Localized K-flats*

Yong Wang[1,2], **Yuan Jiang**[2], **Yi Wu**[1], **Zhi-Hua Zhou**[2]
[1] Department of Mathematics and Systems Science
National University of Defense Technology, Changsha 410073, China
yongwang82@gmail.com, wuyi_work@sina.com
[2] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{jiangy, zhouzh}@lamda.nju.edu.cn

Abstract

K-flats is a model-based linear manifold clustering algorithm which has been successfully applied in many real-world scenarios. Though some previous works have shown that K-flats doesn't always provide good performance, little effort has been devoted to analyze its inherent deficiency. In this paper, we address this challenge by showing that the deteriorative performance of K-flats can be attributed to the usual reconstruction error measure and the infinitely extending representations of linear models. Then we propose Localized K-flats algorithm (LKF), which introduces localized representations of linear models and a new distortion measure, to remove confusion among different clusters. Experiments on both synthetic and real-world data sets demonstrate the efficiency of the proposed algorithm. Moreover, preliminary experiments show that LKF has the potential to group manifolds with nonlinear structure.

Introduction

Traditional clustering algorithms (e.g., K-means) assume each cluster is centered around a single point. However, a large number of recent research efforts have demonstrated that distributions of many real data sometimes naturally fall into clusters grouped around low-dimensional linear manifolds (i.e., linear or affine subspaces) (Vidal, Ma, and Sastry 2005; Yan and Pollefeys 2006). Thus, a reasonable and promising generalization of classical clustering is *linear manifold clustering*, which changes the entity of the cluster center from being a point to that of being a linear manifold.

Generally speaking, existing algorithms on linear manifold clustering can be roughly classified into two main categories: model-based algorithms and similarity-based algorithms. Since each linear manifold can be equivalently modeled by its offset vector and normal vectors, model-based algorithms (such as Generalized Principal Component Analysis (GPCA) (Vidal, Ma, and Sastry 2005) and K-flats (Bradley and Mangasarian 2000; Tseng 2000)) learn these parameters from the data and then assign each point to the nearest linear manifold. On the other hand, similarity-based algorithms, such as Local Subspace Affinity (LSA) (Yan and Pollefeys 2006) and Spectral Curvature Clustering (SCC) (Chen and Lerman 2009), first build an affinity matrix among certain points to measure their possibilities of coming from the same manifold, and then apply spectral clustering (Shi and Malik 2000; Ng, Jordan, and Weiss 2001) to group the samples into K clusters.

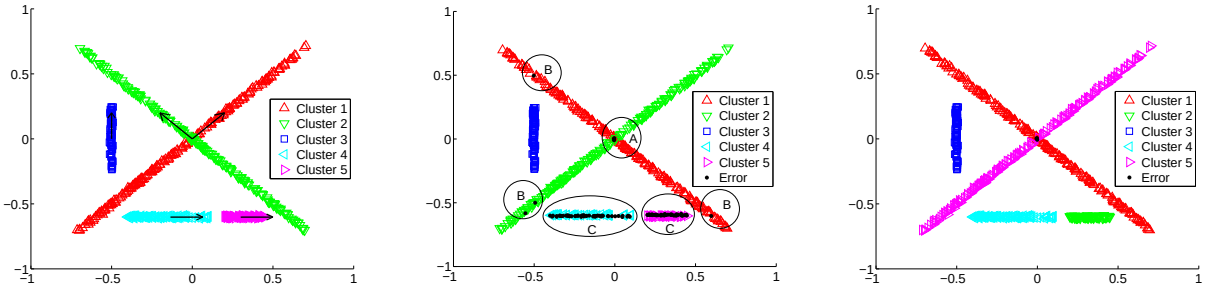
Among these algorithms, K-flats is more attractive due to its low computational complexity and its ability to handle both linear and affine subspaces explicitly. Thus, K-flats has been successfully applied in a variety of applications, such as face clustering and motion segmentation. However, previous works also have shown that clustering using K-flats may deteriorate seriously in many scenarios, for example, when it was used to group affine subspaces (Lu and Vidal 2006; Chen and Lerman 2009). Unfortunately, there has been little effort devoted to analyze its inherent deficiency and then improve its performance. Moreover, it is not clear on how to extend K-flats to group manifolds with nonlinear structure.

