PERGAMON

Neural Networks 15 (2002) 1005–1016

Neural Networks

www.elsevier.com/locate/neunet

2002 Special Issue

Data visualisation and manifold mapping using the ViSOM

Hujun Yin

Department of Electrical Engineering and Electronics, UMIST, P.O. Box 88, Manchester M60 1QD, UK

Abstract

The self-organising map (SOM) has been successfully employed as a nonparametric method for dimensionality reduction and data visualisation. However, for visualisation the SOM requires a colouring scheme to imprint the distances between neurons so that the clustering and boundaries can be seen. Even though the distributions of the data and structures of the clusters are not faithfully portrayed on the map. Recently an extended SOM, called the visualisation-induced SOM (ViSOM) has been proposed to directly preserve the distance information on the map, along with the topology. The ViSOM constrains the lateral contraction forces between neurons and hence regularises the inter-neuron distances so that distances between neurons in the data space are in proportion to those in the map space. This paper shows that it produces a smooth and graded mesh in the data space and captures the nonlinear manifold of the data. The relationships between the ViSOM and the principal curve/surface are analysed. The ViSOM represents a discrete principal curve or surface and is a natural algorithm for obtaining principal curves/surfaces. Guidelines for applying the ViSOM constraint and setting the resolution parameter are also provided, together with experimental results and comparisons with the SOM, Sammon mapping and principal curve methods. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Multidimensional scaling; Principal curves and surfaces; Multivariate data visualisation; Self-organising maps; Dimensionality reduction

1. Introduction

The demand for a meaningful understanding and visualisation of nonlinear multivariate data has never been higher, as operational data and experimental results are being accumulated at astonishing rates in many organisations. Searching for a suitable data projection method has always been an integral objective of multivariate data analysis and pattern recognition. Such a method should enable us to observe and detect underlying data distributions, patterns and structures. Good data analysis tools and methods will not only enable an in-depth view of the data but also reveal the underlying functions. A great deal of effort has been devoted to this subject and a number of useful methods have been proposed and consequently applied to various applications.

Classic projection methods include the linear principal component analysis (PCA) and the multidimensional scaling (MDS). The PCA projects the data onto its principal directions (usually the first or the first two, or any two 'interesting' components). The principal directions are represented by the principal, orthogonal eigenvectors of the covariance matrix of the data. It is the optimal linear projection in the sense of minimum mean-square-error between the original data points and the projected ones on the principal subspace. 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 (Oja, 1989), the generalised Hebbian algorithm (Sanger, 1991), and Rubner and Tavan's network (1989). But 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 dimension is much higher than two, the projection onto a linear plane will provide limited visualisation power. Extension to nonlinear PCA could tackle better, in principle, practical problems. However, there is no single and unique solution to nonlinear PCA (Malthouse, 1998). Various methods have been proposed, for example autoassociative networks (Kramer, 1991), generalised PCA (Karhunen & Joutsensalo, 1995), kernel PCA (Schölkopf, Smola, & Müller, 1998), and principal curves and surfaces (Hastie & Stuetzle, 1989; LeBlanc & Tibshirani, 1994). Other mapping methods include the recently proposed local, geometric based grouping and averaging (Tenenbaum, de Silva, & Langford, 2000) and local linear embedding (Roweis & Saul, 2000).

MDS tries to project data points onto a two-dimensional (2D) sheet by preserving as close as possible the inter-point

E-mail address: h.yin@umist.ac.uk (H. Yin).

PII: S0893-6080(02)00075-8

metrics. It is generally nonlinear and can reveal the overall structure of the data, but cannot provide the underlying mapping function. Sammon (1969) mapping is a widely known example of MDS. The objective of Sammon mapping is to minimise the differences between inter-point distances in the original space and those in the projected plane. The projection of data from an invisible high dimensional space to a low perceptible one can reveal the data structures and cluster tendency. The Sammon mapping has been shown to be useful for data structure analysis (e.g. Sammon, 1969; Ripley, 1996). However, like other MDS methods, the Sammon algorithm is a point-to-point mapping, which does not provide the explicit mapping function and cannot accommodate new data points (Sammon, 1969; Mao & Jain, 1995). For any additional data, the projection has to be re-calculated from scratch based on all data points. This proves 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.

Neural networks present another approach to nonlinear data analysis. They are biologically inspired learning and mapping methods and can learn complex nonlinear relationships of variables from sample data. Mao and Jain (1995) have given an overview on this subject. Kohonen's self-organising map (SOM) is an abstract mathematical model of the mapping between nerve sensory and cerebral cortex (Kohonen, 1982, 1995). As the map is often arranged in a low dimensional grid and the inputs are often drawn from a high dimensional space, the SOM has been used as a visualisation tool for dimensionality reduction (e.g. Ultsch, 1993; Kraaijveld, Mao, & Jain, 1995). One of the greatest properties of the SOM is its topology preservation, i.e. close points in the input space are mapped to nearby neurons in the map space. Such properties can be employed to visualise the *relative* or *qualitative* mutual relationships among the input. The SOM is also an abstraction process and it usually uses fewer representatives for an often large number of data points. Its distribution and convergence properties show that the SOM is naturally an optimal vector quantiser (VQ) in minimising the mean-square-error between reference vectors and data space (Luttrell, 1989, 1994; Yin & Allinson, 1995). The algorithm has found a wide range of applications in VQ, pattern classification, clustering, data mining and visualisation, knowledge discovery and information retrieval (cf. Kohonen, 1995).

However, when the SOM is used for visualisation, the inter-neuron distances are not directly visible or measurable on the map. A colouring scheme such as the *U*-matrix (Ultsch, 1993; Kraaijveld et al., 1995) has to be applied to the trained map for marking relative distances between the neurons according to the difference of their weights referred to the input space. Even so, the structures of data clusters are not apparent and often appear distorted. Although the Sammon mapping has been applied, in a post-processing step, to a trained SOM as a means of displaying the distances on the map (Törönen, Kolehmainen, Wong, & Castrén, 1999), the SOM does not *directly* apply to MDS, which aims to reproduce proximity in (Euclidean) distance on a low visualisation space (Cox & Cox, 1994; Ripley, 1996).

Recently a constrained SOM, termed the visualisation induced self-organising map (ViSOM), has been proposed by the author (Yin, 2001, 2002). The ViSOM projects the high dimensional data in an unsupervised manner as does the SOM, but constrains the lateral contraction force between the neurons and hence regularises the inter-neuron distances with respect to a scaleable 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 visualisation of both the structure and distribution of the data.

This paper provides a further analysis of the ViSOM, its implementation relating to applying the constraint and setting the resolution parameter, and its relationship with other nonlinear mapping methods. The ViSOM is a nonlinear projection for data visualisation but of a simple computational structure. The paper also links it with the principal curves or surfaces proposed by Hastie and Stuetzle (1989). The principal curve is a self-consistency smooth curve passing though the 'middle' points of the data. It is more of a notation than an actual algorithm. Unlike in the linear PCA case, the nonlinear manifold may not be unique and its existence depends on constraints and implementation. The subject has attracted much attention recently. Several algorithms have been proposed for solving for principal curves/surfaces. Some are nonparametric, e.g. the HS algorithm (Hastie & Stuetzle, 1989), while others are semi-parametric, e.g. the generative topographic mapping (GTM) (Bishop, Svensén, & Williams, 1998). The paper reveals that the ViSOM is a natural algorithm for constructing principal curves and surfaces. In Section 2, the ViSOM as an extension of the SOM for direct data visualisation is described, together with detailed explanations and guidelines on its constraint and the resolution parameters. Section 3 links the ViSOM with the principal curve algorithm from a kernel regression prospective, followed by several illustrative examples and experiments and comparisons with other methods in Section 4. Conclusions are given in Section 5.

