PRSOM: A New Visualization Method by Hybridizing Multidimensional Scaling and Self-Organizing Map

Sitao Wu and Tommy W. S. Chow, *Senior Member, IEEE*

***Abstract*—Self-organizing map (SOM) is an approach of nonlinear dimension reduction and can be used for visualization. It only preserves topological structures of input data on the projected output space. The interneuron distances of SOM are not preserved from input space into output space such that the visualization of SOM can be degraded. Visualization-induced SOM (ViSOM) has been proposed to overcome this problem. However, ViSOM is derived from heuristic and no cost function is assigned to it. In this paper, a probabilistic regularized SOM (PRSOM) is proposed to give a better visualization effect. It is associated with a cost function and gives a principled rule for weight-updating. The advantages of both multidimensional scaling (MDS) and SOM are incorporated in PRSOM. Like MDS, The interneuron distances of PRSOM in input space resemble those in output space, which are predefined before training. Instead of the hard assignment by ViSOM, the soft assignment by PRSOM can be further utilized to enhance the visualization effect. Experimental results demonstrate the effectiveness of the proposed PRSOM method compared with other dimension reduction methods.**

***Index Terms*—Curvilinear component analysis (CCA), multidimensional scaling (MDS), probabilistic regularized SOM (PRSOM), Sammon's mapping, self-organizing map (SOM), visualization-induced SOM (ViSOM).**

I. Introduction

PRINCIPAL component analysis (PCA) [1] and multidimensional scaling (MDS) [2] are the two most often used classical methods for dimension reduction and visualization. Linear PCA tries to best represent data by retaining most of its information content after dimension reduction. But linear PCA will lose certain useful information in the case of dealing with highly nonlinear data. Several nonlinear PCA methods have been proposed. Some are neural-network (NN)-based PCA [3]–[6]. Others adopt different criteria for nonlinear data projection [7]–[10], where principal curves [7] and principal surfaces [8] will be discussed later. MDS is a method that projects high-dimensional data to a low (usually two-) dimensional space and preserves the interpoint distances among data as much as possible. MDS gives a cost function associated with the coordinates of input and output points. The final result is obtained by moving the output points in accordance with an optimization problem. Sammon's mapping [11] is one of the popular MDS methods. An NN-based Sammon's mapping [12] and other methods related to Sammon's mapping were proposed in the last decade. However, the computational complexity of MDS is so heavy that it is not suitable for large data sets.

Self-organizing map (SOM), proposed by Kohonen [13], [14], can be used for dimension reduction, vector quantization, and visualization. The recent SOM-based applications in the literature can be seen in [36]–[41]. SOM quantizes input data to a small number of neurons and still preserves the topology of input data. Indeed, SOM can be seen as discrete approximation of principal surfaces in input space [15], [16]. Some visualization methods based on SOM were proposed [17]–[22]. One of the disadvantages of SOM is that it only preserves topological structures of input data on the output space. The interneuron distances of SOM are not preserved from input space into output space such that the visualization of SOM can be degraded. Recently ViSOM, a new visualization method extending SOM, was proposed [23], [24]. ViSOM regularizes the interneuron distances such that the interneuron distances in input space resemble those in output space after completion of training. Since the output topology in ViSOM is predefined as a regular two-dimensional (2-D) grid, the trained neurons are almost regularly distributed in input space. ViSOM is able to preserve the topological information as well as the interneuron distances. Experimental results presented in [23] and [24] show that ViSOM delivers excellent visual exhibition compared with SOM and other visualization methods. However, there is no cost function assigned to ViSOM, which makes the derivation of the weight-updating rule rather heuristic.

In this paper, a new visualization method, probabilistic regularized SOM (PRSOM), is proposed. SOM and MDS are hybridized into PRSOM such that PRSOM reduces the computational burden by using SOM and preserves the interneuron distances after dimension reduction by using MDS. Unlike the hard assignment in SOM and ViSOM, the assignment of PRSOM is soft such that an input datum belongs to a neuron with certain probability. In PRSOM, the sequential weight-updating rule is extended from ViSOM to an optimization of a cost function. Under certain circumstance, ViSOM can be considered as a special case and an accelerated one of PRSOM. In addition to performing visualization by using an assignment method [18] in ViSOM, the probabilistic assignment can be utilized in PRSOM. The accumulated probability for each neuron can be displayed with a coloring scheme on a 2-D output map. It is like the U-matrix method [17] and the visualization method in [19], and may reveal the clustering tendency of input data. PRSOM can also

Manuscript received April 14, 2004; revised January 12, 2005. This work was supported by the City University of Hong Kong under Project 7001599-570.

The authors are with the Department of Electric Engineering, City University of Hong Kong, Hong Kong, China (e-mail: eetchow@cityu.edu.hk).

Digital Object Identifier 10.1109/TNN.2005.853574

be considered as a discrete approximation of principal surfaces like SOM and ViSOM. Like regularization terms used in supervised learning, quantization, and feature extraction [25] to simplify or smooth function and to avoid overfitting, the surfaces of PRSOM are smoothed to have a good mapping effect. Experimental results show that the proposed PRSOM is a promising and effective approach for dimension reduction and visualization.

In Section II, MDS, SOM, ViSOM, and principal curves/surfaces are briefly reviewed. In Section III, PRSOM is presented in detail. In Section IV, experimental results demonstrate that the proposed algorithm is able to perform visualization effectively. Conclusions are drawn in Section V.

II. MDS, SOM, VISOM, AND PRINCIPAL CURVES/SURFACES

A. Multidimensional Scaling (MDS)

MDS is a traditional method used for dimension reduction and visualization. The general objective of MDS is to preserve the interpoint distances in a low (usually 2-D) output space. Let δ_{ij} denote the similarity (or dissimilarity) between two points i and j in input space and d_{ij} denote that between the two points in the corresponding output space. The following sum-of-square-error functions (nonmetric scaling), often called stress, are all reasonable candidates [26]

$$J_{ee} = \frac{\sum_{i<j}(d_{ij}-\delta_{ij})^2}{\sum_{i<j}\delta_{ij}^2} \tag{1}$$

$$J_{ff} = \sum_{i<j}\left(\frac{d_{ij}-\delta_{ij}}{\delta_{ij}}\right)^2 \tag{2}$$

$$J_{ef} = \frac{1}{\sum_{i<j}\delta_{ij}}\sum_{i<j}\left(\frac{(d_{ij}-\delta_{ij})^2}{\delta_{ij}}\right). \tag{3}$$

While J_{ee} used in [27] emphasizes large errors (regardless of whether the distance δ_{ij} are large or small), J_{ff} emphasizes the large fractional errors (regardless of whether the errors $|d_{ij}-\delta_{ij}|$ are large or small). A useful compromise on J_{ef} is to emphasize large products of errors and fractional errors. J_{ef} is also known as the cost function of Sammon's mapping [11].

Once a cost function is selected, an optimal configuration of output data is obtained by minimizing the cost function. Such a configuration can be sought by a standard gradient-descent procedure.

However, MDS has certain disadvantages. First, its computational complexity is $O(M^2)$, where M is the number of input data. Thus, it is impractical to perform MDS on a large data set. Secondly, no explicit mapping function exists in MDS. Therefore, MDS lacks the ability of mapping new input data to output space unless all input data are recomputed. Thirdly, MDS treats large distances in a similar way to small ones. This causes problems if the data to be visualized are high dimensional. This problem can be avoided by SOM, which preserves local distances as discussed in the next subsection. Finally, there are many local minima on the error surface. Usually, MDS is inevitable to get stuck in certain local minimum.

Curvilinear component analysis (CCA) [28] was proposed as an improvement of MDS. It favors local topology conservation. The purpose of CCA is to give a revealing representation of data in low dimension. The cost function of CCA is

$$E = \frac{1}{2}\sum_i\sum_{j\neq i}(\delta_{ij}-d_{ij})^2F(d_{ij},\lambda) \tag{4}$$

where $F(d_{ij},\lambda)$ is chosen as a bounded and monotonically decreasing function. For example, $F(d_{ij},\lambda)$ can be a simple step function

$$F(d_{ij},\lambda) = \begin{cases} 1, & \text{if } d_{ij}\leq\lambda \\ 0, & \text{if } d_{ij}>\lambda \end{cases} \tag{5}$$

where λ is a parameter controlling the scope of local structure. The computational complexity of CCA is $O(M)$, which is less than that of MDS.

B. SOM

SOM consists of N neurons located at a regular low-dimensional grid, usually a 2-D grid. The lattice of the grid is either hexagonal or rectangle. The basic SOM algorithm is iterative. Each neuron i has a d-dimensional feature vector $w_i = [w_{i1},\ldots,w_{id}]^T$. At each training step t, a sample data vector $x(t)$ is randomly chosen from a training set. Distances between $x(t)$ and all the feature vectors are computed. The winning neuron, denoted by c, is the neuron with the feature vector closest to $x(t)$

$$c = \arg\min_i \|x(t)-w_i\|,\ i\in\{1,\ldots,N\}. \tag{6}$$

A set of neighboring nodes of the winning node is denoted as N_c. We define $h_{jc}(t)$ as the neighborhood kernel function around the winning neuron c at time t. The neighborhood kernel function is a nonincreasing function with time and with the distance between neuron j and the winning neuron c in output space. The kernel can be taken as a Gaussian function

$$h_{jc}(t) = e^{-\frac{\|\text{Pos}_j-\text{Pos}_c\|^2}{2\sigma(t)^2}},\ j\in N_c \tag{7}$$

where Pos_j is the coordinates of neuron j on the output grid.

The weight-updating rule in the sequential SOM algorithm can be written as

$$w_j(\text{t}+1) = \begin{cases} w_j(t)+\varepsilon(t)h_{jc}(t)(x(t)-w_j(t)), & \forall j\in N_c \\ w_j(t), & \text{otherwise} \end{cases}. \tag{8}$$

Both the learning rate $\varepsilon(t)$ and the neighborhood $\sigma(t)$ decrease monotonically with time.

One of the disadvantages of SOM is that it only preserves the topology of input data. Since the neurons in output space are always predefined in a rectangular or hexagonal grid, the interneuron distances of SOM are apparently not preserved.

There are many variants of SOM. Soft topographic vector quantization (STVQ) [29], [30] is the one that motivates the core

TABLE I
COMPARISON OF MAPPING BY USING THE RELATIVE STANDARD DEVIATION MEASUREMENT

RSD / Algorithms (computational complexity) / Data sets	PRSOM $(O(MN^2))$	ViSOM $(O(MN^2))$	SOM $(O(MN))$	CCA $(O(M))$	Sammon's mapping $(O(M^2))$
3-D data (3-D)	0.07	0.19	0.54	0.30	0.50
UK university data(7-D)	0.06	0.18	0.66	0.37	0.48
Winsconsin breast cancer data (9-D)	0.04	0.24	0.65	0.63	0.91
Wine data (13-D)	0.03	0.20	0.41	0.77	0.63

idea of PRSOM. The STVQ algorithm gives a cost function (soft quantization error) as follows:

$$F = \sum_{t=1}^{M}\sum_{i=1}^{N} P_i\left(x(t)\right)\sum_{j=1}^{N} h_{ij} D\left(x(t), w_j\right) \tag{9}$$

where M and N are the numbers of input data and neurons, respectively, $P_i(x(t))$ is the probability of assigning an input $x(t)$ to neuron i, h_{ij} is a fixed neighborhood function satisfying $\sum_{j=1}^{N} h_{ij} = 1$, and $D(x(t), w_j)$ is the quantization error between the input $x(t)$ and the weight w_j of neuron j, defined by $D(x(t), w_j) = (1/2)\|x(t) - w_j\|^2$. The entropy of the probabilistic assignments is

$$S = -\sum_{t=1}^{M}\sum_{i=1}^{N} P_i\left(x(t)\right)\ln\left(P_i\left(x(t)\right)\right). \tag{10}$$

In order to maximize the entropy in (10) and minimize the cost function in (9), the regularized cost function to be minimized, under the constraint $\sum_{i=1}^{N} P_i(x(t)) = 1$, becomes

$$E = \beta F - S \tag{11}$$

where β is the fixed regularization parameter.

