International Journal of Neural Systems, Vol. 18, No. 6 (2008) 1–9

World Scientific
www.worldscientific.com

TOPOLOGICAL MAPPINGS OF VIDEO AND AUDIO DATA

COLIN FYFE* and WESAM BARBAKH†
Applied Computational Intelligence Research Unit
The University of The West of Scotland, UK
**colin.fyfe@uws.ac.uk*
†*wesam.barbakh@uws.ac.uk*

WEI CHUAN OOI and HANSEOK KO‡
Department of Electronics and Computer Engineering
Korea University, Korea
‡*hsko@korea.ac.kr*

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.[2] We show visualisation and clustering results on a data set composed of video data of lips uttering 5 Korean vowels. Finally we note that we may dispense with the probabilistic underpinnings of the product of experts and derive the same algorithm as a minimisation of mean squared error between the prototypes and the data. This leads us to suggest a new algorithm which incorporates local and global information in the clustering. Both ot the new algorithms achieve better results than the standard Self-Organizing Map.

Keywords:

1. Introduction

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)[5] and varieties of multi-dimensional scaling.[6] 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.

For the last few years, there has been a large study at Korea University 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 8 KHz 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.

Topographic mappings enable us to perform dimensionality reduction in a nonlinear fashion. In this paper we investigate several different such mappings as well as Kohonen's SOM. The first two alternative mappings are two variants of a mapping which is based on an underlying latent space. One of these, we call the Topographic Products of Experts (ToPoE)[7] and the other is based on the Inverse Weighted K-means.[8] ToPoE is based on a generative model of the experts, and we show how a topology preserving mapping can 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).

Most latent space methods envisage a set of experts who reside in the latent space and take responsibility for generating the data set. The most common model is a mixture of experts[9,10] in which 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 given to it by the experts so that it only requires one expert to give high probability to a data point. There are efficient algorithms, notably the Expectation-Maximization algorithm, for finding the parameters associated with mixtures of experts. Reference 1 cleverly introduced a constraint on the experts' positions in latent space and showed that the resulting mapping 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. Reference 11 discusses how 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. An example of this was called a Gaussian pancake in Ref. 12: a probability density function which is very wide in most dimensions but is very narrow in one dimension. In that paper, it is very elegantly associated with Minor Components Analysis.

In this paper, we first review a method of creating a topology preserving mapping from a product of experts, ToPoE. In fact the final mapping is neither a true product of experts nor a mixture of experts but lies somewhere in between. We then show how this same underlying latent space can be used for exploratory data analysis using a second method which is predicated on the necessity of combining local and global interactions to create globally optimal topographic mappings. However, for completeness, we begin with a review of the well known Self-organizing Map.

2. SOM

Kohonen's algorithm is exceedingly simple and very well known so we will only quickly sketch the details of this network. Each 'neuron' in the output layer has a position in neuron space which is usually one or two dimensional and also has a centre or prototype in data space. When an input is presented to the network, the neuron whose centre/prototype is closest to the input is deemed to have one a competition. however now not only are the prototypes into the winning neuron updated but also the prototypes of its neighbours. Let i^* denote the winning neuron (i.e. its position in neuron space) and i be any other neuron. 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

We begin with a product of K experts with

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

where Θ is the set of current parameters in the model. Hinton[2] notes that using Gaussians alone does not allow us to model e.g. multi-modal distributions, however the Gaussian works very well for our purposes. Thus our first model is

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

The $\mathbf{m}_k$ values are the prototypes generated by the experts in data space and so we first discuss the K experts which are going to generate the K centres/prototypes, $\mathbf{m}_k$. We envisage that the underlying structure of the experts can be represented by K latent points, $t_1, t_2, \ldots, t_K$ whose positions in a latent space exhibit some structure (typically lying at the corners of a grid). We then map those latent points through a set of M basis functions, $f_1(\,), f_2(\,), \ldots, f_M(\,)$. 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 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 through 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 ith column of W and Φ_j to represent the row vector of the mapping of the jth latent point. Thus each basis point is mapped to a point in data space, $\mathbf{m}_j = (\Phi_j W)^T$.

We wish to emulate the success of the GTM and so we allow latent points to have different responsibilities depending on the data point presented. This is used in determining the probability that each expert gives to each data point:

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

where r_{kn} is the responsibility of the kth 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 depending on the responsibilities: 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

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

We wish to maximise the likelihood of the data set $X = \{\mathbf{x}_n : n = 1, \ldots, N\}$ 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 $\mathbf{m}_k$.

