Magnification Factors for the GTM Algorithm

Christopher M. Bishop, Markus Svensén and Christopher K. I. Williams

Neural Computing Research Group, Aston University, UK

Abstract

The Generative Topographic Mapping (GTM) algorithm of Bishop *et al.* (1996) has been introduced as a principled alternative to the Self-Organizing Map (SOM). As well as avoiding a number of deficiencies in the SOM, the GTM algorithm has the key property that the smoothness properties of the model are decoupled from the reference vectors, and are described by a continuous mapping from a lower-dimensional latent space into the data space. Magnification factors, which are approximated by the difference between code-book vectors in SOMs, can therefore be evaluated for the GTM model as continuous functions of the latent variables using the techniques of differential geometry. They play an important role in data visualization by highlighting the boundaries between data clusters, and are illustrated here for both a toy data set, and a problem involving the identification of crab species from morphological data.

1 The GTM Algorithm

Many algorithms have been proposed which seek a representation of a multi-dimensional data set in terms of a reduced number of dimensions. One of the best known is the Self-Organizing Map (SOM) algorithm of Kohonen (1982). However, the SOM suffers from a number of limitations including the absence of a well-defined cost function, the lack of any guarantee of 'self-organisation' or of convergence, and the absence of a probability density function. We have introduced the *Generative Topographic Mapping* algorithm (Bishop *et al.*, 1996) as a principled alternative to the SOM which overcomes these limitations. For completeness, and to establish notation, we begin with a brief overview of the GTM algorithm.

Our goal is to model a probability distribution in a D-dimensional data space in terms of a smaller number, L, of latent, or 'hidden', variables. We denote the coordinates of the data space by $\mathbf{t} = (t_1, \ldots, t_D)^{\mathrm{T}}$ and the latent variables by $\mathbf{x} = (x_1, \ldots, x_L)^{\mathrm{T}}$. The model will be trained using a set of N data vectors $\mathbf{t}_1, \ldots, \mathbf{t}_N$. We first define a non-linear mapping from latent space to data space of the form

$$\mathbf{y} = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) \tag{1}$$

where $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_M)^{\mathrm{T}}$ represents a set of M fixed non-linear basis functions, and $\mathbf{W}$ is a $D \times L$ matrix of parameters. The mapping (1) defines an L-dimensional non-Euclidean manifold $\mathcal{S}$ embedded in the D-dimensional Euclidean data space, as illustrated in Figure 1. A typical choice for the basis functions would be a set of Gaussians centred on a regular grid in latent space, with a common width parameter whose value controls the degree of smoothness of the manifold in data space.

If we introduce a probability distribution $p(\mathbf{x})$ over latent space, then (1) induces a corresponding distribution in data space which will be confined to the L-dimensional manifold. Since our data

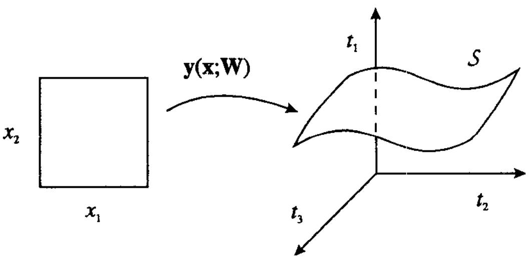

Figure 1: The non-linear mapping $\mathbf{y}(\mathbf{x};\mathbf{W})$ from the L-dimensional latent space $\mathbf{x}$ to the D-dimensional data space $\mathbf{t}$ defines an L-dimensional non-Euclidean manifold $\mathcal{S}$.

will not live exactly on such a manifold, we convolve this distribution with an isotropic Gaussian distribution in data space of the form

$$p(\mathbf{t}|\mathbf{x},\mathbf{W},\beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x};\mathbf{W}) - \mathbf{t}\|^2\right\}. \quad (2)$$

The distribution in $\mathbf{t}$-space, for given values of $\mathbf{W}$ and β, is then obtained by integration over the $\mathbf{x}$-distribution

$$p(\mathbf{t}|\mathbf{W},\beta) = \int p(\mathbf{t}|\mathbf{x},\mathbf{W},\beta)p(\mathbf{x})\,d\mathbf{x}. \quad (3)$$