In this paper, we try to deal with these challenges. Our analysis reveals that K-flats suffers from three kinds of deterioration which are mainly rooted in the reconstruction error measure and the infinitely extending representations of linear models. Then, we propose Localized K-flats algorithm (abbreviated as LKF), which introduces localized representations of linear models and a new distortion measure into the objective function of K-flats, to remove confusion among different clusters. Experiments on both synthetic data sets and real-world applications compare our proposed algorithm favorably against state-of-the-art algorithms. Moreover, preliminary experiments illustrate the potential of LKF to group manifolds with nonlinear structure.

The remainder of this paper is divided into the following parts: Section 2 gives a brief review of the related works. Section 3 presents the details of LKF. In Section 4, we report experiments on both synthetic and real-world data. Section 5 concludes this paper.

Notation: Throughout this paper, A^T and $span^{\perp}(A)$ represent the transpose and the orthogonal complement space of A, respectively. $\|\cdot\|$ and $\|\cdot\|_F$ denote the 2-norm and F-norm, respectively. $\langle A, B\rangle_F = tr(A^T B)$. I always denotes the identity matrix and e denotes a vector of ones, both of their dimensions should be apparent from the context.

*This research was partially supported by NSFC (60975038, 61005003, 61073097) and 973 Program (2010CB327903).



(a) Original data (b) Grouping result of K-flats (c) Grouping result of Localized K-flats

Figure 1: Different performance of K-flats and our proposed LKF on the synthetic data. (a) A set of points in $\Re^2$ drawn from 5 clusters, where clusters 4 and 5 lie approximately around the same affine line but are well-separated. (b) Grouping using K-flats misclassifies several points which can be classified into three kinds of errors: intrinsic errors, infinity errors and co-linear errors, labeled as A, B and C, respectively. (c) Grouping using LKF significantly improves the clustering performance.

Related Work

Let $\{x_i \in \Re^D,\ i = 1, \cdots, N\}$ be a set of points lying approximately on K distinct d-dimensional ($0 < d < D$) linear manifolds (or flats) $\Omega_j = \{x : B_j^T(x - \mu_j) = 0\} \subseteq \Re^D$, $j = 1, \cdots, K$, with normal bases $B_j \in \Re^{D\times(D-d)}$ and $\mu_j \in \Re^D$. The objective of linear manifold clustering is to group the samples into the flat they belong to.

The objective of K-flats is to determine K cluster flats by minimizing the sum of the squared distances of each point to a nearest flat (Bradley and Mangasarian 2000; Tseng 2000). Formally, K-flats solves for μ_j and B_j by minimizing the following cost function:

$$\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{K} w_{ij}\left\|B_j^T(x_i - \mu_j)\right\|^2, \qquad (1)$$

subject to the constraints $B_j^T B_j = I$, $\sum_{j=1}^{K} w_{ij} = 1$, and $w_{ij} \in \{0, 1\}$.

In practice, K-flats alternates between cluster update and cluster assignment to determine K cluster flats and find K clusters. Let $X_1, \cdots, X_K$ be a partition of the data where X_j is the $D \times N_j$ ($\sum_{j=1}^{K} N_j = N$) matrix whose columns correspond to all samples which are assigned to the j-th cluster. Then, *Cluster Update* computes the optimal solution for B_j as the eigenvectors of $X_j(I - ee^T/N_j)X_j^T$ corresponding to its $D - d$ smallest eigenvalues and $\mu_j = X_j e/N_j$. Given μ_j and B_j, *Cluster Assignment* allocates x_i to the closest flat Ω_k according to the reconstruction error measure:

$$\left\|B_k^T(x_i - \mu_k)\right\|^2 = \min_{j=1,\cdots,K}\left\|B_j^T(x_i - \mu_j)\right\|^2. \qquad (2)$$

Due to low computational complexity and wide applicable domain, K-flats has received much attention and has been successfully used in many application scenarios, such as face clustering and motion segmentation (Vidal, Ma, and Sastry 2005; Chen and Lerman 2009). However, existing researches also have shown that clustering using K-flats doesn't always provide good performance. Previous works focus on two issues: good initialization and robustness. K-flats is an iterative scheme and frequently converges to a local minima, thus good initialization is important to its success. Instead of random initialization, (Vidal, Ma, and Sastry 2005) initializes K-flats with GPCA to speed up convergence while (Zhang et al. 2010) searches for a good set of K seed flats from $C > K$ local best-fit candidate flats. To improve the robustness of K-flats to outliers and noise, (Zhang, Szlam, and Lerman 2009) replaces the 2-norm average of equation (1) with the more robust 1-norm average.

