Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp105-110)

Quantization Errors in the Harmonic Topographic Mapping

STEPHEN MCGLINCHEY, MARIAN PEÑA and COLIN FYFE
Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland.

Abstract We review two versions of a new topology preserving mapping, the HaToM. This mapping has previously been investigated as a data visualization tool but, in this paper, we investigate empirically the quantization errors in both versions of the mapping. We show that the more model driven version does not minimise the quantization error either when it is calculated in the usual manner or when we use the Harmonic average to do so. Somewhat surprisingly the model driven method lowers the quantization error more quickly than the data-driven method.

Key-words: Clustering, Topographic maps, quantization errors.

1 Introduction

Topology-preserving mappings such as the Self-organizing Map (SOM) [3] and the Generative Topographic Mapping(GTM) [1] have been very popular for data visualization: we project the data onto the map which is usually two dimensional and look for structure in the projected map by eye. Therefore when we introduced a new topographic mapping, the Harmonic Topographic Mapping (HaToM) [4], our first concern was its use as a data visualization mechanism. However, more recently we have investigated other uses of this mapping such as outlier detection [6] and as a forecasting tool [5].

Forecasting relies on the fact that topographic maps may also be used as data quantiZers: typically these maps are built around a number of centres and we may allocate data points to the centres to which they are closest. This is not the main use of topographic maps but is one to which they are often applied. The Harmonic Topographic Mapping was built on the success of Harmonic K-Means [8, 7] which was designed as a data quantization method. This raises the question which we address in this paper: how good are the variants of HaToM as data quantizers? We first review two versions of HaToM before addressing this question.

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

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

2.1 Harmonic K-Means

The Harmonic Means were applied to the K-Means algorithm in [8] to make the K-means a more robust algorithm. The recursive formula to update the means is

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

where $d_{i,k}$ is the Euclidean distance between the i^{th} data point and the k^{th} centre.

[8] 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.

3 The Harmonic Topograpic Map

The Harmonic Topographic Map (HaToM) was developed as an alternative to the Topographic Product of Experts (ToPoE) [2], which is also based on the GTM. The HaToM has the same structure as the GTM, with a number of latent points that are mapped to a feature space by M Gaussian functions, and then into the data space by a matrix W. Each latent point, indexed by k is mapped, through a set of M basis functions, $\Phi_1(t_k), \Phi_2(t_k), \cdots, \Phi_M(t_k)$ to a centre in data space, $\mathbf{m}_k = \Phi(t_k)W$. The weights in the HaToM are adjusted using the Harmonic K-Means algorithm; it gains its topology-preserving properties from the arrangement of points in the latent space.

The basic batch algorithm often exhibited twists, such as are well-known in the SOM [3], so we developed a growing method that prevents the mapping from developing these twists.

We developed two versions of the algorithm (see [4]). The main structure for the Data-driven HaToM or D-HaToM is as follows:

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, $\mathbf{m}_k$.
   (a) count=0
   (b) For every data point, $\mathbf{x}_i$, calculate $d_{i,k} = ||\mathbf{x}_i - \mathbf{m}_k||$.
   (c) Recalculate centres, $\mathbf{m}_k$, using (2).
   (d) If count<MAXCOUNT, count= count +1 and return to 3b
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.

We do not randomise W each time we augment K. The current value of W is approximately correct and so we need only continue training from this current value.

In the Model-driven HaToM or M-HaToM, we give greater credence to the model by recalculating W and hence the centres, $\mathbf{m}_k$, within the central loop each time. Thus we are explicitly forcing the structure of the M-HaToM model on the data. The projection method is the same as above. In [4], we showed that this version had several advantages over the D-HaToM: in particular, the M-HaToM creates tighter clusters of data points and finds an underlying data manifold smoothly no matter how many latent points are used in creating the manifold. The D-HaToM, on the other hand, is too responsive to the data, but as shown in [6], this quality makes it more suitable for outlier detection.

However, till now it remains an open question as to how the addition of topology-preserving properties affects the data quantization properties of the underlying Harmonic K-Means algorithm. We now investigate this.

4 Simulations

We investigate this issue in the context of an artificial data set so that we may control every aspect of this experimental investigation. We create a data set with 4 clusters of data, each of 250 two dimensional points. The first 250 points are taken from a uniform distribution in $[0,1]\times[0,1]$, the second 250 are in $[0,1]\times[3,4]$, the third 250 are in $[3,4]\times[0,1]$ and the last 250 are in $[3,4]\times[3,4]$. Sample data and the positions of the latent points' projections found by one simulation at 4 separate growing stages of the D-HaToM are shown in Figure 1. We note that, at the third stage shown, with 9 latent points, there is a centre which lies between two clusters of data. This may be useful for visualisation purposes since data points when visualised in latent space may lie between this point and those of its neighbours, however when we consider clustering distances, this point will not feature *directly* since it never is the projection of a latent point which is closest to any data point. However, as we shall see, such points do have an indirect effect on the quantization error (particularly for the M-HaToM) since they pull other latent points' projections away from the centre of the data.

