

Visualising and Clustering Video Data

Colin Fyfe¹, Wei Chuang Ooi², and Hanseok Ko²

¹ Applied Computational Intelligence Research Unit,
The University of Paisley, Scotland.
email:colin.fyfe@paisley.ac.uk

² Department of Electronics and Computer Engineering,
Korea University, Korea
email:hsko@korea.ac.kr

Abstract. We review a new form of self-organizing map which is based on a nonlinear projection of latent points into data space, identical to that performed in the Generative Topographic Mapping (GTM) [1]. But whereas the GTM is an extension of a mixture of experts, this model is an extension of a product of experts [6]. We show visualisation and clustering results on a data set composed of video data of lips uttering 5 Korean vowels and show that the new mapping achieves better results than the standard Self-Organizing Map.

1 Introduction

This paper constitutes a part of a much larger study into whether video data which contains both visual and audio information can be used to better transcribe speech data than with audio data alone. Both video and audio data can be very high dimensional - visual data is captured at 20+ frames per second and each frame may contain 10000+ pixels; audio information is generally captured at 8KHz upwards. Both therefore give high dimensional data and we generally wish to process this information in real time. This suggests the need for some form of dimensionality reduction.

We have previously [3, 2] investigated linear projections of data sets but such global linear projections may not be able to capture the structure of a data set when the data is either locally low dimensional but globally high dimensional or when the data lies on a nonlinear manifold. We therefore consider nonlinear projections in this paper.

A topographic mapping (or topology preserving mapping) is a transformation which captures some structure in the data so that points which are mapped close to one another share some common feature while points which are mapped far from one another do not share this feature. The most common topographic mappings are Kohonen's self-organizing map (SOM) [10] and varieties of multi-dimensional scaling [5]. The SOM was introduced as a data quantisation method but has found at least as much use as a visualisation tool. It does have the disadvantage that it retains the quantisation element so that while its centres may lie on a manifold, the user must interpolate between the centres to infer the shape of the manifold.

In this paper, we review two topology preserving mappings, the familiar Self-Organizing Map [10] and the second we call the Topographic Products of Experts (ToPoE)[4]. Based on a generative model of the experts, we show how a topology preserving mapping can be created from a product of experts in a manner very similar to that used by [1] to convert a mixture of experts to the Generative Topographic Mapping (GTM).

We begin with a set of experts who reside in some latent space and take responsibility for generating the data set. In a mixture of experts [8, 9], the experts divide up the data space between them, each taking responsibility for a part of the data space. This division of labour enables each expert to concentrate on a specific part of the data set and ignore those regions of the space for which it has no responsibility. The probability associated with any data point is the sum of the probabilities awarded to it by the experts. There are efficient algorithms, notably the Expectation-Maximization algorithm, for finding the parameters associated with mixtures of experts. [1] constrained the experts' positions in latent space and showed that the resulting mapping also had topology preserving properties.

In a product of experts, all the experts take responsibility for all the data: the probability associated with any data point is the (normalised) product of the probabilities given to it by the experts. As pointed out in e.g. [7] this enables each expert to waste probability mass in regions of the data space where there is no data, provided each expert wastes his mass in a different region. The most common situation is to have each expert take responsibility for having information about the data's position in one dimension while having no knowledge about the other dimensions at all, a specific case of which is called a Gaussian pancake in [11]: a probability density function which is very wide in most dimensions but is very narrow (precisely locating the data) in one dimension. It is very elegantly associated with Minor Components Analysis in [11].

In this paper, we review a method of creating a topology preserving mapping from a product of experts, ToPoE. The resulting mapping is neither a true product of experts nor a mixture of experts but lies somewhere in between.