Though existing works have solved some limitations of K-flats and improved its performance to a certain extent, they do not touch the inherent deficiency of K-flats (e.g., why K-flats seriously deteriorates when faced with affine subspaces as revealed by the experiments in (Chen and Lerman 2009)) which is more meaningful and challenging.

The Proposed Method

In our opinion, the deteriorative performance of K-flats is mainly rooted in the adopted reconstruction error measure which is the distance from the sample to the linear manifold. To shed more insight on this problem, let us consider a 2-D example in Figure 1. There are five clusters, of which two lie approximately around linear subspaces and three lie around affine subspaces. Different clusters are colored and marked differently. Moreover, there are two clusters lie approximately around the same affine line but are well-separated such that humans might interpret them as different clusters. Figure 1 (b) shows that K-flats misclassifies several points, and we can generally classify them into three kinds of errors: intrinsic errors, infinity errors and co-linear errors. *Intrinsic errors* happen on the points close to the intersection areas which by nature are ambiguous. Without other information, even an expert can not judge which cluster a point around the intersection should be grouped into. *Infinity error* misclassifies some points in one cluster to another cluster when the points are closer to the latter according to the reconstruction error measure. This kind of deterioration is ascribed to the infinite expansibility of linear models, which is contradicted to the intended local representations of clusters. Moreover, it is this kind of errors results in the deteriorative performance of K-flats when faced with affine subspaces, since linear subspaces can only intersect around the

origin. This gives a explanation to the empirical results obtained earlier in (Chen and Lerman 2009). The points from the well-separated but co-linear clusters are indistinguishable (named *co-linear errors*) since they have similar cost with respect to both clusters under the reconstruction error measure. One skill to partly deal with co-linear errors is that we can first deem the two co-linear clusters as one cluster and then use K-flats followed by K-means to finally group them (Lu and Vidal 2006). However, this strategy will be significantly influenced by the infinity errors. Moreover, it could work well only when there are no elongated clusters.

From the above analysis, we can see that intrinsic errors can not be avoided due to their natural ambiguities. While, both infinity errors and co-linear errors can be boiled down to the usual reconstruction error measure and the infinitely extending representations of linear models. Thus, for better performance, a new distortion measure which can localize the representations of linear models seems necessary.

Infinity errors produced by the contradiction between infinitely extending linear models and the intended local representations of clusters suggest that we can introduce some localized information to restrict the linear models. Moreover, we also need some extra information to distinguish between the co-linear points. To address these problems, we bring K-means into the objective function of K-flats. K-means will force the points to center around some prototypes and thus will localize the representations of linear models. More specifically, the new distortion measure (cost function) considers both the reconstruction error and the cluster centers:

$$\sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\left(\left\|B_j^T(x_i-\mu_j)\right\|^2+\lambda\|x_i-\mu_j\|^2\right), \tag{3}$$

where $0<\lambda<1$ is a parameter to control the relative magnitudes of the reconstruction error and the usual Euclidean distance $\|x_i-\mu_j\|^2$. This new cost function will force the representation along the linear manifold to localize around the prototype μ_j since it is strictly monotonically increasing with respect to $\|x_i-\mu_j\|^2$. Thus, λ is also used to control the localization of the linear manifolds.