2. ViSOM and data visualisation

Kohone's SOM is an unsupervised learning algorithm, which uses a finite grid or lattice of neurons to fill and frame the input data. Nodes are usually arranged in a 2D rectangular or hexagonal grid. In the SOM, a neighbourhood learning is adopted to form topological ordering among the neurons in the map. The close data points are likely to be projected to nearby nodes. Thus the map can be used to show the relative relationships among data points. However, the SOM does not directly show the inter-neuron distances on the map. For visualisation, the SOM requires assistance from some colouring scheme to imprint the inter-neuron distances and therefore the clusters and boundaries can be marked. The colour or grey tone of a node or a region between nodes is proportional to the mean or median of the distances between that node and its nearest neighbours. Such a colouring method has been used in many data visualisation applications, e.g. WEBSOM (Honkela, Kaski, Lagus, & Kohonen, 1997) and World Welfare Map (Kaski & Kohonen, 1996). The colouring methods indeed enhance the visualisation ability of the SOM. However, the cluster structures and distribution of the data shown on the map often are not apparent and appear in distorted and unnatural forms. Other techniques to mark the inter neuron distances include calculating magnification factors (Bishop, Svensén, & Williams, 1998) and interpolation (Yin & Allinson, 1999). The SOM can serve as a visualisation map only in showing the relative closeness and relationships among data points and clusters. In many cases, however, a direct and faithful display of structure shapes and distributions of the data is more desirable in visualisation applications.

2.1. ViSOM

For the map to capture the data structure naturally and directly, the distance quantity must be preserved on the map, along with the topology. Ideally the nodes should be uniformly and smoothly placed in the nonlinear manifold of the data space. The distances of any two nearest neighbouring neurons are approximately the same and the distances between a neuron and its further neighbouring neurons increase proportionally and regularly according to the structure of the map grid. So that the positions of the neurons can be served as grades for measuring the distance of any mapped points. The map can be seen as a smooth and graded mesh embedded into the data space, onto which the data points are mapped and the inter-point distances are approximately preserved.

The ViSOM uses the same grid structure of neurons as the SOM. Denoting the input $\mathbf{x} \in \mathbf{R}^n$ as a n-dimensional vector, node index as c ($c = (i,j) \in \Omega$ for a 2D map, where $i = 1, 2, \ldots, M$ and $j = 1, 2, \ldots, N$ for a $N \times M$ map), its associated weight vector as $\mathbf{w}_c = [w_{c1}, w_{c2}, \ldots, w_{cn}]^{\mathrm{T}}$, at time step t, the data input is $\mathbf{x}(t)$, learning rate is $\alpha(t)$, and the neighbourhood function is $\eta(v, c, t)$, where v represents the winner's index, then the basic ViSOM algorithm can be stated as follows (Yin, 2002).

The basic ViSOM algorithm

1. Initialise the map or weights either to the principal components or to small random values.
2. At time step t, given an input vector $\mathbf{x}(t)$, find the winner v according to,
$$v = \arg \min_{c \in \Omega} \|\mathbf{x}(t) - \mathbf{w}_c\| \tag{1}$$
3. Update the winner's weights according to,
$$\mathbf{w}_v(t+1) = \mathbf{w}_v(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_v(t)] \tag{2}$$
4. Update the weights of neighbouring neurons using
$$\begin{aligned}\mathbf{w}_k(t+1) &= \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t) \\ &\times\left([\mathbf{x}(t) - \mathbf{w}_v(t)] + [\mathbf{w}_v(t) - \mathbf{w}_k(t)]\left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1\right)\right)\end{aligned} \tag{3}$$
5. Refresh the map by randomly choosing a neuron and using its weight vector as the input for a small percentage of updating times (e.g. 20% iterations).
6. Repeat steps 2–5 until the map converges.

Here d_{vk} and Δ_{vk} are the distances between nodes v and k in the data space and on the map respectively (nodes are placed at the units of the map grid). The positive pre-specified resolution parameter, λ, specifies the scale of the map, i.e. the scaling of the unit distance on the map with respect to the data space. It can be chosen according to the variance or maximum scope of the data, and can also be made adaptive during the training. It represents the desired inter-neuron distance of two nearest neighbouring nodes reflected in the data space. The smaller the value of λ, the higher resolution the map can provide; subsequently a larger map and more neurons are required in order to cover the entire data space.

A refreshing phase is introduced to ensure the map's smooth expansion to those regions where there are few or no data points. Some nodes may seldom win from the direct input stimulation. Refreshing keeps these nodes active and also regularises the inter-neuron distances among these nodes.

The main difference between the SOM and ViSOM is the constraint, $\beta := d_{vk}/(\Delta_{vk}\lambda) - 1$. Without it the ViSOM becomes the SOM. In the SOM the updating force, $[\mathbf{x}(t) - \mathbf{w}_k(t)]$, can be decomposed into two components, $\mathcal{F}_{kx} := \mathbf{x}(t) - \mathbf{w}_k(t) = [\mathbf{x}(t) - \mathbf{w}_v(t)] + [\mathbf{w}_v(t) - \mathbf{w}_k(t)] := \mathcal{F}_{vx} + \mathcal{F}_{kv}$, as shown in Fig. 1(a). The first force, $\mathcal{F}_{vx}$, represents the updating force from the winner v to the input $\mathbf{x}(t)$, and is the same to the updating force used by the winner in Eq. (2). It adapts the neurons towards the input in a direction that is orthogonal to the tangent plane around the winner. While the second force, $\mathcal{F}_{kv}$, is a lateral force bringing neuron k to the winner v, i.e. a contraction force. It is this contraction force that brings neurons in the neighbourhood towards the winner and thus forms a contraction around the winner at each time step. The constraint is applied to this contraction force. The ViSOM regularises this force so that the distances between the nodes on the map are in proportion to the distances of their weights in the data space.