To change W in online learning, we randomly select a data point, say $\mathbf{x}_i$. We calculate the current responsibility of the jth 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 pth data point and the projection of the qth 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 kth latent point on the dth dimension in data space. Similarly let $x_d^{(n)}$ be the dth 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 nth 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 which means it need only be calculated once.

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. The structure of the two mappings is identical: K latent points in both are 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. In the GTM, the prototypes' positions are given by $\mathbf{y} = \Phi\mathbf{W} = \Phi(\mathbf{t})\mathbf{W}$, 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 we can adjust this position to increase the likelihood of the data. More formally, let

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

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

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

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".

The GTM also optimises 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

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

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.

An empirical comparison between these mappings on the well known algae data set is given in Ref. 7.

4. Visualising and Clustering Voice Data

This work is part of a larger body of work at Korea University 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'[a] 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 each video is first normalised 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

[a]The Korean language *does* have an initial 'w' associated with this sound.

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.

4.2. *Experiments*

We first use each frame as a separate sample: in Fig. 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 which 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

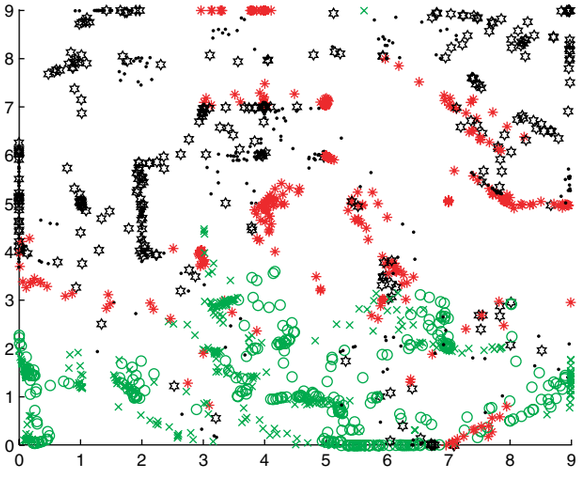

Fig. 1. The ToPoE projection of the visual projections of the lips data. The black stars are 'ah', the red asterisks are 'eh', black dots 'ee', green circles 'oh' and green crosses 'wu'.

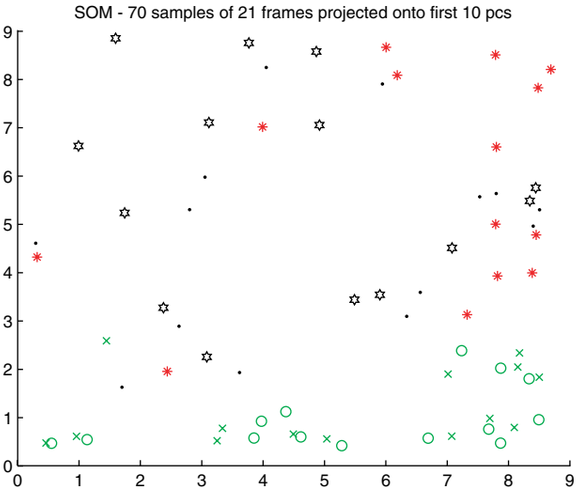

Fig. 2. The SOM projection of the video data when we use 21 frames of the first 10 principal components as 1 data sample. We now have 70 samples = 14 speakers of 5 vowels.

is an attempt to capture some essential features of the data.

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 Fig. 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.

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 is less than satisfactory 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

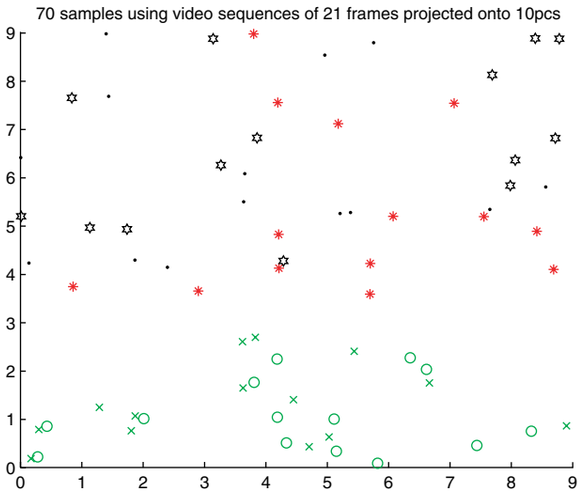

Fig. 3. The SOM projection of the combined audio and visual data. A clearer separation of the two groups of vowels is achieved.

