A Self-Organizing Map with Expanding Force for Data Clustering and Visualization

Wing-Ho Shum, Hui-Dong Jin, Kwong-Sak Leung
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong
{whshum, hdjin, ksleung}@cse.cuhk.edu.hk

Man-Leung Wong
Department of Information Systems
Lingnan University
Hong Kong
mlwong@ln.edu.hk

Abstract

The Self-Organizing Map (SOM) is a powerful tool in the exploratory phase of data mining. However, due to the dimensional conflict, the neighborhood preservation cannot always lead to perfect topology preservation. In this paper, we establish an Expanding SOM (ESOM) to detect and preserve better topology correspondence between the two spaces. Our experiment results demonstrate that the ESOM constructs better mappings than the classic SOM in terms of both the topological and the quantization errors. Furthermore, clustering results generated by the ESOM are more accurate than those by the SOM.

1 Introduction

The Self-Organizing Map (SOM) has been proven to be useful as visualization and data exploratory analysis tools [6]. It maps high-dimensional data items onto a low-dimensional grid of neurons. The regular grid can be used as a convenient visualization surface for showing different features of data [9, 12]. SOMs have been successfully applied in various areas such as full-text and image analysis, and travelling salesman problem [3, 4, 7].

However, because a SOM maps the data from a high-dimensional space to a low-dimensional space which is usually 2-dimensional, a dimensional conflict may occur and a perfect topology preserving mapping may not be generated [1, 5]. For example, consider the two trained SOMs depicted in Fig.1, although they preserve good neighborhood relationships, the SOM depicted in Fig.1(b) folds the neuron string onto data irregularly and loses much topology information in comparison with the SOM shown in Fig.1(a).

There are many research efforts to enhance SOMs for visualization and cluster analysis. Most of them focus on how to visualize neurons clearly and classify data [2, 10, 12].

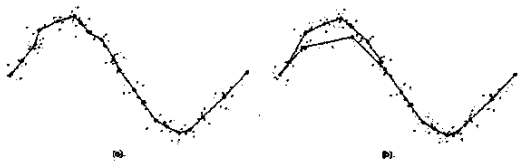

Figure 1. Two SOMs from 2-dimensional space to 1-dimension. The connected dots indicate a string of neurons, and other dots indicate data.

Some work has concentrated on better topology preservation. Kirk and Zurada [5] trained their SOM to minimize the quantization error in the first phase and then minimize the topological error in the second phase.

In this paper, we propose a new learning rule to enhance the topology preservation and overcome the irregularity problem. The paper is organized as follows. We introduce our ESOM in Section 2, followed by its theoretic analysis.. The visualization and clustering results of the ESOM are presented and compared with the SOM in Section 3. A conclusion is given in the last section.

2 Expanding SOM

Besides the neighborhood relationship in the SOM, another topology relationship can be detected and preserved during the learning process to achieve a better topology preserving mapping for data visualization. This is a linear ordering relationship based on the distance between data and their center. A neural network can detect and preserve this ordering relationship. If the distance between a data item and the center of all data items is larger, the distance between the corresponding output neuron and the center is

0-7695-1754-4/02 $17.00 © 2002 IEEE

also larger.

Our Expanding SOM (ESOM) can construct a mapping that preserves both the neighborhood and the ordering relationships. Since this mapping preserves more topology information of the input data, better performance in visualization can be achieved.

We introduce a new learning rule to learn the linear ordering relationship. Different from the SOM, the learning rule of the ESOM has an additional factor, the expanding coefficient $c_j(t)$, which is used to push neurons away from the center of all data items during the learning process. In other words, the flexible neuron network is expanding gradually in our ESOM algorithm. Moreover, the expanding force is specified according to the ordering of the data items. In general, the larger the distance between the corresponding data item and the center, the larger the expanding coefficient $c_j(t)$. Consequently, the associated output neuron is pushed away from the center and the ordering of data items is thus preserved in the output neurons. In the following sub-sections, the ESOM algorithm will be discussed first. Theoretical analysis of the ESOM algorithm will then be described.

2.1 The ESOM algorithm

The ESOM algorithm consists of 6 steps.

