RESEARCH ARTICLE

Novel Online Dimensionality Reduction Method with Improved Topology Representing and Radial Basis Function Networks

Shengqiao Ni, Jiancheng Lv*, Zhehao Cheng, Mao Li

Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, P. R. China

* lvjiancheng@scu.edu.cn

Citation: Ni S, Lv J, Cheng Z, Li M (2015) Novel Online Dimensionality Reduction Method with Improved Topology Representing and Radial Basis Function Networks. PLoS ONE 10(7): e0131631. doi:10.1371/journal.pone.0131631

Editor: Irene Sendina-Nadal, Universidad Rey Juan Carlos, SPAIN

Received: February 26, 2015

Accepted: June 5, 2015

Published: July 10, 2015

Data Availability Statement: All datasets in my study are freely available. You can obtain datasets from the Supporting Information files.

Funding: This work was supported by National Science Foundation of China (http://www.nsfc.gov.cn/) under grant 61375065 and 61432014 and National Program on Key Basic 307 Research Project (973 Program) (http://program.most.gov.cn/) under grant 2011CB302201.

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

Abstract

This paper presents improvements to the conventional Topology Representing Network to build more appropriate topology relationships. Based on this improved Topology Representing Network, we propose a novel method for online dimensionality reduction that integrates the improved Topology Representing Network and Radial Basis Function Network. This method can find meaningful low-dimensional feature structures embedded in high-dimensional original data space, process nonlinear embedded manifolds, and map the new data online. Furthermore, this method can deal with large datasets for the benefit of improved Topology Representing Network. Experiments illustrate the effectiveness of the proposed method.

Introduction

Techniques for dimensionality reduction have attracted much attention in many fields such as machine learning and data mining [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]. Dimensionality reduction methods are used for mapping high-dimensional observations into desired low-dimensional space while preserving the features hidden in the original space. Over the past decades, a number of dimensionality reduction methods have been proposed. Principal Component Analysis (PCA) [11] [12] [13] [14] [15] [16] [17] [18] and Multidimensional Scaling (MDS) [19] [20] [21] have been the two most popular methods because of their relative simplicity and effectiveness. However, PCA is designed to operate when the manifold is embedded linearly or almost linearly in the subspace, and it cannot project previously “unseen” patterns. Classical MDS finds a low-dimensional embedding of patterns with distances in the target space that reflects dissimilarities in the original sample. Both PCA and MDS cannot disclose nonlinearly embedded manifolds because they operate on Euclidean distances. To overcome this limitation, many nonlinear methods have been proposed. Locally Linear Embedding (LLE) [22] maps high-dimensional original data feature space into a single global coordinate system of low dimensionality. Laplacian Eigenmap [23] uses spectral techniques to perform dimensionality reduction. ISOMAP [24] [25] employs classical MDS for geodesic distances in the original data feature space. L-ISOMAP [26] increases ISOMAP's efficiency. It approximates a large global computation in ISOMAP by a much smaller set of calculations.

Because geodesic distances are especially suitable for computing distances among data points embedded in nonlinear manifolds, many methods to build graphs on the data have been proposed. The Topology Representing Network (TRN) [27] [28] [29] [30] is representative because of its effectiveness and simplicity. TRN, which combines the neural gas (NG) vector quantization method with the competitive Hebbian learning rule is used to quantize embedded manifolds and learn the topological relations of the input space without the necessity of prespecifying a topological graph. There are some dimensionality reduction methods based on TRN. Online data visualization using the neural gas network (OVI-NG) [31] is a distance preserving mapping of the codebook vectors (vector quantization) obtained by the NG algorithm. The codebook positions (codebook vectors' projection in low-dimensional space) are adjusted in a continuous output space using an adaptation rule that minimizes a cost function that favors local distance preservation. OVI-NG is not able to disclose nonlinear embedded manifolds because of its use of Euclidean distances. The Geodesic Nonlinear Projection Neural Gas (GNLP-NG) algorithm [32] is an extension of OVI-NG that uses geodesic distances instead of Euclidean distances so that GNLP-NG performs well in the projection of nonlinear embedded manifolds. GNLP-NG and OVI-NG are not able to project new data. The method RBF-NDR [33], which includes the NG algorithm and RBFN, can process data online. Nonetheless, RBF-NDR sometimes has poor mapping quality and sometimes performs well due to minimizing STRESS [33] at each iteration without clear targets.

In this paper, we propose a new method for online and nonlinear dimensionality reduction called ITRN-RBF. We improve the conventional TRN so that it builds a more appropriate topology relationship. That is, the method we call the Improved TRN (ITRN) is more specifically suited to calculating geodesic distances. Furthermore, large amounts of data can be processed by ITRN's vector quantization. We chose the MDS method as the mapping method. In contrast to classical MDS operating on Euclidean distances, our method operates on the geodesic distances of the topology graph reconstructed by ITRN. The mapping between the original high-dimensional space and low-dimensional feature structures embedded is then learned by supervised RBFN, whose target values are generated by the mapping methods. In particular, we give two implementations of RBFN. One is trained by the Widrow-Hoff learning algorithm. The other is an exact RBFN designed by precise mathematical calculation. Finally, the RBFN is used to reduce the dimensions of the original high-dimensional data. ITRN-RBF can process nonlinearly embedded manifolds, preserve the global structure of these manifolds, and project new data online.

Methods

ITRN-RBF comprises two procedures: capturing the topology of the given dataset using ITRN and learning the mapping using RBF. The first procedure learns the topology of the input data embedded in the high-dimensional original data feature space and generates a graph using ITRN. ITRN connects the subgraphs together to ensure the connectivity of the resulting graph. The method for connecting the subgraphs is discussed in the section below. Using the output (codebook vectors with similarity relationships) from the first procedure, the second procedure calculates the pairwise graph distances as geodesic distances and constructs the mapping between the high-dimensional original space and low-dimensional target space. It then uses RBFN to learn this mapping. In particular, there are variety of ways to implement RBFN. We give two different implementations, which are described below. Finally, RBFN is just the dimensionality reduction tool, which has the desired capabilities of processing nonlinearly embedded manifolds and projecting new data online. In the following, ITRN-RBF is introduced and discussed in detail.

ITRN

TRN is one of the vector quantization algorithms that are based on neural network models, which are capable of adaptively quantizing a given set of input data. Given a set of data $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_j \in R^D$, TRN employs a finite set $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, $\mathbf{v}_i \in R^D$ called codebook vectors (or reference vectors, neural units) to encode X. TRN learns the topological relation of X by distributing nodes among the data and connecting them using the competitive Hebbian rule. The purpose of TRN's learning is to reconstruct a topology graph $G = (V, C)$ for X, where C represents the adjacent matrix of V, whose values are constrained to 0 (unconnected) or 1 (connected). The conventional TRN algorithm operates as follows.

1. Set iteration step $t = 0$. Assign initial values to the codebook vectors $\mathbf{v}_i(\mathbf{v}_i \in V, i = 1, 2, \ldots, n)$ and set all connection edges.
2. Randomly select input pattern $\mathbf{x}$ from X.
3. For each codebook vector $\mathbf{v}_i$, calculate rank r_i by determining the sequence $(i_0, i_1, \ldots, i_{n-1})$ by
$$\|\mathbf{x} - \mathbf{v}_{i_0}\| < \|\mathbf{x} - \mathbf{v}_{i_1}\| < \cdots < \|\mathbf{x} - \mathbf{v}_{i_{n-1}}\|. \tag{1}$$
That is, $r_{i_0} = 0, r_{i_1} = 1, \ldots, r_{i_{n-1}} = n{-}1$.
4. Update all nodes $\mathbf{v}_i$ according to
$$\mathbf{v}_i^{new} = \mathbf{v}_i^{old} + \epsilon \cdot e^{-r_i/\lambda(\mathbf{x}-\mathbf{v}_i^{old})}. \tag{2}$$
5. Connect the two nodes closest to the randomly selected input pattern $\mathbf{x}$. Set $c_{i_0 i1} = 1$ and set this connection's age to zero ($t_{i_0 i1} = 0$).
6. Increase the age of all connections of $\mathbf{v}_{i_0}$ by setting $t_{i_0 j} = t_{i_0 j} + 1$ for all nodes $\mathbf{v}_j$ that are connected to node $\mathbf{v}_{i_0}$ ($c_{i_0 j} = 1$).
7. Remove the connections of node $\mathbf{v}_{i_0}$ that have exceeded their lifetime by setting $c_{i_0 j} = 0$ for all j with $c_{i_0 j} = 1$ and $t_{i_0 j} > T$.
8. Increase the iteration step $t = t + 1$. If the maximum number of iterations has not yet been reached ($t < t_{max}$), continue with step 2.

There are many parameters in this algorithm. The codebook vectors' number n and maximum number of iterations t_{max} are both set by the user. The parameter λ, step size ϵ and lifetime T depend on the number of iterations. The time dependent parameters are set according to the form
$$g(t) = g_i \left(\frac{g_f}{g_i}\right)^{\frac{t}{t_{max}}}. \tag{3}$$
Here, g_i is the initial value of the variable, g_f is the final value, t denotes the iteration step and t_{max} represents the maximum number of iterations. Suggestions as to how to tune these parameters have been proposed by Martinetz and Schulten [27].

In fact, to obtain a denser graph that is better for calculating geodesic distances, we implement some improvements. For the randomly selected input patterns at each iteration, the method ITRN creates a connection between the 1_{st} and $(k + 1)_{th}$ nearest nodes ($1 \leqslant k \leqslant kn$, typically $kn \in \{2, 3, 4\}$) instead of only connecting the first and second closest codebook vectors. In addition, we also connect the subgraphs to avoid the existence of infeasible nodes. Specific details about ITRN are presented in the statements below. Steps 1–5 are the same as steps 1–5 in the conventional TRN, hence we only list the steps that follow.

6. If the following condition is satisfied

$$\|\mathbf{v}_{i_s} - \mathbf{v}_{i_k}\| = \min\left(\|\mathbf{v}_{i_0} - \mathbf{v}_{i_k}\|, \|\mathbf{v}_{i_1} - \mathbf{v}_{i_k}\|, \ldots, \|\mathbf{v}_{i_{k-1}} - \mathbf{v}_{i_k}\|\right) \tag{4}$$

for $k = 1, 2, \ldots, kn$, then create a connection between nodes $\mathbf{v}_{i_s}$ *and* $\mathbf{v}_{i_k}$ by setting $c_{i_s ik} = 1$ and $t_{i_s ik} = 0$.

7. Increase the age of all connections of $\mathbf{v}_l(l = i_0, i_1, \ldots, i_{kn-1})$ by setting $t_{lj} = t_{lj} + 1$ for all nodes $\mathbf{v}_j$ that are connected to node $\mathbf{v}_{i_0}$ ($c_{i_0 j} = 1$).

8. Remove the connections of node $\mathbf{v}_l(l = i_0, i_1, \ldots, i_{kn-1})$ that have exceeded their lifetime by setting $c_{lj} = 0$ for all j for which $c_{lj} = 1$ and $t_{lj} > T$.

9. Increase the iteration step:$t = t + 1$. If the maximum number of iterations has not yet been reached ($t < t_{max}$), continue with step 2.

10. If the resulting graph $G = (V, C)$ is unconnected, it is necessary to connect the subgraphs. Assume that $G = \{G_1, G_2, \ldots, G_c\}$, where G_i is the subgraph that is not connected to the others. Calculate $E = \mathbf{e}_{ij}$, where e_{ij} is the shortest edge obtained by connecting the closest nodes in G_i and G_j. Finally, choose a suitable $\mathbf{e}_{ij}$ to add to C and obtain the connected graph $G^E = (V, C^E)$.

