Properties of the Topographic Product of Experts

Colin Fyfe
The University of Paisley
Paisley, Scotland
colin.fyfe@paisley.ac.uk

Abstract - *In this paper, we show how a topographic mapping can be created from a product of experts. We learn the parameters of the mapping using gradient descent on the negative logarithm of the probability density function of the data under the model. We show that the mapping, though retaining its product of experts form, becomes more like a mixture of experts during training.*

Key words - topographic, product of experts

1 Introduction

Recently [2], we introduced a new topology preserving mapping which we called the Topographic Products of Experts (ToPoE). 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).

We begin with a set of experts who reside in some latent space and take responsibility for generating the data set. With a mixture of experts [5, 6], 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. Bishop *et al* [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. [4] 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 [7]: 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 [7].

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

2 Topographic Products of Experts

Hinton [3] investigated a product of K experts with

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

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. To fit this model to the data we can define a cost function as the negative logarithm of the probabilities of the data so that

$$C = \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{\beta}{2}||\mathbf{m}_k - \mathbf{x}_n||^2 \tag{2}$$

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

$$C_1 = \sum_{n=1}^{N}\sum_{k=1}^{K} \frac{\beta}{2}||\mathbf{m}_k - \mathbf{x}_n||^2 r_{kn} \tag{3}$$

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

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

We wish to maximise the likelihood of the data set $X = \{\mathbf{x}_n : n = 1, \cdots, N\}$ under this model. The ToPoE learning rule (6) is derived from the minimisation of C_1 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, $t_1, t_2, \cdots, t_K$. To allow local and non-linear modeling, we map those latent points through a set of M basis functions, $f_1(), f_2(), \cdots, f_M()$. 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, $\mathbf{m}_j = \Phi_j W$.
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,

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

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 weights are close to the data point (the denominator of (5) is zero), we set $r_{ij} = \frac{1}{K}, \forall j$.
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

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

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. It is the combination of the fact that the latent points are mapped through the basis functions and that the latent points are given fixed positions in latent space which gives the ToPoE its topographic properties. We have previously illustrated these on artificial data in [2].

3 Product or Mixture?

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. [3] 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.
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.

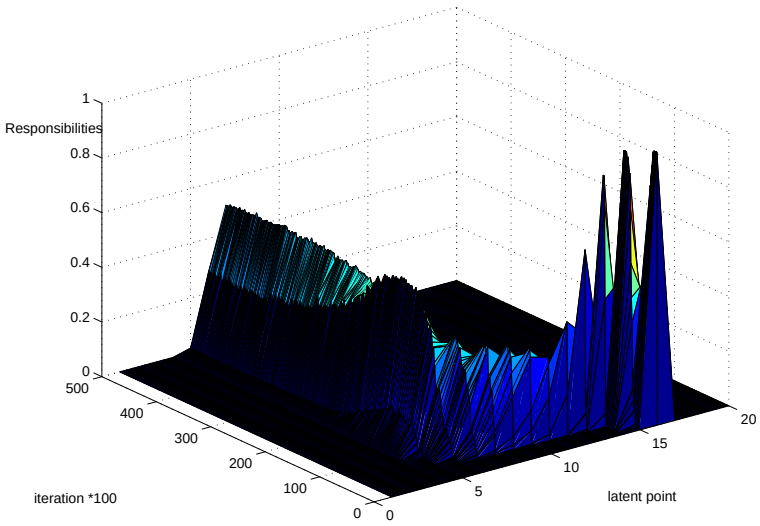

Figure 1: There is an initial competition to take responsibility for a specific data point but quickly converge so that just a few latent points do so.

To illustrate this, we create an artificial 2 dimensional, 60 sample data set which lies close to a single nonlinear manifold and train a ToPoE with 20 latent points arranged in a line in a 1 dimensional latent space. We may show the growth of the responsibilities from either the perspective of a single data point (Figure 1) or from the perspective of an individual latent point (Figure 2). Initially we see the latent points assuming a broad responsibility which is refined in time till each latent point has only a responsibility for a few data points and conversely each data point is being generated (under the model) by only a few latent points: we have moved some way from the product of experts towards a mixture of experts.
However, the responsibilities do not, in general, tend to 0 or 1. Typically the responsibility for a data point is shared between several latent points. In [2], we have shown that this sharpening of the responsibilities takes place even when we use a non-local function to map the latent points to feature space. In that paper, we used $\phi_{kj} = f_j(t_k) = \tanh(jt_k)$.
This feature of not quite having one expert take sole responsibility for a data point is, in fact, rather useful for visualisation. We illustrate with a data 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. Figure 3 shows the projection of the 9 labeled classes (72 samples). Note that few of the samples could be said to be lying at an integer coordinate on the map. Most lie between integral values and the clusters are easily seen. When we zoom into the central part of this mapping (Figure 4, left), we find that we can disambiguate the 8th and 9th classes. However, the right diagram in that figure suggests that the remaining two classes are not completely distinguished. Figure 5 shows the projection of the whole data set including the unlabeled samples. From this, we conjecture that

- there are other classes in the data set which have not yet been identified.