1. Linearly transform the coordinates $\vec{x}_i' = [x'_{1i}, x'_{2i}, \cdots, x'_{Di}]^T$ $(i = 1, \cdots, N)$ of all given data items so that they lie within a sphere S_R centered at the origin with radius R (< 1). Here N is the number of data items, D is the dimensionality of the data set. Hereafter, $[x_{1i}, x_{2i}, \cdots, x_{Di}]^T$ denotes the new coordinate of $\vec{x}_i$. Let the center of all data items be $\vec{x}_C' = \frac{1}{N}\sum_{i=1}^{N}\vec{x}_i'$ and the maximum distance of data from the data center be D_{max}, then

$$\vec{x}_i = \frac{R}{D_{max}}\left(\vec{x}_i' - \vec{x}_C'\right) \text{ for all } i. \qquad (1)$$

2. Set $t = 0$, and the initialize weight vectors $\vec{w}_j(0)$ $(j = 1, \cdots, M)$ with random values within the above sphere S_R where M is the number of output neurons.

3. Select a data item at random, say $\vec{x}_k(t) = [x_{1k}, x_{2k}, \cdots, x_{Dk}]^T$, and feed it to the input neurons.

4. Find the winning output neuron, say $m(t)$, nearest to $\vec{x}_k(t)$ according to the Euclidean metric:

$$m(t) = \arg\min_j \|\vec{x}_k(t) - \vec{w}_j(t)\|. \qquad (2)$$

5. Train neuron $m(t)$ and its neighbors by using the following formula:

$$\begin{aligned}\vec{w}_j(t+1) &= c_j(t)\vec{w}_j'(t+1) \triangleq c_j(t)\{\vec{w}_j(t) \\ &\quad +\alpha_j(t)\left[\vec{x}_k(t) - \vec{w}_j(t)\right]\} \qquad (3)\end{aligned}$$

The parameters include:

- *the interim neuron* $\vec{w}_j'(t+1)$, which indicates the position of the excited neuron $\vec{w}_j(t)$ after moving towards the input data item $\vec{x}_k(t)$;
- *the learning parameter* $\alpha_j(t)(\in [0,1])$, which is specified by a learning rate $\epsilon(t)$ and a neighborhood function $h_{j,m(t)}(\sigma(t))$:

$$\alpha_j(t) = \epsilon(t) \times h_{j,m(t)}(\sigma(t)); \qquad (4)$$

- *the expanding coefficient* $c_j(t)$, which is specified according to

$$c_j(t) = \left[1 - 2\alpha_j(t)\left(1 - \alpha_j(t)\right)\kappa_j(t)\right]^{-\frac{1}{2}}, \qquad (5)$$

where $\kappa_j(t)$ is specified by

$$\begin{aligned}\kappa_j(t) &= 1 - \langle\vec{x}_k(t), \vec{w}_j(t)\rangle - \\ &\quad \sqrt{(1 - \|\vec{x}_k(t)\|^2)(1 - \|\vec{w}_j(t)\|^2)} \qquad (6)\end{aligned}$$

6. Update the neighbor width parameter $\sigma(t)$ and the learning parameters $\epsilon(t)$ with predetermined decreasing schemes. If the learning loop does not reach a predetermined number, go to Step 3 with $t := t + 1$.

The first step facilitates the realization of the expanding coefficient $c_j(t)$. After the transformation, we can use the norm of a data item $\|\vec{x}_k(t)\|$ to represent its distance from the center of the transformed data items since the center is the origin. Thus, the norm $\|\vec{x}_k(t)\|$ can indicate the ordering topology in the data space. This ordering will be detected and preserved in $\|\vec{w}_j(t)\|$ through the expanding process.

The learning rule defined in Eq.(3) is the key point of the proposed ESOM algorithm. Different from the SOM learning rule, it has an additional multiplication factor — the expanding coefficient $c_j(t)$. It is worth pointing out that, although the expanding coefficient $c_j(t)$ is relevant to all data items, the calculation of $c_j(t)$ only depends on $\alpha_j(t)$, $\vec{x}_k(t)$ and $\vec{w}_j(t)$. If $c_j(t)$ is a constant 1.0, the ESOM is simplified to a conventional SOM. Since $c_j(t)$ is always greater than or equal to 1.0, the expanding force pushes the excited neuron away from the center. In other words, the inequality $\|\vec{w}_j(t+1)\| \geq \left\|\vec{w}_j'(t+1)\right\|$ is always true. Fig.2(b) illustrates the expanding functionality. After moving the excited neuron $\vec{w}_j(t)$ towards the input data $\vec{x}_k(t)$, as indicated by

