Evolving Self-Organizing Maps for On-line Learning, Data Analysis and Modelling

Da Deng, Nikola Kasabov
Department of Information Science,
University of Otago,
PO Box 56, Dunedin, New Zealand

Abstract

In real world information systems, data analysis and processing are usually needed to be done in an on-line, self-adaptive way. In this respect, neural algorithms of incremental learning and constructive network models are of increased interest. In this paper we present a new algorithm of evolving self-organizing map (ESOM), which features fast one-pass learning, dynamic network structure, and good visualisation ability. Simulations have been carried out on some benchmark data sets for classification and prediction tasks, as well as on some macroeconomic data for data analysis. Compared with other methods, ESOM achieved better classification with much shorter learning time. Its performance for time series modelling is also comparable, requiring more hidden units but with only one-pass learning. Our results demonstrate that ESOM is an effective computational model for on-line learning, data analysis and modelling.

1 Introduction

Many real world information systems use data from on-line data streams that are updated frequently. To extract useful information hidden among these multivariate data, a number of techniques can be applied, such as scientific visualisation, dynamic clustering, and classification etc. As the data environment is dynamic, it often demands that the computational models should have incremental learning ability, and evolve with the changes in data. By incremental learning we refer to Shaal and Atkeson (1998), addressing a learning scenario in which

- Input and output distributions of data are not known and these distributions may change over time.
- The parameters of the learning system are updated incrementally.
- Only a limited memory is available so that data have to be discarded after they have been used.

Such a situation exists in biological systems as well as in engineering applications, such as robotic systems and process control.

In the context of data clustering and classification, a straightforward approach is the well known K-means algorithm (MacQueen 1967), which calculates each cluster centre as the mean of data vectors within the cluster. The performance of this algorithm has been found to be rather stable, but it has a limitation that the proper number of clusters needs to be specified. This is sometimes difficult if we have no a priori knowledge of the data distribution. Its on-line version may also suffer from confinement to local minima (Martinetz et al. 1993). To avoid this a “soft-max” principle is applied to modify reference vectors (Nowlan 1990), in which not only the “winner” is modified, but all reference vectors are adjusted depending on their proximity to the input vector. Kohonen's self-organizing feature map (SOM) (Kohonen 1982) is another model incorporating a soft-max adapting rule, using a Mexican hat neighbourhood function to modify map nodes. It is a good choice for vector quantisation in applications such as speech and image coding, featuring topology preserving ability and approximation of data distribution. SOM has found great success with a large amount of applications in various fields (Kohonen 1997).

The SOM model has a fixed topology on the feature map and the number of map nodes is also fixed. Once a map is learned it can not change its size to meet the need of new data environment. The feature map space of low dimensionality is intended to facilitate visualisation, but for data of high complexity the feature map can be folded and hence generate poor topolgy representation of the data. During the learning process SOM may also suffer from border effects and special treatment has to be considered (Kohonen 1997). Training data need to be presented to the model for many epochs before a good representation is learned on the feature map, especially to compensate for a potential inappropriate initialisation.

SOM is basically an unsupervised algorithm and it is rarely used for tasks such as time series prediction and pattern classification. Kohonen (1997) suggests to use LVQ in these cases. Meyering and Ritter (1992) proposed a Local Linear Mapping (LLM) network for supervised learning in a computer vision problem. LLM has two layers of weights. The lower layer is trained with the SOM algorithm, but the upper layer updates its weights by a LMS algorithm. Vesanto (1997) incorporates a local linear regression model on the top of a SOM network in a time-series prediction problem and achieved very good results. He constructs local data sets for the prototype vectors and uses linear regression models on these local data sets. Strictly speaking, this is not an incremental learning approach and the complexity of the model is larger than the scale of the number of prototype vectors.

Other variations of the SOM models try to introduce improvements into the computation process and the effectiveness of the feature mapping. The constraints of a low dimensional mapping topology is removed in (Martinetz 1993), where a neural-gas model is proposed with a learning rule similar to SOM, but the reference vectors are organized in the original manifold of the input space. Blackmore and Miikkulainen (1993) proposed an incremental grid growing algorithm, where nodes and connections can be added to, or deleted from, the feature map, which is of fixed low dimensionality. Fritzke (1991,1994) proposed a growing cell structure (GCS), which uses a fixed topology dimension for reference vector space, but there is no pre-defined layout order for network nodes and the topology is much more flexible. The network creates new nodes whenever input data is not closely matched to existing reference vectors, and sets up connections between nodes. In (Fritzke 1995) the fixed graph topology of GCS is removed, giving a new growing neural gas (GNG) model, which has its origin also from the neural gas model. Bruske and Sommer (1995) presented another similar model, dynamic cell structure (DCS-GCS), differing from GNG slightly in the location of node insertion. GCS, GNG and DCS-GCS can be applied to supervised learning by adding an additional output weight layer which adopts a delta learning rule. A comparison on the performance of GNG, GCS and fuzzy ARTMAP is made in (Heinke and Hamker 1998).

Apart from the connection to SOM, there are other approaches which tackle on the problem of incremental learning and automatic resource allocation. Platt (1991) proposed a resource-allocating network (RAN) model, which allows for allocation of new computational units whenever unusual patterns are presented. Network paramters are updated using standard LMS gradient descent. Further improvement on the RAN algorithm is reported by replacing the LMS rules with other learning rules, for example, in (Kadirkamanathan 1993; Rosipal 1998). Schaal and Atkeson (1998) addressed the importance of local receptive field system for robust incremental learning, and proposed a receptive field-weighted regression (RFWR) model. An ECOS paradigm is proposed in (Kasabov 1998a), setting out principles for on-line construction of intelligent information systems using connectionist-based models. The ECOS principles include: fast incremental on-line learning, evolvable network structure, and knowledge interpretation and manipulation. As an instance of ECOS prototypes, Kasabov (1998b) proposed an evolving fuzzy neural network (EFuNN) model, which is found to be very competitive for time series prediction and pattern classification.

