



Finding representative landmarks of data on manifolds

Jun Li*, Pengwei Hao

Queen Mary, University of London, London E1 4NS, UK

ARTICLE INFO

Article history:

Received 30 August 2008
Received in revised form 29 December 2008
Accepted 28 January 2009

Keywords:

Manifold learning
Data representation
Dimensionality reduction

ABSTRACT

Data-driven non-parametric models, such as manifold learning algorithms, are promising data analysis tools. However, to fit an off-training-set data point in a learned model, one must first “locate” the point in the training set. This query has a time cost proportional to the problem size, which limits the model's scalability. In this paper, we address the problem of selecting a subset of data points as the landmarks helping locate the novel points on the data manifolds. We propose a new category of landmarks defined with the following property: the way the landmarks represent the data in the ambient Euclidean space should resemble the way they represent the data on the manifold. Given the data points and the subset of landmarks, we provide procedures to test whether the proposed property presents for the choice of landmarks. If the data points are organized with a neighbourhood graph, as it is often conducted in practice, we interpret the proposed property in terms of the graph topology. We also discuss the extent to which the topology is preserved for landmark set passing our test procedure. Another contribution of this work is to develop an optimization based scheme to adjust an existing landmark set, which can improve the reliability for representing the manifold data. Experiments on the synthetic data and the natural data have been done. The results support the proposed properties and algorithms.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Manifold learning refers to a family of algorithms that analyse the variates non-linearly underlying the distribution of points in high-dimensional data space [17]. However, unlike the classical linear methods, e.g., PCA, these methods learn the embedding of the data points, but not the explicit mapping between the low-dimensional manifold coordinates and the high-dimensional data points. Thus they are not readily generalized to unseen points [6], and rerun the algorithm for each new sample is expensive. Based on the established connection between manifold learning and kernel density estimation [30,21,3], we can estimate the low-dimensional coordinates for a new point with the help of the learned embedding [6,27,5]. To apply these estimating schemes, the near neighbours of the input point on the manifold need to be found. In other words, it is to be “located” on the manifold. This entails a cost proportional to the volume of the training data and limits the scalability of the algorithms.

In order to answer a query for a sample quickly, it is natural to use part of the samples as *landmarks* to represent the full set of the training data. The procedure is as follows: Given a test point, the distances from this point to each landmarks are computed, and the

nearest landmark is found. Then using the established data structure, we retrieve the non-landmark points associated with the nearest landmark and proceed the nearest neighbour search in these points. However, without the knowledge on the manifold, only the Euclidean distance from the new point to the landmarks can be computed. Therefore to allocate the right landmark (the one close to the query point on the manifold) to the newly incoming point needs our landmark set has the “proper” relationship with our manifold: for each point on the manifold, if we choose the landmark that has smallest Euclidean distance to that point from our landmark set, the chosen landmark should be close to the query point in terms of geodesic distance on manifold as well. In other words, we want the nearest landmark to a query point in the *ambient space* to be close to that point on the manifold as well, where “ambient space” refers to the Euclidean space, which contains the data points represented in vectorial form. The term “ambient” indicates the fact that the data manifold is embedded in that space. Such that a sample may be safely located to the proper area on the data manifold, by being compared with only those landmarks. The proper area will contain the position at which the sample should have been located, were the query conducted in the full set of the training data. Therefore using landmarks may reduce the cost of generalizing the learned embedding of the manifold.

Let the geodesic distance from a point to its nearest landmark on the manifold be d and the geodesic distance to the nearest landmark

* Corresponding author.

E-mail addresses: junjy@dcs.qmul.ac.uk (J. Li), phao@dcs.qmul.ac.uk (P. Hao).

in the *ambient* space be d' . It is straightforward to see that $d' \geq d$. However, it is worth asking:

Question 1. Is d' significantly larger than d ?

We will see in the following sections, this question has equivalent forms:

- Does any connection between a point and its nearest landmark in the ambient Euclidean space cause a *short-circuit* of the manifold¹ ?
- Given a set of landmarks, their corresponding Voronoi diagram partitions the ambient space. Intersecting with manifold \mathcal{M} , the Voronoi diagram also splits \mathcal{M} . The question is whether all the Voronoi cells intersects the manifold in a local patch?

In this paper, we analyse Question 1 in a formal way, answer it with an easy-to-implement test and develop a hierarchical structure of the data using the landmarks. We further developed a more applicable and robust test and prove its validity in terms of preserving the topology of the neighbourhood graph of the data point cloud. For a given set of landmarks, if our test indicates that there is risk of changing manifold topology (“short-circuit”) by using such landmarks to represent the manifold, besides the obvious solution of adding more points into the landmark set, we have also developed an optimization-based algorithm to adjust the existing landmarks so that the manifold structure is represented more reliably. Moreover, the optimization procedure is less computationally demanding than performing the test at each time of adding a new landmark.

The rest of this paper are organized as follows: In Section 2, we briefly review the related works. The definitions and symbols are introduced in Section 3. We present our analysis of Question 1 and the simple test criterion for the landmarks in Section 4. In Section 5, how to construct the hierarchical structure of the data is introduced. In Section 6, the connection between the landmark criterion and the topology of the neighbourhood graph is elaborated. A more efficient, robust and applicable test procedure is proposed. We discuss how to adjust existing landmark set in the following section. In Section 8, we report experimental results and we conclude this paper in Section 9.

2. Related work

In the past few years, many manifold learning approaches emerged, e.g., Isomap, by Tenenbaum et al. [28], local linear embedding (LLE) by Roweis and Saul [26], Laplacian eigenmaps, by Belkin and Niyogi [4], local tangent space alignment (LTSA), by Zhang and Zha [35], Hessian eigenmaps, by Donoho and Grimes [12], Riemannian manifold learning, by Lin and Zha [25], charting, by Brand [8], and diffusion maps, by Lafon et al. [22]. These methods discover the components that (generally non-linearly) underlie the variation of the data and show the potential to extract information from the high-dimensional raw data [13,17], therefore, they have found their way to many applications [33,9,15].

However, there are still many problems to be addressed before these approaches become practical tools for data analysis [32]. One of the biggest barriers against manifold learning methods to be applied on practical tasks is that they do not give explicit mapping between the low-dimensional latent space and the high-dimensional data space as the traditional tools, e.g., principle component analysis, do. Obviously, recomputing the embedding of a rich data set for a few test samples is computationally demanding.

¹ We will discuss the meaning of “short-circuit” in details in Section 5. Let us agree here on the intuitive meaning of this metaphor: the connection links two parts of the manifold by bridging a gap in the ambient space that does not belong to the manifold. Without that bridge, it is far to travel from one part to the other along the original manifold.

Researchers have proposed methods of generalizing the learned embedding beyond the training samples for those non-parametric methods. Some of them use a linear model to approximate the relationship between the high-dimensional data points and the learned low-dimensional manifold coordinates [15,31]. To generalize the learned manifold embedding non-linearly, researchers have explored how the point cloud represents the data distribution [19,3,21]. Having the estimation of the density of the manifold in the ambient space, methods of generalizing the functions (including the parametrization) on the manifold to off-training-set points have been developed. Zhang et al. [34] extend the embedding generated by LLE for new face images. Bengio et al. [6] have proposed a more general approach to extending the embedding found by Isomap, LLE and other eigenmaps for the new samples. Belkin et al. [5] address the problem of predicting a labelling function on manifolds for discriminative tasks by adopting the manifold settings as the regularization for the objective function. These proposed algorithms evaluate the objective function, which, e.g., in our case the coordinate function in low-dimensional space, on new points with the help of the computed values for the training data. Therefore, to apply these estimations, one needs to find the nearest neighbours for each incoming point in the training data.

Properly set up landmarks can help fast nearest neighbour searching. In our recent work [24], we propose a preliminary method, which can judge if a set of landmarks preserves the topology of a manifold, given the intrinsic dimension of the manifold. However, the intrinsic dimension may not be easy to estimate for data manifolds in practical applications. Using landmarks for manifold learning algorithms has also been proposed for alleviating the computational complexity when the volume of the training data is large [11,10]. However, their landmarks are to improve computational efficiency, but they do not necessarily respect the topology of the manifold. Keeping the topology of a manifold with only a subset of the samples has been discussed in terms of surface reconstruction or mesh simplification in computer graphics [1,16]. These algorithms deal with the special case of 2D manifold in 3D ambient space.

In terms of nearest neighbour searching, our work is also related to the *spatial accessing methods (SAM)* ([29,18], and references therein). For example, KD-tree represents a rich family of space-splitting data structure to hierarchically organize the data for nearest neighbour searching [7]. *Approximate nearest neighbour (ANN)* was proposed for searching high-dimensional data, with trading off the deterministic “nearest-ness” for the computational efficiency [2]. While *locality sensitive hashing* arrange the data in the memory according to their positions in the feature space [14]. However, compared with these general data accessing algorithms, we pay special attention to the manifold structure.

3. Definitions and symbols

Let us first introduce some denotations for the objects and notions used in the following discussion. In our settings, one has observed N data points in \mathbb{R}^D , $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, and wants to find landmarks to represent those observations for nearest neighbour queries. The data are assumed to be drawn from a probabilistic density, which is supported on a d -dimensional manifold \mathcal{M} embedded in \mathbb{R}^D , where $d < D$; and the landmarks will be chosen from \mathbf{X} . Let the landmarks’ indices be $\mathbf{P} = \{p_1, p_2, \dots\}$. Then \mathbf{x}_{p_k} represents the k th landmark. We write it as \mathbf{p}_k when there is no ambiguity, and use \mathbf{P} interchangeably for the set of landmarks and their indices in the data set.

To reveal the unknown manifold, \mathbf{X} is organized as the nodes in a neighbourhood graph \mathbf{G} . To construct \mathbf{G} , we add an edge between each node and its k (number) or ε (Euclidean distance in \mathbb{R}^D) nearest neighbours. \mathbf{G} can be unweighted, where the edge represents the relation of being neighbours of two nodes; it can also be

weighted, where each edge is weighted according to the Euclidean distance between the two nodes. We use “ \sim ” to denote the adjacency relation in the graph. We also call two subsets of nodes as *adjacent* when there are at least one pair of adjacent nodes belonging to each of the subsets, respectively. $d_E(\cdot, \cdot)$ stands for the Euclidean distance in \mathbb{R}^D . $d_M(\cdot, \cdot)$ measures the geodesic distance in \mathcal{M} , where the subscript “ M ” stands for “manifold”. In our discussion, $d_M(\cdot, \cdot)$ is approximated by the length of the shortest path connecting the two nodes in \mathbf{G} , which is a usual practice in literature [28]. Given $\mathbf{x} \in \mathcal{M}$, $L_E^n(\mathbf{x}; \mathbf{P})$ denotes its n th nearest landmark with respect to $d_E(\cdot, \cdot)$, and $L_M^n(\mathbf{x}; \mathbf{P}, \mathbf{G})$ corresponds to the n th nearest landmark measured with $d_M(\cdot, \cdot)$ on \mathcal{M} . In the following, we write L_E^1 as L_E and L_M^1 as L_M , and without writing the landmark set \mathbf{P} and graph \mathbf{G} explicitly if there is no ambiguity. We denote the inverse map for L_E as $\text{Cell}_E(\mathbf{p}) = \{\mathbf{x} | L_E(\mathbf{x}) = \mathbf{p}\}$ and that for L_M as $\text{Cell}_M(\mathbf{p}) = \{\mathbf{x} | L_M(\mathbf{x}) = \mathbf{p}\}$. The meaning of the “cell-map” is as follows: given a landmark, find the data points that take this landmark as the nearest landmark (with some distance measure). In special cases, a point is exactly equidistant to more than one landmarks and these landmarks are the nearest ones. We remark that if a point has more than one nearest landmarks on manifold or in Euclidean space, then all the corresponding cells of these landmarks should contain this point and thus be adjacent to each other.

4. A simple test

In this section, we analyse Question 1 in details. For a certain set of data and landmark points, we present an easy-to-test criterion for answering the question.

4.1. A view of partitioning the manifold

Let us recall that our original motivation for using landmarks is to guide the nearest neighbour query: to find the nearest neighbour for an incoming point, one finds its nearest landmark in the ambient space, and the landmark should indicate the vicinity of the query point on the manifold. Then the nearest point to the query point can be found there.

From another point of view, this procedure can be taken as: each point on the manifold is represented by the nearest landmark in the Euclidean ambient space. Therefore given a landmark set \mathbf{P} , it partitions \mathbb{R}^D into the quotient space of the relation of “having $\mathbf{p} \in \mathbf{P}$ as the nearest landmark”, which is in fact the *Voronoi diagram*² induced by \mathbf{P} . Intersecting with the diagram of \mathbb{R}^D , the embedded manifold \mathcal{M} is also split. We denote this partition of the manifold as $\mathcal{P}_E = \{\text{Cell}_E(\mathbf{p}) | \mathbf{p} \in \mathbf{P}\}$, where E stands for “Euclidean”.

A *geodesic Voronoi diagram* partitions a manifold in a way analogous to a normal Voronoi diagram partitions an Euclidean space. The difference is that the geodesic distance metric is used instead of the Euclidean metric. We denote the partition of the manifold with the geodesic Voronoi diagram as $\mathcal{P}_M = \{\text{Cell}_M(\mathbf{p}) | \mathbf{p} \in \mathbf{P}\}$, where M stands for “manifold”.

Answering Question 1 is equal to verifying whether \mathcal{P}_E is identical to \mathcal{P}_M . This provides a comprehensive overlook of our problem. However, we choose to address Question 1 rather than directly comparing the two partitions because it is not feasible in practice: constructing the Voronoi diagram for the ambient space, which is of very high dimensions, is computationally prohibitively expensive. Then, we examine whether the two partitions coincide with each other at each individual data point.

² Given a space \mathbb{X} , a set of points $\mathbf{S} \in \mathbb{X}$ and a distance metric, the *Voronoi tessellation* is the partition of \mathbb{X} induced by \mathbf{S} . Each point $\mathbf{s} \in \mathbf{S}$ corresponds to an area in the partition referred to as its *cell*. The cell of \mathbf{s} consists of all points in \mathbb{X} such that the distance measured with the given metric from the point to \mathbf{s} is less than or equivalent to the distance to any other point in \mathbf{S} .

4.2. Inconsistent points

“*Inconsistent points*” for \mathbf{P} refer to a set of points $\{\mathbf{x} \in \mathbf{X} | L_E(\mathbf{x}; \mathbf{P}) \neq L_M(\mathbf{x}; \mathbf{P})\}$. It is readily to show that if there are no inconsistent points, the condition for the landmarks in Question 1 is met. Thus an inconsistent point indicates a possible violation of the condition. Careful study displays that there are different kinds of inconsistent points, because the causes that make them inconsistent are different.

