Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp508-513)

Tight Clusters and Smooth Manifolds with the Harmonic Topographic Map.

MARIAN PEÑA AND COLIN FYFE
Applied Computational Intelligence Research Unit,
The University of Paisley,
Paisley, PA1 2BE
SCOTLAND.

Abstract We review a new form of self-organizing map introduced in [5] which is based on a non-linear projection of latent points into data space, identical to that performed in the Generative Topographic Mapping (GTM) [1]. We discuss a refinement of that mapping (M-HaToM) and show on real and artificial data how it both finds the true manifold on which a data set lies and also clusters data more tightly than the previous algorithm (D-HaToM).

Key-words: Smooth manifold identification, Tight Clustering, Topographic maps.

1 Introduction

Recently [5], we introduced a new topology preserving mapping which we called the Harmonic Topographic Map (HaToM). Based on a generative model of the experts, we showed how a topology preserving mapping could be created from a product of experts in a manner very similar to that used by Bishop *et al* [1] to convert a mixture of experts to the Generative Topographic Mapping (GTM).

A topographic mapping of a data set is a mapping which retains some property of the data set in an ordered manner. For example, in the visual cortex, we have neurons which have optimal response to different orientation of bars. Crucially, however, as we traverse part of the cortex, the optimal orientation changes smoothly and gradually: nearby neurons respond optimally to similar orientations. Topographic mappings are rather ubiquitous in the cortex, appearing for example in the visual, auditory, somatosensory and motor cortex. In this paper, we discuss a new method of finding topographic mappings.

The underlying method uses a (one or two dimensional) latent space of K points, $t_1, ..., t_K$ which are mapped through a set of Gaussian basis functions, $\Phi()$, to a feature space which is subsequently mapped via a matrix W to the data space. The algorithm in [2] maximised the likelihood of the data under a probabilistic model based on the mapping of these latent points. It is readily shown that this method is equivalent to minimising

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K} \parallel \mathbf{x}_i - W\Phi(t_k) \parallel^2 r_{ik} \qquad (1)$$

$$= \sum_{i=1}^{N}\sum_{k=1}^{K} \parallel \mathbf{x}_i - \mathbf{m}_k) \parallel^2 r_{ik}$$

where $\mathbf{m}_k$ is the centre in data space determined by the k^{th} latent point and r_{ik} is the responsibility of the k^{th} latent point for the i^{th} data point. In this paper, we extend the HaToM algorithm so that, when used for cluster identification, the clusters it finds are more tightly defined while, if it is used for manifold identification, the manifold is more robustly identified.

2 Harmonic Averages

Harmonic Means or Harmonic Averages are defined for spaces of derivatives. For example, if

you travel $\frac{1}{2}$ of a journey at 10 km/hour and the other $\frac{1}{2}$ at 20 km/hour, your total time taken is $\frac{d}{10} + \frac{d}{20}$ and so the average speed is $\frac{2d}{\frac{d}{10}+\frac{d}{20}} = \frac{2}{\frac{1}{10}+\frac{1}{20}}$. In general, the Harmonic Average of K points, $a_1, ..., a_K$, is defined as

$$HA(\{a_i, i = 1, \cdots, K\}) = \frac{K}{\sum_{k=1}^{K} \frac{1}{a_k}} \quad (2)$$

2.1 Harmonic K-Means

This has recently [7, 6] been used to make the K-means algorithm more robust. The K-Means algorithm [3] is a well-known clustering algorithm in which N data points are allocated to K means which are positioned in data space. The algorithm is known to be dependent on its initialization: a poor set of initial positions for the means will cause convergence to a poor final clustering. [7, 6] have developed an algorithm based on the Harmonic Average which converges to a better solution than the standard algorithm.

The algorithm calculates the Euclidean distance between the i^{th} data point and the k^{th} centre as $d(\mathbf{x}_i, \mathbf{m}_k)$. Then the performance function using Harmonic averages seeks to minimize

$$Perf_{HA} = \sum_{i=1}^{N} \frac{K}{\sum_{k=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_k)^2}} \quad (3)$$

Then we wish to move the centres using gradient descent on this performance function

$$\frac{\partial Perf_{HA}}{\partial \mathbf{m}_k} \quad (4)$$
$$= -K \sum_{i=1}^{N} \frac{4(\mathbf{x}_i - \mathbf{m}_k)}{d(\mathbf{x}_i, \mathbf{m}_k)^4 \{\sum_{l=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_l)^2}\}^2}$$

Setting this equal to 0 and "solving" for the $\mathbf{m}_k$'s, we get a recursive formula

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^4 (\sum_{l=1}^{K} \frac{1}{d_{i,l}^2})^2} \mathbf{x}_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^4 (\sum_{l=1}^{K} \frac{1}{d_{i,l}^2})^2}} \quad (5)$$

where we have used $d_{i,k}$ for $d(\mathbf{x}_i, \mathbf{m}_k)$ to simplify the notation. There are some practical issues to deal with in the implementation, details of which are given in [7, 6].

