

An Adaptive Multidimensional Scaling and Principled Nonlinear Manifold

Hujun Yin

University of Manchester

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

Email: h.yin@manchester.ac.uk

Keywords: *self-organizing maps, multidimensional scaling, dimension reduction, data visualization*

Abstract— The self-organizing map (SOM) and its variant, the visualization induced SOM (ViSOM), have been linked with principal manifolds. They have also been shown to yield similar results to multidimensional scaling (MDS). However the exact connection has not yet been established. In this paper we first examine their relationship with (generalized) MDS from their cost functions in the aspect of data visualization and dimensionality reduction. The SOM is shown to produce a quantized, qualitative or nonmetric scaling, while the ViSOM is a quantitative metric scaling. Then we propose a way to use the core principle of the ViSOM, i.e. local distance preservation, to adaptively construct a metric local scaling and extract a nonlinear manifold. Comparison with other methods such as ISOMAP and LLE has been made, especially in mapping highly nonlinear subspaces. The advantages over other methods are also discussed.

1 Introduction

A great challenge in an information era is to analyze a vast amount of data in order to extract useful information and to discover meaningful patterns and rules. Clustering, classification and projection of multidimensional data are widely used practices for this purpose in many fields ranging from high-throughput bioinformatics and web information extraction to pattern recognition, decision support, and data and knowledge management. Seeking a suitable and meaningful representation of the data space has always been a key objective of data analysis and pattern recognition. Projecting and abstracting data onto their underlying subspace can reduce the number of features and identify latent variables, detect intrinsic structures, facilitate visualization and analysis of the interactions between variables. With the ever fast increasing quantity and complexity of the data (and pattern features) used for pattern recognition tasks, more sophisticated methods are required and being developed. A great deal of research has been devoted to this emerging topic, mainly on improving and extending the classic methods such as the principal component analysis (PCA) and multi-dimensional scaling (MDS).

PCA has long been considered to be the work horse and widely used in reducing variables and visualizing data in scatter plots or linear subspaces. Singular value decomposition and factor analysis are adopted to perform the task due to various advantages such as direct operation on data matrix, stable results even when data matrices are ill-conditioned, and decomposition at both feature and data level. The linearity of PCA however limits its power for practical, complex and increasingly large data sets, as it cannot capture nonlinear relationships defined by beyond second order statistics. Projection onto a linear plane will only provide limited feature reduction or visualization power. Extension to nonlinear projection, in principle, can tackle the problems better; yet a unique solution is still to be defined [19]. Various methods have been proposed, such as, the auto-associative networks [15], generalized PCA [11], kernel PCA [26], principal curve and surface [9], and local linear embedding (LLE) [24].

MDS is another popular statistical method that tries to project data points onto a lower (often two) dimensional plane by preserving as close as possible the inter-point metrics [2]. Metric MDS generalizes the classic MDS by minimizing a stress function. The mapping is generally nonlinear and can reveal the overall structure of the data. Sammon mapping [25] is a widely known example and uses an inter-point distance in the data space as the weighting for the stress. In contrast to metric MDS, non-metric MDS finds a monotonic relationship (instead of metric ones) between the dissimilarities of the data points in the data space and those of their corresponding coordinates in the low-dimensional space. More general weighting scheme has been proposed recently and the resulting MDS is called generalized MDS [1]. Isomap [27] applies scaling on geodesic instead of Euclidean distances. MDS methods are point-to-point mappings and do not provide generalizing mapping functions or manifolds.

Neural networks present alternative approaches to nonlinear data projection and dimension reduction. They can provide (implicit) generalizing mapping functions. Early examples including feed-forward neural network based mapping [20] and radial basis function based MDS [17]. The self-organizing map (SOM) [12, 14] has become a widely used method for data visualization and



dimensionality reduction. The topology preserving property of the SOM is utilized to extract and visualize relative mutual relationships among the data. Many variants and extensions have since been proposed, including the recent visualization *induced* SOM (ViSOM) [31, 32]. The ViSOM regularizes the inter-neuron distances within a neighborhood so to preserve (local) distances on the map.

