RESEARCH ARTICLE

Multiple Manifold Clustering Using Curvature Constrained Path

Amir Babaeian[1]*, **Alireza Bayestehtashk**[2], **Mojtaba Bandarabadi**[3]

1 Department of Mathematics, University of California San Diego, San Diego, California, United States of America, **2** Department of Computer Science, Oregon Health and Science University, Portland, Oregon, United States of America, **3** Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

* ababaeian@ucsd.edu

OPEN ACCESS

Citation: Babaeian A, Bayestehtashk A, Bandarabadi M (2015) Multiple Manifold Clustering Using Curvature Constrained Path. PLoS ONE 10(9): e0137986. doi:10.1371/journal.pone.0137986

Editor: Xuhui Huang, Hong Kong University of Science and Technology, HONG KONG

Received: January 27, 2015

Accepted: August 25, 2015

Published: September 16, 2015

Data Availability Statement: The FOX video data is available http://math.duke.edu/∼glchen/scc.html. The human motion capture data used in this project was obtained from mocap.cs.cmu.edu. The Coil data set is available from http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Funding: This work was all supported by a grant from the National Science Foundation (DMS- 09-15160).

Competing Interests: The authors have declared that no competing interests exist.

Abstract

The problem of multiple surface clustering is a challenging task, particularly when the surfaces intersect. Available methods such as Isomap fail to capture the true shape of the surface near by the intersection and result in incorrect clustering. The Isomap algorithm uses shortest path between points. The main draw back of the shortest path algorithm is due to the lack of curvature constrained where causes to have a path between points on different surfaces. In this paper we tackle this problem by imposing a curvature constraint to the shortest path algorithm used in Isomap. The algorithm chooses several landmark nodes at random and then checks whether there is a curvature constrained path between each landmark node and every other node in the neighborhood graph. We build a binary feature vector for each point where each entry represents the connectivity of that point to a particular landmark. Then the binary feature vectors could be used as a input of conventional clustering algorithm such as hierarchical clustering. We apply our method to simulated and some real datasets and show, it performs comparably to the best methods such as K-manifold and spectral multi-manifold clustering.

Introduction

We consider the problem of clustering points that are sampled in the vicinity of multiple surfaces embedded in Euclidean space, with a particular interest in the case where these intersect. The goal is multi-manifold clustering, which amounts to labeling each point according to the surface it comes from. This stylized problem may be relevant in a number of applications, such as the extraction of galaxy clusters [1] and road tracking [2] and target tracking [3–7] after some preprocessing. In motion segmentation [8–10] and in face recognition [11–13], the underlying surfaces are usually assumed to be affine or, more generally, algebraic. Here we focus on a nonparametric setting where the main assumption is that the surfaces are smooth—see Fig 1 for an example. This appears to be necessary to remove ambiguities in the problem of separating intersecting surfaces.

Several approaches have been proposed in this context. Most methods are designed for the case where the surfaces do not intersect [14–16], while others work only when the surfaces that intersect have different intrinsic dimension or density [17, 18]. The method of [19] is only able

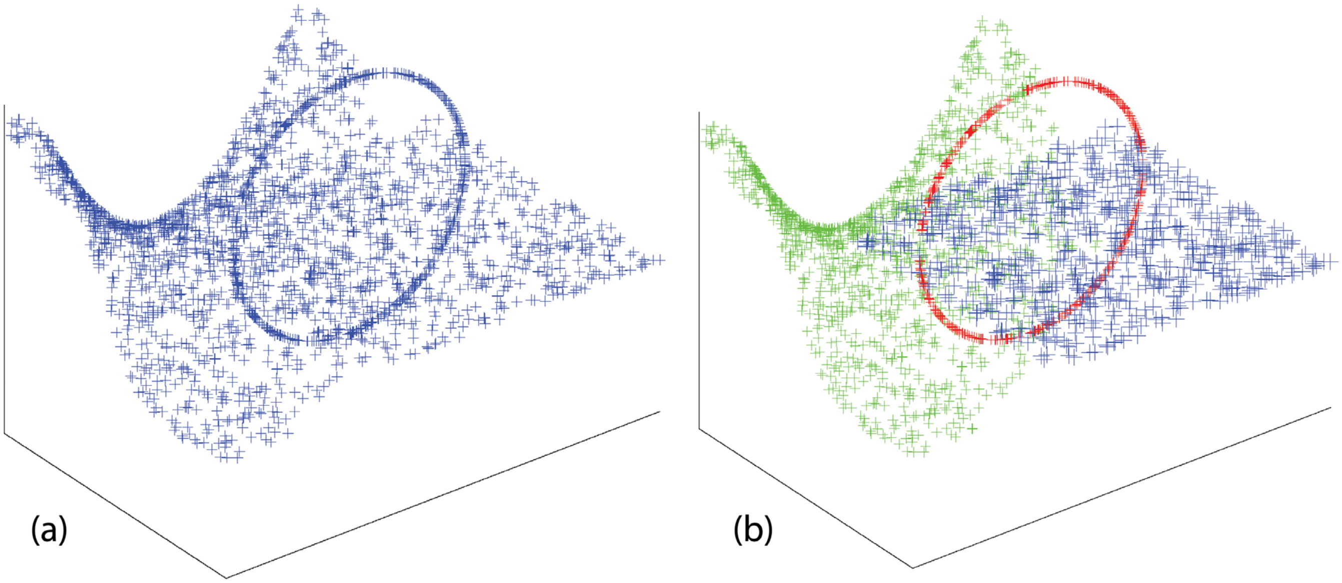

Fig 1. Simulated data illustrating the problem of multi-manifold clustering. Left: 3D data. Right: output from our method.

doi:10.1371/journal.pone.0137986.g001

to separate intersecting curves. Methods that purposefully aim at resolving intersections are fewer. Souvenir et al. [20] implement some variant of K-means [21–23] where the centers are surfaces. Guo et al. [24] propose to minimize a (combinatorial) energy that includes local orientation information, using a tabu search. The state-of-the-art method lies in methods based on local principal component analysis (PCA). An early proposal was the elaborate multiscale spectral method of [25], while the clustering routine of [26]—developed in the context of semi-supervised learning—inspired the works of [27] and [28].

We propose a markedly different approach based on connecting points to landmarks via curvature-constrained paths. It can be seen as a constrained variant of [29]. Isomap was specifically designed for dimensionality reduction in the single-manifold setting, and in particular, cannot handle intersections. It has been used in different applications [30–34]. The curvature constraint on paths is there to prevent connecting points from one cluster to points from a different, intersecting cluster. The resulting algorithm is implemented as a simple variation of Dijkstra's algorithm. Our method is simpler than the previous proposals in the literature and performs comparably to the best methods, both on simulated and real datasets.

The rest of the paper is organized as follows. In the next section we explain the notion of curvature constrained shortest-path and it's connection with the curvature constrained shortest-path. In the algorithm section we present our algorithm for multi-manifold clustering and compare it with three currently applied methods and give a theoretical guarantee for that. In the numerical experiments section we performed multiple numerical experiments on simulated and real data. Robustness of method to noise and choice of constraint is discussed as well. In the discussion section we discuss and outline our future work and development of our algorithm.