Border error points: Consider a case that \mathcal{P}_E and \mathcal{P}_M are similar but not identical. Let $\text{Cell}(\mathbf{p}_x) = \{\mathbf{x}_x, \mathbf{x}_1, \dots, \mathbf{x}_{N_x}\}$ and $\text{Cell}(\mathbf{p}_y) = \{\mathbf{y}_y, \mathbf{y}_1, \dots, \mathbf{y}_{N_y}\}$ be two adjacent cells in \mathcal{P}_E , and \mathbf{p}_x and \mathbf{p}_y be their landmarks, respectively.

For a point $\mathbf{x} \in \text{Cell}(\mathbf{p}_x)$, we have $d_E(\mathbf{x}, \mathbf{p}_y) > d_E(\mathbf{x}, \mathbf{p}_x)$. However, because \mathcal{M} is non-linear, d_M fluctuates from d_E . If \mathbf{x} is close to the border between $\text{Cell}(\mathbf{p}_x)$ and $\text{Cell}(\mathbf{p}_y)$, it is possible that $d_M(\mathbf{x}, \mathbf{p}_y) < d_M(\mathbf{x}, \mathbf{p}_x)$. In this case, $\mathbf{p}_x = L_E(\mathbf{x}) \neq L_M(\mathbf{x})$. It is the fluctuation of the distance metrics that makes \mathbf{x} an *inconsistent point*. Fig. 1³ shows an example of this phenomenon. Fig. 1(a) is a segment of a 1D manifold embedded in \mathbb{R}^2 . Fig. 1(b) shows the fluctuation between the two distance metrics by plotting curve length (vertical) against Euclidean distance, both being measured between samples on the curve and the left landmark (red dot in (c)). Fig. 1(c) shows the *inconsistent points* in this case. We can see that such inconsistent points do not affect seriously the structuring of the data. Moreover, they are inevitable when the number of the samples increases. In our framework, they do not need special treatment.

Topological error points: Another kind of inconsistent points are the *topological error points* (TEPs). We call them “topological error” to indicate that $L_E(\mathbf{x})$ and $L_M(\mathbf{x})$ are not only different, but also faraway from each other on \mathcal{M} , consequently, \mathbf{x} is faraway from $L_E(\mathbf{x})$ on the manifold as well. As $L_E(\mathbf{x})$ is the landmark that will be used to locate \mathbf{x} for any point \mathbf{x} in the testing stage, using the current landmark set to represent the structure of \mathcal{M} will cause significant variance in the nearest neighbour retrieval. In other words, there exists some $\mathbf{p} \in \mathbf{P}$, where $\text{Cell}_E(\mathbf{p})$ contains points that are distant from \mathbf{p} on \mathcal{M} . If an input test sample arises among those points, it will be considered to be near to \mathbf{p} , which is not true. Less rigorously, it can be taken as that some of the manifold’s areas are “stuck” together by cells in \mathcal{P}_E . The cells mistake the gap in the ambient space between those areas, in which the manifold does not present with the manifold itself. This will introduce short-cuts that do not exist on manifold from one point to others and thus for a query point may mix its neighbours with the points that are remote on manifold.

In Fig. 2, we elucidate the topological error. The Voronoi diagram of the landmarks is drawn to show their Euclidean cells. TEPs arise when a cell includes points from more than one different part of the manifold. To see an example, note the landmark A in the figure. Its Euclidean cell intersects the manifold and includes two points (red cross) from the area of E_A , which is far from A on the manifold. But if some test sample comes from E_A , it would be led to A .

To test whether \mathbf{x} is a TEP, one needs to judge that whether \mathbf{x} and $L_E(\mathbf{x})$ are *distant enough* from each other on the *unknown* manifold such that the connection between them makes a short-circuit of that manifold. However, this problem is inherently ill-posed: How to define the criterion for “enough” on an unknown manifold? We thus turn to defining heuristics

Given a test sample \mathbf{x} , we find the number n such that $L_E(\mathbf{x}) = L_M^n(\mathbf{x})$. If the intrinsic dimension of the manifold is d , there should be $n \leq d + 1$. Otherwise, \mathbf{x} is taken as a TEP.

³ We suggest to see this figure and Figs. 3, 4, 5, 12, 13a, 14, 15, and 19 in electronic version with colours.

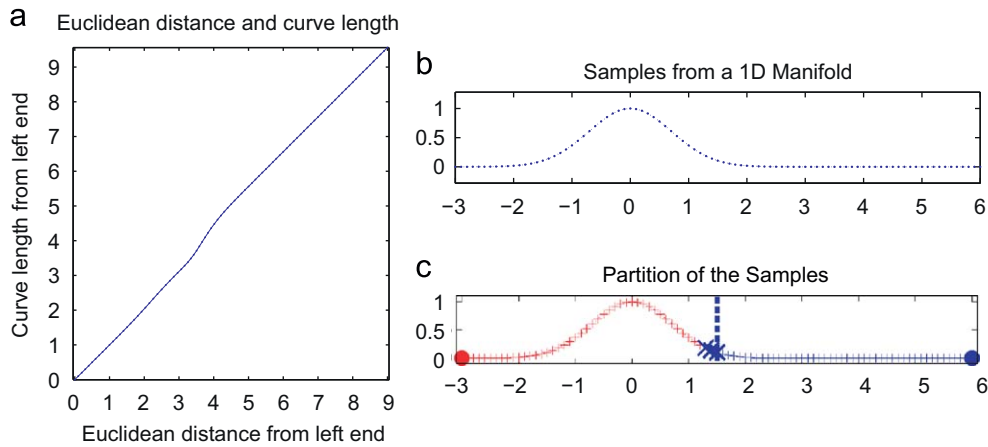


Fig. 1. Border error: (a) samples from a 1D manifold; (b) distance metric fluctuation; (c) partition \mathcal{P}_E (dotted line), \mathcal{P}_M (red and blue colours) and border error (blue crosses) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

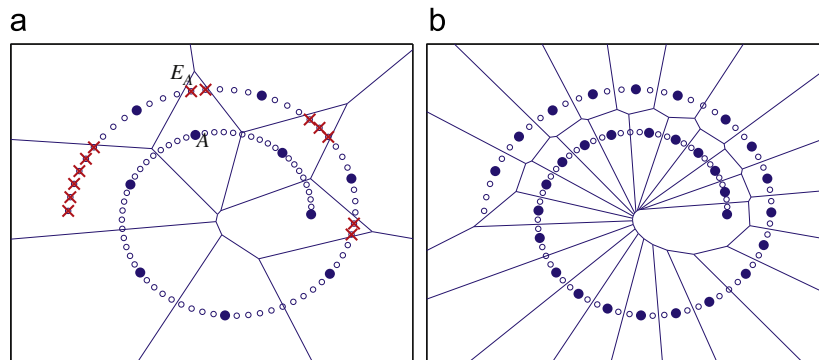


Fig. 2. Partition of manifold with LMPs (big dots): (a) with TEPs (red crosses); (b) TEPs eliminated (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Intuitively speaking, we start from \mathbf{x} , from near to far, search the landmarks on the manifold and examine each of them, until we encounter $L_E(\mathbf{x})$, the nearest landmark to \mathbf{x} in the Euclidean space. Note that here we assume the intrinsic dimensionality d is known.

Our motivation for this heuristics is as follows. For a “border error” inconsistent point \mathbf{x} , it is possible that \mathbf{x} lies in the vicinity of the point, which is equidistant from $L_M^1(\mathbf{x}), \dots, L_M^{d+1}(\mathbf{x})$ as measured along the manifold. Due to the fluctuation between the d_E and d_M , each landmark in $\{L_M^1(\mathbf{x}), \dots, L_M^{d+1}(\mathbf{x})\}$ may be the nearest one in the Euclidean space. Then in this case, we have $n \in \{1, \dots, d + 1\}$. When $n > d + 1$, our judgement is that: because the difference cannot be explained away by distance metric fluctuation, \mathbf{x} is marked as a TEP. Note that in the case where the landmark set is not in general position [20] on the manifold, there may be more than $d + 1$ equidistant nearest landmarks for a sample, thus this rule may identify unnecessary TEPs. However, unnecessary TEPs only incur computational overheads. Thus we err on the safe side.

We show the validity of this test experimentally. Fig. 3 demonstrates the detection of TEPs on 2000 samples of a Swiss roll [28]. There are 50 landmarks randomly selected. In the figure, we display the point-to-landmark distance in 15 out of the 50 cells. The curve in a subplot indicates the geodesic distances from each of the points in the cell to the corresponding landmark in ascending order. A red curve indicates that one or more points in the cell is detected as TEP(s) by the above criterion. Note that, our TEP-detection does not take geometric distance information. The sudden and significant increase of the value of $d_M(\cdot, \mathbf{p}_i)$ coincides with the positive result of

the TEP-detection, which shows that our criterion is effective to detect the empirical risk of breaking manifold topology for a chosen set of landmarks (Fig. 4).

4.3. Elimination of TEPs

To eliminate the TEPs, we add new landmarks, because more landmarks give a finer partition and each part of the partition is less possible to intersect distant areas of the manifold. A simple solution is to choose a new landmark among those TEPs. We first cluster TEPs according to their L_E . The second step is to choose a point in the biggest cluster as the new landmark by picking up the one that is closest to the Euclidean centre of the cluster.

Let $\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \dots, \mathbf{x}_{K_i}^{(i)}$ be detected TEPs in $\text{Cell}(\mathbf{p}_i)$. We choose one of them as the new landmark. This splits the current $\text{Cell}(\mathbf{p}_i)$, and thus makes the partition finer. The computation is as follows.

Firstly, the centre of those points is found:

$$\bar{\mathbf{x}}^{(i)} = \frac{1}{K_i} \sum_{j=1}^{K_i} \mathbf{x}_j^{(i)} \tag{1}$$

Then we find the nearest point to $\bar{\mathbf{x}}^{(i)}$, as the new landmark \mathbf{p}_N :

$$\mathbf{p}_N = \arg \min_{\mathbf{x} \in \text{Cell}(\mathbf{p}_i)} d_E(\mathbf{x}_c^{(i)}, \bar{\mathbf{x}}^{(i)}) \tag{2}$$

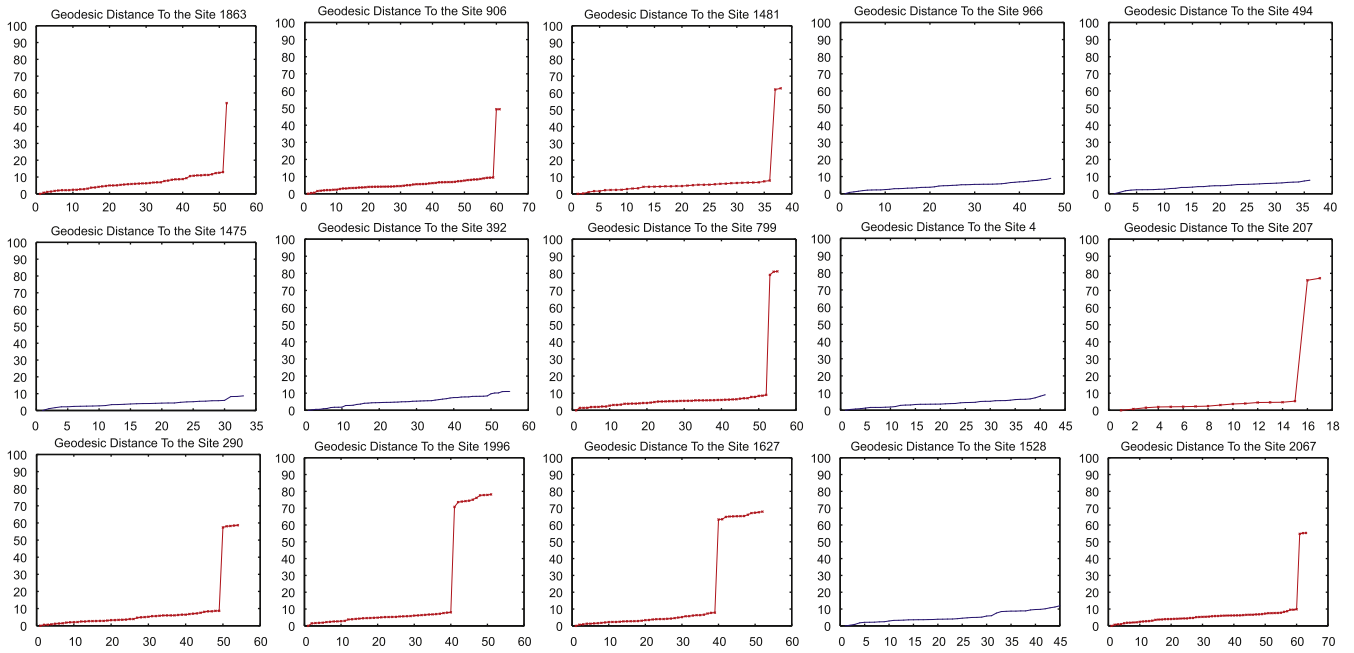


Fig. 3. Geodesic distance from points cells to the site for a partition.