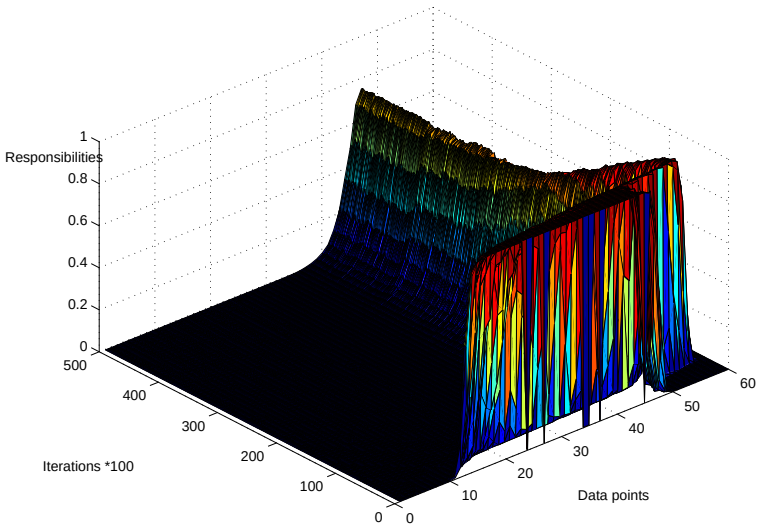

Figure 2: The latent point initially has broad responsibilities but learns to take responsibility for only a few data points.

- some of the unclassified samples belong to classes already identified.
- some may be simply outliers.

These are, however, speculations on our part and must be validated by a scientist with biological expertise.
It is of interest to compare the GTM on the same data: we use a two dimensional latent space with a 10×10 grid for comparison. The results are shown in Figure 6. The GTM makes a very confident classification: we see that the responsibilities for data points are very confidently assigned in that individual classes tend to be allocated to a single latent point. This, however works against the GTM in that, even with zooming in to the map, one cannot sometimes disambiguate the two different classes such as at the points (1,-1) and (1,1). This was not alleviated by using regularisation in the GTM though we should point out that we have a very powerful model for a rather small data set.
In fact, we can control the level of quantisation by changing the γ parameter in (5). For example by lowering γ, we share the responsibilities more equally and so the map contracts to the centre of the latent space to get results such as shown in Figure 7; the different clusters can still be identified but rather less easily. Alternately, by increasing γ, one tends to get the data clusters confined to a single node, that which has sole responsibility for that cluster. We are, in effect, able to control how much our product of experts mapping moves towards a mixture of experts.