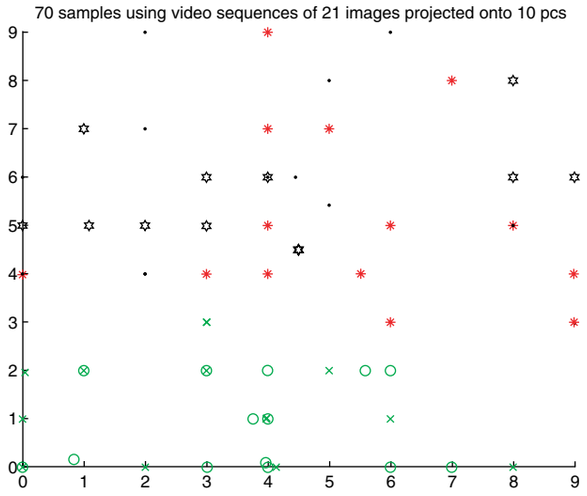

Fig. 4. The ToPoE clearly separates the two groups of vowels very clearly.

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 Fig. 4: the two groups of vowels are clearly separated using only the visual data.

5. Alternative Clustering Criteria

Now if we are not interested in deriving probabilistic learning rules, (6) can clearly be stripped of its probabilistic interpretation and be shown to be a learning rule for minimising the mean squared error (MSE) between the prototypes and the data (while taking into account the mapping from the latent space to the data space). As is well known with the K-means algorithm, the minimisation of the MSE is typically bedevilled by local optima. This has led us to investigate alternative clustering criteria. The deficiency of algorithms minimising MSE is due to their use of solely local information in self-organising. Thus we have attempted to incorporate local and global information[13] into the clustering criteria. One possible criterion we have used in batch mode [8,14] is known as the Inverse Weighted K-Means (IWK) algorithm. We now show an online implementation which optimises this criterion.

5.1. *IWK online algorithm*

In this section we show how it is possible to allow all the units (prototypes) to learn, not only the winner as in K-means or the winner with its neighbors as in Self-organizing map. In this algorithm, it is not necessary to specify any functions for the neighbors as all units learn with every input sample.

Consider the performance function

$$J_{IWK} = \sum_{i=1}^{N} \left[\sum_{j=1}^{K} \frac{1}{\|\mathbf{x}_i - \mathbf{m}_j\|^p} \right] \min_{k=1}^{K} \|\mathbf{x}_i - \mathbf{m}_k\|^n \tag{10}$$

The rationale for this performance function is that we do not wish the situation to arise in which one prototype is optimally responsive to a data point but all the other prototypes are ignoring this data point: we wish all prototypes to take into account all data points. Let $\mathbf{m}_{k*}$ be the closest prototype to $\mathbf{x}_i$. Then

$$\begin{aligned} J_{IWK}(\mathbf{x}_i) &= \left[\sum_{j=1}^{K} \frac{1}{\|\mathbf{x}_i - \mathbf{m}_j\|^p} \right] \|\mathbf{x}_i - \mathbf{m}_{k*}\|^n \\ &= \|\mathbf{x}_i - \mathbf{m}_{k*}\|^{n-p} + \sum_{j \neq k*} \frac{\|\mathbf{x}_i - \mathbf{m}_{k*}\|^n}{\|\mathbf{x}_i - \mathbf{m}_j\|^p} \end{aligned} \tag{11}$$

Therefore the intuition behind this performance function is that it is minimised when the distance between the data and the closest prototype is minimal and the other prototypes are scattered widely

throughout the data to give the largest possible value in the denominator of the second part. To create a learning rule, we calculate

$$\begin{aligned}\frac{\partial J_{IWK}(\mathbf{x}_i)}{\partial \mathbf{m}_{k*}} &= -(n-p)(\mathbf{x}_i - \mathbf{m}_{k*}) \\ &\quad \times \|\mathbf{x}_i - \mathbf{m}_{k*}\|^{n-p-2} - n(\mathbf{x}_i - \mathbf{m}_{k*}) \\ &\quad \times \|\mathbf{x}_i - \mathbf{m}_{k*}\|^{n-2} \sum_{j \neq k*} \frac{1}{\|\mathbf{x}_i - \mathbf{m}_j\|^p} \\ &= (\mathbf{x}_i - \mathbf{m}_{k*}) a_{ik*} \qquad (12)\end{aligned}$$

$$\begin{aligned}\frac{\partial J_{IWK}(\mathbf{x}_i)}{\partial \mathbf{m}_j} &= p(\mathbf{x}_i - \mathbf{m}_j)\frac{\|\mathbf{x}_i - \mathbf{m}_{k*}\|^n}{\|\mathbf{x}_i - \mathbf{m}_j\|^{p+2}} \\ &= (\mathbf{x}_i - \mathbf{m}_j) b_{ij} \qquad (13)\end{aligned}$$

We have found that positive values of p some prototypes tend to ∞ and so we always use negative values. In the experimental section we use $p = -1$ and $p = -0.5$.