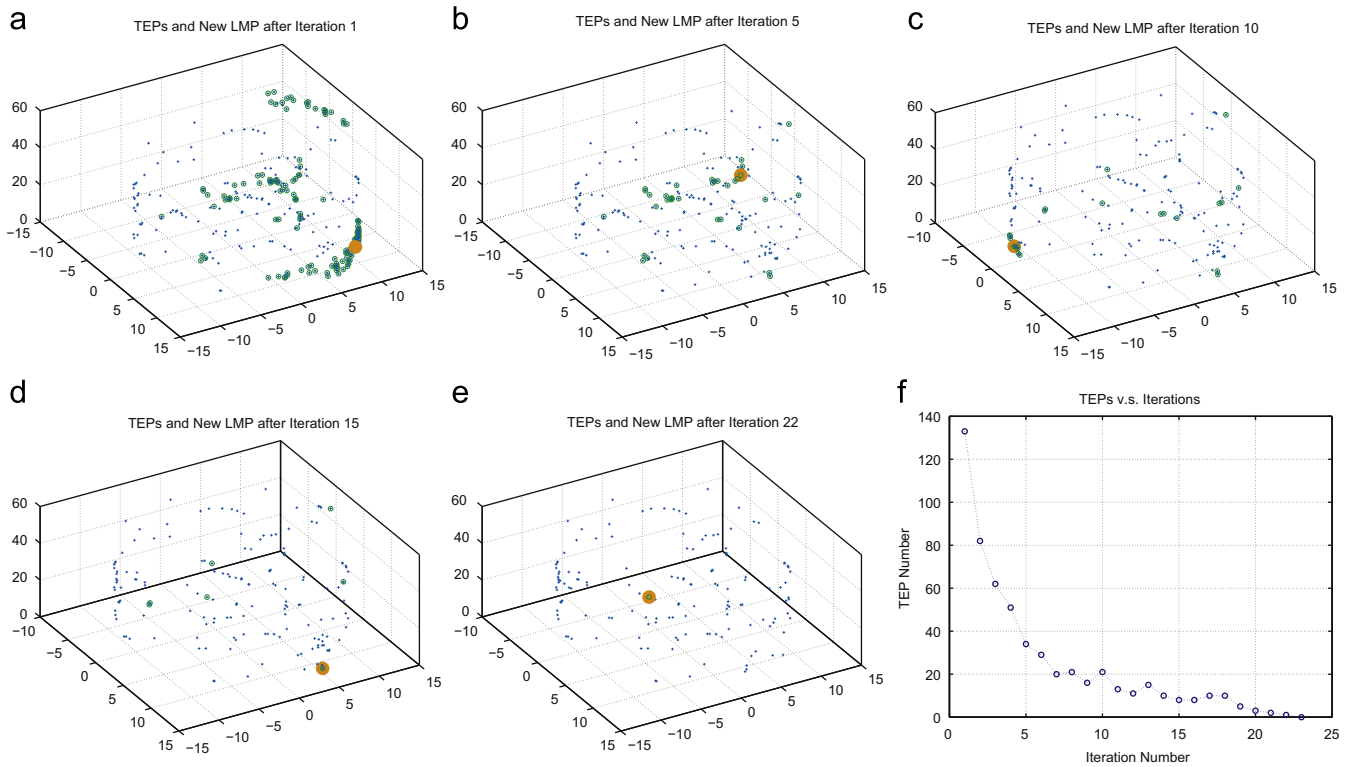


Fig. 4. Adding landmark points (LMPs). TEPs are green dots; new LMP is a bigger orange dot. (a)–(e) TEPs and the new LMPs during the iteration; (f) number of TEPs versus the iterations (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

At last, the landmark set is updated,

$$\mathbf{P}^{(k)} = \{\mathbf{p}_N\} \cup \mathbf{P}^{(k-1)} \quad (3)$$

where k is the iteration number.

The algorithm converges definitely. Because a landmark cannot be a TEP itself, there must be a trivial solution that every point is a

landmark. However, a more helpful landmark set is generated when the algorithm terminated before this trivial end. It is cumbersome to consider all possibilities of the training set of the manifold \mathcal{M} and the starting landmark set $\mathbf{P}^{(0)}$ for the algorithm. We present a loose proof in Appendix A explaining why the algorithm works in practice.

4.4. Implementation and computational complexity

In practice, when a new landmark \mathbf{p}_N is acquired in the k th iteration: $\mathbf{P}^{(k)} = \{\mathbf{p}_N\} \cup \mathbf{P}^{(k-1)}$. It is needed to compute $L_E(\mathbf{x}; \mathbf{P}^{(k)})$ and $L_M^j(\mathbf{x}; \mathbf{P}^{(k)})$, $j = 1, \dots, d+1$ for $\mathbf{x} \in \mathbf{X}$. Rather than starting from nil, they can be updated from $L_E(\mathbf{x}; \mathbf{P}^{(k-1)})$ and $L_M^j(\mathbf{x}; \mathbf{P}^{(k-1)})$ efficiently.

It is straightforward to compute $L_E(\mathbf{x}; \mathbf{P}^{(k)})$ with computational complexity of $\mathcal{O}(N)$, where N is the number of the samples in \mathbf{X} (see Algorithm 4.1).

Algorithm 4.1 (Update the nearest landmark in ambient space).

```

1: for  $\mathbf{x} \in \mathbf{X}$  do
2:   if  $d_E(\mathbf{x}, \mathbf{p}_N) < d_E(\mathbf{x}, L_E(\mathbf{x}; \mathbf{P}^{(k-1)}))$  then
3:      $L_E(\mathbf{x}, \mathbf{P}^{(k)}) \leftarrow \mathbf{p}_N$ 
4:   else
5:      $L_E(\mathbf{x}, \mathbf{P}^{(k)}) \leftarrow L_E(\mathbf{x}; \mathbf{P}^{(k-1)})$ 
6:   end if
7: end for

```

To compute each data point's i th nearest landmark on the manifold according to the adding of \mathbf{p}_N , $i = 1, \dots, d+1$, let us keep a record of the maximum geodesic distance from a data point to its $(d+1)$ th nearest landmark on the manifold to limit the range of samples to be examined:

$$\Delta^{(k-1)} = \max_{\mathbf{x}} d_M(\mathbf{x}, L_M^{(d+1)}(\mathbf{x}; \mathbf{P}^{(k-1)}))$$

For points that are farther away from \mathbf{p}_N than $\Delta^{(k-1)}$, \mathbf{p}_N cannot be one of their nearest $(d+1)$ landmarks, and thus their $L_M^i(\cdot)$, $i = 1, \dots, d+1$ remains unchanged.

We run the Dijkstra algorithm to compute the geodesic distance (shortest path) from the new landmark \mathbf{p}_N to the other points. At each time we obtain the next nearest point to \mathbf{p}_N , adjustment of that point's nearest landmark on the manifold is made if necessary. The procedure is loosely shown in Algorithm 4.2. Computing $d_M(\mathbf{x}, \mathbf{p}_N)$ involves calculating the shortest path from \mathbf{p}_N to a set of nodes \mathbf{x} on the graph. With Fibonacci heap, this takes $\mathcal{O}(KN + N \log N)$ for the set \mathbf{X} , where K is the neighbourhood size that is used in constructing \mathbf{G} and N is the number of nodes in \mathbf{G} . However, our algorithm stops before reaching each node of the graph. In practice, initialization of the heap involves N insertion and thus takes $\mathcal{O}(N)$. On accessing each node, it takes $\mathcal{O}(N)$ to delete the minimum key-valued element in the heap and $\mathcal{O}(K)$ to decrease the key values of the neighbouring nodes. But this repeats only N^- times, where N^- is the number of samples whose shortest path to \mathbf{p}_N is calculated before the algorithm terminates. As K is a small constant, the asymptotic time complexity of the updating is $\mathcal{O}(N^- \log N)$. For uniformly sampled data and landmarks, it can be expected that $\mathbf{P}^{(k-1)}$, N^- is roughly proportional to $N/|\mathbf{P}^{(k-1)}|$.

Algorithm 4.2 (Update the nearest $d+1$ landmark for each point on the manifold).

```

 $d_M^N$  is the buffer storing the distances from each point to the new landmark.
 $\mathbf{Y}$  is the Fibonacci heap storing the points to which the shortest path to be fixed.
1:  $d_M^N(\mathbf{x}) \leftarrow \infty$  for all  $\mathbf{x}$ ;  $d_M^N(\mathbf{p}_N) \leftarrow 0$ ;  $\mathbf{Y} \leftarrow \mathbf{X}$ ;  $\mathbf{y} \leftarrow \mathbf{p}_N$ 
2: while  $d_M^N(\mathbf{y}) < \Delta^{(k-1)}$  do
3:   Find the minimum  $j$ , so that  $d_M(\mathbf{x}, \mathbf{p}_N) < d_M(\mathbf{x}, L_M^j(\mathbf{x}; \mathbf{P}^{(k-1)}))$ .
4:   if  $\exists j \leq d+1$  then
5:      $L_M^j(\mathbf{x}; \mathbf{P}^{(k)}) \leftarrow \mathbf{p}_N$ .

```

```

6:   Adjust  $L_M^{j+1, \dots, d+1}(\mathbf{x}; \mathbf{P}^{(k)})$  accordingly.
7:   end if
8:   if  $\Delta^{(k)} < L_M^{d+1}(\mathbf{x}; \mathbf{P}^{(k)})$  then
9:      $\Delta^{(k)} \leftarrow L_M^{d+1}(\mathbf{x}; \mathbf{P}^{(k)})$ 
10:  end if
11:  Update  $d_M^N$  according to edges connected to  $\mathbf{y}$ 
12:   $\mathbf{y} \leftarrow \arg \min_{\mathbf{y}' \in \mathbf{Y}} d_M^N(\mathbf{y}')$ ;
13: end while

```

The initialization of $L_E(\mathbf{x}; \mathbf{P}^0)$ takes $\mathcal{O}(|\mathbf{P}^0|N)$, and that of $L_M^j(\mathbf{x}; \mathbf{P}^0)$, $j = 1, \dots, d+1$ takes $\mathcal{O}(|\mathbf{P}^0|N \log N)$.

4.5. Hierarchical data structuring and retrieval

With landmarks chosen for a data set, the next problem concerned is how to structure the data with the landmarks and to use the constructed structure for retrieval. We first describe the procedure for retrieval, based on which we design the data structure.

Retrieval: The retrieval procedure is straightforward. Given a test sample \mathbf{x}_t , firstly we find the nearest landmark to \mathbf{x}_t in the Euclidean space, $\mathbf{p}_t = L_E(\mathbf{x}_t)$.⁴ Then we carry on the search in the data points that are “represented” by \mathbf{p}_t .

Structuring: The intuitive way to structure the data with the landmarks is to associate each \mathbf{x}_i with its nearest landmark on manifold $L_M(\mathbf{x}_i)$. The problem with this simple structuring is that: if a test sample \mathbf{x}_t is near the “border area” as described in Section 4.2, it is possible that its neighbourhood is only partially in the subset of data associated with $L_E(\mathbf{x}_t)$. This happens even when the test sample is located exactly to its nearest landmark on manifold, i.e., $L_E(\mathbf{x}_t) = L_M(\mathbf{x}_t)$.

In Fig. 5(a), we show the case where each data point is associated to its nearest landmark only. The yellow shadow areas indicate the “regions” of the two landmarks. When a new sample A comes, it locates the right landmark and in that landmark's region, finds its neighbours successfully. For new sample B, it locates the proper landmark as well as A did. However, as it is close to the border of the landmark's region, its neighbours can only be partially retrieved.

A solution to this problem is to “expand” the subset associated with each landmark, such that, for a sample that is near the border of the cells, it can find its neighbours as well. The expansion should be done according to the size of the neighbourhood that is possibly required (There should be a pre-determined agreement on the size of the neighbourhood to be retrieved.). Fig. 5(b) shows this solution. With the extra associated data, the neighbourhood of B can be found at the cost of slightly increased computational complexity. However, in the construction stage, it is not easy to decide whether a sample should be associated to an landmark such that the set of associated samples of the landmark is properly expanded.

We use a simpler solution in our implementation. Each sample $\mathbf{x} \in \mathbf{X}$ is associated with its nearest $d+1$ landmarks on \mathcal{M} . Fig. 5(c) displays this association scheme: the searching overhead is higher, but the implementation is straightforward.

5. Topological safety revisited

From the analysis in the last section, it can be seen that to answer Question 1 boils down to draw a distinction between the border

⁴ Note that the geodesic distance measurement is not accessible in this case, and $L_M(\mathbf{x}_t)$ is not obtainable. This is our motivation to let the landmark set meet the criterion.

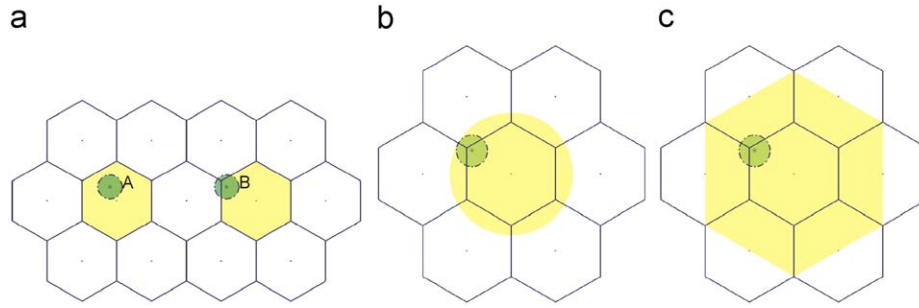


Fig. 5. Association between data points and LMPs. Each data point is associated: (a) to $L_E(\mathbf{x})$ only; (b) ideally; (c) to $L_i^L(\mathbf{x})$, $i \in \{1, \dots, d + 1\}$. The shadow in the figures indicates the areas where the points have association with the central landmark, where the association is established according to the above three schemes.

error points and TEPs. However, in order to make the problem well posed, there are still two important points to be clarified:

- As one may have noticed that, according to the test procedure in the last section, whether a connection between a point \mathbf{x} and $L_E(\mathbf{x})$ is taken as “topological safe” or as a “short-circuit” depends not only on the respective positions of \mathbf{x} and $L_E(\mathbf{x})$ on the manifold, but also on the distribution of the other landmarks. Take an intuitive example, if we have only one landmark, then, however, far from \mathbf{x} to $L_E(\mathbf{x})$, the connection between them causes no “short-circuit”. This makes sense, because in such a circumstance, for any point on the manifold, its nearest landmark, in terms of both Euclidean and geodesic distances, is the only landmark. If we consider the connections in the context of more landmarks presenting, the judgement should change accordingly. Thus it is needed to consider the topology information conveyed by the landmark set as a whole, and how to test the change of the information caused by changing the structuring scheme from assigning the points to the Euclidean nearest landmark to assigning them to the geodesic nearest landmarks.
- There is also a technical issue: the proposed test in the last section requires the intrinsic dimension of the manifold to be known. However, this is not often the case in practice.

In this section, for a given landmark set, we define the topology it applies on the neighbourhood graph. The topology depends also on how we associate each point to some landmark in the set. Thus we analyse Question 1 in terms of the difference of the topology, from the geodesic and Euclidean assignment, respectively. Based on the analysis, we propose a new test scheme, which exploits the topological properties of the neighbourhood graph, and therefore does not require the intrinsic dimension of the manifold.

We firstly elaborate the seemingly intuitive concepts of “topology change” or “short-circuit” in formal language, so that it is well posed. Let us illustrate by an example the argument that whether a link between two points (\mathbf{x} and $L_E(\mathbf{x})$) on a manifold causes short-circuit depends on the context: the manifold geometry, the point cloud sampling, the neighbourhood graph and the distribution of the other landmarks: Fig. 6(a) shows some points from a U-shaped manifold, as well as the links between the neighbouring points on the manifold (dotted lines) and several links (numbered solid lines) to be judged (whether they change the topology). Link 1 should not be considered as a “short-circuit”, otherwise the sampling is inappropriate for the task anyway. From links 2 to 9, the pairs of linked points are farther and farther away on the U-shaped manifold. Which of them should be taken as a “short-circuit”? This question should be answered with care. Even for the link 9, which collapses the whole U-shaped structure and seems to be a definite “short-circuit”, if one

considers in a larger scale, e.g., the case in (b), the link might become acceptable.

