DOI: 10.1007/s11633-007-0294-y

Nonlinear Dimensionality Reduction and Data Visualization: A Review

Hujun Yin

School of Electrical and Electronic Engineering, The University of Manchester, Manchester M60 1QD, UK.

Abstract: Dimensionality reduction and data visualization are useful and important processes in pattern recognition. Many techniques have been developed in the recent years. The self-organizing map (SOM) can be an efficient method for this purpose. This paper reviews recent advances in this area and related approaches such as multidimensional scaling (MDS), nonlinear PCA, principal manifolds, as well as the connections of the SOM and its recent variant, the visualization induced SOM (ViSOM), with these approaches. The SOM is shown to produce a quantized, qualitative scaling and while the ViSOM a quantitative or metric scaling and approximates principal curve/surface. The SOM can also be regarded as a generalized MDS to relate two metric spaces by forming a topological mapping between them. The relationships among various recently proposed techniques such as ViSOM, Isomap, LLE, and eigenmap are discussed and compared.

Keywords: Dimensionality reduction, nonlinear data projection, multidimensional scaling, self-organizing maps, nonlinear PCA, principal manifold.

1 Introduction

Self-organization is a fundamental pattern recognition process, in which intrinsic inter- and/or intra-pattern relationships and structures within the sensory data are discovered. Data projection and visualization methods are becoming increasingly popular tools for many data discovery applications such as decision support[1], financial analysis[2], information retrieval[3], knowledge management[4] and bioinformatics[5]. Searching for a suitable data mapping method has always been an integral objective of multivariate data analysis and pattern recognition. Projecting data onto its underlying subspace can detect its real structures, facilitate functional analysis, and help make a judgment. A great deal of research has been devoted to this subject and a number of methods have been proposed.

Classic projection methods include principal component analysis (PCA) and multidimensional scaling (MDS). The PCA projects the data onto its principal directions, which are represented by the orthogonal eigenvectors of the covariance matrix of the data. The PCA′s linearity has limited its power for practical data, as it cannot capture nonlinear relationships defined by higher than second-order statistics. If the input dimensionality is much higher than two, projection onto a linear plane will provide limited dimension reduction or visualization power. Extension to nonlinear can tackle practical problems better. However there is no single and unique solution to nonlinear PCA[6]. Various methods have been proposed such as. the auto-associative networks[7], generalized PCA[8], kernel PCA[9], and the principal curve and surface[10,11]. Other mapping methods include the recently proposed local, geometric based grouping and averaging[12] and local linear embedding (LLE)[13].

Multidimensional scaling (MDS) tries to project data points onto a two-dimensional plane by preserving as close as possible the inter-point metrics[14,15]. The mapping generally is nonlinear and can reveal the overall structure of the data. Sammon[16] mapping is a widely known example of MDS. However, MDS methods are generally point-to-point mapping, which does not provide the explicit mapping function[16,17]. Neural networks have been used as alternative approaches to nonlinear data projection. A feedforward neural network has been proposed to parameterize the Sammon mapping function and a back-propagation algorithm has been derived for training of the network and minimizing the Sammon stress[17]. Neuroscale[18] is another realization of the MDS using the radial basis function. Recent developments also see the use of geodesic (curvature) distance instead of Euclidean for capturing nonlinear manifold better, e.g. in the Isomap[12].

The self-organizing map (SOM) is an abstract mathematical model of the mapping between nerve sensory and cerebral cortex[19,20]. Modeling and analyzing such mappings are important to understanding how the brain perceives, encodes, recognizes, and processes the patterns it receives and thus, if somewhat indirectly, are beneficial to machine-based pattern recognition. Indeed the SOM has been widely studied and applied in various pattern recognition tasks such as clustering, classification, data mining and visualization. However, the SOM does not directly apply to scaling, which aims to reproduce proximity in (Euclidean) distance on a low visualization space, as it has to rely on a coloring scheme to imprint the distances –that is very crude and often the distributions of the data points are distorted on the map. The recently proposed visualization induced SOM (ViSOM)[21] constrains the lateral contraction force between the neurons in the SOM and hence regularizes the inter-neuron distances with respect to a scale-able parameter that defines and controls the resolution of the map. It preserves the data structure as well as the topology as faithfully as possible. The ViSOM provides a direct visualization of both the structure and distribution of the data.

The remaining of the paper is organized as follows. Section 2 provides a review on the SOM, its convergence and

Manuscript received date May 26, 2007; revised date June 5, 2007

E-mail address: h.yin@manchester.ac.uk

cost function, and related variants, especially the ViSOM, for dimension reduction and data visualization. Then MDS is reviewed and discussed along with recent advances in this approach. Various methods proposed for nonlinear PCA are then described in section 4, followed by the principal curve and surfaces, the principled nonlinear extension of PCA in section 5. The connections and relationships among various approaches to nonlinear projection are discussed next. Conclusions are drawn in the last section.

2 Self-organizing maps: a review

2.1 The SOM background

External stimuli are received by various sensory or receptive fields (e.g. visual-, auditory-, motor-, or somatosensory), coded or abstracted by the living neural networks, and projected through axons onto the cerebral cortex, often to distinct parts of cortex. The different areas of the cortex correspond to different sensory inputs, though some functions require collective responses from various areas. Topographically ordered mappings have been widely observed in the cortex. Detailed areas (associative areas) are developed through self-organization gradually in a topographically meaningful fashion. Studying such topographic projections is important for forming dimension reduction mapping and for the effective representation of sensory information and feature extraction.

Von der Malsburg and Willshaw first developed in mathematical form the self-organizing topographic mappings, mainly from two-dimensional presynaptic sheets to two-dimensional postsynaptic sheets, based on retinatopic mapping: the ordered projection of visual retina to visual cortex[22,23]. Kohonen[19] abstracted this self-organizing learning model and proposed a much simplified mechanism which ingeniously incorporates the Hebbian learning rule and lateral interconnection rules. This simplified model can emulate the self-organization effect. Although the resulting SOM algorithm was more or less proposed in a heuristic manner, it is an abstract and generalized model of the self-organization or unsupervised learning process.

2.2 The SOM algorithm

The SOM uses a set of neurons, often arranged in a 2D rectangular or hexagonal grid or map, to form a discrete, topological mapping of an input space, $\boldsymbol{X}\in \mathbf{R}^n$. At the start of the learning, all the weights $\{\boldsymbol{w}_{\boldsymbol{r}1}, \boldsymbol{w}_{\boldsymbol{r}2}, \ldots, \boldsymbol{w}_{\boldsymbol{r}M}\}$ are initialized to small random numbers. Here $\boldsymbol{w}_{\boldsymbol{r}i}$ is the weight vector associated to neuron i and is a vector of the same dimension, n, of the input. M is the total number of neurons. $\boldsymbol{r}i$ is the location vector (coordinates) of neuron i on the grid. Then the algorithm repeats the following steps.

