ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 117-122

Comparison Of Kohonen, Scale-Invariant and GTM Self-organising Maps for interpretation of Spectral Data

Donald MacDonald, Stephen McGlinchey John Kawala and Colin Fyfe

Department of Computing and Information Systems
Applied Computational Intelligence Research Unit
The University of Paisley
Scotland
e-mail: mcdo-ci0,mcgl-ci0,kawa-ci0,fyfe-ci0@paisley.ac.uk

ABSTRACT

We investigate the use of artificial neural networks in classifying hyperspectral data. Such data when collected from remote sensors provides extremely detailed coverage of e.g. the mineralogical composition of planetary surfaces, however the volume of data supplied often overwhelms traditional classifiers. When we wish to investigate such data sets in an open-ended manner, the use of unsupervised learning is a pre-requisite. A set of remotely sensed spectral images are use to train several different topology preserving neural networks. In each method, the data is projected onto a two dimensional grid designed to visualise the data set in a low dimensional space. Such mappings allow graceful degradation of the classifications given by the mappings since nearby data points are mapped to the same or similar classifications.

Introduction

We investigate the use of three unsupervised techniques as classifiers of astronomical data. The data consists of 65 colour spectra of 115 asteroids used by [1]. We compare the relative performance of three networks on this dataset. Due to the high dimensionality of the spectral data, the classification can result in several different 2-d mappings; this will always occur when mapping high dimensional data onto a lower dimension and is determined by the interaction between the non-linearity and network's initial starting conditions.

To assess the effectiveness of the mappings, we use only the 12 classes proposed by Tholen [2], which has given a different classification from that of Howell et al. [1]. Each of the different techniques used also produces a different mapping. We compare the use of the following three different self-organising artificial neural networks to classify the spectral data while preserving topological relationships:

1. The Kohonen SOM [3] has previously been shown to be useful in mapping such data sets [1].
2. The Scale Invariant Feature Map [4] has been shown to ignore the magnitude of signals and organise on the basis of the distribution of directions of the input data
3. The Generative Topographic Mapping, (GTM) [5], which has been developed as a "principled alternative to the SOM".

Kohonen’s Self-Organising Feature Map

Kohonen’s algorithm consists of a single layer network where competition between output neurons determines which weights are updated. In the SOM not only are the weights attached to the winning neuron, c, updated but weights of nodes that are neighbours to the winning node are also updated

$$\Delta w_{ij} = \eta(x_j - w_{ij})\Lambda(i,c)$$

using the neighbourhood function $\Lambda(i, c)$ of the winning neuron *c*. The neighbourhood function is a monotonically decreasing function of the distance between *i* and *c*. Normally this function is a difference of Gaussians. This results in the network trying to get all of its neurons to cover the input space which can result in neurons that do not cover any data points at all.

The weight parameters of the SOM are normally initialised to small random values, however in this study we have initialised the weights vectors to lie on a hyperplane determined by the first two principal components of the data. This method has already been used by Bishop et al. with the GTM and it is our finding that convergence of the SOM was more consistent and seemed to provide better discrimination than with random initialisation.

Scale Invariant Map

We have previously introduced a network [6] that self-organises to find a mapping of the input data that preserves neighbourhood relations, but the difference between this mapping and the SOM is that the mapping is scale invariant. When the SOM is trained it approximates a Voronoi tessellation of the input space. The scale invariant map, however, creates a mapping where each neuron captures a “pie slice” of the data according to the angular distribution of the input data (see Figure 1).

Consider a network with n dimensional input data and having m output neurons and a $m \times n$ weight vector that is initialised to small random values. A competition takes place between all outputs. The neuron whose weights have the smallest angle with the input vector wins. Let the winning neuron be the cth.

$$c = \arg\min_i(\theta_i) \quad \text{where} \quad \cos(\theta_i) = \frac{\mathbf{x}.\mathbf{w}_i}{\|\mathbf{x}\|.\|\mathbf{w}_i\|}$$

Now the winning neuron excites those neurons close to it, i.e. we have a neighbourhood function $\Lambda(c, j)$ which satisfies $\Lambda(c, j) \le \Lambda(c,k)$ for all $j,k : \| c - j \| \ge \| c - k \|$ where $\|.\|$ is the Euclidean norm. For this experiment the neighbourhood function is a Gaussian whose variance is decreased during the course of training. Then simple Hebbian learning gives:

$$\begin{aligned}\delta w_{ij} &= \eta_t \Lambda(c,i).x_j(t+1) \\ &= \eta_t \Lambda(c,i).(x_j(t) - w_{cj})\end{aligned}$$

where we have used $x_j(t)$ as the activation of the j^{th} input neuron at time t and w_{ij} is the weight between this and the i^{th} output neuron. Note that every weight vector is updated in the direction of the vector $\mathbf{x} - \mathbf{w}_c$ by a magnitude dependant on the neighbourhood function.