In this paper we introduce another computational model from the ECOS family, called Evolving Self-Organizing Map (ESOM). ESOM is proposed as an evolving neural algorithm for unsupervised tasks such as clustering and visualisation. It has close relation with the Kohonen SOM, GCS/GNG and EFuNN. The main features of ESOM include fast incremental learning, self-adaptive network structure, and good topology preserving for data visualisation.

During learning, an ESOM network creates new nodes and sets up connections dynamically. Insertion of nodes are straightforward rather than using mid-point interpolation as in GCS/GNG. This is close to the approach of EFuNN and RFWR, suggesting that data of novelty are first memorised and then adapted with the changing data.

In our model, prototype nodes have a limited number of connections to other nodes, which confine the neighbourhood relationship among them. During learning the updating of learning weight vectors occurs only on the node which wins the competition and on its neighbour nodes. Comparing to algorithms which have rigid neighbourhood definition (such as SOM), or those who are free of neighbourhood definition and undergo a global updating (such as the neural gas), this approach is faster and more effective owing to its localised nature of learning.

ESOM uses the same dimensionality as that of the input space, and it does not pre-assume any topological order and connectivities among network nodes. On the contrary, the network connections are created when necessary and the neighbourhood relationship is built as-it-is. This gives ESOM great flexibility and efficiency to learn a good representation of the input data, especially for data from a complex and high dimensional manifold. SOM-like algorithms typically need to start with large neighbourhood so as to unfold maps of ill initialisation and it is time-consuming. With less geometrical constraints on the mapping space, ESOM is more likely to generate a compact representation of input data with better efficiency.

The rest of the paper is organised as follows. In Section 2, we first give a brief introduction of the ESOM computational model. In section 3, simulations on benchmark data sets for classification and prediction are described and results are compared with other methods. A case study on international macroeconomic data is also used and a world macroeconomic map is generated. Finally, conclusions are given in Section 4.

2 The ESOM Computational Model

2.1 Network structure

The network structure of ESOM is different from that of SOM. No topological constraint is given for the feature map a priori and prototype nodes are not organised onto one- or two-dimensional lattices.

The ESOM network starts without any nodes. During learning, the network updates itself to capture the on-line incoming data, creating new nodes and setting out new connections when necessary. Connections between map nodes are used to maintain the neighbourhood relationships between close nodes. The strength of the neighbourhood relation is determined by the distance between connected nodes. If the distance is too big, it results in a weak connection and the connection can be pruned. In this way the feature map can be split apart and data structures such as clusters and outliers can emerge.

Fig.1 shows an ESOM network with five nodes, each of them labelled with a number in the order of its creation. The newest one, node 5, is created and connections with node 2 and 1 are set up. Node 2 and 1 are the winner and second winner respectively at the current time. The weakest connection between node 2 and 3 is then clipped.

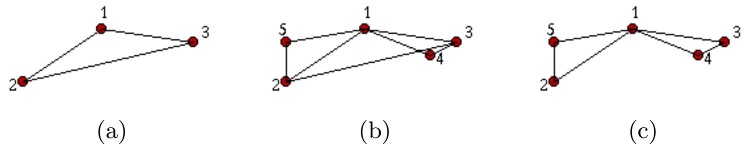

Fig. 1: An ESOM network grown to five nodes. (a) The first three nodes are created; (b) Two more nodes (node 4 and 5) are added; (c) The weakest connection is removed.

2.2 The algorithm

We denote the ESOM network at time t as a triplet of a prototype node set $\Omega \subset E^d$, an interconnection set $\mathcal{C}$, and a parameter set $\mathcal{P}$:

$$\mathcal{E}^t = (\Omega^t, \mathcal{C}^t, \mathcal{P}) \tag{1}$$

with each node $\mathbf{w}_i \in \Omega^t$ as a vector of dimension d, $i = 1...N$, and N is the current number of nodes in Ω^t.

The learning process can be summarised as follows:

1. Input a new data vector $\mathbf{x}$;
2. If there is node generated ($N = 0$), or none of the existing nodes matches the input vector within a distance threshold ϵ, i.e., for all $i = 1, ..., N$,

   $$d(\mathbf{w}_i, \mathbf{x}) = \|\mathbf{w}_i - \mathbf{x}\| > \epsilon,$$

   create a new node in the network which represents exactly the input $\mathbf{x}$:

   $$\begin{gathered} \mathbf{w}_{N+1} = \mathbf{x} \\ \Omega^{t+1} = \Omega^t \cup \mathbf{w}_{N+1} \\ N \leftarrow N + 1 \end{gathered} \tag{2}$$

   Connect the new node with its two nearest neighbours $\mathbf{w}_{n1}$ and $\mathbf{w}_{n2}$, and connect $\mathbf{w}_{n1}$ and $\mathbf{w}_{n2}$ if they are not connected yet:

   $$\mathcal{C}^{t+1} = \mathcal{C}^t \cup c(\mathbf{x}, \mathbf{w}_{n1}) \cup c(\mathbf{x}, \mathbf{w}_{n2}) \cup c(\mathbf{w}_{n1}, \mathbf{w}_{n2}) \tag{3}$$