2 SOM, ViSOM and MDS

In this section, the relationships among SOM, ViSOM, principal manifold and MDS are revealed and discussed.

2.1 SOM

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, $\mathbf{X} \in \mathcal{R}^n$. At the start of the learning, all the weights $\{\mathbf{w}_{r1}, \mathbf{w}_{r2}, \dots, \mathbf{w}_{rm}\}$ are initialized to small random numbers. \mathbf{w}_{ri} 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 (coordinates) 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,

$$v(t) = \arg \min_{k \in \Omega} \|\mathbf{x}(t) - \mathbf{w}_k(t)\| \quad (1)$$
- Updating the weights of winner and its neighbors,

$$\Delta \mathbf{w}_k(t) = \alpha(t) \eta(v, k, t) [\mathbf{x}(t) - \mathbf{w}_k(t)] \quad (2)$$
- Repeat until the map converges.

where $\eta(v, k, t)$ is the neighborhood function and Ω is the set of neuron indexes. Although one can use the original top-hat type of neighborhood function, a Gaussian form,

$$\eta(v, k, t) = \exp\left[-\frac{\|v - k\|^2}{2\sigma(t)^2}\right],$$

is often used in practice with σ representing the effective range of the neighborhood.

The cost function of the SOM has been discussed in the SOM community, e.g. [5, 10, 13, 16, 18, 21, 30]. At least in discrete case, or in approximation, the cost function can be written as,

$$E(\mathbf{w}_1, \dots, \mathbf{w}_m) = \sum_i \int_{V_i} \sum_k \eta(i, k) \|\mathbf{x} - \mathbf{w}_k\|^2 p(\mathbf{x}) d\mathbf{x} \quad (5)$$

which naturally leads to the SOM update algorithm using the sample or stochastic gradient descent method.

The SOM has been widely used for data visualization. However the interneuron distances, as referred to the data space, have to be marked crudely by colors or gray levels. The coordinates of the neurons (the resulting of scaling) are fixed on the lower dimensional (often 2D) grid and do not resemble the distances (dissimilarities) in the data space.

2.2 ViSOM

For metric scaling and data visualization, a direct and faithful display of data structure and distribution is desirable. 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 embedded into the data space, onto which the data points are mapped and the inter-point distances are locally 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: $[\mathbf{x}(t) - \mathbf{w}_v(t)]$ and $[\mathbf{w}_v(t) - \mathbf{w}_k(t)]$. The first term represents the updating force from the weight of 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 neighboring neurons to the winner. In the ViSOM [31], this lateral contraction force is regulated in order to help maintain unified inter-neuron distances $\|\mathbf{w}_v(t) - \mathbf{w}_k(t)\|$ at least locally on the map.

$$\Delta \mathbf{w}_k(t) = \alpha(t) \eta(v, k, t) [\mathbf{x}(t) - \mathbf{w}_k(t)] + \beta [\mathbf{w}_v(t) - \mathbf{w}_k(t)] \quad (13)$$

where the simplest constraint can be $\beta := d_{vk}/(D_{vk}\lambda) - 1$, d_{vk} is the distance of neuron weights in the input space, D_{vk} the distance of neuron indexes on the map, and λ a resolution parameter (scale), i.e. the desired inter-neuron distance on the map referred to the data space.

The ViSOM regularizes the contraction force so that local distances between the nodes on the map are analogous to the distances of their weights in the data space. In addition to SOM' objective for minimizing the quantization error, the aim is also to ensure the interneuron distances on the map in proportion to those in the data space, i.e. $D_{vk} \propto d_{vk}$, or $\lambda D_{vk} \approx d_{vk}$, at least locally. When the data points are eventually projected on a trained map, the distance between two local data points in the original space is proportional to that on the map, subject to the quantization error and map resolution. The key feature of the ViSOM is that the distances between neurons (which data are mapped to) on the map (in a neighborhood) reflect the corresponding distances in the data space. This makes visualization more direct and quantitative and resembles metric MDS. The map resolution can be enhanced by interpolating a trained map or incorporating local linear projections to reduce the computational cost of the training [33]. The neighborhood size determines the locality of such distance-preserving effect and the rigidity or curvature of the map.

