3

Learning Nonlinear Principal Manifolds by Self-Organising Maps

Hujun Yin

The University of Manchester, Manchester, M60 1QD, UK,
hujun.yin@manchester.ac.uk

Summary. This chapter provides an overview on the self-organised map (SOM) in the context of manifold mapping. It first reviews the background of the SOM and issues on its cost function and topology measures. Then its variant, the visualisation induced SOM (ViSOM) proposed for preserving local metric on the map, is introduced and reviewed for data visualisation. The relationships among the SOM, ViSOM, multidimensional scaling, and principal curves are analysed and discussed. Both the SOM and ViSOM produce a scaling and 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. Examples and applications of extracting data manifolds using SOM-based techniques are presented.

Key words: Self-organising maps, principal curve and surface, data visualisation, topographic mapping

3.1 Introduction

For many years, artificial neural networks have been studied and used to construct information processing systems based on or inspired by natural biological neural structures. They not only provide solutions with improved performance when compared with traditional problem-solving methods, but also give a deeper understanding of human cognitive abilities. Among the various existing neural network architectures and learning algorithms, Kohonen's self-organising map (SOM) [35] is one of most popular neural network models. Developed for an associative memory model, it is an unsupervised learning algorithm with simple structures and computational forms, and is motivated by the retina-cortex mapping. Self-organisation in general is a fundamental pattern recognition process, in which intrinsic inter- and intra-pattern relationships within the data set are learnt without the presence of a potentially biased or subjective external influence. The SOM can provide topologically

preserved mapping from input to output spaces. Although the computational form and structure of the SOM are very simple, numerous researchers have already examined the algorithm and many of its properties, there are still many aspects to be exploited.

In this chapter, we review the background, theories and statistical properties and present recent advances of the SOM. The SOM is an optimal for vector quantisation. Its topographical ordering provides the mapping with enhanced fault and noise tolerant abilities. It also extracts a latent structure of the input space, which is applicable to many applications such as dimensionality reduction, data visualisation, clustering and classification. Various extensions of the SOM have been devised since to extend the mapping as optimal solutions for a wide range of applications. In particular, the SOM has been linked with the principal curve and surface [20] and the recently proposed visualisation induced SOM (ViSOM) [78] has been shown to represent a discrete principal curve/surface [79]. Such an analogy is explored and demonstrated and the advantages and shortly comings examined in the context of other methods such as kernel PCA [66], local linear embedding (LLE) [63] and Isomap [69]. Several examples are presented to highlight the potential of this biologically inspired model in nonlinear, principled data analysis.

3.2 Biological Background

Kohonen's self-organising map (SOM) is an abstract mathematical model of topographic mapping from the (visual) sensory to the cerebral cortex. Modelling and analysing the mapping are important to understanding how the brain perceives, encodes, recognises, and processes the patterns it receives and thus, if somewhat indirectly, is beneficial to machine-based pattern recognition. This section looks into the relevant biological models, from two fundamental phenomena involved, lateral inhibition and Hebbian learning, to Willshaw and von der Malsburg's self-organisation retinotopic model, and then to subsequent Kohonen's simplified and abstracted SOM model. Basic operations and the algorithm of the SOM as well as methods for choosing model parameters are also given.

3.2.1 Lateral Inhibition and Hebbian Learning

Human visual perception and brain make up the most complex cognition system and the most complex of all biological organs. Visual information is processed in both retina and brain, but it is widely believed and verified that most processing is done in the retina, such as extracting lines, angles, curves, contrasts, colours, and motions. The retina then encodes the information and sends through optic nerves and optic chiasma, where some left and right nerves are crossed, to the brain cortex at left or right hemispheres. The retina is a complex neural network. Human retina has over 100 million photosensitive

cells (combining rods and cones) processing in parallel the raw images and codes and renders to just over one million optic nerves to be transmitted to the brain cortex.

The *Perceptron* models some cells in the retina, especially the bipolar and ganglion cells. These cells take inputs from the outputs of cells in the previous layer. To put many units together and connect them into layers, one may hope the resulting network, the *multi-layer perceptron*, will have some functionality similar to the retina (despite some horizontally interconnections are ignored). And indeed such a structure has been demonstrated of capable of certain cognitive and information processing tasks.

Cells in neural networks (either in retina or brain) also connect and interact horizontally. The experiment on limulus by Haldan K. Hartline (1967 Nobel Prize Laureate) and his colleagues in 1960s, has confirmed such a processing on limulus retina. They revealed the so-called *lateral inhibition* activities among the retina cells. That is, there exist both short-range excitatory interaction between close cells and long-range inhibitory interaction between long range cells. This consequently explains the so-called "Mach band" phenomenon on the edges or sharp changes of light intensity. Lateral inhibition tells us that neurons in retina do not just feed the information to upper levels, but also perform an important visual processing task: edge detection and enhancement.

Neural networks present completely different approaches to computing and machine intelligence from traditional symbolic AI. The goal is to emulate the way that natural systems, especially brains, perform on various cognitive tasks. When a network of simple processing units interconnect to each other, there are potentially a massive number of synaptic weights available to be configured and modified such that the network will suit a particular task. This configuration and modification process is carried out by a learning procedure, i.e. *learning or training* algorithm. Traditional pattern recognition approaches usually require solving some well-defined functions or models, such as feature extraction, transformation, and discriminant analysis by a series of processing steps. Neural networks can simply learn from examples. Presented repeatedly with known examples of raw patterns and with an appropriate learning or training algorithm, they are able to extract by themselves the most intrinsic nature of the patterns and are able to perform recognition tasks. They will also have ability to carry out similar recognition tasks not only on trained examples but also on unseen patterns. Learning methods and algorithms, undoubtedly play an important role in building successful neural networks.

Although many learning methods have been proposed, there are two fundamental kinds of learning paradigms: *supervised learning* and *unsupervised learning*. The former is commonly used in most feed-forward neural networks, in which the input-output (or input-target) functions or relationships are built from a set of examples. While the latter resembles a *self-organisation* process in the cortex and seeks inter-relationships and associations among the input.

The most representing supervised learning rule is the *error-correction learning*. When presented an input-output pair, learning takes place when

the error exists between a desired response or target output and the actual output of the network. This learning rule applies an adjustment, proportional to this error, to the weights of the neuron concerned. That is, *learning from errors*. Derivation of such a rule can be often traced backed to minimising the mean-square-error function. More details can be found in [21]. A derivative of supervised learning is so-called *reinforcement learning*, which is based trail and error (and reward) [68] and has backings from psychology.

Self-organisation often involves both *competition* and *correlative learning*. When presented with a stimulus, neurons compete among themselves for the possession or ownership of this input. The winners then strengthen their weights or their relationships with this input. *Hebbian learning* is the most common rule for *unsupervised or self-organised learning*. The original Hebb's statement from his book, The Organization of Behaviour [22], was "*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.*"

Mathematically Hebbian learning rule can be directly interpreted as,

$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha x_i(t) y_j(t) \ , \tag{3.1}$$

where α is a positive learning rate, $0 < \alpha < 1$, and x and y are the input and output of the neural system respectively, or can also be regarded as the outputs of two neurons. That is, the change of the synaptic weight is proportional to the correlation between an input and its associated output. If the input and output coherent, the weight connecting them is strengthened (xy is positive), otherwise, weakened (xy is either negative or zero).

The Hebbian learning requires some modification before it can be used in practice, otherwise the weight will easily become saturated or unlimited. One solution is to add a forgetting term to prevent weights from increasing/decreasing monotonically as in the SOM (see the next subsection). Alternative is to normalise the weights. For instance, Oja [54] proposed a weight normalisation scheme on all weights. This introduces naturally a forgetting term to the Hebbian rule,

$$\begin{aligned} w_i(t+1) &= \frac{w_i(t) + \alpha x_i(t) y(t)}{\{\sum_{j=1}^{n} [w_j(t) + \alpha x_j(t) y(t)]^2\}^{1/2}} \\ &\approx w_i(t) + \alpha y(t)[x_i(t) - y(t) w_i(t)] + O(\alpha^2) \ , \end{aligned} \tag{3.2}$$

where $O(\alpha^2)$ represents second- and high-order terms in α, and can be ignored when a small learning rate is used.