5.1. Topology-preserving condition

Due to the consideration above, we do not judge independently if each individual connection ($\mathbf{x}, L_E(\mathbf{x})$) is appropriate. Instead, we compare the Euclidean partition \mathcal{P}_E and the ideal partition on the manifold \mathcal{P}_M to check whether the landmarks represent the manifold in the way that they are “supposed to”, and leave the decision which structure should be preserved to the higher level of processing. The level of details to be preserved may be specified implicitly by providing the initial landmark set.

Generally, with a limited number of landmarks, it is not possible to make the two partitions of the manifold exactly the same. Therefore, we propose

Condition 1. $\forall \mathbf{p}_1, \mathbf{p}_2 \in \mathbf{P}, \text{Cell}_E(\mathbf{p}_1) \cap \text{Cell}_M(\mathbf{p}_2) \neq \emptyset$ implies $\mathbf{p}_1 = \mathbf{p}_2$ or $\text{Cell}_M(\mathbf{p}_1) \sim \text{Cell}_M(\mathbf{p}_2)$.

The condition means that there are two cases that a point \mathbf{x} is “topologically safe”, i.e., not a TEP: (i) $L_E(\mathbf{x}) = L_M(\mathbf{x})$; and (ii) for an *inconsistent point* \mathbf{x} , recall the definition in Section 4.2, we have $\mathbf{p}_1 = L_E(\mathbf{x})$, $\mathbf{p}_2 = L_M(\mathbf{x})$ and \mathbf{p}_1 and \mathbf{p}_2 are distinct. Then their geodesic Voronoi cells $\text{Cell}_M(\mathbf{p}_1)$ and $\text{Cell}_M(\mathbf{p}_2)$ must be neighbours on the manifold.

In the following, we show how this condition implements the concept that “the landmarks represent the data properly”, by first showing an example of quantization and then presenting a theoretical treatment.

Quantization error bound: We take quantization as an example scenario in which there is an intrinsic metric determining the cost of a solution. However, the metric is not represented in an explicit form; thus the actual solution for general inputs has to be designed based on the Euclidean metric. Therefore the pivot point here is to “design the problem”, such that the optimal solution based on the intrinsic metric (indicated by the training samples) is approximated by the optimal solution based on the Euclidean metric. In the context of quantization, this means choosing a proper set of landmarks for the data.

Let \mathbb{E}^A and \mathbb{E}^W measure the average and worst quantization costs, respectively. The quantization cost can be reasonably taken as the geodesic distance between a data point and the quantizer (landmark) to which it is represented, because the geodesic distance of the underlying manifold usually reflects the essential difference between data points. Then we will have

$$\mathbb{E}^A(\mathbf{Q}) = \int_{\mathcal{M}} d_M(\mathbf{x}, \mathbf{Q}(\mathbf{x})) d\mu(\mathbf{x}) \quad (4)$$

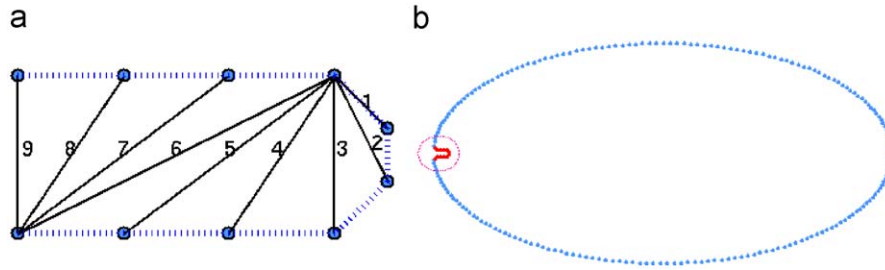


Fig. 6. Which are topology-changing links? (a) Samples from a U-shaped manifold. The dotted links show the true neighbouring relationships. Numbered solid links are to be judged whether they are short-circuits; (b) the global manifold from which (a) is sampled. This is to show that the judgement whether a link changes the topology of a manifold depends on the view points (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

where μ is the probability density function supported on \mathcal{M} , and \mathbf{Q} is the quantization scheme. In practice it is computed as

$$\hat{\mathbb{E}}^A(\mathbf{Q}) = \frac{1}{|\mathbf{X}|} \sum_{\mathbf{x}} d_M(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \quad (5)$$

Hereafter, we do not distinguish the symbols for the ideally continuous and the practically discrete cases. \mathbb{E}^W is the worst-case cost function:

$$\mathbb{E}^W(\mathbf{Q}) = \sup_{\mathbf{x}} d_M(\mathbf{x}, \mathbf{Q}(\mathbf{x})) \quad (6)$$

It is not difficult to see that L_M is the ideal quantization scheme minimizing both \mathbb{E}^A and \mathbb{E}^W : if in some scheme, one point is not quantized by its nearest landmark on manifold, adjusting the scheme to be so will reduce both \mathbb{E}^A and \mathbb{E}^W . However, as we pointed out in the last section, in practice, it cannot be for quantizing new input data points; and we can only use L_E . If Condition 1 holds, we will have

$$\mathbb{E}^W(L_E) \leq \mathbb{E}^W(L_M) + d_{\text{MAX}} \quad (7)$$

$$\mathbb{E}^A(L_E) \leq \mathbb{E}^A(L_M) + d_{\text{MAX}} \quad (8)$$

where d_{MAX} is the greatest geodesic length between adjacent landmarks

$$d_{\text{MAX}} = \sup_{\text{Cell}(\mathbf{p}_1) \sim \text{Cell}(\mathbf{p}_2)} d_M(\mathbf{p}_1, \mathbf{p}_2) \quad (9)$$

Because $d_{\text{MAX}} \leq 2\mathbb{E}^W(L_M)$, for the worst-case cost, we also have

$$\mathbb{E}^W(L_E) \leq 3\mathbb{E}^W(L_M) \quad (10)$$

Similarity of graph topology: For a neighbourhood graph, Condition 1 also ensures that L_E is *topologically similar* to L_M . Let us first make clear the meaning of “topologically similar”.

Given a partition of the neighbourhood graph $L: \mathbf{X} \rightarrow \mathbf{P}$, a *minor* \mathbf{H} of \mathbf{G} can be defined by the following contraction:

1. $\mathbf{G}^{(0)} \leftarrow \mathbf{G}$.
2. Find an edge $\mathbf{x} \sim \mathbf{y}$ in $\mathbf{G}^{(k)}$
 - if $L(\mathbf{x}) = L(\mathbf{y})$ and $\mathbf{x}, \mathbf{y} \notin \mathbf{P}$, contract \mathbf{x} and \mathbf{y} to \mathbf{x} , and take the resultant graph as $\mathbf{G}^{(k+1)}$.
 - if $\mathbf{y} \notin \mathbf{P}$ and $L(\mathbf{y}) = \mathbf{x}$, or vice versa, contract them to the landmark.
3. Repeat Step 2 until no more edges are contractable.

The contraction results in a minor \mathbf{H} of \mathbf{G} consisting of all the landmarks. Let $\mathbf{H}_M = \text{Contract}(\mathbf{G}; L_M)$ and $\mathbf{H}_E = \text{Contract}(\mathbf{G}; L_E)$. Then it can be proved that

Proposition 1. Condition 1 implies that \mathbf{H}_E is similar to \mathbf{H}_M , where “similar” means:

1. If there is an edge between two landmarks \mathbf{p}_i and \mathbf{p}_j in \mathbf{H}_E , \mathbf{p}_i and \mathbf{p}_j are connected in \mathbf{H}_M by a path of the length less than or equal to 3 (Fig. 7(a)–(c)).
2. If there is an edge between two landmarks \mathbf{p}_i and \mathbf{p}_j in \mathbf{H}_M , there is a path in \mathbf{H}_E between \mathbf{p}_i and \mathbf{p}_j , and the path contains only landmarks in the neighbourhood of \mathbf{p}_i and \mathbf{p}_j in \mathbf{H}_M (Fig. 7(d)).

Of the two cases, case 1 is more important. It means that for a point \mathbf{x} , compared to the optimal representation by $L_M(\mathbf{x})$, the Euclidean representation $L_E(\mathbf{x})$ does not take significantly more risk of making a “short-circuit” on the manifold. The proof is provided in Appendix B.

5.2. Implementation of the test

Testing Condition 1 for a given data set and a landmark set \mathbf{P} involves (i) finding the nearest landmark for each non-landmark point on the manifold, (ii) determining whether each pair of landmarks have adjacent cells and (iii) finding the nearest landmark for each non-landmark point in the Euclidean space. Step (i) needs to find the shortest path from landmarks in \mathbf{P} to the other nodes on \mathbf{G} , which has a time complexity of $\mathcal{O}(|\mathbf{P}|N \log N)$ (see also Section 4). Having the first step done, Step (ii) traverse each edge of \mathbf{G} to mark each pair of landmarks and has a time complexity of $\mathcal{O}(kN)$, where k is the neighbourhood size for constructing \mathbf{G} . The complexity of Step (iii) is $\mathcal{O}(|\mathbf{P}|N)$. Therefore the total time complexity is $\mathcal{O}(|\mathbf{P}|N \log N)$, when k is a small number.

For building the landmark set that indexes the data without a topological error risk, the procedure of eliminating the TEPs is analogous to that in Section 4. We start with an initial landmark set, then select any one of the data points at which Condition 1 does not hold, and add it to the current landmark set. By applying this procedure repeatedly, we will eventually have a set of landmarks (generally, non-trivial) satisfying Condition 1 (or the number of TEPs reduces below a threshold).

As in Section 4, when a new landmark is added, Condition 1 does not need to be re-tested at each point. Instead, for each \mathbf{x} , we keep the records of

- its nearest landmark in the Euclidean space, $L_E(\mathbf{x})$;
- the Euclidean distance to $L_E(\mathbf{x})$, $\rho_E(\mathbf{x}) = d_E(L_E(\mathbf{x}), \mathbf{x})$;
- its nearest landmark on the manifold, $L_M(\mathbf{x})$;
- the shortest path length to $L_M(\mathbf{x})$, $\rho_M(\mathbf{x}) = d_M(L_M(\mathbf{x}), \mathbf{x})$.

We also record the adjacency matrix \mathbf{H}_M to track whether two landmarks’ cells are adjacent on the manifold. Then our test can be

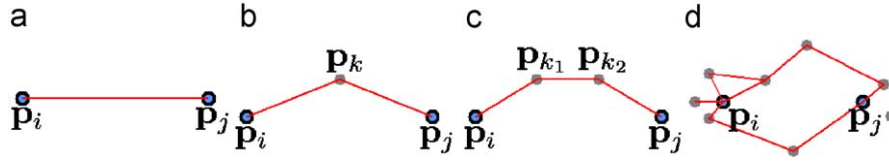


Fig. 7. Graph similarity.

stated as

$$L_E(\mathbf{x}) = L_M(\mathbf{x}) \quad \text{or} \quad \mathbf{H}_M(L_E(\mathbf{x}), L_M(\mathbf{x})) = 1 \quad (11)$$

After a new landmark \mathbf{p}_N is added, it is straightforward to update $L_E(\mathbf{x})$ and $\rho_E(\mathbf{x})$ with a complexity of $\mathcal{O}(N)$.

To adjust $L_M(\mathbf{x})$ and $\rho_M(\mathbf{x})$ according to the newly added landmark \mathbf{p}_N , we use Dijkstra's algorithm to compute the shortest path length from \mathbf{p}_N to each \mathbf{x} . However, as the updating algorithm in Section 4, we do not need to finish the whole procedure of Dijkstra's algorithm and access each individual node of \mathbf{G} . The procedure is listed in Algorithm 5.1. It can be interpreted as: (i) start from the new landmark \mathbf{p}_N . (ii) Compute the shortest paths from \mathbf{p}_N to the other points using Dijkstra's algorithm. (iii) Having computed the shortest path to one point \mathbf{x} , add \mathbf{x} to the cell of \mathbf{p}_N by letting $L_M(\mathbf{x}) \leftarrow \rho_M(\mathbf{x})$. (iv) As all points that are nearer to \mathbf{p}_N than \mathbf{x} have already been in the cell when visiting \mathbf{x} , if \mathbf{x} should not be included in the cell of \mathbf{p}_N , neither should any farther points and thus we stop the updating. As we have discussed in Section 4, when manipulating \mathbf{Y} using a Fibonacci heap, the complexity is $\mathcal{O}(N^- \log N)$, where N^- is the number of the samples that are visited by the algorithm. Roughly speaking, it is the number of points in one cell and thus (averagely) smaller than the N^- in Section 4, which is the number of point in $d + 1$ cells, where d is the intrinsic dimensionality.

Algorithm 5.1 (Update L_M and ρ_M).

d_M^N is the buffer storing the distances from each point to the new landmark.

\mathbf{Y} is the Fibonacci heap storing the points to which the shortest path to be fixed.

- 1: $d_M^N(\mathbf{x}) \leftarrow \infty$ for all \mathbf{x} ; $d_M^N(\mathbf{p}_N) \leftarrow 0$; $\mathbf{Y} \leftarrow \mathbf{X}$; $\mathbf{y} \leftarrow \mathbf{p}_N$
- 2: **while** $d_M^N(\mathbf{y}) < \rho_M(\mathbf{y})$ **do**
- 3: $L_M(\mathbf{y}) \leftarrow \rho_M(\mathbf{y})$; $\rho_M(\mathbf{y}) \leftarrow d_M^N(\mathbf{y})$
- 4: Update d_M^N according to edges connected to \mathbf{y}
- 5: $\mathbf{y} \leftarrow \arg \min_{\mathbf{y}' \in \mathbf{Y}} d_M^N(\mathbf{y}')$;
- 6: **end while**

Note that the stop condition of the while-loop in the algorithm takes advantage of the fact:

If a point $\mathbf{x} \notin \text{Cell}_M(\mathbf{p})$, then $\forall \mathbf{y} \in \text{Cell}_M(\mathbf{p})$, \mathbf{x} is not on the shortest path connecting \mathbf{y} and \mathbf{p} .

Therefore, in the algorithm, when we are sure that a point does not belong to $\text{Cell}_M(\mathbf{p}_N)$, we can safely delete it from \mathbf{Y} and thus avoid searching any path passing that point.