Several improvements have since been made on the original ViSOM. For example, interpolation and a local linear projection (LLP) are proposed to enhance the resolution of the ViSOM or to make the mapping continuous in [33]. In [29], a probabilistic data assignment is used in both the input assignment and the neighborhood function and a second-order constraint is adopted. In [6] the ViSOM principle has been extended to arbitrary, neural gas type of map structure.



2.3 Connection with principal curves

The SOM has been related (e.g. [23]) to the discrete principal curve/surface, a smooth, self-consistent curve/surface that does not intersect itself [9]. However the differences remain in both the projection and smoothing processes. In the SOM data are projected onto the nodes rather than onto the curve. The principal curve/surface performs the smoothing in the data space. The smoothing process in the SOM and ViSOM, as a convergence criterion, is [31],

$$\mathbf{w}_k = \frac{\sum_{i=1}^L \mathbf{x}_i \eta(v, k, i)}{\sum_{i=1}^L \eta(v, k, i)} \quad (15)$$

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 neighborhood function uses 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 $\|\mathbf{w}_k - \mathbf{w}_i\|$ in the SOM. In the ViSOM, however, as the inter-neuron distances on the map represent those in the data space (subject to the resolution parameter), the distances of node indexes on the map are in proportion to the difference of their positions in the data space, i.e. $\|k - i\| \sim \|\mathbf{w}_k - \mathbf{w}_i\|$. Furthermore the LLP method [33] can be used to make this approximation even more precise. Thus the smoothing in the ViSOM resembles that of the principal curve/surface as,

$$\mathbf{w}_k = \frac{\sum_{i=1}^L \mathbf{x}_i \eta(v, k, i)}{\sum_{i=1}^L \eta(v, k, i)} \approx \frac{\sum_{i=1}^L \mathbf{x}_i \eta(\mathbf{w}_v, \mathbf{w}_k, i)}{\sum_{i=1}^L \eta(\mathbf{w}_v, \mathbf{w}_k, i)} \quad (16)$$

It shows that ViSOM is a closer approximation to the principal curves/surfaces than the SOM is. The SOM and ViSOM may be similar only when the data is uniformly distributed, or the number of nodes becomes very large.

2.4 Connection with MDS

The similarities between SOMs and MDS in terms of topographic mapping – mostly the qualitative likeness of the mapping results have been reported and discussed [e.g. 22, 32, 28]. However clear limitations of using the SOM for MDS have been noted [7]. Many applications combine the SOM and MDS for improved visualization of the SOM mapping results. We show that the metric-preserving ViSOM approximates a discrete principal manifold and also produces a similar mapping result as to a metric MDS

Let's take a close look at the cost function (stress) of metric MDS and rewrite it as,

$$\sum_{i,j} (d_{ij} - D_{ij})^2 = \sum_{i,j} (d_{ij}^2 + D_{ij}^2 - 2d_{ij}D_{ij}) \quad (18)$$

(Further weighting is introduced in curvilinear component

analysis [4] and generalized multidimensional scaling [1].)

The first term is a constant as data points are fixed and so is the second term eventually. To minimize the above stress is to maximize the third term. The third term plays a dominant role and explains that the mapping is to form corresponding correlation between inter-distances in the original and mapped spaces. This is closely related to the C measure for topographic mapping [8].

For SOMs, the sample cost is,

$$\sum_k \eta(i, k) \|\mathbf{x} - \mathbf{w}_k\|^2 \quad (19)$$

Note \mathbf{x} is the data set confined to node i . As \mathbf{w}_k is the mean of voronoi region k , let's denote it as $\bar{\mathbf{x}}_k$. Let's also denote $\bar{\mathbf{x}}_i$ as the mean of voronoi region i . Furthermore $\eta(i, k)$ is a function of $\|i - k\|$. Then the above equation can be approximated as,

$$\begin{aligned} \sum_k \eta(i, k) \|\mathbf{x} - \mathbf{w}_k\|^2 &\approx \sum_k f(\|i - j\|) \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_k\|^2 \\ &= \sum_k f(D_{ik}) d_{ik}^2 \end{aligned} \quad (20)$$

