Neighborhood Selection for Dimensionality Reduction

Paola Campadelli, Elena Casiraghi, and Claudio Ceruti(✉)

Dipartimento di Informatica, Università Degli Studi di Milano,
Via Comelico 39-41, 20135 Milano, Italy
{paola.campadelli,elena.casiraghi,claudio.ceruti}@unimi.it

Abstract. Though a great deal of research work has been devoted to the development of dimensionality reduction algorithms, the problem is still open. The most recent and effective techniques, assuming datasets drawn from an underlying low dimensional manifold embedded into an high dimensional space, look for "small enough" neighborhoods which should represent the underlying manifold portion. Unfortunately, neighborhood selection is an open problem, for the presence of noise, outliers, points not uniformly distributed, and to unexpected high manifold curvatures, causing the inclusion of geodesically distant points in the same neighborhood. In this paper we describe our neighborhood selection algorithm, called ONeS; it exploits both distance and angular information to form neighborhoods containing nearby points that share a common local structure in terms of curvature. The reported experimental results show the enhanced quality of the neighborhoods computed by ONeS w.r.t. the commonly used k-neighborhoods solely employing the euclidean distance.

Keywords: Dimensionality reduction · Manifold learning · Neighborhood selection

1 Introduction

When developing automatic solutions to problems in the pattern recognition field, most researchers are confronted with intrinsically low dimensional data lying in a very high dimensional space. This requires `dimensionality reduction` (`dr`) as the first and fundamental processing step, to reduce the data dimensionality without losing important information.

To this aim, several `dr` techniques have been proposed in the past, such as `Multidimensional Scaling` (`MDS`) algorithms [14], [9], [15], the mostly used `Principal Component Analysis` (`PCA` [8]), Curvilinear Component Analysis (`CCA`) [3], `ISOMAP` [16], Local Linear Embedding (`LLE` [12]), `Local Tangent Space Alignment` (`LTSA` [23]) and its variants [17], [21], [20], [18], [22], and the `CycleCut` algoritm [5].

Among them, the most recent and effective methods assume that the input points are uniformly drawn from an underlying locally smooth manifold $\mathcal{M} \subseteq \Re^d$

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 183–191, 2015.
DOI: 10.1007/978-3-319-23231-7_17

embedded into an higher D-dimensional space $\mathcal{M} \subseteq \Re^d \subset \Re^D (d \leq D)$, where d is the `intrinsic dimension` (`id`) of $\mathcal{M}$. This leads to the consideration that, though the manifold local smoothness guarantees that "small enough" manifold neighborhoods can be well approximated by their local tangent spaces, the embedding map might produce unexpected folds in $\mathcal{M}$, that should be properly accounted for when trying to discover the underlying manifold geometry.

Practically, most `dr` algorithms pursue the following steps:

1) `Id` **estimation**: the `id` of the embedded manifold $\mathcal{M}$ is a fundamental information, usually unknown. However, reliable estimates of its value can be computed by one of the `id` estimators recently proposed [10], [2], [6], [11], [13] , [1]

2) **Neighborhoods selection**: it is based on the fact that "small enough" neighborhoods may reliably approximate the underlying manifold's neighborhoods. Two strategies are commonly applied to select the neighborhoods of each point $\mathbf{x}$: one takes the k nearest neighbors of $\mathbf{x}$, the other selects the points in the D-dimensional ball centered in $\mathbf{x}$ and having radius ϵ (both k and ϵ are parameters to be set).

3) **Dimensionality reduction**: a reduction function is found, that preserves neighborhoods relations.

Among the `dr` techniques applying these steps, we consider `LTSA` and its variants. Assuming that point neighborhoods (approximating the underlying manifold's portions) are "smooth enough" to be well approximated by a linear tangent space, `LTSA` firstly applies local `SVD` to estimate the local tangent spaces approximating them; secondly, it computes the global mapping by finding the point coordinates that produce the best alignment among all the local tangent spaces. It must be noted that an accurate global mapping is obtained only if the point neighborhoods are "large enough" to guarantee a good overlap among nearby neighborhoods; unfortunately, the required overlap causes too large neighborhoods, that often include noise, outliers, points not uniformly distributed, or points belonging to geodesically distant regions, due to unexpected folds generated by high manifold curvatures. In this cases the assumptions of `LTSA` are violated since the approximation of the neighborhoods via local tangent spaces is not accurate. For this reason, different variants of `LTSA` have been proposed [17], [21], [20], [18], [22], which mainly modify the neighborhood selection and the local tangent space construction.

