Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp219-224)

Matching the dimensionality of maps with that of the data

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

Abstract Topographic maps are low dimensional maps which retain some topology of the original dataset. Many topographic mappings suffer from having the dimensionality of the map determined beforehand which is certain to be inappropriate for some data sets. In this paper, we develop a method of investigating a data set enabling the local dimensionality of the map to change. Our model of the data allows us to traverse the main manifold on which the data lies while giving information about the local dimensionality of the data around this main manifold.

Key-words: Data dimensionality, Generative Topographic Map.

1 Introduction

We are interested in exploratory data analysis and use unsupervised learning in order to determine some structure in high dimensional data sets. In this paper, we investigate a method of visualising high dimensional data sets. We have previously [4, 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, particularly those known as topographic mappings.

There are several mappings of a data set which are designed to retain some topographic features of the data set. Perhaps the most famous is Kohonen's Self-Organizing Map (SOM) [7]. A more recent innovation is the Generative Topographic Mapping (GTM) [1] which has been described as a principled alternative to the SOM and the Topographic Product of Experts [5] which is a near relative of the GTM. All of these mappings try to map a high dimensional data space onto a lower dimensional representation which retains some features of the data which are present in the original space. Thus points which are close together in data space should be close together in the low dimensional representation while points which are distant in data space should have very different representations. The above maps have the latter property but cannot guarantee the former as illustrated in Figure 1. In this figure, a one dimensional (GTM) mapping is attempting to represent data from a uniform distribution in $[-1, 1] \times [-1, 1]$. Since the dimensionality of the data does not match the dimensionality of the map, it is inevitable that some points which are actually close together will be given representations which are far apart.

It is not enough to say that we must get a representation whose dimensionality exactly fits the data since the data may lie on a nonlinear manifold whose dimensionality is not constant. This is the problem we seek to address in this paper. Intuitively, we string a one dimensional GTM through the centre of the data, allowing it to follow the manifold by finding the main mass of data while using the centres of the GTM to act as knot points from which we extend local probes to determine the dimensionality of the local data. In the next section, we review the GTM before discussing "Gaussian pancakes" and their application to solving the problem of finding the local dimensionality of the mapping.

2 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 mapped through a set of M basis functions and a set of adjustable weights,W, to the data space. In the GTM, the parameters W and β (see below) are updated using the EM algorithm to maximise the likelihood of the data *under this model.*

In detail, the underlying structure of the experts can be represented by K latent points, $t_1, t_2, \cdots, t_K$ which are positioned in a latent space of low dimensionality. Typically, the latent points will lie on a line in a one dimensional space or on the corners of a grid in two dimensional space. 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 changed 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 latent point is mapped to a point in data space, $\mathbf{m}_j = (\Phi_j W)^T$ which acts as the centre of an isotropic Gaussian which is the local probability density of the data.

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

where β is the inverse variance of the map. To change W, we consider a specific data point, say $\mathbf{x}_i$. We calculate the current responsibility of the j^{th} latent point for this data point,

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

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 (2) is zero), we set $r_{ij} = \frac{1}{K}, \forall j$.

The parameters of the combined mapping are adjusted to make the data as likely as possible under this mapping. [1] assume that each of the latent points has equal probability and so

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

i.e. all the data is assumed to be noisy versions of the mapping of the latent points.

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.

The GTM suffers from a common problem with topographic mappings - the latent points determine the topography *a priori* and the model is then made to fit the data as well as possible.

3 Gaussian pancakes

[8] discuss the covariance structure of a D-dimensional Gaussian pancake: intuitively

such a structure has large (equal) variance in $D-1$ dimensions and a small variance in the final dimension. Let C be the covariance matrix of a data set. If we perform a principal component analysis of this data set to get the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_D$ and associated eigenvalues $\lambda_1, \lambda_2, ..., \lambda_D$, then the model discussed in [8] has $\lambda_1 = \lambda_2 = ... = \lambda_{D-1} >> \lambda_D$ and the covariance matrix can be written as

$$C = \sum_{i=1}^{D-1} \mathbf{v}_i \mathbf{v}_i^T \lambda_1 + \mathbf{v}_D \mathbf{v}_D^T \lambda_D$$

Alternatively we may write

$$C^{-1} = \sum_{i=1}^{D-1} \mathbf{v}_i \mathbf{v}_i^T \beta_0 + \mathbf{v}_D \mathbf{v}_D^T \beta_D$$

where $\beta_0 = \lambda_1^{-1}$ and $\beta_D = \lambda_D^{-1}$. This may be generalised to pancakes with different numbers of small variance directions so that $\lambda_1 = \lambda_2 = ... = \lambda_{D-m} >> \lambda_{D-m+1} > ... > \lambda_D$ i.e. we have $D-m$ directions with large variance and m directions with small variance. Thus we may write

$$C^{-1} = \beta_0 I_D + \sum_{i=D-m+1}^{D} \mathbf{v}_i \mathbf{v}_i^T (\beta_i - \beta_0)$$

where I_D is the D-dimensional identity matrix.