However, when a point has the same reconstruction error to two clusters, it will be assigned according to the value of K-means. Unfortunately, it is well-known that K-means will deteriorate when there are elongated clusters (Shi and Malik 2000; Ng, Jordan, and Weiss 2001) since it tries to minimize the within-class variance. For example, it will misclassify some points in the fourth cluster in Figure 1 (a) to the fifth cluster. To circumvent this problem, we can increase the number of training linear models from K to $M>K$. The underlying idea is that each local piece of the elongated clusters are not elongated and thus can be correctly grouped by K-means. After getting the correctly grouped local clusters, we can merge them to obtain the desired K clusters.

Specifically, our method computes μ_m and B_m, $m=1,\cdots,M$, by minimizing the following problem:

$$\sum_{i=1}^{N}\sum_{m=1}^{M} w_{im}\left(\left\|B_m^T(x_i-\mu_m)\right\|^2+\lambda\|x_i-\mu_m\|^2\right), \tag{4}$$

where $B_m^TB_m=I$, $\sum_{m=1}^{M}w_{im}=1$ and $w_{im}\in\{0,1\}$.

Similar to K-flats, our method first alternates between *Cluster Update* and *Cluster Assignment* to learn μ_m and B_m, and then assigns each point into its closest flat. Now, we give the details of these procedures.

Initialization: Assign each point in X to a cluster randomly or using K-means, to give an initial partition $X_1,\cdots,X_M$ such that $w_{im}=1$ when the i-th point is assigned to X_m.

Cluster Update: This step will find a center μ_m and a set of bases B_m for the m-th cluster such that the new cost function is minimum. More precisely, given the membership w_{im}, we can simplify the optimization problem (4) as:

$$\sum_{m=1}^{M}\left(\left\|B_m^T(X_m-\mu_me^T)\right\|_F^2+\lambda\left\|X_m-\mu_me^T\right\|_F^2\right). \tag{5}$$

The global optimal solution of (5) renders the partial derivatives of the Lagrangian of (5) vanish. That is, for

$$\mathcal{L}(B_m,\mu_m,\Lambda_m)= \sum_{m=1}^{M}\begin{pmatrix}\left\|B_m^T(X_m-\mu_me^T)\right\|_F^2+\lambda\left\|X_m-\mu_me^T\right\|_F^2\\ -\left\langle\Lambda_m,B_m^TB_m-I\right\rangle_F\end{pmatrix}, \tag{6}$$

where $\Lambda_m\in\Re^{(D-d)\times(D-d)}$ are symmetric, we have:

$$(1/2)\nabla_{B_m}\mathcal{L}=(X_m-\mu_me^T)(X_m-\mu_me^T)^TB_m-B_m\Lambda_m =0, \tag{7}$$

$$-(1/2)\nabla_{\mu_m}\mathcal{L}=(B_mB_m^T+\lambda I)(X_m-\mu_me^T)e=0, \tag{8}$$

$$-\nabla_{\Lambda_m}\mathcal{L}=B_m^TB_m-I=0. \tag{9}$$

It is easy to show that $B_mB_m^T+\lambda I$ is positive definite, thus equation (8) implies that $\mu_m=X_me/N_m$, which is the m-th cluster center. Substitution for μ_m in (7) gives:

$$X_m(I-ee^T/N_m)X_m^TB_m=B_m\Lambda_m. \tag{10}$$

Similar to Theorem 2.1 in (Tseng 2000), we can prove that B_m is given by the eigenvectors of $X_m(I-ee^T/N_m)X_m^T$ corresponding to its $D-d$ smallest eigenvalues. Interestingly, though our method has different distortion measure to that of K-flats, both μ_m and B_m are formally the same. The difference is that $M>K$ linear models are trained here while K-flats only trains K linear models.

Cluster Assignment: Given μ_m and B_m, the new distortion measure is positive and linear with respect to w_{im}. Under the constraints $w_{im}\in\{0,1\}$ and $\sum_{m=1}^{M}w_{im}=1$, we have

$$w_{im}=\begin{cases}1 \text{ if } (i,m)=\\ \quad\arg\min\left(\left\|B_m^T(x_i-\mu_m)\right\|^2+\lambda\|x_i-\mu_m\|^2\right),\\ 0 \text{ otherwise.}\end{cases} \tag{11}$$

Similar to Theorem 7 in (Bradley and Mangasarian 2000), we can show that this iterative procedure will terminate in a finite number of steps. In practice, we stop whenever there is a non-decrease in the overall objective function.