2 SOM

Kohonen's algorithm is exceedingly simple - the network is a simple 2-layer network and competition takes place between the output neurons; however now not only are the prototypes into the winning neuron updated but also the prototypes of its neighbours. Kohonen defined a neighbourhood function $f(i, i^*)$ of the winning neuron i^* . The neighbourhood function is a function of the distance between i and i^* . A typical function is the Difference of Gaussians function; thus if unit i is at point \mathbf{r}_i in the output layer then

$$f(i, i^*) = a \exp\left(\frac{-|r_i - r_{i^*}|^2}{2\sigma^2}\right) - b \exp\left(\frac{-|r_i - r_{i^*}|^2}{2\sigma_1^2}\right)$$

The algorithm is

1. Select at random an input point.
2. There is a competition among the output neurons. That neuron whose prototype is closest to the input data point wins the competition:

$$\text{winning neuron, } i^* = \arg \min(\| \mathbf{x} - \mathbf{w}_i \|)$$

3. Now update all neurons' prototypes using

$$\Delta w_{ij} = \alpha(x_j - w_{ij}) * f(i, i^*)$$

4. Go back to the start.

Kohonen typically keeps the learning rate constant for the first 1000 iterations or so and then slowly decreases it to zero over the remainder of the experiment. Two dimensional maps can be created by imagining the output neurons laid out on a rectangular grid or sometimes a hexagonal grid.

3 Topographic Products of Experts

Hinton [6] investigated a product of K experts with

$$p(\mathbf{x}_n | \Theta) \propto \prod_{k=1}^K p(\mathbf{x}_n | k) \quad (1)$$

where Θ is the set of current parameters in the model. Hinton notes that using Gaussians alone does not allow us to model e.g. multi-modal distributions, however the Gaussian is ideal for our purposes. Thus our base model is

$$p(\mathbf{x}_n | \Theta) \propto \prod_{k=1}^K \left(\frac{\beta}{2\pi} \right)^{\frac{D}{2}} \exp \left(-\frac{\beta}{2} \|\mathbf{m}_k - \mathbf{x}_n\|^2 \right) \quad (2)$$

We will, as with the GTM, allow latent points to have different responsibilities depending on the data point presented:

$$p(\mathbf{x}_n | \Theta) \propto \prod_{k=1}^K \left(\frac{\beta}{2\pi} \right)^{\frac{D}{2}} \exp \left(-\frac{\beta}{2} \|\mathbf{m}_k - \mathbf{x}_n\|^2 r_{kn} \right) \quad (3)$$

where r_{kn} is the responsibility of the k^{th} expert for the data point, \mathbf{x}_n . Thus all the experts are acting in concert to create the data points but some will take more responsibility than others. Note how crucial the responsibilities are in this model: if an expert has no responsibility for a particular data point, it is in essence saying that the data point could have a high probability as far as it is concerned. We do not allow a situation to develop where no expert accepts responsibility for a data point; if no expert accepts responsibility for a data

point, they all are given equal responsibility for that data point (see below). For comparison, the probability of a data point under the GTM is

$$p(\mathbf{x}) = \sum_{i=1}^K P(i)p(\mathbf{x}|i) = \sum_{i=1}^K \frac{1}{K} \left(\frac{\beta}{2\pi} \right)^{\frac{D}{2}} \exp \left(-\frac{\beta}{2} \|\mathbf{m}_i - \mathbf{x}\|^2 \right) \quad (4)$$

We wish to maximise the likelihood of the data set $X = \{\mathbf{x}_n : n = 1, \dots, N\}$ under this model. The ToPoE learning rule (6) is derived from the minimisation of $-\log(p(\mathbf{x}_n|\Theta))$ with respect to a set of parameters which generate the \mathbf{m}_k .