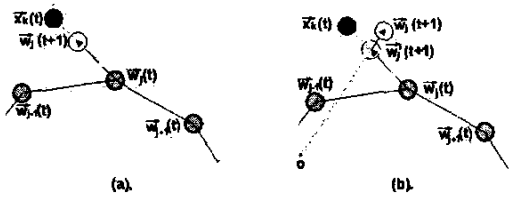

Figure 2. A schematic view of two different learning rules. (a). The learning rule for the SOM; (b). The learning rule for the ESOM. A black disc indicates a data item; a gray disc indicates a neuron; a solid line indicates the neighbor relationship on the grid; a circle indicates the new position of a neuron; a dashed circle indicates a neuron's temporary position; a dashed arrow indicates a movement direction; and 'o' indicates the the center of data.

$\vec{w}_j'(t+1)$, the neuron is then pushed away from the center. So, during the learning process, the flexible neuron net is expanding in the data space. More interestingly, as the expanding force is specified accordingly to the ordering relationships, distant data items are likely to be mapped to the distant neurons, while data items near the center are likely to be mapped to the neurons in the center of map, therefore, $c_j(t)$ can help us to detect and preserve the ordering relationship.

Researchers have introduced several measures to evaluate the quality of a mapping [1, 13]. In this paper, we employ both the quantization error E_Q and the topological error E_T used in [5, 11] to evaluate the mapping obtained by our ESOM. E_T is defined as the proportion of the data items whose closest and second-closest neurons are not adjacent on the grid. The quantization error evaluates how well the weight vectors represent the data set [5, 12]. It is specified as follows:

$$E_Q = \frac{1}{N}\sum_{k=1}^{N} \|\vec{x}_k(t) - \vec{w}_{m_k}(t)\| \tag{7}$$

where m_k is the winner for the data item $\vec{x}_k(t)$. These two criteria usually conflict with each other in the SOM.

2.2 Theoretical analysis

To show the feasibility of the ESOM, we should verify that the ESOM does generate a map that preserve both the good neighborhood relationship and the ordering relationship. In this paper, we only give a theorem on a one-step trend to support the feasibility of the ESOM because it is very difficult to prove the convergence of the SOM-like networks in high dimensional cases. In fact, it is still one of the long-standing open research problems in neural networks [8]. We will perform a more rigorous convergence analysis in our future research work. In the following theorem, we assume that all input data items are located within the sphere S_R and their center coincides with the origin because the preprocessing procedure in Step 1 has been executed.

Theorem 1 *Let S_R be the closed sphere with radius R (< 1) centered at the origin, $\{\vec{x}_k(t) \in S_R\}$ (for $k = 1, \cdots, N$) be the input data and $\{\vec{w}_j(t)\}$ (for $j = 1, \cdots, M$) be the weight vectors of the ESOM at time t. Then, for any $t \geq 0$,*

(i). for $j \in \{1, 2, \cdots, M\}$,

$$1 \leq c_j(t) \leq \frac{1}{\sqrt{1-R^2}}; \tag{8}$$

and $\vec{w}_j(t) \in S_R$, that is,

$$\|\vec{w}_j(t)\| \leq R. \tag{9}$$

(ii). the expanding coefficient $c_j(t)$ increases with $\|\vec{x}_k(t)\|$ when $\|\vec{x}_k(t)\| \geq \|\vec{w}_j(t)\|$.

Proof. (i). We prove Eqs.(8) and (9) together by induction. This is trivially true for $t = 0$ according to Step 2 of the ESOM algorithm. If we assume that both equations hold for certain $t(\geq 0)$, then we find

$$\begin{aligned}
1 - \kappa_j(t) &= \langle \vec{x}_k(t), \vec{w}_j(t) \rangle + \sqrt{(1 - \|\vec{x}_k(t)\|^2)} \\
&\quad \times \sqrt{(1 - \|\vec{w}_j(t)\|^2)} \\
&\leq \left(\sum_{d=1}^{D} x_{dk}^2(t) + \left(\sqrt{(1 - \|\vec{x}_k(t)\|^2)} \right)^2 \right) \\
&\quad \times \left(\sum_{d=1}^{D} w_{dj}^2(t) + \left(\sqrt{(1 - \|\vec{w}_j(t)\|^2)} \right)^2 \right) \\
&= 1.
\end{aligned}$$