After training, we obtain M localized linear models where each model is represented by $\theta_m=\{\mu_m,B_m\}$. Moreover, all the original data are grouped into these models based on our new distortion measure and thus we obtain M localized clusters $X_1,\cdots,X_M$ whose local structural information are characterized by θ_m. These localized clusters

Algorithm **Localized K-flats (LKF)**

Input: Data set X, number of clusters K, dimension of manifolds d, balance parameter λ, number of localized models M, number of neighbors P, power parameter o.

Steps:

1: Train M localized linear models $\theta_m = \{\mu_m, B_m\}$ and partition the data into M localized clusters.

2: Construct an undirected graph on the localized clusters with the affinity matrix A defines through (12).

3: Partition the vertices of the graph into K clusters using spectral clustering methods.

Output: A partition of the data into K disjoint clusters.

Figure 2: Pseudo-code of the Localized K-flats algorithm

should be merged together to get the desired K clusters. To this end, similar to similarity-based linear manifold clustering algorithms, we utilize the localized linear models to define an affinity matrix to measure their possibilities of coming from the same cluster. To be specific, the localized clusters correspond to the vertices of an undirected graph and the affinity between X_k and X_l is defined as follows:

$$A_{kl} = \begin{cases} (\prod_{e=1}^{d} \cos(\theta_e))^o & \text{if } X_k \in P - connected(X_l) \\ & \text{or } X_l \in P - connected(X_k), \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

where $o \in \mathrm{N}^+$ and $X_k \in P - connected(X_l)$ means that X_k has a point which is among the P nearest neighbors of X_l. $0 \leq \theta_1 \leq, \cdots, \leq \theta_d \leq \pi/2$ are a series of principal angles between two spaces $\Theta_k = span^{\perp}(B_k)$ and $\Theta_l = span^{\perp}(B_l)$, which are defined recursively as:

$$\cos(\theta_e) = \max_{\substack{u \in \Theta_k, v \in \Theta_l \\ \|u\|=\|v\|=1}} u^{\mathrm{T}} v = {u_e}^{\mathrm{T}} v_e, \quad e = 1, \cdots, d, \quad (13)$$

where $u^{\mathrm{T}} u_i = 0$, $v^{\mathrm{T}} v_i = 0$, $i = 1, \cdots, e-1$.

Then, we could use spectral clustering algorithms (Shi and Malik 2000; Ng, Jordan, and Weiss 2001) on the affinity matrix A to give the final clustering results. The pseudo-code of our algorithm, named as Localized K-flats (LKF), is shown in Figure 2. Theoretically, the computational complexity of our method is $O(DMNt_1 + (D^3M + DMN^2)t_2 + [M^3 + M^2(Dd^2 + K) + (D+P)N^2] + NK^2 t_3)$, where t_1, t_2 and t_3 are the number of iterations before the convergence of K-means, iterative procedures of LKF, and spectral clustering, respectively.

Experiments

In this section, we experimentally evaluate the performance of LKF on a series of synthetic data sets and real-world applications. Clustering accuracy is used as the performance evaluation criterion, which is defined as the maximum classification accuracy among all possible alignments:

$$\text{clustering accuracy} = \max_{align} \sum_{i=1}^{N} \delta(t_i = c_i)/N, \quad (14)$$

where t_i is the true label and c_i is the obtained cluster label of x_i, $\delta(\cdot)$ is the delta function.

Table 1: Comparison of the clustering accuracy (%) and the average computational time (in seconds) of the different algorithms on the synthetic data set.

Approach	Best	Mean	Time
K-means	60.57	58.21	0.03
GPCA	78.71	78.71	0.01
K-flats	86.57	76.90	0.01
LSA	59.57	59.57	22.40
SCC	95.43	79.20	2.69
LKF	**99.43**	**98.80**	0.51

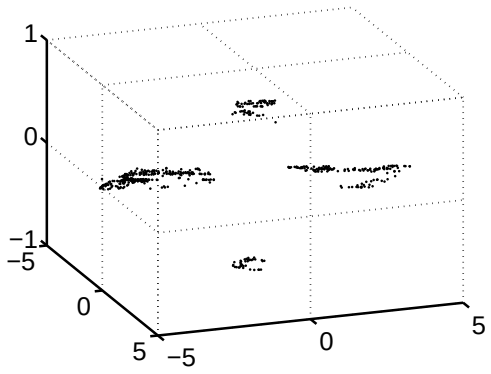


Figure 3: 3D-projections of the tracked feature points of the cars3 data set.