We now turn our attention to the nature of the K experts which are going to generate the K centres, \mathbf{m}_k . We envisage that the underlying structure of the experts can be represented by K latent points, t_1, t_2, \dots, t_K . To allow local and non-linear modeling, we map those latent points through a set of M basis functions, $f_1(), f_2(), \dots, f_M()$. This gives us a matrix Φ where $\phi_{kj} = f_j(t_k)$. Thus each row of Φ is the response of the basis functions to one latent point, or alternatively we may state that each column of Φ is the response of one of the basis functions to the set of latent points. One of the functions, $f_j()$, acts as a bias term and is set to one for every input. Typically the others are gaussians centered in the latent space. The output of these functions are then mapped by a set of weights, W , into data space. W is $M \times D$, where D is the dimensionality of the data space, and is the sole parameter which we change during training. We will use \mathbf{w}_i to represent the i^{th} column of W and Φ_j to represent the row vector of the mapping of the j^{th} latent point. Thus each basis point is mapped to a point in data space, $\mathbf{m}_j = (\Phi_j W)^T$.

We may update W either in batch mode or with online learning. To change W in online learning, we randomly select a data point, say \mathbf{x}_i . We calculate the current responsibility of the j^{th} latent point for this data point,

$$r_{ij} = \frac{\exp(-\gamma d_{ij}^2)}{\sum_k \exp(-\gamma d_{ik}^2)} \quad (5)$$

where $d_{pq} = \|\mathbf{x}_p - \mathbf{m}_q\|$, the euclidean distance between the p^{th} data point and the projection of the q^{th} latent point (through the basis functions and then multiplied by W). If no centres are close to the data point (the denominator of (5) is zero), we set $r_{ij} = \frac{1}{K}, \forall j$.

Now we wish to maximise (4) so that the data is most likely under this model. We do this by minimising the $-\log()$ of that probability: define $m_d^{(k)} = \sum_{m=1}^M w_{md} \phi_{km}$, i.e. $m_d^{(k)}$ is the projection of the k^{th} latent point on the d^{th} dimension in data space. Similarly let $x_d^{(n)}$ be the d^{th} coordinate of \mathbf{x}_n . These are used in the update rule

$$\Delta_n w_{md} = \sum_{k=1}^K \eta \phi_{km} (x_d^{(n)} - m_d^{(k)}) r_{kn} \quad (6)$$

where we have used Δ_n to signify the change due to the presentation of the n^{th} data point, \mathbf{x}_n , so that we are summing the changes due to each latent point's

response to the data points. Note that, for the basic model, we do not change the Φ matrix during training at all.

3.1 Comparison with the GTM

The Generative Topographic Mapping (GTM) [1] is a mixture of experts model which treats the data as having been generated by a set of latent points. These K latent points are also mapped through a set of M basis functions and a set of adjustable weights to the data space. The parameters of the combined mapping are adjusted to make the data as likely as possible under this mapping. The GTM is a probabilistic formulation so that if we define $\mathbf{y} = \Phi\mathbf{W} = \Phi(\mathbf{t})\mathbf{W}$, where \mathbf{t} is the vector of latent points, the probability of the data is determined by the position of the projections of the latent points in data space and so we must adjust this position to increase the likelihood of the data. More formally, let

$$\mathbf{m}_i = \Phi(\mathbf{t}_i)W \quad (7)$$

be the projections of the latent points into the feature space. Then, if we assume that each of the latent points has equal probability

$$p(\mathbf{x}) = \sum_{i=1}^K P(i)p(\mathbf{x}|i) = \sum_{i=1}^K \frac{1}{K} \left(\frac{\beta}{2\pi} \right)^{\frac{D}{2}} \exp\left(-\frac{\beta}{2}\|\mathbf{m}_i - \mathbf{x}\|^2\right) \quad (8)$$

where D is the dimensionality of the data space. i.e. all the data is assumed to be noisy versions of the mapping of the latent points. This equation should be compared with (3) and (4).

In the GTM, the parameters W and β are updated using the EM algorithm though the authors do state that they could use gradient ascent. Indeed, in the ToPoE, the calculation of the responsibilities may be thought of as being a partial E-step while the weight update rule is a partial M-step. The GTM has been described as a ‘‘principled alternative to the SOM’’ however it may be criticised on two related issues:

1. it is optimising the parameters with respect to each latent point independently. Clearly the latent points interact.
2. using this criterion and optimising the parameters with respect to each latent point individually does not necessarily give us a globally optimal mapping from the latent space to the data space.