For SOMs, $f(\|i - k\|)$ is simply the neighborhood function, typically an exponential function. The first term of its Taylor expansion is proportional to $\|i - k\|^2$ (subject to quantization error). This leads the above cost function approximately to,

$$\sum_k -(D_{ik} d_{ik})^2 \quad (21)$$

where D_{ik} represents the distance between indexes of the neurons i and k on the map. Therefore the SOM preserve the correlation between the *orders* of the indexes of the neurons with the distances of their corresponding data regions in the input space. So that the largest d_{ik} matches the largest allowed D_{ik} on the grid. As the grid is not scalable, the data points will be mapped to these pre-fixed grid positions to achieve maximum correlation. This is a qualitative or *nonmetric* scaling.

In the ViSOM, as the $\|i - k\|$ is proportional to $\|\mathbf{w}_i - \mathbf{w}_k\|$, so D_{ik} is $D(\mathbf{w}_i, \mathbf{w}_k)$ and is a function of $\|\mathbf{w}_i - \mathbf{w}_k\|$, which is the mapped distance referred to the input space in the metric MDS sense. Thus this shows why the ViSOM produces similar scaling results as to MDS as observed in [31, 32]. In other words it shows that the ViSOM is a metric MDS. The squared distance correlation terms in Eq. (21) have little different effect as to those non-squared ones in Eq. (18).

As the ViSOM is a discrete principal manifold, at the same time it is also a MDS. This implies that MDS and principal manifolds are performing the same underlying task at least in the context of data visualization and dimension reduction. Finding a principal manifold, a smooth curve/surface passing through the middle of the data [9, 3], may well result in a topographical scaling of the input space onto the lower dimensional manifold. On other hand, although MDS presents a useful scaling of the data on low dimension for visualization, it does not

provide the underlying mapping function, the manifold. The ViSOM can unite these two and provide both metric scaling and nonlinear manifold function.

3 Incremental ViSOM as metric MDS and nonlinear manifold

Although we have shown that SOM and ViSOM are nonmetric and metric MDS respectively, one the difficulties for SOM-based algorithms is to converge to highly nonlinear manifolds such as swissroll data. Indeed high nonlinearity poses problems for many MDS and data projection methods. Isomap [27] adopts geodesic distance within a neighborhood instead of global Euclidean distance in MDS to capture the nonlinearity of the data set. LLE [24] uses local linear embedding to approximate global nonlinearity.

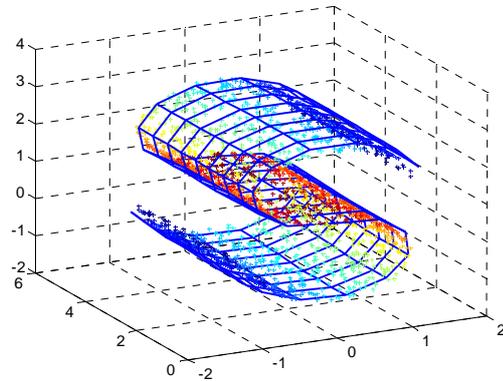
Here an incremental ViSOM or growing ViSOM (gViSOM) is proposed to extract nonlinear manifold and thus provide metric scaling to highly nonlinear manifold.

gViSOM algorithm:

- (1) Initial map can be small, say $N_0 \times N_0$, in either rectangular or hexagonal, though the latter is preferred for better nonlinear abilities. Place the initial map to a linear subspace of either global or local region of the data space. Set the desired resolution and neighborhood (locality) size.
- (2) Randomly draw a data sample from the data space and find the winning neuron with the shortest distance.
- (3) If the sample falls within the neighborhood, update the weights of the neurons of the neighborhood using the ViSOM principle.
- (4) At regular iteration intervals (e.g. 1000 iterations), if the growing condition is met (e.g. the data is underrepresented by the existing map), grow the map by adding one column or row to the side with the highest activities (e.g. winning frequencies). The added column or row is a linear extrapolation of existing map. Other growing structures can be used such as incrementing polygons instead of entire column or row for a free structure of the map and efficient use of neurons.
- (5) As in the ViSOM, at regular intervals (e.g. every certain iterations), refresh the map (neurons) probabilistically.
- (6) Check if the map has converged. If not go back to step 2); if so go to next step.
- (7) Project the data samples onto the map, either to the neurons or by the interpolation or LLP resolution enhancement method [33].