Here we give a sketch of the proof: If this does not hold, consider that one starts from \mathbf{y} and travels along the shortest path to \mathbf{p} arriving at \mathbf{x} . It is not difficult to see that $d_M(\mathbf{y}, L_M(\mathbf{x})) < d_M(\mathbf{y}, \mathbf{p})$, which contradicts $\mathbf{y} \in \text{Cell}_M(\mathbf{p})$.

6. Landmark optimization

In the last two sections, we have discussed the criterion for the landmarks, how to test it and make it satisfied by simply adding new

landmarks. One natural question may arise is that whether the extent to which the proposed criterion is satisfied (or violated) by a set of the landmarks can be estimated quantitatively without performing the test procedure. If we can explicitly estimate the (dis-)satisfaction of condition in terms of the set of landmarks, we will be able to adjust or improve the existing landmark set by optimization so that it represents the manifold structure more reliably. Furthermore, the size of the landmark set can be controlled. In this section, we present the desired estimation and a way of adjusting existing landmarks.

Ideally, we have a function $\mathcal{E} : \mathbf{P} \rightarrow \mathbb{N}$: given the landmarks, \mathcal{E} returns the number of TEPs. Obviously, directly minimizing the \mathcal{E} over \mathbf{P} will lead to a difficult combinatorial programming problem. We can use $\sum d_M^2(\mathbf{x}, L_E(\mathbf{x}))$ as heuristics of the number of TEPs. This heuristic is actually the geodesic distances between the landmarks and the points in the corresponding Euclidean Voronoi cell. We argue without proof that this objective function should favour a landmark distribution that makes the radius of each $\text{Cell}_E(\mathbf{p})$ minimized and thus reduces the TEPs.

However, pre-computing $\forall \mathbf{x}, \mathbf{y}$, $d_M(\mathbf{x}, \mathbf{y})$ requires $\mathcal{O}(N^2 \log N)$ time and $\mathcal{O}(N^2)$ storage, both of which are expensive when the training data are rich. Thus we turn to the estimate of d_M in explicit forms, which allow us to evaluate $d_M(\mathbf{x}, \mathbf{p})$ for $\forall \mathbf{x} \in \text{Cell}_E(\mathbf{p})$ when \mathbf{p} is changing.

6.1. Objective function and optimization

For example, we can use the eigenvector f of the second smallest eigenvalue of the graph Laplacian of \mathbf{G} as a *location indicator* [4]. For a graph with N nodes, f is of N -dimensional, with each entry corresponding to a node. We use $(f(\mathbf{x}) - f(\mathbf{y}))^2$ to estimate $d_M^2(\mathbf{x}, \mathbf{y})$ in the objective function. Therefore, we can write our objective function as

$$\mathcal{E}(\mathbf{P}; \mathbf{G}) = \sum_{\mathbf{p}} \sum_{\mathbf{x} \in \text{Cell}_E(\mathbf{p})} (f(\mathbf{x}) - f(\mathbf{p}))^2 \quad (12)$$

However, searching for a \mathbf{P} that optimizes Eq. (12) still involves complex combinatorial programming. To make gradient-based optimization possible, we apply a “soft” border cells of different landmarks, inspired by Lazebnik and Ragsinsky [23]: a point \mathbf{x} is linked to a landmark \mathbf{p}_k by $w_k(\mathbf{x})$:

$$w_k(\mathbf{x}) = \frac{\exp(-\beta \|\mathbf{x} - \mathbf{p}_k\|_2^2 / 2)}{\sum_j \exp(-\beta \|\mathbf{x} - \mathbf{p}_j\|_2^2 / 2)} \quad (13)$$

where β is the “hardness” parameter. Therefore we rewrite our objective function as

$$\mathcal{E}(\mathbf{P}; \mathbf{G}) = \sum_{i=1}^N \sum_{k=1}^K w_k(\mathbf{x}_i) (f(\mathbf{x}_i) - f(\mathbf{p}_k))^2 \quad (14)$$

We can optimize Eq. (14) w.r.t. \mathbf{p}_k by taking the gradient

$$\frac{\partial \mathcal{E}}{\partial \mathbf{p}_k} = \sum_{i=1}^N \sum_{j=1}^K \left[(f(\mathbf{x}_i) - f(\mathbf{p}_j))^2 \frac{\partial w_j(\mathbf{x}_i)}{\partial \mathbf{p}_k} - 2w_j(\mathbf{x}_i) (f(\mathbf{x}_i) - f(\mathbf{p}_j)) \frac{\partial f(\mathbf{p}_j)}{\partial \mathbf{p}_k} \right] \quad (15)$$

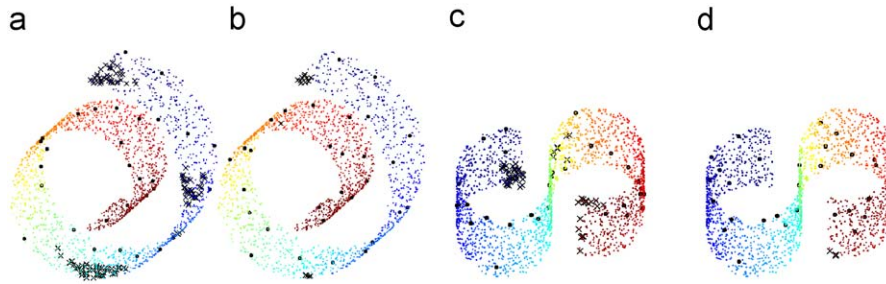


Fig. 8. “Colourizing” function and optimization: (a) and (c) initial landmarks; (b) and (d) after five iterations. “x”: where Condition 1 does NOT hold.



Fig. 9. Manifold samples, partitions (colour), and LMPs (big dot).

where [23]

$$\frac{\partial w_j(\mathbf{x})}{\partial \mathbf{p}_k} = \beta w_k(\mathbf{x})(\delta_{jk} - w_j(\mathbf{x}))(\mathbf{x} - \mathbf{p}_k) \quad (16)$$

and

$$\frac{\partial f(\mathbf{p}_j)}{\partial \mathbf{p}_k} = \delta_{jk} \nabla f|_{\mathbf{p}_k} \quad (17)$$

δ_{jk} is 1 for $j = k$ and 0 otherwise, and ∇f can be pre-computed numerically from the training data. Note that for calculating Eq. (15), the computation only needs to be done for $w_j(x_k) \neq 0$. If the border parameter β is “hard”, for most j , $w_j(\cdot)$ only has one non-zero element.

In the t th step, the potential update for the k th landmark $\mathbf{p}_k^{(t+1)*}$ is found by searching the neighbourhood of $\mathbf{p}_k^{(t)}$ in \mathbf{G} and finding the neighbour to which the vector from $\mathbf{p}_k^{(t)}$ best matches the direction of the gradient in Eq. (15).

$$\mathbf{p}_k^{(t+1)*} = \arg \max_{\mathbf{y} \sim \mathbf{p}_k^{(t)}} \cos \left(\left\langle \mathbf{y} - \mathbf{p}_k^{(t)}, -\frac{\partial \mathcal{E}^{(t)}}{\partial \mathbf{p}_k^{(t)}} \right\rangle \right) \quad (18)$$

A learning rate can be set to decide whether $\mathbf{p}_k^{(t+1)}$ remains the same as $\mathbf{p}_k^{(t)}$ or is updated to $\mathbf{p}_k^{(t+1)*}$. In our implementation, we always update it.

In Fig. 8, we demonstrate the optimization on two synthetic data sets [28,26]. (a) and (b) show the optimization on a Swiss roll, and (c) and (d) show that on an S-shaped manifold. Each data set contains 2000 data points. The neighbourhood size for \mathbf{G} is 8. (a) and (c) show the initial landmarks and (b) and (d) show the resulting landmarks after five iterations of optimization. Condition 1 has been tested for each case. The points that fail the test are also shown.

Link to manifold learning: One may notice that the f happens to be the results of the non-linear dimensionality reduction of the manifold into 1D coordinate space using Laplacian eigenmaps [4]. This

observation reveals that we are essentially looking for a landmark set that results similar *Euclidean* Voronoi tessellations in both the ambient space and the low-dimensional⁵ global manifold coordinate space. Therefore, other manifold coordinate functions (may be vector-valued) may be used for estimating d_M as well. Note that our landmarks are for generalizing learned models, thus requiring the low-dimensional embedding of the data points to be known does not result overheads in practice.

7. Experiments

We conduct experiments on both synthetic and real-world data to verify the proposed criteria, their test procedure and the optimization algorithms.

We first show the results of the simple topological error detection algorithm in Section 4. The experiment is conducted on three data sets of 3D point clouds and one set of synthetic face images. The synthetic sets are used because their intrinsic dimensionality are readily to obtain. Each of the used 3D points contains 2000 samples. They are displayed with the found landmarks in Fig. 9. Fig. 10⁶ shows some of the samples organized in the way that a landmark is displayed with its associated samples. The association is generated by the proposed algorithm. It is shown in the figure that in each of the data set, the cells of the resultant landmarks are fine enough that they do not include points from the different branches of the manifold.

⁵ In this case, it is 1D.

⁶ The face images are from “<http://isomap.stanford.edu/>”. For all the image data set used in our experiment, we firstly represent each image as a feature vector, then constructing the neighbourhood graph \mathbf{G} as described in Section 3. The original high-dimensional feature space is obtained with principle component analysis. The dimension is chosen such that most (95–98%) of the variance in the image set is preserved.

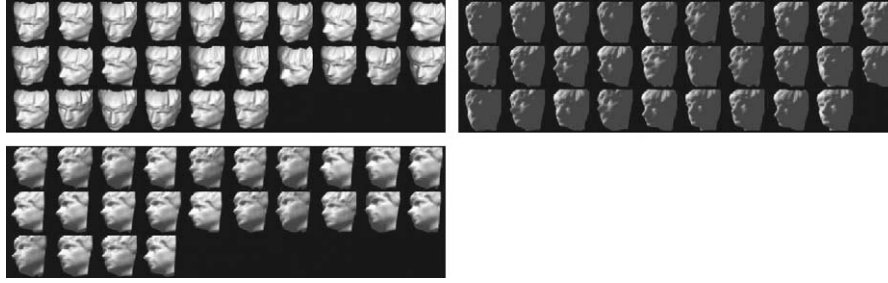


Fig. 10. Face model images. The first image in each group is the LMP.

Table 1
Iteration times on data sets.

Data set	Average iterations	Max. iterations
Swiss roll	23.6	33
Swiss hole	6.5	18
Helix	4.8	8
Faces	34.9	52

To test the effect of the initialization on the size of the resultant landmark set, the procedure of “detect-and-add” has been run on each data set for 100 times. At each time it is given a randomly selected (50 for 3D points and 20 for faces) initial landmarks; and the number of iterations (landmarks added) is recorded. The maximum and the average number of iterations during the 100 runnings for each data set is listed in Table 1. It shows that the procedure does give a reasonable landmark set in practice with arbitrarily chosen initial landmarks. The iteration number loosely reflects the complexity of the manifold.

Then neighbourhood retrieval has been conducted by searching the nearest samples using the obtained structure on each data set. Let $\mathbf{R}(\mathbf{p})$ denote the training samples associated with landmark \mathbf{p} (different from $\text{Cell}_E(\mathbf{p})$ in the structuring stage), and $\eta_k(\mathbf{x}; \mathbf{S})$ denote the k -th nearest (Euclidean distance) point to \mathbf{x} in $\mathbf{S} \subset \mathbf{X}$ (different from $L_E^k(\cdot)$, which refers to the k th nearest landmark). The test is to evaluate the empirical risk of retrieval for a structuring of the data. For each test sample $\mathbf{x} \in \mathbf{X}$, $\eta_k(\mathbf{x}; \mathbf{X})$ and $\eta_k(\mathbf{x}; \mathbf{R}(L_E(\mathbf{x})))$ is compared for $k = 1, 2, \dots$, until for some k they are different. Then k is recorded as $N_R(\mathbf{x})$. Fig. 11 shows the histograms of $\{N_R(\mathbf{x}) | \mathbf{x} \in \mathbf{X}\}$ for the data sets. It shows that the direct query and the one guided by the landmarks for the nearest neighbour yield consistent results for all samples on each of the data sets. In fact, the two queries keep consistent for K -NN with $K > 1$ as well, where K seems to depend on the manifold complexity: for simple manifolds, they will be consistent for large number of K .

In fact, because the structure has topology-keeping nature by construction, when $\eta(\mathbf{x}; \mathbf{X}) \neq \eta(\mathbf{x}; \mathbf{R}(L_E(\mathbf{x})))$, the neighbours retrieved with the structured data are more reliable than those directly from the original data. An experiment is done to demonstrate this point: for every $\mathbf{x} \in \mathbf{X}$, $\eta_k(\mathbf{x}; \mathbf{X})$ and $\eta_k(\mathbf{x}; \mathbf{R}(L_E(\mathbf{x})))$ are found for some k . The geodesic distances are then computed from \mathbf{x} to both $\eta_k(\mathbf{x}, \cdot)$,

$$\delta_k^{\mathbf{X}}(\mathbf{x}) = d_M(\mathbf{x}, \eta_k(\mathbf{x}; \mathbf{X})) \quad (19)$$

$$\delta_k^{\mathbf{R}(L_E(\mathbf{x}))}(\mathbf{x}) = d_M(\mathbf{x}, \eta_k(\mathbf{x}; \mathbf{R}(L_E(\mathbf{x})))) \quad (20)$$

Then the averages are computed,

$$\bar{\delta}_k^{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \delta_k^{\mathbf{X}}(\mathbf{x}_i) \quad (21)$$

$$\bar{\delta}_k^{\mathbf{R}} = \frac{1}{N} \sum_{i=1}^N \delta_k^{\mathbf{R}(L_E(\mathbf{x}_i))}(\mathbf{x}_i) \quad (22)$$

They are compared in Fig. 12 for $k = 1, \dots, 100$ for each data set. Note that, if $\eta_k^{\mathbf{R}(L_E(\mathbf{x}))}$ is not defined for some \mathbf{x} , (when $\mathbf{R}(L_E(\mathbf{x}))$ contains less than k elements), it is simply removed from the average in Eq. (22).

The simple test has been conducted hierarchically for multilayer structuring as well. The algorithm takes a large data set of 50,000 points sampled from the Swiss roll. Two layers of landmarks are generated. First the algorithm yields 1102 landmarks for the original data set, being given 1000 randomly selected initial landmarks (Fig. 13(a)). Taking the 1102 layer-1 landmarks as input, the algorithm generates 55 layer-2 landmarks for 50 layer-1 landmarks randomly selected as initial layer-2 landmarks. Fig. 13(b) shows the landmarks, in which the bigger dots are layer-2 landmarks.