The ToPoE will overcome the second of these shortcomings in that all data points are acting together. Specifically if no latent point accepts responsibility for a data point, the responsibility is shared equally amongst all the latent points.

The GTM, however, does have the advantage that it can optimise with respect to β as well as W . However note that, in (3) and (4), the variance of each expert is dependent on its distance from the current data point via the hyper-parameter, γ . Thus we may define

$$(\beta_k)_{|\mathbf{x}=\mathbf{x}_n} = \beta r_{kn} = \beta \frac{\exp(-\gamma d_{nk}^2)}{\sum_t \exp(-\gamma d_{nt}^2)} \quad (9)$$

Therefore the responsibilities are adapting the width of each expert locally dependent on both the expert’s current projection into data space and the data point for which responsibility must be taken. Initially, $r_{kn} = \frac{1}{K}, \forall k, n$ and so we have the standard product of experts. However during training, the responsibilities are refined so that individual latent points take more responsibility for specific data points. We may view this as the model softening from a true product of experts to something between that and a mixture of experts.

A model based on products of experts has some advantages and disadvantages. The major disadvantage is that no efficient EM algorithm exists for optimising parameters. [6] suggests using Gibbs sampling but even with the very creative method discussed in that paper, the simulation times were excessive. Thus we have opted for gradient descent as the parameter optimisation method.

The major advantage which a product of experts method has is that it is possible to get very much sharper probability density functions with a product rather than a sum of experts.

4 Visualising and Clustering Voice data

This work is part of a larger body of work in which we wish to combine audio and video data in order to better transcribe speakers audio utterances. As part of this work, we investigated clustering and visualisation of the video data alone.

4.1 The data and pre-processing

14 speakers were asked to utter each of 5 Korean vowels and were videoed while doing so. The five vowels were approximately

1. 'ah' as in the English word, 'cat'
2. 'eh' as in the English word, 'bed'
3. 'ee' as in the English word, 'feel'
4. 'oh' as in the English word, 'door'
5. 'wu'³ as in the English word, 'wood'

Each speaker spoke for approximately 1 second on each vowel and between 21 and 27 frames were taken. The video sequences were cropped to a 71×71 pixel region round the mouth so that we have 5041 dimensional data, each pixel of which is in a range from 0 to 255.

The lighting conditions were very different from speaker to speaker and so we first normalised each video so that the pixels varied from -1 to 1 (this is a very crude way to perform this but we wished to minimise the pre-processing requirements). We then performed a principal component analysis of the data and, based on the variances, opted to investigate further processing based on the projection of the data onto 4 and 10 principal components. In practise, there was little difference in the results and in this paper we use the first 10 principal components. Thus we have compressed our 5041 dimensional data down to 10 dimensions and it is in this data that we look for structure.

³ The Korean language *does* have an initial 'w' associated with this sound.

4.2 Experiments

We first use each frame as a separate sample: in Figure 1, we show the projections of the data found by ToPoE. We see that there is some structure in the mapping - the top half is dominated by the open lip data ('ah', 'eh' and 'ee') and the bottom half is dominated by the closed lip data ('oh' and 'wu'). However there is a great deal of overlap between these. This is caused by the fact that in all videos the subjects began the vocalisation in a similar pose. Also a nearest neighbour investigation in this space showed that often the nearest neighbour was a frame of the same person but speaking a different vowel. We therefore subsequently selected the first 21 frames of each of the videos and concatenated these to form one data sample of dimensionality 210 (21 frames of the 10 principal components). Note that this is not the same as performing a principal component analysis of the completed data set (which would have involved a PCA of 21×5041 dimensional data) but is an attempt to capture some essential features of the data.