Constrained path

Neighborhood graph

Neighborhood graphs play a central role in manifold learning, exploiting the fact that smooth submanifolds are locally flat. Recall that a neighborhood graph is a graph with vertices the sample points $x_1, \ldots, x_N$. We consider two types of neighborhood structure [35]:

- ε-*ball*. x_i and x_j are connected if $\|x_i - x_j\| \le \varepsilon$, where $\| \cdot \|$ denotes the Euclidean norm.
- k-*nearest neighbor*. x_i and x_j are connected if x_j is among the k-nearest neighbors of x_i (in the Euclidean metric), or vice-versa.

Angle constraint

The central idea in this paper is the use of *constrained* shortest-path distances in a neighborhood graph. The paths are constrained in order to control their smoothness. The constrained shortest-path distances are used to estimate geodesic distances reliably, even when the surface self-intersects, thus allowing us to mimic Isomap. We use the fact that the constrained and unconstrained shortest-path distances are similar for points belonging to the same submanifold, while usually different for points belonging to different submanifolds.

For an ordered triplet of points (x, y, z) in $\mathbb{R}^D$, define its angle as

$$\angle(x, y, z) = \angle(\overrightarrow{xy}, \overrightarrow{yz}) = \cos^{-1}\left(\frac{\langle y - x, z - y\rangle}{\| y - x \| \| z - y \|}\right) \in [0, \pi] \tag{1}$$

We say that a sequence of points $(x_{i_1}, \ldots, x_{i_m})$ is θ-angle constrained if the angles between successive segments are all bounded by θ, meaning

$$\angle(x_{i_{t-1}}, x_{i_t}, x_{i_{t+1}}) \le \theta, \quad \forall t = 2, \cdots, m - 1. \tag{2}$$

Curvature constraint

For an ordered triplet of points (x, y, z) in $\mathbb{R}^D$, we define the curvature as

$$\operatorname{curv}(x, y, z) = \begin{cases} (R(x, y, z))^{-1}, & \text{if } \angle(x, y, z) < \frac{\pi}{2}, \\ \infty, & \text{otherwise,} \end{cases} \tag{3}$$

where ∠ stands for the angle and $R(x, y, z)$ is the radius of the circle passing through x, y, z.

$$R(x, y, z) = \frac{\sqrt{\| x - y \|^2 + \| z - y \|^2 + 2 \| x - y \| \| z - y \| \cos\angle(x, y, z)}}{\sin\angle(x, y, z)}. \tag{4}$$

with $R(x, y, z) = \infty$ if x, y, z are aligned.

Definition. For a curvature $\kappa > 0$, we say that a path $(x_{i_1}, \ldots, x_{i_m})$ is κ-constrained if $\operatorname{curv}(x_{i_{t-1}}, x_{i_t}, x_{i_{t+1}}) \le \kappa, \quad \forall t = 2, \ldots, m - 1$.

Definition. For every point y in the graph the annulus neighborhood of that point is a set of points on the graph that are within $\varepsilon/2$ and ε distance of point y.

Fig 2 shows three D-dimensional points x, y, z which form vertices of a triangle such that x and z belong to the annulus neighborhood of point y. Under above assumption the angle constraint $\angle(x, y, z) < \theta$ where $\theta < \pi/2$ implies curvature constraint $\operatorname{curv}(x, y, z) < \kappa$ where $\kappa = 2sin(\theta)/\sqrt{\frac{\varepsilon^2}{2}(1 + cos(\theta))}$.

Our algorithm

To compute these constrained shortest-path distances we use a simple modification of Dijkstra's algorithm. Then input of the algorithm is the *KNN* graph, angle or curvature constraint and landmark points. Algorithm compute the constrained shortest path distance between

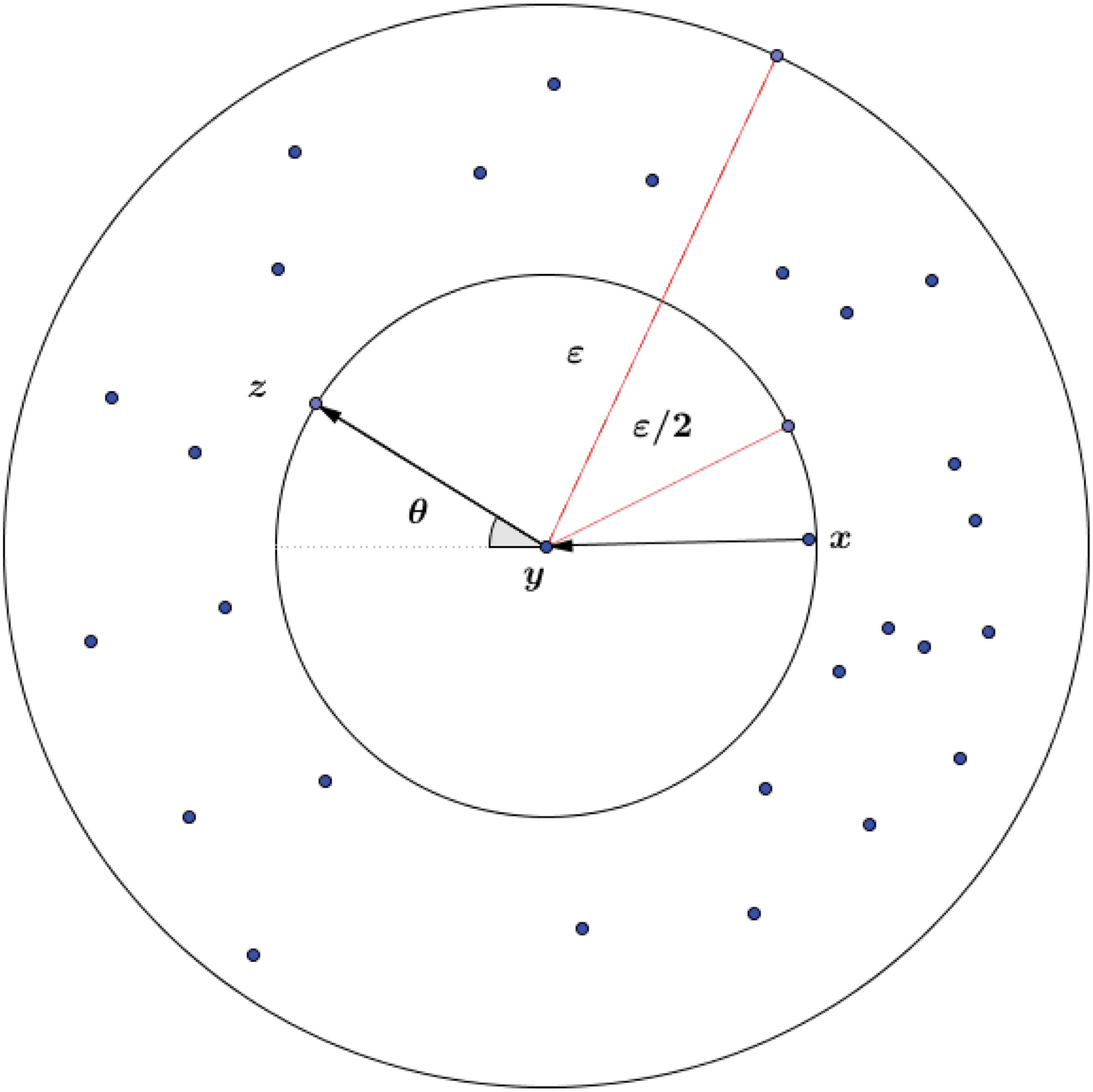

Fig 2. *x* and *z* lie in annulus neighborhood of point *y*.