[7] have extensive simulations showing that this algorithm converges to a better solution (less prone to finding a local minimum because of poor initialisation) than both standard K-means or a mixture of experts trained using the EM algorithm.

2.2 The Harmonic Topographic Map

With this learning rule on the latent space model of Section 1, we get a mapping which has elements of topology preservation but which often exhibits twists, such as are well-known in the SOM [4]. In [5] we opted to begin with a small value of K (for one dimensional latent spaces, K=2, for two dimensional latent spaces and a square grid, K=2*2) and grew the mapping. We do not randomise W each time we augment K. The current value of W is approximately correct and so we need only to continue training from this current value. Also we use a pseudo-inverse method for the calculation of W from the positions of the centres. In [5] we leave the data to control the changes while the algorithm iteratively recalculate the centres (see below), and only when we add a new latent point K, do we update the W and project the $\mathbf{m}_k$ centres into data space with it; so it is a more data-driven algorithm (D-HaTom). The algorithm is

1. Initialise K to 2. Initialise the W weights randomly and spread the centres of the M basis functions uniformly in latent space.

2. Initialise the K latent points uniformly in latent space.

3. Calculate the projection of the latent points to data space. This gives the K centres, $\mathbf{m}_k$. Set count=0.

   (a) For every data point, $\mathbf{x}_i$, calculate $d_{i,k} = ||\mathbf{x}_i - \mathbf{m}_k||$.

(b) Recalculate means using (5).

(c) If count<MAXCOUNT, count= count +1 and return to 3a

4. Recalculate W using $(\Phi^T\Phi + \delta I)^{-1}\Phi^T\mathbf{\Xi}$ where $\mathbf{\Xi}$ is the matrix containing the K centres, I is identity matrix and δ is a small constant, necessary because initially $K < M$ and so the matrix $\Phi^T\Phi$ is singular.

5. If $K < K_{max}$, $K = K + 1$ and return to 2.

In the simulations in this paper, MAXCOUNT was set at 20. Figure 1 shows the result of a simulation in which we have 20 latent points deemed to be equally spaced in a one dimensional latent space, passed through 5 Gaussian basis functions and then mapped to the data space by the linear mapping W. We generated 60 two dimensional data points, (x_1, x_2), from the function $x_2 = x_1 + 1.25\sin(x_1) + \mu$ where μ is noise from a uniform distribution in [0,1]. We see that, for a sufficiently small number of latent points, the one dimensional nature of the data set is revealed but when the number of latent points exceeds 15, the manifold found begins to wander across the true manifold. The M-HaToM algorithm corrects this. Similar results can be achieved using an underlying two dimensional latent space.

3 An improved algorithm

However, since

$$\begin{aligned}\frac{\partial Perf_{HA}}{\partial W} &= \sum_{k=1}^{K}\frac{\partial Perf_{HA}}{\partial \mathbf{m}_k}\frac{\partial \mathbf{m}_k}{\partial \mathbf{W}} \\ &= \sum_{k=1}^{K}\frac{\partial Perf_{HA}}{\partial \mathbf{m}_k}\Phi_k \qquad (6)\end{aligned}$$

we can use the algorithm directly in a learning rule.

The Model-driven algorithm (M-HaToM) is

1. Initialise K to 2. Initialise the W weights randomly and spread the centres of the M basis functions uniformly in latent space.

2. Initialise the K latent points uniformly in latent space. Set count=0.

3. Calculate the projection of the latent points to data space. This gives the K centres, $\mathbf{m}_k = \phi_k^T W$.

4. For every data point, $\mathbf{x}_i$, calculate $d_{i,k} = ||\mathbf{x}_i - \mathbf{m}_k||$.

5. Recalculate centres using (5).

6. Recalculate W using

$$W = \begin{cases} (\Phi^T\Phi + \delta I)^{-1}\Phi^T\mathbf{\Xi} & \text{if } K < M \\ (\Phi^T\Phi)^{-1}\Phi^T\mathbf{\Xi} & \text{if } K \geq M \end{cases} \qquad (7)$$

7. If count<MAXCOUNT, count = count +1 and return to 3

8. If $K < K_{max}$, $K = K + 1$ and return to 2.

This is a model-driven algorithm (M-HaToM) that forces the data to follow the model continuously (i.e. calculating W and $\mathbf{m}_k$ inside the loop), so that the manifold is smoother and gets tighter clustering as we will see with the experiments below.

4 Simulations

In this section we review the examples seen in [5] and above, comparing the performance of the two HaToM algorithms.