The resulting Oja's learning algorithm is a so-called *principal component network*, which learns to extract the most variant directions among the data set. Other variants of Hebbian learning include many algorithms used for *Independent Component Analysis* [55, 26].

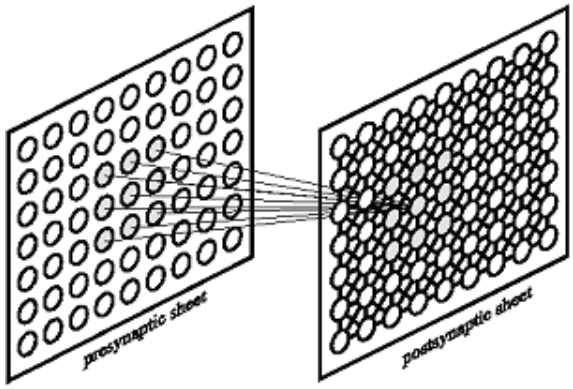


Fig. 3.1. Von der Malsburg's self-organising map model. Local clusters in a presynaptic sheet are connected to local clusters in a postsynaptic sheet. There are lateral interconnections within the postsynaptic sheet (solid lines are used to indicate such connections)

3.2.2 From Von Marsburg and Willshaw's Model to Kohonen's SOM

Stimuli from the outside world are received by various sensory or receptive fields (e.g. visual-, auditory-, motor-, or somato-sensory), coded or abstracted by the living neural networks, and projected through axons onto the cerebral cortex, often to distinct parts of cortex. In other words, the different areas of the cortex (cortical maps) correspond to different sensory inputs. Topographically ordered maps have been widely observed in the cortex. The main structures (primary sensory areas) of the cortical maps are established before birth (cited in [76, 36]), in a predetermined topographically ordered fashion. Other more detailed areas (associative areas), however, are developed through self-organisation gradually during life and in a topographically meaningful order. Therefore studying such topographically ordered projections, which had been ignored during the early period of neural information processing development [37], is clearly important for forming dimension-reduction mapping and for the effective representation of sensory information and feature extraction.

The self-organised learning behaviour of brains has been studied for a long time by many people. Many pioneering works include [2, 7, 17, 22, 35, 52, 74, 75, 76]. von der Malsburg and Willshaw [74, 76] first developed, in mathematical form, self-organising topographical 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 (see Fig. 3.1).

The model uses short-range excitatory connections between cells so that activity in neighbouring cells becomes mutually reinforced, and uses long-range inhibitory interconnections to prevent activity from spreading too far.

The postsynaptic activities $\{y_j(t),\ j{=}1,\ 2,{...}N_y\}$, at time t, are expressed by

$$\frac{\partial y_i(t)}{\partial t} + cy_i(t) = \sum_j w_{ij}(t)x_i(t) + \sum_k e_{ik}y_k^*(t) - \sum_{k'} b_{ik'}y_{k'}^*(t)\ , \tag{3.3}$$

where c is the membrane constant, $w_{ij}(t)$ is the synaptic strength between cell $i$and cell $j$in pre- and post-synaptic sheets respectively; $\{x_i(t),\ i{=}1,\ 2,{...}N_x\}$, the state of the presynaptic cells, equal to 1 if cell $i$is active or 0 otherwise; e_{kj} and b_{kj} are short-range excitation and long-range inhibition constants respectively; and $y_j^*(t)$ is an active cell in postsynaptic sheet at time t. The postsynaptic cells fire if their activity is above a threshold, say,

$$y_j^*(t) = \begin{cases} y_j(t) - \theta\ , \text{ if } y_j(t) > \theta\ ; \\ 0\ , \qquad\qquad \text{otherwise}\ . \end{cases} \tag{3.4}$$

The modifiable synaptic weights between pre- and post-synaptic sheets are then facilitated in proportion to the product of activities in the appropriate pre- and postsynaptic cells (direct realisation of Hebbian learning):

$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha x_i(t)y_j^*(t),\ \text{subject to}\ \frac{1}{N_x}\sum_i w_{ij} = \text{constant}\ , \tag{3.5}$$

where α is a small constant representing the learning rate. To prevent the synaptic strengths becoming unstable, the total strength associated with each postsynaptic cell is limited by normalisation to a constant value after each iteration.

Kohonen [35] abstracted the above self-organising learning principles and proposed a much simplified learning mechanism which cleverly incorporates the Hebb's learning rule and lateral interconnection rules and can emulate the self-organising learning effect. Although the resulting SOM algorithm was more or less proposed in a heuristic manner [40], it is a simplified and generalised model of the above self-organisation process.

In Kohonen's model, the postsynaptic activities are similar to Eq. (3.3). To find the solutions of this equation and ensure they are non-negative properties, a sigmoid type of nonlinear function is applied to each postsynaptic activity:

$$y_j(t+1) = \varphi\left(\mathbf{w}_j^T\mathbf{x}(t) + \sum_i h_{ij}y_i(t)\right)\ , \tag{3.6}$$

where h_{kj} is similar to e_{kj} and b_{kj}, the input is described as a vector as the map can be extended to any dimensional input. A typical mapping is shown in Fig. 3.2.

A spatially-bounded cluster or *bubble* will then be formed among the postsynaptic activities and will stabilise at a maximum (without loss of generality which is assumed to be unity) when within the bubble, or a minimum (i.e. zero) otherwise,

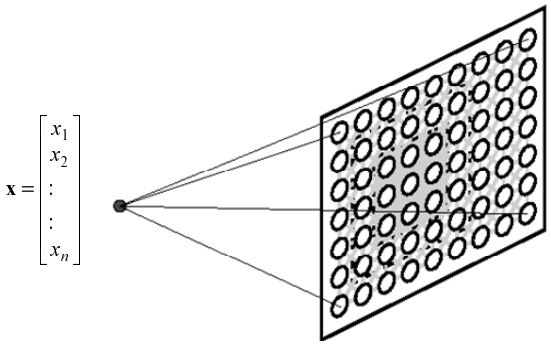


Fig. 3.2. Kohonen's self-organising map model. The input is connected to every cell in the postsynaptic sheet (the map). The learning makes the map localised, i.e. different local fields will respond to different ranges of inputs. The lateral excitation and inhibition connections are emulated by a mathematical modification, i.e. local sharing, to the learning mechanism. (So there are no actual connections between cells, or in a sense we can say the connections are virtual. Hence grey lines are used to indicate these virtual connections)

$$y_j(t+1) = \begin{cases} 1\,, & \text{if neuron } j \text{ is inside the bubble} \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{3.7}$$

The bubble is centred on a postsynaptic cell whose synaptic connection with the presynaptic cells is mostly matched with the input or presynaptic state, i.e. the first term in the function in Eq. (3.6) is the highest. The range or size, denoted by $\eta(t)$, of the bubble depends on the ratio of the lateral excitation and inhibition. To modify the Hebbian learning rule, i.e. Eq. (3.5), instead of using normalisation, a forgetting term, $\beta y_j(t) w_{ij}(t)$, is added. Let $\alpha = \beta$, and apply the function (3.7), the synaptic learning rule can then be formulated as

$$\begin{aligned} \frac{\partial w_{ij}(t)}{\partial t} &= \alpha y_j(t) x_i(t) - \beta y_j(t) w_{ij}(t) = \alpha [x_i(t) - w_{ij}(t)] y_j(t) \\ &= \begin{cases} \alpha [x_i(t) - w_{ij}(t)], & \text{if } j \in \eta(t)\,; \\ 0, & \text{if } j \notin \eta(t)\,. \end{cases} \end{aligned} \tag{3.8}$$

At each time step the best matching postsynaptic cell is chosen according to the first term of the function in Eq. (3.6), which is the inner product, or correlation, of the presynaptic input and synaptic weight vectors. When normalisation is applied to the postsynaptic vectors, as it usually is, this matching criterion is similar to the Euclidean distance measure between the weight and input vectors. Therefore the model provides a very simple computational form. The lateral interconnection between neighbouring neurons

and the "*Mexican-hat*" excitatory or inhibitory rules are simulated (mathematically) by a simple local neighbourhood excitation centred on the winner. Thus the neuron's lateral interconnections (both excitatory and inhibitory) have been replaced by neighbourhood function adjustment. The neighbourhood function's width can simulate the control of the exciting and inhibiting scalars. The constrained (with a decaying or forgetting term) Hebbian learning rule has been simplified and becomes a competitive learning model. Most of Kohonen's work has been in associative memories [32, 33, 34, 35, 36, 37]. In his studies, he has found that the spatially ordered representation of sensory information in the brain is highly related to the memory mechanism, and that the inter-representation and information storage can be implemented simultaneously by an adaptive, massively parallel, and self-organising network [37]. This simulated cortex map, on the one hand can perform a self-organised search for important features among the inputs, and on the other hand can arrange these features in a topographically meaningful order.

