On-line Pattern Analysis by Evolving Self-Organizing Maps

D. Deng and N. Kasabov

Department of Information Science, University of Otago, Dunedin, New Zealand

(E-mail: {ddeng, nkasabov}@infoscience.otago.ac.nz)

Abstract

Many real world data analysis and processing tasks require systems with the ability of on-line, self-adaptive learning. In this paper present some theoretical background for the Evolving Self-Organising Map (ESOM) model and further apply it in solving some on-line pattern analysis problems. Results are compared with some benchmarks.

Keywords: on-line learning, self-organizing, clustering, classification

1. Introduction

On-line data analysis is catching more attention as the volume of information in computer networks keep on exploding. Despite of the rapid development in the theories of computational intelligence, it remains a challenging problem with a number of considerable obstacles, such as in the cases when the statistical models of data are unknown or time-dependent, and the parameters of the learning system need to be updated incrementally while only a partial glimpse of incoming data is available.

In the context of data clustering and vector quantisation, a straightforward approach is the well known *K*-means algorithm [13], which calculates each cluster centre as the mean of data vectors within the cluster. Given an input $\mathbf{x}$, its on-line version is applied without a priori knowledge of data distribution:

$$\Delta \mathbf{w}_j = \begin{cases} \gamma(\mathbf{x} - \mathbf{w}_j), & \text{if } j = i(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where γ is the learning rate, and $i(\mathbf{x})$ is the winner among nodes. This learning rule is also referred as 'local k-means algorithm' [14]. It is of Winner-Takes-All scheme and can operate in a dynamic environment with continuously arriving data. One drawback is that it can suffer from confinement to local minima [15]. To avoid this some "soft" computing schemes are proposed [16][14], in which not only the "winner" prototype is modified, but all reference vectors are adjusted depending on their proximity to the input vector.

Kohonen's self-organizing map (SOM) [10] further introduces topology preserving ability by means of applying neighbourhood functions in feature map for node modification. The map is typically organised on a two dimensional lattice for visualisation purpose. The learning rule of SOM is

$$\Delta \mathbf{w}_i = h_{i,b}(\mathbf{x} - \mathbf{w}_i) \quad (2)$$

where $\mathbf{w}_b$ denotes the winner node, and $h_{i,b}$ is a neighbourhood usually defined as a Gaussian or a bubble function of the node indexes i and b.

Such a learning rule is linked to an optimisation process which targets on achieving both a minimum representation error on the best matching prototype for the input, and topological preserving of the reduced prototype space.

In SOM the topology order of the prototype nodes, indicated by the node indexes, are pre-determined and the learning process is to move the initialised nodes onto appropriate positions in the low dimensional feature map. As the original input manifold can be complicated with an inherent dimension larger than that of the feature map (usually set as 2 or 3 for visualisation purpose), the dimension reduction in SOM can be too drastic, generating a folded feature map.

The topology constraint on the feature map with a low dimension is removed in [15], where a neural-gas model is proposed with a learning rule similar to that of SOM, but the prototype vectors are organised in the original manifold of the input space. The weight updating rule is

$$\Delta \mathbf{w}_i = \gamma h_\lambda(k_i(\mathbf{x}, \mathbf{w})(\mathbf{x} - \mathbf{w}_i) \quad i = 1, ..., N. \quad (3)$$

where γ is the learning rate, and k_i is the *neighbourhood rank* of the *i*-th prototype corresponding to the current input. The neighborhood function is defined as

$$h_\lambda(k_i(\mathbf{x}, \mathbf{w}) = e^{-k_i(\mathbf{x}, \mathbf{w})/\lambda} \quad (4)$$

with $\lambda > 0$. Each time when the weights are updated the neighbourhood rank, i.e. the matching rank of prototypes, needs to be computed. This brings up the time complexity for one adapting step of the algorithm to the scale of $N \log N$ in a serial implementation, while searching for the best matching unit in K-means or SOM scales only with N.

Fritzke [5] proposed the growing neural gas (GNG) model, which allows the neural gas model to grow by adding new nodes adaptively. Bruske and Sommer [2] presented a similar model called dynamic cell structure

(DCS-GCS). Both GNG and DCS-GCS need to calculate local resources for prototypes, which introduces extra computational effort and reduces their efficiency.