   Here $c(\cdot, \cdot)$ denotes a connection between two nodes.
3. Otherwise update the matching node and each of its neighbours, denoted as $\mathbf{w}$, according to their distances to the input vector $\mathbf{x}$, a relation represented by a function f:

   $$\Omega^{t+1} = f(\Omega^t, \mathbf{x}), \tag{4}$$

   with each node being modified as

   $$\Delta\mathbf{w} = \gamma\ e^{-d^2(\mathbf{w},\mathbf{x})/2\sigma^2}(\mathbf{x} - \mathbf{w}) \tag{5}$$

   where γ is a small constant learning rate, and σ controls the spread of neighbourhood. Usually we set $\sigma = \epsilon$.
4. Reset the strength of connections between the winner (or the newly created node) and its neighbours. The connection strength $s(i, j)$, for the connection $c(\mathbf{w}_i, \mathbf{w}_j)$, is defined as

   $$s(i, j) = \epsilon / d(\mathbf{w}_i, \mathbf{w}_j) \tag{6}$$

5. After every T_p steps of learning time, prune the weakest connection. If isolated nodes appear, prune them as well;

6. Repeat all steps above.

The parameter set is defined as $\mathcal{P} = \{\epsilon, \sigma, \gamma, T_p\}$.

With a relatively small learning rate and a data sequence which is long enough, it can be expected that after the presentation of a certain number of data examples an optimum set of prototypes representing the data stream will be evolved. A similar case on competive learning is analysed in (Heskes and Kappen 1991) using Gaussian approximation. As ESOM is aimed at achieving "life-long" learning, strict convergence of the algorithm is not a critical issue.

The weight vector update rule in Eq.(5) is similar to that of SOM, except that for the neighbourhood function the vector distance between nodes is used, rather than the grid distance as in SOM. SOM needs to start with a large neighbourhood so as to unfold ill initialised maps, which may result in a longer learning time and the final map may have a border effect. To deal with these problems special treatment is needed, such as using local-linear smoothing (Mulier and Cherkassky 1995), or introducing a heuristic weighting rule (Kohonen 1997, p.138). On the other hand, ESOM with straightforward node allocation and localised learning, does not manifest these problems.

In our computational model the way of node insertion is different from that of GNG. In the GNG, node insertion is basically controlled by a time factor, i.e. when ever learning time reaches the integer multiple of certain constant, a node is inserted. Thus with GNG it is easy to control the network size with respect to the learning time. Yet it is time-consuming to find the location for node insertion. GNG needs to calculate accumulated error for each network node, and insertion is occurred between the node with the maximum accumulated error and its neighbour node with the maximum accumulated error. In ESOM, however, the node insertion criterion is similar to EFuNN, only requiring a distance threshold to be specified. The control on network size is not direct. But in circumstances such as vector quantisation, we start with an estimation of intended quantisation accuracy and may not intend to put on constraint on the network size. Then it is straightforward to use the accuracy value for the node insertion criterion. Nevertheless, the distance threshold will influence the evolved network size. Like EFuNN, ESOM inserts the new node exactly as the input datum, by doing so we believe it is natural for a learning system to remember stimuli of novelty, code them, and have them adapted at later times.

Given the distance threshold in different scales, it is easy to construct hierarchical mappings with ESOM, with maps generated in multi-resolution. This will fascilate the application of ESOM in image coding and information retrieval.

2.3 Extension for supervised learning

ESOM is by nature an unsupervised learning algorithm. But just like SOM and other clustering methods it can be applied to supervised learning tasks such as classification and time series prediction. This can be done by augmenting input vectors with target output values or class labels, and using the augmented data as input to the network, an approach similar to the self-organizing semantic map (Ritter and Kohonen 1991, Bezdek and Pal 1995). By doing this, however, the output components should have comparable variance as to the input components, and sometimes standardisation of the data set is needed to meet this requirement. A diagram that represents this idea is shown in Fig.2. Given a data set with class labels or target output values, a map of the augmented data vectors can be learned using the ESOM algorithm. When testing, classification or prediction is done by comparing the input part of the augmented data vectors with the input parts of augmented prototype vectors. The output value of the best matching unit (BMU) is taken.

For classification tasks, this mechanism can be further extended using a k-Nearest Neighbour method if we take into account the contribution of the first k best matching prototypes.

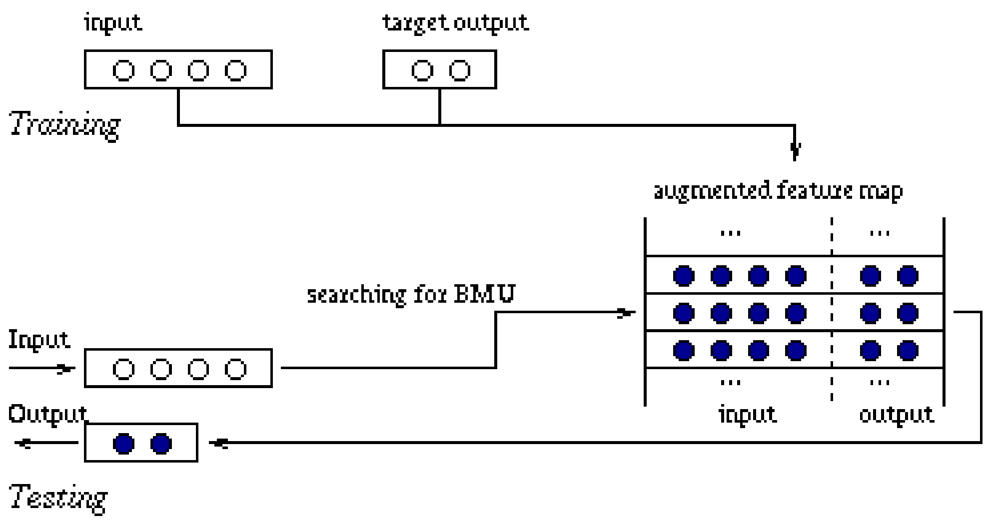

Fig. 2: ESOM extended for supervised learning by using augmented feature mapping.