Though these variants are theoretically sound and the reported experiments seem promising, it is quite difficult to objectively compare them since none of them has been tested on standard databases; besides, they are mostly tested on classical manifolds, such as the Swiss Roll, and results are only visually presented. Though other `dr` works [19], [7], [4] are also focusing on the "bottleneck" of neighborhoods selection, the problem is still crucial and open.

For this reason in this paper we present our proposal, called `Optimal Neighborhood Selection` (`ONeS`). Based on the reliable results obtained by the `id` estimator employing both distance and angular information [1], `ONeS` builds

the point neighborhoods by analyzing not only pairwise distances but also angular informations.

This paper is organized as follows: in Sect. 2 we outline `ONeS`; in Sect. 3 we describe the obtained results and report future works.

2 Algorithm

In this section we describe `ONeS`, which exploits the local angular distribution to improve the construction of the local point neighborhoods, each describing the local structure of the underlying manifold portion.

More formally, given a D-dimensional dataset $\mathbf{X}_n = \{\mathbf{x}_i, \cdots, \mathbf{x}_n\} \subset \Re^D$ composed by points sampled from a locally smooth d-dimensional manifold $\mathcal{M} \subset \Re^d$ (with $d \leq D$), we define the distance based k-neighborhoods of a point, $\mathbf{x}_i \in \mathbf{X}_n$, as the set containing the k nearest points in terms of euclidean distance, and we indicate it as $\mathcal{N}_{dist}(\mathbf{x}_i, k)$.

This k-neighborhood is commonly used by most of the manifold learning and `dr` techniques; we also exploit this structure to discover the local angular distribution describing the curvature of the manifold portion underlying each neighborhood. Precisely, for each sample $\mathbf{x}_i \in \mathbf{X}_n$, $\mathcal{N}_{dist}(\mathbf{x}_i, k)$ is firstly centered in $\mathbf{x}_i$; we then measure the angle between each point $\mathbf{x}_j \in \mathcal{N}_{dist}(\mathbf{x}_i, k)$ and each of the D axis of the canonical base of $\Re^D$, $\mathbf{e}_d$ with $d \in \{1, \cdots, D\}$ ($\mathbf{e}_d \in \Re^D$ is a vector composed by zeros, except for the value 1 in the position d). In this way, given the k neighbors $\mathbf{x}_j \in \mathcal{N}_{dist}(\mathbf{x}_i, k)$ of $\mathbf{x}_i$, we obtain for each $\mathbf{e}_d$ ($d = 1, \cdots, D$), the k angles:

$$\alpha_j = acos\left(\frac{< \mathbf{x}_j, \mathbf{e}_d >}{\|\mathbf{x}_j\|}\right) \tag{1}$$

where *acos* is the inverse of the cosine function. Afterwards, for each axis $\mathbf{e}_d$ we take the computed k angles α_j and we build an histogram of their distribution. More precisely, we split the interval $[0, \pi]$ in a fixed number of equal bins (which we experimentally set to 8), and we count the numbers of angles that fall inside each of them. As a result, we obtain D histograms, $\{h_1(\mathbf{x}_i), \cdots, h_D(\mathbf{x}_i)\}$ for the point $\mathbf{x}_i$, one for each axis; their concatenation produces a single histogram $\mathbf{h}(\mathbf{x}_i) = [h_1(\mathbf{x}_i), \cdots, h_D(\mathbf{x}_i)]$.

Having computed the histograms $\mathbf{h}(\mathbf{x}_i)$ for all the dataset points $\mathbf{x}_i \in \mathbf{X}_n$, we assign to each point $\mathbf{x}_i$ the average of the k histograms $\mathbf{h}(\mathbf{x}_j)$ computed for its k neighbors:

$$\overline{\mathbf{h}}(\mathbf{x}_i) = \frac{\mathbf{h}(\mathbf{x}_i) + \sum_{j=1}^{k} \mathbf{h}(\mathbf{x}_j)}{k+1} \qquad where \qquad \mathbf{x}_j \in \mathcal{N}_{dist}(\mathbf{x}_i, k) \tag{2}$$

This reduces the variability between histograms of nearby points, which could be influenced by the presence of outliers, noise and variation in the density of the points sampled from the underlying manifold.