Compared with conventional TRN, we note that:

- ITRN modifies the TRN strategy to establish the connections in steps 6–8 (see Fig 1) and connect subgraphs in step 10 (see Fig 2).
- Conventional TRN causes deviation because it ignores some topological relations of the codebook vectors. However, ITRN connects multiple points so that more topological relations can be established. In addition, a relation caused by miscalculation will be removed when its lifetime exceeds the limit. An experiment shows the different construction, as shown in Fig 3.
- The distance ratio defined as follows:

$$ratio = \frac{GD_{ij}}{ED_{ij}} \tag{5}$$

can be used to quantitatively evaluate the connection quality. Where GD_{ij} denotes the geodesic distance and ED_{ij} denotes the Euclidean distance between codebook vectors $\mathbf{v}_i$ and $\mathbf{v}_j$. The bar chart shown in Fig 4 displays the statistical results (the x-axis is distance ratio interval and y-axis is the node number). Fig 4a and 4b are based on the dataset that is shown in Fig 2. Fig 4c and 4d use the Swiss roll dataset (shown in Fig 5a). The ITRN's bar chart has a larger gradient and much more restricted *ratio* range, both of which are desirable.

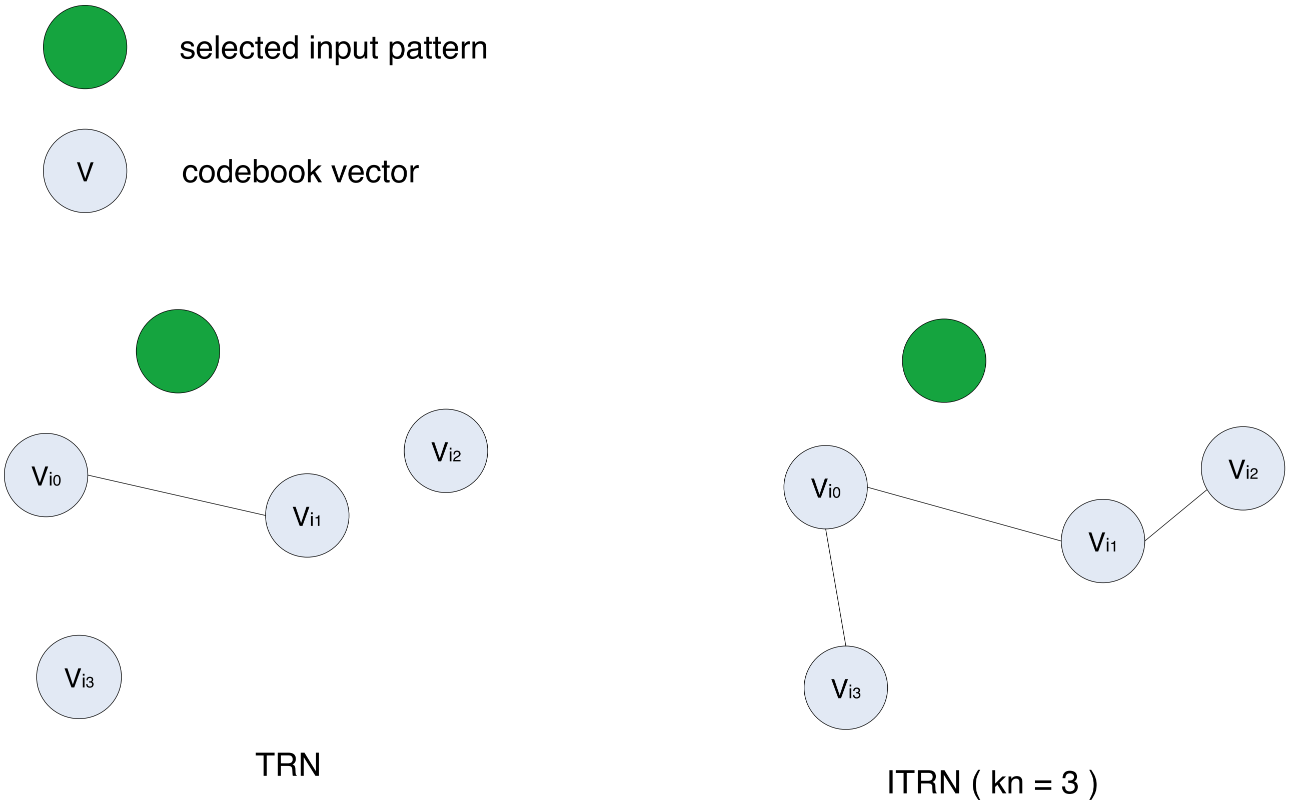

Fig 1. Different strategies to establish connections.

doi:10.1371/journal.pone.0131631.g001

RBFN

In this section, we propose two methods to train or design an RBFN. The first approach, called the training RBFN (TRBF), is a D-h-d network that includes an input layer with D units (equal to the codebook vectors' dimensionality), hidden layers with h units (set by users), and an output layer with d units (equal to the dimensionality of the output space). The second approach, named exact RBFN (ERBF), is a D-n-d network with the same parameters as the training RBFN. Especially the number of hidden layer units n is equal to the number of codebook vectors. All of them have the same codebook vector input s obtained by ITRN and the same training targets given by MDS. What is more important, MDS is based on geodesic distances that are calculated from the graph $G^E = (V, C^E)$ and the training targets defined as $T = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$, $\mathbf{t}_i \in R^d$ are certain, so we can obtain a stable RBFN. For more details, the interested reader can refer to [34] [35] [36] [37] [38] [39].

TRBF. In terms of TRBF, we chose a Gaussian function as the activation function, defined as follows:

$$\phi_i(\mathbf{x}_j) = e^{-\frac{1}{2}\frac{\|\mathbf{x}_j-\mathbf{c}_i\|^2}{\sigma_i^2}} = e^{-\frac{1}{2}\sum_{l=1}^{D}\frac{(x_{lj}-c_{li})^2}{\sigma_{li}^2}}. \quad (6)$$

The hidden layer output is defined as

$$H = \{\mathbf{h}_1, \mathbf{h}_1, \ldots, \mathbf{h}_h\}, \mathbf{h}_{ij} = \phi_i(\mathbf{v}_j), i \in [1, h], j \in [1, n]. \quad (7)$$

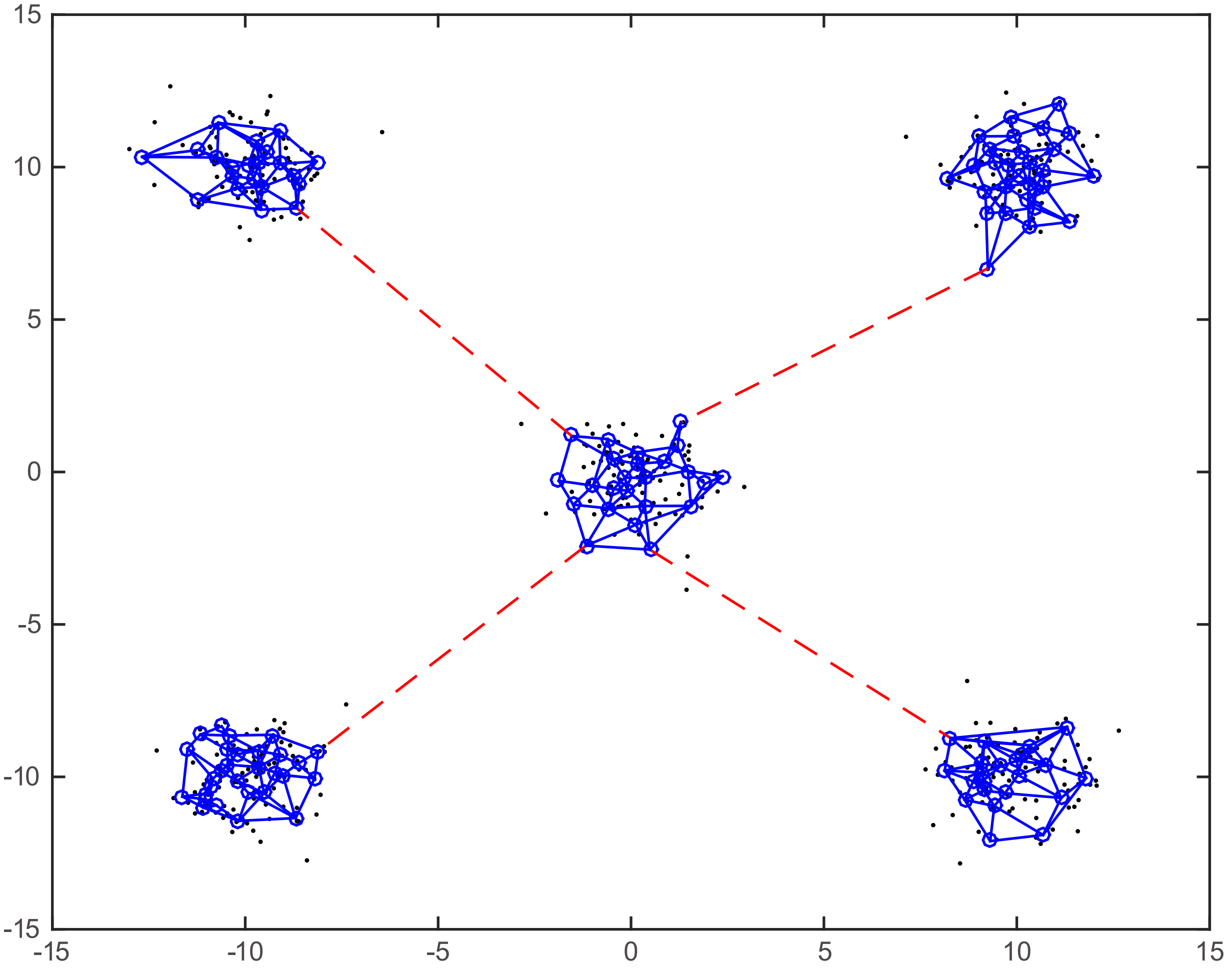

Fig 2. Connecting the subgraphs in ITRN step 10. The dataset is formed of randomly generated nodes comprising five non-overlapping clusters (S1 Dataset). Black dots indicate the training patterns (500 nodes), and blue circles indicate the codebook vectors (100 vectors). In addition, the blue solid lines are established by ITRN steps 1–9 and the dotted lines are established by ITRN step 10.

doi:10.1371/journal.pone.0131631.g002

In addition, the loss function is given by

$$E_j = \frac{1}{2}\mathbf{e}_j^2 = \frac{1}{2}\|\mathbf{t}_j - \mathbf{y}_j\|^2 = \frac{1}{2}\sum_{k=1}^{d}(t_{kj} - y_{kj})^2. \tag{8}$$

The TRBF network provides four types of adjustable parameters: center c_{li}, widths σ_{li}, weights w_{ik} and bias $\mathbf{b}_k$. Based on the Widrow-Hoff learning algorithm, the calculation equations of each parameter are given by:

$$c_{li} = c_{li} + \eta_c \sum_{k=1}^{d}(t_{kj} - y_{kj})\frac{w_{ik}}{\sigma_{li}^2}\phi_i(\mathbf{x}_j)(x_{lj} - c_{li}), l \in [1, D], i \in [1, h], \tag{9}$$

$$\sigma_{li} = \sigma_{li} + \eta_\sigma \sum_{k=1}^{d}(t_{kj} - y_{kj})\frac{w_{ik}}{\sigma_{li}^3}\phi_i(\mathbf{x}_j)(x_{lj} - c_{li})^2, l \in [1, D], i \in [1, h], \tag{10}$$