The evolving self-organising map (ESOM) [3] model was proposed in the light of the works mentioned above. It is similar to GNG but does not require local resource calculation. Its node insertion mechanism also allows the prototypes evolve quicker than DCS and GNG. We have applied the model in macro-economic data analysis and some pattern recognition experiments. In the following parts of this paper, we will first give a brief introduction to the ESOM model and then present simulation results done on some benchmark problems.

2. ESOM revisited

The ESOM network structure is similar to that of GNG. The algorithm starts with a null network. Nodes embedded in the original data space are created incrementally. When new input is presented the prototypes in the network compete with each other. The winner node sets up connections with its first two nearest neighbours.

Assume the current input is $\mathbf{x}$, and the existing prototype set is $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_N\}$. If

$$\|\mathbf{w}_i - \mathbf{x}\| > \epsilon, \ \ \forall \mathbf{w}_i \in \mathcal{W} \tag{5}$$

where ϵ is a distance threshold, then a new node is inserted as

$$\mathbf{w}_{N+1} = \mathbf{x} \tag{6}$$

The new node is inserted representing exactly the poorly matched input vector, resulting in a maximum activation on the new node. This simple approach brings advantages especially when handling clustered data. Although direct allocation is sensitive to noise and may introduce some artifacts in clustering, this can be mitigated by automatic deletion of obsolete nodes.

If the new input matches well to some prototypes, the activation on the prototype nodes is defined as

$$a_i = e^{-2\|\mathbf{x}-\mathbf{w}_i\|^2/\epsilon^2} \tag{7}$$

indicating the closeness of the current input to weight vector i. To update the weight vectors of the matched prototypes, we consider the following generalised learning rule:

$$\Delta\mathbf{w}_i = \gamma h_i(\mathbf{x}, b)(\mathbf{x} - \mathbf{w}_i), \quad i = 1, ..., N \tag{8}$$

where b is the index for the winner node, γ is the learning rate. If we let $h_i(\cdot) = h_{i,b}$, i.e., the neighbourhood function in SOM, it gives the Kohonen rule. Defining $h_i(\cdot) = k_i(\mathbf{x})$, i.e, the neighbourhood rank, then the learning rule becomes that of neural gas.

To overcome the topological restriction of the SOM model, and avoid the time-consuming sorting procedure involved in neighbourhood ranking as in neural gas, here we define $h_i(\mathbf{x}) = a_i(\mathbf{x}) / \sum_k a_k(\mathbf{x})$. Hence the learning rule of ESOM is

$$\Delta\mathbf{w}_i = \gamma \frac{a_i}{\sum_k a_k(\mathbf{x})}(\mathbf{x} - \mathbf{w}_i) \tag{9}$$

On the other hand, it has been pointed out that such a learning rule leads to a stochastic process to minimise the Kullback discrepancy between the input data and the internal representation of the network [1]. The Kullback criterion can be written as:

$$G(\mu, h) = \int \mu(\mathbf{x}) \log \frac{\mu(\mathbf{x})}{h(\mathbf{x})} d\mathbf{x} \tag{10}$$

where μ is the probability density of the input data and h is the internal representation presumed as a mixture of Gaussians:

$$h(\mathbf{x}) = \frac{1}{N(\sqrt{2\sigma})^d} \sum_{i=1}^{N} \exp(-\frac{\|\mathbf{x} - \mathbf{w_i}\|^2}{2\sigma^2}) \tag{11}$$

where d is the dimension of the input space. By carrying out a gradient descent on $G(\cdot,$
$cdot)$ the learning rule in Eq.(9) can be derived.

By removing geometric constraints in the SOM model, ESOM allows for more flexibility in the prototype space. This trait, however, becomes a disadvantage when visualisation of the prototype nodes is required, as they are now embedded in the original data space usually of a high dimension. It needs to turn to dimension reduction algorithms such as PCA and Sammon's projection to visualise the *feature map* [3].

3. Simulations

Previous studies made on some benchmark data sets have shown that ESOM works faster and more effectively in classification tasks and macro-economic data analysis [3]. In this paper we introduce more simulations of pattern analysis tasks. We have also strengthened the evaluation of classification results using cross validation.