We have also applied the simple test on a real-world data set with *a priori* about the underlying factors. The test images are from one subject from the YaleB face database.⁷ They are varying in illumination directions (2D) and poses (1D). The figure of the resultant landmarks is shown in Fig. 14. In the figure, (a) is drawn such that: the first image I_p is a landmark; the others are samples in $\text{Cell}_E(I_p)$. (b) shows the same landmark, with all samples associated to it. Note that some of the associated samples are far from the landmark by Euclidean distance, however, they share the similar poses and illumination configurations. While (c) displays the parameter space of the data set, with landmarks drawn in dark blue and samples in (b) drawn in green.

Then we use two synthetic manifolds to illustrate the condition proposed in Section 5, and the topology of the neighbourhood graph it requires to preserve. The results on the Swiss roll and S-shaped manifolds are shown in Fig. 15. The left column ((a) and (d)) shows the TEPs detected. In the figure, each inconsistent point \mathbf{x}^8 is linked with $L_E(\mathbf{x})$, and the TEPs detected are shown with red dots. We can see that not all inconsistent points cause short-cuts on the manifold, for those who do cause (in this case, we observe this by intuition), Condition 1 will not hold. The middle column ((b) and (e)) shows the minor graph given by the contraction algorithm with the randomly selected 40 landmarks, and the right column ((c) and (f)) shows the minor graph of the resultant landmarks. From the previous discussion we may derive that if two landmarks are connected in the minor graph, there will be points on the manifold that are equidistant to both of the two landmarks, and farther away to any of the other landmarks. Therefore if there are short-cuts of the manifold in the minor graph, there will be risk of attributing points to a remote landmark on the manifold. Or, roughly speaking, the better the minor graph preserves the manifold structure, the safer to use the landmarks to represent the manifold. The minor graph in (c) and (f) keeps the manifolds much better than those in (b) and (e), which confuses different parts of the manifold together. Thus we have satisfying landmarks when Condition 1 holds.

⁷ <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>

⁸ Recall that it means $L_E(\mathbf{x}) \neq L_M(\mathbf{x})$.

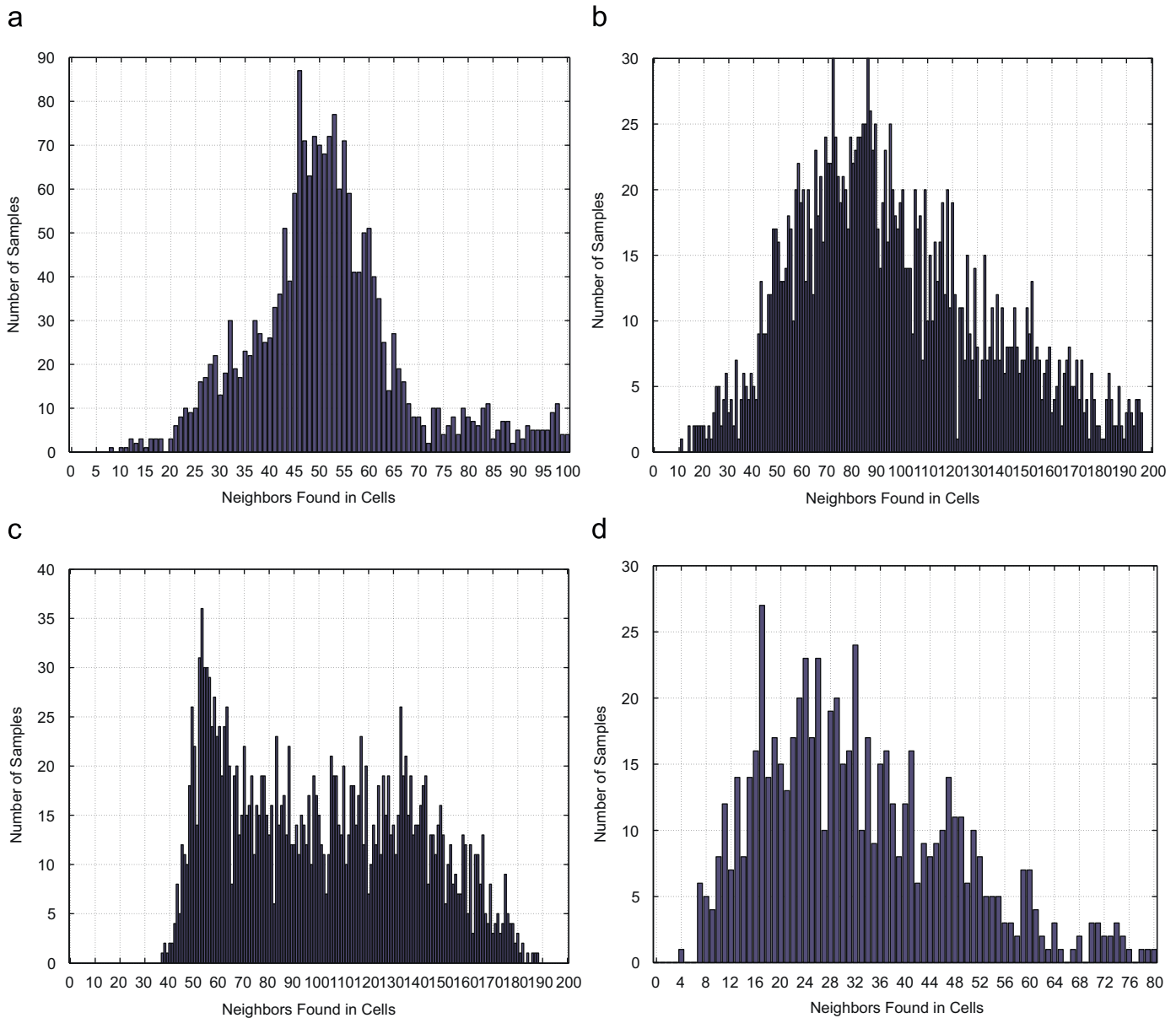


Fig. 11. Histograms of N_r array: (a) Swiss roll; (b) Swiss hole; (c) helix; (d) face images.

We then apply the test on 12,000 handwritten digit images, whose intrinsic dimension is not clear.⁹ The principal component analysis is applied on the images to initially lower the dimensionality to 100. The neighbourhood size is set to 3 for constructing \mathbf{G} . We randomly select 100 samples as the initial landmarks. In Fig. 16(a), we check the TEPs detected in the class of digit “4”. In each triple of the images, the middle one is the sample \mathbf{x} , the left one is $L_E(\mathbf{x})$, and the right one is $L_M(\mathbf{x})$. In most of these TEPs, L_E is an inappropriate landmark for the sample and $L_M(\mathbf{x})$ suggests a proper one. The exceptions are highlighted in the figure, where L_E is appropriate and L_M is not. In table (b), for each digit from “0” to “9”, we list the statistics of: (i) TEPs detected, (ii) $L_E(\mathbf{x})$ with incorrect label, (iii) erroneous $L_E(\mathbf{x})$ but correct $L_M(\mathbf{x})$ (potential gains) and (iv) correct $L_E(\mathbf{x})$ but erroneous $L_M(\mathbf{x})$ (potential loss). Most of the TEPs indicate short-cuts

on the digit manifold, and can be corrected by representing the TEP with the corresponding $L_M(\mathbf{x})$.

By adding more landmarks, the TEPs can be eliminated. In Fig. 17(a), we plot the number of the TEPs versus the number of the landmarks. During the first phrase of the processing, the number of TEPs increases with the number of landmarks. A possible explanation is that as the landmarks increase, the partition of the manifold becomes finer, but is not completely consistent (in the Euclidean space and on the manifold). Therefore, some structural details on the manifold may come out (considering the case of the small U-shaped structure we discussed in Section 3). In Fig. 17(b), several examples of the Euclidean cells of the resultant (when the number of landmarks is 590) landmarks are drawn. We have also tested the added landmarks within a classification procedure. When we add new landmarks to eliminate the TEPs for the digits, we save the first 110, 120, ..., 590 landmarks and apply the nearest neighbour classifier on the 10,000 test samples. For each of these accumulating sets, K-Means is done with the same number of randomly selected

⁹ <http://www.cs.toronto.edu/~roweis/data.html>

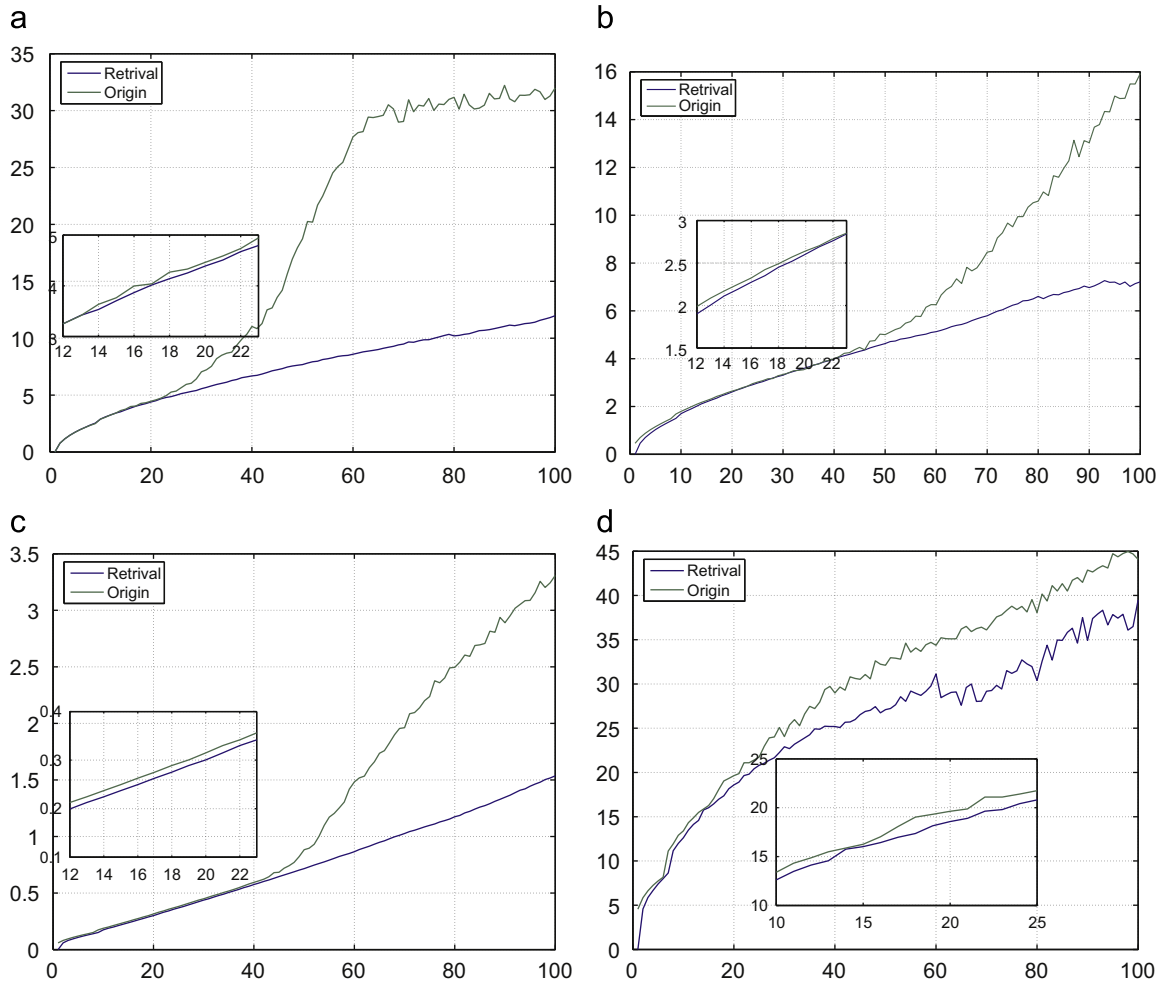


Fig. 12. Geodesic distances to retrieved neighbours: (a) Swiss roll; (b) Swiss hole; (c) helix; (d) face.

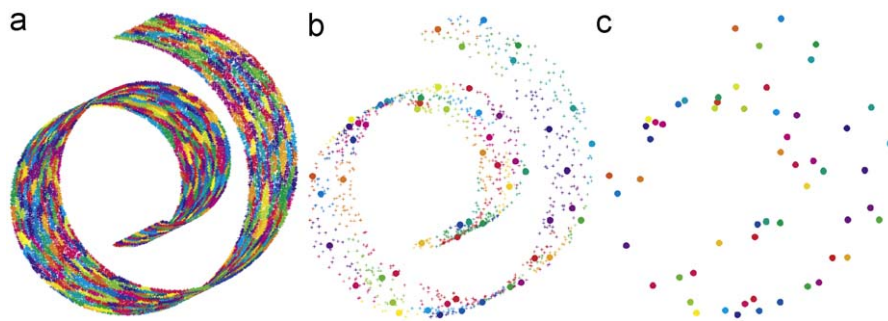


Fig. 13. Hierarchical partition of 50,000 points from Swiss roll: (a) data with 1102 layer-1 LMPs; (b) layer-1 LMPs with 55 layer-2 LMPs; (c) layer-2 LMPs.

landmarks. The whole procedure are done for 10 times. These results are shown in Fig. 17(c), which suggests that our proposed topology safe landmarks represents the data distribution in a constantly better way.

In the following experiment, we test the optimization-based algorithm for adjusting the landmarks. To validate the effectiveness of the adjustment, we use fewer landmarks: 30 in the Swiss roll and the S-shaped point clouds, respectively. Starting with the randomly selected landmarks, we apply both our optimization and the K-Means

algorithms.¹⁰ After each iteration, the training points with the resultant landmarks are tested with Condition 1 and the number of TEPs are recorded and shown in Fig. 18. The fewer TEPs caused by the landmark set from our optimization algorithm indicates the algorithm favours the landmark sets which distribute on the manifold

¹⁰ They both converge in most cases in 10 iterations. We set the maximum iteration number to 20.

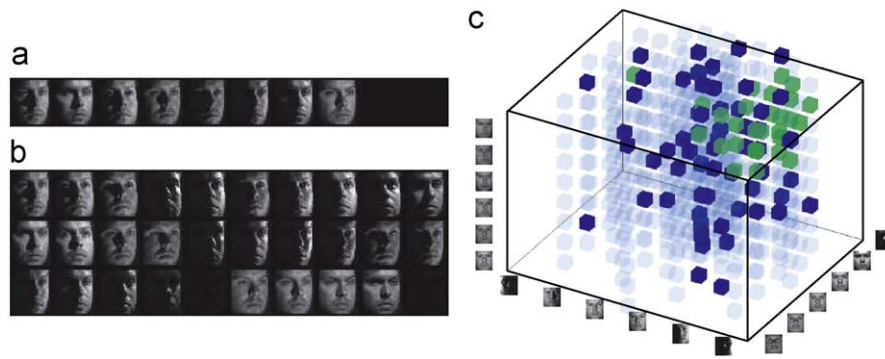


Fig. 14. Structuring of YaleB face database: (a) LMP with samples in a Voronoi cell; (b) LMP with associated samples; (c) parameter space.