It can be seen that if the d_{vk} is larger than $\Delta_{vk}\lambda$, i.e. the $\mathbf{w}_k$

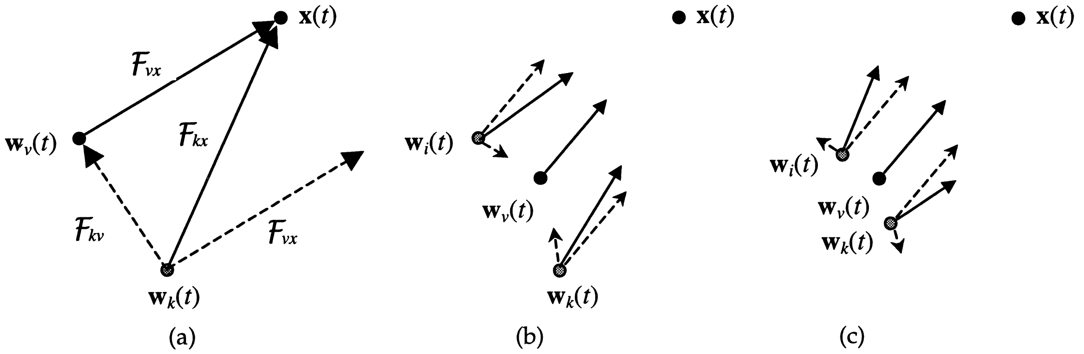


Fig. 1. (a) Decomposition of the SOM updating force, (b) contraction, (c) expansion.

is farther away from $\mathbf{w}_v$ under the specified resolution, the constraint is positive, so a contraction force remains. Otherwise, the constraint becomes negative, so an opposite or expansion force applies. These are shown in Fig. 1(b) and (c). The aim is to adjust inter-neuron distances on the map in proportion to those in the data space, i.e. $\Delta_{vk} \propto d_{vk}$. When the data points are eventually projected on a trained map, the distance between point i and j on the map is proportional to that of the original space, subject to the quantisation error (the distance between a data point and its neural representive). This has a similar effect to Sammon mapping, which also aims at achieving this proportionality, $D_{ij} \propto d_{ij}$, though here D_{ij} represents the distance of two mapped data points. When the number of nodes increases, the quantisation errors reduces, so the ViSOM becomes similar to the Sammon mapping. However, the Sammon mapping is a point to point mapping and a batch operation, so cannot provide a mapping function or accommodate new data points. Computationally the Sammon mapping is generally more costly, as it requires calculation of both first and second order gradients for each data point (Sammon, 1969). It also requires an $L \times L$ distance matrix, where L is the number of data point. This becomes disadvantageous or even impractical for a large data set. For example, for 1000 data points 1,000,000 variables are needed for holding all the inter-point distances and they are calculated at every iteration. The ViSOM, however, is an adaptive and effective method. It is also a principled approach as it can provide the mapping function. Another difference lies in the way of defining the scope where distance similarities are preserved. In the Sammon mapping, because it uses intermediate normalisation (one local and one global normalisers, see Biswas, Jain, & Dubes, 1981; Cox & Cox, 1994), it gives an averaged view of local distributions and global structure of the data. The ViSOM uses the neighbourhood function to provide a *flexible* control over the scope of the constraint. Large neighbourhood size means a broad constraint that makes most nodes comply with the distance proportionalities, equivalent to a global view of the data structure. A small scope provides a local constraint, leading to a detailed display of local distributions.

2.2. Smooth constraint and resolution parameter

In practice, it is not necessary to follow exactly the basic version of the algorithm listed in Section 2.1, as long as the ViSOM constraining principle is obeyed. The constraint can be introduced gradually for a smooth convergence. Define a smooth variable ξ that varies from 1 to 0 gradually with time during the training course, then one can replace the update rule Eq. (2) with the following one.

$$\begin{aligned}\mathbf{w}_k(t+1) &= \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t)(\mathcal{F}_{vx} + [\xi + (1-\xi)\beta]\mathcal{F}_{kv}) \\ &= \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t) \\ &\quad \times \left([\mathbf{x}(t) - \mathbf{w}_v(t)] + \left[\xi + (1-\xi)\left(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1 \right) \right] \right. \\ &\quad \left. \times [\mathbf{w}_v(t) - \mathbf{w}_k(t)] \right) \qquad (4)\end{aligned}$$

At the start of training, when ξ is close to 1, the ViSOM behaves almost like a SOM as almost no constraint is applied. Gradually with ξ decreasing with time, the constraint takes effect. Such an application of the constraint provides a smooth transition from the SOM to ViSOM.

The choice of resolution parameter λ depends on the size of the map and the variance or breadth of the data. It can also be a subjective figure, i.e. how fine does one want the representation. For a fixed resolution, a map of sufficient size has to be used in order to cover the entire data region. Otherwise, the map will only reveal the centre part of the data space correctly, and the boundary regions will be cramped on the edges of the map. If the size of the map is chosen a priori then the resolution parameter must be set appropriately. If it is too small, the map is too fine and does not cover the entire data space. While if it is too large the map becomes too coarse and over stretched to the outsides of the data space thus giving a poor resolution. For many data analysis problems, the maximum scope/span, $\text{Span}_{\max}$, or the maximum variance, $\text{Var}_{\max}$, of the data can be obtained beforehand then the resolution, λ, can be set to

$$\lambda = 1 \sim 1.5 \times \frac{\text{Span}_{\max}}{\min\{M, N\}} \tag{5a}$$

or

$$\lambda = 1 \sim 1.5 \times \frac{4 \times \sqrt{\text{Var}_{\max}}}{\min\{M, N\}} \tag{5b}$$

As practical data are likely to lie on a nonlinear manifold, such a measure based on spans or variances only provide a linear estimate of the stretch. A factor of $1 \sim 1.5$ is to allow additional stretch or curvature due to the nonlinearity. Larger factors can be used for stronger nonlinear cases. If no prior knowledge on data is available, the SOM can be used for the first few hundreds of iterations, then λ can be set to $2 \sim 5$ times of the average inter-neuron distance of the SOM. The ViSOM then comes into action.

Although the size of the map and the resolution are usually pre-specified and fixed, they both can be made adaptive during the training in order for the map to represent the entire data space effectively. Additional rows or columns can be either added or deleted for varying the size of the map as for the growing grid structure (Fritzke, 1995). One way to adapt the resolution is to monitor the firing activities and quantisation errors of the nodes. If these quantisation errors or activities along the edges are higher than those of inner nodes, it means that the map or resolution is too small. The resolution can be increased according to the quantisation errors. If the firing activities (on the data rather than refreshing weights) of all boundary nodes are low or zero, it means that the map or resolution is set too large, then the resolution should be decreased according to the range from the dormant nodes to the edges of the map.

The key feature of the ViSOM is that the distances between the neurons on the map (in a neighbourhood) reflect the corresponding distances in the data space. That is, the distant measure is preserved on the map. When the map is trained and data points mapped, the distances between mapped data points on the map will resemble approximately those in the original space (subject to the resolution of the map). This makes visualisation more direct, quantitatively measurable, and visually appealing. The size or covering range of the neighbourhood function can also be decreased from an initially large value to a smaller final one. The final neighbourhood, however, should not just contain the winner. The rigidity or curvature of the map is controlled by the ultimate size of the neighbourhood, σ_f. The larger of this size the flatter is the final map in the data space.

3. ViSOM and principal curves/surfaces

The extension from linear PCA to nonlinear PCA has not been straightforward due to the lack of a unified mathematical structure, efficient and reliable algorithms, and in some cases to excessive freedom in selection of representative basis functions (Malthouse, 1998). Several methods have been proposed for nonlinear PCA such as, a five-layer feedforward associative network (Kramer, 1991), a generalised Hebbian algorithm based method (Karhunen & Joutsensalo, 1995), and a kernel-based PCA (Schölkopf et al., 1998). The principal curves and principal surfaces (Hastie & Stuetzle, 1989; LeBlanc & Tibshirani, 1994) were primary nonlinear extension of PCA, but a valid algorithm is required for a good implementation. This section discusses the relationship between principal curves and the ViSOM.

3.1. Definition of principal curves

The principal curve was first defined by Hastie and Stuetzle (1989) as a smooth and self-consistency curve, which does not intersect itself. Denote $\mathbf{x}$ 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 $\mathbf{R}^n$, parametrised by the arc length ρ (from one end of the curve) over $\Lambda \in \mathbf{R}$, a closed interval.