Similarly,

$$\begin{aligned}
1 - \kappa_j(t) &= \langle \vec{x}_k(t), \vec{w}_j(t) \rangle + \sqrt{(1 - \|\vec{x}_k(t)\|^2)} \\
&\quad \times \sqrt{(1 - \|\vec{w}_j(t)\|^2)} \\
&\geq -\frac{1}{2}(\|\vec{x}_k(t)\|^2 + \|\vec{w}_j(t)\|^2) + \sqrt{(1-R^2)} \\
&\quad \times \sqrt{(1-R^2)} \\
&\geq 1 - 2R^2.
\end{aligned}$$

Thus,

$$0 \leq \kappa_j(t) \leq 2R^2$$

On the other hand, for any learning parameter $\alpha_j(t) \in [0,1]$, the following inequality is true,

$$0 \leq \alpha_j(t)\,(1-\alpha_j(t)) \leq 0.25.$$

According to Eq.(5), we get $1 \leq c_j(t) \leq \frac{1}{\sqrt{1-R^2}}$. According to the ESOM learning rule, we have

$$\begin{aligned} 1-\|\vec{w}_j(t+1)\|^2 &= \left[\begin{array}{c} [c_j(t)]^{-2} \\ -\|\vec{w}_j(t)+\alpha_j(t)(\vec{x}_k(t)-\vec{w}_j(t))\|^2 \end{array}\right] \\ &\quad \times (c_j(t))^2 \qquad (10) \\ &= \frac{1}{(c_j(t))^{-2}} \times \\ &\quad \left[\begin{array}{c} (1-\alpha_j(t))\sqrt{1-\|\vec{w}_j(t)\|^2} \\ +\alpha_j(t)\sqrt{1-\|\vec{x}_k(t)\|^2} \end{array}\right]^2 \\ &\geq \left[\begin{array}{c} (1-\alpha_j(t))\sqrt{1-R^2} \\ +\alpha_j(t)\sqrt{1-R^2} \end{array}\right]^2 \\ &= 1-R^2. \end{aligned}$$

This implies that $\|\vec{w}_j(t+1)\| \leq R$ for any $j = 1, \cdots, M$. Thus, by induction, $\vec{w}_j(t) \in S_R$ for any j and t.

(ii). We rewrite $\vec{x}_k(t)$ and $\vec{w}_j(t)$ as follows,

$$\vec{x}_k(t) = \rho \times \vec{e}_{x_k}\,, \quad \vec{w}_j(t) = r \times \vec{e}_{w_j}$$

Here $\vec{e}_{x_k}$ and $\vec{e}_{w_j}$ are two unit vectors, and $\rho = \|\vec{x}_k(t)\|$ and $r = \|\vec{w}_j(t)\|$. According to the assumption that $\rho \geq r$ holds. Let

$$\begin{aligned} F(\rho) &= \langle \vec{w}_j(t), \vec{x}_k(t) \rangle \\ &\quad + \sqrt{(1-\|\vec{w}_j(t)\|^2)(1-\|\vec{x}_k(t)\|^2)} \qquad (11) \\ &= \rho \cdot r \cdot \langle \vec{e}_{w_j}, \vec{e}_{x_k} \rangle + \sqrt{(1-\rho^2)(1-r^2)}. \end{aligned}$$

According to Eq.(5), it is obvious that $F(\rho) = 1 - \frac{1-c_j^{-2}(t)}{2\alpha_j(t)(1-\alpha_j(t))}$. $F(\rho)$ decreases with the expanding coefficient $c_j(t)$. So, to justify the increasing property of $c_j(t)$, it is sufficient to show that $F(\rho)$ decreases with ρ whenever $\rho \geq r$. A direct calculation shows

$$\begin{aligned} \frac{\partial F(\rho)}{\partial \rho} &= r \cdot \langle \vec{e}_{w_j}, \vec{e}_{x_k} \rangle - \frac{\rho}{\sqrt{1-\rho^2}}\sqrt{1-r^2} \qquad (12) \\ &\leq r - \rho \leq 0. \qquad (13) \end{aligned}$$

This implies the decreasing property of $F(\rho)$ on ρ when $\rho \geq r$. ■

Theorem 1 (i) says that the expanding coefficient $c_j(t)$ is always larger than or equal to 1.0. In other words, it always pushes neurons away from the origin. Thus, during learning, the neuron net is expanding. Furthermore, though the expanding force is always greater than or equal to 1.0, it will never push the output neurons to infinite locations. In fact, it is restricted by sphere S_R in which the data items are located. This point is substantiated by Eq.(9). This supports the feasibility of our ESOM.