1) At each time t, present an input, $\boldsymbol{x}(t)$, select the winner,

$$v(t) = \arg\min_{k\in\Omega} \|\boldsymbol{x}(t) - \boldsymbol{w}_k(t)\|\,. \tag{1}$$

2) Updating the weights of winner and its neighbors,

$$\Delta\boldsymbol{w}_k(t) = \alpha(t)\eta(v,k,t)[\boldsymbol{x}(t) - \boldsymbol{w}_k(t)]. \tag{2}$$

3) Repeat until the map converges.

where $\eta(v,k,t)$ is the neighborhood function and Ω is the set of neuron indexes. Although one can use the original top-hat type of neighborhood function, a Gaussian form, $\eta(v,k,t) = \exp(-||v-k||^2/2\sigma(t)^2)$, is often used in practice with σ representing the effective range of the neighborhood.

The SOM algorithm vector-quantizes or clusters the input space and produces a map which preserves topology. It can also be and has been used for classification. In this case, the map is trained on examples of known categories. The nodes are then classified or labeled so that the map can be used to classify unseen samples. The classification performance can be further improved by the LVQ[20].

2.3 Convergence and cost function

The SOM is an unsupervised, associative memory mechanism[24,25]. Such a mechanism is also related to vector quantization (VQ) in coding terms. The SOM has been shown to be an asymptotically optimal VQ[26,27]. More importantly, with the neighborhood learning, the SOM is an error tolerant VQ and Bayesian VQ[28,29].

Convergence and ordering has only been proved in one dimensional case and the full proof of both convergence and ordering in multidimensional may not exist, though there have been several attempts, e.g. [30-34]. Such an issue may be due to the fact that no clearly agreed definition of ordering exists and may also be linked to the claimed lack of an exact cost function that the algorithm follows as shown in [31, 32]. Recent work by various researchers, however, has shed light on this intriguing issue surrounding the SOM. In [27] the Central Limit Theorem is extended and used it to show that with diminishing neighborhood as in the original SOM, the weight vectors are asymptotically Gaussian distributed and will converge in mean square sense to the means of the Voronoi cells. In [27], Yin and Allinson have also proved that the initial state has diminishing effect on the final weights when the learning parameters follow the convergence conditions. Such an effect has been verified by de Bolt *et al*[35] using Monte-Carlo bootstrap cross validation. The ordering was not considered.

Luttrell[28] first related hierarchical noise tolerant coding theory to the SOM. When the transmission channel noise is considered, a two-stage optimization has to be done by minimizing the representation distortion (as in the VQ) as well as the distortion caused by the channel noise. The SOM can be interpreted as such a coding algorithm. The neighborhood function acts as the model for the channel noise distribution and should not go to zero as in the original SOM. Such a noise tolerant VQ has the following objective function[28]:

$$D_2 = \int \mathrm{d}\boldsymbol{x}p(\boldsymbol{x}) \int \mathrm{d}\boldsymbol{n}\pi(\boldsymbol{n})\, \|\boldsymbol{x} - \boldsymbol{w}_k\|^2 \tag{3}$$

where $\boldsymbol{n}$ is the noise variable and $\pi(\boldsymbol{n})$ is the noise distribution. Durbin and Mitchison[36] and Mitchison[37] have also linked the SOM and this noise tolerant VQ with minimal wiring of cortex like maps.

When the codebook (the map) is finite, the noise can be considered as discrete, then the cost function can be re-expressed as

$$D_2 = \sum_i \int_{V_i} \sum_k \pi(i,k) \left\| \boldsymbol{x} - \boldsymbol{w}_k \right\|^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \qquad (4)$$

where V_i is the Voronoi region of cell i. When the channel noise distribution is replaced by a neighborhood function (analogous to inter-symbol dispersion), this gives to the cost function of the SOM. The neighborhood function can be interpreted as channel noise model. Such a cost function has been discussed in the SOM community, e.g. [15, 26, 38-40]. The cost function is therefore

$$E(\boldsymbol{w}_1, ...\boldsymbol{w}_M) = \sum_i \int_{V_i} \sum_k \eta(i,k) \left\| \boldsymbol{x} - \boldsymbol{w}_k \right\|^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \qquad (5)$$

It leads naturally to the SOM update algorithm based on the stochastic or sample gradient descent method. That is, for each Voronoi region, the sub cost function is

$$E_i(\boldsymbol{w}_1, ...\boldsymbol{w}_M) = \int_{V_i} \sum_k \eta(i,k) \left\| \boldsymbol{x} - \boldsymbol{w}_k \right\|^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}. \qquad (6)$$

Then the optimization for the weights $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_M\}$ can be sought using the sample gradients., which are

$$\frac{\partial \widehat{E}_i(\boldsymbol{w}_1, ...\boldsymbol{w}_M)}{\partial \boldsymbol{w}_j} = 2\eta(i,j)(\boldsymbol{x} - \boldsymbol{w}_j). \qquad (7)$$

This results in the SOM updating rule (2). Note, although the neighborhood function $\eta(i,j)$ is inexplicitly related to $\boldsymbol{w}_j$, it does not contribute to the weight optimization, nor does the weight optimization lead to its adaptation as neighborhood adaptation is controlled by a pre-specified scheme, unrelated to the weight adaptation. Thus the neighborhood can be omitted from taking partial differentiation.

It has been argued however that this energy function is violated at boundaries of Voronoi cells where input $\boldsymbol{x}$ has exactly the same smallest distance to two neighboring neurons. Thus this energy function holds mainly for discrete cases where the probability of such boundary input points is close to zero, or the local (sample) cost function should be used in deciding the winner[40]. When spatial-invariant neighborhood function is used as it is often the case, assigning the boundary input to either cells will lead to the same local sample cost or error, therefore any input data on the boundary can be assigned to either Voronoi cells that have the same smallest distance to it. Only when the neurons lie on the borders of the map, such violation occurs as unbalanced neighborhoods of the neurons. The result is a slightly more contraction towards to the center or inside of the map for the border neurons compared to the common SOM algorithm as shown in [38]. Using either the simple distance or local distortion measure as the winning rule will result in border neurons be contracted towards inside the map, especially when the map is not fully converged or when the effective range of the neighborhood function is great. With the local distortion rule, this boundary effect is heavier as greater local error is incurred for the border neurons due to its few neighboring neurons than any inside neurons.

To exactly follow the cost function, the winning rule should be modified to follow the local sample cost function $\widehat{E}_i$ or the local distortion measure (instead of the simplest nearest distance)

$$v = \arg\min_i \sum_k \eta(i,k) \left\| \boldsymbol{x} - \boldsymbol{w}_k \right\|^2. \qquad (8)$$

When the neighborhood function is symmetric as it is often the case and when the data density function is smooth, this local distortion winning rule is the same as to the simplest nearest distance rule for most, non-boundary nodes, especially as the number of nodes is large. On the borders of the map, however, differences exist due to the unbalance of the nodes presented in the neighborhoods. Such differences become negligible to the majority of the neurons especially when a large map is used and when the neighborhood function shrinks to its minimum scale or just the winner.