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

$$\rho_f(\mathbf{x}) = \sup_{\rho \in \Lambda} \left\{ \rho : \|\mathbf{x} - f(\rho)\| = \inf_{\vartheta} \|\mathbf{x} - f(\vartheta)\| \right) \tag{6}$$

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

$$f(\rho) = E[\mathbf{X}|\rho_f(\mathbf{X}) = \rho] \tag{7}$$

The principal component is a special case of the principal curves if the distribution is ellipsoidal. Although 1D principal curves have been mainly studied, extension to higher dimension, e.g. principal surfaces is feasible in principle. However, in practice, a good implementation of principal curves/surfaces relies on an effective and efficient algorithm.

3.2. Principal curve algorithms

The principal curves/surfaces are more of a concept that invites practical implementations. The HS algorithm is a nonparametric method (Hastie & Stuetzle, 1989), which directly iterates the two steps of the above definition. It is similar to the LBG VQ algorithm (Linde, Buzo, & Gray, 1980) combined with some smoothing techniques.

HS algorithm:

Initialisation: Choose the first linear principal component as the initial curve, $f^{(0)}(\mathbf{x})$.
Projection: Project the data points onto the current curve and calculate the projections index, i.e. $\rho^{(t)}(\mathbf{x}) = \rho_{f(t)}(\mathbf{x})$.
Expectation: For each index, take the mean of data points projected onto it as the new curve point, i.e. $f^{(t+1)}(\rho) = E[\mathbf{X}|\rho_{f(t)}(\mathbf{X}) = \rho]$.

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 finite data, the density ρ is often unknown, the expectation step is replaced by a smoothing method such as the locally weighted running-line smoother or smoothing splines. 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 (Hastie & Stuetzle, 1989). Banfield and Raftery (1992) have modified the HS algorithm by taking the expectation of the residual of the projections in order to reduce the bias. Kegl, Krzyzak, Linder, and Zeger (1998) have proposed an incremental, e.g. segment by segment, and arc length constrained method for practical construction of principal curves.

Tibshirani (1992) 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 maximisation (EM) method was employed to estimate the parameters. Other options for finding the nonlinear manifold include the GTM (Bishop et al., 1998) and probabilistic principal surfaces (PPS) (Chang & Ghosh, 1999). These methods model the data by a means of a latent space. They belong to the semi-parametrised mixture model, although types and orientations of the local distributions vary from method to method.

3.3. Discrete principal curves/surfaces

The SOM has also been related to the discrete principal curve/surface algorithm (Ritter, Martinetz, & Schulten, 1992; Mulier & Cherkassky, 1995; Der, Balzuweit, & Herrmann, 1996). However, the difference remains in the projection process as it is known. In the SOM the data are projected onto the nodes rather than onto the curve (Kegl, Krzyzak, Linder, & Zeger, 2001). In the following, it is shown that a difference also exists in the expectation or smoothing step. The SOM's neighbourhood smoothing is governed by the indexes of the neurons, while the principal curves perform the smoothing entirely in the data space. First, comparing the kernel smoothing used by the principal curves, Eq. (8), and the equivalent batch SOM's kernel function (Mulier & Cherkassky, 1995), Eq. (9),

$$\text{Kernel regression}: \quad F(\rho) = \frac{\sum_{i=1}^{L} \mathbf{x}_i \kappa(\rho, \rho_i)}{\sum_{i=1}^{L} \kappa(\rho, \rho_i)} \tag{8}$$

$$\text{SOM}: \quad \mathbf{w}_k = \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(v, k, i)}{\sum_{i=1}^{L} \eta(v, k, i)} \tag{9}$$

Here the kernel function $\kappa(\cdot)$ and neighbourhood function $\eta(\cdot)$ are similar and often chosen from a symmetrical function family such as Gaussian. L denotes the number of data points.

Eq. (9) is also valid or at least as a convergence criterion for the common recursive SOM and ViSOM. The proof is given in Appendix A.

The kernel regression uses the arc length parameters (ρ, ρ_i) or $\|\rho - \rho_i\|$ exactly, while the neighbourhood function uses the node indexes (k, i) or $\|k - i\|$. Arc lengths reflect the curve distances between the data points. However, node indexes are integer numbers denoting the nodes not the positions of the nodes, so $\|k - i\|$ does not resemble $\|\mathbf{w}_k - \mathbf{w}_i\|$. The two smoothing functions, therefore, differ in their effects despite their similar appearance.

In the ViSOM as the inter-neuron distances on the map represent those in the data space (subject to the resolution of the map), the difference of node indexes are in proportion to the difference of their positions in the data space, i.e. $\|k - i\| \sim \|\mathbf{w}_k - \mathbf{w}_i\|$. The smoothing process in the ViSOM resembles that of the principal curves, as shown below,

$$\text{ViSOM}: \quad \mathbf{w}_k = \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(v, k, i)}{\sum_{i=1}^{L} \eta(v, k, i)} \approx \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(\mathbf{w}_v, \mathbf{w}_k, i)}{\sum_{i=1}^{L} \eta(\mathbf{w}_v, \mathbf{w}_k, i)} \tag{10}$$

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, and/or when the number of nodes becomes large, in which case both the SOM and ViSOM will closely approximate the principal curves/surfaces.

When sufficient number of nodes is used, the resolution of the ViSOM will be high. The line segment between two nearest neurons will be small. The projection onto the nearest segment is little different to the projection to any of its end nodes, the ViSOM emulates a natural principal curve algorithm. Another point worth noting is that the SOM is an entropy (or density) related, thus non-uniform, quantiser, while the ViSOM is a uniform quantiser. Smoothing for equally spaced knots becomes simpler (Phillips & Taylor, 1973).

Both the SOM and HS algorithms need a post-adjustment for the end points of a learnt curve, as such a point has an unbalanced smoothing due to the lack of data to one side of the point. The ViSOM has an ability of extending beyond the end points. Most principal curve algorithms such as HS algorithm are mainly proposed for solving 1D curves. There have been few published attempts to extend to higher dimensions. Semi-parametrised methods such as the GTM can be directly applied to higher dimensions. However, the number of model parameters increases with the data dimensions. Instead, the ViSOM or SOM-based methods

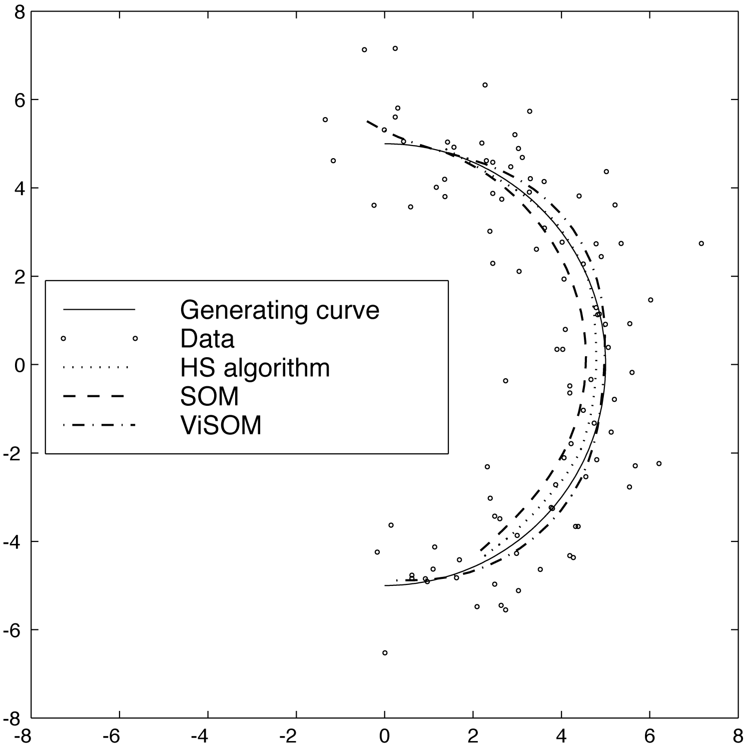


Fig. 2. Comparison of the HS algorithm, SOM and ViSOM for principal curve.

are much simpler in computational complexity and can be easily and more readily applied to tackling high dimensional nonlinear manifolds.