2.4 Visualisation in ESOM

For data of high dimensionality, visualisation of the ESOM prototype space is in question as it is also of high dimensionality. This problem can be solved, however, using Sammon's algorithm (Sammon 1969) which projects high dimensional data into low dimensional space while keeping the distance ordering as best as possible.

The goal is to project prototype vectors $\mathbf{w}_i$, $i = 1, 2, ..., N$, onto a set of points $\mathbf{v}_i$, $i = 1, 2, ..., N$, in a two dimensional plane. Here we assume all identical prototype vectors are removed beforehand. Sammon projection tries to minimise the mapping error defined as

$$E = \frac{1}{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d(\mathbf{w}_i, \mathbf{w}_j)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{[d(\mathbf{w}_i, \mathbf{w}_j) - d(\mathbf{v}_i, \mathbf{v}_j)]^2}{d(\mathbf{w}_i, \mathbf{w}_j)} \tag{7}$$

Here $d(\cdot, \cdot)$ is the distance between two vectors. The mapping vectors are first randomly initialised and then updated using a gradient descent algorithm:

$$\Delta \mathbf{v}_i = -\alpha \frac{\delta E}{\delta \mathbf{v}_i} \frac{\delta^2 E}{\delta \mathbf{v}_i^2} \tag{8}$$

where α is a *magic factor* empirically set to 0.3 or 0.4. The close form of Eq.(8) can be found in (Sammon 1969). Typically this iteration process takes about one or two hundred steps to reach fairly good mapping for small sets of prototypes.

Compared with ESOM, the visualisation of SOM is straightforward as the prototype nodes are usually laid upon a two-dimensional lattice. The topology preserving property of SOM gurantees that similar feature vectors are projected onto close positions in the map. But this does not hold vice-visa. Feature vectors of neighbour nodes can vary significantly. To overcome this drawback, Mao and Jain (1995) proposed a NP-SOM technique to visualise the distance between map node and neighbours by using grayscale graphics. Similar approaches are also discussed in (Kaski 1997). By applying the Sammon projection, ESOM generates a two-dimensional visualisation of the prototype space with a grid structure displaying distances between network nodes. In our point of view, this approach gives more accurate and intuitive representation of the structural information in data.

3 Simulations

3.1 Data analysis and visualisation

Experiments have been done on a macroeconomic data set used in our case study for risk analysis of European Monetary Union economy (Kasabov et al. 2000), which employs a number of economic and financial indicators to predict possible shocks, and develops a computational system for analyzing and anticipating signals of abrupt changes of volatility in financial markets. Here we focus on the problem of generating a world macroeconomic map to evaluate performance and development in national and regional macroeconomy. Macroeconomic data in the period from 1994 to 1998 are collected for fifteen EMU countries, UK, US, and Asian countries such as Japan (JP) and Thailand(TH). The data are taken from the *Monthly Bulletin* of European Central Bank and from the *DataStream* on-line source. The data set has four attributes, namely annual change percentage of stock market (PCH), debt over GDP (DBT/GDP), deficit over GDP (DEF/GDP), and inflation rate. Each data entry carries a label composed by country code and time numbers, which will be used later for the generation of a labelled map.

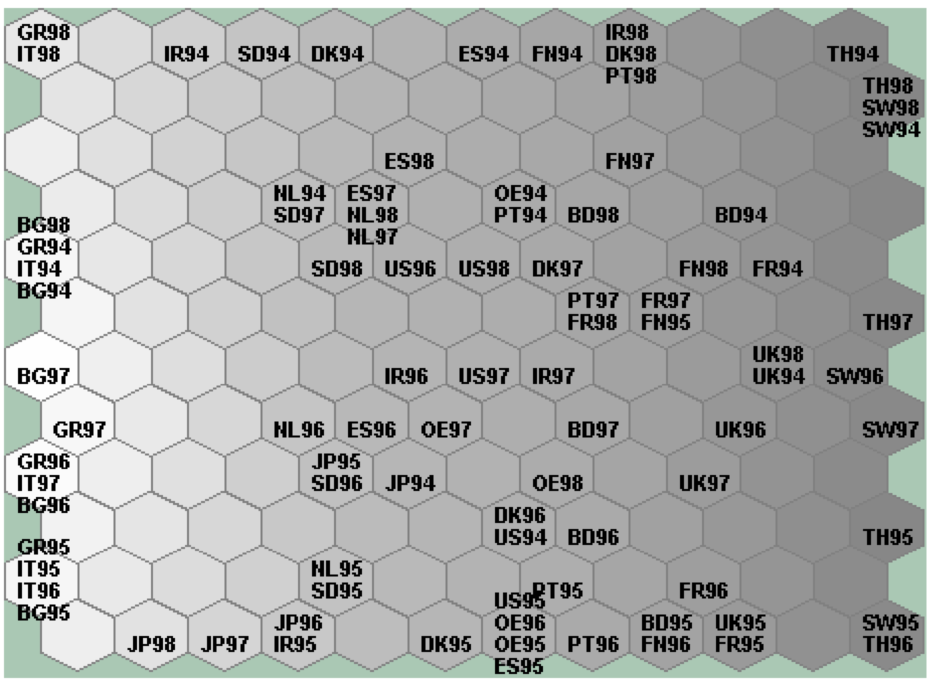

Fig. 3: The annual macroeconomic map for EMU countries etc. obtained with SOM. The debt/GDP value of each prototype vector is used to paint the corresponding map cell in grayscale.