3.2.3 The SOM Algorithm

The SOM uses a set of neurons, often arranged in a 2D rectangular or hexagonal grid, to form a discrete topological mapping of an input space, $\boldsymbol{X} \in \Re^n$. At the start of the learning, all the weights $\{\mathbf{w}_{\mathrm{r}1}, \mathbf{w}_{\mathrm{r}2}, \ldots, \mathbf{w}_{\mathrm{r}m}\}$ are initialised to small random numbers. $\mathbf{w}_{\mathrm{r}i}$ 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. $\mathbf{r}i$ is the location vector of neuron i on the grid. Then the algorithm repeats the following steps.

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

$$v(t) = \arg\min_{k \in \Omega} \|\mathbf{x}(t) - \mathbf{w}_k(t)\| \ . \tag{3.9}$$

- Updating the weights of winner and its neighbours,

$$\Delta \mathbf{w}_k(t) = \alpha(t)\eta(v, k, t)[\mathbf{x}(t) - \mathbf{w}_v(t)] \ . \tag{3.10}$$

- Repeat until the map converges,

where $\eta(v, k, t)$is the neighbourhood function and Ω is the set of neuron indexes. Although one can use the original stepped or top-hat type of neighbourhood function (is one when the neuron is within the neighbourhood or zero otherwise), a Gaussian form is often used in practice, i.e. $\eta(v, k, t) = \exp[-\frac{||v-k||^2}{2\sigma(t)^2}]$, with σ representing the changing effective range of the neighbourhood.

The coefficients $\{\alpha(t),\ t \geq 0\}$, termed *adaptation gain*, or *learning rate*, are *scalar-valued*, *decrease monotonically*, and satisfy [36],

$$\text{(i) } 0 < \alpha(t) < 1; \text{ (ii) } \lim_{t\to\infty} \sum \alpha(t) \to \infty; \text{ (iii) } \lim_{t\to\infty} \sum \alpha^2(t) < \infty\,. \quad (3.11)$$

They are the same as to those used in stochastic approximation [62]. The third condition in (11) has been relaxed [60] to a less restrictive one, namely, $\lim_{t\to\infty} \alpha(t) \to 0$.

If the inner product similarity measure is adopted as the best matching rule, i.e.

$$v(t) = \arg\min_{k\in\Omega}[\mathbf{w}_k^T(t)\mathbf{x}(t)]\,, \quad (3.12)$$

then the corresponding weight updating should become [39]

$$\mathbf{w}_k(t+1) = \begin{cases} \frac{\mathbf{w}_k(t)+\alpha(t)\mathbf{x}(t)}{\|\mathbf{w}_k(t)+\alpha(t)\mathbf{x}(t)\|}\,; \\ \mathbf{w}_k(t)\,. \end{cases} \quad (3.13)$$

Such a form is often used in text/document mining applications (e.g. [16]).

3.3 Theories

3.3.1 Convergence and Cost Functions

Although the SOM algorithm has a simple computational form, a formal analysis of it and the associated learning processes and mathematical properties is not easily obtainable. Some important issues still remain unanswered. Self-organisation, or more specifically the ordering process, has been studied in some depth; however a universal conclusion has been difficult to obtain, if not impossible. This section reviews the statistical and convergence properties of the SOM and associated cost functions, the issue that still causes confusions to many even today. Various topology preservation measures will be analysed and explained.

The SOM was proposed to model the sensory to cortex mapping thus the unsupervised associated memory mechanism. Such a mechanism is also related to vector quantisation or vector quantiser (VQ) [46] in coding terms. The SOM has been shown to be an asymptotically optimal VQ [82]. More importantly, with the neighbourhood learning, the SOM is an error tolerant VQ and Bayesian VQ [48, 49, 50].

Convergence and ordering has only formally been proved in trivial one dimensional case. A full proof of both convergence and ordering in multidimensional are still outstanding, though there have been several attempts (e.g. [13, 14, 45, 47, 60, 82]. Especially Erwin, Obermayer and Schulten [13, 14] showed that there was no cost function that the SOM would follow *exactly*. Such an issue is also linked to the claimed lack of an exact cost function that the algorithm is following. Recent work by various researchers has already shed light on this intriguing issue surrounding the SOM. Yin and Allinson [82] extended the Central Limit Theorem and used it to show that when

the neighbourhood is reducing to just winner as in the original SOM, the weight vectors (code references) are asymptotically Gaussian distributed and will converge in mean square sense to the means of the Voronoi cells, i.e. an optimal VQ (with the SOM's nearest distance winning rule),

$$\mathbf{w}_k \rightarrow \frac{1}{P(X_k)} \int_{V_k} \mathbf{x} p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \; , \tag{3.14}$$

where V_k is the Voronoi cell (the data region) that weight vector $\mathbf{w}_k$ is responsible, and $p(\mathbf{x})$ is the probability density function of the data. In general cases with the effect of the neighbourhood function, the weight vector is a kernel smoothed mean [79],

$$\mathbf{w}_k \rightarrow \frac{\sum_{t=1}^{T} \eta(v,k,t)\mathbf{x}(t)}{\sum_{t=1}^{T} \eta(v,k,t)} \; . \tag{3.15}$$

Yin and Allinson [82] 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 recently verified by de Bolt, Cottrell and Verleysen [10] using Monte-Carlo bootstrap cross validation. The ordering was not considered. (In practice, good initialisation can be used to guide a faster or even better convergence, due to the limited training time and samples, as well as much relaxed learning rates. For example, initialising the map to a principal linear submanifold can reduce the ordering time, if the ordering process is not a key requirement.)

Luttrell [48, 49] first related hierarchical noise tolerant coding theory to the SOM. When the transmission channel noise is considered, a two-stage optimisation has to be done not only to minimise the representation distortion (as in the VQ) but also to minimise the distortion caused by the channel noise. He revealed that the SOM can be interpreted as such a coding algorithm. The neighbourhood 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 [48, 49],

$$D_2 = \int \mathrm{d}\mathbf{x} p(\mathbf{x}) \int \mathrm{d}\mathbf{n} \pi(\mathbf{n}) \|\mathbf{x} - \mathbf{w}_k\|^2 \; , \tag{3.16}$$

where $\mathbf{n}$ is the noise variable and $\pi(\mathbf{n})$ is the noise distribution. Durbin and Mitchison [12] and Mitchison [53] have also linked the SOM and this noise tolerant VQ with minimal wiring of cortex like maps.