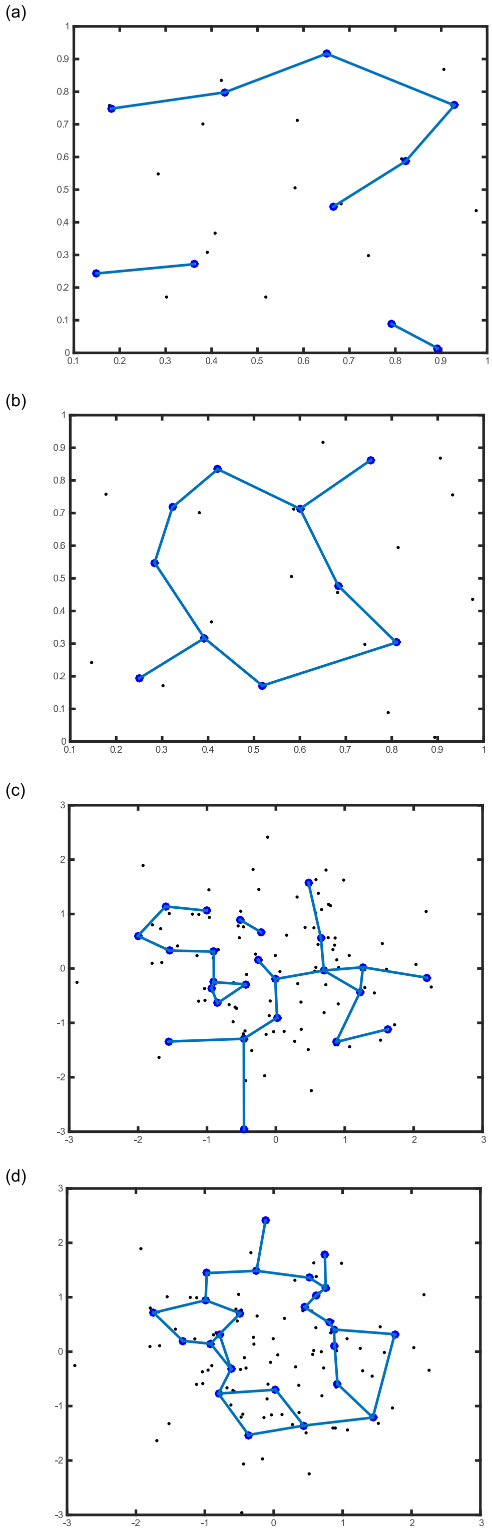

Fig 3. Comparison of TRN and ITRN. Black dots indicate the training patterns, and blue circles indicate codebook vectors. In the first experiment, 20 randomly generated training patterns (S1 Dataset) and 10 codebooks were selected, and (a) and (b) show the results generated by TRN and ITRN, respectively. In the second experiment, 100 randomly generated training patterns (S1 Dataset) and 25 codebooks were selected, and (c) and (d) show the results generated by TRN and ITRN, respectively.

doi:10.1371/journal.pone.0131631.g003

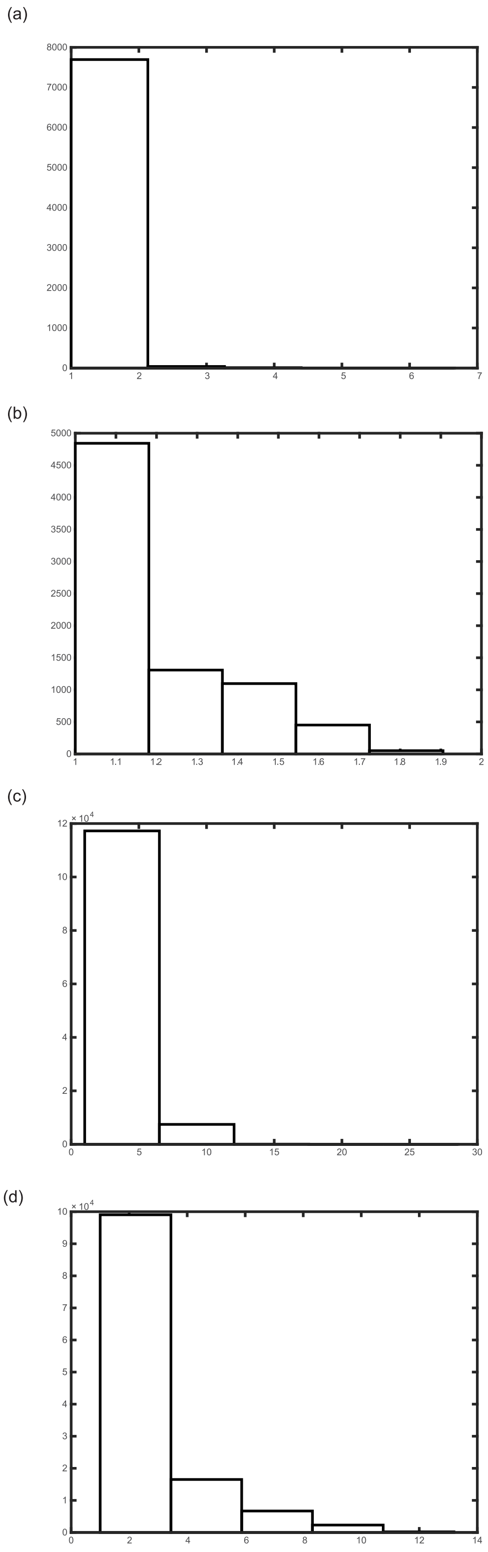

Fig 4. Comparison of distance ratio. (a) and (b) show the ratios for TRN and ITRN, respectively, calculated with an artificial point set, and (c) and (d) show the ratio s for TRN and ITRN, respectively, calculated with a Swiss roll dataset.

doi:10.1371/journal.pone.0131631.g004

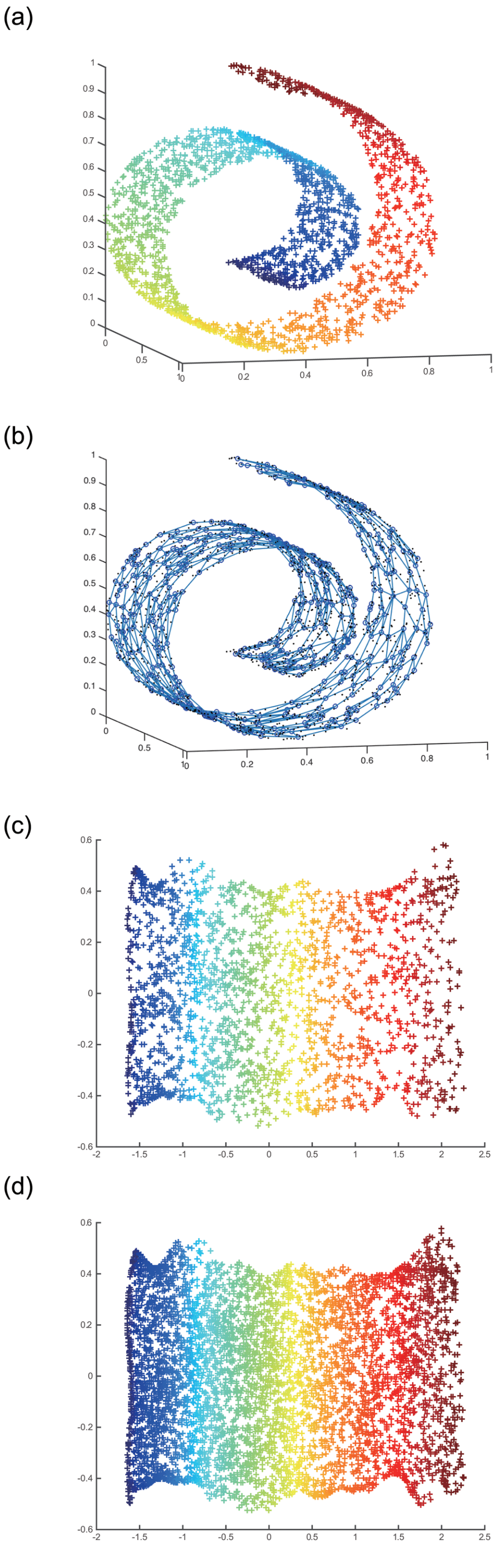

Fig 5. ITRN-ERBF results for Swiss roll. (a) shows Swiss roll dataset, (b) shows learing result by ITRN, (c) shows mapping of the training patterns, (d) shows mapping of the new dataset.

doi:10.1371/journal.pone.0131631.g005

$$w_{ik} = w_{ik} + \eta_w(t_{kj} - y_{kj})e^{-\frac{1}{2}\sum_{l=1}^{D}\frac{(x_{lj}-c_{li})^2}{\sigma_{li}^2}}, i \in [1, h], k \in [1, d] \tag{11}$$

$$\mathbf{b}_k = \mathbf{b}_k + \eta_b(t_{kj} - y_{kj}), k \in [1, d], \tag{12}$$

where η_c, η_σ, η_w, and η_b which are individual step sizes for c_{li}, σ_{li}, w_{ik}, and $\mathbf{b}_k$, respectively, can be defined by users.

ERBF. ERBF's weight W and output layer bias B are obtained by mathematical calculation, so the RBFN can ensure zero error, in theory. The linear equations are given as follows:

$$\{W, B\} \cdot \{H, \mathbf{ones}\}^T = T. \tag{13}$$

The input layer bias b_{in} is set as $\sqrt[2]{-log0.5}/spread$ so there is only one parameter that needs to be set by users. How to set the *spread* is described in the results section.

ITRN-RBF method

The detailed algorithm process is as follows:

1. Construct graph $G^E = (V, C^E)$ using ITRN. In reality, the graph is connected.
2. Calculate the geodesic distances on G^E.
3. Construct the mapping between the high-dimensional original space and low-dimensional target space by using MDS operating on the geodesic distances of the topology graph. For every $\mathbf{v}_j$, we get output $\mathbf{t}_j$ as an expectation.
4. Train or design an RBFN with explicit inputs V and outputs T. In this step, any appropriate RBFNs such as ERBF or TRBF could be applied.
5. Use the RBFN to map the dataset.

Results

In this section, ITRN-RBF is used for visualization and feature extraction, and is also compared with others including methods based on TRN and classical dimensionality reduction methods such as ISOMAP, L-ISOMAP and PCA. We also present the computational complexity analysis of the method and a table with running times.

There are many parameters for experimental data. The common parameters of TRN, OVI-NG and GNLP-NG were set as follows: $t_{max} = 20n$, $\epsilon_i = 0.1$, $\epsilon_f = 0.05$, $\lambda_i = 0.05n$, $\lambda_f = 0.01$, $T_i = 0.05n$, and $T_f = n$. The auxiliary parameters of the OVI-NG and GNLP-NG were set as $\alpha_i = 0.3$, $\alpha_f = 0.001$, $\sigma_i = 0.7n$, and $\sigma_f = 5$. The extra parameter for ITRN kn was set to two (for the Swiss roll) or three (for the artificial faces, handwritten digit "2" and UMist faces datasets). The parameters of RBFN in the Swiss roll experiment were set as follows: $\eta_c = 0.03$, $\eta_\sigma = 0.03$, $\eta_w = 0.2$. For the image processing experiments, they were changed to $\eta_c = 0.002$, $\eta_\sigma = 0.002$, and $\eta_w = 0.05$. The ERBF's parameter *spread* can be obtained as follows:

$$spread = max(d_{ij}), \tag{14}$$

where d_{ij} denotes the Euclidean distances between the codebook vectors. The number of neighbors used in the compuations for ISOMAP and L-ISOMAP is set to 12. The number of landmarks used in L-ISOMAP is set to $0.1n$.

Comparison with the methods based on TRN

We chose two standard metrics for mapping quality. They are widely used for analysing dimensionality reduction methods based on TRN.

- Distance preservation: This value evaluates the distance difference between nodes in input space and nodes in output space. We chose the classical MDS [19] [20] and Sammon stress functions [40] to quantify this value. Their expressions are as follows:

$$E_{MDS} = \sum_{i<j}^{n} (d_{ij} - \hat{d}_{ij})^2, \tag{15}$$

$$E_{SM} = \frac{1}{\sum_{i<j}^{n} d_{ij}} \sum_{i<j}^{n} \frac{(d_{ij} - \hat{d}_{ij})^2}{d_{ij}}, \tag{16}$$

  where d_{ij} is the distance between nodes in the original space and $\hat{d}_{ij}$ is the distance between nodes in output space. Moreover, when the mapping method uses geodesic distances, the expression is calculated using geodesic distances. Otherwise, the method uses Euclidean distances.