SOM has been used in economic and financial data analysis in a number of studies, such as Serrano-Cinca (1996), Kaski (1997), and Deboeck (1999), to name but a few. Here we use the SOM algorithm to generate an annual map of macroeconomic performance and compare it with the map generated by using ESOM.

A 12×12 two-dimensional map is first trained using SOM. The size of the map is selected empirically so as to obtain a well-expanded mapping space. The map learned from the annual macroeconomic data is shown in Fig.3, where map nodes are displayed with hexagons labelled with best-matching data entries. It can be seen that an EMU cluster is formed in the central part of the map, including the following countries: OE(Austria), NL(Netherland), DK(Denmark), IR(Ireland), SD(Sweden), BD(Germany) and FR(France). Non-EMU countries UK and US also fall into this

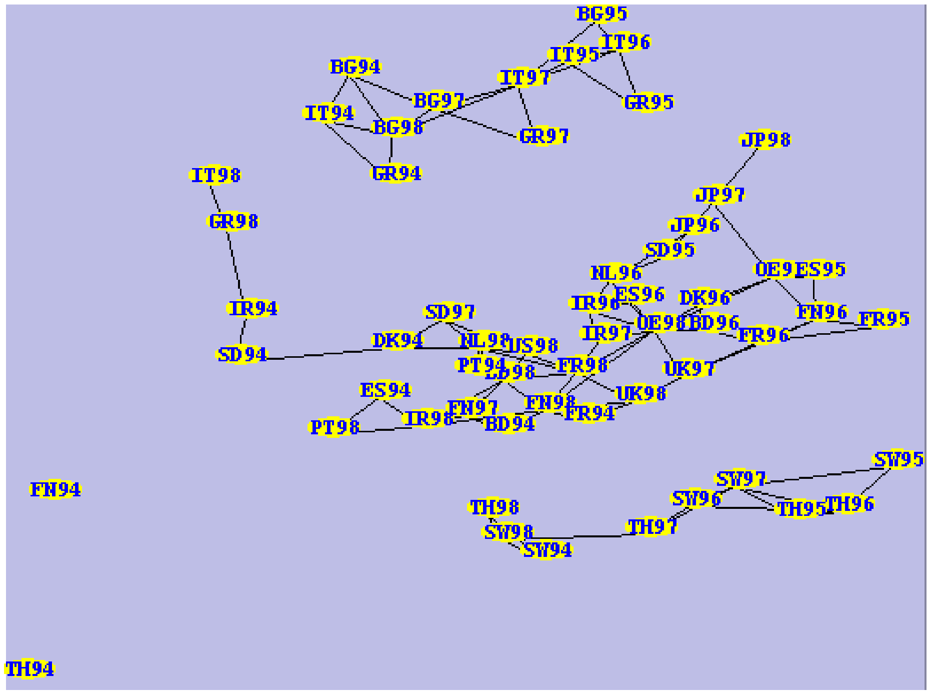

Fig. 4: The ESOM visualised with Sammon projection shows the data structure.

cluster. Four EMU countries fall out apparently: IT(Italy), BG(Belgium), GR(Greece) in area of high inflation rate on the right, and SW(Switzerland) on the left side.

By using colour palette or different gray-scale on nodes of different component values, the two dimensional map like Fig.3 presents a very useful tool to evaluate macroeconomic performance of different countries. Tracking down the movement of certain country on the map, it is also helpful to evaluate its developing trend in different years. But as we mentioned before the map distance between nodes may not match their distance in the feature space, this can be misleading. With SOM it is also difficult to find clusters visually.

Another annual map is next evolved with the same data set and shown in Fig.4. The map is first clustered using ESOM algorithm, and then projected onto a two-dimensional plane for visualisation using Sammon's algorithm. Weak connections are then clipped away. The layout of labelled nodes is quite similar to that of Fig.3, but the ESOM map gives more explicit data structure such as clusters and outliers. Here we find two major clusters, the EMU cluster with countries like FR, BD, FN, IR etc. plus UK and US, the fall-out cluster with GR, IT, and BG (from year up to 1997). It can also be seen that IT98 and GR98 are associated with the main cluster, which implies a tendency for these two economies towards the EMU cluster. The advantage of applying ESOM in this study is that it presents better visualisation quality, and it is open for further adaptation by evolving with online incoming data.

3.2 Classification of benchmark data sets

To test the classification ability of the ESOM model, we tried two data sets from the CMU Learning Benchmark Archive, so as to make comparison with previous studies such as (Fritzke 1994) and (Bruske and Sommer 1995).

Table 1: ESOM Classification Results on the Two Spirals Problem

Training						Testing		
e_{min}	n_{min}	e_{mean}	n_{mean}	σ_e	σ_n	e_{min}	e_{mean}	σ_e
0	105	0.16%	107	0.29%	2.07	0	1.5%	0.6%

3.2.1 The Two-spirals problem

This well-known benchmark generates data points from two spirals in given density. We use a training set of density 1, consisting of 194 data entries, each of which has a pair of X-Y coordinates and one class label (either 1 or -1). The testing set has a density of 4, consisting of a total of 770 data entries. Analogous to other approaches, the training and testing processes of ESOM are repeated for 20 times with the training data samples presented in random order. The performance statistics are listed in Table 1, where the following characteristics are given for both the training set and the testing set: the minimum error rate (e_{min}), the minimum number of nodes (n_{min}), mean error rate (e_{mean}), mean number of nodes (n_{min}), standard deviation of error rates (σ_e), and standard deviation of node numbers (σ_n). These results are obtained with $\epsilon = 0.9$, and $\gamma = 0.05$. With the peak performance the ESOM manages to generate a map of 105 nodes, with zero classification error on the training set and and testing set. This number of units is smaller than GCS (Fritzke 1994) and DCS-GCS (Bruske and Sommer 1995). The average performance of ESOM is also better than the DCS-GCS model as reported.