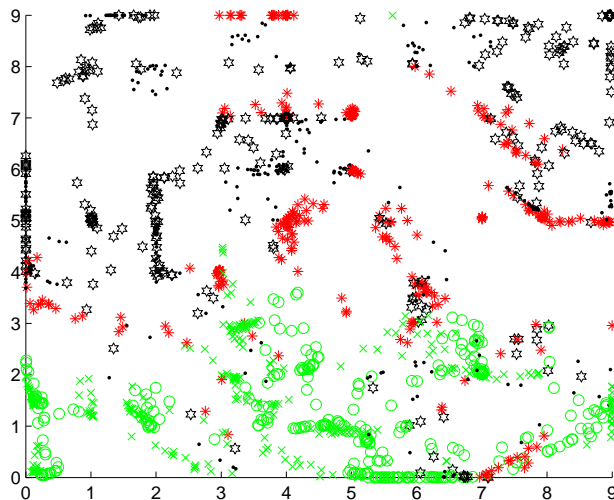


Fig. 1. The ToPoE projection of the visual projections of the lips data. The black stars are 'ah', the red asterisks are 'eh', black dots 'ee', green circles 'oh' and green crosses 'wu'.

Therefore we now have 70 samples (14 speakers each saying 5 vowels) of 210 dimensional data. The SOM projection of this data is shown in Figure 2: we see a very good separation of the open mouth vowels from the rounded mouth vowels but it is not perfect - there is some overlap between the two groups.

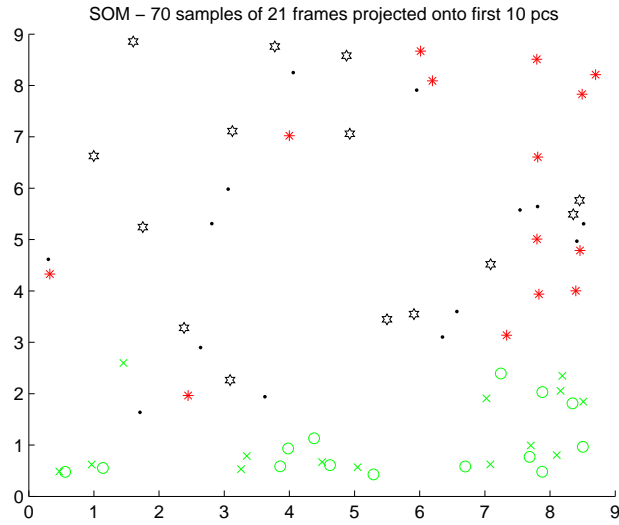


Fig. 2. The SOM projection of the video data when we use 21 frames of the first 10 principal components as 1 data sample. We now have 70 samples = 14 speakers of 5 vowels.

We can alleviate this by using the audio data too. Each audio signal consisted of between 10000 and 16000 samples. We therefore select the first 10000 samples of the audio signal of each video and concatenate these to create a 10000×70 data set. We again performed a PCA on this data set and projected each sample onto the first 10 principal components.

Figure 3 shows the SOM projection when we use 70 samples (14 speakers of 5 vowels) with the combined audio and video data. We see a far better separation of the two groups of vowels; note that treating this data as two separate data streams which can be subsequently conjoined means that we do not have to worry about the problem of matching the audio and visual data streams in time. However this process left us somewhat dissatisfied in that our original investigation was into utilising the information from the visual data to assist the transcription of the audio data. The results here certainly show that we can use one to assist in differentiating the other but we are actually using the audio data to assist in optimising the projection of the visual data.

We therefore investigate the use of the ToPoE on only the visual data as above. The results are shown in Figure 4: the two groups of vowels are clearly separated using only the visual data.