Taking the gradient of E with respect to $P_j(x(t))$ and w_j to zero, i.e.,

$$\begin{aligned}\frac{\partial E}{\partial P_j\left(x(t)\right)} &= \beta\sum_{i=1}^{N} h_{ji} D\left(x(t), w_i\right)) + \ln P_j\left(x(t)\right) + 1 \\ &= 0,\end{aligned} \tag{12}$$

$$\begin{aligned}\frac{\partial E}{\partial w_j} &= -\beta\sum_{t=1}^{M}\sum_{i=1}^{N} h_{ij} P_i\left(x(t)\right)\left(x - w_j\right) \\ &= 0\end{aligned} \tag{13}$$

the weights can be obtained by the following iterative steps in the EM algorithm [31]:

1) E step

$$P_j\left(x(t)\right) = \frac{\exp\left(-\beta\sum_{i=1}^{N} h_{ji} D(x(t), w_i)\right)}{\sum_{k=1}^{N}\exp\left(-\beta\sum_{i=1}^{N} h_{ki} D(x(t), w_i)\right)}, \quad j = 1, 2, \ldots, N \tag{14}$$

2) M step

$$w_j = \frac{\sum_{t=1}^{M} x(t)\sum_{i=1}^{N} h_{ij} P_i\left(x(t)\right)}{\sum_{t=1}^{M}\sum_{i=1}^{N} h_{ij} P_i\left(x(t)\right)}, \quad j = 1, 2, \ldots, N. \tag{15}$$

In (14), β is a parameter of inverse temperature. The optimized weights can be obtained by deterministic annealing from low to high values of β [29] so as to avoid being stuck at local minima of the cost function in (11). The above steps of the STVQ algorithm are of batch type and can be modified to the batch type SOM [32] if we set h_{ij} to a delta function δ_{ij}, and $\beta \to \infty$ in (14) [30]. It is noted that the neighborhood function h_{ij} is kept constant in STVQ, while it decreases with time in SOM and ViSOM.

C. ViSOM

ViSOM [23], [24] is a new algorithm to preserve topology as well as interneuron distances. The final map can be seen as a smooth net embedded in input space. The distances between any pairs of neurons in input space resemble those in output space. ViSOM uses the same network architecture as SOM. The only difference between the two networks is that the neighboring neurons of winner neuron are updated differently. In SOM, the weight-updating rule is (8). The weight-updating rule for the neighboring neurons of winner neuron in ViSOM is

$$\begin{aligned}w_i(\mathrm{t}+1) = w_i(t) + \varepsilon(t) h_{ic}(t)\left(\left[x(t) - w_c(t)\right]\right. \\ \left. + \left[w_c(t) - w_i(t)\right]\left(\frac{d_{ci} - \lambda\Delta_{ci}}{\lambda\Delta_{ci}}\right)\right), \quad \forall i \in N_c\end{aligned} \tag{16}$$

where d_{ci} and Δ_{ci} are the distances between the neuron c and i in input space and output space, respectively, and λ is a resolution parameter.

The basic idea behind ViSOM is that the force $F_{ix} = x(t) - w_i(t)$ can be decomposed into two parts: $F_{ix} = [x(t) - w_c(t)] + [w_c(t) - w_i(t)] = F_{cx} + F_{ci}$. F_{cx} is a force from the winner neuron c to the input x. F_{ci} is a lateral force from the neuron i

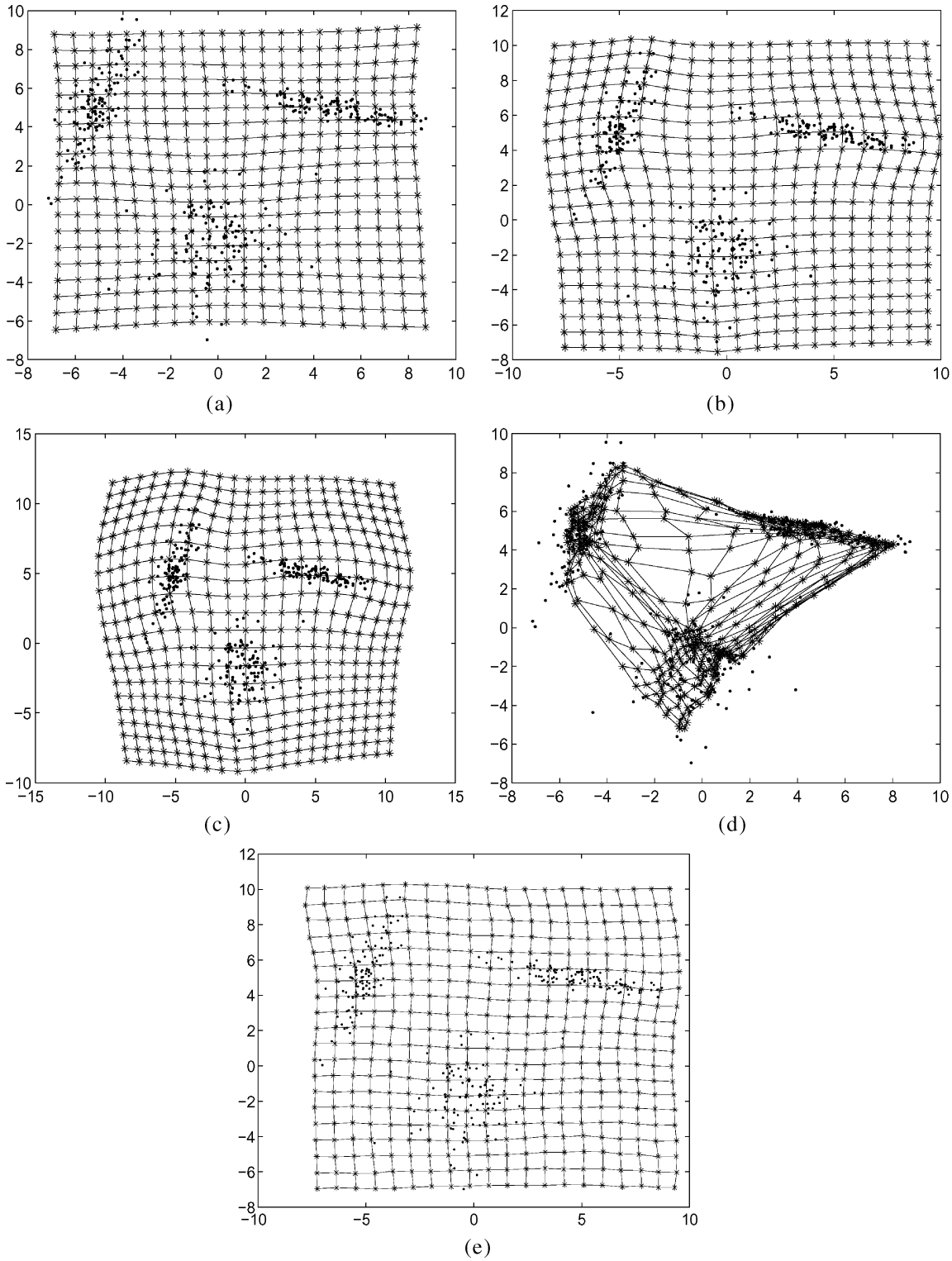


Fig. 1. The maps in the input space for the 2-D synthetic data set. (a) PRSOM (map size 20 × 20, $\lambda = 0.5$). (b) PRSOM (map size 20 × 20, $\lambda = 0.9$). (c) PRSOM (map size 20 × 20, $\lambda = 1.5$). (d) SOM (map size 20 × 20). (e) ViSOM (map size 20 × 20, $\lambda = 0.9$).

to the winner neuron c. ViSOM constrains the lateral force F_{ci} by multiplying a coefficient $(d_{ci} - \lambda\Delta_{ci})/\lambda\Delta_{ci}$. The objective is to maintain the preservations of distances between any neurons. The discrete surface constructed by neurons is then regularized to be smooth for good visualization. In order to keep the rigidity of final maps, the final neighborhood size σ_f should not include only the winner neurons. The larger the σ_f, the flatter the map in input space. The resolution parameter λ controls the resolution of the map. Small values of λ generate maps with high resolution while large values of λ generate maps with low resolution.

D. Principal Curves and Surfaces

Principal curves [7] generalize a principal component line, providing a smooth one-dimensional curved approximation to a

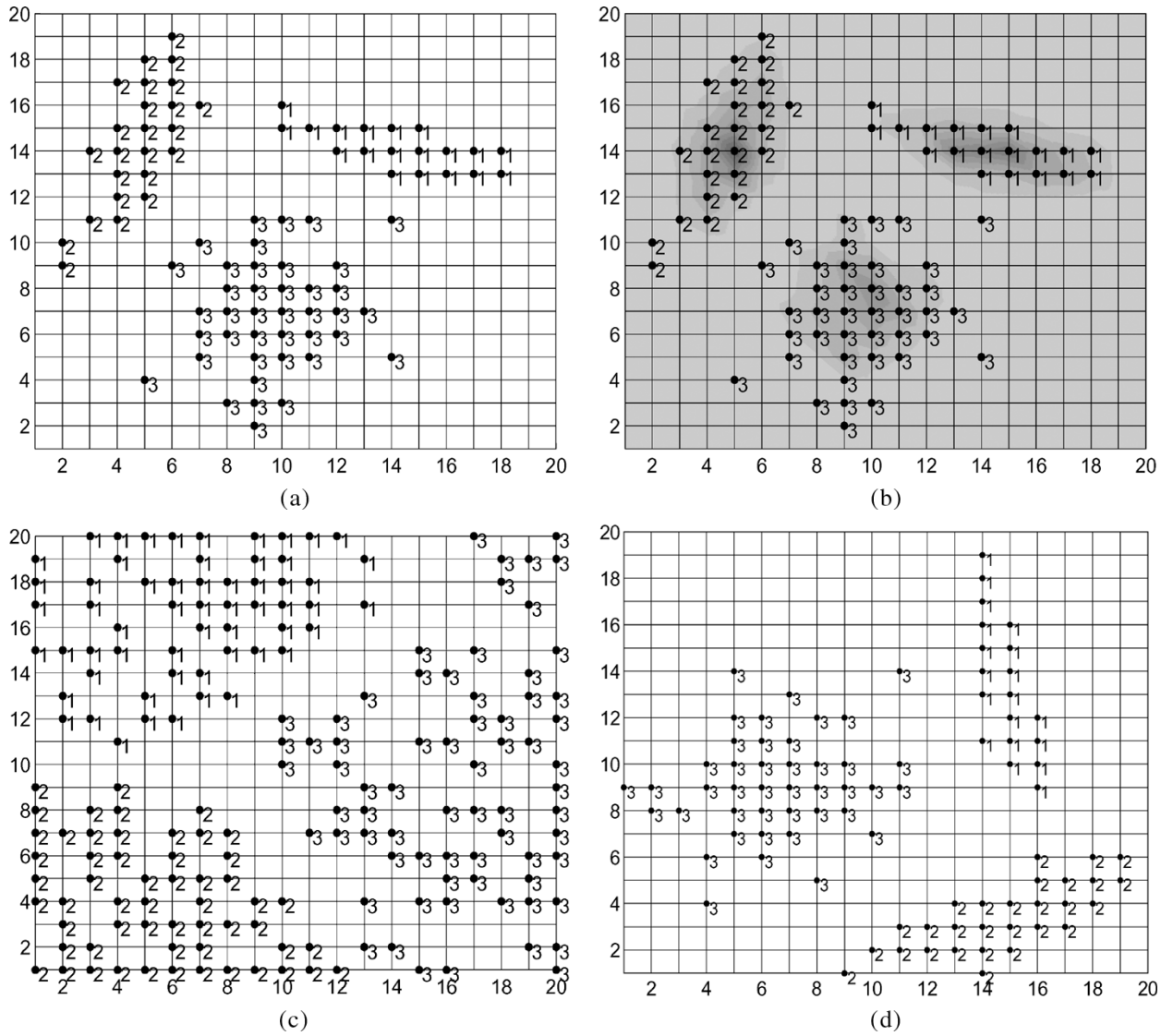


Fig. 2. Visualization of the 2-D synthetic data set. Activated neurons are plotted with big dots and corresponding number of Gaussians. (a) The assignment method of PRSOM (map size 20×20, $\lambda = 0.9$). (b) AP matrix method of PRSOM (map size 20×20, $\lambda = 0.9$). (c) The assignment method of SOM (map size 20×20). (d) The assignment method of ViSOM (map size 20×20, $\lambda = 0.9$).

set of data points. Principal surfaces [8] are more general, providing a curved manifold approximation of dimension two or more. SOM is related to the discrete principal curves/surfaces [15], [16]. The kernel smoothing processes of principal curves and SOM are similar

$$\text{Kernel regression}: \ F(\rho) = \frac{\sum_{k=1}^{M} x_k \kappa(\rho, \rho_k)}{\sum_{k=1}^{M} \kappa(\rho, \rho_k)} \tag{17}$$

$$\text{SOM}: \quad w_i = \frac{\sum_{k=1}^{M} x_k h_{ci}(k)}{\sum_{k=1}^{M} h_{ci}(k)} \tag{18}$$

where ρ and ρ_k are the densities of the input data x and x_k, respectively, $\kappa(\bullet, \bullet)$ is the kernel function, and $h_{ij}(\bullet)$ is the neighborhood function.