When trained with classification data sets, the goodness of prototype sets is evaluated with its discrimination ability in the input data space. This is done by categorising each map vector with a class label using the nearest-neighbour rule, operating on the distances between map vectors and data entries. Each map vector takes a class label from its best matching data entry. We show in Fig.5 how the discrimination ability develops as the map learns and converges. The spiral data set is used in both SOM and ESOM modules. In both cases we classify the data set using map vectors labeled by the nearest-neighbour rule and calculate the classification error rate during learning. Here we plot representative results from the following modules:

- Two ESOM modules. ESOM I with $\epsilon = 1.0$, $\gamma = 0.01$. After around 200 steps the network grows to 78 nodes with stable performance of an error rate around 15.46%; ESOM II with $\epsilon = 0.9$, $\gamma = 0.01$, which stabilises with 114 nodes with zero classification error rate.

- Two SOM modules. SOM I is of 12×12 size, with an initial learning rate of 0.05, and an initial neighbourhood width $N = 8$; SOM II is of 9×9 size, initial learning rate 0.05, initial neighbourhood width $N = 4$. For both modules a number of experiments are carried out with different random initial weights. Final error rates are around 22%. In this case the map size has not significant impact on the discrimination ability of the feature map.

We also tried the same data set with the LVQ_PAK package (Kohonen et al. 1996) and found that even after fine-tuning with LVQ2.1 and LVQ3, the error rates of LVQ networks of equivalent size are bigger than their ESOM counterparts, having in mind that LVQ is a supervised learning algorithm but ESOM is basically unsupervised.

The performance of compared algorithms is summerised in Table 2. Data in the upper part are taken from (Fritzke 1994) and (Bruske and Sommer 1995).

The spiral data of two dimensional space is visualised using map vectors obtained by SOM and ESOM, as shown in Fig.6. Obviously the ESOM result gives a more accurate representation of the spiral shape. The decision regions of the ESOM are displayed in Fig.7.

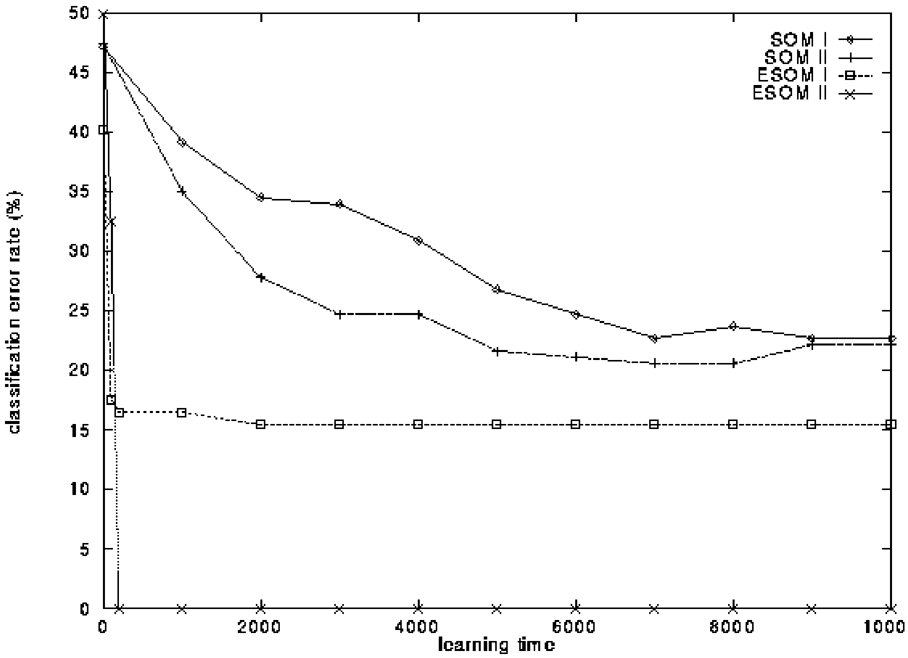

Fig. 5: Discrimination ability develops as SOM and ESOM learn the spiral data. ESOM modules learn much faster, and are more accurate.

Table 2: Comparison of Classification Performance on Two Spirals Problem

Network model	Number of units	Error rate	Number of epochs
GCS(supervised)	145	0%	180
DCS-GCS	135	0%	135
LVQ	114	11.9%	50
SOM	144	22.2%	50
ESOM	105	0%	1

3.2.2 Speaker independent vowel recognition

The vowel data set is another benchmark from the CMU collection. It consists of 990 frames of speech signals from four male and four female speakers, with 528 frames for training and the other 462 for testing. Details about speech processing procedures to construct these data vectos are given in (Robinson 1989).

Here again we present the training set in random order for 20 times and apply the trained ESOM networks on the test set. Results for offline testing are collected and the statistics is given in Table 3. For all the cases we set $\epsilon = 0.5$ and $\gamma = 0.05$ with one-pass learning. When on-line learning is applied, the ESOM is first evolved with the training data and then with the test data. The on-line classification performance of ESOM network is given in Table 4, with an overall error rate of 3.4% tested with the two data sets. Similar result is obtained when the data sets are presented in reversed order. This suggests that ESOM adapts quickly to the on-line incoming data and meanwhile gains good generalisation ability.