doi:10.1371/journal.pone.0137986.g002

every land mark points and all the points in the graph by taking in to the account that if the path between three consecutive points violate the constraint then that path would be considered as infinity or actually there is no path. The main difference of our algorithm with Dijkstra's algorithm is that we need to check the constraint during Dijkstra's process when we consider the neighboring nodes of our current node. Suppose we are in current node x and y and z are the only neighbors of x and these two nodes in Dijkstra's process are not visited yet and the weight from x to y is less than weight of x to z, But the path from $parnts[x]$, x, y violate the constraint but $parnt[x]$, x, z does not violate, so in this case we consider $parent[x]$, x, z as the right path. There is no guarantee always there exist such a path, which means in this case there is no path from that land mark and a given point in the graph. See algorithm 1 below. When applied to a neighborhood graph with maximum degree Δ, its computational complexity is $O(\Delta N \log N)$ per source point.

```
Algorithm 1 Curvature Constrained Shortest Path Algorithm
Input: neighborhood graph $\mathcal{G}$ including weights of all the connected vertecies, the landmark $x_\ell$ where $\ell = 1, \ldots, L$, angle or curvature constraint $\theta$.
for $\ell = 1$ to $L$ do
  $ss = \ell$, $t = \ell$, $m = 0$.
  For each vertex $i$ of the graph ($i = 1, \ldots, N$): $distance[i] = Inf$, $cost[i] = Inf$, $parent[i] = Inf$, $temporary[i] = 0$.
  At the beginning for each vertex $i$ of the graph, there is no path from that vertex to the landmark $x_\ell$, i.e., $path[\ell][i] = [\,]$.
  for $j = 1$ to $N$ do
    Update $distance[i]$ for each $i \in neighbors(t)$ by the weight of edge between the vertex $i$ and $t$.
    for $i = 1$ to $N$ do
      if $distance[i] + m < cost[i]$ then
        if curv($i$, $t$, $ss$) $< \kappa$ or $t == ss$ then
          Update $parent[i] = t$ and $cost[i] = distance[i] + m$.
        end if
      end if
    end for
    Compute $temporary + cost$ vector then find the minimum element of these vector as well as the vertex $I$ with minimum element.
    Update $m$, i.e., $m = min(temporary + cost)$.
    if $parent[i] \sim= Inf$ then
      Update $path[\ell][I]$ by appending vertex $I$ to the end of $path[\ell][parent[I]]$.
    end if
    $temporary[I] = Inf$.
    $distance[i] = Inf$ for all $i = 1, \ldots, N$.
    Update $t = I$ and choose $ss$ as the parent of t, i.e., $ss = parent[t]$.
    Update weights of edge from vertex $ss$ to $t$ and from vertex $ss$ to $t$ by $Inf$ in order to avoid revisiting vertices. Our graph is a directed graph, so it is possible to have edges in both directions between two vertices.
  end for
end for
Output: Constrained shortest-Path from each vertex $i$ of the graph and each landmark $\ell$.
```

Multi-Manifold Clustering

Existing methods

The last decade saw a flurry of propositions aiming at clustering data points when the underlying clusters are not convex, and in particular, in the situation where the points are sampled near low-dimensional objects. We gave a few references in the Introduction and now want to elaborate on three of them, [20, 36] and [27], as we will use them as benchmarks in our experiments. Our choice was dictated by performance, code availability and relevance to our particular setting.

The method of [25] renders impressive results but is hard to tune, having many parameters, while the method of [28] is very similar to that of [27] and the code was not publicly available at the moment of writing this paper. The other methods for multi-manifold clustering that we know of were not designed to resolve intersections of clusters of possibly identical intrinsic dimensions and sampling densities.

We chose the subspace clustering method of [36] among a few others methods that perform well in this context.

K-Manifolds. Souvenir et al. [20] suggest an algorithm that mimics K-means, replacing centroid points with centroid submanifolds. The method starts like Isomap by building a neighborhood graph and computing shortest path distances within the graph. After randomly initializing a K-by-n weight matrix $W = (w_{ki})$, where w_{ki} represent the probability that point i belongs to the kth cluster, it alternates between an M-Step and an E-Step. In the M-Step, for each k, the points are embedded in $\mathbb{R}^K$ using a weighted variant of multidimensional scaling using the weights $(w_{ki} : i = 1, \ldots, n)$. In the E-Step, for each k and i, the normal distance of point x_i to the cluster k is estimated as

$$\delta_{ki} = \frac{\sum_j w_{kj}(d(x_i, x_j) - d_k(x_i, x_j))}{\sum_j w_{kj}},$$

where $d(x_i, x_j)$ denotes the shortest path distance in the neighborhood graph and $d_k(x_i, x_j)$ denotes the Euclidean distance in the kth embedding, between points x_i and x_j. The weights are then updated as $w_{ki} \propto \exp(-d_{ki}^2/\sigma^2)$ such that $\Sigma_k w_{ki} = 1$ for all i, where σ^2 is chosen automatically.

Spectral Curvature Clustering. Chen et al. [36] proposed a spectral method for subspace clustering—the setting where the underlying surfaces are affine. We will compare our method to theirs when the surfaces are affine, and also when the surfaces are curved. The latter is done as a proof of concept, for it will be very clear that it cannot handle curved surfaces, like any other method for subspace clustering we know of. The procedure assumes that all subspaces are of same dimension d, which is a parameter of the method. For each (d+2)-tuple, $x_{i_1}, \ldots, x_{i_{d+2}}$, it computes a notion of curvature $C_{i_1, \ldots, id+2}$ which measure how well approximated this $(d + 2)$-tuple is by an affine subspace of dimension d. After reducing the tensor $\mathbf{C} = (C_{i_1, \ldots, id+2} : i_t = 1, \ldots, N)$ spectral graph partitioning [15] is applied.

Spectral Multi-Manifold Clustering. Wang et al. [27] suggest a spectral method using a dissimilarity that factors in the Euclidean distance and the discrepancy between the local orientation of the data. The surfaces are assumed to be of same dimension d known to the user. First, a mixture of probabilistic principal component analyzers [37] are fitted to the data, approximating the point cloud by a union of d-planes. This is used to estimate the tangent subspace at each data point. The dissimilarity between two points is then an increasing function of their Euclidean distance and the principal angles between their respective affine subspaces. These dissimilarities are fed to the spectral graph partitioning method of [15].