In [24], ViSOM is also a discrete approximation of principal surfaces. The smoothing process is

$$\text{ViSOM}: \ w_c = \frac{\sum_{k=1}^{L} x_k h_{ci}(k)}{\sum_{k=1}^{L} h_{ci}(k)} \tag{19}$$

where neuron c is the winner neuron, x_k is one of the input data selecting neuron c as winner neuron, and L is the total number of input data selecting neuron c as winner neuron.

III. PRSOM

A. Cost Function of PRSOM

PRSOM tries to minimize soft quantization error like STVQ. But it also has a regularization term that makes the surfaces constructed by neurons smooth for good visualization.

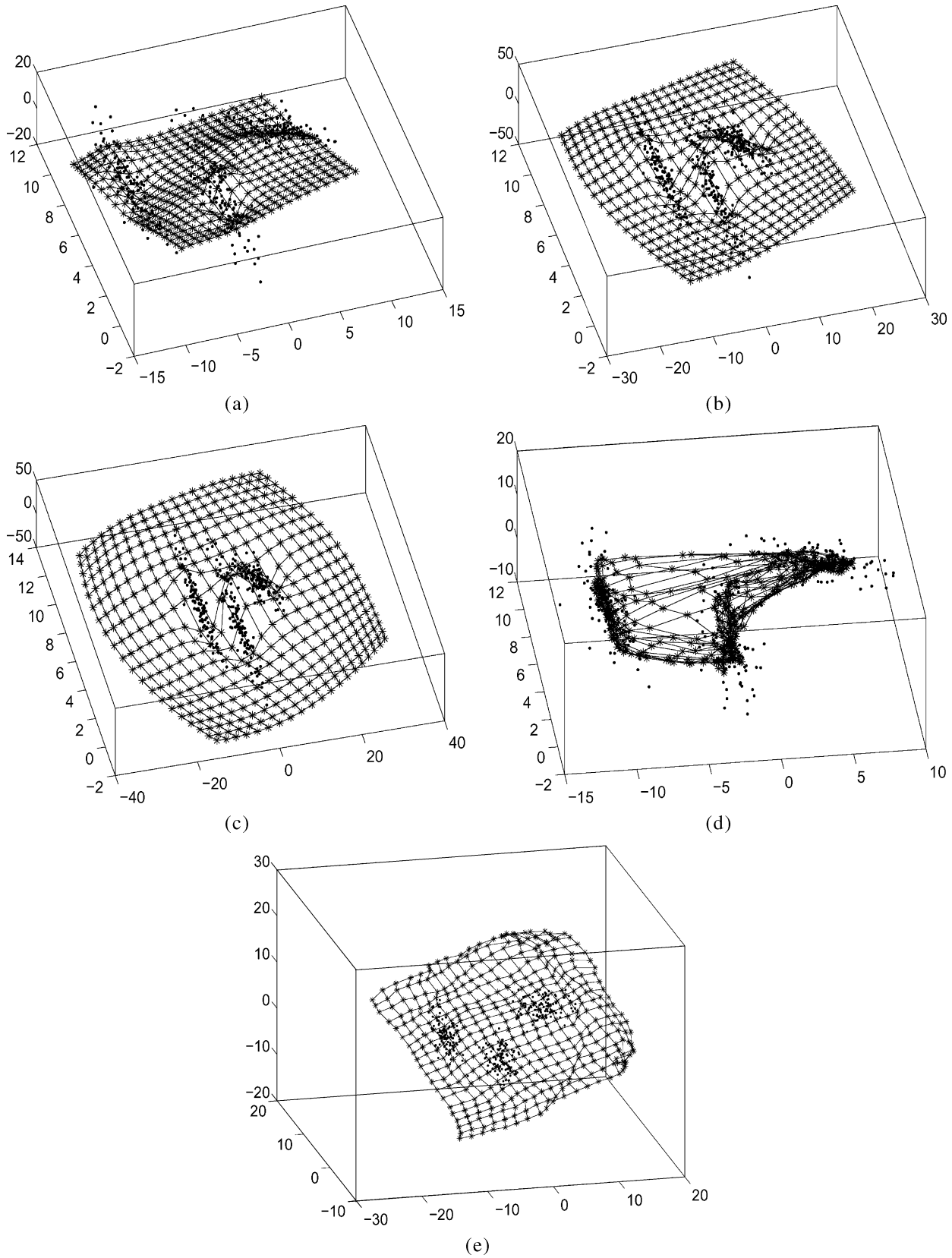


Fig. 3. The maps in the input space for the 3-D synthetic data set generated by (a) PRSOM (map size 20×20, $\lambda = 1.0$); (b) PRSOM (map size 20×20, $\lambda = 6.0$); (c) PRSOM (map size 20×20, $\lambda = 12.0$); (d) SOM (map size 20×20); and (e) ViSOM (map size 20×20, $\lambda = 1.7$).

First, we introduce noised probabilistic assignments. Let $p_j(x(t))$ denote the noised probabilistic assignment of neuron j

$$p_j(x(t)) = \sum_{i=1}^{N} h_{ij} P_i(x(t)) \tag{20}$$

where $P_j(x(t))$ is the probabilistic assignment of neuron j for input $x(t)$ and h_{ij} is a neighborhood constant satisfying $\sum_{j=1}^{N} h_{ij} = 1$. Here the term "noised" means $p_j(x(t))$ is affected by leaked probabilistic assignments from other neighboring neurons. Therefore $p_j(x(t))$ is the probabilistic assignment of neuron j that considers the effects of other neurons. Note that $p_j(x(t))$ can be considered as a weight since

$$\begin{aligned}\sum_{j=1}^{N} p_j(x(t)) &= \sum_{j=1}^{N}\sum_{i=1}^{N} h_{ij} P_i(x(t)) \\ &= \sum_{i=1}^{N} P_i(x(t)) \sum_{j=1}^{N} h_{ij} = 1.\end{aligned} \tag{21}$$

The cost function of PRSOM is then soft vector quantization error

$$\begin{aligned}F_{vq} &= \frac{1}{2}\sum_{t=1}^{M}\left\|\sum_{j=1}^{N} p_j(x(t))\,[x(t) - w_j]\right\|^2 \\ &= \frac{1}{2}\sum_{t=1}^{M}\left\|x(t) - \sum_{j=1}^{N} p_j(x(t))\, w_j\right\|^2\end{aligned} \tag{22}$$

which computes the sum of square errors between the input data and the average weights for all input data.

To control the complexity of the above model, or ensure the solution is simple or smooth, we added the following metric MDS term:

$$F_{\text{reg}} = \frac{1}{8}\sum_{t=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{N} p_j(x(t))\, p_m(x(t)) \frac{\left(d_{jm}^2 - \lambda\Delta_{jm}^2\right)^2}{\left(\lambda\Delta_{jm}^2 + I_{jm}\right)} \tag{23}$$

where $d_{jm} = \|w_j - w_m\|$ is the distance in input space, Δ_{jm} is the corresponding distance between neuron j and m in 2-D output space, λ is a resolution parameter like ViSOM, and the identity matrix I is introduced to avoid the case that the denominator of the fractional term would be zero when $j = m$.

F_{reg} in (23) tries to preserve pairwise distance of neurons in input and output space. It emphasizes large products of errors and fractional errors like Sammon's mapping. It also can be considered as the restriction of PRSOM for the smoothness of discrete approximation of the principal surfaces.

Then a regularized cost function of PRSOM is

$$\begin{aligned}E = F_{vq} + \gamma F_{reg} &= \frac{1}{2}\sum_{t=1}^{M}\left\|\sum_{j=1}^{N} p_j(x(t))\,[x(t) - w_j]\right\|^2 \\ &+ \frac{\gamma}{8}\sum_{t=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{N} p_j(x(t))\, p_m(x(t)) \frac{\left(d_{jm}^2 - \lambda\Delta_{jm}^2\right)^2}{\left(\lambda\Delta_{jm}^2 + I_{jm}\right)}\end{aligned} \tag{24}$$

where γ is a regularization parameter.

B. Weight-Updating and Probability Assignment of PRSOM

Equation (24) can be reexpressed as

$$E = \sum_{t=1}^{M} E(t)$$

where

$$\begin{aligned}E(t) = \frac{1}{2}&\left\|\sum_{j=1}^{N} p_j(x(t))\,[x(t) - w_j]\right\|^2 \\ &+ \frac{\gamma}{8}\sum_{j=1}^{N}\sum_{m=1}^{N} p_j(x(t))\, p_m(x(t)) \frac{\left(d_{jm}^2 - \lambda\Delta_{jm}^2\right)^2}{\left(\lambda\Delta_{jm}^2 + I_{jm}\right)}.\end{aligned}$$

Since the left and right terms in $E(t)$ are always positive, the minimization of E is equal to the minimization of each $E(t)$.

Taking the gradient of $E(t)$ with respect to w_j, i.e.,

$$\begin{aligned}\frac{\partial E(t)}{\partial w_j} = &-p_j(x(t))\sum_{i=1}^{N} p_i(x(t))\,(x - w_i) - \gamma p_j(x(t)) \\ &\times \sum_{i=1}^{N} p_i(x(t))\,(w_i - w_j)\left(\frac{d_{ij}^2 - \lambda\Delta_{ij}^2}{\lambda\Delta_{ij}^2 + I_{ij}}\right)\end{aligned}$$

the following weight-updating rule is obtained:

$$\begin{aligned}w_j(t+1) &= w_j(t) - \varepsilon(t)\frac{\partial E(t)}{\partial w_j} \\ &= w_j(t) + \varepsilon(t) p_j(x(t)) \\ &\quad\times\left[\sum_{i=1}^{N} p_i(x(t))\,([x(t) - w_i(t)]\right. \\ &\qquad\left.+\gamma\,[w_i(t) - w_j(t)]\left(\frac{d_{ij}^2 - \lambda\Delta_{ij}^2}{\lambda\Delta_{ij}^2 + I_{ij}}\right)\right).\end{aligned} \tag{25}$$

In (25), $\varepsilon'(t) = \varepsilon(t)p_j(x(t))$ is the learning rate of the weight-updating rule of PRSOM. To avoid small values of the learning rate $\varepsilon'(t)$, the noised probabilistic assignment (or fuzzy neighborhood function) $p_j(x(t)) = \sum_{i=1}^{N} h_{ij}(t) P_i(x(t))$ can be set $p'_j(x(t)) = p_j(x(t))/\max_k(p_k(x(t)))$. Then the resultant updating rule is

$$\begin{aligned}w_j(t+1) = w_j(t) + \varepsilon(t) p'_j(x(t))&\left[\sum_{i=1}^{N} p_i(x(t))\,([x(t) - w_i(t)]\right. \\ &\left.+\gamma\,[w_i(t) - w_j(t)]\left(\frac{d_{ij}^2 - \lambda\Delta_{ij}^2}{\lambda\Delta_{ij}^2 + I_{ij}}\right)\right].\end{aligned} \tag{26}$$

The probabilistic assignment $P_i(x(t))$ is (14) in STVQ. But the additional parameter β, the inverse temperature, must be carefully selected and tuned from low to high values. If we used the same technique in PRSOM, we add the entropy into the cost function (24)

$$E = \beta(F_{vq} + \gamma F_{\text{stress}}) + \sum_{t=1}^{M}\sum_{i=1}^{N} P_i(x(t)) \ln(P_i(x(t))) \tag{27}$$

where β is a fixed regularization parameter. Taking the gradient of (27) with respect to $P_j(x(t))$, we obtained the expression of $P_j(x(t))$ shown in (28) at the bottom of the next page, which is a fixed-point iteration. However, (28) may not converge in practical situations.

A more convenient and heuristic way to compute $P_i(x(t))$ can be taken as

$$P_j\left(x(t)\right)=\frac{1}{C}\left(\frac{1}{\left\|\sum_{k=1}^{N}h_{jk}\left(x(t)-w_k\right)\right\|^2}\right) \tag{29}$$

where C is a normalization constant. No iteration is needed in (29). Since neighborhood function h_{jj} is much larger than any other $h_{jk}(k\neq j)$, $P_j(x(t))$ achieves the highest probability assignment if w_j is the feature vector of the nearest neuron from the input $x(t)$. Equation (29) is then reasonable in that the closer a neuron is to input, the higher the assignment probability is.