Finally, we are ready to build neighborhoods based on the similarity between the average local angular distribution of each point. In particular, we use the χ^2 distance for measuring the similarity between histograms. It is defined as:

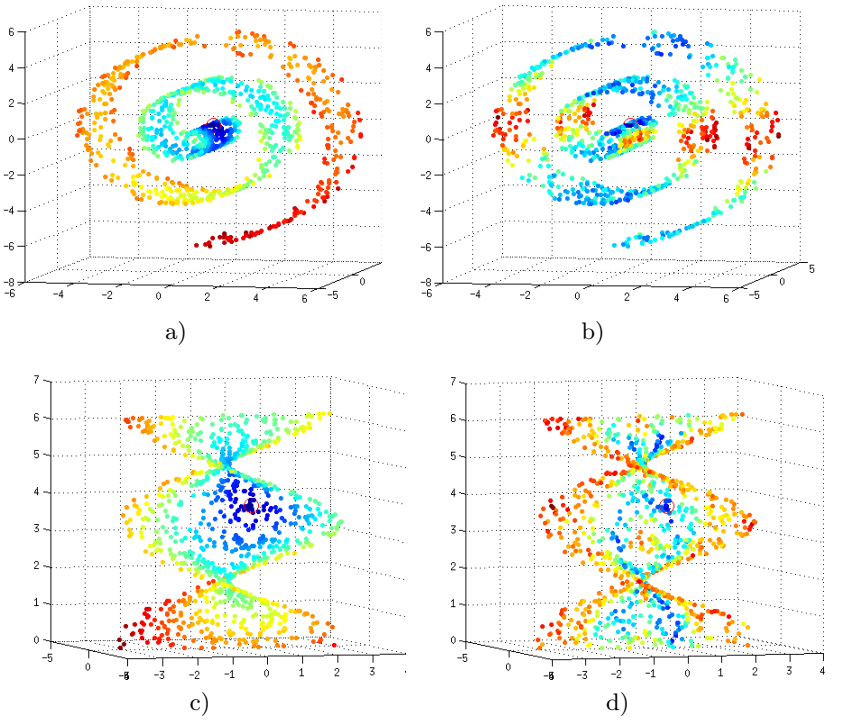

Fig. 1. Reported figures depict the distances from a given point, circled in red, measured by means of the euclidean distance and the χ^2 histogram distance of angular distribution (blue points are the closest ones, whilst the red points are the farthest): a) Swiss Roll with Euclidean distance b) Swiss Roll with histogram distance c) Helix with Euclidean distance d) Helix with histogram distance

$$\chi^2(h_1, h_2) = \sum_{n=1}^{bins} \frac{(h_1(n) - h_2(n))^2}{h_1(n)} \tag{3}$$

Taking for each $\mathbf{x}_i \in \mathbf{X}_n$ the k closest points in terms of the χ^2 histogram distance, we get the angular based neighborhood $\mathcal{N}_{ang}(\mathbf{x}_i, k)$. By doing so, for each point we build two different neighborhoods which embody two different informations: the distance based neighborhood $\mathcal{N}_{dist}(\mathbf{x}_i, k)$, which is based on the proximity between points on the underlying manifold, and the $\mathcal{N}_{ang}(\mathbf{x}_i, k)$ neighborhood, which is based on local angular distribution and allows to select as neighbors those points that share a common local structure in terms of curvature (see Figure 1). A joint use of these two structures allows to obtain local neighborhoods that not only have small radius, but are also less affected by the curvature due to the manifold embedding.