2.4 Topological order measures

The quality of the mapping in terms of topographic or topological preservation is measured for its topological ordering, in addition to the overall quantization error. Such a measure is not unique (unless input and map dimensions are equal) and there is no clear definition of order[41]. Among several proposed measures, Bauer and Pawelzik[42] proposed a measure called the topology product to measure the topological ordering of the map,

$$P = \frac{1}{M^2 - M} \sum_i \sum_j \log \left(\prod_{l=1}^{j} \frac{d^D(\boldsymbol{w}_i, \boldsymbol{w}_{\eta^O(l,i)})}{d^D(\boldsymbol{w}_i, \boldsymbol{w}_{\eta^D(l,i)})} \frac{d^O(i, \eta^o(l,i))}{d^O(i, \eta^D(l,i))} \right)^{\frac{1}{2k}} \qquad (9)$$

where d^D and d^O represent the distance measures in the input or data space and on the map respectively. $\eta(l{,}i)$ represents the l-th neighbor of node i in either data (D) or map(O) space.

The first ratio in the product measures the ratio or match of weight distance sequences of a neighborhood (upto j) on the map and in the data space. The second ratio is the index distance sequences of the neighborhood on the map and in the data space. The topographic product measures the product of the two ratios of all the neighborhoods.

Villmann *et al*[43] proposed a topographic function to measure the neighborhoodness of weight vectors in data space as well as on the map. While the neighborhoodness of the weight vectors is defined by the adjacent Voronoi cells of the weights. The function measures the degree of weight vectors are ordered in the data space as to their indexes on the lattice, as well as how well the indexes are preserved when their weight vectors are neighbors.

Defining a fully ordered map can be straightforward using the distance relations[26]. For example, if all the nearest neighboring nodes on the map have their nearest neighboring nodes$'$ weights in their nearest neighborhood in the data space, we can call the map is a 1^{st}-order (ordered) map[26],

$$d(\boldsymbol{w}_i, \boldsymbol{w}_j) \leq d(\boldsymbol{w}_i, \boldsymbol{w}_k), \ \forall i \in \Omega; \ j \in \eta_i^1; \ k \notin \eta_i^1 \qquad (10)$$

where Ω is the map and η_i^1 denotes the 1^{st}-order neighborhood of node i.

Similarly if the map is a 1^{st}-order ordered map, and all the 2^{nd} nearest neighboring nodes have their weights in their 2^{nd} nearest neighborhood in the data space, we can call the map a 2^{nd}-order (ordered) map. For the 2^{nd} ordered map, the distance relations to be satisfied are

$$d(\boldsymbol{w}_i, \boldsymbol{w}_j) \leq d(\boldsymbol{w}_i, \boldsymbol{w}_k) \leq d(\boldsymbol{w}_i, \boldsymbol{w}_l),$$
$$\forall i \in \Omega;\ j \in \eta_i^1;\ k \notin \eta_i^1\ \&\ k \in \eta_i^2;\ l \notin \eta_i^2. \quad (11)$$

And so forth to define higher ordered maps with interneuron distance hierarchies. An m-th order map is optimal for tolerate the channel noise spreading or inter-symbol dispersion upto m-th neighboring code. Such a fully ordered map however may not be achievable, especially when the mapping is a dimension-reduction one. Then the degree or percentage to which the nodes and their weights are ordered can be measured, together with the probabilities that the nodes are utilized, can determine the topology preservation and that to what degree and to what order the map can tolerate the (channel) noise.

Goodhill and Sejnowski[41] proposed the C measure, a correlation between the similarity of stimulus in the data space and the similarity of their prototypes in the map space, to quantify the topological preservation

$$C = \sum_i \sum_j F(i,j) G[M(i), M(j)] \quad (12)$$

where F and G are symmetric similarity measures in the input and map spaces respectively and can be problem specific, and $M(i)$ and $M(j)$ are the mapped points or weight vectors of node i and j respectively.

The C measure directly evaluates the correlation between distance relations between two spaces. Various other topographic mapping objectives can be unified under the C measure such as multidimensional scaling, minimal wiring, and travel salesperson problem, and noise tolerant VQ. It has also been shown that if a mapping that preserves ordering exists then maximizing C will find it. Thus the C measure is also the objective function of the mapping, an important property different from other topology preservation measures and definitions.

The above list of measures is by no means complete. Other measures for ordering exist in the literature. One can always use the underlying cost function (5) to measure the goodness of the mapping including the topology preservation, at least one can use a temporal window to take a sample of it as suggested in [38]. The (final) neighborhood function specifies the level of topology (ordering) the mapping is likely to achieve or is required. To make an analogy to the above C measure, the neighborhood function can be interpreted as the G measure used in (12) and term $||\boldsymbol{x} - \boldsymbol{w}_k||^2$ represents the F measure. Indeed, the input $\boldsymbol{x}$ and weight $\boldsymbol{w}_j$ are mapped on the map as node index i and j and their G measure is the neighborhood function such as exponentials. Such an analogy also sheds light on the scaling effect of the SOM. Multidimensional scale also aims to preserve local similarities on a mapped space (see section 6 for more detailed account).

2.5 ViSOM

The SOM is optimal for vector quantization. Its topographic ordering provides the mapping with enhanced fault and noise tolerant abilities. It also provides a latent structure of the input space and is applicable to many applications, e.g. dimensionality reduction for face recognition[44]. In the aspect of data visualization and dimensionality reduction, the SOM has been linked with the principal curves and surfaces[45]. However the SOM does not preserve distance on the map. Instead it tries to establish topological order of the mapping between data points and their corresponding nodes on the map.

For scaling and data visualization, a direct and faithful display of data structure and distribution is desirable. The visualization induced SOM (ViSOM) has been proposed to extend the SOM for distance preservation on the map[21], instead of using a crude coloring scheme, which imprints qualitatively the interneuron distances as colors or grey levels on the map. For the map to capture the data structure naturally and directly, (local) distance quantities must be preserved on the map, along with the topology. The map can be seen as a smooth and graded mesh or manifold embedded into the data space, onto which the data points are mapped and the inter-point distances are approximately preserved.

In order to achieve that, the updating force, $[\boldsymbol{x}(t) - \boldsymbol{w}_k(t)]$, of the SOM algorithm, is decomposed into two components: $[\boldsymbol{x}(t) - \boldsymbol{w}_v(t)] + [\boldsymbol{w}_v(t) - \boldsymbol{w}_k(t)]$. The first term represents the updating force from the winner v to the input $\boldsymbol{x}(t)$, and is the same to the updating force used by the winner. The second force is a lateral contraction force that brings the neighboring neurons to the winner. In the ViSOM, this lateral contraction force is regulated in order to help maintain a uniform inter-neuron distance $||\boldsymbol{w}_v(t) - \boldsymbol{w}_k(t)||$, at least locally on the map.

The update rule of the ViSOM is[21]

$$\begin{aligned}\boldsymbol{w}_k(t+1) =& \boldsymbol{w}_k(t) + \alpha(t)\eta(v,k,t)\{[\boldsymbol{x}(t) - \boldsymbol{w}_v(t)] + \\ & \beta[\boldsymbol{w}_v(t) - \boldsymbol{w}_k(t)]\}. \end{aligned} \quad (13)$$

where β is the constraint -the simplest constraint can be $\beta = d_{vk}/(D_{vk}\lambda)$-1, d_{vk} is the distance of neuron weights in the input space, D_{vk} is the distance of neuron indexes on the map, and λ is a (required) resolution constant.