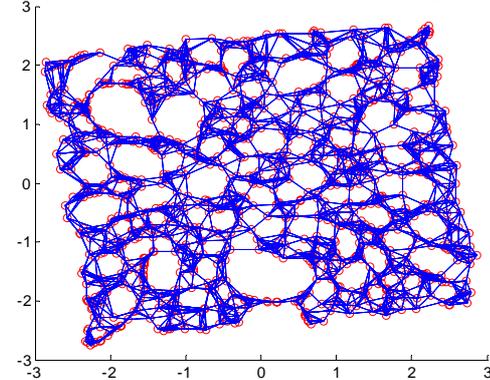
Several typical results are shown in Figs. 1-3. The proposed growing ViSOM is applied to 2000-point 3D nonlinear "S" shape dataset and the resulting embedding

is shown in Fig. 1(a), together with the data. The results of the Isomap and the LLE are shown in Fig. 1 (b) and (c) respectively.

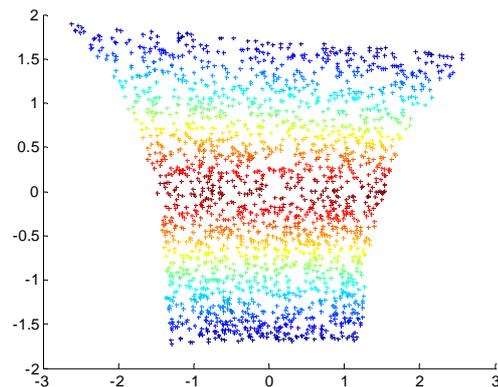


(a) gViSOM embedding in "S" data cloud.

Two-dimensional Isomap embedding (with neighborhood graph).



(b) Isomap scaling.



(c) LLE projection

Figure 1: Comparison of ViSOM with Isomap and LLE on "S" shape dataset.

Here 2000 data points were generated according to [24]. The gViSOM started with 5×5 grid and finally settled to 8×17 . The resolution was set to 0.5. Typical

results of the Isomap and LLE were obtained using the code provided by their authors [24, 27]. Projected data scatter on the final ViSOM with the LLP resolution enhancement is shown in Fig. 2. Correct topology and scale is clearly shown.

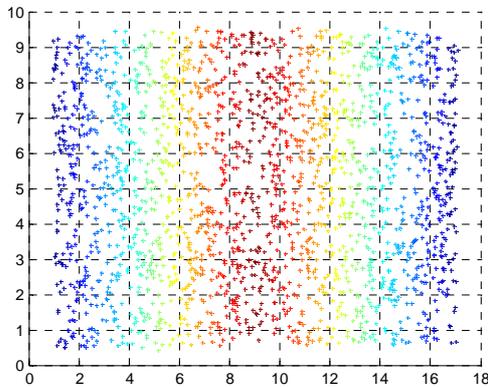


Figure 2: gViSOM projection with LLP resolution enhancement of the 'S' shape data. Note $\lambda=0.5$.

Fig. 3 shows the gViSOM embedding in the highly nonlinear swiss roll data set. Here the gViSOM started with 5×5 grid and finally settled to 18×70 . The resolution was set to 1.5. Isomap and LLE results, obtained using the code provided by their authors, are shown in Fig. 4 (a) and (b) respectively.