- Neighborhood preservation: This value evaluates the degree to which adjacent patterns in input space are close in output space. The measures of trustworthiness $M_1(k)$ and continuity $M_2(k)$ [41] [42] are suitable. Their expressions are given below:

$$M_1(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{\mathbf{v}_j \in U_k(\mathbf{v}_i)} (r_{ij} - k), \tag{17}$$

$$M_2(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{\mathbf{v}_j \in V_k(\mathbf{v}_i)} (\hat{r}_{ij} - k), \tag{18}$$

  where $U_k(\mathbf{v}_i)$ is the set of nodes that are in the k-size neighborhood of the codebook vector i in the output space but not in the original space. In contrast, $V_k(\mathbf{v}_i)$ denotes the set of nodes that belong to the k-size neighborhood of codebook vector i in the original space rather than in output space. Rank r_{ij} refers to rank in the original space, but $\hat{r}_{ij}$ denotes the order in output space. In fact, trustworthiness and continuity are functions of the number of neighbors k.

Three methods, OVI-NG, GNLP-NG, and RBF-NDR, were selected for comparison. In particular, OVI-NG and GNLP-NG can only map the codebook vectors. Hence, to keep the comparison fair, we used the RBFN obtained by RBF-NDR and ITRN-RBF to map the codebook for comparison. All methods' line charts with respect to trustworthiness and continuity are given after each experiment, except for OVI-NG, because the method cannot process nonlinear embedded manifolds. (We only show the results separately in the Swiss roll experiment.) Table 1 presents the stress functions for the different methods.

Swiss roll. The Swiss roll (S2 Dataset) corresponds to a two-dimensional pattern distributed uniformly on a plane and embedded nonlinearly in 3D (Fig 5a). We used ITRN to learn this manifold and ensure the connectivity of the resulting graph. The graph given in Fig 5b shows the reconstructed manifold embedded in the high-dimensional original data feature space by ITRN. We then trained an RBFN to reduce the dimensionality. The projection estimated by the ERBF module is given in Fig 5c and 5d. Fig 5c shows the mapping of the training

Table 1. Stress functions for different methods.

Methods	Swiss roll		AF		"2"	
	E_{MDS}	E_{SM}	E_{MDS}	E_{SM}	E_{MDS}	E_{SM}
ITRN-ERBF	2.5204E+03	0.0094	6.2399E+03	0.0125	5.4540E+07	0.9379
ITRN-TRBF	3.6855E+03	0.0153	1.3094E+04	0.0295	4.8321E+07	0.9851
RBF-NDR	2.6040E+03	0.0114	3.1725E+04	0.0812	4.4090E+06	0.0892
GNLP-NG	3.4730E+03	0.0116	2.6486E+04	0.0507	4.6615E+06	0.1064

doi:10.1371/journal.pone.0131631.t001

pattern (2000 nodes), and Fig 5d shows the mapping of the new dataset (5000 nodes) that was taken from the Swiss roll by random sampling. We observe that TRN-RBF is able to recover the intrinsic two-dimensionality of the Swiss roll and process a new dataset.

The different mappings of the Swiss roll's codebook vectors are presented in Fig 6. All methods disclose the embedded manifolds of the Swiss roll except OVI-NG. The neighborhood preservation achieved by OVI-NG is presented in Fig 7. This method shows such a poor performance, only ITRN-RBF, RBF-NDR, and GNLP-NG are discussed in the following. Moreover, for RBF-NDR and GNLP-NG, the purpose of iterative adjustment is to minimize the stress function, hence they have similar mapping structures.

Analyzing each of the measures shown in Fig 8 and Table 1, it is clear that ITRN-ERBF retains two distinct advantages with respect to distance and neighborhood preservation. Closest to ITRN-ERBF in performance is RBF-NDR. Methods GNLP-NG and ITRN-TRBF perform almost as well.

Artificial and real-world images. The artificial images (S3 Dataset) are from the domain of visual perception. The dataset contains 698 artificially generated images of faces (image size: 64 × 64, 688 images for training and 10 for testing, referred to as AFs) under different poses and different illumination conditions.

The real-world images (S4 Dataset) come from the Mixed National Institute of Standards and Technology (MNIST) database. We chose the handwritten digit "2" (image size: 28 × 28, 1000 images for training and 10 for testing, referred to as "2") for this experiment because of its varied forms.

In particular, for the different datasets, there are two treatments: AF are preprocessed by PCA. The principal components that contribute less than 0.1% to the explained variance are discarded, hence dimensionality reduction methods are used for mapping the primed dataset. However, for "2," we chose the original dataset as the training patterns.

ITRN-ERBF and other methods were used for the task of visual perception. The resulting two-dimensional projection of training patterns obtained by ITRN-ERBF is given in Figs 9 and 10. A comparison of the mapping quality is presented in Figs 11 and 12 as well as Table 1. Blue plusses represent the two-dimensional projections of training patterns and red circles represent testing patterns' position. For easy inspection, only part of the training patterns' corresponding images were plotted. The major articulation features of the AF, left-right (x-axis) and up-bottom (y-axis), are captured from the input space. For the "2" dataset, the bottom loop (x-axis) and lean (y-axis) are captured from input space.

In terms of mapping quality, ITRN-ERBF has a high adaptability and performance. In contrast, ITRN-TRBF, GNLP-NG, and RBF-NDR perform less well. In very rare cases, GNLP-NG shows the best distance preservation feature because the goal of GNLP-NG is to minimize the stress function.

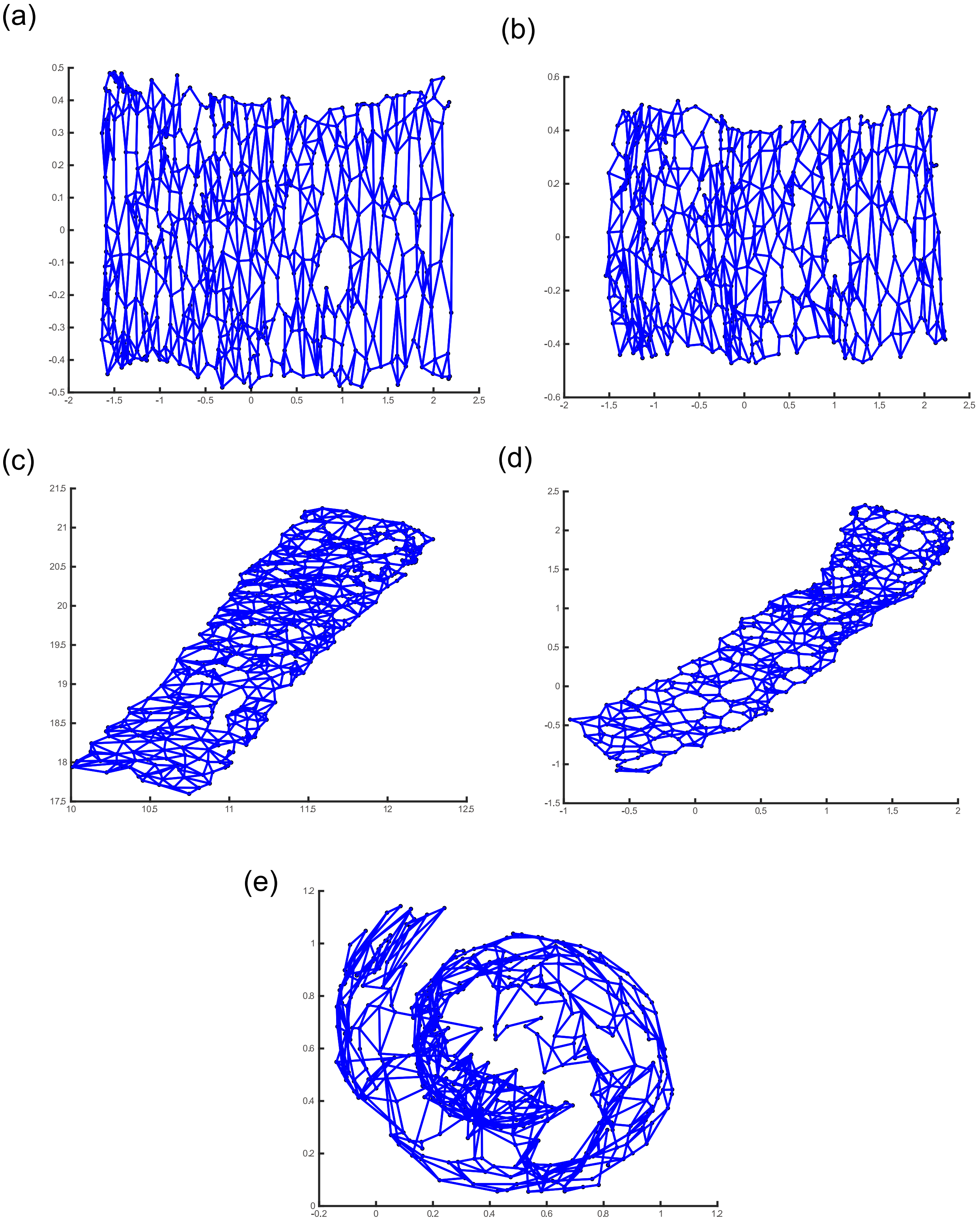

Fig 6. Different mappings of the Swiss roll's codebook vectors for different methods. (a) ITRN-ERBF, (b) ITRN-TRBF, (c) RBF-NDR, (d) GNLP-NG, and (e) OVI-NG.

doi:10.1371/journal.pone.0131631.g006

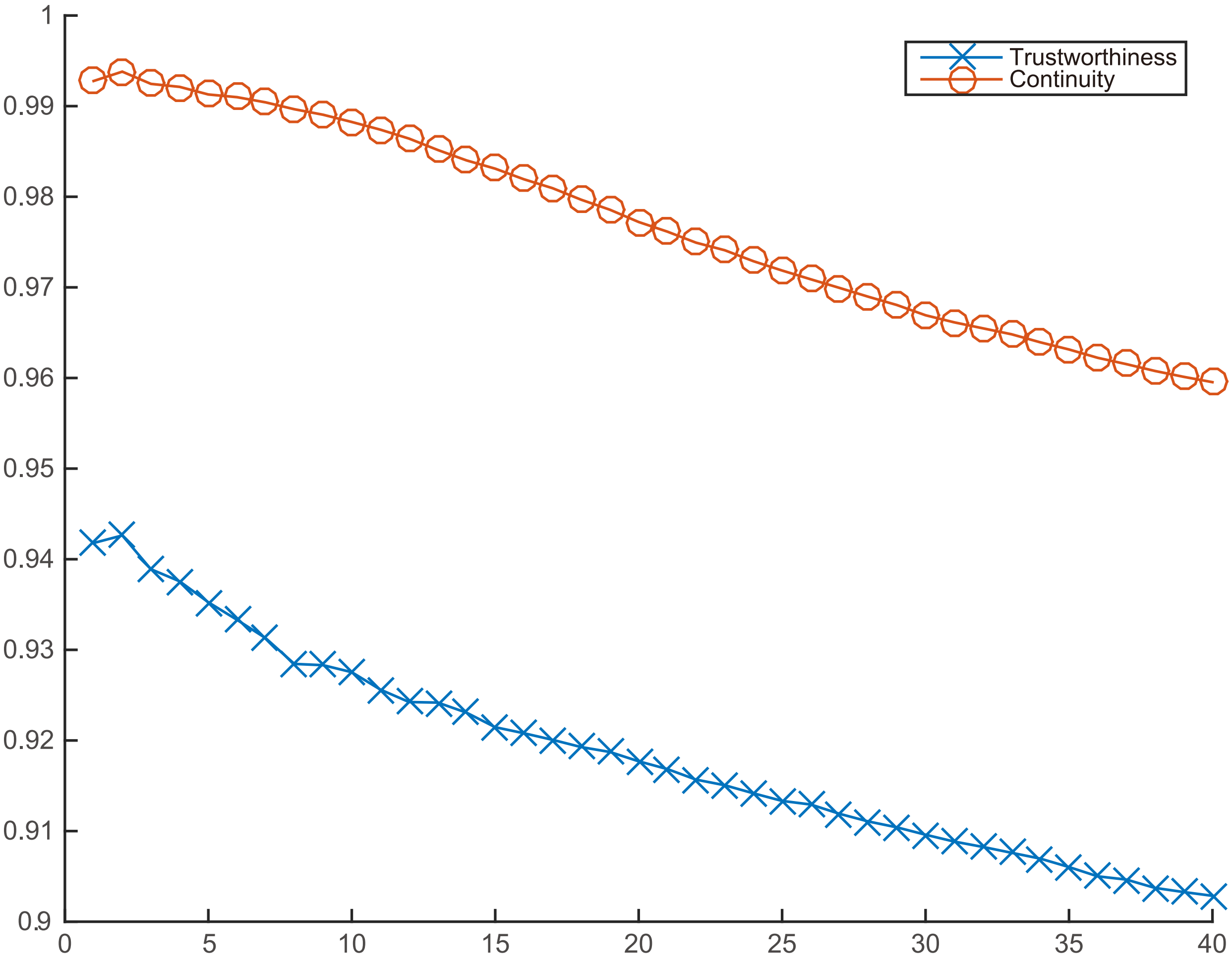

Fig 7. OVI-NG mapping quality for Swiss roll.