3.1. Colour image quantisation

Let us first consider colour image quantisation as a problem of on-line data clustering. Colour image quantisation is a process for reducing the number of colours of a digital colour image. It is one of the most frequently used operations in computer graphics and image processing and is closely related to image compression. There are two basic approaches: pre-clustering (with methods such as median-cut[8] and octree[7] etc.) and post-clustering (with methods such as geometric clustering and neural network algorithms). Although widely studied for many years, it remains a

time consuming task which is difficult to achieve an efficient on-line implementation.

We apply the ESOM algorithm in colour image quantisation and compare the results with those achieved by other methods including median-cut, octree, Wu's method [19], and local K-means (LKM). Among these methods, only ESOM and local K-means operate in on-line mode, in which the image can be quantised pixel-by-pixel. The others mostly need to manipulate image colour histograms and have to be carried out in a batch mode after all pixel data are made available.

Three 24-bit true-colour images are chosen for this study: Pool Balls, Mandrill, and Lenna. The Pool Balls image is artificial and contains smooth colour tones and shades with a size of 510×383. The Mandrill image is of 262144 (512×512) pixels but has a very large number of colours (230427). The Lenna image (size 512×480) contains both smooth areas and fine details. All three images are popularly used in image processing literature.

Images are all quantised in RGB space onto 256 colours except for ESOM, whose colour numbers are slightly smaller. For fair comparison dithering process is not introduced for all methods. We denote the quantisation process as a mapping from the input colour $I_i = (r_i, g_i, b_i)$, to the best-matching colour c_m in a reduced palette $\mathcal{C} = \{c_j | j = 1, 2, ..., 256\}$. The normalised root mean square error (NRMSE) of quantisation is defined as

$$E_q = \sqrt{\frac{1}{N} \sum_{i=1}^{M} \|I_i - c_m\|^2} \qquad (12)$$

M is the number of pixels in image I. The variance of quantisation error is another factor which influences the visual quality of the quantised image. The standard deviation of quantisation error is define as

$$\sigma_q = \sqrt{\frac{\sum_i (\|I_i - c_m\| - E_q)^2}{M}} \qquad (13)$$

Table 1: Quantisation performances: E_q / σ_q

Methods	Images		
	Pool Balls	Mandrill	Lenna
Median-cut	2.6/8.3	11.3/5.6	6.0/3.5
Octree	4.2/3.6	13.2/5.0	7.6/3.8
Wu's	2.2/2.2	9.9/4.6	5.5/2.9
LKM	3.5/2.8	11.5/5.4	6.7/2.8
ESOM	2.4/2.6	9.5/3.9	5.3/2.4

Performance of different methods are compared in Table 1. From these results we can see that ESOM achieved the best results for the natural images (Mandrill and Lenna). For the artificial Pool Balls image its performance is comparable with Wu's method, the best among the popular colour quantisation methods. Generally ESOM not only achieves a very small value of average quantisation error, but its error variance is also the smallest. This is consistent with the observation that images quantised by ESOM have better visual quality than those done by other methods. To demonstrate this the zoomed region of the quantised Lenna images are compared in Fig.1. To evaluate the convergence