Our algorithm

We consider the following problem of surface clustering. Given a sample $x_1, \ldots, x_n \in \mathbb{R}^D$ sampled from $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_K$, where for each k, $\mathcal{S}_k$ is a smooth, but possibly self-intersecting surface, label each point according to the surface it belongs to. Our algorithm is quite distinct from all the other methods for multi-manifold clustering we are aware of, although it starts by building a q-nearest neighbor graph like many others. The idea is very simple and amounts to clustering together points that are connected by an angle-constrained path in the neighborhood graph. Take two surfaces S_1 and S_2 intersecting at a strictly positive angle. Then for 'most' pairs of data points $x_{i_1} \in S_1$ and $x_{i_2} \in S_2$, a path in the graph going from x_{i_1} to x_{i_2} has at least one large angle between two successive edges, on the order of the incidence angle between the surfaces; while for 'most' pairs of data points $x_{i_1}, x_{i_2} \in S_1$, there is a path with all angles between successive edges relatively small. To speedup the implementation, we select M landmarks (with M slightly larger than K) at random among the data points and only identify what data points are connected to what landmark via a κ-constrained path in the graph. M and κ are parameters of the algorithm. Let $\xi_{\ell i} = 1$ if point i and landmark ℓ are connected that way, and $\xi_{\ell i} = 0$ if not. We use $\xi_i := (\xi_{\ell i} : \ell = 1, \ldots, M)$ as feature vectors that we group together and cluster using hierarchical clustering with complete linkage.

```
Algorithm 2 Path-Based Clustering (PBC)
Input: data $(x_i)$; parameters $q$, $K$, $M$, $\kappa$
Build $q$-nearest neighbor graph
Choose $M$ landmarks are random
for $i = 1$ to $n$ do
    For each landmark $\hat{x}_\ell$, identify which points $x_i$ it is connected to via a $\kappa$-constrained path in the graph, and set $\xi_{\ell i} = 1$ if so, and $\xi_{\ell i} = 0$ otherwise.
end for
Group and then apply hierarchical clustering to the feature vectors $\xi_1, \ldots, \xi_n$ to find $K$ clusters, where $\xi_i := (\xi_{\ell i} : \ell = 1, \ldots, M)$.
```

Intersections. We are most interested in the case where the surfaces intersect. Concretely, given K compact, simply connected submanifolds $S_1, \ldots, S_K \subset \mathbb{R}^D$ of maximum pointwise curvature bounded by $\kappa < \infty$, we consider the noisy mixture distribution

$$x = s + z, \quad s \sim \sum_{k=1}^{K} \pi_k \mu_{S_k} \quad z \sim \mu_{B(0,\tau)}, \tag{5}$$

where μ_S denotes the uniform distribution over set S.

Definition. We say that two smooth submanifolds S_1 and S_2 have incidence angle $\alpha \in (0, \pi/2)$ at an intersection point $x \in S_1 \cap S_2$ if the smallest nonzero principal angle between the tangent subspaces of S_1 and S_2 at x is equal to α.

We assume that, for any pair of underlying surfaces S_k and S_ℓ, the minimum incidence angle between them at any point along their intersection is at least $\alpha > 0$. The compactness of the surfaces imply the existence of a function $\omega(\varepsilon) \to 0$ with $\varepsilon \to 0$ such that, for any $\varepsilon > 0$, if dist $(x, S_k) \vee$ dist $(x, S_\ell) \leq \varepsilon$, then dist $(x, S_k \cap S_\ell) \leq \omega(\varepsilon)$. Otherwise, there is $C > 0$ such that, for all $m \geq 1$, there is x_t satisfying dist $(x_t, S_k) \vee$ dist $(x_t, S_\ell) \leq 1/m$ and dist $(x_t, S_k \cap S_\ell) \geq C$. By the fact that (x_t) is bounded (since S_k and S_ℓ are), there is a subsequence that converges to some x, which is necessarily in both S_k and S_ℓ since these sets are closed. At the same time, dist $(x, S_k \cap S_\ell) \geq C$ by continuity of the distance function, which is a contradiction. In fact, the assumption on the incidence angle implies that $\omega(\varepsilon) \leq C\varepsilon$ for some $C > 0$ not depending on ε—but this is much longer to prove.

Landmarks. A key ingredient to the success of the procedure is that there is at least one landmark chosen from each cluster that is far away from any other cluster. That said, we work with the stronger condition that *all* chosen landmarks are more than ε away from the other clusters, which leads to simplifications latter on in the discussion.

Theorem 1 $p_M^*(\varepsilon + 2\tau) \to 1$, *the probability that all M landmarks are away from the other clusters by at least* ε *converges to 1.*

Proof. The probability of selecting a point from Eq (5) with $s \in S_k$ away from any other cluster by at least ε is $p_k(\varepsilon) := \pi_k(1 - \mu_{S_k}(\bigcup_{\ell \neq k} S_\ell^\varepsilon))$, where S^ε denotes the points in $\mathbb{R}^D$ within distance ε of S. Then the probability that all M landmarks are away from the other clusters by at least ε is equal to

$$p_M^*(\varepsilon + 2\tau) := (\sum_k p_k(\varepsilon + 2\tau))^M \tag{6}$$

where we used the triangle inequality. By dominated convergence, when $\varepsilon, \tau \to 0$, $p_k(\varepsilon + 2\tau) \to$

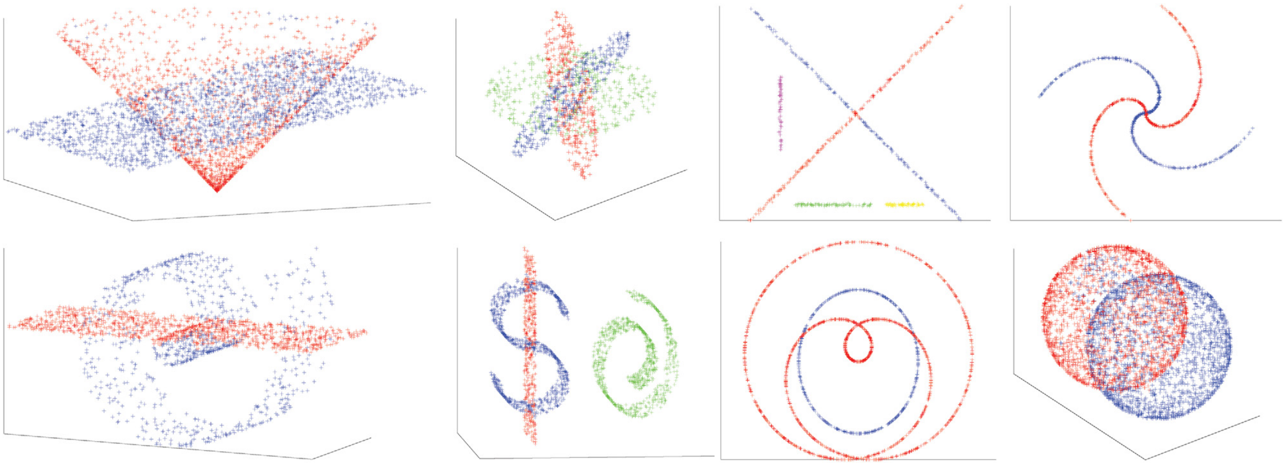

Fig 3. Result of our method on 8 synthetic datasets.

doi:10.1371/journal.pone.0137986.g003

$\pi_k(1 - \mu_{S_k}(\cup_{\ell \neq k} S_\ell)) = \pi_k$, implying $p^*_M(\varepsilon + 2\tau) \longrightarrow 1$, if we assume that $\mu_{S_k}(S_\ell) = 0$ when $\ell \neq k$, which is a very mild assumption we make hereafter.

Numerical Expriments

Synthetic Data

In our experiments we use annulus neighborhood of points to build our graph. The constraint is angle constraint which is easier to tune. The synthetic datasets we generated are similar to those appearing in the literature. We have applied our method in 8 synthetic data including: Three Planes(TP), Two Spirals (TS1), Five Segments(FS), Dollar-Sign and Plane and Roll (DPR), Roll and Plane(RP), Cone and Plane(CP), Two Spheres(TS2), Rose Curve and Circle (RC). Fig 3 shows the performance of our algorithm on eight synthetic data set. The misclustering rates for our method, and the other three methods, are presented in Table 1, where we see that our method achieves a performance at least comparable to the best of the other three methods on each dataset. To compute the accuracy of clustering we remove a few ambiguous points close by intersection. Spectral Curvature Clustering (SCC) works well on linear manifolds (as expected) while it fails when there is curvature (Fig 4a). K-Manifolds fails in the more complicated examples (Fig 4c and 4d). We found that this algorithm is very slow since it has to compute the shortest path between all the points, so that we could not apply it to some of the largest datasets. We mention that it assumes that clusters intersect, and otherwise does not

Table 1. Clustering accuracy on synthetic data.