doi:10.1371/journal.pone.0131631.g007

Comparison with RBF-NDR. Most dimensionality reduction methods can process new datasets because of RBFN. However, an imprecise RBFN could lead to imprecise projections. Hence, ITRN-ERBF, ITRN-TRBF, and RBF-NDR were selected to determine whether they were able to generate definitive results. All of them use RBFN to project the dataset.

All methods ran 20 times on a uniform Swiss roll dataset. At each iteration, the manifold learning procedure was executed afresh and the RBFN was also designed or trained again. The results are shown in Fig 13. Here, the x-axis denotes the iterations and the y-axis represents the value of E_{MDS} or E_{SM}. We observe that ITRN-ERBF has the smoothest line, indicating that ITR-N-ERBF has the most definitive results. In contrast, ITRN-TRBF and RBF-NDR have obvious fluctuations because of their trained RBFN, which could not minimize the stress function or loss function.

Comparison against the classical methods

In this section, ITRN-RBF was compared with classical dimensionality reduction methods including ISOMAP, L-ISOMAP, PCA. Three quality metrics [43], namely, stress function, the

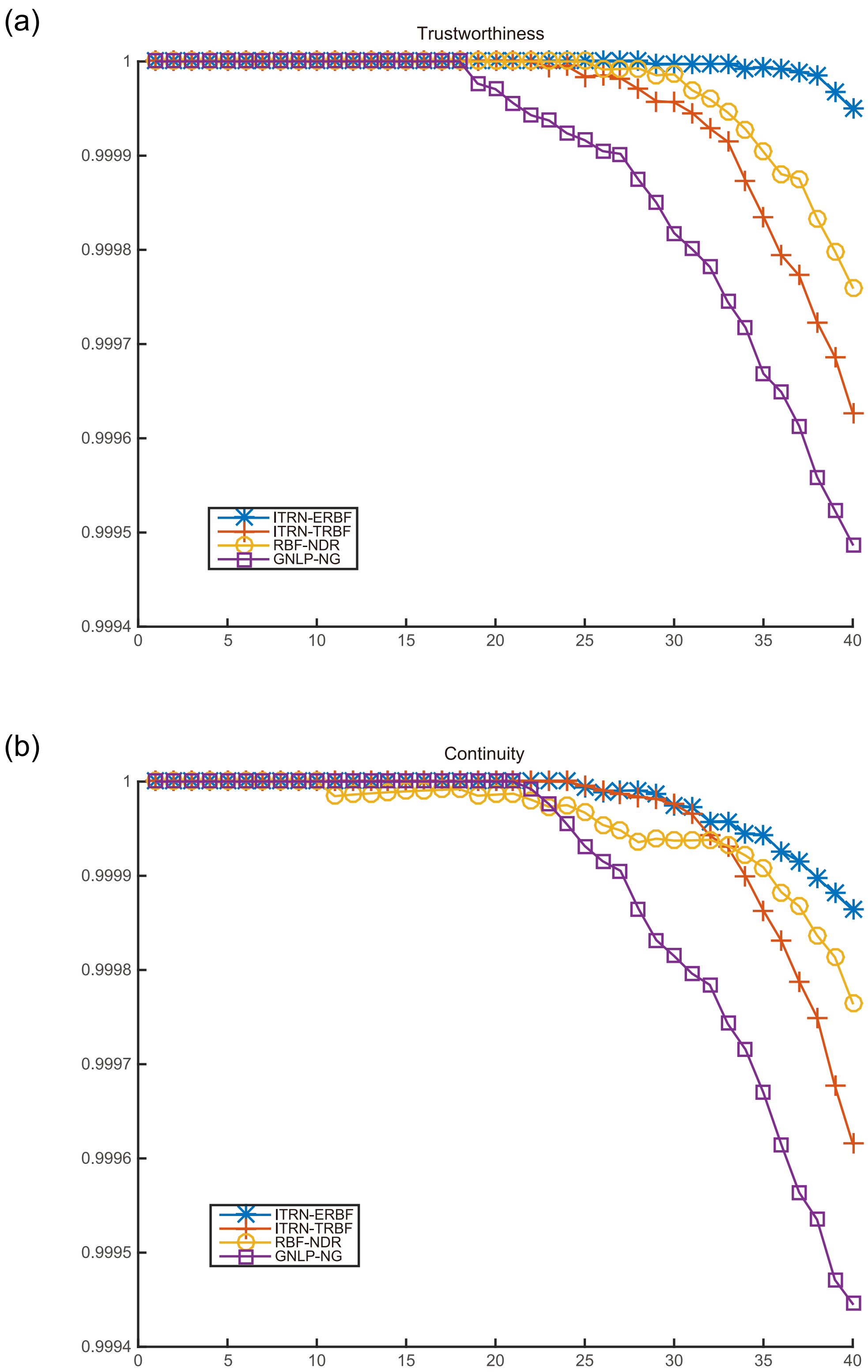

Fig 8. Mapping quality for the Swiss roll.

doi:10.1371/journal.pone.0131631.g008

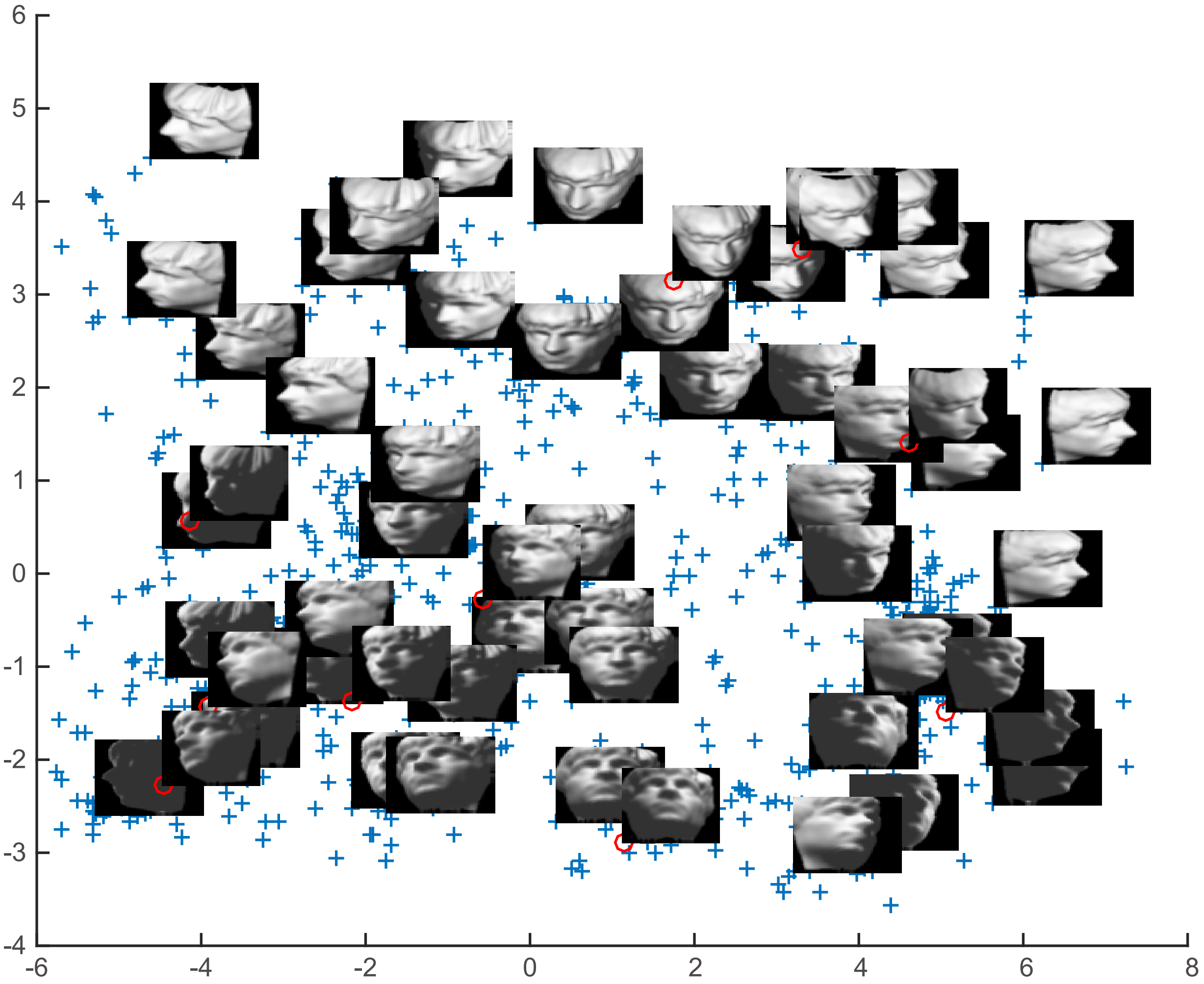

Fig 9. AF results.

doi:10.1371/journal.pone.0131631.g009

correlation coefficient, smooth neighborhood preservation were used for analysis. We detail the three quality metrics in the following.

- Stress function. You can refer to Eq 16.
- Correlation coefficient. This value measures how distances in the original space are correlated to those in the visual space. The expression are as follows:

$$E_{CC} = 1 - \frac{< D \odot \hat{D} > - < D >< \hat{D} >}{\sigma_D \sigma_{\hat{D}}}, \tag{19}$$

  where D and $\hat{D}$are the upper triangular distance metrics before and after projection, $\odot$ is the element-by-element product, $<>$ is the average operator and σ is the standard deviation of the vector's elements. The smaller the value of E_{CC}, the better the performance of the visualization is.