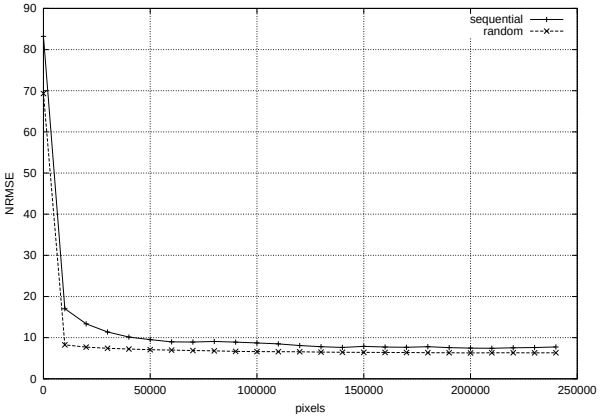


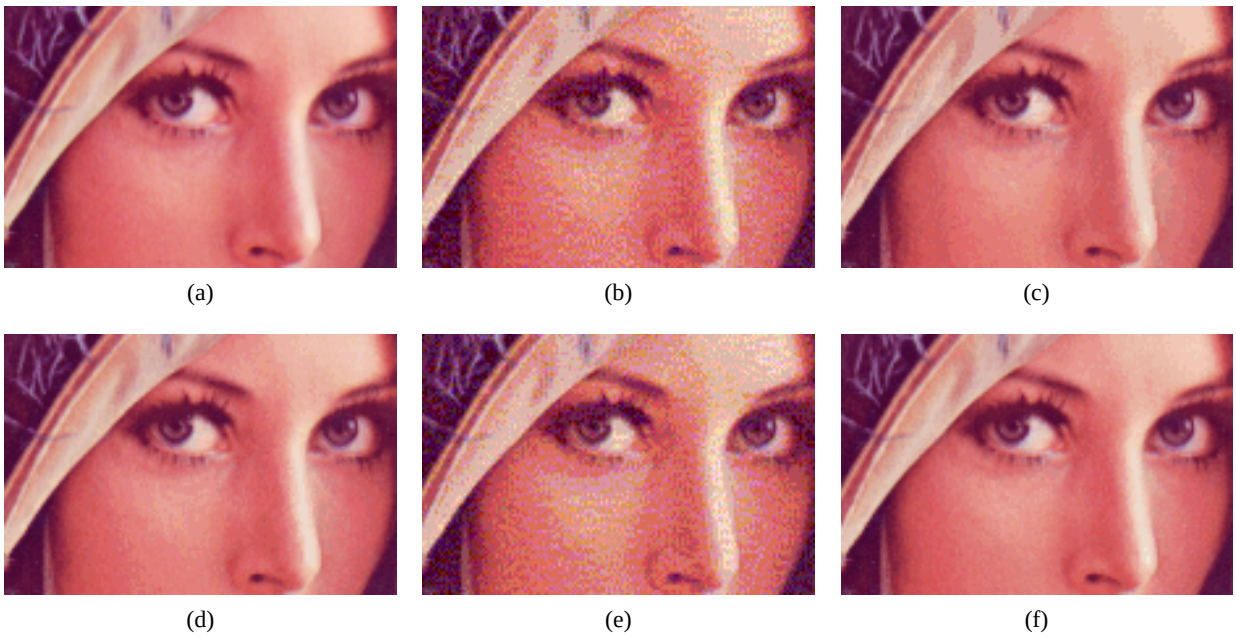
Figure 2: Quantisation error versus number of pixels learned in different orders.

of the algorithm, we compute on-line the NRMSE of using the allocated colour maps to quantise the whole image. As strong correlation typically exists among image pixels, it is found that a small fraction of the image data (less than 10%) can lead the quantisation error quickly into convergence, as shown in Fig.2 for the Lenna image. Instead of quantising image pixels in sequential order, convergence is quicker by presenting image pixels to the algorithm in random order. Judged by visual quality, however, the network needs to learn about 20% of the image data before generating a good display. The simulation program of ESOM written in C runs on an Intel Pentium II machine with Linux 2.2. To get the final palette it takes 2 seconds to fully scan the colour images.

The advantage of using an on-line algorithm in colour quantisation is that a progressive display mode can be enabled and enhance the quality of image and video display on low-end computers connected to the Internet. Although the algorithm running as a computer program is not practical for this purpose because of the intensity of computation, a hardware implementation could make use of parallelism and therefore greatly speed up the quantisation process for real time applications.

3.2. Pattern classification

ESOM is by nature an unsupervised learning algorithm. But just like SOM and other clustering methods, it can be applied to supervised classification tasks. Here we adopt a plain 1-NN approach, i.e., the prototypes keep their labels while being updated, and the



(a) (b) (c)

(d) (e) (f)

Figure 1: Quantisation quality comparison: (a)The zoomed region of the original Lenna image, and those quantised by (b)median cut, (c)octree, (d)Wu's method, (e)local K-Means, and (f)ESOM.

class label of the winner is used for classification.

Two benchmark data sets are used in our experiments. The Pima Indian diabetes data is from the University of California Irvine UCI machine learning repository. The data set has eight inputs, and a total of 768 examples in two classes. The speaker independent vowel data set is the CMU artificial intelligence repository. It consists of 990 frames of speech signals from four male and four female speakers. A 10-fold cross validation process is adopted to train and test the ESOM modules. For each data set, different partition of the data is tried for 5 times and the average performance is calculated.