In Table 5 the results of different algorithms tested with the vowel recognition problem are listed. Results from algorithms other than ESOM are taken from (Bruske and Sommer 1995), referencing works such as (Robinson 1989) and (Fritzke 1993). ESOM performance is comparable to GCS and

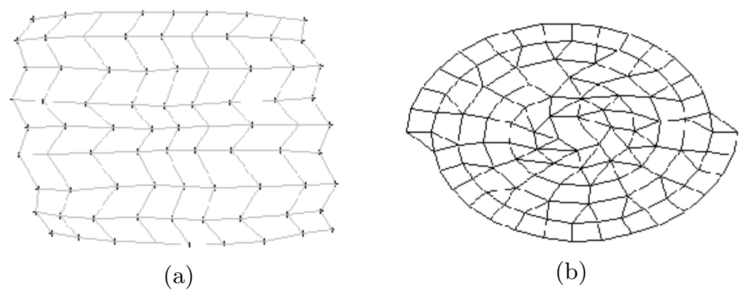

Fig. 6: Visualisation of the spiral data. (a) The SOM grids after training; (b) The ESOM nodes with trimmed connections.

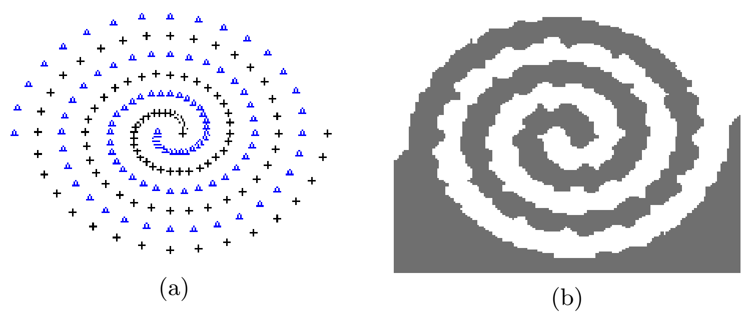

Fig. 7: (a) Data points of the two spirals; (b) decision regions for ESOM.

Table 3: ESOM Offline Classification Performance on Vowel Recognition Problem.

e_{min}	n_{min}	e_{mean}	n_{mean}	σ_e	σ_n
35.5	264	38.2	275.9	1.3%	3.9%

Table 4: On-line classification performance

Learning set I	(training set)	(testing set)
Number of samples	582	462
Number of nodes	286	242
Classification errors	31(5.3%)	24(5.2%)
Learning set II	(testing set)	(training set)
Number of samples	462	582
Number of nodes	525	526
Classification errors	4(0.9%)	9(1.5%)
Overall on-line classification error rate	3.4%	3.2%

Table 5: Performance Comparison on Vowel Recognition Problem.

Classifier	Hidden units	Recognition rate
Single layer perceptron		33
Multilayer perceptron	88	51
Modified Kanerva model	528	50
Radial basis function	528	53
Gaussian node network	528	55
Square node network	88	55
Nearest-neighbor		56
5D GCS	135	66
DCS-GCS	108	65
ESOM	279	65

Table 6: Prediction Performance on the Mackay-Glass Data.

Method	Units	On-line NRMSE
Neural gas	1000	0.062
RAN (Rosipal)	113	0.373
RAN-GQRD	24	0.170
SOM + linear regression	1225	0.022
ESOM	114	0.320
ESOM	1000	0.044

DCS-GCS with a larger number of nodes. The learning time, however, is much shorter. Here ESOM does a one-pass learning, while GCS needs 80 epochs (Fritzke 1994). The learning time of DCS-GCS is not given, but since of the similarity between GCS and DCS-GCS, the numbers of epochs for training may not differ too much.

3.3 Time series modelling

Following previous studies on time series modelling with the use of neural network models, here we apply ESOM to predict the Mackay-Glass time series, also from the CMU Machine Learning Benchmark. The data sets are generated from the benchmark using embedded data vectors consisting of four values of the time series:

$$\mathbf{x}(n) = [x(n), x(n-6), x(n-12), x(n-18)]^T$$

Each input vector relates to an output value of x(n+85), i.e., we train the network to predict the value at time n+85. A training set of 3000 samples (from $n = 200$ to 3200) and a testing set of 500 samples (from $n = 5000$ to 5500)are used.

After training, the prediction results of ESOM networks are comparable to the results of other constructive models, such as neural gas (Martinetz 1993) and RAN (Rosipal 1997). The SOM plus linear regression model (Vesanto 1995) gives the best result, but since it stores local data sets, the complexity scale of the model is much larger than the number of units. An ESOM network of 1000 nodes is obtained by setting the sensitivity threshold of $\epsilon = 0.03$. The normalised RMS error (NRMSE) is 0.044, after on-line testing on the 500 test samples. We believe this is a good result for a one-pass on-line learning process.

4 Conclusion

This paper introduces an evolving self-organizing map (ESOM) as an evolving variation of the Kohonen SOM, featuring its one-pass on-line learning ability, a feature map of less geometric constraint, good topology representation, and good prototyping accuracy.

Results from benchmark study have shown that ESOM is an effective computational model for on-line data clustering, dynamic data analysis, and scientific visualisation. The supervised extension of ESOM also works well for classification problems, achieving good classification results in one-pass learning. For challenging problems such as chaotic time series prediction, ESOM surprisingly achieved comparable accuracy using a relatively simple algorithm, but it needs a network size bigger than that of supervised incremental learning algorithms such as RAN. Further improvement of the ESOM computational model can be achieved, by applying an additional aggregation procedure to tune system parameters and local receptive fields and to reduce network size. Applications for information retrieval and on-line web computing (Lawrence and Giles 1998) are also anticipated.