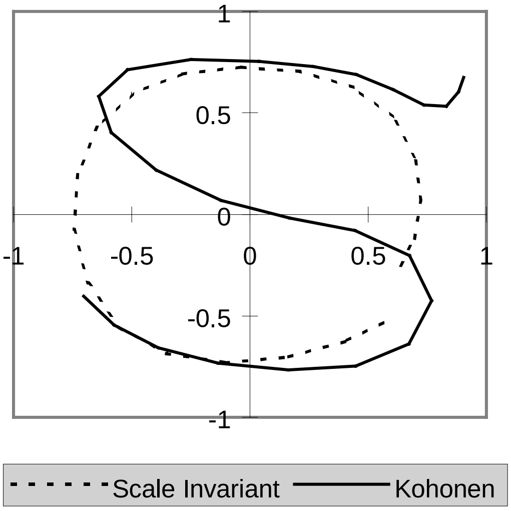

Figure 1: A one-dimensional scale invariant feature map and a Kohonen map on the data drawn from a uniform distribution.

Generative Topographic Mapping

The GTM was introduced in 1996 as a more principled alternative to the SOM. Like the SOM, the GTM consists of a regular lattice of K nodes arranged in a latent space that normally has fewer dimensions than the data space. The latent space is typically one or two-dimensional. These nodes are mapped into data space to give a non-linear manifold arranged in the high-dimensional space. With the SOM, we normally consider a data vector, $\mathbf{x}$, to be coded as a qualitative value, c, which is given by the position of the winning node. However, with the GTM, a data vector is presented to the network and a responsibility for the observed input (posterior probability) is calculated for each node in the lattice. Therefore, a data vector stimulates a posterior distribution across the whole map – not just a single winning node. It is still possible to map data to a single point in latent space by considering, for example, the mean or the mode of the posterior. During training, the responsibilities are used to update the weights in order to maximise the expectation of the observed input using the EM algorithm.

In the case of the SOM, the mapping from latent to data space $d(\mathbf{y}_i;\mathbf{W})$ consists of selecting a D-dimensional weight vector corresponding to each node, but with the GTM, there is a two-stage mapping. Firstly a set of M basis functions $\phi_j(\mathbf{y})$ fixed in the latent space is used to map points (the nodes of the lattice) from L-space to M-space, and secondly, a weight matrix $\mathbf{W}$ maps from basis space to data space $\mathbf{W}\phi(\mathbf{y})$. The activation of each basis node is determined by a function of the Euclidean distance between the latent space point and the centre of the basis function. This is similar to the first layer of a radial basis function network. The centres of the basis functions are usually distributed uniformly amongst the nodes of the lattice. In our experiments, we used Gaussian basis functions as described below, although it is possible to use other probability distributions. When a point is mapped from latent to data space a probability density is generated in data space using a noise model determined by the basis functions.

$$p(\mathbf{x}_n \mid \mathbf{y}_i, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\left\|d(\mathbf{y};\mathbf{W}) - \mathbf{x}\right\|^2\right\}$$

(where β is the inverse variance of the noise model.)
After all latent space nodes have generated distributions in the data space, the observed data vector, $\mathbf{x}$, is then used to evaluate a responsibility for each node in the latent space lattice. This calculation completes the E-step.

$$\begin{aligned} R_{in}(\mathbf{W}_{old}, \beta_{old}) &= p(\mathbf{y}_i \mid \mathbf{x}_n, \mathbf{W}_{old}, \beta_{old}) \\ &= \frac{p(\mathbf{x}_n \mid \mathbf{y}_i, \mathbf{W}_{old}, \beta_{old})}{\sum_{i'=1}^{K} p(\mathbf{x}_n \mid \mathbf{y}_{i'}, \mathbf{W}_{old}, \beta_{old})} \end{aligned}$$

During the M-step a system of linear equations is solved to calculate a new weight matrix using

$$\mathbf{\Phi}^{\mathbf{T}}\mathbf{G}_{old}\mathbf{\Phi}\mathbf{W}_{new}^{\mathbf{T}} = \mathbf{\Phi}^{\mathbf{T}}\mathbf{R}_{old}\mathbf{X}$$

where $\mathbf{\Phi}$ is a K×M matrix with elements $\phi_{ij} = \phi(\mathbf{y}_i)$, $\mathbf{X}$ is the matrix of training vectors and $\mathbf{G}$ is a diagonal K×K matrix with elements $\mathbf{G}_{ii}$ being a sum of responsibilities of lattice node i over all data vectors in the training set. A new inverse noise parameter, β, is given by

$$\frac{1}{\beta_{new}} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{i=1}^{K} R_{in}(\mathbf{W}_{old}, \beta_{old})\left\|\mathbf{W}_{new}\phi(\mathbf{y}_i) - \mathbf{x}_n\right\|^2 .$$

There are several advantages to be gained over the SOM: it is a principled method with a well-defined cost function and convergence can be proven; the cost function gives a good indication of the degree of convergence during training and it is therefore useful as a stopping criterion; the GTM has few parameters that require to be chosen by the user, and it is not necessary to select ad hoc parameters for specific data sets; topology preservation is always guaranteed with the GTM due to its smooth mapping function $d(\mathbf{y}_i;\mathbf{W})$ whereas the SOM is topology preserving only after self-organisation has taken place.