We show in Figure 2, the decrease in the total distance between each of the 1000 data samples and the nearest latent point to each sample as we train a one dimensional D-HaToM. The 'one-dimensional' refers to the dimensionality of the latent space: we begin with 2 latent points in a line and increase this gradually throughout the simulation till we have 21 latent points in a line. We note that this data set is not ideal for the mapping in that there is a mismatch between the dimensionality of the data and that of the map, however it allows us to illustrate the main points with respect to quantization error. Each time we augment the number of latent points we train the D-HaToM for 20 iterations and refer to this as the growing cycle. Unsurprisingly we see that the total quantization error decreases as we add latent points; however, later additions to the number of latent points do not monotonically decrease the quantization error as they should with K-Harmonic Means.

We show the results of repeating this with the M-HaToM in Figure 3. We see the same gross features however the details are rather different. We see that addition of latent points at first can cause an increase in total distance but that this then falls. However later in the simulation when we add a new latent point the total quantization error often decreases as we add a new latent point and then increases slightly as we go through the growing cycle. In the right diagram of Figure 3, we see this happening as we go from $15 \rightarrow 16 \rightarrow 17 \rightarrow 18$. Thus the re-imposition of the model is having an adverse effect on total quantization error.

This might suggest that we can remove the growing cycle from the simulation: simply increase the number of latent points and omit the growing cycle training. We show the quantization error from such a simulation on the same data as before with the D-HaToM [1] in Figure 4 in which we can see that the quantization error does in fact decrease. However a substantial growing cycle length is essential for good visualization as we illustrate in the right diagram of that figure; the two dimensional D-HaToM has been run on this data set with a growing cycle of just 2 iterations but has failed to fill the data space.

5 Conclusion

We have investigated quantization errors in both the D-HaToM and the M-HaToM. Previous research had concentrated on their use as visualization tools, however these mappings can also be used as quantizers. We have shown that we may estimate the number of clusters in a data set using the quantization error but that the decrease in quantization error is not monotonic. This is true of both versions of the mapping, however it is most pronounced for the M-HaToM in which we are re-asserting the

[1] There is actually no difference between the varieties when we remove the growing cycle training.

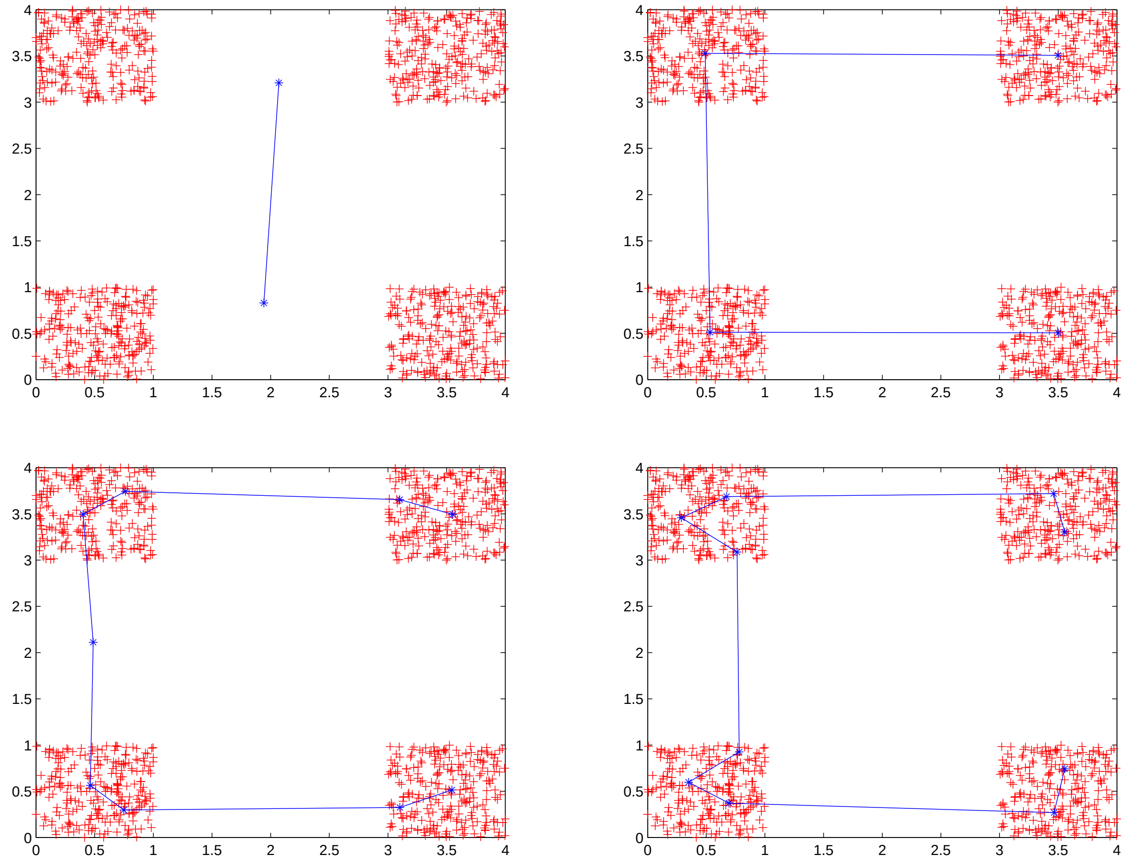

Figure 1: The positions of the data and the latent points' projections with D-HaToM when the number of latent points is 2, 4, 9 and 10 at the end of each growing cycle.

model's priority at each iteration. Indeed with the M-HaToM, we often see the quantization error decrease with the addition of a new latent point only to increase as training with the new latent point continues. However, we have shown that, while this may be true of quantization error, a map which is trained with a very short growing cycle is not so useful since it fails to capture the main features of the data.