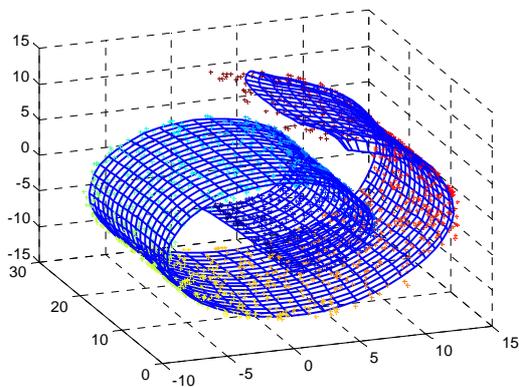


Figure 3: Swissroll data and gViSOM embedding

The advantages of the gViSOM are evident as it produces much more faithful metric scaling and extract the manifold function well. In addition, it can cope with discontinuities of the manifold (e.g. holes in manifold or separated manifold), and adapt to the dynamics (slow changes) of the manifold, which both the Isomap and LLE (and other scaling methods) have to re-capture with the entire data set once any part or whole is updated. Neural methods also have better abilities handling noise.

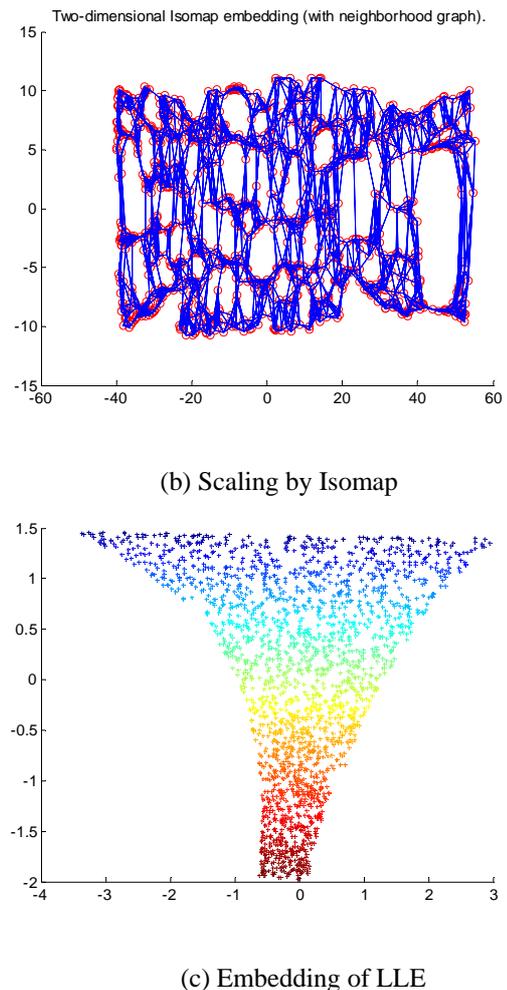


Figure 4: Isomap scaling and LLE projection on swiss roll data.

4 Conclusions

In this paper we have shown the effect of the SOM and ViSOM in performing multidimensional scaling. The SOM and ViSOM can be regarded as a generalized MDS and nonlinear manifold. While the standard SOM produces a nonmetric MDS, the ViSOM provide a metric MDS and principal manifold. In order to guide the ViSOM to a highly nonlinear manifold, a growing ViSOM is proposed to gradually and locally extract the manifold and scaling the data samples metrically on the map. Superior scaling performances have been achieved. Examples and comparison with the Isomap and LLE methods are given.

References

- [1] A.M. Bronstein, M.M. Bronstein, and R. Kimmel, "Generalized multidimensional scaling: A framework for