Experiments on Synthetic Data

In this subsection, we illustrate LKF on a synthetic toy problem to give some insight about its effectiveness. We set $\lambda = 0.005$, $M = 50$, $P = 10$, and $o = 8$ here, the influence of parameters on LKF will be studied later.

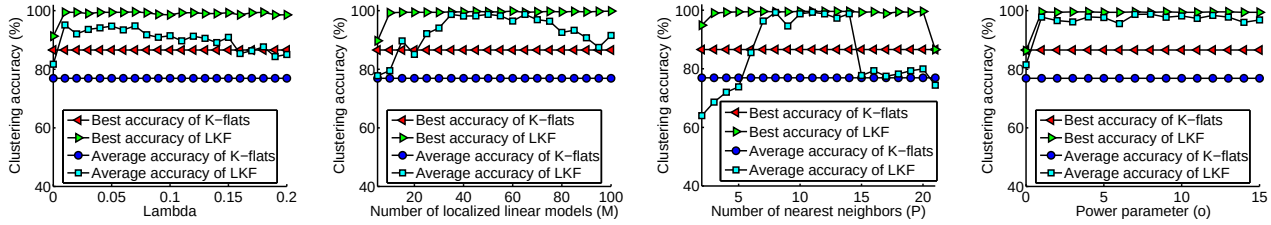
The synthetically generated data points, as illustrated in Figure 1 (a), are sampled from five clusters in $\Re^2$ where two of them (i.e., cluster 4 and cluster 5) lie approximately around the same affine line but are well-separated. Each of the first two clusters has 200 points while each of the last three clusters has 100 samples. We also add zero-mean Gaussian noise to the sample points. As we have shown in Figure 1 (b), the grouping result using K-flats in terms of the usual reconstruction error measure and the infinitely extending representations of linear models is not very satisfying. However, as we can see from Figure 1 (c), the clustering result of LKF is surprisingly good: Even when there are approximately co-linear and elongated clusters, our method reliably groups the samples into the clusters consistent with what a human would have chosen. Our method still misclassifies some points near the intersection of two clusters, however, this is acceptable as we have pointed out that these points are essentially ambiguous.

Since LKF is designed to improve the performance of K-flats which is one of the linear manifold clustering algorithms, it is interesting and meaningful to compare it with state-of-the-art algorithms. Thus, in the following, we compare LKF with K-means, GPCA, K-flats, LSA, and SCC.

We conduct 30 independent trials with random initialization for all the algorithms which may be plagued by local

Table 2: Clustering accuracy (mean±std) of the different algorithms on five video sequences. The best are boldfaced.

Data set	Cars2-06	Cars2B	Cars3	Cars4	Trucks2
Points/Frames/Clusters	123/15/4	535/30/4	548/20/4	147/54/3	331/22/3
K-means	0.784±0.042	0.760±0.023	0.617±0.047	0.761±0.177	0.613±0.015
GPCA	0.919±0.000	0.598±0.000	0.938±0.000	0.721±0.000	0.731±0.000
K-flats	0.927±0.093	0.702±0.091	0.722±0.043	0.809±0.088	0.860±0.103
LSA	0.846±0.000	0.677±0.000	0.583±0.001	0.857±0.000	0.646±0.005
SCC	0.812±0.049	0.908±0.099	0.717±0.089	0.737±0.054	0.649±0.062
LKF	**1.000±0.000**	**1.000±0.000**	**0.996±0.016**	**0.999±0.004**	**1.000±0.000**



(a) Different λ values (b) Different localized models M (c) Different neighbors P (d) Different o values

Figure 4: Parameter influence on Localized K-flats. See text for details on each display.

minima, such as K-means and SCC. The results are tabulated in Table 1, which show that: a). Generally, all the linear manifold clustering algorithms perform better than K-means which is based on the assumption that each cluster is centered around a prototype. b). The performance of all other algorithms are not very satisfying due to the infinite expansibility of linear models and the impact of the co-linear clusters. c). Our proposed LKF does the best.