Data set	K-Manifolds	SCC	SMMC	PBC-Angle-Annulus	PBC-Curvature
TP	97.1%	97.8%	99.5%	99.6%	99.6%
TS1	95.2%	54.8%	99.7%	99.2%	99.1%
FS	59.1%	94.9%	99.6%	98.1%	98.0%
DPR	50.2%	-	99.6%	99.7%	99.5%
RP	56.5%	-	97.6%	96.7%	96.9%
CP	-	-	99.6%	97.9%	98.1%
TS2	-	-	96.7%	98.6%	98.4%
RCC	62.9%	-	64.8%	99.8%	99.7%

doi:10.1371/journal.pone.0137986.t001

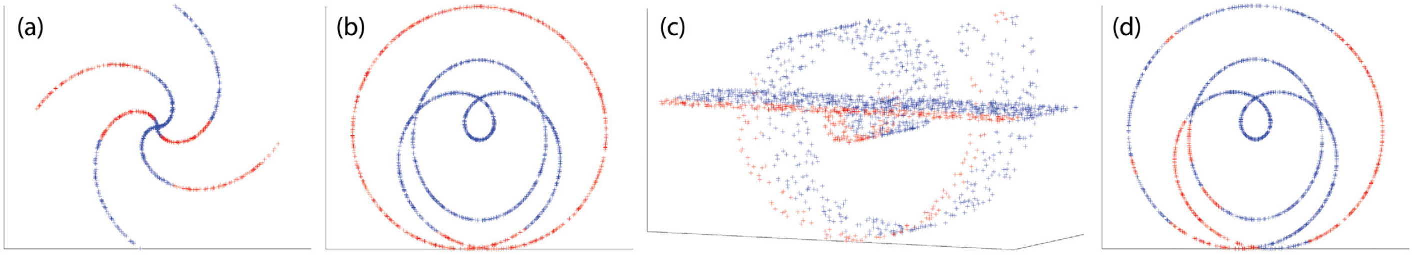

Fig 4. Examples where the other methods fail. (a) SCC, (b) SMMC, (c, d) K-Manifolds.

doi:10.1371/journal.pone.0137986.g004

work properly. Our method and Spectral Multi-Manifolds Clustering (SMMC) perform comparably on most datasets, but SMMC fails in the Rose Curve and Circle example (Fig 4b). Since the path from self-intersecting rose is very smooth at the intersection SMMC cannot capture the correct manifold. The local geometric information of the sampled data can not be used to construct a suitable affinity matrix. Pairwise affinity between two local tangent spaces at the intersection is very similar. For more detail please look at Fig 3 in SMMC paper. Our method can take care of very smooth self-intersecting manifolds or multiple manifolds at the intersection with a choice of an appropriate constraint. We note that K-Manifold, SCC and SMMC all require that all surfaces are of same dimension, which is a parameter of these methods, while our method does not need knowledge of the intrinsic dimensions of the surfaces and can operate even when these are different.

Clustering of 2D Image Data

In this section we apply our method on the COIL-20 dataset (Fig 5) which includes 1440 grayscale images of 20 objects. Each object contains 72 images taken by a camera at different angles. The original resolution of each image is 128 × 128. We first projected the dataset onto the top 10 principal components, then apply our path-based clustering algorithm. We tested our method on the three very similar objects 3, 6 and 19. The algorithm is 99% accurate (misclusters only 2 images out of 216) bringing a significant improvement over the state-of-the-art result of 70% reported in [27]. Lastly, we evaluated our method on the whole dataset obtaining an 83.6% accuracy, improving on the 70.7% accuracy reported in [27]. (Here we used the top 20 principal components.) Since in this case we have 20 different classes, we increased the number of landmarks to 100 to make sure we sampled that at least a few landmarks from each class.

Clustering of Human Motion Sequences

In computer vision clustering of human motion sequences into different class of activities performed by a subject is referred to temporal segmentation. In this section we test our algorithm

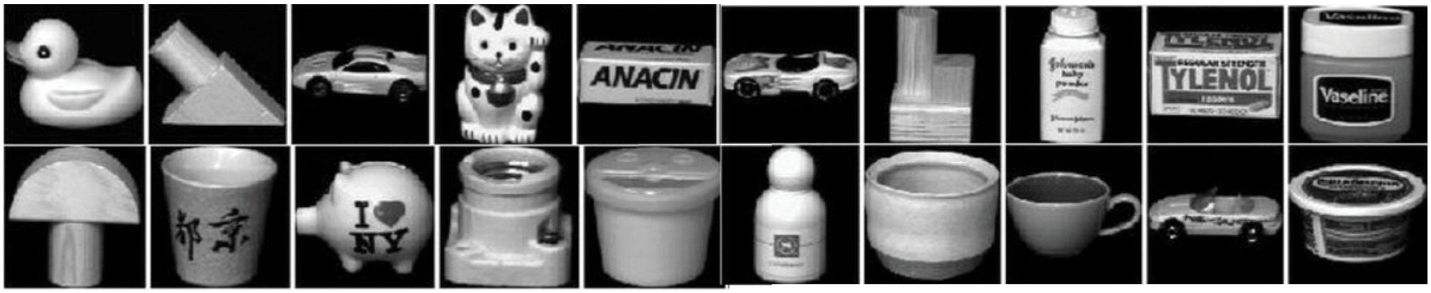

Fig 5. The 20 objects from the COIL-20 database. This figure is similar but not identical to the original image, and is therefore for illustrative purposes only.

doi:10.1371/journal.pone.0137986.g005

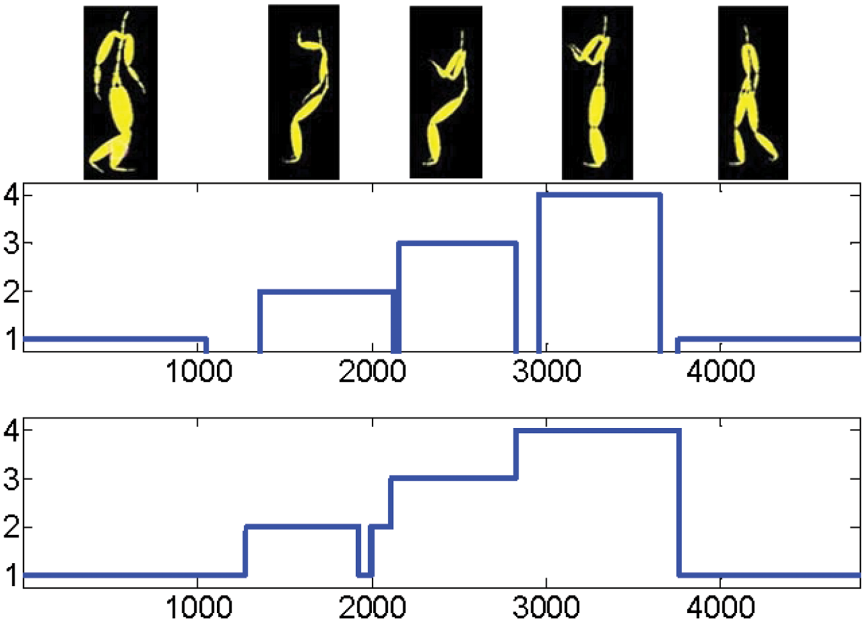

Fig 6. Result of human activity segmentation using Path-Based Clustering. There are 4 activities: walking (1), looking (2), sitting (3) and standing (4). Top: a sample of the sequence. Middle: ground truth. Bottom: output of our algorithm.