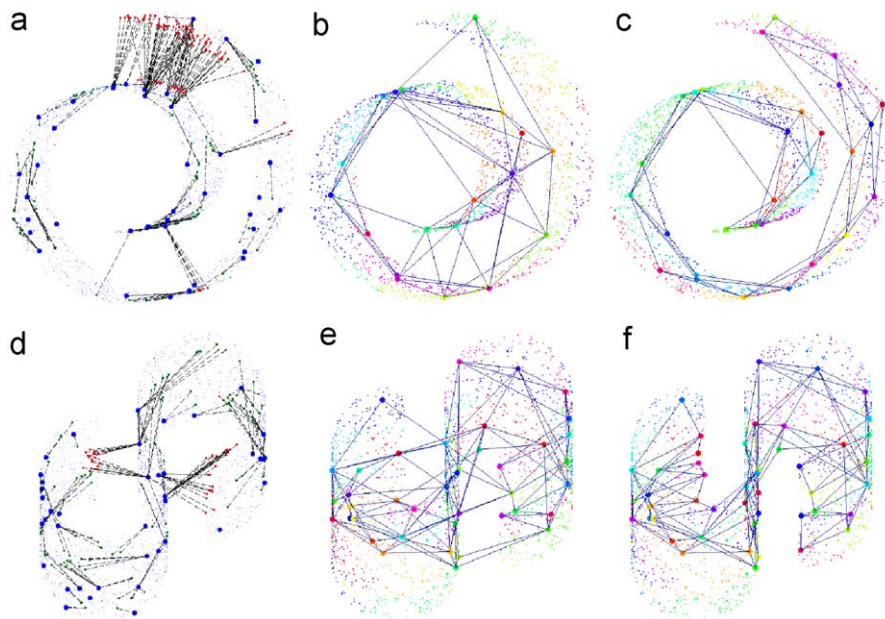


Fig. 15. Condition 1 and the topology of the manifolds: (a) and (d) big blue dots: landmarks; red dots: TEPs; green dots: $L_E \neq L_M$, but NOT TEPs. Links are made between a point and its L_E ; (b) and (e) minor graph of the initial landmarks; (c) and (f) minor graph of the resultant landmarks (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

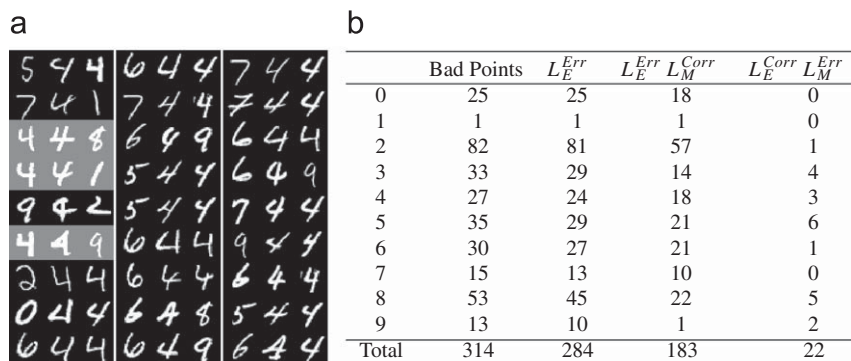


Fig. 16. Detecting TEPs on handwritten digits: (a) TEPs found for “4”. In the table (b), *Err*: the landmark gives the incorrect label; *Corr* is for correct.

in such a way that the points on the manifold are less possible to be represented by a landmark that is remote on the manifold. Thus the representation is more reliable than that from the K-Means.

Finally, let us show an example of the possible application on semi-supervised classification. The data are shown in Fig. 19. In both of the data sets, there are 5000 training samples and another 5000

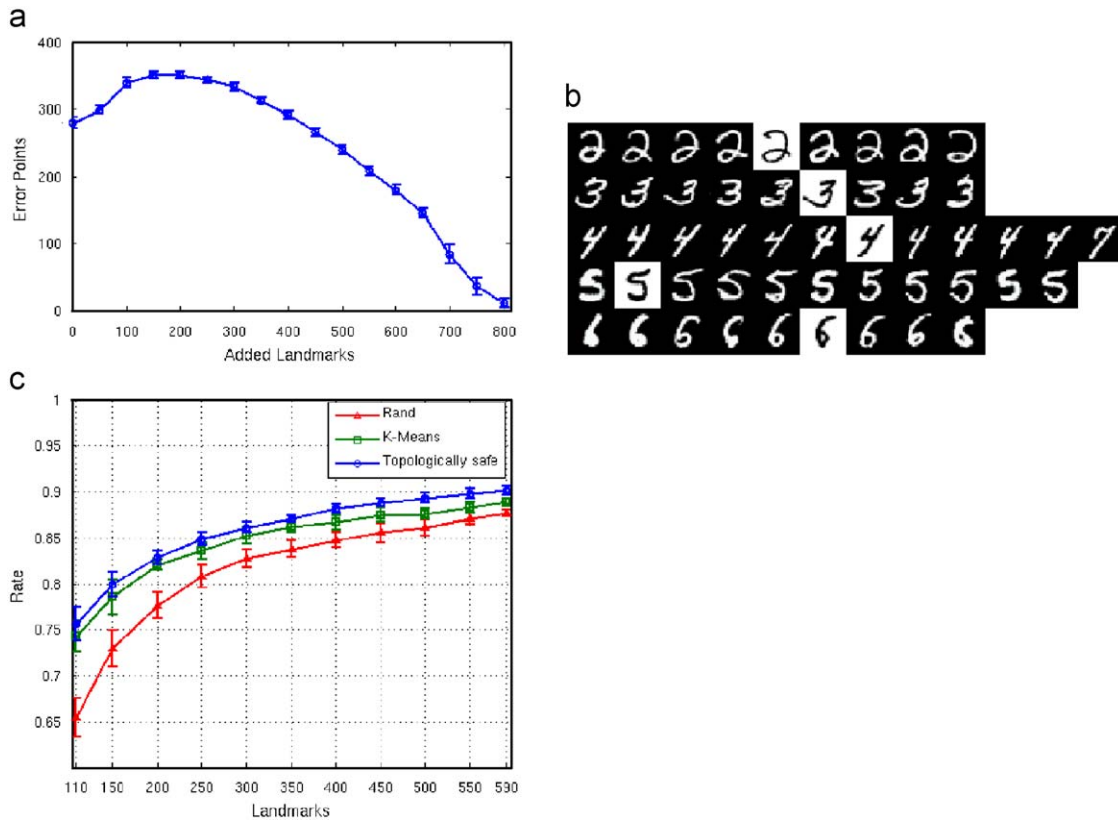


Fig. 17. Eliminate TEPs for handwritten digits: (a) TEPs versus number of landmarks; (b) example cells; (c) recognition rate with landmarks: (i) added according to our topological safe condition, (ii) from K-Means and (iii) randomly selected.

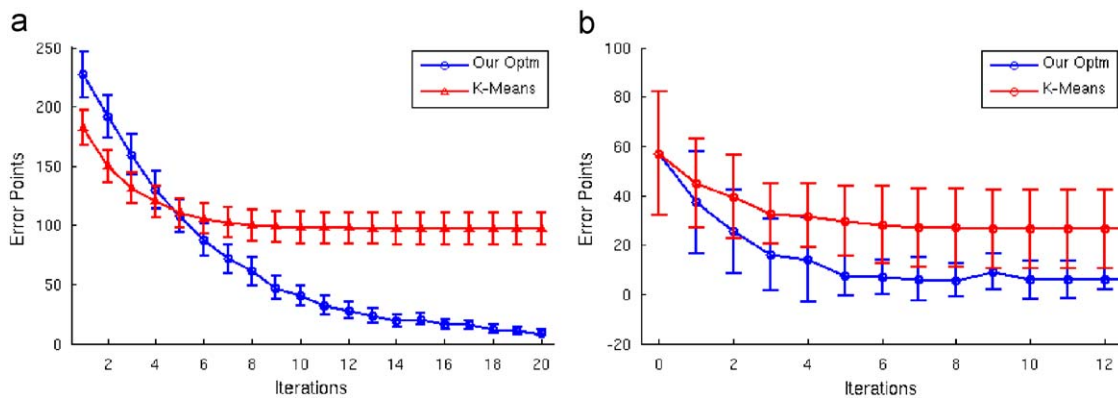


Fig. 18. Optimizing landmarks. TEPs versus iteration number: (a) on the Swiss roll; (b) on the S-shaped manifold.

for testing. The neighbourhood size is 8. The experiment scheme is as follows:

1. Manually select a few points as the initial landmarks \mathbf{P}_{init} and label them.
2. Detect and eliminate (by simply adding new landmarks) the TEPs:
 - (a) Each newly added landmark is labelled according to its nearest landmark on the manifold in the current landmark set.
 - (b) After all TEPs have been eliminated, the number of added landmarks N_{Add} is recorded.
3. Test the optimization algorithm, N_{Add} random selected landmarks are added to \mathbf{P}_{init} and labelled (as known), so that the landmark set being optimized has the same number of landmarks as that

of the result of the adding procedure in the previous step. The optimization is conducted.

4. Initialized with the same landmark set as that for the optimization, K-Means is applied.

Then we use each of these resulting landmark sets to classify the test points by labelling each point \mathbf{x} according to $L_E(\mathbf{x})$ in the corresponding landmark set. The error is recorded and shown in the figures. Both of our schemes outperform K-Means. This is because the natural metric in the data differs from the Euclidean metric, which K-Means uses for evaluating and optimizing its objective function. In contrast, the add-and-test scheme considers the non-Euclidean structure in the data explicitly; and our optimization-based scheme

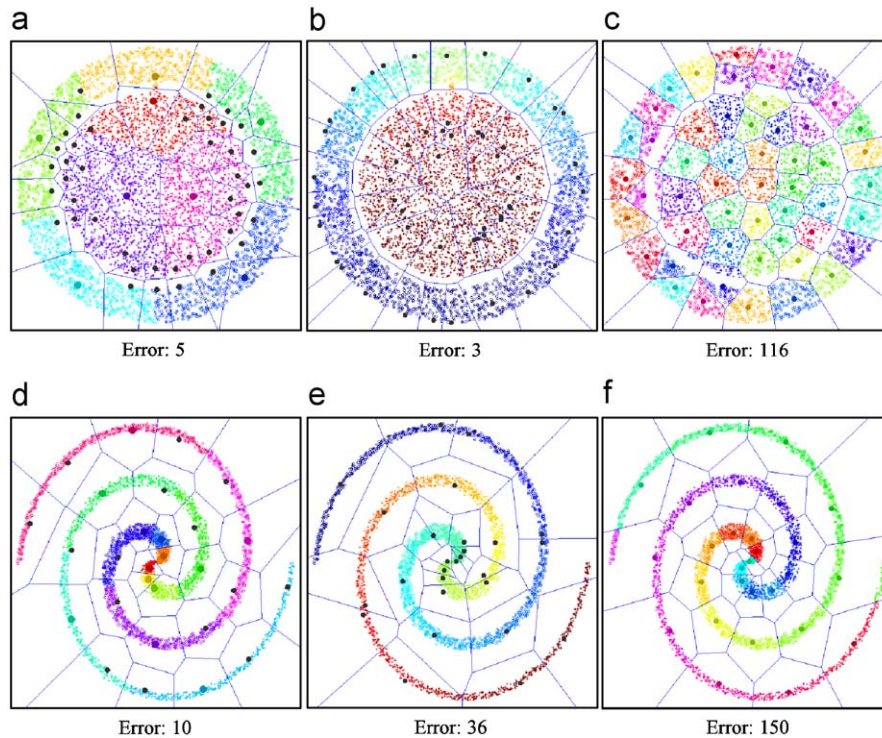


Fig. 19. Semi-supervised classification: (a) and (d) colourized according to π_M in the initial landmark set (colourful big dots). The added landmarks are black big dots; (b) and (e) colourized according to the value of f . The landmarks are big dots; (c) and (f) colourized according to π_E in the resulting landmarks. The landmarks are big dots. In each pane, “+”/“o” indicate the true classes of the points. The Voronoi tessellations of the resulting landmarks are drawn. The error number below is from the classification of 5000 test samples from the same distribution using the landmarks in the corresponding pane (a, d) add; (b, c) optimization; (c, f) K-Means. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

defines its objective function based on the metric in the parametrization space from Laplacian eigenmaps.

8. Conclusion and discussion

In this paper, the problem of choosing representative samples from a point cloud on a manifold has been addressed. Different from previous work, we do not choose landmarks for training the manifold models, but for speeding up the nearest neighbour search on a manifold, which is a critical and time-consuming step for using the learned manifold models, e.g., out-of-sample extensions. We propose a criterion of testing whether a subset of the data samples can accurately and reliably represent the manifold. Our condition ensures that the individual Voronoi cells generated by the landmarks in the ambient space do *not* intersect the manifold at distant locations and thus preserves the manifold topology. Therefore, the landmarks satisfying this criterion can help locate a new test sample on the manifold, so that its closest neighbours can be found. We have also proposed an optimization method to improve the distribution of an existing landmark set on the manifold.

Computational issues for the optimization: As we have mentioned, to compute the objective function of the optimization, one needs to compute the low-dimensional coordinates of the manifold in the first place. This may arise criticism about the computational cost. However, recall that the proposed method is *not* used for training the manifold models. Instead, it is to generalize the learned models. Under such circumstances, the required low-dimensional coordinates of the points on the manifold are already known. Therefore, to set up the optimization procedure would not cause much overheads in practice. However, the estimator for d_M used in the objective function must comply with the learned model.

Scaling issues: Another concern may be that: the proposed criterion depends on the representation of the data. It becomes invalid, if the variables (dimensions) scale non-uniformly. To this issue, our argument above still applies. As our method is not for learning the whole point cloud’s embedding, but for expending the learned embedding to a new point, it is reasonable to assume that the new point has the same scaling with that of the points in the training point cloud, from which the embedding is learned.

Future directions: What we have done is to test the empirical risk for a landmark set. However, this depends on the sampling density and noise. In the future work, we will study the expected risk by taking a statistical point of view of the data.