Experiments on Real Data

In this experiment, we test LKF and all other algorithms on five motion segmentation sequences from (Tron and Vidal 2007). The first two rows of Table 2 list the name of the sequences, the number of points N, frames F and clusters K, respectively. (Vidal, Ma, and Sastry 2005) has shown that each object spans a different subspace under the affine camera model and hence all types of motion segmentation problems are equivalent to finding different subspaces. However, as pointed out by (Boult and Brown 1991) "such definitions, in terms of subspaces, may not totally capture the intuitive notion of 'different motions'. If there are two well separated clusters within a single subspace, humans might interpret them as different motions". One example is illustrated in Figure 3 by projecting the cars3 data set onto the 3D space for visualization using PCA. Obviously, there are four clusters where two of them approximately lie in the same linear subspace. The proposed LKF and all other algorithms are used to segment these five video sequences into multiple spatiotemporal regions. Except for GPCA which requires to work in a low-dimensional space, all other algorithms are executed in the original $D = 2F$ dimensional space. GPCA is performed in a 5-dimensional ambient space following the original authors' suggestion (Vidal, Ma, and Sastry 2005). Table 2 shows the results of the different algorithms and our proposed method achieves significantly better performance.

Parameter Influence

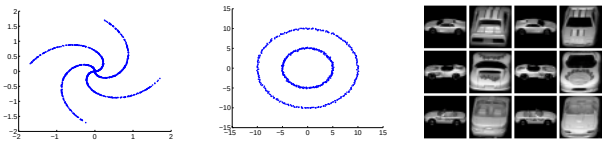
There are four adjustable parameters in our LKF algorithm (i.e., λ, M, P and o) whose optimal values generally depend on the distribution of the samples and the noise level. Traditionally, we can best set them by using domain knowledge. However, it is also interesting and sensible to examine the impact of these parameters, and then obtain some guidelines to their setting. Figure 4 shows the performance of our proposed method as a function of the concerned parameter by fixing other parameters on the synthetic data.

From these results, it is easy to see that LKF achieves significantly better results than K-flats over a large range of parameter setting. Also, we can get several interesting observations: a). LKF works well when λ is relatively small but larger than zero. This phenomenon is reasonable as $\lambda = 0$ corresponds to the original K-flats and thus the representations of the linear models extend infinitely in the data space, while large λ makes the algorithm prone to K-means. b). The performance of LKF is insensitive to both M and P, as long as they are neither too small nor too large, which is consistent with many existing observations (Roweis, Saul, and Hinton 2002; Saul and Roweis 2004). The reason is that small M is not enough to avoid the misclassification of points from the elongated clusters, while large M will cause the local models sensitive to noise (Roweis, Saul, and Hinton 2002). Moreover, there may be many disconnected sub-clusters when P is too small, while local restriction will lose when it is too large (Saul and Roweis 2004). c). LKF seems insensitive to o as long as it is large enough.

From these observations, we recommend to set $\lambda = 0.005$ and $o = 8$. M and P will be tuned according to the distribution of the samples and the noise level.

Handling Nonlinear Structure

Generally, linear manifold clustering algorithms are designed to group linear manifolds. However, manifolds in real



(a) Two-spirals data (b) Two-circles data (c) Three-cars data

Figure 5: Three nonlinear data sets: (a) Two spirals data ($N = 1000$). (b) Two circles data ($N = 3000$). (c) Twelve images of three cars ($N = 216$).

data are generally nonlinear (Saul and Roweis 2004) and all the linear algorithms fail to deliver good performance in the presence of nonlinear structure. Therefore, nonlinear models have received much attention recently (Souvenir and Pless 2005; Wang et al. 2010). Though LKF is initially proposed to deal with the infinity errors and co-linear errors of the linear K-flats algorithm, it fortunately has the potential ability to group manifolds with nonlinear structure. The reason is that we can usually characterize the manifolds with complicated global nonlinear structure by a collection of local linear models (Roweis, Saul, and Hinton 2002) and our proposed method exactly has this ability due to its localized representations of manifolds.