References

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

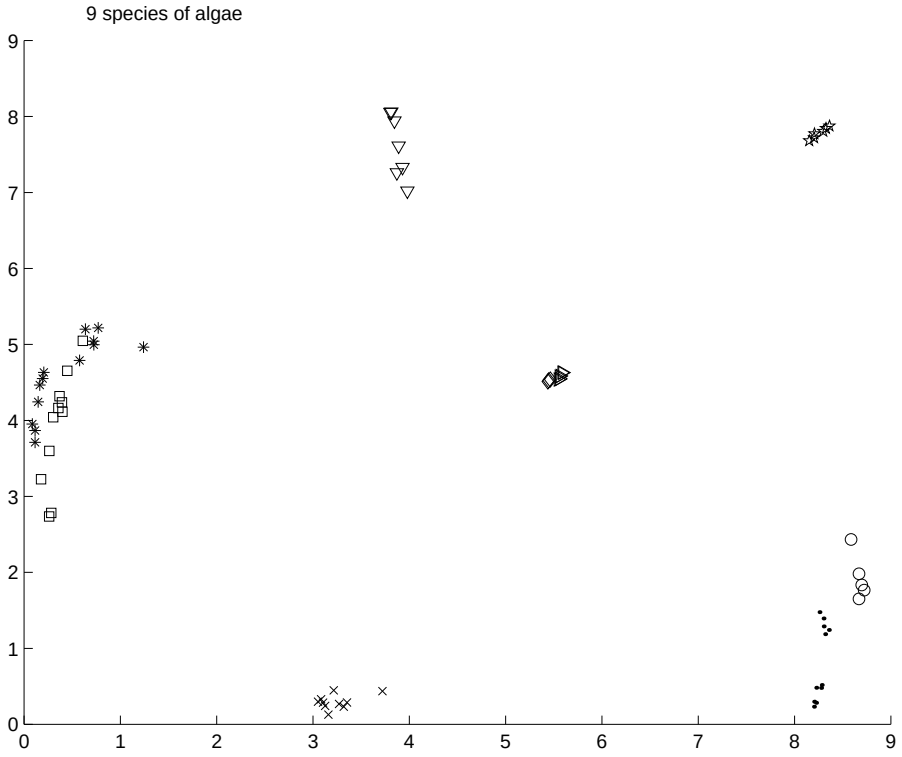

Figure 3: Projection of the 9 classes by the ToPoE.

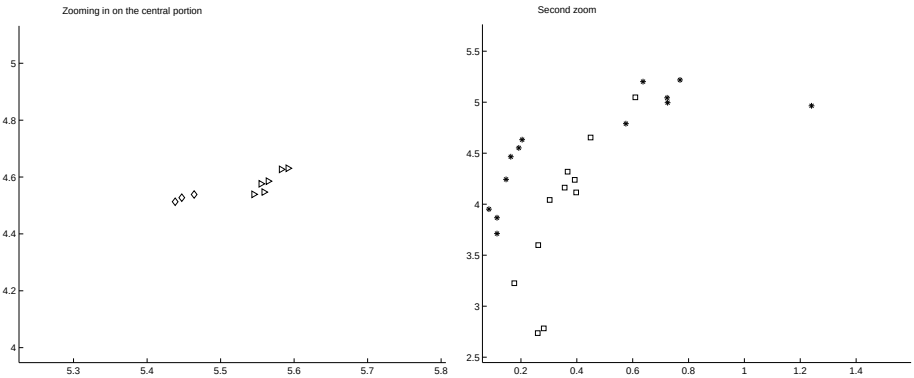

Figure 4: Left: zooming in on the central portion. Right: zooming in on the left side.

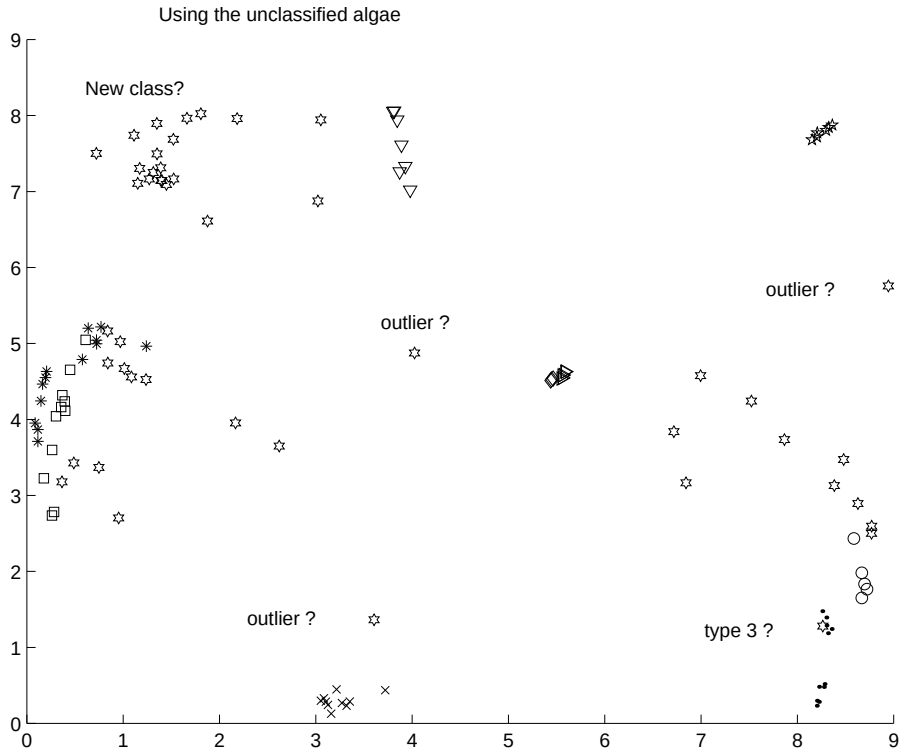

Figure 5: The projection of the whole data set by the ToPoE.