4. Experiments

Several experiments have been conducted and their results are presented here. The first is to demonstrate the similarities and differences of the HS algorithm, SOM and ViSOM in learning a nonlinear principal curve. The remaining shows the usefulness of the ViSOM in mapping manifolds and visualising multivariate data and its advantages over other methods.

The first example used a set of 100 data points uniformly distributed along a half circle of radius, $r = 5$, with further additional disturbance of a zero-mean, unit-variance and uncorrelated Gaussian. Three methods, the HS algorithm, SOM and ViSOM, produced similar performance, though the SOM exhibits more bias than the other two, as shown in Fig. 2. In this experiment, both the SOM and ViSOM used a 40-neuron chain. The resolution parameter, λ, of the ViSOM was set to 0.4, so that the chain can cover entire data span. The length of the final ViSOM chain, i.e. $40 \times 0.4 = 16$, matches the total arc length of the half circle, $\pi r = 15.7$. As can be seen unlike the SOM and HS algorithms, the ViSOM does not suffer from a 'contraction' problem at the end points. If the positions of all nodes had been displayed, it would be seen that the nodes of the ViSOM are uniformly distributed along the learnt curve; while the nodes of the SOM and the vertexes of the HS algorithm are spread uniformly only in the middle part of the curves but appear squeezed at each end.

The second data set is 1000 3D points, shown in Fig. 3. Data are distributed along a nonlinear surface (sine wave) with some disturbances (small normal distributions). A 20×20 ViSOM was used, with a resolution set according to Eq. (5b) and a data span of 10 obtained from the data (i.e. $\lambda = 1.5 \times 10/20 = 0.75$). The smoothing constraint, i.e. Eq. (4), was applied after the first 1000 iterations of the normal SOM. The smoothing variable ξ was set to be $500/(t - 500)$. The resultant ViSOM grid is shown in Fig. 3. The ViSOM has captured well the nonlinear manifold.

The next application used the well-known benchmark of Fisher's Iris dataset, made of 150 4D vectors from three Iris categories, each of which has 50 examples (Fisher, 1936). A 100×100 hexagonal ViSOM was applied to the data set and the result (i.e. the projected data on the map) is shown in Fig. 4(d). For comparison, a SOM of the same size and structure has also been applied to map the data and the result, after further applying the U-matrix colouring method, is presented in Fig. 4(c). The results of the PCA and Sammon mapping are shown in Fig. 4(a) and (b), respectively. The initial states of the Sammon mapping, SOM and ViSOM were all placed on a plane spanned by the first two principal components of the data. The final neighbourhood size, σ_f, for the ViSOM was set to 4, and resolution $\lambda = 0.075$ (the maximum inter-data distance is 7.085). As can be seen, the ViSOM result closely resembles that of the Sammon mapping.

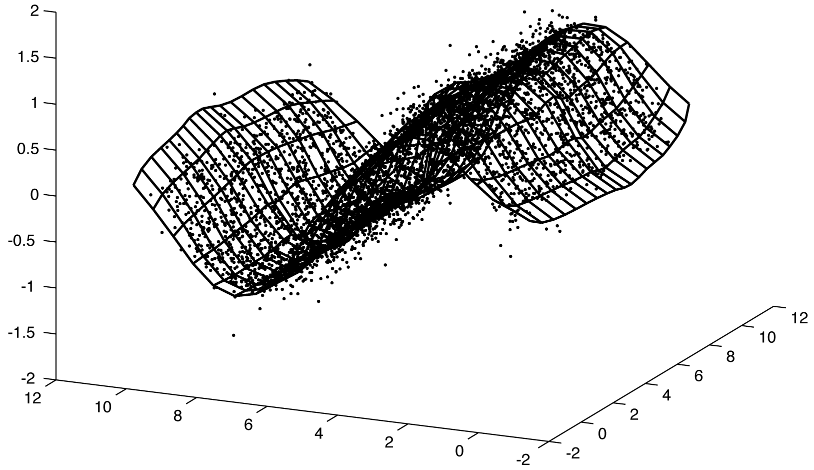
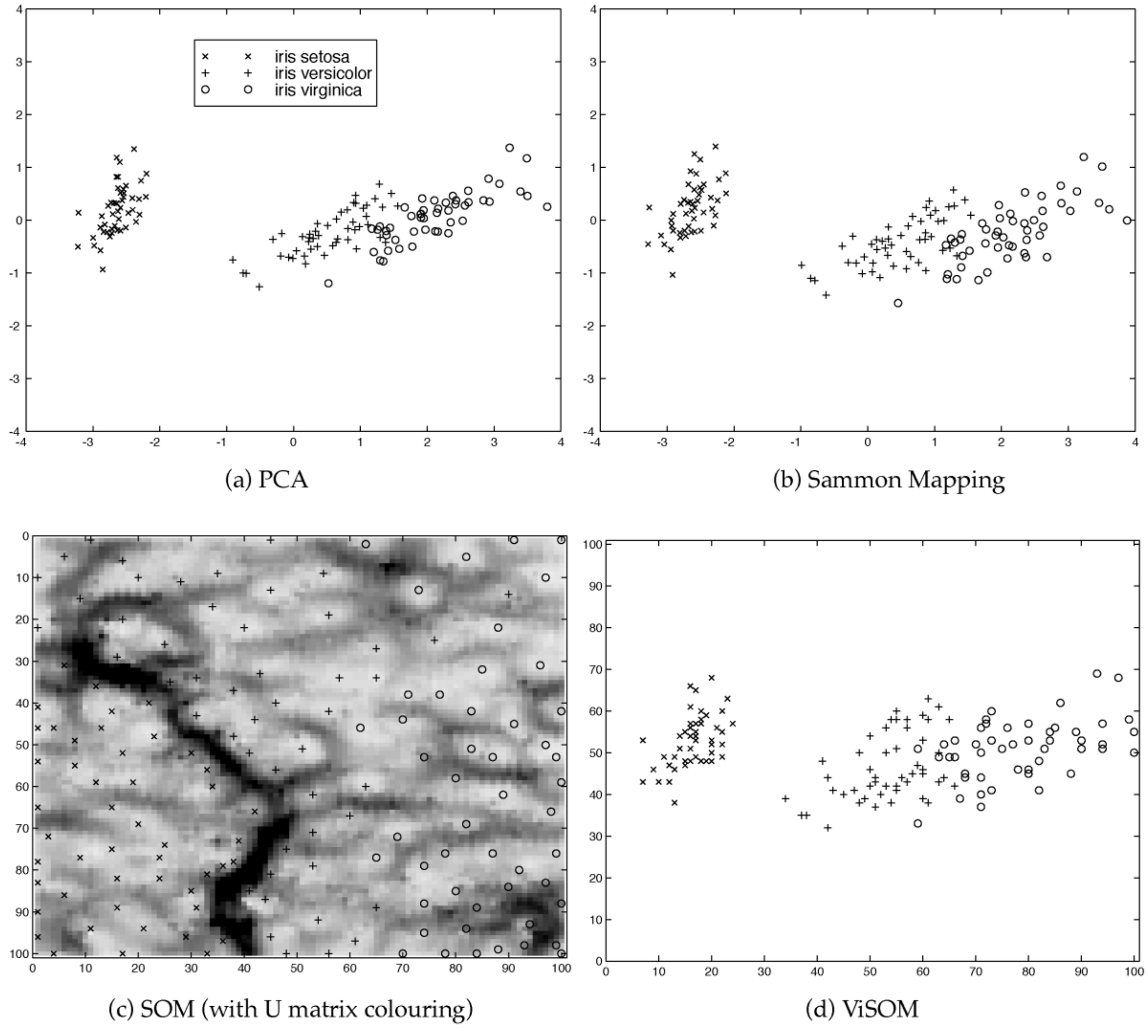