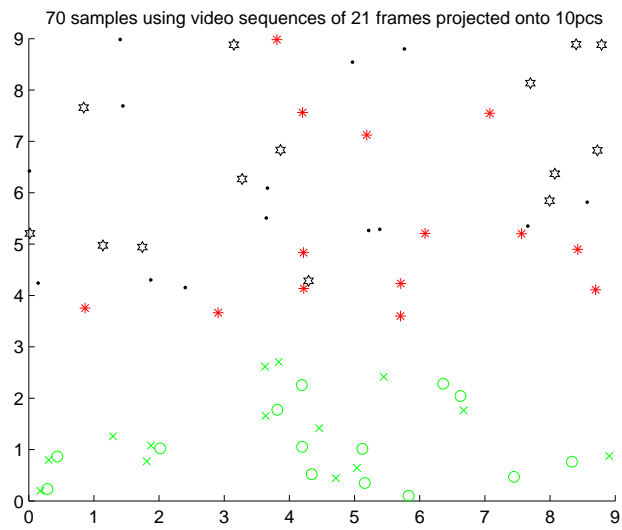


Fig. 3. The SOM projection of the combined audio and visual data. A clearer separation of the two groups of vowels is achieved.

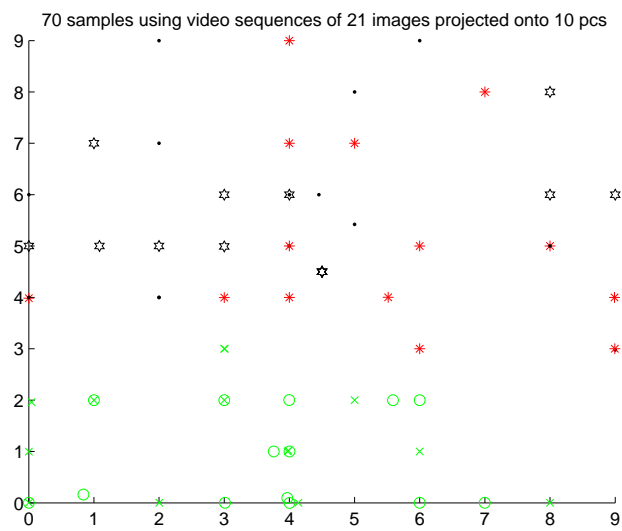


Fig. 4. The ToPoE clearly separates the two groups of vowels very clearly.

5 Conclusion

We investigated the task of finding a good projection of visual data so that different classes of visual data can be clearly identified. We have shown that the Topographic Product of Experts gives a better projection than the standard Self-Organizing Map, though if we add audio information the difference between the mappings is much less.

Future work will continue on these lines with a larger data set, the current work being only a proof of concept. We will also investigate other projections of these data sets such as principal curves, isomap and so on.

Acknowledgement This research was supported by the MIC (Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement).

References

1. C. M. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 1997.
2. E. Corchado, D. MacDonald, and C. Fyfe. Maximum and minimum likelihood hebbian learning for exploratory projection pursuit. *Data Mining and Knowledge Discovery*, 8:203–225, 2004.
3. C. Fyfe. A comparative study of two neural methods of exploratory projection pursuit. *Neural Networks*, 10(2):257–262, 1997.
4. C. Fyfe. Two topographic maps for data visualization. *Data Mining and Knowledge Discovery*, 14:207–224, 2007. ISSN 1384-5810.
5. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
6. G. E. Hinton. Training products of experts by minimizing contrastive divergence. Technical Report GCNU TR 2000-004, Gatsby Computational Neuroscience Unit, University College, London, <http://www.gatsby.ucl.ac.uk/>, 2000.
7. G.E. Hinton and Y.-W. Teh. Discovering multiple constraints that are frequently approximately satisfied. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 227–234, 2001.
8. R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
9. M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.
10. Tuevo Kohonen. *Self-Organising Maps*. Springer, 1995.
11. C. Williams and F. V. Agakov. Products of gaussians and probabilistic minor components analysis. Technical Report EDI-INF-RR-0043, University of Edinburgh, 2001.