Acknowledgements

This work is supported by the Foundation for Research, Science and Technology (FRST), New Zealand, under grant UOO-808.

Unless otherwise specified, all algorithms used in this study are implemented within the Repository of Connectionist-Based Intelligent Systems (RICBIS), which is on-line available on our site at URL http://www.otago.ac.nz/informationscience/kel/CBIIS/cbiis-ricbis.html.

References

Bezdek J.C., Pal N.R. 1995. A note on self-organizing semantic maps. *IEEE Trans. on Neural Networks*, 6(5):1029-1036.

Blackmore J., and Miikkulainen R. 1993. Incremental Grid Growing: Encoding High-Dimensional Structure into a Two-Dimensional Feature Map, *Proc. ICNN'93*, Int. Conf. on Neural Networks, Vol. I, 450-455, IEEE Service Center.

Bruske, J., and Sommer G. 1995. Dynamic cell structure learns perfectly topology preserving map. *Neural Comp.*, **7**, 845-865.

Deboeck, G. 1999. Investment maps for emerging markets. *Neuro-fuzzy techniques for intelligent information systems* (N.Kasabov and R.Kozma Eds.). Physica Verlag, 373-395.

Fritzke, B. 1991. Unsupervised clustering with growing cell structures. *Proc. IJCNN 91*, 531-536.

Fritzke, B. 1994. Growing cell structures - a self-organizing network for unsupervised and supervised learning. *Neural Networks*, **7**, 1441-1460.

Fritzke, B. 1995. A growing neural gas network learns topologies, in *Advances in neural information processing Systems* , D. Touretzky, T.K. Keen eds., pp.625-632, Cambridge MA: MIT Press.

Heinke, D., and Hamker, F.H. 1998. Comparing neural networks: a benchmark on growing neural gas, growing cell structures, and fuzzy ARTMAP. *IEEE Trans. on Neural Networks*, **9**, 1279-1291.

Heskes, T.M., and Kappen B. 1991. Learning processes in neural networks. *Physical Review A*, **44**, 2718-2726.

Kadirkamanathan, V., Niranjan, M. 1993. A function estimation approach to sequential learning with neural networks, *Neural Comp.*, **5**, 954-975.

Kasabov, N. 1998a. The ECOS framework and the ECO learning method for evolving connectionist systems. *Jour. of Advanced Computational Intelligence*, **2**, 1-8.

Kasabov, N. 1998b. Evolving fuzzy neural networks - algorithms, applications and biological motivation. in: Yamakawa and Matsumoto (Eds.) *Methodologies for the Conception, Design, and Application of Soft Computing*, World Scientific, 271-274.

Kasabov, N., Erzegoveri, L. et. al. 2000. Hybrid intelligent decision support systems and applications for risk analysis and prediction of evolving economic clusters in Europe. To appear in N.Kasabov (ed), *Future Directions for Intelligent Systems and Information Sciences*, Physica Verlag (Springer Verlag).

Kaski, S. 1997. Data exploration using self-organizing maps. Mathematics, Computing and Management in Engineering Series No. 82, Acta Polytechnica Scandinavica.

Kohonen T. 1982. Self-organizing formation of topologically correct feature maps, *Biological Cybernetics*, v. 43, 59-69.

Kohonen, T., Hynninen, J., Kangas, J. et al. 1996. LVQ_PAK: The learning vector quantization program package, Report A30, Laboratory of Computer and Information Science, Helsinki University of Technology.

Kohonen, T. 1997. *Self-Organizing Maps*, second edition, Springer.

Lawrence, S., and Giles, C.L. 1998. Searching the world wide web, *Science*, Apr. 3, 1998, pp.98-100.

Mao J., and Jain, A.K. 1995. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. on Neural Networks*, **6**, 296-317.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, *Proc. 5th Berkeley Symp. on Mathematics*, L.M. LeCam and J. Neyman eds., pp.281-297.

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

Meyering, A. and Ritter H. 1992. Learning 3d-shape-perception with local linear maps, in *Proc. of IJCNN 92*, pp.IV:432-436, Baltimore.

Mulier, F. and Cherkassky, V. 1995. Self-organization as an interative kernel smoothing process, *Neural Comp.*, **7**, 1165-1177.

Nowlan, S.J. 1990. Maximum likelihood competitive learning, in *Advances in Neural Information Processing Systems 2*, D. Touretzky ed., pp.574-582, New York: Morgan Kauffman.

Platt, J. 1991. A resource-allocating network for function interpolation. *Neural Comp.*, **3**, 213-225.

Ritter H., Kohonen T. 1989. Self-organizing semantic maps, *Biological Cybernetics*, **61**, 241-254.

Robinson A.J. 1989. Dynamic error propagation networks. Ph.D. thesis, Cambridge University.

Rosipal R., Koska M., and Farkas, I. 1998. Prediction of chaotic time-series with a resource-allocating RBF network. *Neural Processing Letters*, **7**, 185-197.

Sammon, Jr., J. 1969. A non-linear mapping for data structure analysis. *IEEE Trans. on Computers*, **18**, 401-09.

Serrano-Cinca, C. 1996. Self organizing neural networks for financial diagnosis. *Decision Support Systems*, **17**, 227-38.

Schaal, S., & Atkeson, C.G. 1998. Constructive incremental learning from only local information. *Neural Comp.*, 10, 2047-2084.

Vesanto, J. 1997. Using the SOM and local models in time-series prediction. *Proc. of WSOM'97*, pp.209-214. Helsinki University of Technology, Neural Networks Research Centre, Finland.