doi:10.1371/journal.pone.0137986.g006

on a sequence of video frames including different activities performed by a subject. We choose 4 mixed actions from subject 86, trial number 9 of the CMU MoCap dataset. This is the data set used by [20].? The data consists in a temporal sequence of 62-dimensional representation of the human body via markers in $\mathbb{R}^3$. One motion sequence of 4794 frames and corresponding result of path-based multi-manifold clustering are given in Fig 6. Four activities are labeled from 1 to 4. We do not label the frames where the subject switches from one action to another because of the uncertainty about the true activity. In experimental results we used the same data set as K-Manifold used in their paper. You can refer to their paper for more details. In this data SCC fails since it is suitable for linear surfaces with the intersection, it wont work for nonlinear manifolds.

Segmentation of Video Sequences

In this section we consider the problem of partitioning a video sequence into different scenes. We consider the same video sequence used in [36, 38]. The video is an interview from Fox News containing 135 image frames of size 294 × 413. Firstly we change each RGB image frame to the gray scale intensity image, then resize it to an 74 × 104 image. After concatenating all pixels of each image and putting into a vector of size 7696, we construct a matrix of size 135 × 7696 where each row represents a frame of the original video sequence. Applying our algorithm on this matrix we get a perfect clustering (100%). We repeated the experiment, this time projecting the data onto the top 10 principal components as done in [36, 38], obtaining a matrix of size 135 × 10. We still get a 100% accuracy, for an even wider range of parameters.

Discussion

Robustness to Noise

We note that all the other methods we know of for multi-manifold clustering do not perform well unless the noise level is quite small. As it appears in our method when we increase the

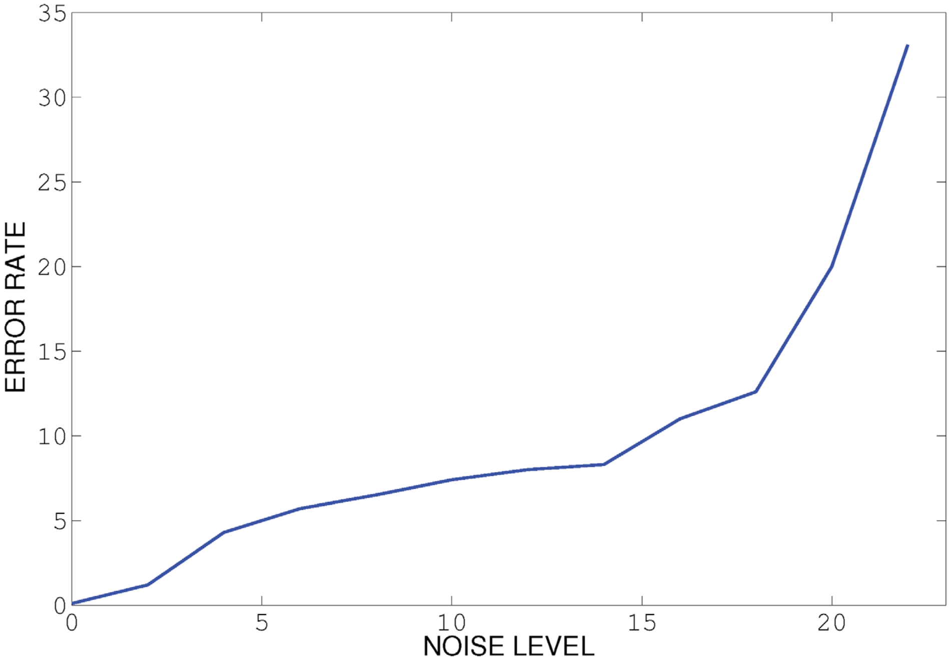

Fig 7. Effect of noise on performance of our algorithm on two intersecting curves shown in Fig 8.

doi:10.1371/journal.pone.0137986.g007

amount of noise the possibility of connecting points from different surface with curvature constrained path increases. Fig 7 shows the error rate for two intersecting curves with addition of standard uniform noise, as it can be illustrated, when we increase the noise, notion of two different surface or manifold would be ambiguous where we can say all the points belong to one manifold. All other three methods fail to capture the correct manifold with even small noise where our method still perform well with 20% noise. See Fig 8 for an example with a substantial amount of noise, where SMMC fails while PBC succeeds.

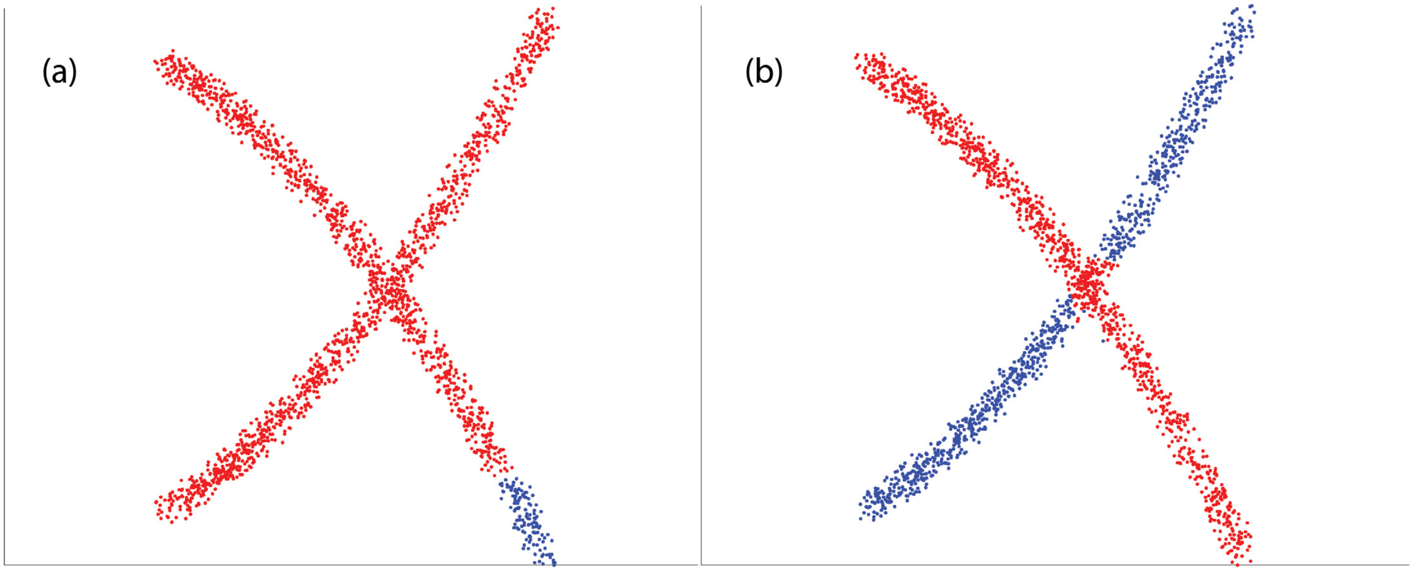

Fig 8. Example of noisy data. (a) output of SMMC. (b) output of PBC.