The ViSOM regularizes the contraction force so that the distances between the nodes on the map are analogous to the distances of their weights in the data space locally. The aim is to make inter-neuron distances on the map proportional to those in the data space, i.e. $D_{vk} \propto d_{vk}$ or exactly $\lambda D_{vk} \approx d_{vk}$. When the data points are eventually projected on a trained map, the distance between points i and j on the map reflects the distance between these two points in the data space, subject to the quantization error (the distance between a data point and its neural representative). This makes visualization more direct and quantitatively measurable. The resolution of the map can be enhanced by interpolating a trained (small) map[46] or by incorporating local linear projections[47]. The size or covering range of the neighborhood function can also be decreased from an initially large value to a final small one. The final neighborhood, however, should not contain just the winner. The rigidity or curvature of the map is controlled by the ultimate size of the neighborhood. The larger of this size the flatter the final map is in the data space. Guidelines for setting these parameters can be found in [48].

Several improvements have since been made since. For example in [49] a probabilistic data assignment is used in both the input assignment and the neighborhood function and an improved second order constraint is adopted. In [50] the ViSOM has been extended to arbitrary, neural gas type of map structure.

3 Metric multidimensional scaling

Multidimensional scaling (MDS) is a traditional subject related to dimension reduction and data visualization. MDS tries to project data points onto an often two-dimensional plane by preserving as closely as possible the inter-point metrics[14]. The projection is generally nonlinear and can reveal the overall structure of the data. A general fitness function or the so-called stress function is defined as

$$S=\frac{\sum\limits_{i,j}(d_{ij}-D_{ij})^2}{\sum\limits_{i,j}D_{ij}^2} \tag{14}$$

where d_{ij} represents the proximity (or dissimilarity) of data points i and j in the original data space, D_{ij} represents the distance (usually Euclidean) between mapped points i and j in the projected space.

MDS relies on an optimization algorithm to search for a configuration that gives as low stress as possible. A gradient method is commonly used for this purpose. Inevitably, various computational problems such as local minima and divergence may occur to the optimization process. The methods are also often computationally intensive. The final solution depends on the starting configuration and parameters used in the algorithm.

Sammon mapping[16] is a well-known example of MDS. The objective of Sammon mapping is to minimize the differences between inter-point (Euclidean) distances in the original space and those in the projected plane.

$$S_{\text{Sammon}}=\frac{1}{\sum\limits_{i<j}d_{ij}}\sum_{i<j}\frac{(d_{ij}-D_{ij})^2}{d_{ij}}. \tag{15}$$

In Sammon mapping intermediate normalization (of original space) is used to preserve good local distributions and at the same time to maintain a global structure. A second-order Newton optimization method is used to recursively solve the optimal configuration. It converges faster than the simple gradient method, but the computational complexity is higher. It may still have the local minima and inconsistency problems. The Sammon mapping has been shown to be useful for data structure analysis. However, like other MDS methods, the Sammon algorithm is a point-to-point mapping, which does not provide an explicit mapping function and cannot naturally accommodate new data points. It also requires computing and storing all the inter-point distances. This can prove difficult or even impossible for many practical applications where data arrives sequentially, the quantity of data is large, and/or memory space for the data is limited.

In addition to being computationally costly for large data sets and not adaptive, another major drawback of MDS is lack of an explicit projection function. Thus for any new input data, the mapping has to be recalculated based on all available data. Although some methods have been proposed to accommodate the new arrivals using triangulation[51,52], the methods are generally not adaptive. However, such drawbacks can be overcome by implementing or parameterizing MDS using neural networks[17,18]. Recently Isomap was proposed to use geodesic (curvature) distance instead for better nonlinear scaling[12]. Geodesic distance is calculated (or cumulated) along the manifold (instead of a direct Euclidean distance) and is often implemented *via* neighborhood graphs or neighboring points. Selecting a suitable neighborhood size can be a difficult task and often needs cross-validation procedure. Isomap has been reported being unstable[53].

4 Nonlinear PCA

PCA is a classic linear projection method aiming at finding orthogonal principal directions from a set of data, along which the data exhibit the largest variances. By discarding the minor components, the PCA can effectively reduce data variables and display the dominant ones in a linear, low dimensional subspace. It is the optimal linear projection in the sense of the mean-square error between original points and projected ones, i.e.,

$$\min\sum_{\boldsymbol{x}}\left(\boldsymbol{x}-\sum_{j=1}^{m}(\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{x})\boldsymbol{q}_j\right)^2 \tag{16}$$

where $\{\boldsymbol{q}_j,\ j=1,2,\ldots,m,\ m\leq n\}$ are orthogonal eigenvectors representing principal directions. They are the first m principal eigenvectors of the covariance matrix of the input. The second term in the above bracket is the reconstruction or projection of $\boldsymbol{x}$ on these eigenvectors. The term $\boldsymbol{q}_j^{\mathrm{T}}\boldsymbol{x}$ represents the projection of $\boldsymbol{x}$ onto the j-th principal dimension. Traditional methods for solving eigenvector problem involve numerical methods. Though fairly efficient and robust, they are not usually adaptive and often require the presentation of the entire data set. Several Hebbian-based learning algorithms and neural networks have been proposed for performing PCA such as, the subspace network[54] and the generalized Hebbian algorithm[55]. The limitation of linear PCA is obvious, as it cannot capture nonlinear relationships defined by higher than the second order statistics. If the input dimension is much higher than two, the projection onto linear principal plane will provide limited visualization power.

The extension to nonlinear PCA is not unique, due to the lack of a unified mathematical structure and an efficient and reliable algorithm, and in some cases due to excessive freedom in selection of representative basis functions[6,8]. Several methods have been proposed for nonlinear PCA such as, the five-layer feedforward associative network[7] and the kernel PCA[9]. The first three layers of the associative network project the original data on to a curve or surface, providing an activation value for the bottleneck node. The last three layers define the curve and surface. The weights of the associative network are determined by minimizing the following objective function

$$\min\sum_{\boldsymbol{x}}\|\boldsymbol{x}-\boldsymbol{f}\{s_f(\boldsymbol{x})\}\|^2 \tag{17}$$

where $\boldsymbol{f}$: $\mathbf{R}^1 \to \mathbf{R}^n$ (or $\mathbf{R}^2 \to \mathbf{R}^n$), the function modeled by the last three layers, defines a curve (or a surface), s_f: $\mathbf{R}^n \to \mathbf{R}^1$ (or $\mathbf{R}^n \to \mathbf{R}^2$), the function modeled by the first three layers, defines the projection index.

The kernel-based PCA uses nonlinear mapping and kernel functions to generalize PCA to nonlinear and has been used for various pattern recognition. The nonlinear function $\boldsymbol{\Phi}(\boldsymbol{x})$ maps data onto high-dimensional feature space, where the standard linear PCA can be performed *via* kernel functions: $k(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{y}))$. The projected covariance matrix is then,

$$Cov = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\Phi}(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_i)^{\mathrm{T}}. \tag{18}$$

The standard linear eigenvalue problem can now be written as $\lambda \boldsymbol{V} = \boldsymbol{K}\boldsymbol{V}$, where the columns of $\boldsymbol{V}$ are the eigenvectors and $\boldsymbol{K}$ is a $N \times N$ matrix with elements as kernels $K_{ij} := k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j))$.