Theorem 1 (ii) gives a theoretic support that the ESOM aims to detect and preserve the ordering relationship among the training data items. It points out that the expanding coefficient $c_j(t)$, or the expanding force, is different for various data items. The larger the distance between a data item and the center of all data items is, the stronger the expanding force will be on the associated output neuron. Consequently, the neuron will be pushed away from the center.

We now briefly discuss another interesting trend based on the proof procedure. If $\vec{w}_j(t)$ is far away from $\vec{x}_k(t)$[1], $\langle \vec{e}_{w_j}, \vec{e}_{x_k} \rangle$ will be very small or even less than 0. From Eq.(12), $\frac{\partial F(\rho)}{\partial \rho} \approx -\frac{\rho}{\sqrt{1-\rho^2}}\sqrt{1-r^2} \leq 0$. In other words, the expanding coefficient $c_j(t)$ increases with ρ which is the distance of the input data item $\vec{x}_k(t)$ from the center. So, the ordering of $\|\vec{x}_k(t)\|$ is reflected by the expanding coefficient $c_j(t)$ and then is learned by $\vec{w}_j(t)$. This also explains why the topological error of the ESOM decreases more quickly than that of the SOM at the beginning of learning. A typical example can be found in Fig.4 in Section 3.

3 Experimental results

3.1 Experimental setting

We examined the ESOM on 3 synthetic data sets and 1 real-life data set. All experimental results are compared with those of the SOM in terms of both the quantization and the topological errors. All data sets were preprocessed using the same linear transformation as in Eq.(1) in order to compare results fairly. All experiments were done with the same set of parameters. The initial values of the learning rate ϵ, the neighbor width parameter σ, and the radius R were 0.5, 0.9 and 0.999 respectively. Both the learning rate α and the neighbor width parameter σ were decreased by factor of 0.998 per iteration. Except for the last data set, we used a rectangular grid with 20*20 neurons. All experiments were run for 2000 iterations in total.

The three synthetic data sets are quite interesting in both their special cluster shapes and locations as illustrated in Fig.3. The conventional clustering algorithms such as K-means and Expectation-Maximization (EM) are unable to identify the clusters. The first data set has 3,000 data items in 2-dimensional space with 3 clusters. The most inside cluster looks like a triangle which is surrounded by two strip-like clusters. This data set is designed to demonstrate the topology preservation capacity of SOMs, because we

[1] the case is common at the beginning of learning since the weight vector $\vec{w}_j(t)$ is randomly initialized.

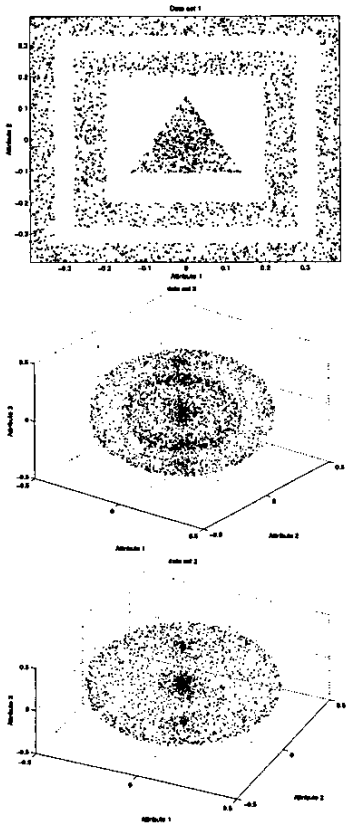

Figure 3. Illustration of Data set 1 (Upper), Data set 2 (Middle) and Data set 3 (Lower)

can compare the resultant mapping with the original data set directly. The second data set contains 2,000 data items in 3-dimensional space with 2 clusters (the middle one in Fig.3). The third data set has 3,000 3-dimensional data items (the lower one in Fig.3). The last data set, Iris, is a benchmark problem and is downloadable from UCI Machine Learning Repository (www.ics.ucs.edu/umlearn/ULSummary.html). It has 150 data items in 4-dimensional space with 3 classes. As there are not many data items in the last data set, we use an output grid with 10 * 10 neurons for both algorithms.