doi:10.1371/journal.pone.0137986.g008

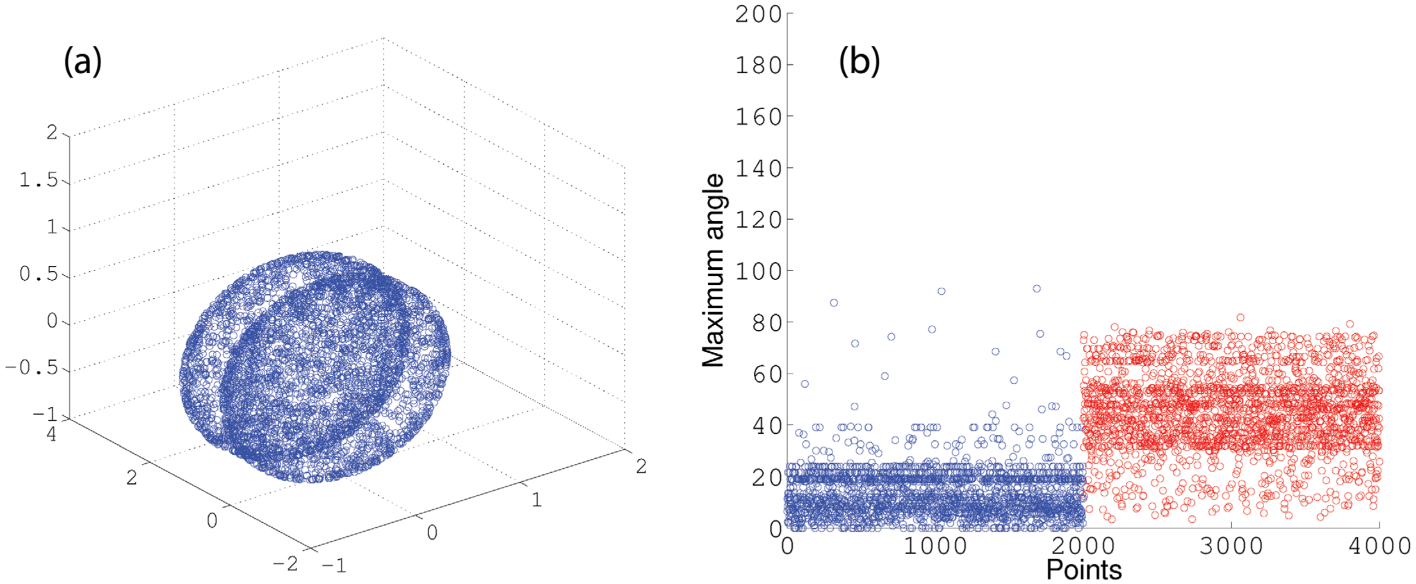

Fig 9. (a) two intersecting Sphere. (b) the distribution of maximum angle in unconstrained shortest-paths between points and a given landmark.

doi:10.1371/journal.pone.0137986.g009

The choice of constraint

One of the main challenges of our algorithm is the choice of the angle constraint with annulus graph, since we deal with multiple manifolds with intersection. The large angle constraint causes the points from different manifolds to be connected using constrained shortest-path. This ultimately leads to multiple manifolds being clustered as one class. We also considered implementing the small constraint, however, this constraint does not allow us to accurately capture the structure of the manifold. Fig 9 shows two intersecting spheres and the distribution of maximum angle in an unconstrained shortest-path between all the points and a given landmark. The distributions of maximum angle for the points within the same sphere as landmark belongs to (blue) is separable from the distributions of maximum angle for the points within the sphere that landmark does not belong to (red). This illustration guides us to the idea that with the small amount of labeled points we are able to find the appropriate angle constraint. In another experiment we started with an angle constraint of 50° and used 1% of the points in each cluster as labeled data. We then compared the performance of our algorithm on the labeled data. In order to find the optimum angle constraint we increased or decreased our angle constraint by a certain factor. We initially begin with dividing our angle constraint by a factor of 2, until the error ceases to decrease. In the case that the error increases, we increase the angle constraint by a factor of 4/3. In most cases we were able to find the optimal angle constraint within 5 iterations. As it can be understood from Fig 9 the distribution of the maximum angle of the points within a class follows a flat distribution. By having a small number of labeled points we are able to capture the distribution of the maximum angle for the rest of the points in that class.

Computational Complexity

The algorithm is quite fast. Building a symmetric q-nearest neighbor graph using cover trees [39] takes order $O(qN \log N)$, where the implicit constant depends exponentially on the intrinsic dimensions of the surfaces and linearly on the ambient dimension D. The angle-constrained pathfinder routine is a simple variant of Dijkstra's algorithm, whose implementation by

Fibonacci heaps runs in $O(qN \log N)$. Hence, calling this routine once for each landmark costs $O(qMN \log N)$. Grouping the feature vectors $O(FMN)$ and then clustering them by complete linkage costs $O(F^2 \log F)$, where F is the (data-dependent) number of distinct feature vectors ξ_i, often of the order of K in our experiments.

Relation to Landmark-Isomap

As [14] argued that Local Linear Embedding [40] could be readily used for clustering non-intersecting manifolds, in a sense, we show that a constrained version of Landmark-Isomap [41] can be used to cluster possibly intersecting manifolds

Conclusion

In this paper we proposed a new method to cluster multiple manifolds with the intersection which works based on shortest constrained path. We applied our method to synthetic and some real datasets and demonstrated that it performs comparably to the best methods such as K-manifold and spectral multi-manifold clustering. We are currently experimenting with variants—some based on other constraints—that would lead to path-based clustering algorithms that perform at least as well in practice as algorithm 1, and are consistent in the large-sample limit.

Acknowledgments

This work was supported by a grant from the National Science Foundation (DMS- 09-15160).

Author Contributions

Conceived and designed the experiments: A. Babaeian MB. Performed the experiments: A. Babaeian. Analyzed the data: A. Babaeian. Contributed reagents/materials/analysis tools: A. Babaeian MB A. Bayestehtashk. Wrote the paper: A. Babaeian MB A. Bayestehtashk.

References

1. Martínez VJ, Saar E. Statistics of the Galaxy Distribution. Boca Raton: CRC press; 2002.
2. Geman D, Jedynak B. An active testing model for tracking roads in satellite images. IEEE Trans Pattern Anal Mach Intell. 1996; 18:1–14. doi: 10.1109/34.476006
3. Babaeian A, Rastegar S, Bandarabadi M, Rezaei M. Mean shift-based object tracking with multiple features. In: System Theory, 2009. SSST 2009. 41st Southeastern Symposium on; 2009. p. 68–72.
4. Babaeian A, Rastegar S, Bandarabadi M, Erza M. Modify kernel tracking using an efficient color model and active contour. In: System Theory, 2009. SSST 2009. 41st Southeastern Symposium on; 2009. p. 59–63.
5. Rastegar S, Babaeian A, Bandarabadi M, Toopchi Y. Airplane detection and tracking using wavelet features and SVM classifier. In: System Theory, 2009. SSST 2009. 41st Southeastern Symposium on; 2009. p. 64–67.
6. Babaeean A, Tashk AB, Bandarabadi M, Rastegar S. Target Tracking Using Wavelet Features and RVM Classifier. In: Natural Computation, 2008. ICNC'08. Fourth International Conference on. vol. 4; 2008. p. 569–572.
7. Babaeean A, Tashk AB, Barzin F, Hosseini SM. Target Tracking Using Mean Shift and Dynamic Directional Gradient Vector Flow. In: System Theory, 2008. SSST 2008. 40th Southeastern Symposium on; 2008. p. 366–370.
8. Ma Y, Yang AY, Derksen H, Fossum R. Estimation of Subspace Arrangements with Applications in Modeling and Segmenting Mixed Data. SIAM Review. 2008; 50(3):413–458. doi: 10.1137/060655523
9. Fu Z, Hu W, Tan T. Similarity based vehicle trajectory clustering and anomaly detection. In: ICIP; 2005. p. II-602–5.
10. Vidal R, Ma Y. A unified algebraic approach to 2-D and 3-D motion segmentation and estimation. JMIV. 2006; 25(3):403–421. doi: 10.1007/s10851-006-8286-z