- Smooth neighborhood preservation. This is also a neighborhood preservation metric, but it's based on distance instead of rank order compared with trustworthiness and continuity. The

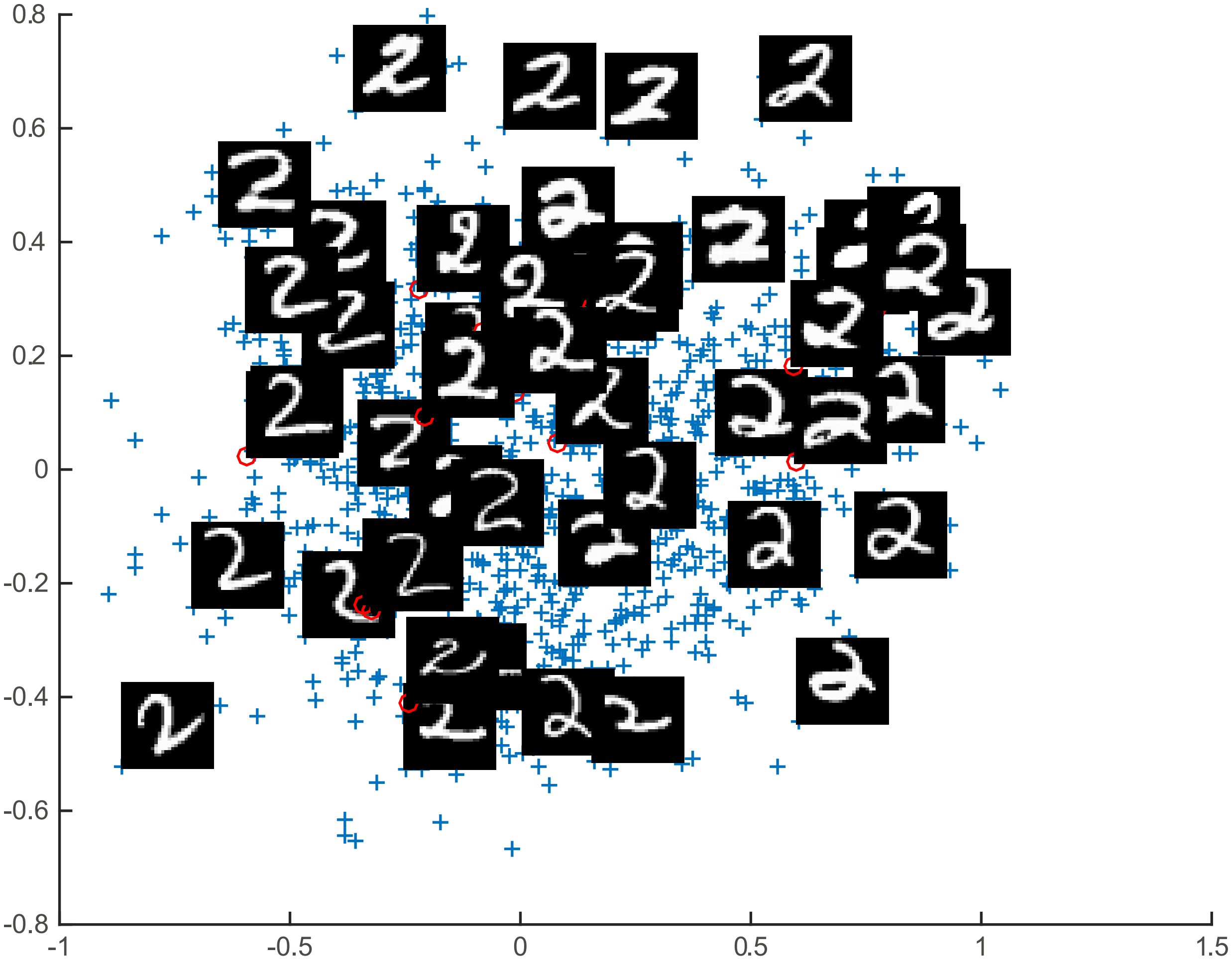

Fig 10. Handwritten digit “2” results.

doi:10.1371/journal.pone.0131631.g010

local misplacing metrics are defined as follows:

$$W^{T}(\mathbf{v}_i) = \begin{cases} \frac{1}{|N^{T}(\mathbf{v}_i)|}\sum_{\mathbf{v}_j \in N^{T}(\mathbf{v}_i)} w(\hat{r}_i, \hat{d}_{ij}) & N^{T}(\mathbf{v}_i) \neq \phi, \\ 0 & others, \end{cases} \tag{20}$$

$$W^{FN}(\mathbf{v}_i) = \begin{cases} \frac{1}{|N^{FN}(\mathbf{v}_i)|}\sum_{\mathbf{v}_j \in N^{FN}(\mathbf{v}_i)} w(r_i, d_{ij}) & N^{FN}(\mathbf{v}_i) \neq \phi, \\ 0 & others, \end{cases} \tag{21}$$

where $N^{T}(\mathbf{v}_i)$ is the set of nodes in the k-nearest neighborhood (we set $k = 12$ for this analysis) of an node i that are not mapped among the k-nearest neighbors of i in the output space and $N^{FN}(\mathbf{v}_i)$ is the set of nodes that are not among the k-nearest neighbors of i but are mapped among the k-nearest neighbors of i in the output space, $|W|$ is the number of elements in the

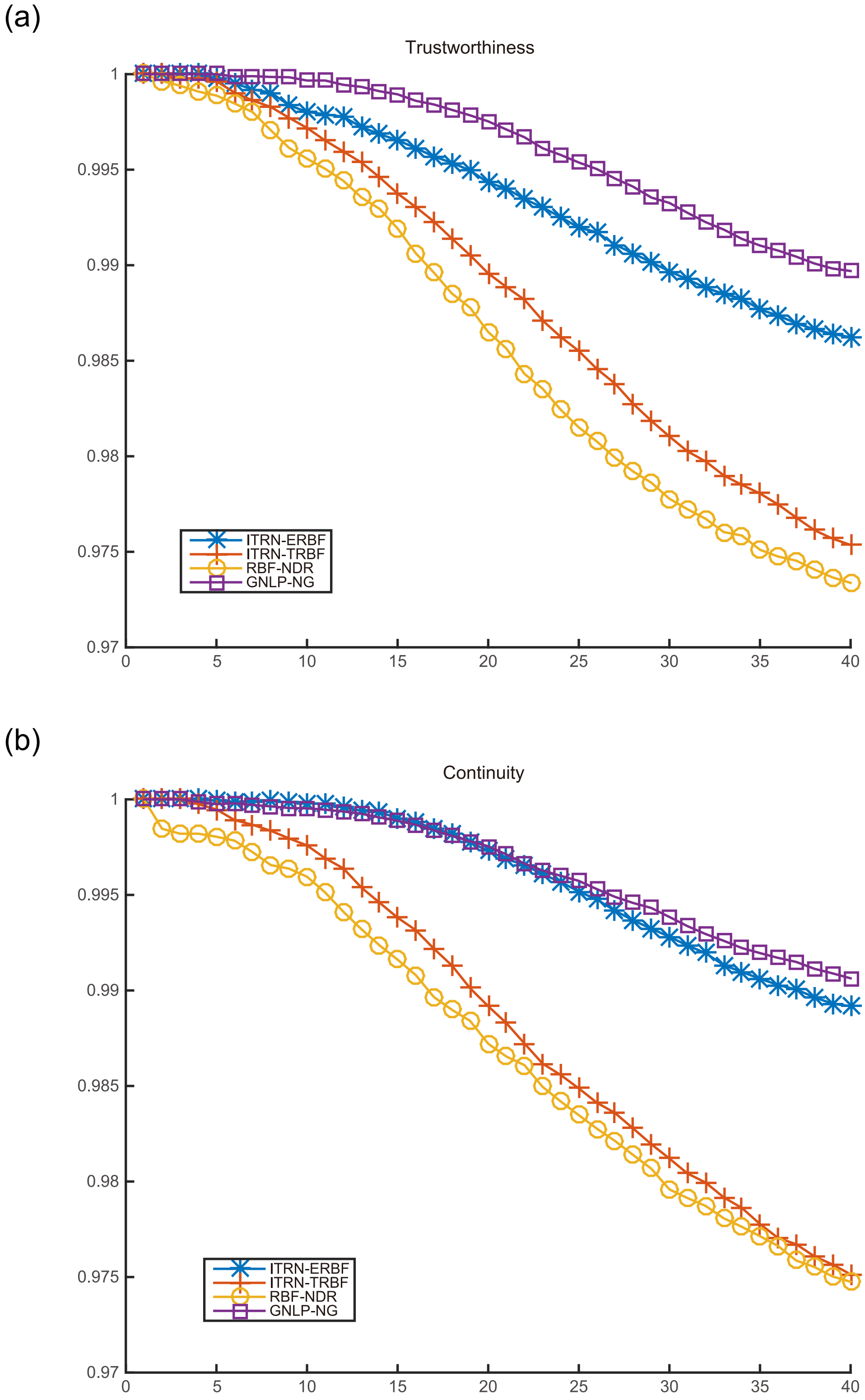

Fig 11. Mapping quality for AFs.

doi:10.1371/journal.pone.0131631.g011

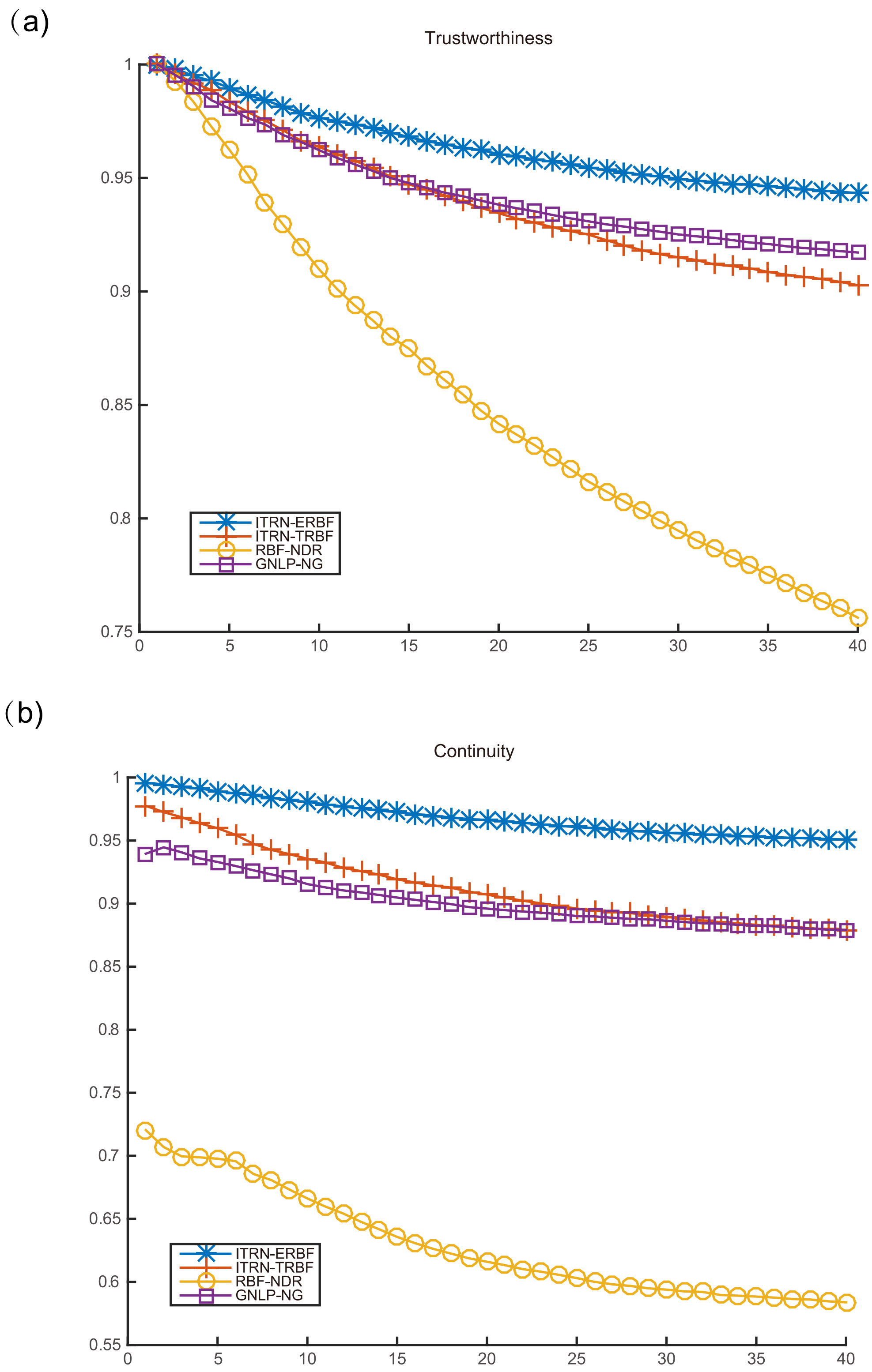

Fig 12. Mapping quality for handwritten digit “2”.