3.2 Results for the first data set

First, we check how good the mappings the two algorithms can generate in terms of the two criteria given in Subsection 2.1. Fig.4 illustrates the quantization and the topological errors during a typical run of both algorithms. It is clearly seen that the quantization error decreases gradually as the learning process continues. The quantization error of

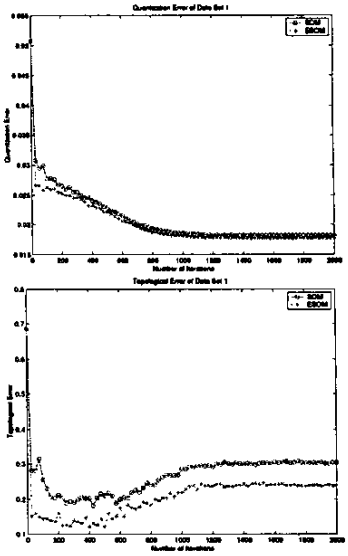

Figure 4. The quantization error (Upper) and the topological error (Lower) during the learning of the ESOM and the SOM.

the trained ESOM is 0.017 which is a bit smaller than that of the trained SOM, 0.018. During learning, the topological error curve clearly has three stages: decreasing, increasing and convergence. At the very beginning of the training process, the neuron's weights are fairly dislike, while some of them even contain remnants of random initial values, thus higher topological errors are got. After several iterations, the topological error decreases dramatically. Because the learning rate ϵ is large and the neighborhood function is also large then, the neurons adjacent on the grid may move much closer the input data item together. At this stage, the ESOM can learn the ordering topology of data items very quickly. As shown in Fig.4, the topological error of the ESOM is much smaller than that of the SOM. Though the topological errors of both algorithms increase later, the ESOM keeps the gain and always has smaller topological error than the SOM. Finally, the topological errors of the ESOM and the SOM are 0.238 and 0.304 respectively. ESOM makes about 20% improvement on the topological error in comparison with the SOM, and the ESOM gets slightly smaller quantization error than the SOM. Thus, the ESOM can generate better topology preserving maps than the SOM.

Now let us see how well the trained ESOM identifies the clusters in the first data set. Fig.5 illustrates the trained ESOM and SOM in the form of U-matrix. The x-axis and y-axis of the U-matrix indicate a neuron's position on the grid, and the z-axis is the average Euclidean distance of

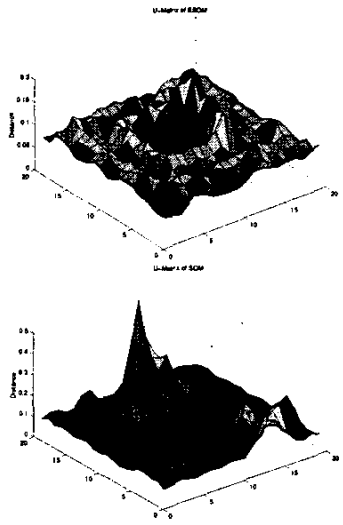

Figure 5. U-Matrix of the trained ESOM (Upper) and the trained SOM (Lower).

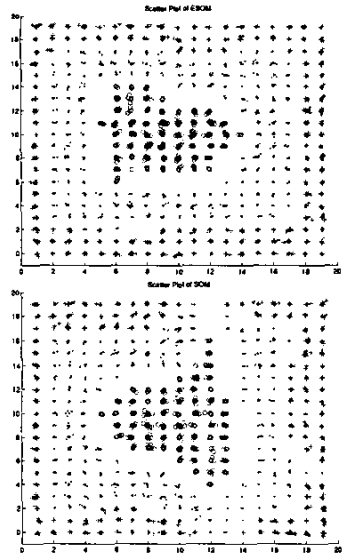

Figure 6. Scatter plot the trained ESOM (Upper) and the trained SOM (Lower).