Fig. 3. The ViSOM for a 3D data set.



(a) PCA

(b) Sammon Mapping

(c) SOM (with U matrix colouring)

(d) ViSOM

Fig. 4. Mappings of Iris data set: (a) PCA, (b) Sammon mapping, (c) SOM with *U*-matrix colouring, (d) ViSOM. Map size for both SOM and ViSOM is 100×100, $\lambda = 0.075$, $\sigma_f = 4$.

Table 1
Top quarter of the 2000 ranking of UK universities (source: The Sunday Times, 18 September 2000). F1–F7 are teaching quality, research quality, A-level points, employment, First/2:1s awarded student–staff ratio and dropout rate, respectively

Ranking	University	*F*1	*F*2	*F*3	*F*4	*F*5	*F*6	*F*7	Total
1	Cambridge	241	182	247	97	88	100	50	1005
2	Oxford	214	175	244	97	81	100	30	941
3	LSE	200	175	233	97	68	100	50	923
4	Imperial	203	154	232	98	67	100	10	864
5	York	206	143	208	94	63	76	60	850
6	UCL	172	152	210	95	71	100	30	830
7	St Andrews	139	131	194	96	73	91	100	824
8	Warwick	153	155	215	97	69	86	20	795
9	Bath	132	142	211	97	66	83	60	791
9	Nottingham	176	125	218	96	74	72	30	791
11	Bristol	145	131	218	96	75	94	20	779
11	Durham	163	132	207	91	64	72	50	779
11	Edinburg	106	145	218	96	74	100	40	779
14	Lancaster	156	144	186	95	62	63	50	756
15	UMIST	135	144	188	97	58	100	30	752
16	Birmingham	146	127	204	96	67	87	20	747
17	Loughborough	162	115	177	95	57	66	60	732
18	Southampton	143	124	180	93	55	71	50	716
19	King's College	135	126	204	96	63	100	− 10	714
20	Newcastle	134	117	193	97	60	87	20	708
21	Manchester	125	134	198	96	66	98	− 10	707
22	Leeds	122	127	199	97	61	74	20	700
23	Sheffield	143	125	213	97	61	72	− 20	691
24	East Anglia	125	127	176	96	63	60	40	687
24	Leicester	125	120	183	94	52	93	20	687

The Sammon method is better than the linear PCA in revealing fine structural details. The ViSOM produces a similar display to the Sammon mapping that not only preserves the inter-cluster structures but also captures the details of intra-cluster and inter-point distributions. An important point, however, is that the ViSOM can provide the projection function, so that any new data points can find their appropriate places on the map, but this is not the case for the Sammon map. Moreover, the Sammon mapping is computationally more intensive, since it requires the first- and second-order derivatives of the stress function at each iteration; and it is also sensitive to the initial position. The standard SOM can show the cluster boundaries with the help of a colouring scheme, but it is impossible for the SOM to reveal inter-cluster and intra-cluster distributions.

In multivariate data analysis, tabling is one of the traditional ways to rank and compare individual members. All attributes of a member are summed, and a table of all members in either ascending or descending order of the sum is often used. Such an example is the league table of UK universities. Each year, the UK Sunday Times newspaper collects data ranging from teaching quality, research achievement and entry-level, to employment rate and dropout rate for each university in the UK. For the September 2000 version, seven factors were considered for the 99 higher education institutions. The top quarter of the 2000 league table is shown in Table 1. Weighting has been taken into account in the data by the newspaper. For example, weightings for teaching, research and A-level entrance points are 2.5, 2.0 and 2.5, respectively. Other factors have a weighting of 1.0. The Sammon mapping and ViSOM have been applied to this data set. The results provide an alternative 'see wood for the trees' view to a simple ranking.

Universities in the UK fall into two basic types, those whose foundations predate or those after the UK Government's decision in 1992 to convert all polytechnics into fully accredited universities. These are generally referred to as the 'pre-92' and 'post-92' universities. Both results, shown in Figs. 5 and 6, indicate that there are two large distinctive clusters and a clear gap between them. All pre-92 universities appear in the upper cluster, although some have dropped to the lower half of the table. For example, Salford University, an old university, is listed 64th in the league table. A few, which stand out of the 'pack', are the 'Golden Triangle' of world-class institutions such as Cambridge University, Oxford University, London School of Economics and Political Science (LSE) and Imperial College. Most post-92 or 'new' universities are in the lower stretched cluster. The similarities between the universities can be directly quantified according to the distances between them. Such a direct and global view of cluster form and division cannot be revealed in the tabling method, nor by the usual SOM algorithm.