For a batch mode algorithm, we set the derivatives to 0 and solve. For the online algorithm, we update the prototypes proportional to these derivatives.

5.1.1. *Implementation (Online Mode)*

1. Initialization:
   — initialize the cluster prototype vectors $\mathbf{m}_1, \ldots, \mathbf{m}_K$ randomly.
2. Loop for M iterations:
   — from (12) and (13) and we have:

$$\begin{aligned}\mathbf{m}_{k^*}^{(new)} &= \mathbf{m}_{k^*} - \zeta \frac{\partial J_{IWK}}{\partial \mathbf{m}_{k*}} \\ &= \mathbf{m}_{k^*} - \zeta(\mathbf{x}_i - \mathbf{m}_{k^*}) a_{ik^*} \qquad (14) \\ \mathbf{m}_k^{(new)} &= \mathbf{m}_k - \zeta \frac{\partial J_{IWK}}{\partial \mathbf{m}_j} \\ &= \mathbf{m}_k - \zeta(\mathbf{x}_i - \mathbf{m}_k) b_{ik} \qquad (15)\end{aligned}$$

   — for each data vector $\mathbf{x}_i$, set

$$k^* = \arg\min_{k=1}^{K}(\|\mathbf{x}_i - \mathbf{m}_k\|)$$

   and update all the prototypes $\mathbf{m}_k$ as

$$\mathbf{m}_{k^*}^{(new)} = \mathbf{m}_{k^*} - \zeta a_{ik^*}(\mathbf{x}_i - \mathbf{m}_{k^*})$$

   where for $p = -1$

$$\begin{aligned}a_{ik^*} &= -(n+1)\|\mathbf{x}_i - \mathbf{m}_{k^*}\|^{n-1} \\ &\quad - n\|\mathbf{x}_i - \mathbf{m}_{k^*}\|^{n-2} \sum_{j \neq k^*} \|\mathbf{x}_i - \mathbf{m}_j\| \\ \mathbf{m}_k^{(new)} &= \mathbf{m}_k - \zeta\left(\frac{-\|\mathbf{x}_i - \mathbf{m}_{k^*}\|^n}{\|\mathbf{x}_i - \mathbf{m}_k\|}\right)(\mathbf{x}_i - \mathbf{m}_k)\end{aligned}$$

   where ζ is a learning rate.

Note: if we want to choose a bigger value for n, we will need to choose smaller value for ζ (thus we tend to choose $n = 1$).

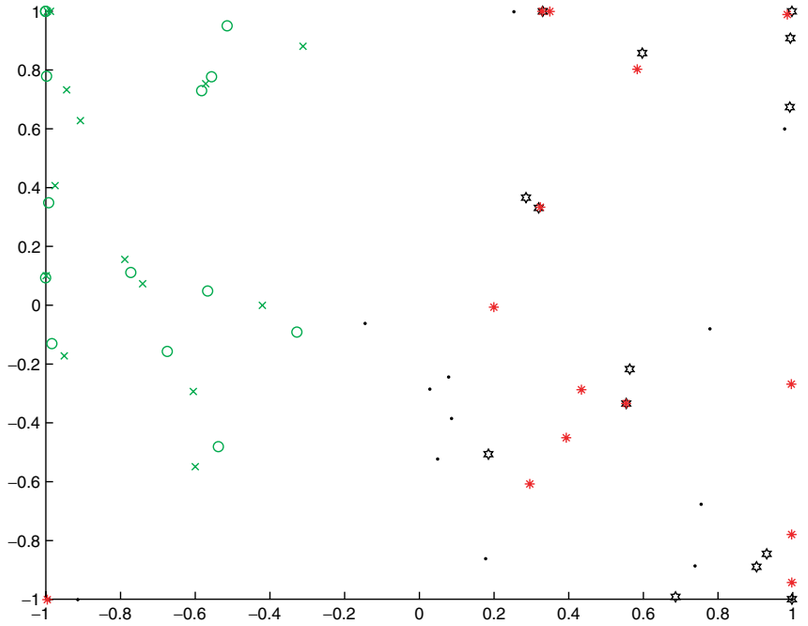

Fig. 5. IWKO ($p = -1$) results, the projections of the data onto the latent space.