Algorithm 1. ONeS

```
Require: Dataset X_n, k, k̂
  return Neighborhood set N_final(x_i, k) for each x_i ∈ X_n
  for i = 1, ··· , n do
    N_dist(x_i, k) = Find_nn_euclid(X_n, x_i, k)
    [The function Find_nn_euclid(X_n, x_i, k) finds the k neighbors of the point x_i in X_n
    using Euclidean distance.]
    Ñ_dist(x_i, k) = N_dist(x_i, k) − x_i
    for d = 1, ··· , D do
      for x_j ∈ Ñ_dist(x_i, k) do
        α_j = ∠x_j, e_d
      end for
      h_d(x_i) = Histogram({α_j})
    end for
    h(x_i) = [h_1(x_i), ··· , h_D(x_i)]
  end for
  for i = 1, ··· , n do
    h̄(x_i) = (1/(k+1))h(x_i) + Σ_{j=1}^{k} h(x_j) where x_j ∈ N_dist(x_i, k)
  end for
  for i = 1, ··· , n do
    N_ang(x_i, k̂) = Find_nn_χ²(H_n, h̄(x_i), k̂)
    [H_n is the collection of the mean histogram for each point of the dataset X_n. The
    function Find_nn_χ² is the analogue of Find_nn_euclid using χ² histogram distance
    instead of the Euclidean distance.]
    N_dist(x_i, k̂) = Find_nn_euclid(X_n, x_i, k̂)
    N_final(x_i, k) = Borda_count(N_dist(x_i, k̂), N_ang(x_i, k̂), k)
    [The Borda_count function is the method Borda Count as described in this section].
  end for
```

Table 1. Brief description of the synthetic datasets employed in our experiments, where d is the intrinsic dimension and D is the embedding space dimension.

Name	d	D	Description
$\mathcal{M}_2$	3	5	Affine space.
$\mathcal{M}_3$	4	6	Concentrated figure, confusable with a $3d$ one.
$\mathcal{M}_4$	4	8	Nonlinear manifold.
$\mathcal{M}_6$	6	36	Nonlinear manifold.
$\mathcal{M}_7$	2	3	Swiss-Roll.
$\mathcal{M}_8$	12	72	Nonlinear manifold.
$\mathcal{M}_9$	20	20	Affine space.
$\mathcal{M}_{11}$	1	3	Möebius band 10-times twisted.

To perform a proper mixture of distance and angular neighborhoods we fix $\hat{k} > k$, build two new sets $\mathcal{N}_{dist}(\mathbf{x}_i, \hat{k})$ and $\mathcal{N}_{ang}(\mathbf{x}_i, \hat{k})$, as described above, and

use the simple and well known Borda count method to select the k points that are the closest to $\mathbf{x}_i$ with respect to both the euclidean and the angular distance. More precisely, given two sets of size $\hat{k}$, sorted in ascending order, the Borda count method selects the points sharing the top positions between the two indexed sets; to this aim, it assigns the score $\hat{k}$ to the first element of each set, the score $\hat{k}-1$ to the second element, and so on, till the value 1 is assigned to the last element of the sets. The algorithm then sums the scores obtained by each element in the two sets (if an element is not present in a set its score is zero), orders the elements according to the resulting scores, and takes the first k elements. In this way the neighborhood set $\mathcal{N}_{final}(\mathbf{x}_i, k)$ obtained for each point $\mathbf{x}_i$ is a combination of the two sets $\mathcal{N}_{dist}(\mathbf{x}_i, \hat{k})$ and $\mathcal{N}_{ang}(\mathbf{x}_i, \hat{k})$.

In the next section, experiments on synthetic and real datasets show that the algorithm `ONeS` builds local neighborhoods preserving the proximity relations between points and being less affected by the manifold curvature.

3 Experimental Results and Future Works

In order to assess the quality of `ONeS`, we compare the neighborhood sets it computes with the commonly used k-neighborhoods, obtained by employing the euclidean distance.

The comparison employs a measure, which we call residual, that indicates how much the point neighborhoods are affected by the manifold's folds generated by the embedding. Under the assumption that the manifold has `id` d, once a neighborhood is computed, its residual is obtained by building its local d-dimensional tangent space and calculating the mean distance between each neighborhood point and its projection on the tangent space, normalized by the neighborhood radius (i.e. the distance of the farthest point from the center). The residual for a given dataset is the average residual over all the neighborhoods.

Table 2. Percentage values of the relative improvement using `ONeS`. Since manifold $\mathcal{M}_9$ has an `id` of 20, we need to fix $k = 22, \hat{k} = 33$ and $k = 22, \hat{k} = 44$, in order to estimate a d-dimensional tangent space having $k > d$ points.