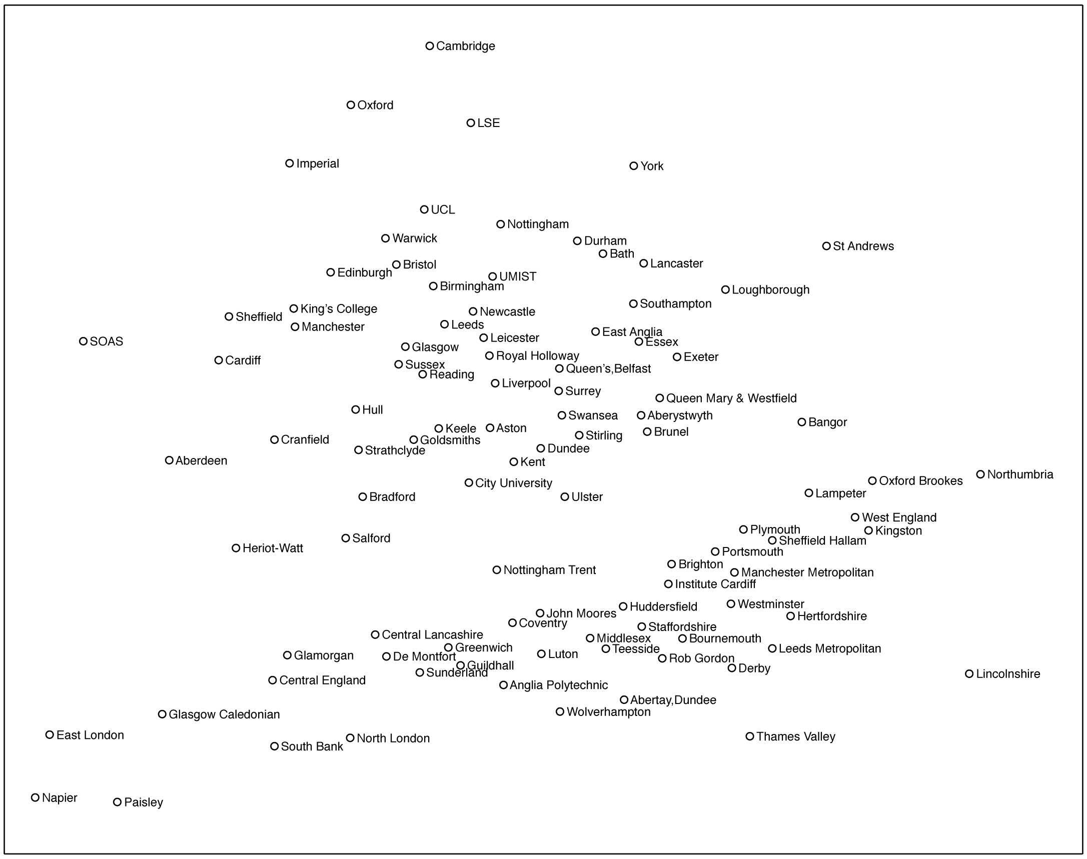


Fig. 5. Sammon mapping of the 2000 league table of the UK universities.

For the ViSOM, a 50×50 map was used, with $\lambda = 8$ ($\text{Span}_{\max} = 380.1$), $\sigma_f = 10$. The results produced by the two methods are very similar, although some local distributions differ slightly. The ViSOM is a principled approach as it provides a (discrete) mapping function. It projects and visualises the data onto the obtained nonlinear manifold. This has advantages over the point-to-point Sammon mapping. If a data point (a University in this particular example) was initially missed in the survey, then it can be added to the map by simply projecting it on the trained ViSOM map. The ViSOM can be implemented in parallel, while the Sammon mapping is basically a batch and numerical method.

5. Conclusions

In this paper, the recently proposed variant of the SOM, namely the ViSOM, is analysed for multivariate data visualisation and nonlinear manifold projections of high dimensional data. It has been compared with Sammon mapping and formally related to principal curves/surfaces-the principal description of the nonlinear manifold. The ViSOM is similar in structure to the SOM, but constrains the lateral contraction force within the updating neighbourhood. As a result, the map preserves the inter-neuron distances as well as the topology as faithfully as possible. The ViSOM produces a smooth and evenly graded mesh through the data points and enables a quantitative, direct and visually appealing measure of inter-point distances. The ViSOM has been shown to have a similar capability to the Sammon mapping in preserving the distributions on the map and can be considered as a principled or functional approach to MDS. The ViSOM is a natural algorithm for finding the principal curve/surface. Its distance preserving property makes the effect of the neighbourhood function equivalent to the kernel smoothing of principal curves/surfaces. Guidelines have also been given to the practical implementation of the ViSOM, and original ViSOM algorithm has been enhanced. Various examples and applications confirm the potential and usefulness of the ViSOM for multivariate data visualisation and nonlinear principal projections.

Acknowledgments

The author wishes to thank the reviewers for their valuable and useful comments on the manuscript.

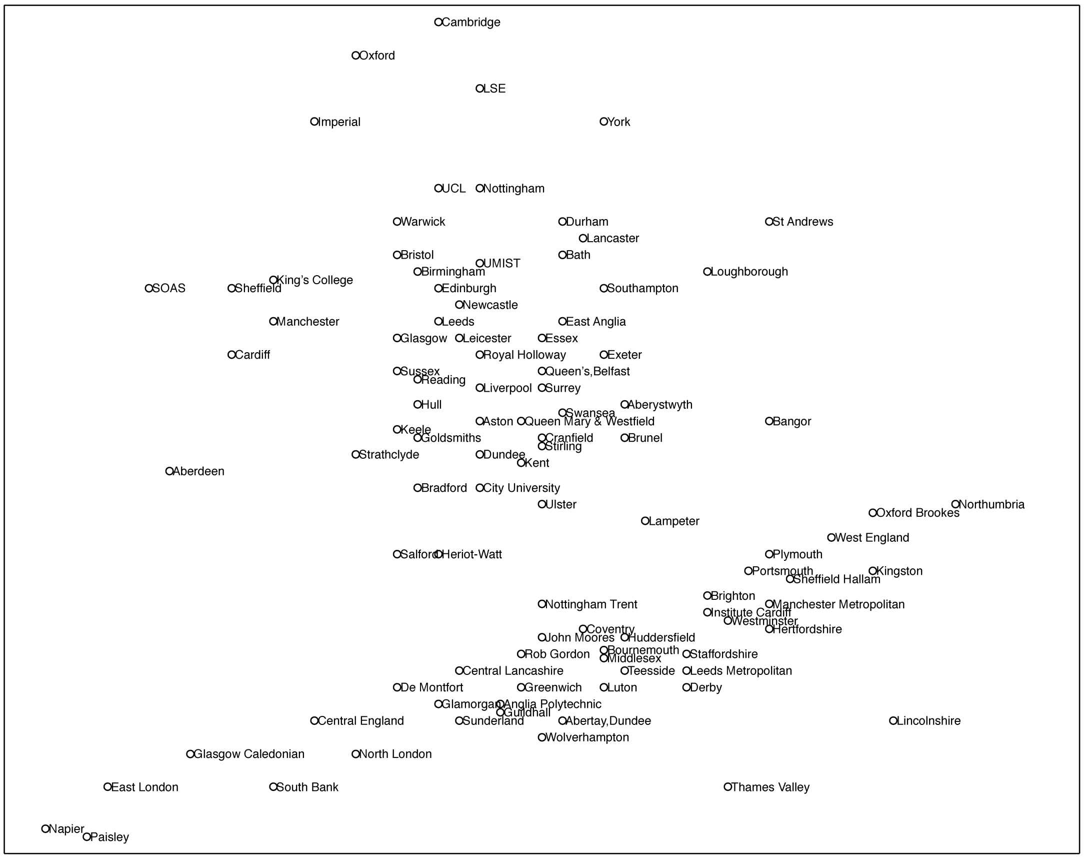


Fig. 6. ViSOM of the 2000 league table of the UK universities.