C. Connections to STVQ, SOM, and ViSOM

The cost function (9) of STVQ can be rewritten as

$$\begin{aligned}F&=\sum_{t=1}^{M}\sum_{j=1}^{N}\left(\sum_{i=1}^{N}h_{ij}P_i\left(x(t)\right)\right)D\left(x(t),w_j\right)\\&=\sum_{t=1}^{M}\sum_{j=1}^{N}p_j\left(x(t)\right)D\left(x(t),w_j\right)\\&=\frac{1}{2}\sum_{t=1}^{M}\sum_{j=1}^{N}p_j\left(x(t)\right)\left\|x(t)-w_j\right\|^2.\end{aligned} \tag{30}$$

If $\sum_{j=1}^{N}p_j(x(t))\|x(t)-w_j\|^2$ is considered as the soft topographic quantization error for the input $x(t)$, F is the sum of the soft topographic quantization errors for the whole data.

F_{vq} in (22) of PRSOM can be expanded as the following:

$$\begin{aligned}F_{vq}&=\frac{1}{2}\sum_{t=1}^{M}\left\|\sum_{j=1}^{N}p_j\left(x(t)\right)\left[x(t)-w_j\right]\right\|^2\\&=\frac{1}{2}\sum_{t=1}^{M}\left(\sum_{j=1}^{N}p_j\left(x(t)\right)\left[x(t)-w_j\right]\right)^T\\&\quad\times\left(\sum_{j=1}^{N}p_j\left(x(t)\right)\left[x(t)-w_j\right]\right)\\&=\frac{1}{2}\sum_{t=1}^{M}\left\{\sum_{i=1}^{N}p_i^2\left(x(t)\right)\left\|x(t)-w_i\right\|^2+\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}p_j\left(x(t)\right)\right.\\&\qquad\left.\times p_k\left(x(t)\right)\left[x(t)-w_j\right]^T\left[x(t)-w_k\right]\right\}.\end{aligned} \tag{31}$$

The first term of (31) minimizes the probabilistic quantization error like (30) in STVQ. In order to minimize the second term of (31), the angles between every pairs of $x(t)-w_j$ and $x(t)-w_k$ $(k\neq j)$ should be obtuse, i.e., neurons in input space should be repelled from one another around input data as far as possible. The topographic information in PRSOM is embodied in the lateral function h_{ij}, which can cause quantization error of neurons j leaking into that of neuron i.

We can express the weight-updating in (15) as an online version. First, we introduce a state variable $B_j(M)$ for neuron j at iteration $M: B_j(M)=\sum_{t=1}^{M}\sum_{i=1}^{N}h_{ij}P_i(x(t))$. Then

$$\begin{aligned}w_j(M-1)&=\frac{\sum_{t=1}^{M-1}x(t)\sum_{i=1}^{N}h_{ij}P_i\left(x(t)\right)}{\sum_{t=1}^{M-1}\sum_{i=1}^{N}h_{ij}P_i\left(x(t)\right)}\\&=\frac{\sum_{t=1}^{M-1}x(t)\sum_{i=1}^{N}h_{ij}P_i\left(x(t)\right)}{B_j(M-1)}.\end{aligned} \tag{32}$$

From (32), the following equation is satisfied:

$$w_j(M-1)B(M-1)=\sum_{t=1}^{M-1}x(t)\sum_{i=1}^{N}h_{ij}P_i\left(x(t)\right). \tag{33}$$

Then the weight of neuron j can be sequentially adapted by (34) as shown at the bottom of the next page,where $\varepsilon_i(t)$ is the learning rate decreasing to zero during the training course and $p_i(x(t))$ is the neighborhood function. This updating rule can be also derived by taking the gradient of the cost function in (30).

It should be noted that the SOM algorithm can be expressed by [33]

$$\begin{aligned}w_j(t)=w_j(t-1)+\varepsilon(t)&\left(\sum_{i=1}^{N}h_{ij}(t)\delta_i\left(x(t)\right)\right)\\&\times\left(x(t)-w_j(t-1)\right)\end{aligned} \tag{35}$$

where the activation function $\delta_i(x(t))$ is given by

$$\delta_j\left(x(t)\right)=\begin{cases}1, & \text{when } \|x(t)-w_j\|\leq\|x(t)-w_i\|,\ \forall i\\0, & \text{otherwise}\end{cases}. \tag{36}$$

The online updating form of STVQ then uses soft assignment, while SOM uses hard assignment.

$$P_j\left(x(t)\right)=\frac{\exp\left\{-\beta\left\{\left(\sum_{k=1}^{N}\sum_{i=1}^{N}h_{ik}P_i\left(x(t)\right)(x(t)-w_k)\right)^T\left[x(t)-\sum_{i=1}^{N}w_ih_{ji}\right]+\frac{\gamma\sum_{k=1}^{N}\left(\sum_{i=1}^{N}h_{ik}P_i(x(t))\right)\sum_{m=1}^{N}h_{jm}\left(d_{mk}^2-\lambda\Delta_{mk}^2\right)^2}{\left(\lambda\Delta_{mk}^2+I\right)}\right\}\right\}}{\sum_{n=1}^{N}\exp\left\{-\beta\left\{\left(\sum_{k=1}^{N}\sum_{i=1}^{N}h_{ik}P_i(x(t))(x(t)-w_k)\right)^T\left[x(t)-\sum_{i=1}^{N}w_ih_{ni}\right]+\frac{\gamma\sum_{k=1}^{N}\left(\sum_{i=1}^{N}h_{ik}P_i(x(t))\right)\sum_{m=1}^{N}h_{nm}\left(d_{mk}^2-\lambda\Delta_{mk}^2\right)^2}{\left(\lambda\Delta_{mk}^2+I\right)}\right\}\right\}} \tag{28}$$

The weight-updating rule in (16) in ViSOM can be reexpressed by using hard assignment

$$w_j(\mathrm{t}+1) = w_j(t) + \varepsilon(t)\left(\sum_{i=1}^{N} h_{ij}(t)\delta_i\left(x(t)\right)\right) \times \left(\sum_{i=1}^{N}\delta_i(x(t)([x(t)-w_i(t)]+[w_i(t)-w_j(t)] \times \left(\frac{d_{ij}-\lambda\Delta_{ij}}{\lambda\Delta_{ij}+I_{ij}}\right)\right)\right) \quad (37)$$

where identity matrix I is the same as that used in PRSOM.

Equation (37) can be written in a probabilistic form

$$\begin{aligned} w_j(\mathrm{t}+1) &= w_j(t) + \varepsilon(t)\left(\sum_{i=1}^{N} h_{ij}(t)P_i(x(t))\right) \times\left[\sum_{i=1}^{N} p_i(x(t))([x(t)-w_i(t)]+[w_i(t)-w_j(t)] \times \left(\frac{d_{ij}-\lambda\Delta_{ij}}{\lambda\Delta_{ij}+I_{ij}}\right)\right)\right] \\ &= w_j(t) + \varepsilon(t)p_j(x(t)) \times\left[\sum_{i=1}^{N} p_i(x(t))([x(t)-w_i(t)]+[w_i(t)-w_j(t)] \times \left(\frac{d_{ij}-\lambda\Delta_{ij}}{\lambda\Delta_{ij}+I_{ij}}\right)\right)\right] \end{aligned} \quad (38)$$

where $\sum_{i=1}^{N} p_i(x(t)) = 1$. Note that if $\gamma = 1$ in (25) and d_{ij} in (38) is taken as the square distance, i.e., $d_{ij} = \|w_i - w_j\|^2$, (25) and (38) are equivalent.

D. The PRSOM Algorithm

The architecture of PRSOM is the same as SOM or ViSOM. By using the same notation of SOM in Section II-B, the sequential PRSOM algorithm is described as follows:

```
Step 1) Randomly select an input x(t)
from a data set.
Step 2) Compute the assignment prob-
ability of x(t) for all neurons ac-
cording to (29).
Step 3) Perform the weight-updating
rule for all neurons according to
(26).
Step 4) Terminate the algorithm until
certain criterion is satisfied. Oth-
erwise, go to Step 1).
```

The above sequential algorithm is affected by the ordering of training samples. To avoid this problem, it is better to use the following batch algorithm of PRSOM.

```
Step 1) Compute the assignment proba-
bility of x(t) for all input data and
neurons according to (29).
Step 2) Perform the batch weight-up-
dating rule for all neurons
```

$$w_j(k+1) = w_j(k) + \frac{\varepsilon(t)}{N}\sum_{t=1}^{M} p_j\left(x(t)\right)\left(\sum_{i=1}^{N} p_i(x(t) \times \left([x(t)-w_i(k)]+\gamma\left[w_i(k)-w_j(k)\right]\left(\frac{d_{ij}^2-\lambda\Delta_{ij}^2}{\lambda\Delta_{ij}^2+I_{ij}}\right)\right)\right)$$

$$\begin{aligned} w_j(M) &= \frac{\sum_{t=1}^{M} x(t)\sum_{i=1}^{N} h_{ij}P_i\left(x(t)\right)}{\sum_{t=1}^{M}\sum_{i=1}^{N} h_{ij}P_i\left(x(t)\right)} = \frac{\sum_{t=1}^{M-1} x(t)\sum_{i=1}^{N} h_{ij}P_i\left(x(t)\right) + x(M)\sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)}{B_j(M)} \\ &= \frac{w_j(M-1)B_j(M-1) + x(M)\sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)}{B_j(M)} \\ &= \frac{w_j(M-1)\left(B_j(M-1) + \sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)\right) + \left(x(M)-w_j(M-1)\right)\sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)}{B_j(M)} \\ &= \frac{w_j(M-1)B_j(M) + \left(x(M)-w_j(M-1)\right)\sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)}{B_j(M)} \\ &= w_j(M-1) + \frac{1}{B_j(M)}\left(\sum_{i=1}^{N} h_{ij}P_i\left(x(M)\right)\right)\left(x(M)-w_j(M-1)\right) \\ &= w_j(M-1) + \varepsilon_j(M)p_j\left(x(M)\right)\left(x(M)-w_j(M-1)\right), \quad j=1,2,\ldots,N, \\ \text{i.e., } w_j(t) &= w_j(\mathrm{t}-1) + \varepsilon_j(t)p_j\left(x(t)\right)\left(x(t)-w_j(t-1)\right), \quad j=1,2,\ldots,N \end{aligned} \quad (34)$$

where k is current epoch and k+1 is the next epoch.
Step 3) Terminate the algorithm until certain criterion is satisfied. Otherwise, go to Step 1).

The computational complexity of PRSOM and ViSOM is $O(MN^2)$, where M and N are the number of input data and neurons, respectively. If N^2 is significantly less than M, the computational complexity of PRSOM is less than that of MDS, i.e., $O(M^2)$. The computational complexity of SOM, i.e., $O(MN)$), and that of CCA, i.e., $O(M)$, are less than that of PRSOM. The computational complexities of these mapping methods are listed in Table I.

$\varepsilon(t)$ in PRSOM should be decreased from high values to nearly zero like SOM (or ViSOM). The selection of regularization coefficient γ can be set from 0.5 to 10 practically according to the emphasis of the MDS term (second term) in (24). h_{ij} in PRSOM can be selected like (7) with the constraint $\sum_{j=1}^{N} h_{ij}$=1

$$h_{ij} = \frac{\exp\left(-\frac{\|\text{Pos}_i - \text{Pos}_j\|^2}{2\sigma^2}\right)}{\sum_{k=1}^{N} \exp\left(-\frac{\|\text{Pos}_i - \text{Pos}_k\|^2}{2\sigma^2}\right)} \tag{39}$$

where the neighborhood radius σ is a constant. The value of σ is important for the training of PRSOM. The neighborhood function curves are steep when the value of σ is small. As a result, only few neurons around neuron j can be included in the computation of the weight of the neuron j. This may have an effect of generating folded or disordered maps. On the other hand, the area of the neighborhood function is enlarged to neurons that are far from the neuron j when σ is set to a large value. Large σ flattens the neighborhood function curves and results in contracted maps. This would degrade the performance of competitive learning. In this paper, σ is set to 0.5, which results in maps with good mapping effects.

The neighborhood function in PRSOM is $p_j(x(t)) = \sum_{i=1}^{N} P_i(x(t))h_{ij}(t)$. σ in (39) can be set to a small value, e.g., 0.5, such that $p_j(x(t))$ affects not only the winner neuron due to the leaked information from other neighboring neurons.

The most important property of PRSOM is the cost function in (24), which gives the meaning of the weight-updating rule. From the definition of the cost function, the probabilistic quantization error F_{vq} in (31) is different from that of STVQ in (9). Optimization of only the first term F_{vq} will not generate the similar result with SOM. This should be also true for ViSOM if the regularized term in the updating rule is left out. The implication of F_{vq} is to not only minimize the probabilistic quantization error but also repulse neurons from one another. The meaning of the regularized term F_{reg} is similar to MDS. But PRSOM tries to preserve the interneuron distance in input space, which is a reverse direction compared with MDS.