Local linear embedding (LLE)[13] is another way of forming nonlinear principal subspace. The local linearity is defined on a local neighborhood, say k nearest neighbors. Then the linear contributions or weightings, W_{ij}, of these neighboring points are calculated through

$$\min \sum_i \left\| \boldsymbol{x}_i - \sum_{j=1} W_{ij}\boldsymbol{x}_j \right\|^2. \tag{19}$$

The embedding is computed *via*

$$\min \sum_i \left\| Y_i - \sum_{j=1} W_{ij} Y_j \right\|^2 \tag{20}$$

where Y is the embedding coordinates.

Recently eigenmap[56] has been proposed to form a local linear mapping by converting the problem to a generalized eigenproblem and the solution becomes easily traceable.

First, the weightings (of local neighboring points) or heat kernels are constructed

$$W_{ij} = \exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{t}). \tag{21}$$

Then the embedding is computed *via* the generalized eigenproblem

$$L\boldsymbol{f} = \lambda D\boldsymbol{f} \tag{22}$$

where $D_{ii} = \sum_j W_{ji}$ and $L = D - W$

The data is then projected to the subspace spanned by the principal eigen functions $(\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_m)$. This approach is also related to spectral clustering[57].

5 Principal manifolds

The principal curves and principal surfaces[10,11,58] are the principled nonlinear extension of PCA. The principal curve is defined as a smooth and self-consistency curve, which does not intersect itself, passing through the middle of the data. Denote $\boldsymbol{x}$ as a random vector in $\mathbf{R}^n$ with density p and finite second moment. Let $f(\cdot)$ be a smooth unit-speed curve in $\mathbf{R}^n$, parameterized by the arc length ρ (from one end of the curve) over $\Lambda \in \mathbf{R}$, a closed interval.

For a data point $\boldsymbol{x}$, its projection index on f is defined as

$$\rho_f(\boldsymbol{x}) = \sup_{\rho \in \Lambda}\{\rho : \|\boldsymbol{x} - f(\rho)\| = \inf_{\vartheta} \|\boldsymbol{x} - f(\vartheta)\|\}. \tag{23}$$

The curve is called self-consistent principal curve of ρ if

$$f(\rho) = E[\boldsymbol{X}|\rho_f(\boldsymbol{X}) = \rho]. \tag{24}$$

The principal component is a special case of the principal curves if the distribution is ellipsoidal. Although principal curves have been mainly studied, extension to higher dimension, e.g. principal surfaces or manifolds is feasible in principle. However, in practice, a good implementation of principal curves/surfaces relies on an effective and efficient algorithm. The principal curves/surfaces are more of a concept that invites practical algorithm and implementations. The HS algorithm is a nonparametric method[10] that directly iterates the two steps of the above definition. It is similar to the standard LGB VQ algorithm[59] combined with some smoothing techniques.

HS Algorithm

Initialization: Choose the first linear principal component as the initial curve, $f^{(0)}(\boldsymbol{x})$.

Projection: Project the data points onto the current curve and calculate the projections index, i.e. $\rho^{(t)}(\boldsymbol{x}) = \rho_{f^{(t)}}(\boldsymbol{x})$.

Expectation: For each index, take the mean of data points projected onto it as the new curve point, i.e., $f^{(t+1)}(\rho) = E[\boldsymbol{X}|\rho_{f^{(t)}}(\boldsymbol{X}) = \rho]$.

The projection and expectation steps are repeated until a convergence criterion is met, e.g. when the change of the curve between iterations is below a threshold.

For a finite data set, the density p is often unknown, the above expectation is replaced by a smoothing method such as the locally weighted running-line smoother or smoothing splines. For kernel regression, the smoother is

$$f(\rho) = \frac{\sum\limits_{i=1}^{N} \boldsymbol{x}_i \kappa(\rho, \rho_i)}{\sum\limits_{i=1}^{N} \kappa(\rho, \rho_i)}. \tag{25}$$

The arc length is simply computed from the line segments. There are no proofs of convergence of the algorithm, but no convergence problems have been reported, though the algorithm is biased in some cases[10]. Banfield and Raftery[60] have modified the HS algorithm by taking the expectation of the residual of the projections in order to reduce the bias. Kegl *et al*[61] have proposed an incremental, e.g. segment by segment, and arc length constrained method for practical construction of principal curves.

Tibshirani[58] has introduced a semi-parametric model for the principal curve. A mixture model was used to estimate the noise along the curve; and the expectation and maximization (EM) method was employed to estimate the parameters. Other options for finding the nonlinear manifold include the GTM[62] and probabilistic principal surfaces[63]. These methods model the data by a means of a latent space. They belong to the semi-parameterized mixture model, although types and orientations of the local distributions vary from method to method.

6 Connections and comparisons between SOMs and others approaches

The SOM has been related to the discrete principal curve/surface algorithm[45]. However the differences remain in both the projection and the smoothing processes. In the SOM data are projected onto the nodes rather than onto the curve. The principal curves perform the smoothing entirely in the data space. The smoothing process in the SOM and ViSOM, as a convergence criterion, is[48],

$$\boldsymbol{w}_k = \frac{\sum\limits_{i=1}^{N} \boldsymbol{x}_i \eta(v,k,i)}{\sum\limits_{i=1}^{N} \eta(v,k,i)}. \tag{26}$$

The smoothing is governed by the indexes of the neurons in the map space. The kernel regression uses the arc length parameters $(\rho,\ \rho_i)$ or $||\rho-\rho_i||$ exactly, while the neighborhood function uses the node indexes (k,i) or $||k-i||$. Arc lengths reflect the curve distances between the data points in the data space. However, the node indexes are integer numbers denoting the nodes or the positions on the map grid, not the positions in the data space. So $||k-i||$ does not resemble $||\boldsymbol{w}_k-\boldsymbol{w}_i||$ in the SOM. In the ViSOM, however, as the inter-neuron distances on the map represent those in the data space (subject to the resolution of the map), the distances of nodes on the map are in proportion to the difference of their positions in the data space, i.e. $||k-i|| \approx ||\boldsymbol{w}_k-\boldsymbol{w}_i||$. The smoothing process in the ViSOM resembles that of the principal curves as shown below:

$$\boldsymbol{w}_k = \frac{\sum\limits_{i=1}^{N} \boldsymbol{x}_i \eta(v,k,i)}{\sum\limits_{i=1}^{N} \eta(v,k,i)} \approx \frac{\sum\limits_{i=1}^{N} \boldsymbol{x}_i \eta(\boldsymbol{w}_v,\boldsymbol{w}_k,i)}{\sum\limits_{i=1}^{N} \eta(\boldsymbol{w}_v,\boldsymbol{w}_k,i)}. \tag{27}$$

It shows that the ViSOM is a better approximation to the principal curves/surfaces than the SOM. The SOM and ViSOM are similar only when the data are uniformly distributed, or when the number of nodes becomes very large, in which case both the SOM and ViSOM will closely approximate the principal curves/surfaces.