The Gaussian pancake used with a product of experts is very elegantly associated with Minor Components Analysis in [8].

4 Dimensionality Matching

We are going to augment the GTM with Gaussian pancakes in the following manner. We begin with a one dimensional GTM which we string through the data set, initially along its first principal component. The projected latent points form the centres of Gaussian pancakes which determine the probability of the data given the map. We iterate the following steps

1. Calculate the local covariance matrix around the centres; express as a Gaussian pancake.
2. Use this pancake to calculate the responsibilities of the latent points for the data.
3. Use the responsibilities to calculate the new centres of the GTM.

Convergence is fast (typically 3 or 4 iterations). Step 2 is the E step for the GTM and step 3 is the standard M step for the GTM. Step 1 calculates the parameters for the local pancake.

Initially, every latent point is given equal responsibility for the data which we use to calculate the covariance matrix of the data taking account of responsibilities:

$$\Sigma_k = \sum_{n=1}^{N} r_{kn} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T$$

where r_{kn} is the responsibility which the k^{th} latent point has for the n^{th} data point (initially $\frac{1}{K}$. The Gaussian pancakes are fitted to this covariance matrix as above to get the parameters, $\beta_0^{(k)}, \beta_i^{(k)}$, and $\mathbf{v}_i^{(k)}$ for each pancake. We also estimate the local dimensionality of the data as $D-m$, the directions with large variance. Then for data point, $\mathbf{x}_n$,

$$P(\mathbf{x}_n|k) = \frac{1}{Z_k} \exp(-(\mathbf{x}_n - \mathbf{m}_k)^T C_k^{-1} (\mathbf{x}_n - \mathbf{m}_k))$$

where C_k is the covariance matrix of the k^{th} pancake and $\mathbf{m}_k$ is the centre determined by the projection of the k^{th} latent point into data space. $Z_k \propto C_k$ is the normalisaton factor. As with the GTM, we could assume that all the latent points have equal probability, so that

$$P(\mathbf{x}_n) = \sum_{k=1}^{K} \frac{1}{K} \frac{1}{Z_k} \exp(-(\mathbf{x}_n - \mathbf{m}_k)^T C_k^{-1} (\mathbf{x}_n - \mathbf{m}_k))$$

However trained GTMs are known not to fit this assumption and so we opt to calculate $P(k)$ from $\int d\mathbf{x} P(k|\mathbf{x}) P(\mathbf{x})$, where the last term is the Dirac δ function. This is the second major change from the standard GTM, in that we are not relying wholly on a generative model; we are giving some acknowledgement to the existence of the data. There are several models which combine top-down generative models with bottom-up data driven modelling e.g. [6, 9]. We then use this in

$$P(\mathbf{x}_n) = \sum_{k=1}^{K} P(k) \frac{1}{Z_k} \exp(-(\mathbf{x}_n - \mathbf{m}_k)^T C_k^{-1} (\mathbf{x}_n - \mathbf{m}_k))$$

We wish to maximise the likelihood of the data under this model, and so, assuming the data are drawn independently from the data distribution, we use the standard GTM cost function

$$L = \sum_{n=1}^{N} \log(P(\mathbf{x}_n)) \quad (3)$$

We use the EM [3] algorithm to do so having as the E-step, the calculation of the parameters of the model using auxiliary variables denoting the responsibility which each pancake has for each data point:

$$r_{kn} = P(k|\mathbf{x}_n) = \frac{P(\mathbf{x}_n|k)P(k)}{\sum_{j=1}^{K} P(\mathbf{x}_n|j)P(j)} \quad (4)$$

which are then used to calculate values for the means and hence the mapping W using [1]

$$W_{new}^T = (\Phi^T G \Phi)^{-1} \Phi^T R X \quad (5)$$

where R is the matrix of responsibilities, G is a diagonal matrix with $G_{ii} = \sum_j r_{ij}$ and X is the $N \times D$ data matrix. We also recalculate the parameters of the individual covariance matrices

$$\Sigma_k = \sum_{n=1}^{N} r_{kn} ||\mathbf{m}_k - \mathbf{x}_n||^2 \quad (6)$$

Note that these two operations can be performed independently of each other. We then return to re-fit the Gaussian pancakes to these covariance matrices separately so that we have new parameters $\beta_0^{(k)}, \beta_i^{(k)}$, and $\mathbf{v}_i^{(k)}$ for each pancake and the process begins again. An alternative to (6) is to have a cut off point and include, for each covariance matrix, only data which have responsibility greater than a particular value. Both methods result in similar maps.

We note that no real data is liable to have $\lambda_1 = \lambda_2 = ... = \lambda_{D-m}$ and so we approximate β_0 with the inverse of the average of the first $D - m$ eigenvalues. Also, we have found in practice that it is beneficial to use regularisation when we are inverting the variances.