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

$$D_2 = \sum_i \int_{V_i} \sum_k \pi(i,k) \|\mathbf{x} - \mathbf{w}_k\|^2 p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \; , \tag{3.17}$$

where V_i is the Voronoi region of cell i. When the channel noise distribution is replaced by a neighbourhood function (analogous to intersymbol dispersion), this gives to the cost function of the SOM algorithm. The neighbourhood function can be interpreted as channel noise model. Such a cost function has been discussed in the SOM community (e.g. [38, 42, 82, 58, 23]). The cost function is therefore [23],

$$E(\mathbf{w}_1, ...\mathbf{w}_N) = \sum_i \int_{V_i} \sum_k \eta(i,k) \|\mathbf{x} - \mathbf{w}_k\|^2 p(\mathbf{x})\, \mathrm{d}\mathbf{x} \; . \tag{3.18}$$

It leads naturally to the SOM update algorithm using the sample or stochastic gradient descent method [62]. That is, for each Voronoi region, the sub cost function is,

$$E_i(\mathbf{w}_1, ...\mathbf{w}_N) = \int_{V_i} \sum_k \eta(i,k) \|\mathbf{x} - \mathbf{w}_k\|^2 p(\mathbf{x})\, \mathrm{d}\mathbf{x} \; . \tag{3.19}$$

The optimisation for all weights $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$ can be sought using the sample gradients. The sample gradient for $\mathbf{w}_j$ is,

$$\frac{\partial \widehat{E}_i(\mathbf{w}_1, ...\mathbf{w}_N)}{\partial \mathbf{w}_j} = \frac{\partial \sum\limits_k \eta(i,k) \|\mathbf{x} - \mathbf{w}_k\|^2}{\partial \mathbf{w}_j} = 2\eta(i,j)(\mathbf{x} - \mathbf{w}_j) \; , \tag{3.20}$$

which leads to the SOM updating rule, Eq. (3.10). Note, although the neighbourhood function $\eta_{i,k}$ is inexplicitly related to $\mathbf{w}_j$, it does not contribute to the weight optimisation, nor does the weight optimisation lead to its adaptation (neighbourhood adaptation is often controlled by a pre-specified scheme, unrelated to the weight adaptation); thus the neighbourhood can be omitted from taking partial differentiation. This is the point that has caused problems in interpreting the cost function of the SOM in the past.

It has however been argued that this energy function is violated at boundaries of Voronoi cells where input has exactly the same smallest distance to two neighbouring neurons. Thus this energy function holds mainly for the discrete case where the probability of such boundary input points is close to zero or the local (sample) cost function $\widehat{E}_i$ should be used in deciding the winner [23]. When spatial-invariant neighbourhood 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 just like as in the ordinary manner (e.g. using the first-come-first-served fashion in the programming). Only when the neurons lie on the borders of the map, such violation occurs as unbalanced neighbourhood of the neurons. The result is a slightly more contraction towards to the centre 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,

esp. when the map is not fully converged or when the effective range of the neighbourhood function is large. With the local distortion rule, this boundary effect is heavier as greater local error is incurred for the border neurons due to its few neighbouring 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. That is,

$$v = \arg\min_i \sum_k \eta(i,k) \|\mathbf{x} - \mathbf{w}_k\|^2 . \tag{3.21}$$

When the neighbourhood 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, the differences exist due to the unbalance of the nodes presented in the neighbourhoods. Such differences become negligible to the majority of the neurons especially when a large map is used and when the neighbourhood function shrinks to its minimum scale.

3.3.2 Topological Ordering Measures

The ordering to a large extent is still an outstanding and subtle issue, largely due to the fact that there is no clear (or agreed) definition of order [18]. This is the very reason that why a full self-organisation convergence theorem including both the statistical convergence and ordering and the exact cost function are still subject to debate, the fact that has prompted many alternatives such as [67, 19, 5]. The ordering and an ordered map are clearly defined only in 1-D trivial case. Extending to higher dimension proves to be difficult if not impossible. Bauer and Pawelzik [4] have proposed a measure termed topology product to measure the topological ordering of the map,

$$P = \frac{1}{N^2 - N} \sum_i \sum_j \log \left(\prod_{l=1}^{j} \frac{d^D(\mathbf{w}_i, \mathbf{w}_{\eta^O(l,i)})}{d^D(\mathbf{w}_i, \mathbf{w}_{\eta^D(l,i)})} \frac{d^O(i, \eta^o(l,i))}{d^O(i, \eta^D(l,i))} \right)^{\frac{1}{2k}} , \tag{3.22}$$

where d^D and d^O represent the distance measures in the input or data space and output or map space respectively; $\eta(l,i)$ represents the l-th neighbour 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 neighbourhood (upto j) on the map and in the data space. The second ratio is the index distance sequences of the neighbourhood on the map and in the data space. The topographic product measures the product of the two ratios of all possible neighbourhoods.

Villmann et al. [73] proposed a topographic function to measure the neighbourhoodness of weight vectors in data space as well as on the lattice. While the neighbourhoodness 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 neighbours.

Goodhill and Sejnowski [18] proposed the C measure, a correlation between the similarity of stimuli in the data space and the similarity of their prototypes in the map space, to quantify the topological preservation,

$$C = \sum_i \sum_j F(i,j) G[M(i), M(j)] \ , \tag{3.23}$$

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 across two spaces. Various topographic mapping objectives may be unified under the C measure such as multidimensional scaling, minimal wiring, and travel salesperson problem (TSP), and noise tolerant VQ. It has also been shown that if a mapping that preserves ordering exists then maximising 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.

One can always use the underlying cost function, Eq. (3.18), to measure the goodness of the resulting map including the topology preservation, at least one can use a temporal window to take a sample of it as suggested in [38]. The (final) neighbourhood 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 neighbourhood function can be interpreted as the G measure used in (3.25) and term $||\mathbf{x} - \mathbf{w}_k||^2$ represents the F measure. Indeed, the input $\mathbf{x}$ and weight $\mathbf{w}_j$ are mapped on the map as node index i and j and their G measure is the neighbourhood function such as exponentials. Such an analogy also sheds light on the scaling effect of the SOM. Multidimensional scaling also aims to preserve local similarities on a mapped space.

3.4 SOMs, Multidimensional Scaling and Principal Manifolds

3.4.1 Multidimensional Scaling

The SOM is often associated with VQ and clustering. However it is also associated with data visualisation, dimensionality reduction, nonlinear data projection, and manifold mapping. A brief review on various data projection methods and their relationships has been given before [80].

Multidimensional Scaling

Multidimensional scaling (MDS) is a traditional subject related to dimension reduction and data projection. MDS tries to project data points onto an often two-dimensional sheet by preserving as closely as possible the inter-point metrics [9]. 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,

$$S = \frac{\sum_{i,j} (d_{ij} - D_{ij})^2}{\sum_{i,j} D_{ij}^2} , \tag{3.24}$$

where d_{ij} represents the proximity (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,.

The MDS relies on an optimisation 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 optimisation process. The methods are also often computationally intensive. The final solution depends on the starting configuration and the parameters used in the algorithm.

Sammon mapping is a well-known example of MDS [65]. In Sammon mapping intermediate normalisation (of original space) is used to preserve good local distributions and at the same time maintain a global structure. The Sammon stress is expressed as,

$$S_{Sammon} = \frac{1}{\sum_{i<j} d_{ij}} \sum_{i<j} \frac{(d_{ij} - D_{ij})^2}{d_{ij}} . \tag{3.25}$$

A second order Newton optimisation method is used to recursively solve the optimal configuration. It converges faster than the simple gradient method, but the computational complexity is even higher. It still has 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 the explicit mapping function and cannot naturally accommodate new data points. It also requires to compute and store all the inter-point distances. 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.

In addition to being computationally costly, especially for large data sets, and not adaptive, another major drawback of MDS methods including Sammon mapping 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 [11, 44], the methods are generally not adaptive.

3.4.2 Principal Manifolds

Principal component analysis

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.,

$$\min \sum_{\mathbf{x}} \left[\mathbf{x} - \sum\nolimits_{j=1}^{m} (\mathbf{q}_j^T \mathbf{x}) \mathbf{q}_j \right]^2 , \tag{3.26}$$

where $\{\mathbf{q}_j,\ j{=}1{,}2,\ \ldots m,\ m \leq n\}$ 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 $\mathbf{x}$ on these eigenvectors. The term $\mathbf{q}_j^T \mathbf{x}$ represents the projection of $\mathbf{x}$ 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 generalised Hebbian algorithm [64]. 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 visualisation power.