For the diabetes data, all ESOM modules have $\epsilon = 0.4$ and the average network size is of 97 nodes. Results are listed in Table 2 in comparison with other studies using the same experiment process. Results on k nearest neighbourhood (k-NN), classification and regression trees (CART), multi-layer perceptron (MLP), learning vector quantization (LVQ) and linear discriminant analysis (LDA) are from [18], and CART-DB from [17].

For the vowel recognition data, the same experiment process is repeated. We set $\epsilon = 1.2$. The average number of nodes in the network is 233. In Table.3, the performance of ESOM is compared with those two approaches in [17].

3.3. Predictive classification

Owing to the on-line learning ability of ESOM, whenever a new example arrives the network can classify it in advance using existing prototypes. If the classification is correct compared with the class label

Table 2: Performance Comparison on the Diabetes Problem.

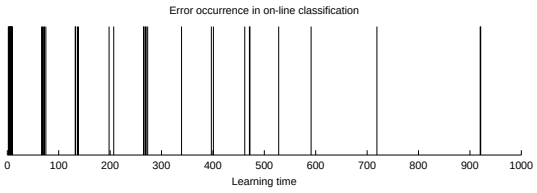
Classifier	% Correct (Average)
k-NN	71.9
CART	72.8
CART-DB	74.4
MLP	75.2
LVQ	75.8
LDA	77.5
ESOM	78.4 ± 1.6

Table 3: Performance Comparison on Vowel Recognition using Cross Validation

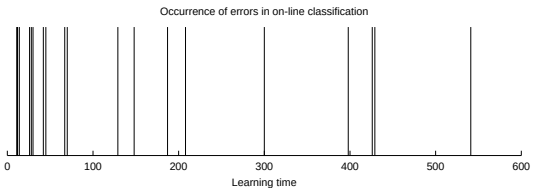
Classifier	% Correct (Average)
CART	78.2
CART-DB	90.0
ESOM	95.0 ± 0.5

carried by the new example, the network then adapts itself to the new example. Otherwise a new node is generated for the new example. In this way the network continually improves its performance in an on-line mode. We call such a process as *predictive classification*.

Good results have obtained on both the vowel and diabetes data sets, indicating the fast adaptive ability of ESOM in on-line learning. The performance of on-line predictive classification is shown in Fig. 3. Only 36 classification errors are made when the ESOM network is evolved from the data set, i.e. with an overall error rate of 3.4%. After exposure to the first 528 entries, the ESOM module makes only 4 errors for the left 462 entries for predictive classification. Similar performance is obtained for the diabetes data.



(a)



(b)

Figure 3: On-line predictive classification error occurrence over time for (a) the vowel data and (b) the diabetes data.

These predictive classification simulations work without node pruning and aggregation. In practice as generally there will be a limit on number of nodes for the network, node aggregation and pruning processes are necessary and consequently introduce forgetting effects and reduce the classification rate. We argue, however, that this is not a bad thing at all when dealing with dynamic data environments.

4. Discussion and conclusion

This paper tries to present the ideas of ESOM algorithm with some historical background, and to justify its application in on-line pattern analysis tasks by carrying out a few benchmark studies. The results have been promising. Nevertheless, there are a few open-questions left untackled so far for the algorithm.

As ESOM is aimed at achieving "life-long" learning, the learning rate should not drop to zero asymptotically, otherwise it will fail to learn novel examples at later stage, or to follow the possible fluctuation of statistical properties of incoming data. We simply choose a small constant value as the learning rate, which, as pointed out by Heskes and Kappen [9] for a competitive learning scenario, can achieve the trade-off of adaptability and accuracy for the network.

Our algorithm can actually be categorised as an on-line implementation of the leader-follower clustering [4], where the adaptive resonance theory (ART) was given as an instance. The distance threshold in ESOM algorithm plays a similar role as the vigilance threshold in the ART model, whose value determines implicitly the number of clusters to be formed. With little information about the data in an on-line mode, however, there is no guideline in finding the right value for the threshold. So far setting the threshold in leader-follower clustering has been done through trial-and-error, until the 'proper' number of clusters are formed. Some future work may be done in the effort to adaptively tune the threshold, and find guidelines to split or merge clusters.