5 Simulations

We begin with an example from an artificial data set: we create 3 dimensional data from $\{t, t + \cos(t) + \mu_1, t + 2\sin(t) + \mu_2\}$ where t is drawn from a uniform distribution in $[0, 2\pi]$ and μ_i are drawn from a zero mean Gaussian of standard deviation 0.3. We create a one dimensional GTM with Gaussian pancakes centred on the projections of 20 latent points into data space and achieve the result shown in Figure 2.

We see that the mapping has captured the gross topography of the data.

However this experiment, while useful for visualisation of the mapping, does not exhibit the ability to match different dimensionalities in different parts of the manifold; for this, we create 5 dimensional data of the form $\{t, t + \cos(t), t + 2\sin(t-1), \cos(2t) - \sin(t+1), \sin(2t) + t\}$ where t is drawn from a uniform distribution in $[0, 2\pi]$. This data, though 5 dimensional, lies on a one dimensional manifold. We add varying degrees of noise so that the manifold is one dimensional at one end, becomes two dimensional, then three and so on up to 5 dimensional. We train a one dimensional GTM with pancakes as above. The local values of m are 1,2, 1,1,1,2,1,1,2,2,2,2,2,2,2,3,3,3,4,4 i.e. the mapping is adapting to the local dimensionality of the data. Typically we will have 5 non-zero variances for the pancakes at one end of the GTM while we see a single non-zero variance at the other end.

6 Conclusion

We have illustrated an extension to the GTM which is designed to be a first step in an exploratory data investigation. We use a one dimensional GTM to create a core manifold through the data set and use the projections of the latent points as local centres with which to investigate the local data dimensionality. We have made two significant changes to the GTM

1. We have used Gaussian pancakes with parameters derived locally instead of a single variance parameter, β.
2. We have given some priority to the data in our calculation of $P(k|\mathbf{x})$ rather than have a wholly generative model where this is prescribed *a priori*.

We have shown the algorithm performing with two artificial data sets.

This algorithm should be seen as an exploratory tool: an investigator will move sequentially along the GTM centres (the projections of the latent points in data space) investigating the projections of the local data onto the Gaussian pancakes which are of relatively low dimension. Within this local investigation, a number of options are possible. For example, our future work will consider adding secondary GTM probes within the local Gaussian pancakes and attempt to match these between proximate pancakes.

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] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* Wiley-interscience, (second edition) edition, 2001.

[4] C. Fyfe. A comparative study of two neural methods of exploratory projection pursuit. *Neural Networks*, 10(2):257–262, 1997.

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

[6] G.E. Hinton, P. Dayan, B.J. Frey, and R.M. Neal. The 'wake-sleep' algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

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

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

[9] L. Xu. Byy harmony learning, structural rpcl, and topological self-organizing on mixture models. *Neural Networks*, 15:1125–1151, 2002.

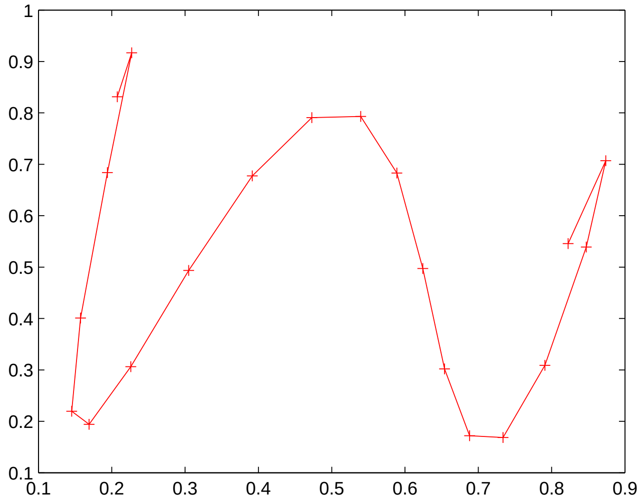

Figure 1: The centres of a one dimensional GTM trained on data drawn iid from $[0, 1] \times [0, 1]$. The GTM quantises the data but loses its topology-preserving properties.

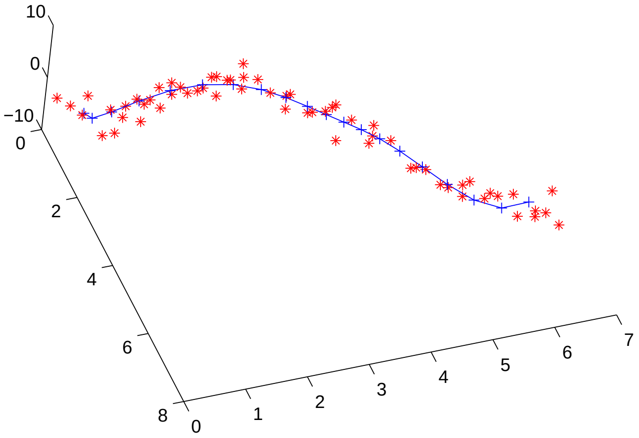

Figure 2: The GTM centres go through the middle of the data set.