Nonlinear PCA and principal manifolds

The extension to nonlinear PCA (NLPCA) 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 [51, 28]. Several methods have been proposed for nonlinear PCA such as, the five-layer feedforward associative network [41] and the kernel PCA [66]. 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 NLPCA network are determined by minimising the following objective function,

$$\min \sum_{\mathbf{x}} ||\mathbf{x} - \mathbf{f}\{s_f(\mathbf{x})\}||^2 , \tag{3.27}$$

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

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

$$Cov = \frac{1}{N}\sum_{i=1}^{N} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T \ . \tag{3.28}$$

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

The principal curves and principal surfaces [20, 43] are the principled nonlinear extension of PCA. The principal curve is defined as a smooth and self-consistency curve, which does not intersect itself. Denote $\mathbf{x}$ as a random vector in R^n with density p and finite second moment. Let $f(\cdot)$ be a smooth unit-speed curve in R^n, parameterised by the arc length ρ (from one end of the curve) over $\Lambda \in \mathrm{R}$, a closed interval.

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

$$\rho_f(\mathbf{x}) = \sup_{\rho\in\Lambda}\{\rho : \|\mathbf{x} - f(\rho)\| = \inf_{\vartheta}\|\mathbf{x} - f(\vartheta)\|\} \ . \tag{3.29}$$

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

$$f(\rho) = E[\mathbf{X}|\rho_f(\mathbf{X}) = \rho] \ . \tag{3.30}$$

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 implementations. The HS algorithm [20] proposed by Hastie and Stuezle is a nonparametric method, which directly iterates the two steps of the above definition. It is similar to the standard LGB VQ algorithm [46] combined with some smoothing techniques.

HS algorithm:

- Initialisation: *Choose the first linear principal component as the initial curve,* $f^{(0)}(\boldsymbol{x})$.
- Projection: *Project 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 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,

$$f(\rho) = \frac{\sum_{i=1}^{N} \mathbf{x}_i \kappa(\rho, \rho_i)}{\sum_{i=1}^{N} \kappa(\rho, \rho_i)} \,. \tag{3.31}$$

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 [20]. Banfield and Raftery [3] have modified the HS algorithm by taking the expectation of the residual of the projections in order to reduce the bias. Kegl et al [31] have proposed an incremental, e.g. segment by segment, and arc length constrained method for practical construction of principal curves.

Tibshirani [70] 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 [5] and probabilistic principal surfaces [6]. These methods model the data by a means of a latent space. They belong to the semi-parameterised mixture model, although types and orientations of the local distributions vary from method to method.

3.4.3 Visualisation Induced SOM (ViSOM)

For scaling and data visualisation, a direct and faithful display of data structure and distribution is highly desirable. ViSOM has been proposed to extend the SOM for directly distance preservation on the map [78], instead of using a colouring scheme such as U-matrix [72], which imprints qualitatively the interneuron distances as colours 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, $\mathbf{x}(t) - \mathbf{w}_k(t)$, of the SOM algorithm is decomposed into two elements $[\mathbf{x}(t)\text{-}\mathbf{w}_v(t)] + [\mathbf{w}_v(t) - \mathbf{w}_k(t)]$. The first term 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. The second force is a lateral contraction force bringing neighbouring neuron k to the winner v. In the ViSOM, this lateral contraction force is constrained or regulated in order

to help maintain a unified local inter-neuron distance $||\mathbf{w}_v(t)-\mathbf{w}_k(t)||$ on the map.

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t)\{[\mathbf{x}(t) - \mathbf{w}_v(t)] + \beta[\mathbf{w}_v(t) - \mathbf{w}_k(t)]\} \ , \quad (3.32)$$

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

The ViSOM regularises 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. The aim is to adjust inter-neuron distances on the map in proportion to those in the data space, i.e. $\lambda D_{vk} \propto d_{vk}$. 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 representative). This has a similar effect to Sammon mapping, which also aims at achieving this proportionality, $D_{ij} \propto d_{ij}$. The key feature of the ViSOM is that the distances between the neurons (which data are mapped to) on the map (in a neighbourhood) reflect the corresponding distances in the data space. 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 map resolution can be enhanced by interpolating a trained map or incorporating local linear projections [81]. The size or covering range of the neighbourhood function can also be decreased from an initially large value to a final smaller one. The final neighbourhood, however, should not contain just the winner. The rigidity or curvature of the map is controlled by the ultimate size of the neighbourhood. The larger of this size the flatter the final map is in the data space. Guidelines for setting these parameters have been given in [79]. An example on data visualisation will be shown in the next section.

Several authors have since introduced improvements and extensions on the ViSOM. For example, in [77], a probabilistic data assignment [19] is used in both the input assignment and the neighbourhood function and an improved second order constraint is adopted. The resulting SOM has a clearer connection to an MDS cost function. In [15] the ViSOM has been extended to arbitrary, neural gas type of map structure. Various existing variants of the SOM such as hierarchical, growing and hierarchical and growing structures are readily extendable to the ViSOM for various application needs.

The SOM has been related to the discrete principal curve/surface algorithm [61]. However the differences remain in both the projection and smoothing processes. In the SOM the data are projected onto the nodes rather than onto the curve. The principal curves perform the smoothing entirely in the data space –see Eq. (3.31). The smoothing process in the SOM and ViSOM, as a convergence criterion, is [79],

$$\mathbf{w}_k = \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(v,k,i)}{\sum_{i=1}^{L} \eta(v,k,i)} \ . \tag{3.33}$$

The smoothing is governed by the indexes of the neurons in the map space. 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 or the positions on the map grid, not the positions in the input space. So $||k-i||$ does not resemble $||\mathrm{w}_k - \mathrm{w}_i||$ in the common SOM. In the ViSOM, however, as the local 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. $||k-i|| \sim ||\mathrm{w}_k - \mathrm{w}_i||$. The smoothing process in the ViSOM resembles that of the principal curves as shown below,

$$\mathbf{w}_k = \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(v,k,i)}{\sum_{i=1}^{L} \eta(v,k,i)} \approx \frac{\sum_{i=1}^{L} \mathbf{x}_i \eta(\mathbf{w}_v,\mathbf{w}_k,i)}{\sum_{i=1}^{L} \eta(\mathbf{w}_v,\mathbf{w}_k,i)} \ . \tag{3.34}$$

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

3.5 Examples

There have been reported thousands of applications of the SOM and its variants [39, 29, 56] since its introduction, too many to list here. There are a dedicated international Workshop on SOMs (WSOM) and focused sessions in many neural networks conferences. There have also been several special issues dedicated to the advances in the SOM and related topics [1, 27, 8]. Still many new applications are being reported in many relevant journals today. SOMs will remain an active topic in their continued extension, combination and applications in the years to come.

Typical applications include image and video processing and retrieval; density or spectrum profile modelling; nonlinear ICA; classification (LVQ); cross-modal information processing and associations; data visualisations; text and document mining and management systems; gene expression data analysis and discovery;. novelty detection; robotics and computer animation. In this section, we take a slice of typical applications and present several examples on high dimensional data visualisation and scaling only.

3.5.1 Data Visualisation

Data projection and visualisation has become a major application area for neural networks, in particular for the SOMs [39], as its topology preserving property is unique among other neural models. Good projection and visualisation methods help to identify clustering tendency, to reveal the underlying functions and patterns, and to facilitate decision support. A great deal of research has been devoted to this subject and a number of methods have been proposed.

