the estimation of the number of principal components of the data set. This problem can be tackled by combining PCA with existing model selection techniques, such as minimum description length, Akaike information criterion, or Bayesian information criterion (Duda et al., 2000). Given d_1, the number of clusters m_1 can be determined by combining the Kmeans cost functional with the aforementioned model selection criteria.

When the number of subspaces is $n > 1$, the problem is much more challenging. One possible solution is to employ model selection algorithms for subspace and central clustering separately in a sequential manner, to determine n first, then d_j and then m_j. As shown in (Vidal et al., 2005), GPCA provides a way of determining n from a rank constraint on a polynomial embedding of the data. Given n, one may cluster the data using GPCA, and then determine the dimension of each subspace as the number of principal components of the data points that belong to each subspace. Given n and d_j, one can use the model selection procedure mentioned earlier to determine the number of clusters m_j in Kmeans. However, this three-stage solution is clearly not optimal. Ideally one would like a model selection criteria that integrates both types of clustering into one joint or combined process. This is obviously more difficult than combining the clustering algorithms, and is under current investigation.

4. Conclusions and Future Work

We have proposed an intuitive and easy to implement algorithm for clustering data lying in a union of subspaces with multiple clusters within each subspace. By minimizing a cost function that incorporates both central and subspace distances, our algorithm can handle situations in which Kmeans and Ksubspaces/GPCA fail, e.g., when data are close to the intersection of two subspaces, or when cluster centers in different subspaces are spatially close. Future work includes using model selection to automatically determine the number of subspaces and cluster centers. Also, we believe it should be possible to extend the proposed combined central and subspace clustering formulation to recognize multiple complex curved manifolds. An example application is to find which movie a given images appear in. Each manifold will be composed of multiple subspaces where each subspace is spatially constrained by central distances among data samples. Once the movie models are learned (similarly to shot detection), the likelihood evaluation for a new data sample is based on computing its combined central and subspace distances to the given models.

Acknowledgments

This work was supported by Hopkins WSE startup funds, and by grants NSF CAREER IIS-0447739, NSF CRS-EHS-0509101, and ONR N00014-05-1-0836.

References

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1–38.

Duda, R., Hart, P., & Stork, D. (2000). *Pattern classification*. Wiley, New York. 2nd edition.

Ho, J., Yang, M.-H., Lim, J., Lee, K.-C., & Kriegman, D. (2003). Clustering apperances of objects under varying illumination conditions. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 11–18).

Tipping, M., & Bishop, C. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation*, *11*, 443–482.

Vidal, R., & Ma, Y. (2004). A unified algebraic approach to 2-D and 3-D motion segmentation. *European Conference on Computer Vision* (pp. 1–15).

Vidal, R., Ma, Y., & Sastry, S. (2005). Generalized Principal Component Analysis (GPCA). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *27*, 1–15.

Weinberger, K. Q., & Saul, L. K. (2004). Unsupervised learning of image manifolds by semidefinite programming. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 988–995).