- isometry-invariant partial surface matching,” *PNAS*, vol. 103, pp. 1168–1172, 2006.
- [2] T.F. Cox & M.A.A. Cox, *Multidimensional Scaling*, Chapman & Hall, 1994.
- [3] P. Delicado, “Another look at principal curves and surfaces,” *J. Multivariate Analysis*, vol. 77, pp. 84–116, 2001.
- [4] P. Demartines & J. Héroult, “Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets,” *IEEE Trans. Neural Networks*, vol. 8, pp. 148–154, 1997.
- [5] R. Durbin & G. Mitchison, “A dimension reduction framework for understanding cortical maps,” *Nature*, vol. 343, pp. 644–647, 1990.
- [6] P.A. Estévez & C.J. Figueroa, “Online data visualization using the neural gas network,” *Neural Networks*, vol. 19, pp. 923–934, 2006.
- [7] A. Flexer, “Limitations of self-organizing maps for vector quantization and multidimensional scaling,” *Proc. NIPS’97*, pp. 445–451, 1997.
- [8] G.J. Goodhill & T. Sejnowski, “A unifying objective function for topographic mappings,” *Neural Computation*, vol. 9, pp. 1291–1303, 1997.
- [9] T. Hastie & W. Stuetzle, “Principal curves,” *J. American Statistical Association*, vol. 84, pp. 502–516, 1989.
- [10] T. Heskes, “Energy functions for self-organizing maps,” *Kohonen Maps*, E. Oja and S. Kaski (eds.), pp. 303–315, 1999.
- [11] J. Karhunen & J. Joutsensalo, “Generalization of principal component analysis, optimization problems, and neural networks,” *Neural Networks*, vol. 8, pp. 549–562, 1995.
- [12] T. Kohonen, “Self-organized formation of topologically correct feature map,” *Biological Cybernetics*, vol. 43, pp. 56–69, 1982.
- [13] T. Kohonen, “Self-organizing maps: optimization approaches,” *Artificial Neural Networks*, T. Kohonen, et al. (eds.), North-Holland, vol. 2, pp. 981–990, 1991.
- [14] T. Kohonen, *Self-Organizing Maps*, 2nd edition, Springer, 1997.
- [15] M.A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AICHE Journal*, vol. 37, pp. 233–243, 1991.
- [16] J. Lampinen & E. Oja, “Clustering properties of hierarchical self-organizing maps,” *J. Mathematical Imaging and Vision*, vol. 2, pp. 261–272, 1992.
- [17] D. Lowe & M.E. Tipping, “Neuroscale: novel topographic feature extraction using RBF networks,” *Neural Computing and Applications*, vol. 4, pp. 83–95, 1996.
- [18] S.P. Luttrell, “Derivation of a class of training algorithms,” *IEEE Trans. Neural Networks*, vol. 1, pp. 229–232, 1990.
- [19] E.C. Malthouse, “Limitations of nonlinear PCA as performed with generic neural networks,” *IEEE Trans. on Neural Networks*, vol. 9, pp. 165–173, 1998.
- [20] J. Mao & A. K. Jain, “Artificial neural networks for feature extraction and multivariate data projection,” *IEEE Trans. on Neural Networks*, vol. 6, pp. 296–317, 1995.
- [21] G. Mitchison, “A type of duality between self-organising maps and minimal wiring,” *Neural Computation*, vol. 7, pp. 25–35, 1995.
- [22] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
- [23] H. Ritter, T. Martinetz & K. Schulten, *Neural Computation and Self-organising Maps: An Introduction*. Addison-Wesley Publishing Company, 1992.
- [24] S.T. Roweis & L.K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.
- [25] J.W. Sammon, “A nonlinear mapping for data structure analysis,” *IEEE Trans. Computer*, vol. 18, pp. 401–409, 1969.
- [26] B. Schölkopf, A. Smola, & K.R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [27] J.B. Tenenbaum, V. de Silva & J.C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, pp. 2319–2323, 2000.
- [28] J. Venna & S. Kaski, “Local multidimensional scaling,” *Neural Networks*, vol. 19, pp. 889–899, 2006.
- [29] S. Wu & T.W.S. Chow, “PRSOM: A new visualization method by hybridizing multidimensional scaling and self-organizing map,” *IEEE Trans. Neural Networks*, vol. 16, pp. 1362–1380, 2005.
- [30] H. Yin, *Self-Organising Maps: Statistical Analysis, Treatment and Applications*, PhD Thesis, University of York, 1996.
- [31] H. Yin, “ViSOM-A novel method for multivariate data projection and structure visualization,” *IEEE Trans. Neural Networks*, vol. 13, pp. 237–243, 2002.
- [32] H. Yin, “Data visualization and manifold mapping using the ViSOM,” *Neural Networks*, vol. 15, pp. 1005–1016, 2002.
- [33] H. Yin, “Resolution enhancement for the ViSOM,” *Proc. Workshop on Self-Organizing Maps (WSOM’03)*, pp. 208–212, 2003.