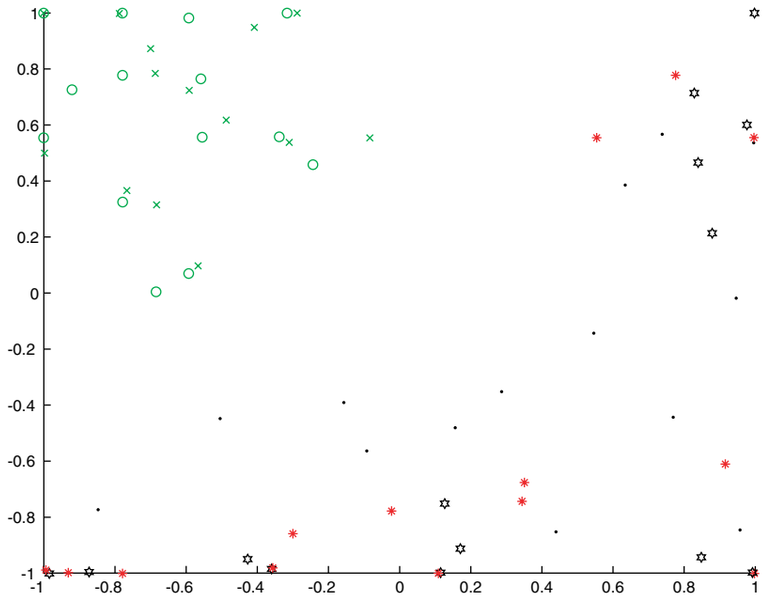

Fig. 6. IWKO ($p = -0.5$) results, the projections of the data onto the latent space.

5.1.2. *Simulation*

We have applied IWKO algorithm with the same underlying latent structure to the same real data set. We show two separate projections in Figs. 5 and 6; in both figures, we can see that we have two separate groups. 'oh' and 'wu' overlap but are separated from the others. In the first figure, more of the space is used to visualise the data but arguably the second is slightly better in that the 'oh' and 'wu' group is very tight and clearly separate from the others and also that 'ee' is somewhat separate from the other two open mouthed vowels.

6. 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.

We have subsequently shown that an alternative mapping based on a different criterion also provides a better mapping than the Self-Organizing Map (better in the sense that the two groups of vowels are clearly identified).

This second mapping was also based on the same underlying latent space that the Generative Topographic Mapping uses which suggests that there may be inherent advantages in having self-organisation based on the mapping from underlying latent spaces rather than self-organisation based on the learning rule as in Kohonen's algorithm. Future work will investigate such mappings based on Bregman divergences, special cases of which are mean squared error and Kullback-Leibler divergences.

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. 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).

3. C. Fyfe, A comparative study of two neural methods of exploratory projection pursuit, *Neural Networks* **10**(2) (1997) 257–262.
4. E. Corchado, D. MacDonald and C. Fyfe, Maximum and minimum likelihood hebbian learning for exploratory projection pursuit, *Data Mining and Knowledge Discovery* **8** (2004) 203–225.
5. Tuevo Kohonen, *Self-Organising Maps* (Springer, 1995).
6. T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning* (Springer, 2001).
7. C. Fyfe, Two topographic maps for data visualization, *Data Mining and Knowledge Discovery*, **14** (2007) 207–224. [ISSN 1384–5810].
8. W. Barbakh, *Local versus Global Interactions in Clustering Algorithms*. PhD thesis, School of Computing, University of the West of Scotland (2008).
9. R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, Adaptive mixtures of local experts, *Neural Computation* **3** (1991) 79–87.
10. M. I. Jordan and R. A. Jacobs, Hierarchical mixtures of experts and the em algorithm, *Neural Computation* **6** (1994) 181–214.
11. 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* (2001) 227–234.
12. C. Williams and F. V. Agakov, Products of gaussians and probabilistic minor components analysis. Technical Report EDI-INF-RR-0043, University of Edinburgh (2001).
13. W. Barbakh and C. Fyfe, Local vs global interactions in clustering algorithms, *International Journal of Knowledge Based Intelligent Engineering Systems* (2008).
14. W. Barbakh, M. Crowe and C. Fyfe, A family of novel clustering algorithms. In *7th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL2006* (2006).