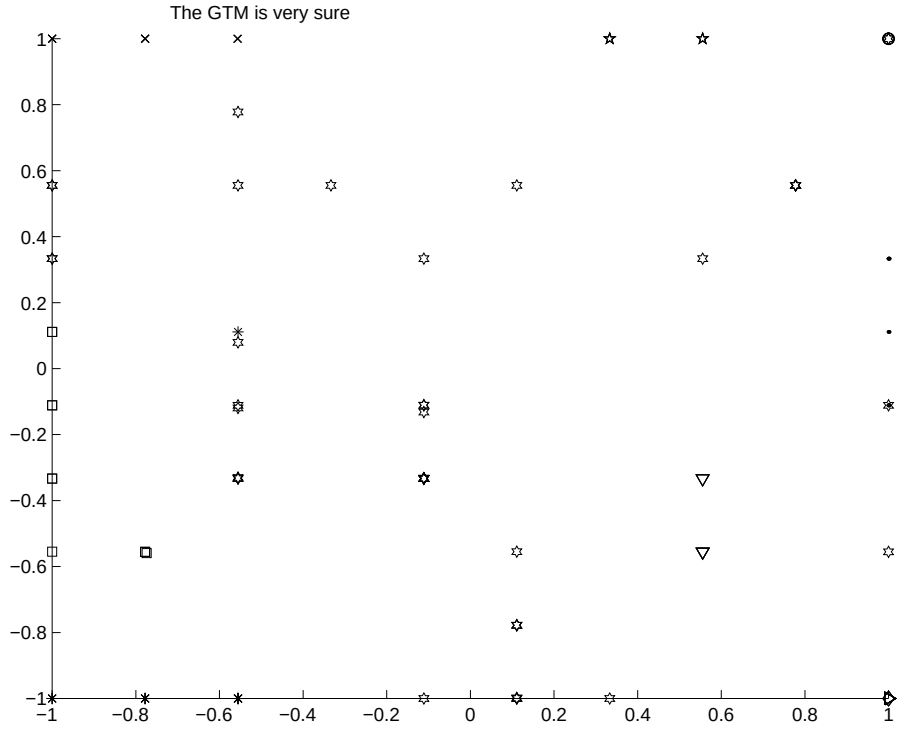

Figure 6: The projection of the algae data given by the GTM.

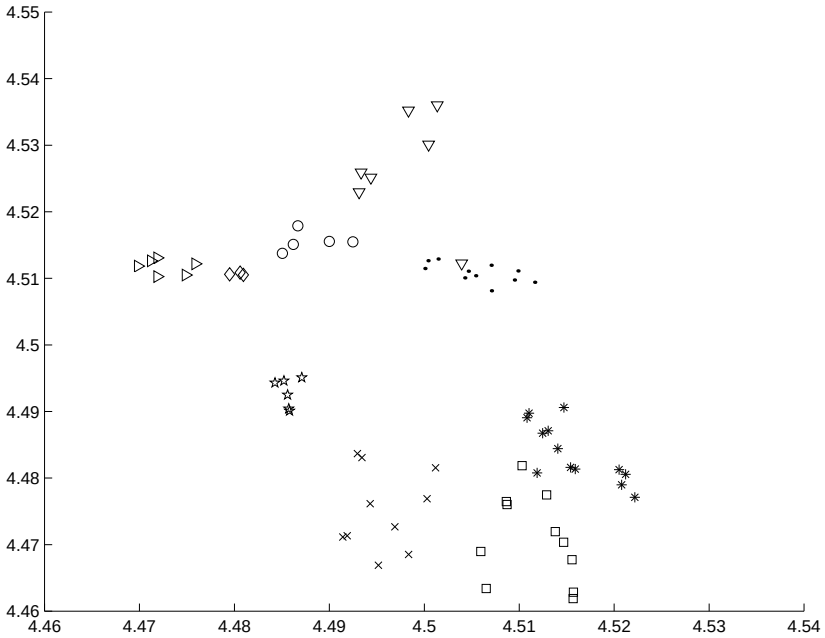

Figure 7: By lowering the γ parameter, the ToPoE map is contracted.

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

[3] 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.

[4] 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.

[5] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[6] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.

[7] C. Williams and F. V. Agakov. Products of gaussians and probabilistic minor components analysis. Technical Report EDI-INF-RR-0043, University of Edinburgh, 2001.