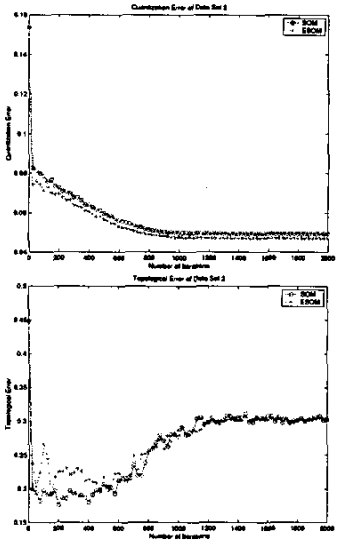

Figure 7. The quantization error (Upper) and the topological error (Lower) in the learning of the ESOM and the SOM.

neurons from its adjacent ones [2]. A sequence of consecutive peaks can be regarded as a boundary among clusters, while the basins beside are regarded as clusters. There are clearly two sequences of peaks in the ESOM's U-matrix, which indicate three clusters in the first data set. The layer structure in the data set has been clearly illustrated. In contrast, The boundaries in the SOM's U-matrix is not so clear because some high peaks blur the boundaries. The scatter plots shown in Fig.6 illustrate data clusters on the grid. In the scatter plot, each marker represents a mapping of a data item and the shape of the marker indicates its cluster label. The marker is placed on the winning neuron of the data item. To avoid overlapping, the marker has plotted with a small offset which is specified according to the data item's Euclidean distance from the winning neurons. The ESOM maps data items in well-organized layers. We can easily find the three clusters in its scatter plot which is quite similar with the original data set as shown in Fig.3. However, as we can see from Fig.6, the SOM cannot map data very well. The outer cluster in Fig.3 is even separated into three subclusters (indicated by '+').

3.3 Results for the second data set

Fig.7 shows the quantization and the topological errors during a typical run of both algorithms for the second data set. Though the topological errors of both the ESOM and

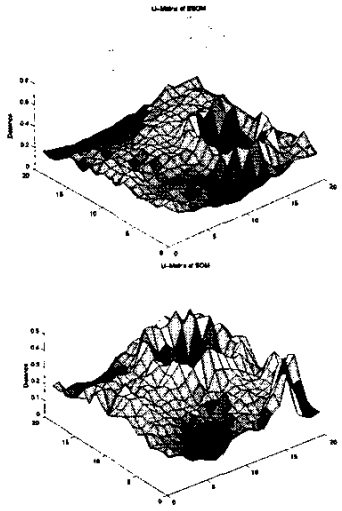

Figure 8. U-Matrix of the trained ESOM (Upper) and the trained SOM (Lower).

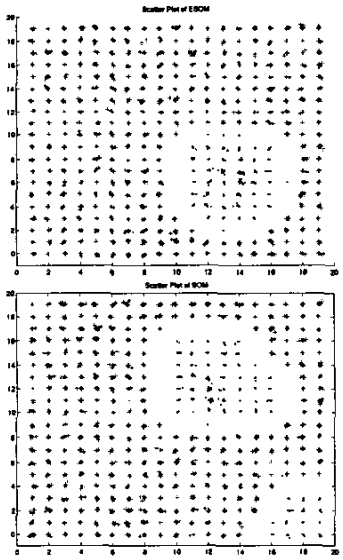

Figure 9. Scatter plot the trained ESOM (Upper) and the trained SOM (Lower).

the SOM are very close, the ESOM has smaller quantization error than the SOM. Observed from Fig.8, the U-matrix of the ESOM clearly shows two clusters as one cluster is separated by a sequence of peaks. However, the U-matrix of the SOM indicates three clusters. An extra cluster has been added at the bottom-right corner. This results is further confirmed by the scatter plots of the two maps as illustrated in Fig.9. The SOM separates one of the clusters in the original data set into two clusters(represented by '.'). In contrast, the scatter plot of the ESOM clearly shows two clusters where one cluster surrounds the other as in the original data set depicted in Fig.3.

Table 1. Average Quantization Errors based on 10 independent runs.

Data Set Name	ESOM	SOM	Improvement (%)
Data Set 1	0.01767	0.01800	1.80060
Data Set 2	0.04753	0.04832	1.64508
Data Set 3	0.05991	0.06042	0.83417
Data Set 4	0.02907	0.02774	-4.81993

Table 2. Average Topological Errors based on 10 independent runs.