The resolution parameter λ must be chosen carefully. If λ is too large, some useful data structure may not be well displayed on the output map. Some neurons far outside input data may be wasteful for visualization. If λ is too small, the resultant map is embedded in input data and cannot well display input data. A practical equation for the selection of λ has been proposed in [24]:

$$\lambda = 1 \sim 1.5 \times \frac{\text{Span}_{\max}}{\min\{a,b\}}, \text{ or } \lambda = 1 \sim 1.5 \times \frac{4 \times \sqrt{\text{Var}_{\max}}}{\min\{a,b\}} \tag{40}$$

where a and b are the number of rows and columns of the map, respectively. However, the selection of λ for PRSOM may be out of the range according to (40) because of high input dimension or nonlinearity.

The soft assignment in PRSOM can be exploited like that in STVQ. The accumulated probability in each neuron forms an accumulated probability matrix (AP matrix) like U-matrix. The element AP_{ij} of neuron k located at the ith row and jth column of the map is defined by

$$AP_{ij} = \sum_{t=1}^{M} P_k\left(x(t)\right). \tag{41}$$

By assigning different colors to different accumulated probabilities, we can obtain a colored map with some colors corresponding to clusters and some colors corresponding to empty regions. This is a powerful visualization technique in addition to the method by simply assigning input data to their nearest neurons [18].

SOM and ViSOM are both discrete approximations of principal surfaces. But SOM cannot well display the data boundary at the boundaries of output map since it is a density-based quantizer. ViSOM instead can well represent the data boundary because ViSOM is a uniform quantizer and some neurons are outside input data if parameters of ViSOM are properly chosen. PRSOM is also a discrete approximation of principal surface like ViSOM. As the interneuron distances in input space are regularized to resemble those in output grid, the regularized MDS term (second term) in (25) can be very small or neglectable after the completion of training. We further consider only the nearest neuron c by using hard assignment. The updating rule in (25) now becomes

$$w_j(t+1) = w_j(t) + \varepsilon(t)h_{cj}\left(x(t) - w_c(t)\right). \tag{42}$$

Then the adaptation rule in the final stage leads to the smoothing process

$$\text{PRSOM } w_c = \frac{\sum_{t=1}^{L} x(t)h_{cj}}{\sum_{t=1}^{L} h_{cj}} \tag{43}$$

which is similar to kernel smoothing of principal curves in (17), batch SOM's kernel smoothing function in (18), and ViSOM's kernel smoothing function in (19). Here h_{cj} is fixed for all time in PRSOM, which is different from (18) and (19) for SOM and ViSOM, respectively.

E. Quality Measurement of Mapping

In order to compare the mapping effects of different mapping methods, quality measurement of mapping is proposed. The measurement is evaluated by judging if the distances between a data point and its neighboring data points in input space are proportional to those in output space. For example, for any data point x in the input space, compute the distances between x and its nearest k neighboring data points (except the data points identical to x) in input space. Then compute the corresponding distances in output space. After all the distances between data points and their neighboring ones in input and output space are computed, the ratios of the distances in input space to the corresponding distances in output space are computed. Then the mean (μ) and standard deviation (σ) of the ratios are obtained. Finally, the relative standard deviation (RSD) is computed by RSD$= \sigma/\mu$. For ideal mapping, all the ratios are equal such that RSD must be zero. In the real world, the closer to zero the RSD, the better the mapping effect.

In this paper, the value of k in RSD is chosen as four. For CCA and Sammon's mapping, RSD is computed by input data and their projected data. For SOM, ViSOM, and PRSOM, RSD is computed by the trained weights of neurons in input space and the 2-D coordinates of neurons on the output map.

IV. Experimental Results

The advantages of the proposed PRSOM are demonstrated through two synthetic and three real data sets, i.e., wine data set, UK university data set, and Wisconsin breast cancer data set. The batch type of PRSOM algorithm is used in this paper. We will present the visualization effects of PRSOM compared with those of SOM, ViSOM, CCA, and Sammon's mapping.

A. 2-D Synthetic Data Set

The first data set was used for PRSOM to demonstrate the mapping effects of different values of λ and the difference among PRSOM, SOM, and ViSOM. It is a 2-D synthetic data set consisting of three mixtures of Gaussians. The number of data in each Gaussian is 100. Their mean vectors are [5.0 -5.0]$^{\mathrm{T}}$, [-5.0 5.0]$^{\mathrm{T}}$, and [0 2.0]$^{\mathrm{T}}$. Their covariance matrices are $\begin{bmatrix} 5.0 & -1.0 \\ -1.0 & 0.3 \end{bmatrix}$, $\begin{bmatrix} 0.2 & 1.2 \\ 1.2 & 3.0 \end{bmatrix}$, and $\begin{bmatrix} 2.0 & 0.2 \\ 0.2 & 3.0 \end{bmatrix}$. The three Gaussians are well separated in the 2-D input space. The number of epochs is 1000. The learning rate monotonically decreases from 0.90 to 0.01. The regularized parameter γ is set to 3.0. The size of the PRSOM map is set to 20×20. The neighborhood σ is set to 0.5 since the leaked information from other neighboring neurons causes neighborhood function $p_i(x(t))$ to affect not only the winner neuron but also its neighboring neurons. The resolution parameter λ can be set to 0.85$\sim$1.27 or 0.87$\sim$1.30 according to (40). In order to see the mapping effects of different values of λ, we plotted the final maps in the 2-D input space with $\lambda = 0.5, 0.9$, and 1.5 in Fig. 1(a)–(c), respectively. In Fig. 1(a), some input data are not covered by the map, quantization error is large, and resolution is high. In Fig. 1(c) the resolution is a little coarse and some boundary neurons are useless to represent the input data. As shown in Fig. 1(b), $\lambda = 0.9$ is appropriate for the 20×20 map and is inside the range according to (40).

The visualization of PRSOM can be used not only by assigning data to the nearest neurons but also by the AP matrix in (41) in Section III. Corresponding to Fig. 1(b), the visualizations by the assignment method and AP matrix method are shown in Fig. 2(a) and (b), respectively. Clearly the three clusters are clearly separated in the output maps in Fig. 2(a) and (b). In the AP matrix method, the larger the accumulated probabilities, the darker the corresponding neurons on the 2-D output map. Hence the clusters, noises, and outliers can be found by the AP matrix methods, while the assignment method is worse to deal with them.

We also used SOM on the 2-D data. The map size of SOM is also 20×20 and the learning rate decreases from 0.90 to 0.01 with time. The total epochs of SOM are 1000. The final map in the input space is shown in Fig. 1(d), where most neurons concentrate inside the three Gaussians. The corresponding output map with assignment visualization is illustrated in Fig. 2(c). Although the three Gaussians are clearly separated from each another on the output map, some of the data boundaries are clipped at the outside boundaries of the output map since the neurons of SOM cannot extend outside the whole data structure. The quality of mapping effects of PRSOM (map size 20×20, $\lambda = 0.9$) is better than that of SOM as listed in Table I.

We also used ViSOM (map size 20×20, $\lambda = 0.9$) on the 2-D data. As shown in Fig. 1(e), the final map in the input space is similar to that by PRSOM (map size 20×20, $\lambda = 0.9$). The three Gaussians are well separated and not clipped in the 2-D output space, as shown in Fig. 2(d).

B. 3-D Synthetic Data Set

The three-dimensional (3-D) synthetic data set consists of three mixed Gaussians with 100 points in each Gaussian. The mean vectors of the three Gaussians are [5.0 7.0 6.0]$^{\mathrm{T}}$, [-2.0 5.0 -3.0]^{T}, and [-10.0 6.0 2.0]^{T}. Their corresponding covariance matrices are

$$\begin{bmatrix} 5.0 & -1.0 & 0.3 \\ -1.0 & 0.3 & -1.0 \\ 0.3 & 1.5 & 4.0 \end{bmatrix}, \quad \begin{bmatrix} 1.0 & 0.1 & 0.5 \\ -2.0 & 5.0 & 1.0 \\ 1.3 & -0.2 & 3.0 \end{bmatrix}, \quad \begin{bmatrix} 0.1 & -1.0 & -0.2 \\ 1.2 & 2.3 & 1.4 \\ -0.3 & 1.2 & 4.0 \end{bmatrix}.$$

The three Gaussians are well separated in the 3-D input space. The total epochs are set to 1000. The learning rate monotonically decreases from 0.90 to 0.01. The regularized parameter γ is set to 2.0. The size of the PRSOM map is set to 20×20. The neighborhood σ is set to 0.5 like the first data set. The resolution parameter λ can be set to 1.17$\sim$1.75 or 1.26$\sim$1.88 according to (40). We tried three different values of λ in PRSOM: $\lambda = 1.0, 6.0$, and 12.0. The corresponding final maps in the 3-D input space are shown in Fig. 3(a)–(c). However, the visualization effect is best with $\lambda = 6.0$, which is outside the range according to (40). The visualization by the AP matrix is shown in Fig. 4(a), where the three clusters are easy to find.

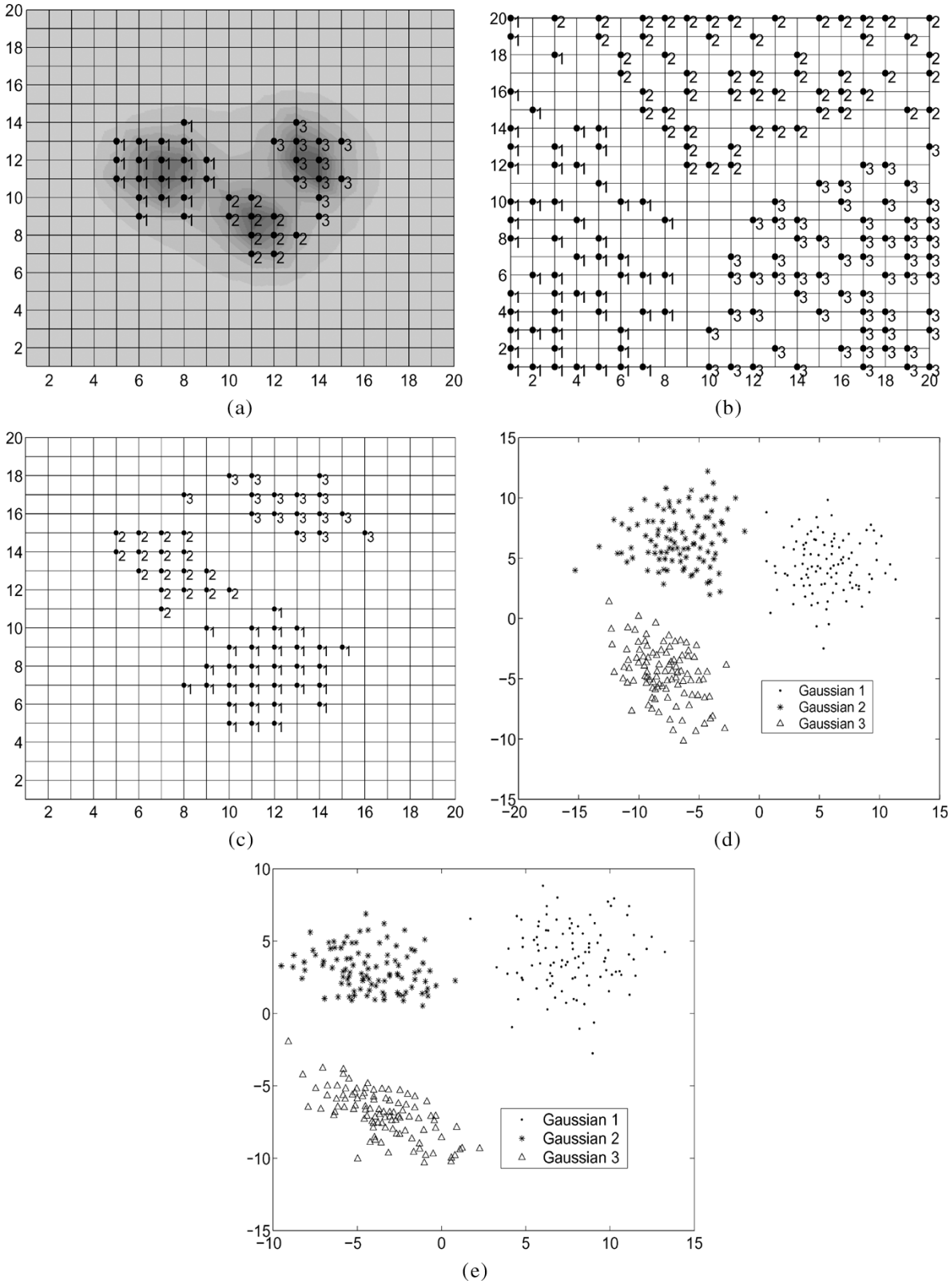


Fig. 4. Visualization of the 3-D synthetic data set. Activated neurons are plotted with big dots and corresponding number of Gaussians. (a) AP matrix method of PRSOM (map size 20×20, $\lambda = 6.0$). (b) The assignment method of SOM (map size 20×20). (c) The assignment method of ViSOM (map size 20×20, $\lambda = 1.7$). (d) Nonlinear mapping by CCA. (e) Nonlinear mapping by Sammon's mapping.