Name	$k = 12, \hat{k} = 18$	$k = 12, \hat{k} = 24$
$\mathcal{M}_2$	62%	57%
$\mathcal{M}_3$	69%	67%
$\mathcal{M}_4$	29%	23%
$\mathcal{M}_6$	19%	13%
$\mathcal{M}_7$	48%	32%
$\mathcal{M}_8$	9%	6%
$\mathcal{M}_9^*$	32%*	24%*
$\mathcal{M}_{11}$	47%	42%

Name	$k = 12, \hat{k} = 18$	$k = 12, \hat{k} = 24$
$\texttt{MNIST}_0$	5%	9%
$\texttt{MNIST}_1$	5%	7%
$\texttt{MNIST}_2$	3%	5%
$\texttt{MNIST}_3$	5%	8%
$\texttt{MNIST}_4$	3%	6%
$\texttt{MNIST}_5$	2%	5%
$\texttt{MNIST}_6$	7%	13%
$\texttt{MNIST}_7$	2%	5%
$\texttt{MNIST}_8$	3%	6%
$\texttt{MNIST}_9$	4%	8%

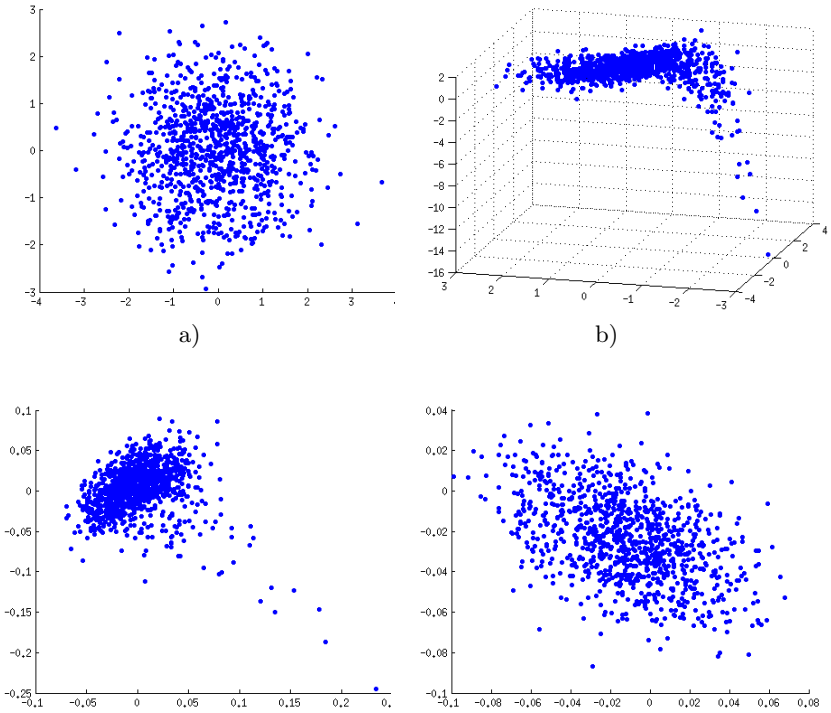

Fig. 2. a) Two dimensional dataset composed by 1000 points b) Non linear embedding in three dimensions c) Reduced dataset obtained by `LTSA` using Euclidean distance based neighborhood sets ($k = 12$) d) Reduced dataset obtained by `LTSA` using neighborhood sets calculated by `ONeS` ($k = 12, \hat{k} = 60$)

In order to perform experiments on datasets composed by points sampled from manifolds of both low and high `id`, linearly and nonlinearly embedded in an higher dimensional space (see Table 1), we use the datasets generator proposed in [6]. Besides, we test `ONeS` on the digit test set contained in the standard `MNIST` database, which contains 784-dimensional points, and has an estimated `i.d.` in the range $[8, \cdots, 11]$ [6]. In Table 2 we report the results obtained by fixing $k = 12$, a commonly used value in the literature, and $\hat{k} = 1.5 * k$, $\hat{k} = 2 * k$. The results are expressed as percentage values, which show the relative improvement in the reduction of the residual. Precisely, being R the residual estimated using the k-neighborhoods, and $\tilde{R}$ the residual estimated using `ONeS`, the percentage is obtained as $100 - \frac{\tilde{R}}{R} * 100$.

The reported results show the enhanced quality of the neighborhood set computed by `OnES` both on the synthetic and on the real datasets, the latter being noisy and sparse. It is notable to observe that neighborhoods built on noisy and sparse dataset may benefit by using higher values of $\hat{k}$, that is by considering more candidate neighbors. s a further visual experiment and example, we gen-

erate a two dimensional dataset (Figure 2.a) embedded in a three dimensional space (Figure 2.b), and having folds characterized by different curvatures. On this dataset we test `LTSA` using two different neighborhood sets, the set built by the k-neighborhoods employing the euclidean distances, and the sets built by `ONeS`. The neighborhood sets obtained by `ONeS` allow `LTSA` to obtain a better preservation of the original data structure.