4.1 1D Artificial Data

Figure 2 shows how the M-HaToM algorithm solves the problem of the D-HaToM, i.e. we can increment the number of latent points as long as we want, without losing the manifold shape. The reason for the creation of the smooth manifold compared to the M-HaToM algorithm is twofold:

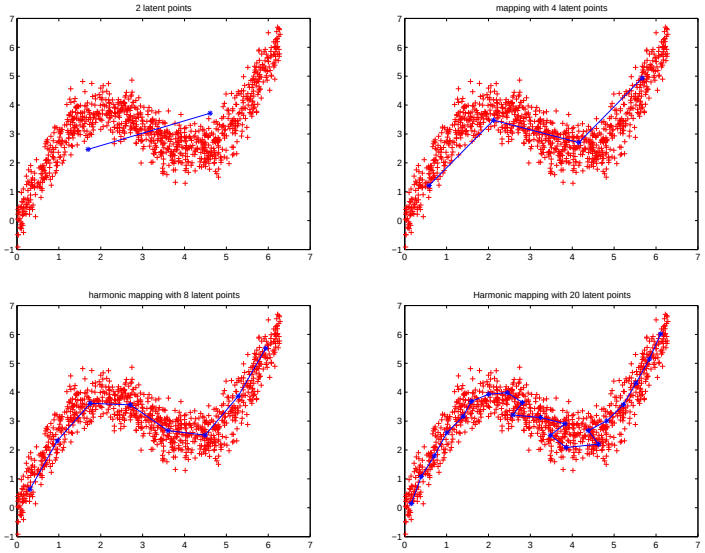

Figure 1: The D-HaToM mappings with 2, 4, 8 and 20 latent points.

1. The δI is a regularising term which ensures that the manifold does not wander about the data space but sticks closely to the manifold.

2. However, even when we remove this term (for $K \geq M$), the regularisation continues since we are compressing the reconstruction of $\mathbf{\Xi}$ into M dimensions:

$$\begin{aligned} \mathbf{\Xi} &= \Phi^T W \\ \text{where } W &= (\Phi\Phi^T)^{-1}\Phi\Xi \\ \text{Therefore } \mathbf{\Xi} &= \Phi^T(\Phi\Phi^T)^{-1}\Phi\Xi \end{aligned}$$

4.2 The Algae data set

This is a set of 118 samples from a scientific study of various forms of algae some of which have been manually identified. Each sample is recorded as an 18 dimensional vector representing the magnitudes of various pigments. 72 samples have been identified as belonging to a specific class of algae which are labeled from 1 to 9. 46 samples have yet to be classified and these are labeled 0.The M-HaToM algorithm has given a much better clustering of the algae data. Figure 4 shows a clustering with a complete separation of the different classes. We lose however the spread of the data where different subclasses seems to appear given by the D-HaToM algorithm (see Figure 3).

5 Conclusion

We have shown how updating the weights which determine the centres of a topographic mapping within the algorithm has both a smoothing property which allows a smooth manifold to be discovered in a data set and a clustering property which enables the identification of very tight clusters in a data set. We anticipate that future work will continue to compare these two algorithms over a variety of properties.

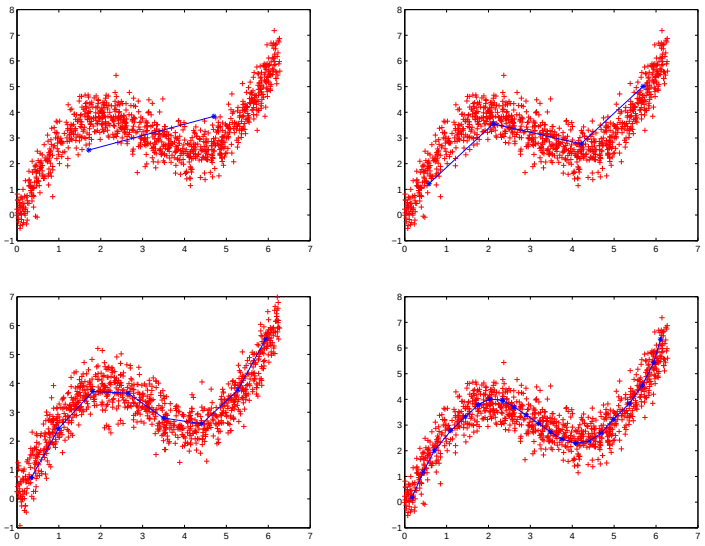

Figure 2: The M-HaToM mappings with 2, 4, 8 and 20 latent points. All the latent points stay in the manifold