Finally we investigated the effect of training on the quantization error calculated by harmonic averages and found similar behaviour in general but different in details. The D-HaToM quantization error so calculated takes longer to decrease than that of the M-HaToM but eventually gets to a lower figure.

Future work will investigate quantization errors on real data sets.

References

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

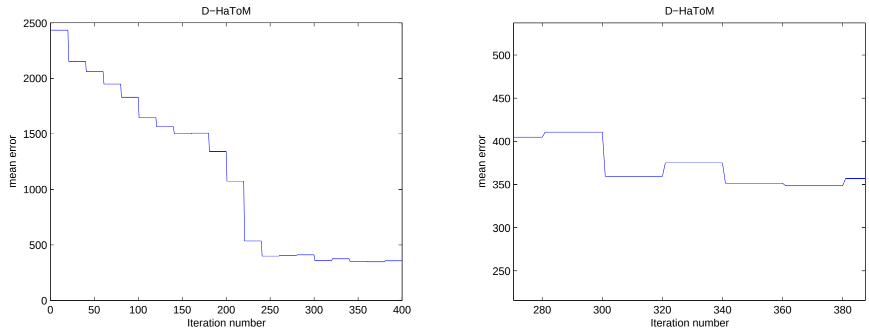

Figure 2: The decrease in the total distance between data and the nearest latent point projection as the number of latent points is grown from 2 to 21 in the D-HaToM. We use 20 iterations per growing cycle. In the right diagram, we have zoomed in to a section of the map.

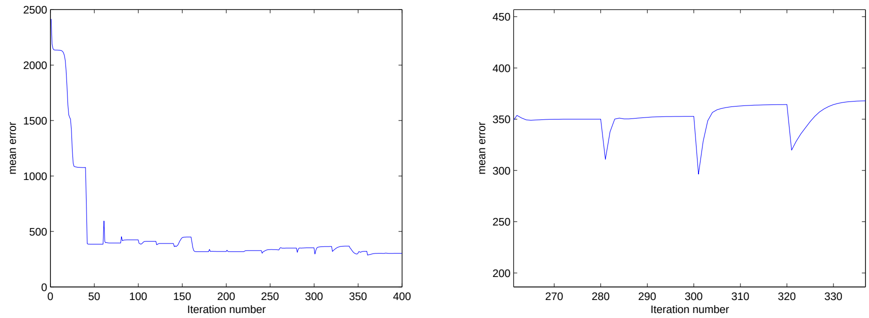

Figure 3: The decrease in the total distance between data and the nearest latent point projection as the number of latent points is grown from 2 to 21 in the M-HaToM. We use 20 iterations per growing cycle. In the right diagram we show the change in quantization error as the number of latent points grows from 15 to 18.

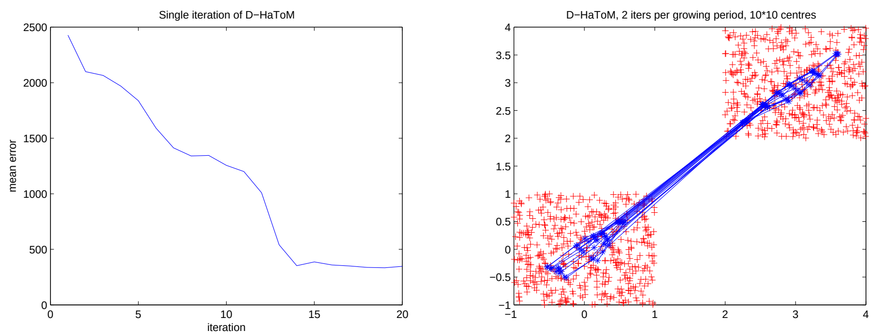

Figure 4: Decrease in the total distance between data and the nearest latent point projection as the number of latent points is grown from 2 to 21 with the D-HaToM and a single iteration in the growing cycle. Right: a two dimensional D-HaToM with a growing cycle length of 2 fails to map the data.

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

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

[4] M. Peña and C. Fyfe. Model- and data-driven harmonic topographic maps. *WSEAS Transactions on Computers*, 4(9):1033–1044, 2005.

[5] M. Peña and C. Fyfe. Forecasting with the growing harmonic topographic mapping. In *International Symposium on Forecasting, ISF06*. Springer, 2006.

[6] M. Peña and C. Fyfe. Outlier identification with the harmonic topographic mapping. In *14th European Symposium on Artificial Neural Networks*, 2006.

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

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