We used SOM with map size 20×20 on the 3-D data. The learning rate decreases from 1.0 to 0.01 with time and the total epochs of SOM are 1000. The final map in the input space is shown in Fig. 3(d), where most neurons concentrate inside the three Gaussians like the 2-D case. The corresponding output map with assignment visualization is illustrated in

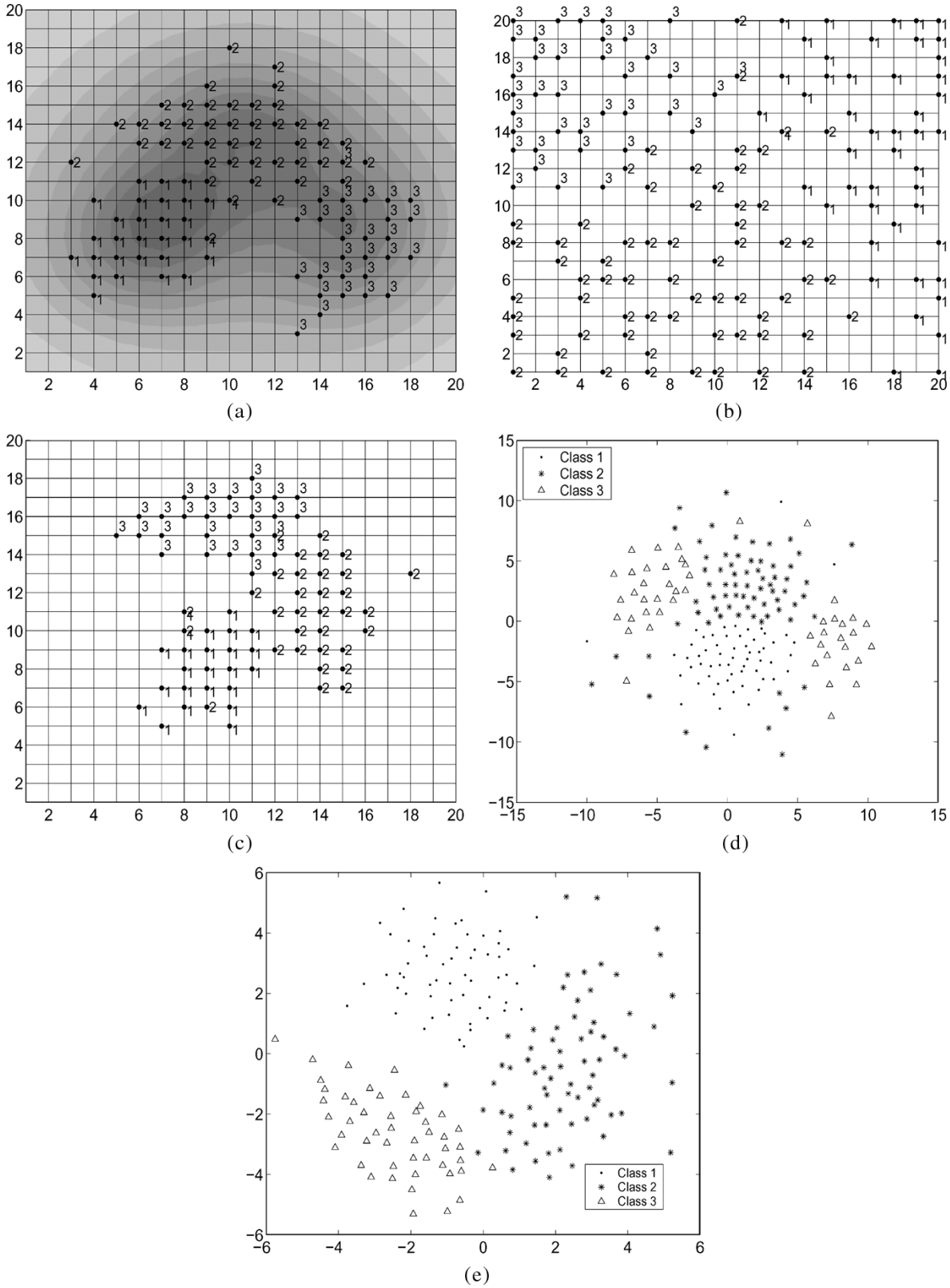


Fig. 5. Visualization of the wine data set. Activated neurons are plotted with big dots and corresponding number of classes. (a) AP matrix method of PRSOM (map size 20×20, $\lambda = 0.3$). (b) The assignment method of SOM (map size 20×20). (c) The assignment method of ViSOM (map size 20×20, $\lambda = 0.8$). (d) Nonlinear mapping by CCA. (e) Nonlinear mapping by Sammon's mapping.

Fig. 4(b). Although the three Gaussians are clearly separated from each another, the Gaussian data structures can not be well displayed in it because the outside data boundaries are clipped in SOM.

For ViSOM (map size 20×20, $\lambda = 1.7$), the final map is similar to that by PRSOM (map size 20×20, $\lambda = 6.0$). The maps in the input and output space are shown in Figs. 3(e) and 4(c), respectively.

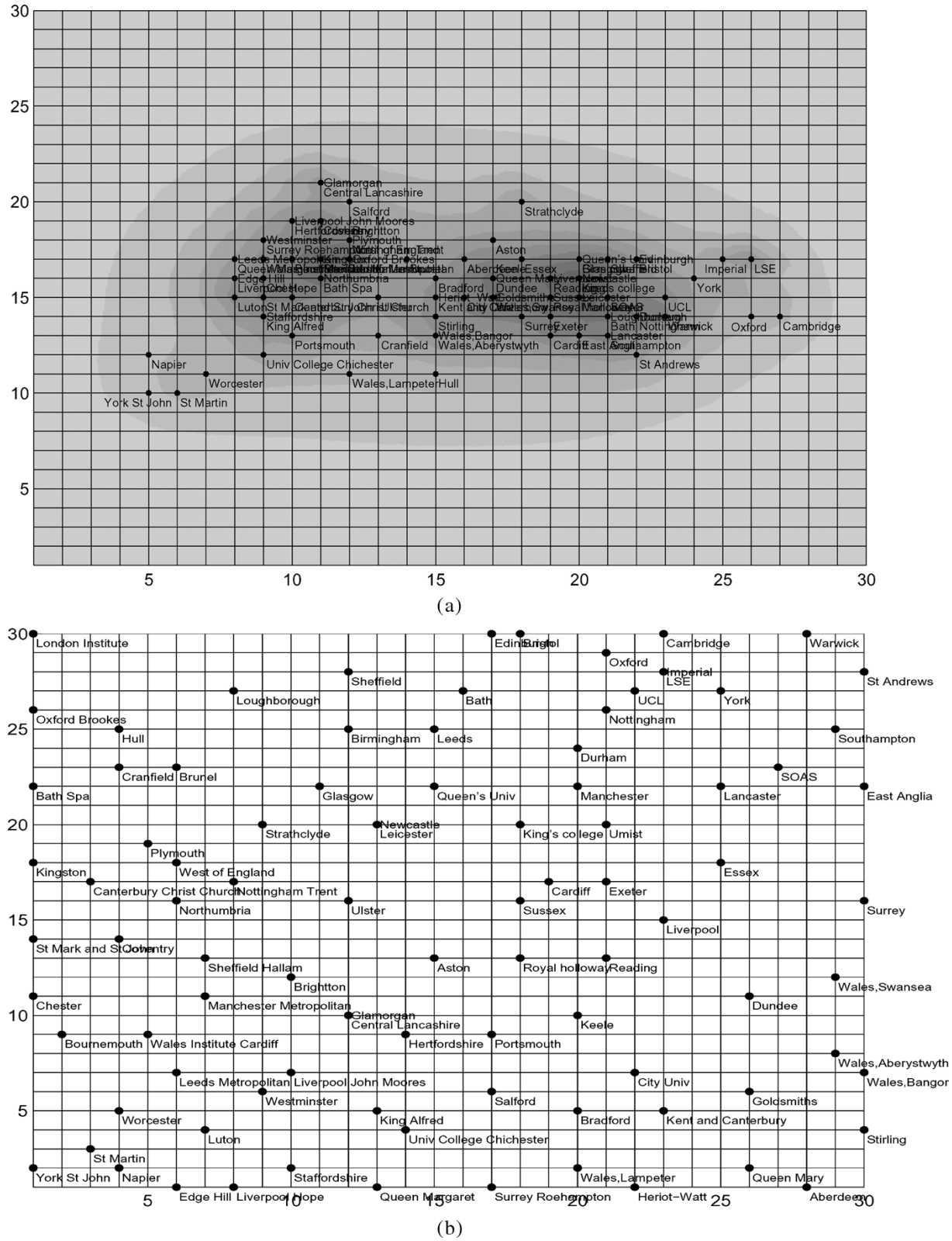
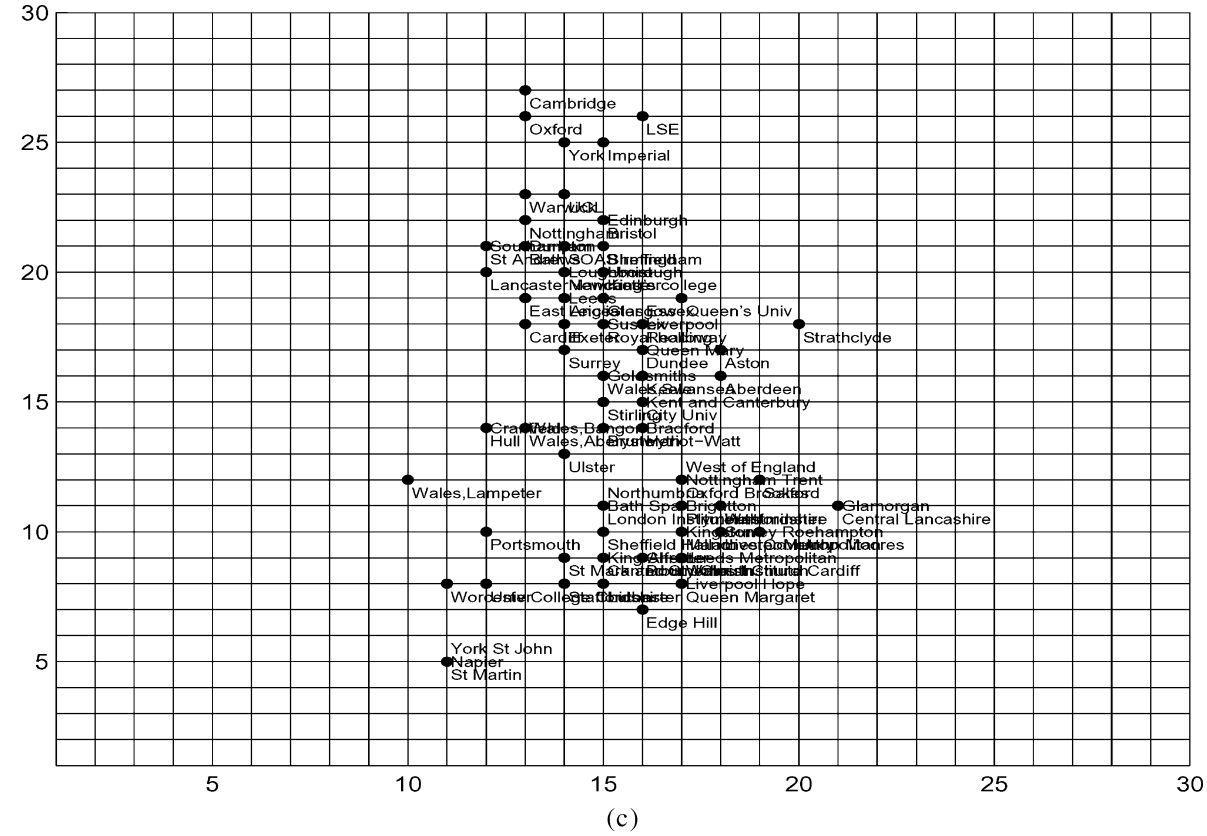


Fig. 6. Visualization of the U.K. university data set. Activated neurons are plotted with big dots and corresponding U.K. universities. (a) AP matrix method of PRSOM (map size 30×30, $\lambda = 200.0$). (b) The assignment method of SOM (map size 30×30).