Appendix A

A.1. Weight convergence of the recursive SOM

Recalling the SOM updating rule, $\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha\eta(v,k,t)[\mathbf{x}(t) - \mathbf{w}_k(t)]$. If we re-arrange it as,

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(0 - \eta(v,k,t)[\mathbf{w}_k(t) - \mathbf{x}(t)])$$

This can be considered as equivalent to a Robbins–Monro stochastic gradient method (Robbins & Monro, 1951), from which, in reverse order, one can observe that the term, $\eta(v,k,t)[\mathbf{w}_k(t) - \mathbf{x}(t)] := \mathbf{y}_k$, is the instantaneous gradient for $\mathbf{w}_k(t)$, and the true gradient, which is the mean of $\{\mathbf{y}_k(t)\}$, i.e. $\int \mathbf{y}_k(t)\mathrm{d}t$ or discrete form $\sum_{t=1}^{T} \mathbf{y}_k(t)$, equals to zero. That is,

$$\sum_{t=1}^{T} \mathbf{y}_k(t) = \sum_{t=1}^{T} \eta(v,k,t)[\mathbf{w}_k(t) - \mathbf{x}(t)] := 0$$

$$\sum_{t=1}^{T} \eta(v,k,t)\mathbf{w}_k(t) = \sum_{t=1}^{T} \eta(v,k,t)\mathbf{x}(t)$$

With $\mathbf{w}_k(t)$ converging to $\mathbf{w}_k$, the earlier equation can be written as,

$$\mathbf{w}_k \rightarrow \frac{\sum_{t=1}^{T} \eta(v,k,t)\mathbf{x}(t)}{\sum_{t=1}^{T} \eta(v,k,t)}, \; k \in \Omega \tag{A1}$$

And hence it follows Eq. (9).

A.2. Weight convergence of the recursive ViSOM

With the equal distance constraint, the nodes become regularised or evenly placed and the lateral force becomes small or negligible. Then the ViSOM updating rule, Eq. (3), tends to, $\mathbf{w}_k(t+1) \approx \mathbf{w}_k(t) + \alpha\eta(v,k,t)[\mathbf{x}(t) - \mathbf{w}_v(t)]$. As above, we can re-arrange it as,

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(0 - \eta(v,k,t)[\mathbf{w}_v(t) - \mathbf{x}(t)])$$

From the earlier proof, we can see that this adaptation leads to,

$$\mathbf{w}_v \rightarrow \frac{\sum_{t=1}^{T} \eta(v,k,t)\mathbf{x}(t)}{\sum_{t=1}^{T} \eta(v,k,t)}, \ v \in \Omega \tag{A2}$$

References

Banfield, J. D., & Raftery, A. E. (1992). Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association*, *87*, 7–16.

Bishop, C. M., Svensén, M., & Williams, C. K. I. (1998). GTM: The generative topographic mapping. *Neural Computation*, *10*, 215–235.

Biswas, G., Jain, A. K., & Dubes, R. C. (1981). Evaluation of project algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-3*, 701–708.

Chang, K.-Y., & Ghosh, J. (1999). Probabilistic principal surfaces. *Proceedings 1999 IEEE International Conference on Neural Networks*. Washington DC, July.

Cox, T. F., & Cox, M. A. A. (1994). *Multidimensional scaling*. London: Chapman and Hall.

Der, R., Balzuweit, G., & Herrmann, M. (1996). Constructing principal manifolds in sparse data sets by self-organising maps with self-regulating neighbourhood width. *Proceedings of the International Conference on Neural Networks*, 480–483.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics*, *7*, 178–188.

Fritzke, B. (1995). Growing grid—A self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, *2*, 9–13.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, *84*, 502–516.

Honkela, T., Kaski, S., Lagus, K., & Kohonen, T. (1997). WEBSOM-self-organizing maps of document collections. *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*. Espoo, Finland, June 4–6 (pp. 310–315).

Karhunen, J., & Joutsensalo, J. (1995). Generalisation of principal component analysis, optimisation problems, and neural networks. *Neural Networks*, *8*, 549–562.

Kaski, S., & Kohonen, T. (1996). Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In A.-P. N. Refenes, Y. Abu-Mostafa, J. Moody, & A. Weigend (Eds.), *Neural networks in financial engineering* (pp. 498–507). Singapore: World Scientific.

Kegl, B., Krzyzak, A., Linder, T., & Zeger, K. (1998). A polygonal line algorithm for constructing principal curves. *Neural Information Processing Systems (NIPS'98)*, 11, (pp. 501–507).

Kegl, B., Krzyzak, A., Linder, T., & Zeger, K. (2001). Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-22*, 281–292.

Kohonen, T. (1982). Self-organised formation of topologically correct feature map. *Biological Cybernetics*, *43*, 56–69.

Kohonen, T. (1995). *Self-organising maps* (2nd ed). Berlin: Springer.

Kraaijveld, M. A., Mao, J., & Jain, A. K. (1995). A nonlinear projection method based on Kohonen's topology preserving maps. *IEEE Transactions on Neural Networks*, *6*, 548–559.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AICHE Journal*, *37*, 233–243.

LeBlanc, M., & Tibshirani, R. J. (1994). Adaptive principal surfaces. *Journal of the American Statistical Association*, *89*, 53–64.

Linde, Y., Buzo, A., & Gray, R. M. (1980). An algorithm for vector quantiser design. *IEEE Transactions on Communications*, *28*, 84–95.

Luttrell, S. P. (1989). Hierarchical self-organising networks. *Proceedings of IEE International Conference on Artificial Neural Networks* (pp. 2–6).

Luttrell, S. P. (1994). A Bayesian analysis of self-organising map. *Neural Computation*, *6*, 767–794.

Malthouse, E. C. (1998). Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Transactions on Neural Networks*, *9*, 165–173.

Mao, J., & Jain, A. K. (1995). Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, *6*, 296–317.

Mulier, F., & Cherkassky, V. (1995). Self-organisation as an iterative kernel smoothing process. *Neural Computation*, *7*, 1165–1177.

Oja, E. (1989). Neural networks, principal components, and subspaces. *International Journal of Neural Systems*, *1*, 61–68.

Phillips, G. M., & Taylor, P. J. (1973). *Theory and applications of numerical analysis*. New York: Academic Press.

Ripley, B. D. (1996). *Pattern recognition and neural network*. Cambridge, UK: Cambridge University Press.

Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural computation and self-organising maps: An introduction*. Reading, MA: Addison-Wesley.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*, 400–407.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*, 2323–2326.

Rubner, J., & Tavan, P. (1989). A self-organising network for principal component analysis. *Europhysics Letters*, *10*, 693–698.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computer*, *18*, 401–409.

Sanger, T. D. (1991). Optimal unsupervised learning in a single-layer linear feedforward network. *Neural Networks*, *2*, 459–473.

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*, 2319–2323.

Tibshirani, R. (1992). Principal curves revisited. *Statistics and Computation*, *2*, 183–190.

Törönen, P., Kolehmainen, K., Wong, G., & Castrén, E. (1999). Analysis of gene expression data using self-organising maps. *FEBS Letters*, *451*, 142–146.

Ultsch, A. (1993). Self-organising neural networks for visualisation and classification. In O. Opitz, B. Lausen, & R. Klar (Eds.), *Information and classification* (pp. 864–867).

Yin, H. (2001). Visualisation induced SOM (ViSOM). In N. Allinson, H. Yin, L. Allinson, & J. Slack (Eds.), *Advances in self-organising maps* (pp. 81–88). *Proceedings WSOM'01*, London: Springer.

Yin, H. (2002). ViSOM—A novel method for multivariate data projection and structure visualisation. *IEEE Transactions on Neural Networks*, *13*, 237–243.

Yin, H., & Allinson, N. M. (1995). On the distribution and convergence of feature space in self-organising maps. *Neural Computation*, *7*, 1178–1187.

Yin, H., & Allinson, N. M. (1999). Interpolating self-organising map (iSOM). *Electronics Letters*, *35*, 1649–1650.