The current study can be seen as on the consistency between Voronoi tessellation in two Riemannian spaces, which are both determined by the respective distance metrics. We will consider the possibility that generalizing this to multiple metric spaces.

Appendix A. Why adding landmarks works

Let $\mathbf{P}^{(k)}$ be the current landmark set. Denote the maximum *radius* of the cells in the corresponding Voronoi diagram as

$$R(\mathbf{P}^{(k)}; \mathcal{M}) = \max_{\mathbf{x} \in \mathcal{M}} d_E(\mathbf{x}, L_E(\mathbf{x}; \mathbf{P}^{(k)})) \quad (23)$$

Because the manifold \mathcal{M} is constant, we denote the equation as $R(\mathbf{P}^{(k)})$. Suppose there is a function that judges whether connecting two points on a manifold is topologically safe, i.e., connection between them does not cause a “short-circuit” on \mathcal{M} ,

$$f(\mathbf{x}_1, \mathbf{x}_2; \mathcal{M}) = \begin{cases} 0 & \text{if } (\mathbf{x}_1, \mathbf{x}_2) \text{ causes short-cut} \\ 1 & \text{if } (\mathbf{x}_1, \mathbf{x}_2) \text{ is topologically safe} \end{cases} \quad (24)$$

Let

$$d_{TS}(f; \mathcal{M}) = \max_{f(\mathbf{x}_1, \mathbf{x}_2; \mathcal{M})} d_E(\mathbf{x}_1, \mathbf{x}_2) \quad (25)$$

For a manifold \mathcal{M} and an appropriately defined f , d_{TS} is a positive constant, which means the “longest” “topologically safe” connection between an arbitrary pair of points on \mathcal{M} . In the k -th iteration, one adds a new landmark into $\mathbf{P}^{(k-1)}$ to make $\mathbf{P}^{(k)}$, thus $\mathbf{P}^{(0)} \subset \mathbf{P}^{(1)} \subset \dots$. According to the definition in Eq. (23), for $\mathbf{P}^{(k_1)} \subset \mathbf{P}^{(k_2)}$, we have $R(\mathbf{P}^{(k_1)}) \supseteq R(\mathbf{P}^{(k_2)})$. Therefore $R(\mathbf{P}^{(0)}) \supseteq R(\mathbf{P}^{(1)}) \supseteq \dots$, and when $\mathbf{P}^{(k)}$ approaches \mathbf{X} , $R(\mathbf{P}^{(k)})$ reaches 0. Thus when $R(\mathbf{P}^{(k)}) < d_{TS}(f; \mathcal{M})$, the algorithm exits with success. If we substitute a finite collection of samples of the manifold for \mathcal{M} itself, the statements above also hold. So our method works fine in practice.

Appendix B. Proof of Proposition 1

Proof. (1) If $\mathbf{p}_i \sim \mathbf{p}_j$ in \mathbf{H}_E , according to the contraction procedure, we have $\text{Cell}_E(\mathbf{p}_i) \sim \text{Cell}_E(\mathbf{p}_j)$ in \mathbf{G} (see below). Therefore, there exist equidistant points on \mathcal{M} to \mathbf{p}_i and \mathbf{p}_j . Take one of these points, \mathbf{x} , and let $L_M(\mathbf{x}) = \mathbf{p}_k$. By Condition 1, $\{\mathbf{x}\} \subset \text{Cell}_E(\mathbf{p}_i) \cap \text{Cell}_M(\mathbf{p}_k) \neq \emptyset$ implies $\text{Cell}_M(\mathbf{p}_i)$ and $\text{Cell}_M(\mathbf{p}_k)$ are adjacent, and thus $\mathbf{p}_i \sim \mathbf{p}_k$ in \mathbf{H}_M . Using the same deduction, we have $\mathbf{p}_j \sim \mathbf{p}_k$ in \mathbf{H}_M as well. If $k = i$ or $k = j$, then $\mathbf{p}_i \sim \mathbf{p}_j$ in \mathbf{H}_M (Fig. 7(a)). Otherwise, \mathbf{p}_i and \mathbf{p}_j are connected in \mathbf{H}_M by a path $(\mathbf{p}_i, \mathbf{p}_k, \mathbf{p}_j)$ (Fig. 7(b)).

In practice, however, there are generally no such equidistant points in the discrete point set sampled from \mathcal{M} . Nevertheless, because $\text{Cell}_E(\mathbf{p}_i) \sim \text{Cell}_E(\mathbf{p}_j)$ in \mathbf{G} , $\exists \mathbf{x}_1 \sim \mathbf{x}_2$ in \mathbf{G} , where $L_E(\mathbf{x}_1) = \mathbf{p}_i$ and $L_E(\mathbf{x}_2) = \mathbf{p}_j$. Consider $L_M(\mathbf{x}_1) = \mathbf{p}_{k_1}$ and $L_M(\mathbf{x}_2) = \mathbf{p}_{k_2}$, because $\mathbf{x}_1 \sim \mathbf{x}_2$, we have $\text{Cell}_M(\mathbf{p}_{k_1}) \sim \text{Cell}_M(\mathbf{p}_{k_2})$ in \mathbf{G} and equivalently $\mathbf{p}_{k_1} \sim \mathbf{p}_{k_2}$ in \mathbf{H}_M . And as discussed in the continuous case above, we have $\mathbf{p}_i \sim \mathbf{p}_{k_1}$ and $\mathbf{p}_j \sim \mathbf{p}_{k_2}$ in \mathbf{H}_M . Depending on whether (a) $k_1 = i, k_2 = j$, (b(i)) $k_1 = i, k_2 \neq j$, (b(ii)) $k_1 \neq i, k_2 = j$ or (d) $k_1 \neq i, k_2 \neq j$, the path between \mathbf{p}_i and \mathbf{p}_j in \mathbf{H}_M corresponds to one of the cases (a)–(c) in Fig. 7.

(2) If $\mathbf{p}_i \sim \mathbf{p}_j$ in \mathbf{H}_M , consider the shortest path $(\mathbf{p}_i = \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n = \mathbf{p}_j)$ in \mathbf{G} . It is not difficult to see, all points on the path is within either $\text{Cell}_M(\mathbf{p}_i)$ or $\text{Cell}_M(\mathbf{p}_j)$. Thus by Condition 1, if $\text{Cell}_E(\mathbf{p})$ contains these points, either (i) \mathbf{p} is $\mathbf{p}_i, \mathbf{p}_j$ or (ii) $\text{Cell}_M(\mathbf{p})$ is adjacent to $\text{Cell}_M(\mathbf{p}_i)$ or $\text{Cell}_M(\mathbf{p}_j)$. Let $\mathbf{U}_{ij}^M = \{\mathbf{p} | \mathbf{p} \sim \mathbf{p}_i \text{ or } \mathbf{p} \sim \mathbf{p}_j\}$ in \mathbf{H}_M . There exists a path between \mathbf{p}_i and \mathbf{p}_j in \mathbf{H}_E consisting only nodes in \mathbf{U}_{ij}^M (Fig. 7(d)). \square

B.1. Proof of cell adjacency

In the following, we provide a sketch of the proof that: $\mathbf{p}_i \sim \mathbf{p}_j$ in $\text{Contract}(\mathbf{G}, L) \Leftrightarrow \text{Cell}(\mathbf{p}_i) \sim \text{Cell}(\mathbf{p}_j)$ in \mathbf{G} .

Proof. “ \Leftarrow ”: If $\text{Cell}(\mathbf{p}_i) \sim \text{Cell}(\mathbf{p}_j)$ in \mathbf{G} , $\exists \mathbf{x}_1 \sim \mathbf{x}_2$ in \mathbf{G} , where $L(\mathbf{x}_1) = \mathbf{p}_i$ and $L(\mathbf{x}_2) = \mathbf{p}_j$. Let $\mathbf{y}_1^{(0)} = \mathbf{x}_1$ and $\mathbf{y}_2^{(0)} = \mathbf{x}_2$, after the k -th contraction, let $\mathbf{y}_1^{(k)}$ and $\mathbf{y}_2^{(k)}$ in $\mathbf{G}^{(k)}$ be the nodes $\mathbf{y}_1^{(k-1)}$ and $\mathbf{y}_2^{(k-1)}$ in $\mathbf{G}^{(k-1)}$ or the nodes they are contracted to in the k -th step of contraction. In intuitive words, “trace” them. Note that $\mathbf{y}_1^{(k)} \sim \mathbf{y}_2^{(k)}$ is true for all k . If the contraction ends after k_N steps, we have $\mathbf{y}_1^{(k_N)} = \mathbf{p}_i \sim \mathbf{p}_j = \mathbf{y}_2^{(k_N)}$.

“ \Rightarrow ”: Because contraction does not change the connectivity of a graph and $\mathbf{p}_i \sim \mathbf{p}_j$ in $\mathbf{G}^{(k_N)}$, for all $k = 0, \dots, k_N$, there are paths between \mathbf{p}_i and \mathbf{p}_j in $\mathbf{G}^{(k)}$. Consider a minimum number k_0 , such that all paths between \mathbf{p}_i and \mathbf{p}_j in $\mathbf{G}^{(k_0)}$ consist of nodes from only $\text{Cell}(\mathbf{p}_i) \cup \text{Cell}(\mathbf{p}_j)$ in \mathbf{G} . It is easy to check, if there exists such k_0 , then $k_0 = 0$ and there exists a pair of nodes on the path belonging to $\text{Cell}(\mathbf{p}_i)$ and $\text{Cell}(\mathbf{p}_j)$, respectively. Therefore, $\text{Cell}(\mathbf{p}_i)$ and $\text{Cell}(\mathbf{p}_j)$ are adjacent in \mathbf{G} . \square

References

- [1] N. Amenta, M. Bern, M. Kamvysselis, A new voronoi-based surface reconstruction algorithm, in: Proceedings of Siggraph, 1998, pp. 415–421.
- [2] S. Arya, D. Mount, N. Netanyahu, R. Silverman, A. Wu, An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, Journal of the ACM 45 (1998) 891–923.
- [3] M. Belkin, Problems of learning on manifolds, Ph.D. Thesis, Department of Mathematics, University of Chicago, 2003.
- [4] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation 15 (6) (2003) 1373–1396.
- [5] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.
- [6] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N.L. Roux, M. Ouimet, Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering, in: Advances in Neural Information Processing Systems (NIPS), vol. 16, 2004.
- [7] J. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (9) (1975) 509–517.
- [8] M. Brand, Charting a manifold, in: Advances in Neural Information Processing Systems (NIPS), vol. 15, 2002, pp. 961–968.
- [9] H. Chang, D.-Y. Yeung, Y. Xiong, Super-resolution through neighbor embedding, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2004, pp. 275–282.
- [10] V. de Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, in: Advances in Neural Information Processing Systems (NIPS), vol. 15, 2003, pp. 705–712.
- [11] V. de Silva, J.B. Tenenbaum, Selecting landmark points for sparse manifold learning, in: Advances in Neural Information Processing Systems (NIPS), vol. 18, 2006, pp. 1241–1248.
- [12] D.L. Donoho, C. Grimes, Hessian eigenmaps: new locally linear embedding techniques for highdimensional data, Technical Report TR2003-08, Department of Statistics, Stanford University, 2003.
- [13] D.L. Donoho, C. Grimes, Image manifolds which are isometric to Euclidean space, Journal of Mathematical Imaging and Vision 23 (1) (2005) 5–24.
- [14] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Very Large Data Bases Conference (VLDB), 1999, pp. 518–529.
- [15] X. He, S. Yan, Y. Hu, P. Niyogi, Face recognition using Laplacian faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.
- [16] H. Hoppe, Progressive meshes, in: In Proceedings of Siggraph, 1996, pp. 99–108.
- [17] X. Huo, A.K. Smith, Performance analysis of a manifold learning algorithm in dimension reduction, Technical Report, Statistics Group, Georgia Institute of Technology, 2006.
- [18] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
- [19] R. Jenssen, D. Erdogmus, J. Principe, T. Eltoft, The Laplacian PDF distance: a cost function for clustering in a kernel feature space, in: Advances in Neural Information Processing Systems, vol. 17, 2005, pp. 625–632.
- [20] A. Kolmogorov, S. Fomin, Introductory Real Analysis, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970.
- [21] S. Lafon, Diffusion maps and geometric harmonics, Ph.D. Thesis, Yale University, 2004.
- [22] S. Lafon, Y. Keller, R.R. Coifman, Data fusion and multicue data matching by diffusion maps, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006) 1784–1797.
- [23] S. Lazebnik, M. Raginsky, Learning nearest-neighbor quantizers from labeled data by information loss minimization, in: Proceedings of International Conference on Artificial Intelligence and Statistics, 2007.
- [24] J. Li, P. Hao, Hierarchical structuring of data on manifolds, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
- [25] T. Lin, H. Zha, Riemannian manifold learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (5) (2008) 796–809.
- [26] S.T. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
- [27] L.K. Saul, S.T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (1) (2003) 119–155.
- [28] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
- [29] P. van Oosterom, Spatial Access Methods, vol. 1, Wiley, New York, 1999, pp. 385–400 (Chapter 2).
- [30] N. Vasiloglou, A.G. Gray, D.V. Anderson, Parameter estimation for manifold learning through density estimation, in: Machine Learning for Signal Processing, 2006, pp. 211–216.
- [31] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (1) (2007) 40–51.
- [32] M.-C. Yeh, I.-H. Lee, G. Wu, Y. Wu, E. Chang, Manifold learning, a promised land or work in progress? in: Proceedings of IEEE Conference on Multimedia and Expo, 2005.
- [33] C. Zhang, J. Wang, N. Zhao, D. Zhang, Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction, Pattern Recognition (2004) 325–336.

- [34] J. Zhang, H. Shen, Z.-H. Zhou, Unified locally linear embedding and linear discriminant analysis algorithm (ULLELDA) for face recognition, in: *SINOBIOMETRICS*, 2004.
- [35] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, *SIAM Journal on Scientific Computing* 26 (1) (2005) 313–338.

About the Author—JUN LI is a Ph.D. student in Queen Mary, University of London. He received his B.Sc. in Computer Science from Shandong University, China. Then he studied in Peking University and obtained his M.Sc. in 2006.

About the Author—PENGWEI HAO was awarded the Ph.D. in 1997 from the Institute of Remote Sensing Applications, Chinese Academy of Sciences. He then worked at the Center for Information Science, Peking University and became an associate professor in 1999. In 2000, he worked as a visiting scientist with Prof. Maria Petrou at CVSSP, University of Surrey. In June 2001, he joined Queen Mary, University of London.