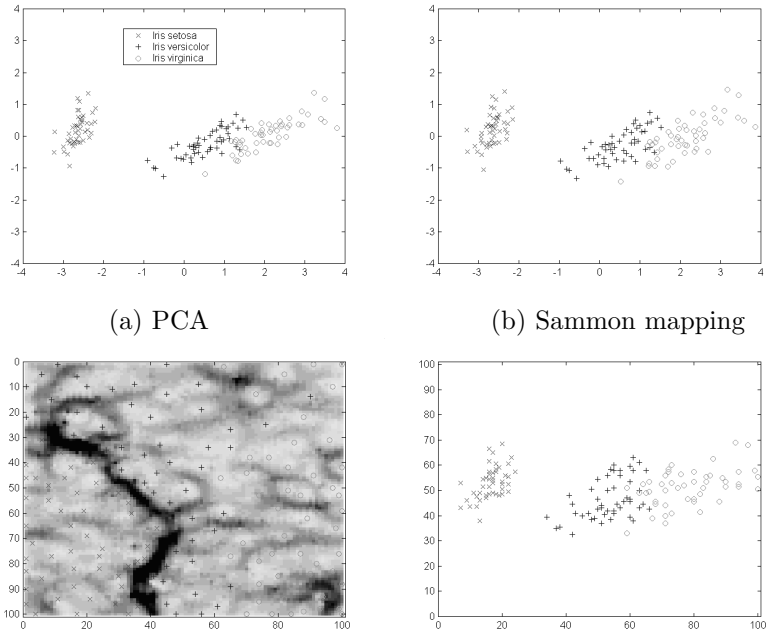
The SOM has been widely used as a visualisation tool for dimensionality reduction (e.g. [25, 30, 39, 72]). The SOM's unique topology preserving property can be used to visualise the relative mutual relationships among the data. However, the SOM does not directly apply to scaling, which aims to reproduce proximity in (Euclidean) distance on a low visualisation space, as it has to rely on a colouring scheme (e.g. the U-matrix method [72] to imprint the distances crudely on the map. Often the distributions of the data points are distorted on the map. The ViSOM [78, 79] constrains the lateral contraction force between the neurons in the SOM 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. An example is shown in Fig. 3.3, where the ViSOM of 100×100 (hexagonal) was used to map the 4-D Iris data set and it gives direct visualisation of data distribution, similar to the sammon mapping. Although, the SOM with colouring can show the gap between iris setosa and the rest, it is impossible to capture the data structure and represent the data proximity on the map.

Usually for a fine mapping, the resolution parameter needs to be set to small value and a large number of nodes, i.e. a large map, is required, as for all discrete mappings. However such computational burden can be greatly reduced by interpolating a trained map [83] or incorporating a local linear projection on a trained low resolution map [81].

A comparison with other mapping methods- PCA, Sammon and LLE - on an "S" shape manifold is also shown in Fig. 3.4.

3.5.2 Document Organisation and Content Management

With drastically increasing amount of unstructured content available electronically within an enterprise or on the web, it is becoming inefficient if not impossible to rely on human operators to manually annotate electronic documents. (Web) content management systems have become an important area of research in many applications such as e-libraries, enterprise portals, e-commerce, software contents management, document management and knowledge discovery. The documents, generated in an enterprise either centrally or locally by employees, are often unstructured or arranged in ad hoc manner



(a) PCA (b) Sammon mapping

(c) SOM with U matrix colouring (d) ViSOM

Fig. 3.3. Mapping and visualisation of Iris data set

(e.g. emails, reports, web pages, presentations). Document management addresses many issues storage, indexing, security, revision control, retrieval and organization of documents. Many existing full-text search engines return a large ranked list of documents, many of which are irrelevant. This is especially true when queries are short and very general words are used. Hence the document organization has become important in information retrieval and content management.

Documents can be treated as feature vectors. There are many methods to extract features such as word frequencies from documents (cf. [16]). The SOM has been applied to organise and visualise vast amount of textual information. Typical examples include the Welfaremap [30] and the WEBSOM [25]. Many variants of the SOM have been proposed and applied to document organization, e.g. TreeGCS [24] and growing hierarchical-SOM (GH-SOM) [57]. The main advantage of the SOM is the topology preservation of input space, which makes similar topics appear closely on the map. Most these applications however are based on 2D maps and grids, which are intuitive for digital library idea. However such a 2D grid presentation of information (mainly document files) is counter to all existing computer file organisers and explorers such as Windows Explorer. A new way of utilising the SOM as a

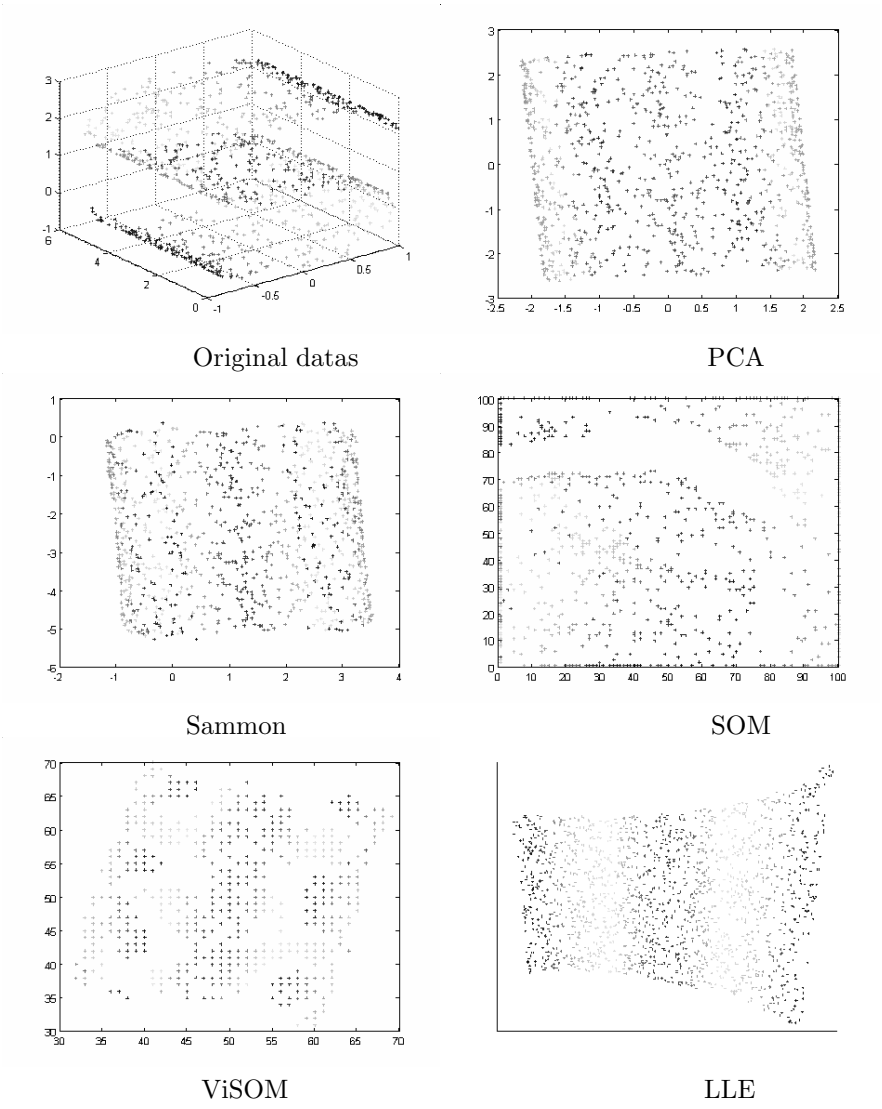


Fig. 3.4. Manifold mapping obtained by various methods

topology-preserving tree structure for content management and knowledge discovery has been proposed [16]. The method can generate a taxonomy of topics from a set of unannotated, unstructured documents. It consists of a hierarchy of self-organizing growing chains, each of which can develop independently in terms of size and topics. The dynamic development process is validated continuously using a proposed entropy-based Bayesian information

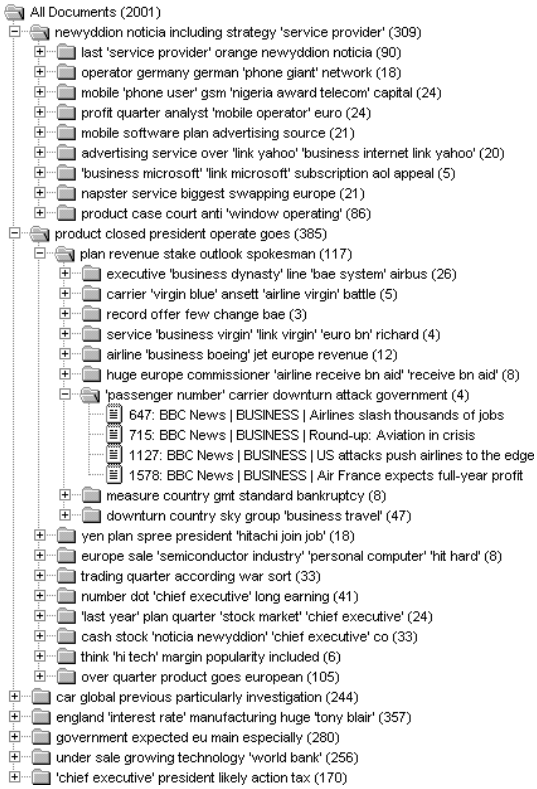


Fig. 3.5. A typical result of topological tree structure for organising documents

criterion. Each chain meeting the criterion spans child chains, with reduced vocabularies and increased specializations.