11. Ho J, Yang M, Lim J, Lee K, Kriegman D. Clustering appearances of objects under varying illumination conditions. In: CVPR; 2003. p. 11–18.

12. Basri R, Jacobs D. Lambertian Reflectance and Linear Subspaces. IEEE PAMI. 2003; 25(2):218–233. doi: 10.1109/TPAMI.2003.1177153

13. Epstein R, Hallinan P, Yuille A. 5 ± 2 Eigenimages Suffice: An Empirical Investigation of Low-dimensional Lighting Models. In: IEEE Workshop on Physics-based Modeling in Computer Vision; 1995. p. 108–116.

14. Polito M, Perona P. Grouping and dimensionality reduction by locally linear embedding. Advances in Neural Information Processing Systems. 2001; 14:1255–1262.

15. Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems. 2002; 2:849–856.

16. Elhamifar E, Vidal R. Sparse Manifold Clustering and Embedding. In: Shawe-Taylor J, Zemel RS, Bartlett P, Pereira FCN, Weinberger KQ, editors. Advances in Neural Information Processing Systems 24; 2011. p. 55–63.

17. Gionis A, Hinneburg A, Papadimitriou S, Tsaparas P. Dimension induced clustering. In: KDD'05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. New York, NY, USA: ACM; 2005. p. 51–60.

18. Haro G, Randall G, Sapiro G. Stratification learning: Detecting mixed density and dimensionality in high dimensional point clouds. Advances in Neural Information Processing Systems. 2007; 19:553.

19. Arias-Castro E, Chen G, Lerman G. Spectral Clustering Based on Local Linear Approximations. Electron J Statist. 2011; 5:1537–1587. doi: 10.1214/11-EJS651

20. Souvenir R, Pless R. Manifold clustering. In: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. vol. 1; 2005. p. 648–653 Vol. 1. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber = 1541315.

21. Bayestehtashk A, Shafran I. Parsimonious multivariate copula model for density estimation. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on; 2013. p. 5750–5754.

22. Bayestehtashk A, Asgari M, Shafran I, McNames J. Fully automated assessment of the severity of Parkinson's disease from speech. Computer Speech & Language. 2013;.

23. Tashk ARB, Sayadiyan A, Mahale P, Nazari M. Pattern Classiffication using SVM with GMM Data Selection Training Methode. In: Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on; 2007. p. 1023–1026.

24. Guo Q, Li H, Chen W, Shen IF, Parkkinen J. Manifold Clustering via Energy Minimization. In: ICMLA'07: Proceedings of the Sixth International Conference on Machine Learning and Applications. Washington, DC, USA: IEEE Computer Society; 2007. p. 375–380.

25. Kushnir D, Galun M, Brandt A. Fast multiscale clustering and manifold identification. Pattern Recogn. 2006; 39(10):1876–1891. doi: 10.1016/j.patcog.2006.04.007

26. Goldberg AB, Zhu X, Singh A, Xu Z, Nowak R. Multi-manifold semi-supervised learning. In: AISTATS; 2009. p. 169–176.

27. Wang Y, Jiang Y, Wu Y, Zhou ZH. Spectral Clustering on Multiple Manifolds. Neural Networks, IEEE Transactions on. 2011; 22(7):1149–1161. doi: 10.1109/TNN.2011.2147798

28. Gong D, Zhao X, Medioni G. Robust Multiple Manifolds Structure Learning. In: Proc. 29th Intl. Conf. on Machine Learning (ICML); 2012..

29. Tenenbaum JB, de Silva V, Langford JC. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science. 2000; 290(5500):2319–2323. doi: 10.1126/science.290.5500.2319

30. Tauro F, Grimaldi S, Porfiri M. Unraveling Flow Patterns through Nonlinear Manifold Learning. PLoS ONE. 2014 03; 9(3):e91131. Available from: http://dx.doi.org/10.1371%2Fjournal.pone.0091131. doi: 10.1371/journal.pone.0091131

31. Shen H, Tao D, Ma D. Dual-Force ISOMAP: A New Relevance Feedback Method for Medical Image Retrieval. PLoS ONE. 2013 12; 8(12):e84096. Available from: http://dx.doi.org/10.1371%2Fjournal.pone.0084096. doi: 10.1371/journal.pone.0084096

32. Hoang Duc AK, Modat M, Leung KK, Cardoso MJ, Barnes J, Kadir T, et al. Using Manifold Learning for Atlas Selection in Multi-Atlas Segmentation. PLoS ONE. 2013 08; 8(8):e70059. Available from: http://dx.doi.org/10.1371%2Fjournal.pone.0070059. doi: 10.1371/journal.pone.0070059

33. Hira ZM, Trigeorgis G, Gillies DF. An Algorithm for Finding Biologically Significant Features in Microarray Data Based on *A Priori* Manifold Learning. PLoS ONE. 2014 03; 9(3):e90562. Available from: http://dx.doi.org/10.1371%2Fjournal.pone.0090562. doi: 10.1371/journal.pone.0090562

34. Yang X, Tan MZ, Qiu A. CSF and Brain Structural Imaging Markers of the Alzheimer's Pathological Cascade. PLoS ONE. 2012 12; 7(12):e47406. Available from: http://dx.doi.org/10.1371%2Fjournal.pone.0047406. doi: 10.1371/journal.pone.0047406

35. Maier M, Hein M, von Luxburg U. Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. Theoretical Computer Science. 2009; 410(19):1749–1764.

36. Chen G, Lerman G. Spectral Curvature Clustering (SCC). IJCV. 2009; 81(3):317–330. doi: 10.1007/s11263-008-0178-9

37. Tipping M, Bishop C. Mixtures of Probabilistic Principal Component Analysers. Neural Computation. 1999; 11(2):443–482. doi: 10.1162/089976699300016728

38. Vidal R, Ma Y, Sastry S. Generalized principal component analysis (GPCA). Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2005; 27(12):1945–1959. doi: 10.1109/TPAMI.2005.244

39. Beygelzimer A, Kakade S, Langford J. Cover trees for nearest neighbor. In: ICML'06: Proceedings of the 23rd international conference on Machine learning. New York, NY, USA: ACM; 2006. p. 97–104.

40. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290 (5500):2323–2326. doi: 10.1126/science.290.5500.2323

41. Silva V, Tenenbaum JB. Global versus local methods in nonlinear dimensionality reduction. Advances in neural information processing systems. 2002; 15:705–712.