On the other hand, given the distance threshold in different scales, it is easy to construct hierarchical mappings with ESOM, with maps generated in a multi-resolution mode. This may facilitate the application of ESOM in information retrieval for instance.

Last but not least, ESOM is proposed as a computational model for on-line information processing from an engineering point of view. Hence the biological plausibility aspect of neural modeling is not considered. Kohonen proclaims the biological plausibility in SOM [11]. Some cognitive and psychological studies, for example [12], also suggest that there is a semantic dimension as low as 2 in human mind. The human brain still remains fascinating yet mystic to us. From an engineering point of view, however, number of computational units can not be compared to that of human brain and it is often required to keep a condensed set of prototypes. Under such circumstance it makes sense to explore artificial neural network models such as ESOM etc. which may not fit in a biological background.

Acknowledgement This work is supported by the Foundation for Research, Science and Technology (FRST) New Zealand, under grant UOOX0016.

References

[1] M. Benaim and L. Tomasini, Approximating functions and predicting time-series with multi-sigmoidal basis functions, in I. Aleksander and J. Taylor eds, Artificial neural networks 2 (Elsevier 1992), 407-411.

[2] J. Bruske, and G. Sommer, Dynamic cell structure learns perfectly topology preserving map, Neural Computation 7 (1995) 845-865.

[3] D. Deng, and N. Kasabov, ESOM: An algorithm to evolve self-organizing maps from on-line data streams, Proc. of IJCNN 2000, (IEEE Press, 2000), VI:3-8.

[4] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, 2nd Ed. (John Wiley & Sons 2001).

[5] B. Fritzke, Growing cell structures - a self-organizing network for unsupervised and supervised learning. Neural Networks 7 (1994) 1441-1460.

[6] B. Fritzke, A growing neural gas network learns topologies, in D. Touretzky and T.K. Keen eds., Advances in neural information processing Systems 7, (MIT Cambridge MA, 1995) 625-632.

[7] M. Gervautz and W. Purgathofer, A simple method for color quantization: octree quantization, in A. Glassner ed., Graphics Gems, (Academic, New York, 1990) 287-293.

[8] P. Heckbert, Color image quantization for frame buffer display. Computer Graphics (SIGGRAPH) 16 (1982) 297-307.

[9] T.M. Heskes and B. Kappen, Learning processes in neural networks. Physical Review A 44 (1991) 2718-2726.

[10] T. Kohonen, Self-organizing formation of topologically correct feature maps. Biological Cybernetics 43 (1982) 59-69.

[11] T. Kohonen, Self-Organizing Maps, second edition, (Springer, Berlin, 1997).

[12] W. Lowe, What is the dimensionality of human semantic space, Proc. of the 6-th Neural Computation and Psychology Workshop, Springer Verlag, pp.303-311, 2000.

[13] J. MacQueen, Some methods for classification and analysis of multivariate observations, in L.M. LeCam and J. Neyman eds., Proc. 5th Berkeley Symp. on Mathematics, (University of California Press, Berkeley, 1967) 281-297.

[14] J.L. Marroquin and F. Girosi, Some extension of the K-means algorithm for image segmentation and pattern classification, Technical Report 1390, Massachusetts Institute of Technology, 1993.

[15] T.M. Martinetz, S.G. Berkovich and K. J. Schulten, Neural-Gas network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4 (1993) 558-569.

[16] S.J. Nowlan, Maximum likelihood competitive learning, in D. Touretzky ed., Advances in Neural Information Processing Systems 2 (Morgan Kauffman, New York, 1990) 574-582,

[17] N. Shang and L. Breiman, Distribution based trees are more accurate, in Proc. of ICONIP'96, (Springer, Hong Kong, 1996) 133-138.

[18] B. Ster and A. Dobnikar, Neural networks in medical diagnosis: comparison with other methods, in A. Bulsari et al. eds, Proceedings of Inter. Conf. on Engineering Applications with Neural Networks (London, 1996) 427-430.

[19] X. Wu, Color quantization by dynamic programming and principal analysis. ACM Trans. on Graphics 11 (1992) 348-372.