This results in a topological tree hierarchy, which can be browsed like a table of contents directory or web portal. A typical tree is shown in Fig. 3.5. This approach has been tested and compared with several existing methods on real world web page datasets. The results have clearly demonstrated the advantages and efficiency in content organization of the proposed method in terms of computational cost and representation. The preserved topology provides a unique, additional feature for retrieving related topics and confining the search space.

An application prototype developed based this method is shown in Fig. 3.5. The left panel displays the generated content tree with various levels and preserved topology on these levels. The right panel shows the details of a selected level or branch or a particular document. The method bears familiar interface to the most popular Windows explorer style.

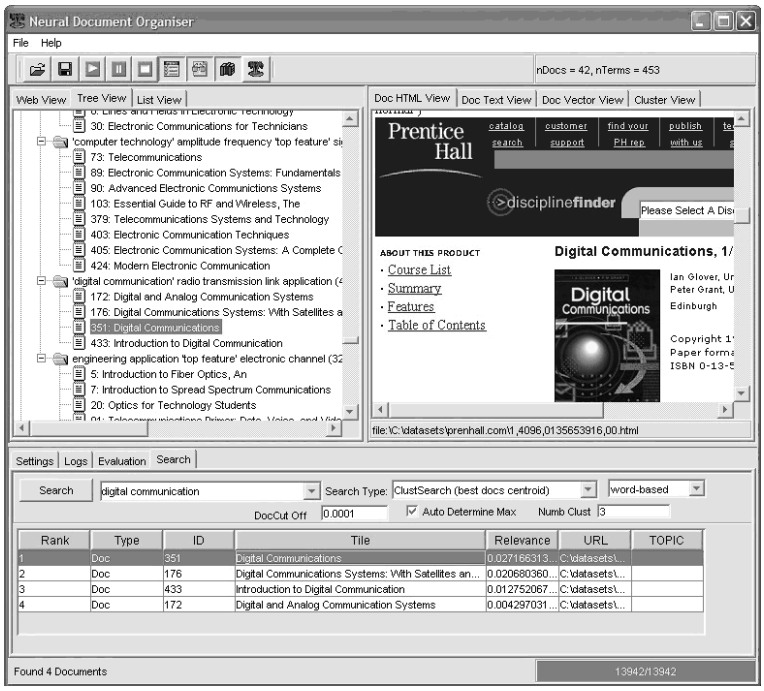


Fig. 3.6. A screen shot of a document management system using topological tree structure

References

1. Allinson, N.M., Obermayer, K. and Yin, H.: Neural Networks, Special Issue on New Developments in Self-Organising Maps, **15**, 937–1155 (2002)
2. Ameri, S.-I.: Topographic organisation of nerve fields. *Bulletin of Mathematical Biology*, **42**, 339–364 (1980)
3. Banfield, J.D. and Raftery, A.E.: Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. Journal of the American Statistical Association, **87**, 7–16 (1992)
4. Bauer, H.-U. and Pawelzik, K.R.: Quantifying the neighborhood preservation of self-organizing feature maps. IEEE Trans. Neural Networks, **3**, 570–579 (1992)
5. Bishop, C.M., Svensén, M., and Williams, C.K.I.: GTM: The generative topographic mapping. Neural Computation, **10**, 215–235 (1998)
6. Chang, K.-Y. and Ghosh, J.: A unified model for probabilistic principal surfaces. IEEE Trans. Pattern Analysis and Machine Intelligence, **23**, 22–41 (2001)
7. Cottrell, M. and Fort, J.C.: A stochastic model of retinotopy: a self-organising process. Biological Cybernetics, **53**, 405–411 (1986)
8. Cottrell, M. and Verleysen, M.: Neural Networks, Special Issue on Advances in Self-Organizing Maps, **19**, 721–976 (2006)
9. Cox, T.F. and Cox, M.A.A.: Multidimensional Scaling, Chapman & Hall (1994)
10. de Bolt, E., Cottrell, M., and Verleysen, M.: Statistical tools to assess the reliability of self-organising maps. Neural Networks, **15**, 967–978 (2002)

11. De Ridder, D. and Duin, R.P.W.: Sammon mapping using neural networks: a comparison. Pattern Recognition Letters, **18**, 1307–1316 (1997)
12. Durbin, R. and Mitchison, G.: A dimension reduction framework for understanding cortical maps. Nature, **343**, 644–647 (1990)
13. Erwin, E., Obermayer, K. and Schulten, K.: Self-organising maps: ordering, convergence properties and energy functions. Biological Cybernetics, **67**, 47–55 (1992)
14. Erwin, E., Obermayer, K. and Schulten, K.: Self-organising maps: stationary states, metastability and convergence rate. Biological Cybernetics, **67**, 35–45 (1992)
15. Estévez, P.A and Figueroa, C.J.: Online data visualization using the neural gas network. Neural Networks, **19**, 923–934 (2006)
16. Freeman, R. and Yin, H., Adaptive topological tree structure (ATTS) for document organisation and visualisation. Neural Networks, **17**, 1255–1271 (2004)
17. Gaze, R.M.: The Information of Nerve Connections, Academic Press (1970)
18. Goodhill, G.J. and Sejnowski, T.: A unifying objective function for topographic mappings. Neural Computation, **9** 1291–1303 (1997)
19. Graepel, T. Burger, M. and Obermayer, K.: Phase transitions in stochastic self-organizing maps. Phys. Rev. E, **56** 3876–3890 (1997)
20. Hastie, T. and Stuetzle, W.: Principal curves. Journal of the American Statistical Association, **84**, 502–516 (1989)
21. Haykin, S.: Neural Networks: A Comprehensive Foundation, second edition, Prentice Hall, (1998)
22. Hebb, D.: Organisation of Behaviour, John Wiley & Sons (1949)
23. Heskes, T.: Energy functions for self-organizing maps. Kohonen Maps, E. Oja and S. Kaski (eds.), Elsevier, Amsterdam, 303–315 (1999)
24. Hodge, V.J. and Austin, J.: Hierarchical growing cell structures: TreeGCS. IEEE Trans. Knowledge and Data Engineering, **13**, 207–218 (2001)
25. Honkela, T., Kaski, S., Lagus, K., and Kohonen, T.: WEBSOM-self-organizing maps of document collections. Proc. Workshop on Self-Organizing Maps (WSOM'97), 310–315 (1997)
26. Hyvärinen, A, Karhunen, J. and Oja, E.: Independent Component Analysis. John Wiley & Sons, Inc. (2001)
27. Ishikawa, M., Miikkulainen R. and Ritter, H.: Neural Network, Special Issue on New Developments in Self-Organizing Systems, **17**, 1037–1389 (2004)
28. Karhunen, J., and Joutsensalo, J.: Generalisation of principal component analysis, optimisation problems, and neural networks. Neural Networks, **8**, 549–562 (1995)
29. Kaski, S., Kangas, J. and Kohonen, T.: Bibliography of self–organizing map (SOM) papers: 1981–1997. Neural Computing Surveys, **1**, 1–176 (1998)
30. Kaski, S., and Kohonen, T.: Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In: Refenes, A.-P.N., Abu-Mostafa, Y., Moody, J., and Weigend, A. (eds.) Neural Networks in Financial Engineering, World Scientific, 498–507 (1996)
31. Kegl, B., Krzyzak, A., Linder, T., and Zeger, K.: A polygonal line algorithm for constructing principal curves. Neural Information Processing Systems (NIPS'98), **11**, 501–507 (1998)
32. Kohonen, T.: Correlation matrix memory. IEEE Trans. Computers, **21**, 353–359 (1972)