The similarities between SOMs and metric MDS in terms of topographic mapping – mostly the qualitative likeness of the mapping results have been reported. However clear limitations of using the SOM for MDS have been noted[64]. Many applications combine the SOM and MDS for improved visualization of the SOM mapping results.

In [15], it is argued that the SOM is closer to MDS than to principal manifold. In [48], it is shown that the metric preserving ViSOM is a close approximate to a discrete principal manifold, and in [65] it is also shown that the ViSOM produces a similar mapping result as to the metric MDS.

Let's take a close look at the cost function of metric MDS, e.g. (14). Its denominator is a normalizing constant for all, while the numerator, which plays an important role in establishing the topological mapping, can be rewritten as[65]

$$\sum_{i,j}(d_{ij}-D_{ij})^2 = \sum_{i,j}(d_{ij}^2+D_{ij}^2-2d_{ij}D_{ij}). \tag{28}$$

The first term is a constant as data points are fixed and second term will be eventually fixed as it is to match the first term. To minimize the above stress is to maximize the third term (without the sign). The third term plays a dominant role and explains that the mapping is to form corresponding correlation between inter-distances in the original and mapped spaces. This is closely related to the C measure.

From the cost function of the SOM (5), we can see that the sample cost function – the integrand of (6), can be expressed as (for the data contained in Voronoi region i)

$$\sum_k \eta(i,k)\,||\boldsymbol{x}-\boldsymbol{w}_k||^2. \tag{29}$$

As $\boldsymbol{w}_k$ is the mean of Voronoi region k, let's denote it as $\bar{\boldsymbol{x}}_k$. Let's also denote $\bar{\boldsymbol{x}}_i$ as the mean of Voronoi region i. Furthermore $\eta(i,\ k)$ is a function of $||i-k||$. Then the above equation can be approximated as[65]

$$\sum_k \eta(i,k)\,||\boldsymbol{x}-\boldsymbol{w}_k||^2 = \sum_k f(||i-k||)\,||\bar{\boldsymbol{x}}_i-\bar{\boldsymbol{x}}_k||^2 = \sum_k f(D_{ik})d_{ik}^2. \tag{30}$$

For the SOM, $f(||i-k||)$ is simply the neighborhood function, typically an exponential function. The first term of its Taylor expansion is proportional to $-||i-k||^2$ (subject to the resolution parameter). This leads the above cost function with approximately[65],

$$\sum_k -(D_{ik}d_{ik})^2 \tag{31}$$

where D_{ik} represents the distance between the indexes of the neurons i and k on the map. Therefore the SOM preserves the correlation between the orders of the indexes of the neurons with the distances of their corresponding data regions in the input space. So that the largest d_{ik} matches the largest allowed D_{ik} on the grid. As the grid is not scalable, the data points will be mapped to these pre-fixed grid positions to achieve maximum correlation. This is a qualitative scaling and does not preserve the metric on the mapped space. This however can also be regarded as a kind of generalization as two spaces are no longer required to be in the same space. The SOM thus can be used to relate any two metric spaces by forming such a topological mapping.

In the ViSOM, as the $||i-k||$ is proportional to $||\boldsymbol{w}_i-\boldsymbol{w}_k||$, so D_{ik} is $D(\boldsymbol{w}_i,\boldsymbol{w}_k)$ and is a function of $||\boldsymbol{w}_i-\boldsymbol{w}_k||$, which is the mapped distance referred to the input space in the metric MDS sense. Thus this shows why the ViSOM produces similar scaling results as to MDS as observed in [21, 48] and many other reports. In other words it shows that the ViSOM is a metric MDS. The squared distance correlation terms in (31) have little different effect as to those non-squared ones of MDS in (28). The local distance preserving property of the ViSOM also enables it to capture highly nonlinear manifolds better compared to global distance preserving MDS.

As the ViSOM is a discrete principal manifold, at the same time it is also a MDS. This implies that MDS and principal manifold perform the same underlying task at least in the context of data visualization and dimension reduction. Finding a principal manifold – a smooth curve/surface passing through the middle of the data[10,66] – may well result in a topographic metric scaling of the input space onto the lower dimensional manifold. On other hand, although MDS presents a useful scaling of the data on a low dimensional space for visualization, it does not provide the underlying mapping function, the manifold.

A comparison between various classic mapping methods: PCA, Sammon mapping and SOM on a nonlinear "S" shape manifold is shown in Fig. 1. These methods have limited power in extracting highly nonlinear manifold. In Fig. 2 various recent methods such as the ViSOM, Isomap and LLE on embedding this manifold are demonstrated. The success of these methods can be easily seen, together with the superior performance of the ViSOM.

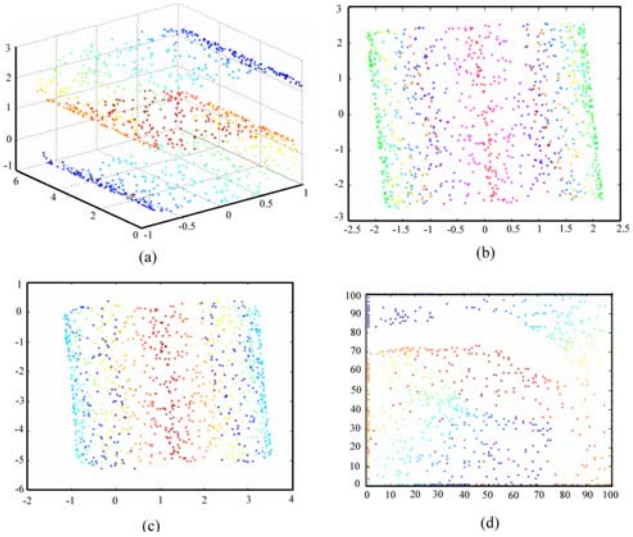


Fig. 1 Projection of S shape data((a)dataset; (b)PCA; (c)Sammon mapping; (d)SOM).

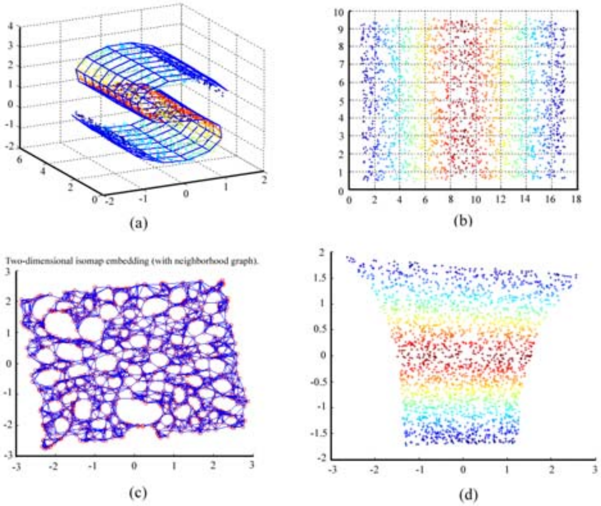


Fig. 2 Projections of S shape manifold((a)ViSOM embedding in the data space; (b)ViSOM projection; (c)Isomap; (d)LLE).