doi:10.1371/journal.pone.0131631.g012

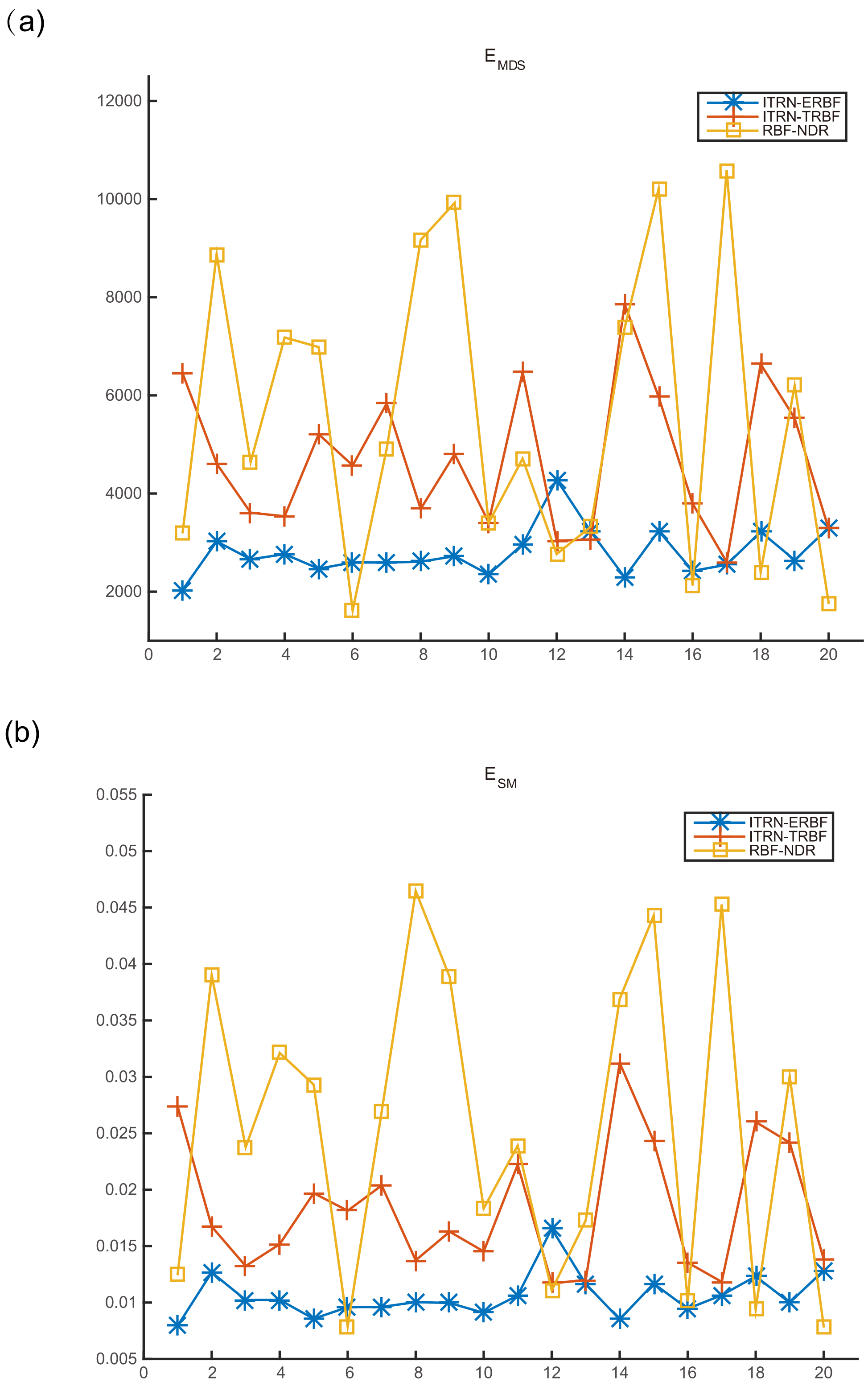

Fig 13. Comparison with RBF-NDR.

doi:10.1371/journal.pone.0131631.g013

set and $w(r, t)$ is given below:

$$w(r,t) = \begin{cases} \frac{28}{5}\left(\frac{t-r}{r}\right)^{5} - 14\left(\frac{t-r}{r}\right)^{4} + \frac{46}{5}\left(\frac{t-r}{r}\right)^{3} + \frac{1}{5}\left(\frac{t-r}{r}\right)^{2} & r \leqslant t \leqslant 2r, \\ 1 & others. \end{cases} \tag{22}$$

Smooth Neighborhood preservation can be obtained by simply computing:

$$E_{NP} = \frac{1}{2|S|}\sum_{\mathbf{v}_i \in S}(W^{T}(\mathbf{v}_i) + W^{FN}(\mathbf{v}_i)), \tag{23}$$

where S is the set of nodes under analysis. The smaller the value of E_{NP} means the better neighborhood preservation.

We add a dataset, three people's face images (S5 Dataset) in UMist Faces database (575 total images, 112 × 92 size, manually cropped by Daniel Graham [44]), for showing feature extraction (Fig 14). Table 2 presents the quality metrics' value for the different methods. We observe that PCA has poor performance because of nonlinear datasets. ISOMAP is better than L-ISOMAP because L-ISOMAP approximates a large global computation. ITRN-ERBF is better than ITRN-TRBF because ITRN-TRBF is trained and it has less center nodes in network. ITRN-RBF, ISOMAP and L-ISOMAP have similar results. In some cases, ITRN-RBF performs better than ISOMAP and L-ISOMAP. That illustrates the effectiveness of ITRN-RBF.

Computational complexity analysis

Assume that input space's nodes number is N, codebook vectors number is n, TRN's epochs is k_1 and TRBF's epochs is k_2. The most time consuming part of TRN corresponds to sorting the distances for rank r_i which goes with $O(Nlog_2N)$. Our improvement of TRN increases time cost because of building connecting graph. The extra time cost is $O(n^2)$. However, in most applications, this time cost can be neglectable because of the small value of n. The MDS has complexity $O(n^3)$. The TRBF is $O(k_2n)$ and the ERBF is $O(n)$. So ITRN-RBF runs in $O(k_1Nlog_2N + n^3 + k_2n)$ (based on TRBF) or $O(k_1Nlog_2N + n^3)$ (based on ERBF).

We list the running times in Table 3. Specially, training RBF and mapping dataset are separated, so the extent to which RBFN maps the dataset fast are quite remarkable. We note that:

- In most applications, $n < < N$, so MDS and training RBFN run faster.
- If we get RBFN, mapping the dataset only costs $O(N)$.
- ITRN-TRBF is slower than ITRN-ERBF because trained RBFN has iterative procedure. However, if we get RBFN, the mapping based TRBF is always faster than ERBF's, because ERBF has larger number of center nodes in network.

Discussion

The classical dimensionality reduction methods, such as PCA and MDS cannot disclose nonlinear embedded manifolds. ISOMAP and L-ISOMAP uses geodesic distantce to improve MDS, providing good performance. ITRN-RBF offers performance near that, but has a faster mapping speed and an ability to deal with new data.

For the dimensionality reduction methods based on TRN, OVI-NG can also not process nonlinear dataset because it uses Euclidean distances in the observation space. GNLP-NG makes improvements that are similar to ISOMAP's. Both of OVI-NG and GNLP-NG cannot project new data online.

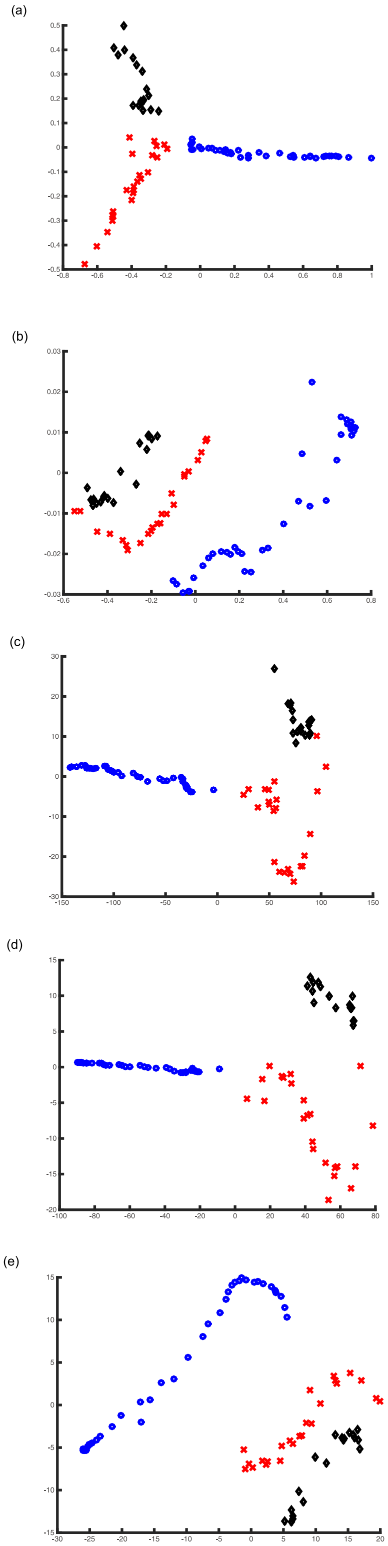

Fig 14. Visualizations of the UMist faces dataset. Different people's faces denote different marks (black rhomb, red cross, blue circle). (a) ITRN-ERBF, (b) ITRN-TRBF, (c) ISOMAP, (d) L-ISOMAP and (e) PCA.

doi:10.1371/journal.pone.0131631.g014

Table 2. Values of quality metrics for ITRN-RBF and classical dimensionality reduction methods.

		Swiss roll			
Quality Metrics	ITRN-ERBF	ITRN-TRBF	ISOMAP	L-ISOMAP	PCA
E_{CC}	6.7798E-04	0.0067	3.0124E-04	5.3887E-04	0.1536
E_{NP}	0.0493	0.1082	0.1330	0.1537	0.7373
E_{SM}	0.0094	0.0153	0.0012	0.0923	0.4921
		AF			
Quality Metrics	ITRN-ERBF	ITRN-TRBF	ISOMAP	L-ISOMAP	PCA
E_{CC}	0.0313	0.2569	0.1760	0.1326	0.1111
E_{NP}	0.3271	0.4159	0.4752	0.4199	0.5887
E_{SM}	0.0152	0.0295	0.0857	0.1608	0.1700
		“2”			
Quality Metrics	ITRN-ERBF	ITRN-TRBF	ISOMAP	L-ISOMAP	PCA
E_{CC}	0.2069	0.3200	0.2316	0.2641	0.4014
E_{NP}	0.5184	0.5504	0.5657	0.5837	0.5806
E_{SM}	0.9379	0.9851	0.1219	0.2529	0.3814
		UMist face			
Quality Metrics	ITRN-ERBF	ITRN-TRBF	ISOMAP	L-ISOMAP	PCA
E_{CC}	0.0057	0.0429	0.0080	0.0051	0.0583
E_{NP}	0.0252	0.1813	0.1788	0.1129	0.1729
E_{SM}	0.9877	0.9840	0.0138	0.1177	0.1305

doi:10.1371/journal.pone.0131631.t002

Table 3. Running times (specified in seconds) for different methods.