33. Kohonen, T.: A new model for randomly organised associative memory. Int. Journal of Neuroscience, **5**, 27–29 (1973)
34. Kohonen, T.: An adaptive associative memory principle. IEEE Trans. Computers, **23**, 444–445 (1974)
35. Kohonen, T.: Self-organised formation of topologically correct feature map. Biological Cybernetics, **43**, 56–69 (1982)
36. Kohonen, T.: Self-organization and associative memory, Springer (1984)
37. Kohonen, T.: Representation of sensory information in self-organising feature maps, and relation of these maps to distributed memory networks. Proc. SPIE, **634**, 248–259 (1986)
38. Kohonen, T.: Self-organizing maps: optimization approaches. In: Artificial Neural Networks, vol. 2. North-Holland: Amsterdam, 981–990 (1991)
39. Kohonen, T.: Self-Organising Maps, second edition, Springer (1997)
40. Kohonen, T.: Comparison of SOM point densities based on different criteria. Neural Computation, **11**, 2081–2095 (1999)
41. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. AICHE Journal, **37**, 233–243 (1991)
42. Lampinen, J. and Oja, E.: Clustering properties of hierarchical self-organizing maps. Journal of Mathematical Imaging and Vision, **2**, 261–272 (1992)
43. LeBlanc, M., and Tibshirani, R. J.: Adaptive principal surfaces. J. Amer. Statist. Assoc. **89**, 53–64 (1994)
44. Lee, R.C.T., Slagle, J.R., and Blum, H.: A triangulation method for the sequential mapping of points from n-space to two-space. IEEE Trans. Computers, **27**, 288–292 (1977)
45. Lin, S. and Si, J.: Weight-value convergence of the SOM algorithm for discrete input. Neural Computation, **10**, 807–814 (1998)
46. Linde, Y., Buzo, A. and Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. Communications, **28**, 84–95 (1980)
47. Lo, Z.P. and Bavarian, B.: On the rate of convergence in topology preserving neural networks. Biological Cybernetics, **65**, 55–63 (1991)
48. Luttrell, S.P.: Derivation of a class of training algorithms. IEEE Trans. Neural Networks, **1**, 229–232 (1990)
49. Luttrell, S.P.: Code vector density in topographic mappings: Scalar case, IEEE Trans. Neural Networks, **2**, 427–436 (1991)
50. Luttrell, S.P. A Bayesian analysis of self-organising maps. Neural Computation, **6**, 767–794 (1994)
51. Malthouse, E.C.: Limitations of nonlinear PCA as performed with generic neural networks. IEEE Trans. Neural Networks, **9**, 165–173 (1998)
52. Marr, D.: A theory of cerebellar cortex. Journal of Physiology, **202**, 437–70 (1969)
53. Mitchison, G.: A type of duality between self-organising maps and minimal wiring, Neural Computation, **7**, 25–35 (1995)
54. Oja, E.: Neural networks, principal components, and subspaces. Int. Journal of Neural Systems, **1**, 61–68 (1989)
55. Oja, E.: PCA, ICA, and nonlinear Hebbian learning. Proc. Int. Conf. on Artificial Neural Networks (ICANN'95), 89–94 (1995)
56. Oja, M., Kaski, S. and Kohonen, T.: Bibliography of self-organizing map (SOM) papers: 1998–2001 addendum. Neural Computing Surveys, **3**: 1–156 (2003)

57. Rauber, A., Merkl, D., and Dittenbach, M.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Trans. Neural Networks, **13** 1331 –1341 (2002)
58. Ripley, B.D.: Pattern Recognition and Neural Networks, Cambridge University Press (1996)
59. Ritter, H.: Asymptotical level density for class of vector quantisation processes. IEEE Trans. Neural Networks, **2**, 173–175 (1991)
60. Ritter, H. and Schulten, K.: Convergence properties of Kohonen's topology conserving maps: fluctuations, stability, and dimension selection. Biological Cybernetics, **60**, 59–71 (1988)
61. Ritter, H., Martinetz, T., and Schulten, K.: Neural Computation and Self-organising Maps: An Introduction. Addison-Wesley Publishing Company (1992)
62. Robbins, H. and Monro, S.: A stochastic approximation method. Annals of Math. Statist., **22**, 400–407 (1952)
63. Roweis, S. T., and Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding. Science, **290**, 2323–2326 (2000)
64. Sanger, T. D.: Optimal unsupervised learning in a single-layer linear feedforward network. Neural Networks, **2**, 459–473 (1991)
65. Sammon, J. W.: A nonlinear mapping for data structure analysis. IEEE Trans. Computer, **18**, 401–409 (1969)
66. Schölkopf, B., Smola, A., and Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, **10**, 1299–1319 (1998)
67. Sum, J., Leung, C.-S. , Chan, L.-W., and Xu, L.: Yet another algorithm which can generate topography map. IEEE Trans. Neural Networks, **8**, 1204–1207 (1997)
68. Sutton, R.S., Barto, A.G., and Williams, R.J. (1991). Reinforcement learning is direct adaptive optimal control. Proc. American Control Conference, 2143–2146,.
69. Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. Science, 290: 2319–2323.
70. Tibshirani, R.: Principal curves revisited. Statistics and Computation, **2**, 183–190 (1992)
71. Törönen, P., Kolehmainen, K., Wong, G., and Castrén, E.: Analysis of gene expression data using self-organising maps. FEBS Letters, **451**, 142–146 (1999)
72. Ultsch, A.: Self-organising neural networks for visualisation and classification. In: Opitz, O. Lausen, B., and Klar, R. (eds.) Information and Classification, 864–867 (1993)
73. Villmann, T., Der, R., Herrmann, M., and Martinetz, T.M.: Topology preservation in self-organizing feature maps: exactdefinition and measurement. IEEE Trans. Neural Networks, **8**, 256–266 (1997)
74. von der Malsburg, C. and Willshaw, D.J.: Self-organization of orientation sensitive cells in the striate cortex. Kybernetik, **4**, 85–100 (1973)
75. Willshaw, D.J., Buneman, O.P., and Longnet-Higgins, H.C.: Non-holographic associative memory. Nature, **222**, 960–962 (1969)
76. Willshaw, D.J. and von der Malsburg, C.: How patterned neural connections can be set up by self-organization, Proc. Royal Society of London. Series B, **194**, 431–445 (1976)
77. Wu, S. and Chow, T.W.S.: PRSOM: A new visualization method by hybridizing multidimensional scaling and self-organizing map. IEEE Trans. Neural Networks, **16**, 1362–1380 (2005)

78. Yin, H.: ViSOM-A novel method for multivariate data projection and structure visualisation. IEEE Trans. Neural Networks, **13**, 237–243 (2002)
79. Yin, H.: Data visualisation and manifold mapping using the ViSOM. Neural Networks, **15**, 1005–1016 (2002)
80. Yin, H.: Nonlinear multidimensional data projection and visualisation. Proc. IDEAL'03, 377–388 (2003)
81. Yin, H.: Resolution enhancement for the ViSOM. Proc. Workshop on Self-Organizing Maps, 208–212 (2003)
82. Yin, H. and Allinson, N.M.: On the distribution and convergence of the feature space in self-organising maps. Neural Computation, **7**, 1178–1187 (1995)
83. Yin, H. and Allinson, N. M.: Interpolating self-organising map (iSOM). Electronics Letters, **35**, 1649–1650 (1999)