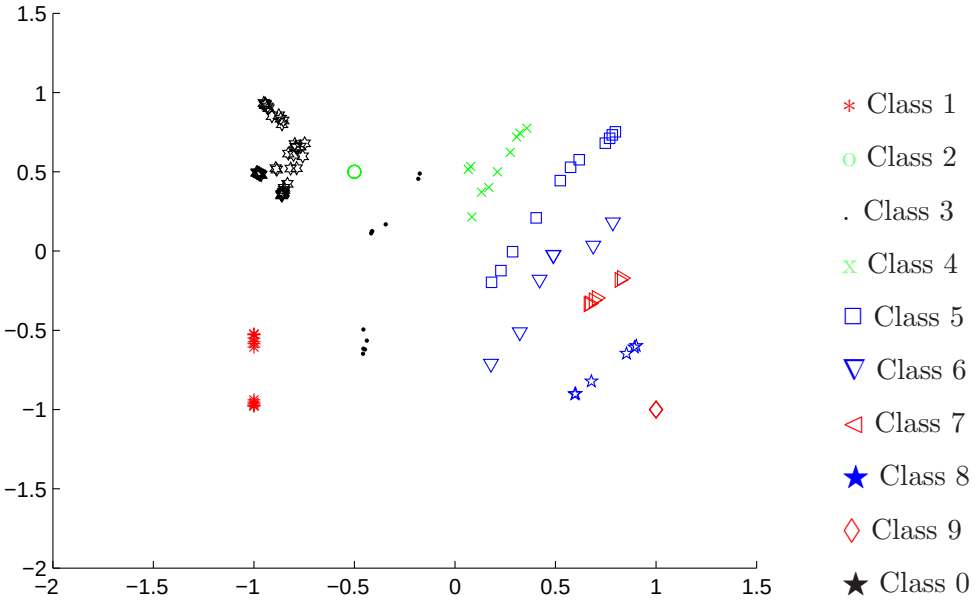

Figure 3: The D-HaToM projection of the 9 labelled classes and one unlabeled on a harmonic mapping with a 2 dimensional set of 5*5 latent points.

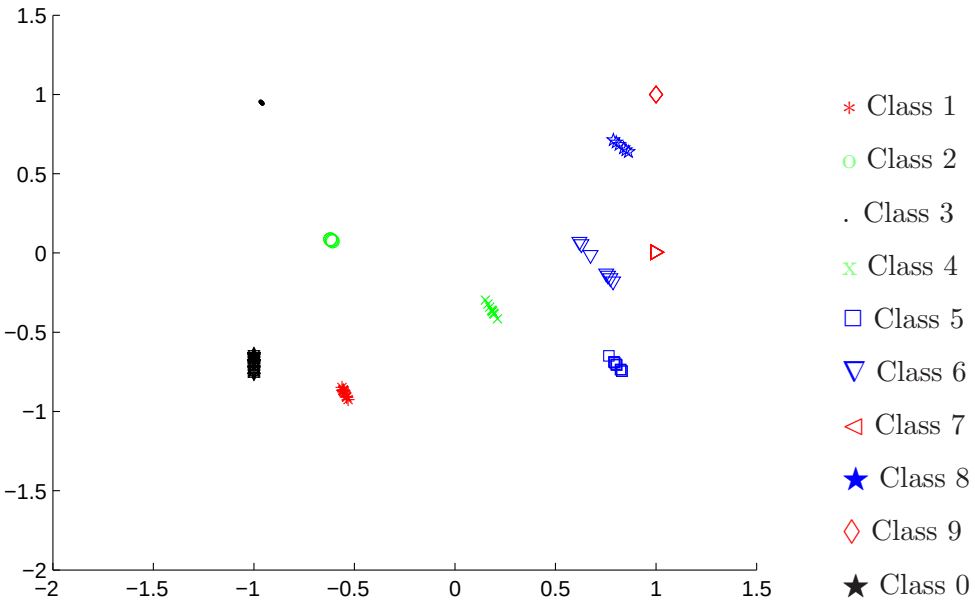

Figure 4: The M-HaToM projection of the 9 labelled classes and one unlabeled on a harmonic mapping with a 2 dimensional set of 5*5 latent points.

References

[1] C. M. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 1997.

[2] C. Fyfe. The topographic product of experts. In *International Conference on Artificial Neural Networks, ICANN2005*, 2005.

[3] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[4] Tuevo Kohonen. *Self-Organising Maps*. Springer, 1995.

[5] M. Peña and C. Fyfe. The harmonic topographic map. In *The Irish conference on Artificial Intelligence and Cognitive Science, AICS05*, 2005.

[6] B. Zhang. Generalized k-harmonic means – boosting in unsupervised learning. Technical report, HP Laboratories, Palo Alto, October 2000.

[7] B. Zhang, M. Hsu, and U. Dayal. K-harmonic means - a data clustering algorithm. Technical report, HP Laboratories, Palo Alto, October 1999.