7 Conclusions

This paper provides a review on nonlinear dimensionality reduction and data visualization, from a self-organized approach. It also reviews the SOM and the issues surrounding its cost functions and topology measures, and reveals the connection between the SOM or its variant ViSOM and nonlinear principal manifolds and MDS through analyzing their weights and cost functions. Both the SOM and ViSOM are multidimensional scaling methods and produce nonlinear dimension-reduction mapping or manifold of the input space. The SOM is shown to be a qualitative scaling method, while the ViSOM is a metric scaling and approximates a discrete principal curve/surface. The SOMs can be seen as a generalized (and quantized) MDS that connecting or ordering two possibly different metric spaces.

It also reveals that metric MDS and principal manifold essentially produce the same topographic mapping for visualization. However metric MDS is a point to point mapping, while the principal manifold can establish an explicit mapping function.

References

[1] E. Condon, B. Golden, S. Lele, S. Raghavan, E. Wasil. A Visualization Model Based on Adjacency Data. *Decision Support Systems*, vol. 33, no. 4, pp. 349–362, 2002.

[2] H. Ni, H. Yin. Recurrent Self-organizing Maps and Local Support Vector Machine Models for Exchange Rate Prediction, In *Proceedings of International Symposium on Neural Networks*, Chengdu, China, vol. 3, pp. 504–511, 2006.

[3] R. Freeman, H. Yin. Web Content Management by Self-organization. *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1256–1268, 2005.

[4] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paetero, A. Saerelo. *Self-organization of Massive Text Document Collection, in Kohonen Maps*, E. Oja and S. Kaski (eds.), pp. 171–182, 1999.

[5] P. Toronen, K. Kolehmainen, G. Wong, E. Castren. Analysis of Gene Expression Data Using Self-organizing Maps. *FEBS Letters*, vol. 451, no. 2, pp. 142–146, 1999.

[6] E. C. Malthouse. Limitations of Nonlinear PCA as Performed with Generic Neural Networks. *IEEE Transactions on Neural Networks*, vol. 9, no. 1, pp. 165–173, 1998.

[7] M. A. Kramer. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.

[8] J. Karhunen, J. Joutsensalo. Generalization of Principal Component Analysis, Optimization Problems, and Neural Networks. *Neural Networks*, vol. 8, no. 4, pp. 549–562, 1995. [51] B.

[9] B. Schőlkopf, A. Smola, K. R. Müller, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, vol. 10, no. 5. pp. 1299–1319, 1998.

[10] T. Hastie, W. Stuetzle. Principal Curves. *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502–516, 1989.

[11] M. LeBlanc, R. J. Tibshirani. Adaptive Principal Surfaces. *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 53–64, 1994.

[12] J. B. Tenenbaum, V. de Silva, J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, vol. 290, pp. 2319–2323, 2000.

[13] S. T. Roweis, L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, vol. 290, pp. 2323–2326, 2000.

[14] T. F. Cox, M. A. A. Cox. *Multidimensional Scaling*, Chapman & Hall, 1994.

[15] B. D. Ripley. *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.

[16] J. W. Sammon. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computer*, vol. C-18, no. 5, pp. 401–409, 1969.

[17] J. Mao, A. K. Jain. Artificial Neural Networks for Feature Extraction and Multivariate Data Projection. *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 296–317, 1995.

[18] D. Lowe, M. E. Tipping. Feed-forward Neural Networks and Topographic Mappings for Exploratory Data Analysis. *Neural Computing & Applications*, vol. 4, no. 2, pp. 83–95, 1996.

[19] T. Kohonen. Self-organized Formation of Topologically Correct Feature Map. *Biological Cybernetics*, vol. 43, no. 1, pp. 56–69, 1982.

[20] T. Kohonen. *Self-organizing Maps*, Springer-Verlag, Berlin, 1997.

[21] H. Yin. ViSOM-A Novel Method for Multivariate Data Projection and Structure Visualization. *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 237–243, 2002.

[22] C. von der Malsburg, D. J. Willshaw. Self-organization of Orientation Sensitive Cells in the Striate Cortex. *Biological Cybernetic*, vol. 14, no. 2, pp. 85–100, 1973.

[23] D. J. Willshaw, C. von der Malsburg. How Patterned Neural Connections Can be Set Up by Self-Organization. *Proceedings of Royal Society of London, Series B, Biological Sciences*, vol. 194, no. 1117, pp. 431–445, 1976.

[24] T. Kohonen. *Self-organization and Associative Memory*, Springer-Verlag, Berlin, 1984.

[25] T. Kohonen. Representation of Sensory Information in Self-Organizing Feature Maps, and Relation of These Maps to Distributed Memory Networks, *Proceedings of SPIE*, vol. 634, pp. 248–259, 1986.

[26] H. Yin. Self-Organizing Maps: Statistical Analysis, Treatment and Applications, Ph.D. dissertation, University of York, UK, 1996.

[27] H. Yin, N. M. Allinson. On the Distribution and Convergence of the Feature Space in Self-organizing Maps, *Neural Computation*, vol. 7, pp. 1178–1187, 1995.

[28] S. P. Luttrell. Derivation of a Class of Training Algorithms. *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 229–232, 1990.

[29] S. P. Luttrell. A Bayesian Analysis of Self-organizing Maps. *Neural Computation*, vol. 6, no. 5, pp. 767–794, 1994.

[30] H. Ritter, K. Schulten. Convergence Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability, and Dimension Selection. *Biological Cybernetics*, vol. 60, no. 1, pp. 59–71, 1988.

[31] E. Erwin, K. Obermayer, K. Schulten. Self-organizing Maps: Ordering, Convergence Properties and Energy Functions. *Biological Cybernetics*, vol. 67, no. 1, pp. 47–55, 1992.

[32] E. Erwin, K. Obermayer, K. Schulten. Self-organizing Maps: Stationary States, Metastability and Convergence Rate. *Biological Cybernetics*, vol. 67, no. 1, pp. 35–45, 1992.

[33] Z.P. Lo, B. Bavarian. On the Rate of Convergence in Topology Preserving Neural Networks. *Biological Cybernetics*, vol. 65, no. 1, pp. 55–63, 1991.

[34] S. Lin, J. Si. Weight-value Convergence of the SOM Algorithm for Discrete Input. *Neural Computation*, vol. 10, no. 4, pp. 807–814, 1998.

[35] E. de Bolt, M. Cottrell, M. Verleysen. Statistical Tools to Assess the Reliability of Self-organising Maps. *Neural Networks*, vol. 15, no. 8–9, pp. 967–978, 2002.

[36] R. Durbin, G. Mitchison. A Dimension Reduction Framework for Understanding Cortical Maps. *Nature*, vol. 343, no. 6259, pp. 644–647, 1990.

[37] G. Mitchison. A Type of Duality Between Self-organizing Maps and Minimal Wiring. *Neural Computation*, vol. 7, no. 1, pp. 25–35, 1995.