For CCA and Sammon's mapping, the three Gaussians are well separated and not clipped in the reduced 2-D space as shown in Fig. 4(d) and (e), respectively.

The effects of different mapping methods are compared by RSD as listed in Table I. PRSOM (map size 20×20, $\lambda = 6.0$) has the best quality of mapping effects than other methods since the measurement of mapping by PRSOM is 0.07 that is closest to zero. The measurement by ViSOM is a little less than that by PRSOM. The measurements by SOM, CCA, and Sammon's mapping are much larger than that by PRSOM.



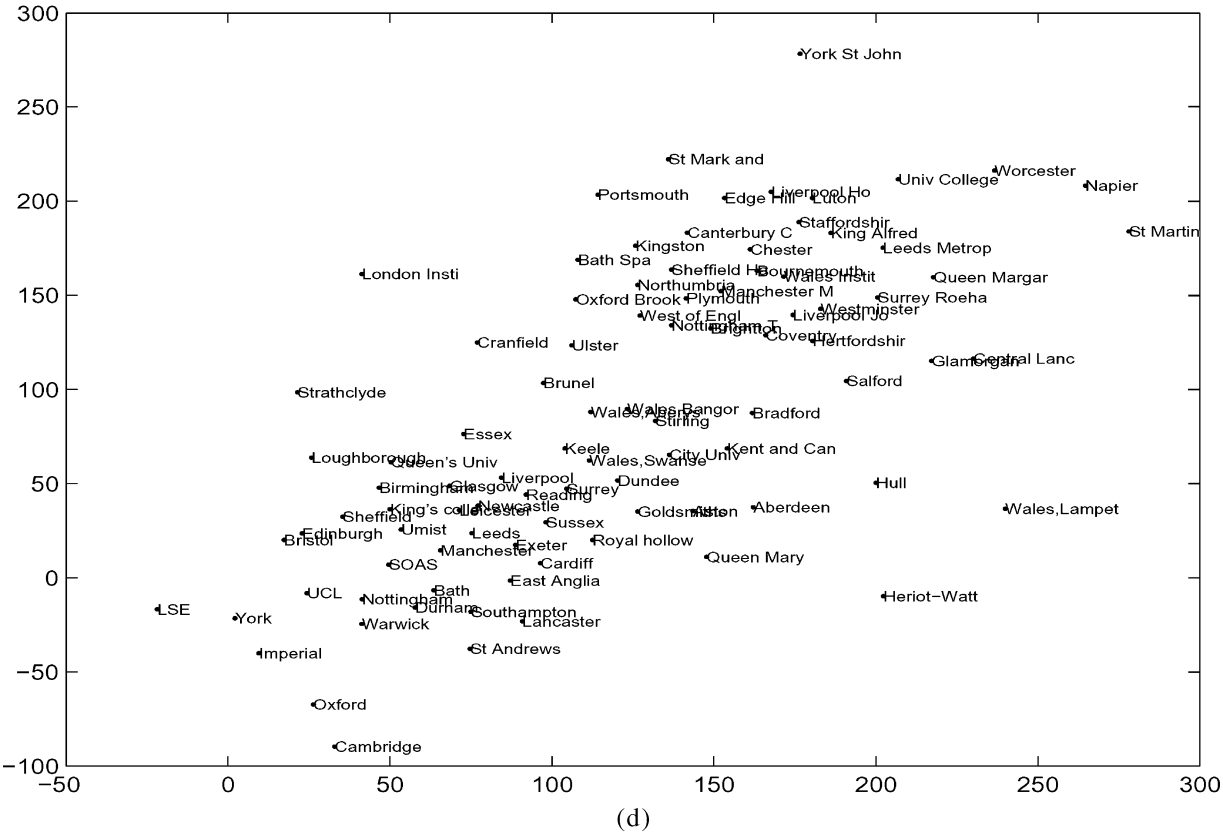
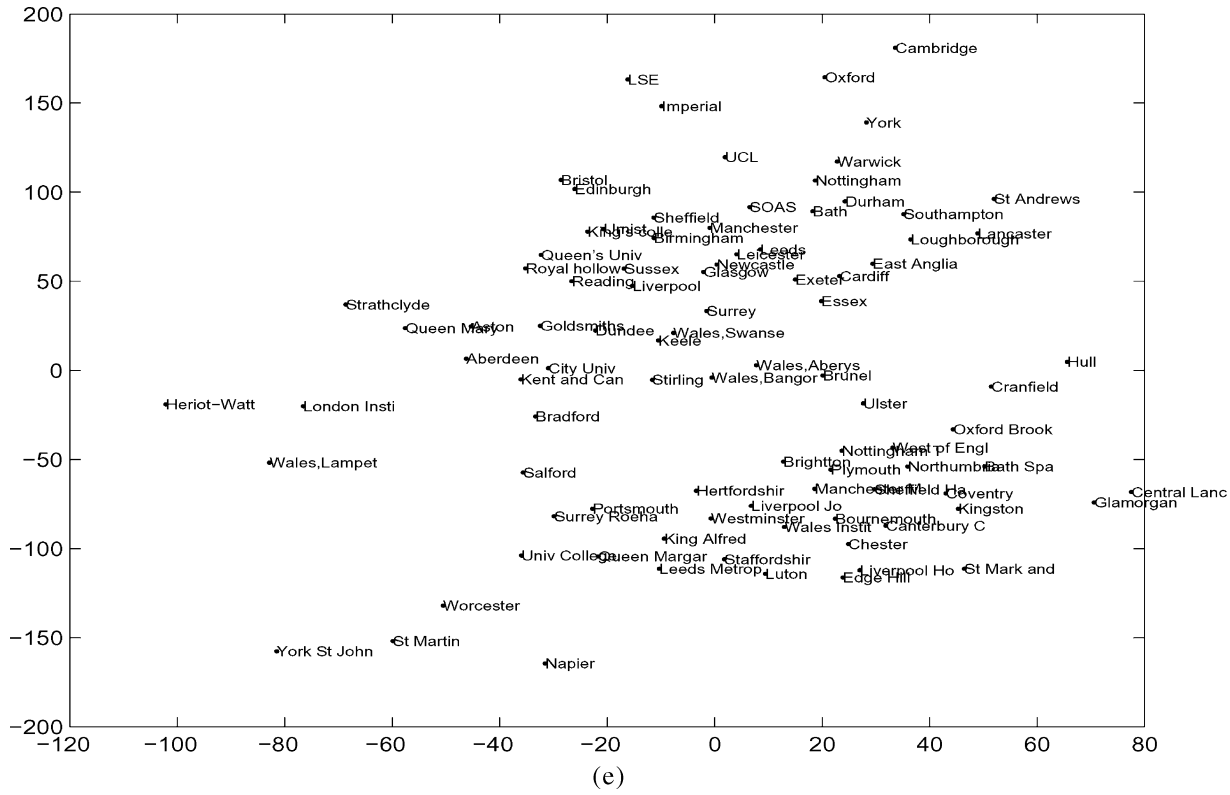


Fig. 6. (*Continued.*) Visualization of the U.K. university data set. Activated neurons are plotted with big dots and corresponding U.K. universities. (c) The assignment method of ViSOM (map size 30×30, $\lambda = 15.0$). (d) Nonlinear mapping by CCA.



(e)

Fig. 6. (*Continued.*) Visualization of the U.K. university data set. Activated neurons are plotted with big dots and corresponding U.K. universities. (e) Nonlinear mapping by Sammon's mapping.

C. Wine Data Set

The wine data set [34] consists of 178 data points with 13 dimensions. The data are divided into three classes. The numbers of data points in class 1, 2, and 3 are 59, 71 and 48, respectively. The three classes are not well separated. Since there is a large difference in different dimensions, the data were normalized such that the mean and variance of data in each dimension are zero and unit, respectively. The total epochs are set to 1000. The learning rate monotonically decreases from 0.90 to 0.01. The regularized parameter γ is set to 5.0. The size of the PRSOM map is set to 20×20. The neighborhood σ is set to 0.5 like the first and second data sets. The resolution parameter λ can be set to 0.34∼0.51 or 0.20∼0.30 according to (40). We found $\lambda = 0.3$ is an appropriate resolution parameter and is in the range according to (40). The visualization by the AP matrix method is shown in Fig. 5(a). There is only one dark area meaning the three classes are mixed to some extent. Class 1 and class 3 are well separated. But class 2 has some overlapping with the other two classes.

The data set was also trained by SOM with map size 20×20. The visualization of SOM on the wine data is shown in Fig. 5(b). The three classes are also not well separated and the outside data boundaries are also clipped. ViSOM (map size 20×20, $\lambda = 0.8$) and Sammon's mapping have similar visualization, as shown in Fig. 5(c) and (e), respectively. CCA has the worst visualization since each class is not well clustered, as shown in Fig. 5(d).

The mapping measurement of different methods is listed in Table I. PRSOM (map size 20×20, $\lambda = 0.3$) has the best quality of mapping effects since the measurement of mapping by PRSOM is 0.03, which is closest to zero. The measurement by ViSOM is a little less than that by PRSOM. The measurements by SOM, CCA, and Sammon's mapping are larger than that by PRSOM.

D. U.K. University Data Set

The U.K. university data set was taken from the *Sunday Times* newspaper. The newspaper ranks the U.K. universities every year from seven aspects. We chose the ranking on September 15, 2002. Ninety-three higher educational institutions with seven attributes, e.g., teaching quality, research achievement, employment rate, dropout rate, etc., were in the ranking list. Among these institutions, there are two types. Those in one type were founded before 1992. The other type was converted from polytechnics to fully accredited universities after 1992. The two types of universities are separated from each other to some extent. The total epochs are set to 1000. The learning rate monotonically decreases from 0.90 to 0.01. The regularized parameter γ is set to 1.0. The size of the PRSOM map is set to 30×30. The neighborhood σ is set to 0.5 like the first to third data set. The resolution parameter λ can be set to 11.80∼17.70 or 10.56∼15.84 according to (40). We found $\lambda = 200.0$ is an appropriate resolution parameter and is far outside the range according to (40).

The visualization by the AP matrix method is shown in Fig. 6(a). Clearly there are two clusters in the 2-D output map. Most post-1992 or new universities are in the left cluster. The pre-1992 or old universities are in the right clusters. Note that the first four universities, i.e., Cambridge University, Oxford