Results

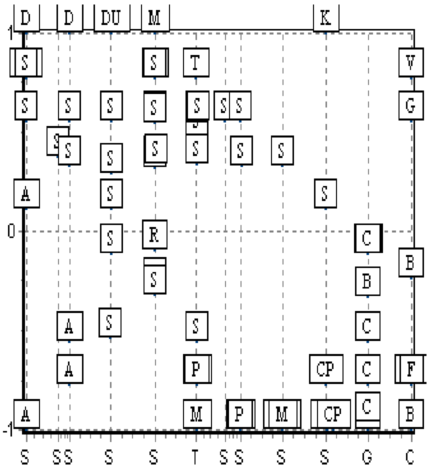

Figure 2 **GTM**
Projection of the data set onto the GTM space

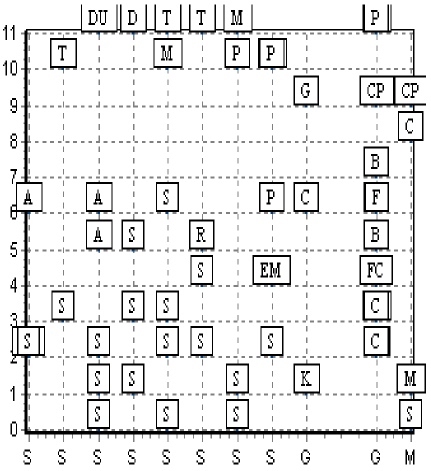

Figure 3 **Scale Invariant**
Projection of the data set onto the Scale Invariant mapping.

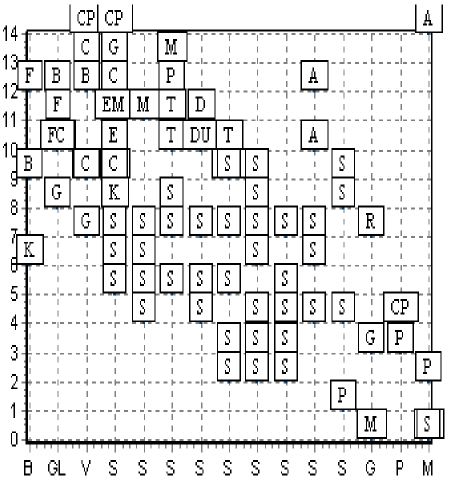

Figure 4 : SOM

Projection of the data set onto the converged SOM space

The results show that all three maps are able to classify the data. The interesting points are the differences in the classification. The first thing to note is that the GTM unlike the other two maps is capable of non-integer mappings. Visually the GTM has achieved better cluster separation, whereas the SOM has mapped the data points in a diagonal where the large areas of the edges of the mapping are redundant. The SOM was also prone to over-train on the data; the decision as to when to stop training the SOM is very subjective. The SOM has mispositioned an S type asteroid at position 14,0 (S type 138 tolosa is positioned on top of 2 M types, 92 undina and 849 ara). The GTM has not made the same classification errors for the S type asteroids. The errors that have occurred with the GTM are on the border where asteroids with similar spectra overlap. This is because the GTM has a continuous mapping function unlike the SOM, which quantises similar data vectors to the same point on the map.

308 polyxo is a type T asteroid that has been misclassified by all three networks since it has different spectra from the other type T asteroids. The misclassification is unavoidable and most pronounced in the scale invariant map as the map classifies on orientation and not magnitude.

Each of the three networks were also tested on data that was excluded from the training set. In each case, classification was correct with the exception of a T asteroid. This asteroid (308 polyxo) was also incorrectly classified when it was included in the training set and therefore this misclassification should be expected. The other correct classifications show that each of the three algorithms have successfully produced general classifiers.

Conclusions

Using visual inspection, the network that had the best performance on this data was the GTM, which had the fewest misplacements (where 1 node was used for two types of asteroids).

We have also compared the mappings with the new classes derived by Howell et al [1]; with the parameter set used above, the difference between the SOM and the GTM was negligible. However, it is our experience that a small change of parameter values can lead to substantially different mappings with all three methods. This suggests that this problem is more usefully tackled using a data mining method in which the interaction between skilled human expert and exploratory neural method will lead to the best results.

The GTM is a probabilistic method which provides a pdf of responsibilities. We are currently investigating methods of combining such a pdf with the more discrete decisions of the other two networks in order to perform robust classification.

References

[1] E. Merenyi, Self-organising ANNs for Planetary Surface Composition Research.*Journal of Geophysical Research*, 99:E5, 10847-10865, 1994.

[2] D. J. Tholen, Asteroid Taxonomy from cluster analysis of photomerty, Ph.D. dissertation, University of Arizona, Tucson, 1984

[3] T. Kohonen, Self-Organising Maps, Springer,1995.

[4] C. Fyfe. A scale invariant feature map, *Network:Computation in Neural Systems*,7:269-275,1996.

[5] C. Bishop. M. Svensen C. Williams, GTM: The Generative Topographic Mapping, Technical Report NCRG 96-015, Aston University.

[6] S. McGlinchey C. Fyfe An Angular Quantising SOM for Scale Invariant Classification, Workshop on Self-Organising Maps, Helsinki, June 1997