[38] T. Kohonen. Self-organizing Maps: Optimization Approaches. *Artificial Neural Networks*, vol. 2, no. 5, pp. 981–990, 1991.

[39] J. Lampinen, E. Oja. Clustering Properties of Hierarchical Selforganizing Maps. *Journal of Mathematical Imaging and Vision*, vol. 2, no. 2-3, pp. 261–272, 1992.

[40] T. Heskes. *Energy Functions for Self-organizing Maps. Kohonen Maps*, E. Oja and S. Kaski (eds.), pp. 303–315, 1999.

[41] G. J. Goodhill, T. Sejnowski. A Unifying Objective Function for Topographic Mappings. *Neural Computation*, vol. 9, no. 6, pp. 1291–1303, 1997.

[42] H. -U. Bauer, K. R. Pawelzik. Quantifying the Neighborhood Preservation of Self-organizing Feature Maps. *IEEE Transactions on Neural Networks*, vol. 3, no. 4, pp. 570–579, 1992.

[43] T. Villmann, R. Der, M. Herrmann, T. M. Martinetz. Topology Preservation in Self-organizing Feature Maps: Exact Definition and Measurement. *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 256–266, 1997.

[44] S. Lawrence, C .L. Giles, A. C. Tsoi, A. D. Back. Face Recognition: A Convolutional Neural-network Approach. *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[45] H. Ritter, T. Martinetz, K. Schulten. *Neural Computation and Self-organizing Maps: An Introduction*, Addison Wesley Publishing Company, Danvers, 1992.

[46] H. Yin, N. M. Allinson. Interpolating Self-organizing Map (iSOM). *Electronics Letters*, vol. 35, no. 19, pp. 1649–1650, 1999.

[47] H. Yin. Resolution Enhancement for the ViSOM, In *Proceedings of the Workshop on Self-organizing Maps*, Kitakyushu, Japan, pp. 208–212. 2003.

[48] H. Yin. Data Visualization and Manifold Mapping Using the ViSOM. *Neural Networks*, vol. 15, no. 8–9, pp. 1005–1016, 2002.

[49] S. Wu, T. W. S. Chow. PRSOM: A New Visualization Method by Hybridizing Multidimensional Scaling and Self-organizing Map. *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1362–1380, 2005.

[50] P. A. Estévez, C. J. Figueroa. Online Data Visualization Using the Neural Gas Network. *Neural Networks*, vol. 19, no. 6–7, pp. 923–934, 2006.

[51] R. C. T. Lee, J. R. Slagle, H. Blum. A Triangulation Method for the Sequential Mapping of Points from N-Space to Two-Space. *IEEE Transactions on Computers*, vol. C-26, no. 3, pp. 288–292, 1977.

[52] D. de Ridder, R. P. W. Duin. Sammon′s Mapping Using Neural Networks: A Comparison. *Pattern Recognition Letters*, vol. 18, no. 11–13, pp. 1307–1316, 1997.

[53] M. Balasubramanian, E. L. Schwartz. The Isomap Algorithm and Topological Stability. *Science*, vol. 295, no. 5552, pp.7, 2002.

[54] E. Oja. Neural Networks, Principal Components, and Subspaces. *International Journal of Neural Systems*, vol. 1, no. 1, pp. 61–68, 1989.

[55] T. D. Sanger. Optimal Unsupervised Learning in a Single-layer Linear Feedforward Network. *Neural Networks*, vol. 2, no. 6, pp. 459–473, 1989.

[56] M. Belkin, P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[57] Y. Weiss. Segmentation Using Eigenvectors: A Unifying View. In *Proceedings of the 7th IEEE International Conference on Computer Vision*, Kerkyra, Greece, pp. 975–982, 1999.

[58] R. Tibshirani. Principal Curves Revisited. *Statistics and Computation*, vol. 2, no. 4, pp. 183–190, 1992.

[59] Y. Linde, A. Buzo, R. M. Gray. An Agorithm for Vctor Qantizer Dsign. *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[60] J. D. Banfield, A. E. Raftery. Ice Floe Identification in Satellite Images Using Mathematical Morphology and Clustering about Principal Curves. *Journal of the American Statistical Association*, vol. 87, no. 417, pp. 7–16, 1992.

[61] B. Kegl, A. Krzyzak, T. Linder, K. Zeger. A Polygonal Line Algorithm for Constructing Principal Curves, In *Proceeding of Neural Information Processing Systems*, Computer Press, Denver Colorado, USA, pp. 501–507, 1998.

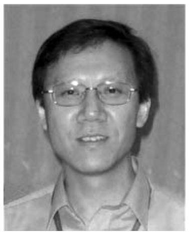
[62] C. M. Bishop, M. Svensn, C. K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, vol. 10, no. 1, pp. 215–235, 1998.

[63] K. Y. Chang, J. Ghosh. A Unified Model for Probabilistic Principal Surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 22–41, 2001.

[64] A. Flexer. Limitations of Self-organizing Maps for Vector Quantization and Multidimensional Scaling, In *Proceedings of Neural Information Processing Systems*, Computer Press, Denver Colorado, USA, pp. 445–451, 1997.

[65] H. Yin. Connection Between Self-organizing Maps and Metric Multidimensional Scaling, In *Proceedings of International Joint Conference on Neural Networks*, Orlando, Florida, USA, pp. 12–17, 2007.

[66] P. Delicado. Another Look at Principal Curves and Surfaces. *Journal of Multivariate Analysis*, vol. 77, no. 1, pp. 84–116, 2001.



Hujun Yin Senior lecturer at the University of Manchester, School of Electrical and Electronic Engineering. He received his B.Eng., M.Sc. from Southeast University and Ph.D. from University of York in 1983, 1986 and 1996 respectively. His research interests include neural networks, self-organizing systems in particular, pattern recognition, image processing and bioinformatics.

Dr. Yin has studied, extended and applied the self-organizing map (SOM) and related topics (principal manifolds and data visualization) extensively in the last ten years and proposed a number of extensions including the Bayesian SOM and ViSOM. He has served on the Programme Committee for more than twenty international conferences. He was the Organizing and Programme Committee Chair and General Chair for a number of conferences, including, International Workshop on Self-Organizing Maps (WSOM′01), International Conference on Intelligent Data Engineering and Automated Learning (IDEAL′03 –IDEAL′07), and International Symposium on Neural Networks (ISNN′04-ISNN′06). He sits on the Steering Committee of the WSOM series. He was a guest editor of *Neural Networks*: 2002 Special Issue on New Developments in Self-Organizing Maps, among two other special issues on two other international journals. He has received research funding from the EPSRC, BBSRC and DTI.

Dr. Yin has published more than 90 peer-reviewed articles. He is a senior member of the IEEE and a member of the EPSRC Peer Review College. He has also been a regular referee for the EPSRC, BBSRC and Royal Society grant proposals, Hong Kong Research Grant Councils, etherlands Organization for Scientific Research, and Slovakia Research and Development Council. He is an associate editor of the *IEEE Transactions on Neural Networks* and a member of the Editorial Board of the *International Journal of Neural Systems*.