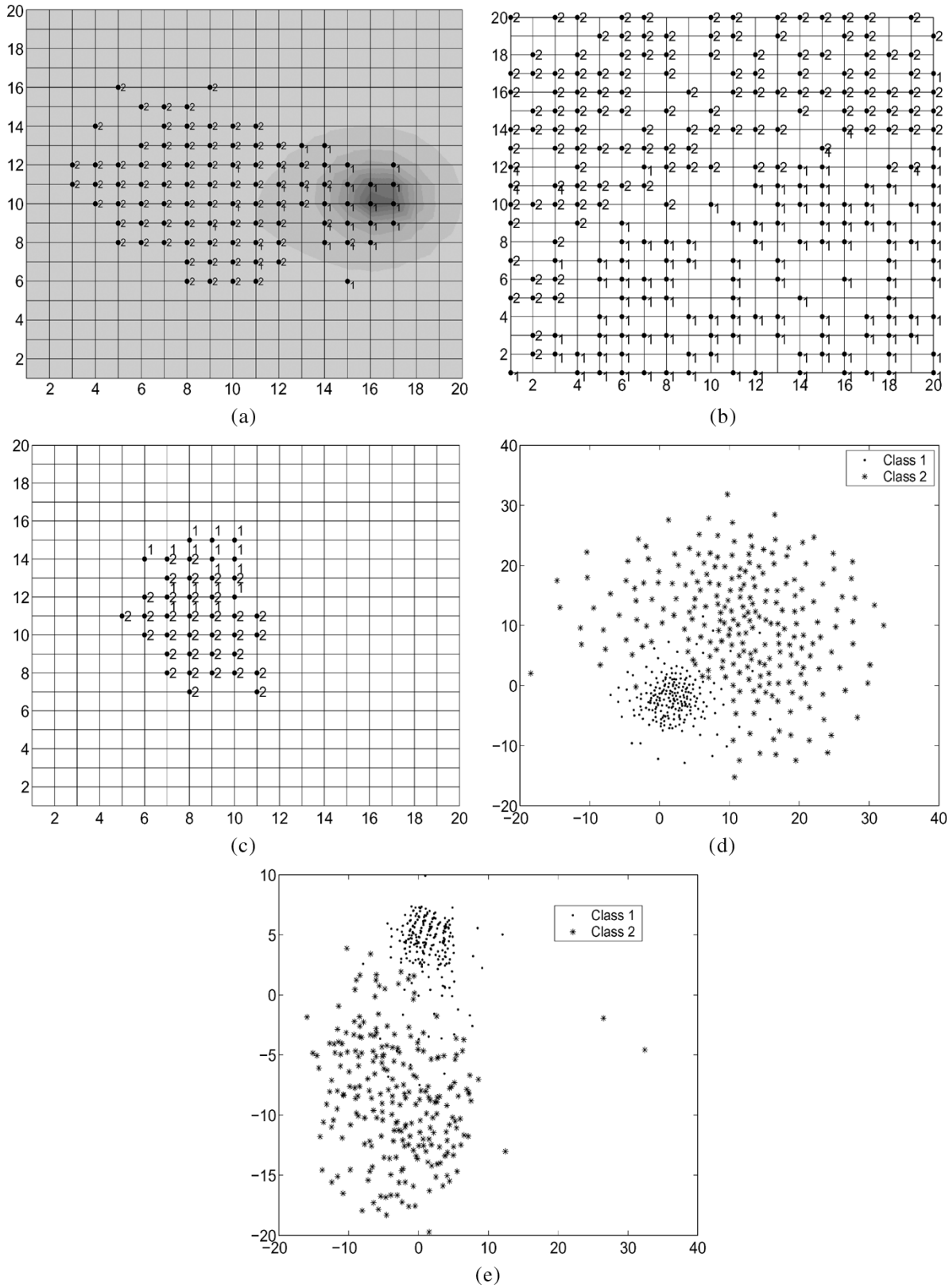


Fig. 7. Visualization of the Wisconsin breast cancer data set. Activated neurons are plotted with big dots and corresponding number of classes (class 1: benign, class2: malignant). (a) AP matrix method of PRSOM (map size 20×20, $\lambda = 3.0$). (b) The assignment method of SOM (map size 20×20). (c) The assignment method of ViSOM (map size 20×20, $\lambda = 3.0$). (d) Nonlinear mapping by CCA. (e) Nonlinear mapping by Sammon's mapping.

University, London School of Economics and Political Science, and Imperial College, are at the right corner of the map, meaning their corresponding high ranks in the ranking list. The lowest two universities, Central Lancashire and Glamorgan, lie at the top left corner of the map. The data set was also trained by SOM with map size 30×30. Like the visualization of the

previous three data sets by SOM, the outside data boundaries are also clipped in SOM, as shown in Fig. 6(b). ViSOM (map size 30×30, $\lambda = 15.0$), CCA, and Sammon's mapping have similar mapping, as shown in Fig. 6(c)–(e), respectively.

The mapping measurement of different method is listed in Table I. PRSOM (map size 30×30, $\lambda = 200.0$) has the best quality of mapping effects among the methods since the measurement of mapping by PRSOM is 0.06, which is closest to zero. The measurement by ViSOM is a little less than that by PRSOM. The measurements by SOM, CCA, and Sammon's mapping are larger than that by PRSOM.

E. Wisconsin Breast Cancer Data

The Wisconsin breast cancer data set [34], [35] consists of 599 instances with nine attributes, e.g., clump thickness, uniformity of cell size, marginal adhesion, etc. The data are divided into two classes: benign and malignant instances. But there are 16 instances that contain a single missing attribute value. The 16 instances were deleted for convenient processing. Thus the total numbers of instances used in this paper are 583. The numbers of benign and malignant instances are 444 and 239, respectively. There is no clear gap between the two classes. The total epochs are set to 1000. The learning rate monotonically decreases from 0.90 to 0.01. The regularized parameter γ is set to 3.0. The size of the PRSOM map is set to 20×20. The neighborhood σ is set to 0.5 like the first to fourth data set. The resolution parameter λ can be set to $0.45 \sim 0.68$ or $0.73 \sim 1.09$ according to (40). We found $\lambda = 3.0$ is an appropriate resolution parameter and is outside the range according to (40).

The visualization by the AP matrix method is shown in Fig. 7(a). There is only one dark area, where most of the data in the benign class concentrate, meaning the two classes are not well separated. The data set was trained by SOM with map size 20×20. The visualization of SOM on the Wisconsin breast cancer data is similar to that on the third and fourth data set, where the outside data boundaries are also clipped. ViSOM (map size 20×20, $\lambda = 3.0$), CCA, and Sammon's mapping have similar visualization, as shown in Fig. 7(c)–(e), respectively.

The mapping measurement of different methods is listed in Table I. PRSOM (map size 20×20, $\lambda = 3.0$) has the best quality of mapping effects since the measurement of mapping by PRSOM is 0.04, which is closest to zero. The measurement by ViSOM is a little less than that by PRSOM. The measurements by SOM, CCA, and Sammon's mapping are larger than that by PRSOM.

V. Conclusion

In this paper, a new visualization method, called PRSOM, is proposed. PRSOM hybridizes MDS and SOM in one algorithm. Therefore it reduces the computational burden by using SOM and preserves the interneuron distance after dimension reduction by using MDS. PRSOM is associated with a cost function such that its weight-updating rule is a principled optimization. PRSOM gives better mapping effects than SOM. Due to the probabilistic assignment of each input datum, the AP matrix method provides a better visualization tool than the conventional assignment method used in ViSOM. The regularization is to constrain the interneuron distances in input space resemble those in output space as much as possible. ViSOM can be considered as a simplification, hard assignment, and fast algorithm of PRSOM. Although a large amount of neurons is required and hence the computation is heavy, experiments demonstrate that PRSOM is an effective approach for dimension reduction and visualization compared with SOM, ViSOM, CCA, and Sammon's mapping.

Acknowledgment

The authors are grateful to the reviewers for their detailed and useful comments.

References

[1] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[2] R. N. Shepard and J. D. Carroll, "Parametric representation of nonlinear data structures," in *Proc. Int. Symp. Multivariate Anal.*, P. R. Krishnaiah, Ed. New York: Academic, 1965, pp. 561–592.

[3] E. Oja, "Neural networks, principal components, and applications of numerical analysis," *Int. J. Neural Syst.*, vol. 1, pp. 61–68, 1989.

[4] J. Rubner and P. Tavan, "A self-organizing network for principal component analysis," *Europhys. Lett.*, vol. 10, pp. 693–698, 1989.

[5] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural network," *AICHE J.*, vol. 37, pp. 233–243, 1991.

[6] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward network," *Neural Netw.*, vol. 2, pp. 459–473, 1991.

[7] T. Hastie and W. Stuetzle, "Principal curves," *J. Amer. Statist. Assoc.*, vol. 84, pp. 502–516, 1989.

[8] M. LeBlanc and R. J. Tibshirani, "Adaptive principal surfaces," *J. Amer. Statist. Assoc.*, vol. 89, pp. 53–64, 1994.

[9] J. Karhunen and J. Joutsensalo, "Generalization of principal component analysis, optimization problems, and neural networks," *Neural Netw.*, vol. 8, pp. 549–562, 1995.

[10] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, pp. 1299–1319, 1998.

[11] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-18, pp. 401–409, 1969.

[12] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 296–317, 1995.

[13] T. Kohonen, "Self-organized formation of topologically correct feature map," *Biol. Cybern.*, vol. 43, pp. 56–69, 1982.

[14] ——, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1997.

[15] H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps: An Introduction*. Reading, MA: Addison-Wesley, 1992.

[16] F. Mulier and V. Cherkassky, "Self-organization as an iterative kernel smoothing process," *Neural Comput.*, vol. 7, pp. 1165–1177, 1995.

[17] A. Ultsch and H. P. Siemon, "Kohonen's self organizing feature maps for exploratory data analysis," in *Proc. Int. Neural Networks Conf.*, Paris, France, 1990, pp. 305–308.

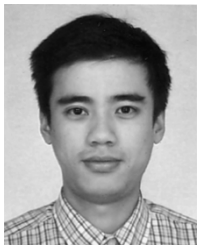
[18] X. Zhang and Y. Li, "Self-organizing map as a new method for clustering and data analysis," in *Proc. Int. Joint Conf. Neural Networks*, 1993, pp. 2448–2451.

[19] M. A. Kraaijveld, J. Mao, and A. K. Jain, "A nonlinear projection method based on the Kohonen's topology preserving maps," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 548–559, 1995.

[20] N. R. Pal and V. K. Eluri, "Two efficient connectionist schemes for structure preserving dimensionality reduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1142–1154, 1998.

[21] A. Köng, "Interactive visualization and analysis of hierarchical neural projections for data mining," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 615–624, 2000.

[22] M. C. Su and H. T. Chang, "A new model of self-organizing neural networks and its application in data projection," *IEEE Trans. Neural Netw.*, vol. 12, no. 1, pp. 153–158, 2000.
[23] H. Yin, "ViSOM: A novel method for multivariate data projection and structure visualization," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 237–243, 2002.
[24] ——, "Data visualization and manifold mapping using the ViSOM," *Neural Netw.*, vol. 15, no. 8–9, pp. 1005–1016, 2002.
[25] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002.
[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.
[27] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1–27, 1964.
[28] P. Demartines and J. Hérault, "Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 148–154, 1997.
[29] T. Graepel, M. Burger, and K. Obermayer, "Phase transitions in stochastic self-organizing maps," *Phys. Rev. E*, vol. 56, pp. 3876–3890, 1997.
[30] ——, "Self-organizing maps: generalizations and new optimization techniques," *Neurocomputing*, vol. 21, pp. 173–190, 1998.
[31] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.
[32] Y. Cheng, "Convergence and ordering of Kohonen's batch map," *Neural Comput.*, vol. 9, pp. 1667–1676, 1997.
[33] K. Kiviluoto and E. Oja, "S-map: a network with a simple self-organization algorithm for generative topographic mappings," in *Advances in Neural Information Processing Systems 10*, M. Jordan, M. Kearns, and S. Solla, Eds. Cambridge, MA: MIT Press, 1998, pp. 549–555.
[34] C. L. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases* : Dept. of Information and Computer Science, Univ. of California at Irvine, 1998.
[35] W. H. Wolberg and O. L. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology," *Proc. Nat. Acad. Sci.*, vol. 87, pp. 9193–9196, 1990.
[36] C.-H. Chang, P. Xu, R. Xiao, and T. Srikanthan, "New adaptive color quantization method based on self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 237–249, 2005.
[37] A. Hirose and T. Nagashima, "Predictive self-organizing map for vector quantization of migratory signals and its application to mobile communications," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1532–1540, 2003.
[38] G. A. Barreto and A. F. R. Araujo, "Identification and control of dynamical systems using the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1244–1259, 2004.
[39] W. Hu, D. Xie, and T. Tan, "A hierarchical self-organizing approach for learning the patterns of motion trajectories," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 135–144, 2004.
[40] S. Wu and T. W. S. Chow, "Induction machine fault detection: using SOM-based RBF neural networks," *IEEE Trans. Ind. Electron.*, vol. 51, no. 1, pp. 183–194, 2004.
[41] S. Wu, M. K. M. Rahman, and T. W. S. Chow, "Content-based image retrieval using growing hierarchical self-organizing quadtree map," *Pattern Recognition*, vol. 38, no. 5, pp. 707–722, 2005.



Sitao Wu received the B.E. and M.E. degrees from the Department of Electrical Engineering, Southwest Jiaotong University, Chengdu, China, in 1996 and 1999, respectively, and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, China, in 2004.

His research interests are in neural networks and pattern recognition and their applications.

Tommy W. S. Chow (M'94–SM'03) received the B.Sc. (first honors) degree and the Ph.D. degree from the University of Sunderland, U.K.

His doctoral work involved a collaborative project between The International Research and Development, Newcastle Upon Tyne, U.K., and the Ministry of Defense (Navy), U.K. He joined the City University of Hong Kong, where currently he is a Professor in the Department of Electronic Engineering. He has been working on different consultancy projects with the Mass Transit Railway, Kowloon-Canton Railway Corporation, Hong Kong. He has also conducted other collaborative projects with Kong Electric Co., Ltd., Royal Observatory Hong Kong, and MTR Hong Kong on the application of neural networks for machine fault detection and forecasting. His main research has been in the area of learning theory and optimizations, system identification, and machine fault diagnostics. He is author or coauthor of numerous published works, including book chapters and more than 100 journal articles related to his research. He was Chairman of Hong Kong Institute of Engineers, Control Automation and Instrumentation Division from 1997 to 1998.

Prof. Chow received the Best Paper Award at the 2002 IEEE Industrial Electronics Society Annual Meeting, Seville, Spain.