Future works will be aimed at the experimental analysis of the relation between the parameters k, $\hat{k}$, the dimensionality of the given dataset, its cardinality, and the `id` of the manifold from which the dataset points are sampled. Besides, we are currently searching for an evaluation measure that could be more general than the residual, and different techniques to select the final neighborhood set since we are aware that the Borda count method is based on an independency assumption which is not guaranteed to be true. Further efforts will be devoted to the development of a `dr` technique using the neighborhood sets calculated by `ONeS`, and its comparison with state-of-the-art `dr` techniques, which use an adaptive neighborhood selection. Particularly, we will focus on the improvement of both the `dr` accuracy and its computational efficiency.

References

1. Ceruti, C., Bassis, S., Rozza, A., Lombardi, G., Casiraghi, E., Campadelli, P.: DANCo: an intrinsic Dimensionalty estimator exploiting Angle and Norm Concentration. Pattern Recognition **47**(8), 2569–2581 (2014)
2. Costa, J.A., Hero, A.O.: Learning intrinsic dimension and entropy of high-dimensional shape spaces. In: Proc. of EUSIPCO, pp. 231–252 (2004)
3. Demartines, P., Herault, J.: Curvilinear component analysis: A self-organizing neural network for nonlinear mapping in cluster analysis. IEEE Trans. on Neural Networks **8**(1), 148–154 (1997)
4. Gashler, M., Martinez, T.: Robust manifold learning with cyclecut. Connection Science, **24**(1) (2012)
5. Gashler, M., Martinez, T.: Tangent space guided intelligent neighbor finding, pp. 2617–2624 (2011)
6. Hein, M., Audibert, J.Y.: Intrinsic dimensionality estimation of submanifolds in euclidean space. In: Proc. of ICML, pp. 289–296 (2005)
7. Jing, L., Shao, C.: Selection of the suitable parameter value for isomap. Journal of Software **6**(6), 1034–1041 (2011)
8. Jollife, I.T.: Principal component analysis. Springer Series in Statistics. Springer-Verlag, New York (1986)
9. Kruskal, J.B.: Linear transformation of multivariate data to reveal clustering, vol. I. Ac. Press (1972)
10. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: Proceedings of NIPS, vol. 1, pp. 777–784 (2004)
11. Little, A.V., Maggioni, M., Rosasco, L.: Multiscale Geometric Methods for Data Sets I: Multiscale SVD, Noise and Curvature. MIT-CSAIL-TR-2012-029 (2012)
12. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science **290**, 2323–2326 (2000)

13. Rozza, A., Lombardi, G., Ceruti, C., Casiraghi, E., Campadelli, P.: Novel high intrinsic dimensionality estimators. Machine Learning Journal **89**(1–2), 37–65 (2012)
14. Shepard, R.N., Carroll, J.D.: Parametric representation of nonlinear data structures. Ac. Press (1969)
15. Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In: Advances in Neural Information Processing Systems, vol. 15, pp. 705–712 (2002)
16. Tenenbaum, J., Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
17. Wang, J.: Improve local tangent space alignment using various dimensional local coordinates. Neurocomputing **71**(1618), 3575–3581 (2008)
18. Wang, J., Jiang, W., Gou, J.: Extended local tangent space alignment for classification. Neurocomputing **77**(1), 261–266 (2012)
19. Wei, J., Peng, H., Lin, Y.S., Huang, Z.M., Wang, J.B.: Adaptive neighborhood selection for manifold learning. In: International Conference on Machine Learning and Cybernetics, pp. 380–384 (2008)
20. Zhan, Y., Yin, J.: Robust local tangent space alignment via iterative weighted PCA. Neurocomputing **74**(11), 1985–1993 (2011)
21. Zhang, P., Qiao, H., Zhang, B.: An improved local tangent space alignment method for manifold learning. Pattern Recognition Letters **32**(2), 181–189 (2011)
22. Zhang, Z., Wang, J., Zha, H.: Adaptive Manifold Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(2), 253–265 (2012)
23. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimension reduction via local tangent space alignment. SIAM Journal of Scientific Computing **26**, 313–338 (2002)