In these preliminary experiments, LKF is compared with all linear manifold clustering algorithms and two nonlinear algorithms (i.e., K-manifolds (Souvenir and Pless 2005) and mumCluster (Wang et al. 2010)) on three highly nonlinear data sets (see Figure 5): two synthetic data sets and one real data set from the processed COIL-20 database[1]. The corresponding clustering accuracy of the different algorithms are shown in Table 3. The results reveal several interesting observations: a). Traditional linear algorithms (i.e., GPCA, K-flats, LSA and SCC) do not work well on these nonlinear databases, due to their linear property. b). Generally, nonlinear models (i.e., K-manifolds, mumCluster and LKF) give better performance. c). It is easy to see that the proposed LKF performs superior to all other algorithms.

Conclusions

In this paper, we analyze the inherent deficiency of the original K-flats algorithm in detail and propose LKF to remove confusion among different linear models. Experimental results on both synthetic data sets and real-world applications have demonstrated the efficiency of the proposed method and its potential to group manifolds with nonlinear structure.

References

Boult, T. E., and Brown, L. G. 1991. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, 179–186.

Bradley, P. S., and Mangasarian, O. L. 2000. k-plane clustering. *Journal of Global Optimization* 16(1):23–32.

[1]http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Table 3: Clustering accuracy (mean±std. followed by the highest accuracy in the parentheses) of the different algorithms on three data sets with nonlinear structure.

Approach	Two-spirals	Two-circles	Three-cars
GPCA	0.524±0.000 (0.524)	0.501±0.000 (0.501)	0.375±0.000 (0.375)
K-flats	0.550±0.001 (0.552)	0.500±0.000 (0.501)	0.351±0.013 (0.384)
LSA	0.500±0.000 (0.500)	0.500±0.000 (0.500)	0.365±0.002 (0.366)
SCC	0.552±0.020 (0.602)	0.502±0.001 (0.504)	0.343±0.004 (0.352)
K-manifolds	0.784±0.189 (0.976)	0.528±0.026 (0.575)	0.417±0.032 (0.495)
mumCluster	0.969±0.000 (0.969)	**1.000±0.000** (1.000)	0.507±0.031 (0.519)
LKF	**0.975±0.061** (0.995)	**1.000±0.000** (1.000)	**0.632±0.108** (0.875)

Chen, G. L., and Lerman, G. 2009. Spectral curvature clustering (SCC). *IJCV* 81(3):317–330.

Lu, L., and Vidal, R. 2006. Combined central and subspace clustering for computer vision applications. In *ICML*, 593–600.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Roweis, S.; Saul, L. K.; and Hinton, G. E. 2002. Global coordination of local linear models. In *NIPS*, 889–896.

Saul, L. K., and Roweis, S. T. 2004. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research* 4(2):119–155.

Shi, J. B., and Malik, J. 2000. Normalized cuts and image segmentation. *PAMI* 22(8):888–905.

Souvenir, R., and Pless, R. 2005. Manifold clustering. In *ICCV*, 648–653.

Tron, R., and Vidal, R. 2007. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 41–48.

Tseng, P. 2000. Nearest q-flat to m points. *Journal of Optimization Theory and Applications* 105(1):249–252.

Vidal, R.; Ma, Y.; and Sastry, S. 2005. Generalized principal component analysis (GPCA). *PAMI* 27(12):1945–1959.

Wang, Y.; Jiang, Y.; Wu, Y.; and Zhou, Z.-H. 2010. Multi-manifold clustering. In *PRICAI*, 280–291.

Yan, J. Y., and Pollefeys, M. 2006. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 94–106.

Zhang, T.; Szlam, A.; Wang, Y.; and Lerman, G. 2010. Randomized hybrid linear modeling by local best-fit flats. In *CVPR*, 1927–1934.

Zhang, T.; Szlam, A.; and Lerman, G. 2009. Median k-flats for hybrid linear modeling with many outliers. In *IEEE Workshops on ICCV*, 234–241.