Data Set Name	ESOM	SOM	Improvement (%)
Data Set 1	0.24088	0.26039	7.49154
Data Set 2	0.30215	0.30555	1.11241
Data Set 3	0.33132	0.34763	4.69203
Data Set 4	0.33200	0.33667	1.38624

Table 3. Average Execution Times based on 10 independent runs.

Data Set Name	ESOM	SOM	Difference (%)
Data Set 1	2371.6	2283.2	3.87176
Data Set 2	2068.9	2051.2	0.86291
Data Set 3	2904.7	2872.1	1.13506
Data Set 4	28.8	28.5	1.05263

Tables 1, 2 and 3 show the average quantization errors, topological errors and execution time for all the four data sets, based on 10 independent runs, respectively. We can see that the ESOM has better capability in topology preservation, while the execution times are comparable with those of SOM. However, the ESOM have not much improvement on quantization error yet, because topological error and quantization error are in conflict and it is difficult to optimize both of them at the same time.

4 Conclusion

In this paper, we have proposed an Expanding Self-Organizing Map (ESOM) to detect and preserve better topology correspondence between the input data space and the output grid. During the learning process of our ESOM, the flexible neuron net is expanding and the neuron corresponding to a distant data item gets large expanding force. Besides the neighborhood relationship as in the SOM (Self-Organizing Map), the ESOM can detect and preserve an linear ordering relationship as well. Our experiment results have substantiated that the ESOM constructs better visualization results than the classic SOM in terms of both the topological and the quantization errors. Furthermore, clustering results generated by the ESOM are more accurate than those obtained by the SOM on both synthetic and real-life data sets.

In our future work, we will study different methods to scale-up the ESOM algorithm for large data sets and investigate various techniques for visualizing large data sets efficiently, and compare the performance of the ESOM with other extensions of SOM.

Acknowledgement

This research was supported by RGC Earmarked Grant for Research CUHK 4212/01E of Hong Kong.

References

[1] H. U. Bauer, M. Herrmann, and T. Villmann. Neural maps and topographic vector quantization. *Neural Networks*, 12:659–676, 1999.

[2] J.A.F. Costa and M.L. de Andrade Netto. A new tree-structured self-organizing map for data analysis. In *Proceedings of International Joint Conference on Neural Networks*, volume 3, pages 1931–1936, 2001.

[3] Huidong Jin, Kwong-Sak Leung, and Man-Leung Wong. An integrated self-organizing map for the traveling salesman problem. In Nikos Mastorakis, editor, *Advances in Neural Networks and Applications*, pages 235–240, World Scientific and Engineering Society Press, Feb. 2001.

[4] Huidong JIN, Kwong-Sak Leung, Man-Leung Wong, and Zongben Xu. An efficient self-organizing map designed by genetic algorithms for the traveling salesman problem. Accepted by IEEE Trans. SMC Part B., May 2002.

[5] J. S. Kirk and J. M. Zurada. A two-stage algorithm for improved topography preservation in self-organizing maps. In *2000 IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 2527–2532. IEEE Service Center, 2000.

[6] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, 1997.

[7] Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko SalojAvi, Vesa Paatero, and Antti Saarela. Organization of a massive document collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3):574–585, May 2000.

[8] Siming Lin and Jennie Si. Weight-value convergence of the SOM algorithm for discrete input. *Neural Computation*, 10(4):807–814, 1998.

[9] Mu-Chun Su and Hsiao-Te Chang. A new model of self-organizing neural networks and its application in data projection. *IEEE Transactions on Neural Networks*, 12(1):153 –158, Jan. 2001.

[10] Juha Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3:111–26, 1999.

[11] Juha Vesanto. Neural network tool for data mining: SOM toolbox. In *Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems (TOOLMET2000)*, pages 184–196, Oulu, Finland, 2000.

[12] Juha Vesanto and Esa Alhoniemi. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 11(3):586–600, May 2000.

[13] Thomas Villmann, Ralf Der, Michael Herrmann, and Thomas M. Martinetz. Topology preservation in self-organizing feature maps: exact definition and measurement. *IEEE Transactions on Neural Networks*, 8(2):256 –266, March 1997.