Dataset	ITRN-ERBF		ITRN-TRBF	
	Training RBFN	Mapping	Training RBFN	Mapping
Swiss roll	24.3739	0.0626	47.5088	0.0031
AF	14.2524	0.2210	199.2901	0.2271
“2”	81.6783	2.8812	387.0357	0.3082
UMist face	5.7719	0.2558	588.1310	0.3176

Dataset	RBF-NDR	GNLP-NG	ISOMAP	L-ISOMAP	PCA
Swiss roll	18.2267	175.9537	11.7621	5.1544	0.0016
AF	41.6311	99.7312	1.1107	0.7062	0.0152
“2”	184.3450	106.8090	2.8570	1.4513	0.2179
UMist face	———	———	0.0443	0.0422	0.0293

doi:10.1371/journal.pone.0131631.t003

ITRN-RBF and RBF-NDR overcome these problems. They can project nonlinear data for using geodesic distances and can map new data because of RBFN. In this paper, we proposed two methods to obtain the RBFN. Each has distinct advantages and disadvantages. ERBF has only one parameter, its *spread*. Larger *spread* will generate more robust networks, but too large a *spread* will cause mathematical calculation problems. ERBF only calculates once without accumulating error, hence it is fast and exact. However, a large number of training patterns will result in a large-scale network. ITRN uses the vector quantization technique to decrease the number of training patterns, hence ERBF is the recommended approach to obtain an RBFN.

The other method, TRBF, obtains a training RBFN, which requires a large number of adjustable parameters and calculation time.

Compared with RBF-NDR, ITRN-RBF has definitive results and high mapping quality. ITRN-RBF has good scalability with reasonable hardware costs. That is, if more effective methods for getting RBFN are adopted, better performance is obtained.

To sum up, the proposed ITRN-RBF that uses ITRN, which is suitable for geodesic distances because it builds a more appropriate topology relationship, does well with nonlinearly embedded manifolds, large amounts of data, and the online projection of new data. This method can be applied to a wide range of applications including visualization, feature extraction, and other applications.

Supporting Information

S1 Dataset. Randomly generated nodes dataset.
(ZIP)

S2 Dataset. Swiss roll dataset.
(ZIP)

S3 Dataset. AF dataset.
(ZIP)

S4 Dataset. Handwritten digit "2" dataset.
(ZIP)

S5 Dataset. UMist face dataset.
(ZIP)

Acknowledgments

This work was supported by the National Science Foundation of China under grant 61375065 and 61432014, partially supported by the National Program on Key Basic Research Project (973 Program) under grant 2011CB302201.

Author Contributions

Conceived and designed the experiments: SN JL ZC. Performed the experiments: SN ZC. Analyzed the data: SN JL ZC ML. Contributed reagents/materials/analysis tools: SN JL ML. Wrote the paper: SN ZC.

References

1. Xu HM, Sun XW, Qi T, Lin WY, Liu NJ, Lou XY. Multivariate dimensionality reduction approaches to identify gene-gene and gene-environment interactions underlying multiple complex traits. Plos One. 2014; 9(9). PubMed PMID: WOS:000342685600046. doi: 10.1371/journal.pone.0108103
2. Cattaert T, Urrea V, Naj AC, De Lobel L, De Wit V, Fu M, et al. FAM-MDR: a flexible family-based multifactor dimensionality reduction technique to detect epistasis using related individuals. Plos One. 2010; 5(4). PubMed PMID: WOS:000276952600021. doi: 10.1371/journal.pone.0010304 PMID: 20421984
3. Tang L, Peng SL, Bi YM, Shan P, Hu XY. A new method combining LDA and PLS for dimension reduction. Plos One. 2014; 9(5). PubMed PMID: WOS:000336653300053. doi: 10.1371/journal.pone.0096944
4. Bharti KK, Singh PK. Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering. Expert Systems with Applications. 2015; 42(6):3105–14. PubMed PMID: INSPEC:14856234. doi: 10.1016/j.eswa.2014.11.038

5. Li B, Li J, Zhang XP. Nonparametric discriminant multi-manifold learning for dimensionality reduction. Neurocomputing. 2015; 152:121–6. PubMed PMID: INSPEC:14857289. doi: 10.1016/j.neucom.2014.11.012

6. Ingram S, Munzner T. Dimensionality reduction for documents with nearest neighbor queries. Neurocomputing. 2015; 150:557–69. PubMed PMID: WOS:000346952300022. doi: 10.1016/j.neucom.2014.07.073

7. Dominguez M, Alonso S, Moran A, Prada MA, Fuertes JJ. Dimensionality reduction techniques to analyze heating systems in buildings. Information Sciences. 2015; 294:553–64. PubMed PMID: WOS:000346542800039. doi: 10.1016/j.ins.2014.06.029

8. Espezua S, Villanueva E, Maciel CD, Carvalho A. A Projection Pursuit framework for supervised dimension reduction of high dimensional small sample datasets. Neurocomputing. 2015; 149:767–76. PubMed PMID: WOS:000346550300030. doi: 10.1016/j.neucom.2014.07.057

9. Boutsidis C, Zouzias A, Mahoney MW, Drineas P. Randomized dimensionality reduction for k-means clustering. IEEE Transactions on Information Theory. 2015; 61(2):1045–62. PubMed PMID: INSPEC:14863774. doi: 10.1109/TIT.2014.2375327

10. Gisbrecht A, Schulz A, Hammer B. Parametric nonlinear dimensionality reduction using kernel t-SNE. Neurocomputing. 2015; 147:71–82. PubMed PMID: WOS:000343337600007. doi: 10.1016/j.neucom.2013.11.045

11. Jolliffe IT. Principal component analysis. Springer Series in Statistics. 2002; 87(100):41–64.

12. Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology. 1933; 24:417–41. PubMed PMID: WOS:000202766900037. doi: 10.1037/h0070888

13. Ma JZ, Amos CI. Principal components analysis of population admixture. Plos One. 2012; 7(7). PubMed PMID: WOS:000306354700027. doi: 10.1371/journal.pone.0040115

14. Bollen J, Van de Sompel H, Hagberg A, Chute R. A principal component analysis of 39 scientific impact measures. Plos One. 2009; 4(6). PubMed PMID: WOS:000267465900001. doi: 10.1371/journal.pone.0006022

15. Ye M, Zhang Yi, Lv JC. A globally convergent learning algorithm for PCA neural networks. Neural Computing & Applications. 2005; 14(1):18–24. PubMed PMID: WOS:000228978000003. doi: 10.1007/s00521-004-0435-y

16. Lv JC, Tan KK, Zhang Yi, Huang SN. A family of fuzzy learning algorithms for robust principal component analysis neural networks. Ieee Transactions on Fuzzy Systems. 2010; 18(1):217–26. PubMed PMID: WOS:000274217500020. doi: 10.1109/TFUZZ.2009.2038711

17. Shang LF, Lv JC, Zhang Y. Rigid medical image registration using PCA neural network. Neurocomputing. 2006; 69(13–15):1717–22. PubMed PMID: WOS:000239015000038. doi: 10.1016/j.neucom.2006.01.007

18. Lv JC, Zhang Y, Tan KK. Global convergence of GHA learning algorithm with nonzero-approaching adaptive learning rates. Ieee Transactions on Neural Networks. 2007; 18(6):1557–71. PubMed PMID: WOS:000250789100001. doi: 10.1109/TNN.2007.895824

19. Torgerson WS. Theory and methods of scaling. Biometrika. 1958.

20. Borg I, Groenen P. Modern mutlidimensional scaling: theory and applications. Springer Berlin. 2005; 40(3).

21. Wei M, Aragues R, Sagues C, Calafiore GC. Noisy Range Network Localization Based on Distributed Multidimensional Scaling. Sensors Journal IEEE. 2015; 15(3):1872–83.

22. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290(5500):2323. PubMed PMID: WOS:000165995800050. doi: 10.1126/science.290.5500.2323 PMID: 11125150

23. Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation. 2003; 15(6):1373–96. PubMed PMID: WOS:000182530600005. doi: 10.1162/089976603321780317

24. Tenenbaum JB, de Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000; 290(5500):2319. PubMed PMID: WOS:000165995800049. doi: 10.1126/science.290.5500.2319 PMID: 11125149

25. Shen H, Tao D, Ma D. Dual-Force ISOMAP: A new relevance feedback method for medical image retrieval. Plos One. 2013; 8(12). PubMed PMID: WOS:000329323900034. doi: 10.1371/journal.pone.0084096

26. Silva VD, Tenenbaum JB, editors. Global versus local methods in nonlinear dimensionality reduction. Advances in Neural Information Processing Systems 15; 2003.

27. Martinetz T, Schulten K. A neural-gas network learns topologies. Artificial Neural Networks, Vols 1 and 2. 1991:397–402. PubMed PMID: WOS:A1991BV08V00064.

28. Martinetz T, Schulten K. Topology representing networks. Neural Networks. 1994; 7(3):507–22. PubMed PMID: WOS:A1994NJ80100009. doi: 10.1016/0893-6080(94)90109-0

29. Fritzke B. A growing neural gas network learns topologies. Advances in Neural Information Processing Systems 7. 1995:625–32. PubMed PMID: INSPEC:5211272.

30. Tokunaga K. Growing topology representing network. Applied Soft Computing. 2014; 22:311–22. PubMed PMID: WOS:000338706600026. doi: 10.1016/j.asoc.2014.04.028

31. Estévez PA, Figueroa CJ. Online data visualization using the neural gas network. Neural Networks. 2006; 19(6–7):923–34. PubMed PMID: WOS:000240269600022. PMID: 16806817

32. Estévez PA, Chong AM, Held CM, Perez CA. Nonlinear projection using geodesic distances and the neural gas network. Artificial Neural Networks—Icann 2006, Pt 1. 2006; 4131:464–73. PubMed PMID: WOS:000241472100049. doi: 10.1007/11840817_49

33. Tomenko V. Online dimensionality reduction using competitive learning and Radial Basis Function network. Neural Networks. 2011; 24(5):501–11. PubMed PMID: WOS:000290510000010. doi: 10.1016/j.neunet.2011.02.007 PMID: 21420831

34. Haykin S. Neural networks. 1998.

35. Park J, Sandberg IW. Universal approximation using radial-basis-function networks. Neural Computation. 1991; 3(2):246–57. PubMed PMID: INSPEC:3983597. doi: 10.1162/neco.1991.3.2.246

36. Fasshauer GE. Solving differential equations with radial basis functions: multilevel methods and smoothing. Advances in Computational Mathematics. 1999; 11(2–3):139–59. PubMed PMID: WOS:000083107200004. doi: 10.1023/A:1018919824891

37. Gan M, Chen CLP, Li HX, Chen L. Gradient radial basis function based varying-coefficient autoregressive model for nonlinear and nonstationary time series. Ieee Signal Processing Letters. 2015; 22 (7):809–12. PubMed PMID: WOS:000345569500003. doi: 10.1109/LSP.2014.2369415

38. Jinna L, Hongping H, Yanping B. Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm. Neurocomputing. 2015; 152:305–15. PubMed PMID: INSPEC:14857309. doi: 10.1016/j.neucom.2014.10.065

39. Dai XJ, Shao XX, Yang FJ, He XY. Non-destructive strain determination based on phase measurement and radial basis function. Optics Communications. 2015; 338:348–58. PubMed PMID: INSPEC:14812613. doi: 10.1016/j.optcom.2014.10.055

40. Sammon JW. A nonlinear mapping for data structure analysis. Ieee Transactions on Computers. 1969; C 18(5):401. PubMed PMID: WOS:A1969D511700001. doi: 10.1109/T-C.1969.222678

41. Kaski S, Nikkilä J, Oja M, Venna J, Törönen P, Castren E. Trustworthiness and metrics in visualizing similarity of gene expression. Bmc Bioinformatics. 2003; 4. PubMed PMID: WOS:000186690600001. doi: 10.1186/1471-2105-4-48 PMID: 14552657

42. Venna J, Kaski S. Local multidimensional scaling with controlled tradeoff between trustworthiness and continuity. 2005.

43. Pagliosa P, Paulovich FV, Minghim R, Levkowitz H, Nonato LG. Projection inspector: assessment and synthesis of multidimensional projections. Neurocomputing. 2015; 150:599–610. doi: 10.1016/j.neucom.2014.07.072

44. Graham DB, Allinson NM. Characterizing virtual eigensignatures for general purpose face recognition. 1998.