The GTM algorithm corresponds to a particular form of this model in which we consider $p(\mathbf{x})$ to be a sum of delta functions centred on the nodes of a regular lattice in latent space

$$p(\mathbf{x}) = \frac{1}{K}\sum_{l=1}^{K}\delta(\mathbf{x} - \mathbf{x}_l). \quad (4)$$

Note that this lattice is typically much finer than the grid of points used to define the centres of the basis functions. Each point $\mathbf{x}_l$ is then mapped to a corresponding point $\mathbf{y}(\mathbf{x}_l;\mathbf{W})$ in data space, which then forms the centre of a Gaussian density function. From (3) and (4) we see that the distribution function in data space then takes the form

$$p(\mathbf{t}|\mathbf{W},\beta) = \frac{1}{K}\sum_{l=1}^{K}p(\mathbf{t}|\mathbf{x}_l,\mathbf{W},\beta) \quad (5)$$

which represents a mixture of Gaussians in which the centres of the Gaussian functions are constrained to lie in the L-dimensional manifold $\mathcal{S}$. The parameters $\mathbf{W}$ and β can be determined by maximum likelihood using the EM (expectation-maximization) algorithm (Dempster *et al.*, 1977; Bishop, 1995). Often the latent space is chosen to be two-dimensional so that the algorithm can be applied to the problem of data visualization. The latent space density $p(\mathbf{x})$ can be regarded as a prior distribution, with the corresponding posterior distribution $p(\mathbf{x}|\mathbf{t},\mathbf{W},\beta)$, for a given data point $\mathbf{t}$, given by Bayes' theorem. For a two-dimensional latent space this posterior distribution can be visualized using, for example, pseudo-colour. In order to visualize a *set* of data points, each of the corresponding posterior distributions can conveniently be summarized by its mean (or mode), which is easily evaluated.

2 Magnification Factors

The concept of a magnification factor arose originally in the context of topographic maps in the brain, such as those found in the visual and somatosensory areas of the cortex, where it relates the two-dimensional spatial density of sensors to the two-dimensional spatial density of the corresponding cortical cells. In the context of data analysis, the analogous concept plays an equally important role. When a small region of the latent space is mapped to data space it may be compressed or stretched as the mapping is optimized to fit the data. One consequence of this is that well-separated clusters of points in data space will appear to be more nearly uniform in latent space, and so inhomogeneities in the data can be obscured.

This problem has been addressed in the context of the SOM by Ultsch (1993) who uses a gray-scale scheme to display the Euclidean distances between code-book vectors on the visualization plot. This necessarily gives a discrete representation of the local magnification since the effective surface in data space for the SOM is defined only in terms of the positions of the code-book vectors. A key difference between GTM and the SOM is that in the GTM algorithm the lower-dimensional manifold $\mathcal{S}$ in data space is defined directly by the mapping (1), while in the SOM it is determined only indirectly by the locations of the finite number of code-book vectors. We now show how the local magnification factor for GTM can be evaluated in terms of the mapping (1) using the techniques of differential geometry.

Consider a standard set of Cartesian coordinates x^i in the latent space. Since each point P in latent space is mapped by a continuous function to a corresponding point P' in data space, the mapping defines a set of curvilinear coordinates ξ^i in the manifold in which each point P' is labelled with the coordinate values $\xi^i = x^i$ of P, as illustrated in Figure 2. Throughout this paper we shall use the standard notation of differential geometry in which raised indices denote contravariant components and lowered indices denote covariant components, with an implicit summation over pairs of repeated covariant-contravariant indices.

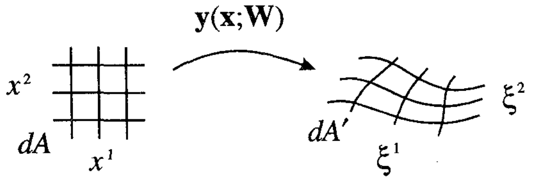

Figure 2: This diagram shows the mapping of the Cartesian coordinate system x^i in latent space onto a curvilinear coordinate system ξ^i in the L-dimensional manifold $\mathcal{S}$.

Our goal is to find an expression for the area[1] dA' of the region of $\mathcal{S}$ corresponding to an infinitesimal rectangle in latent space with area $dA = \prod_i dx^i$. We first discuss the metric properties of the manifold $\mathcal{S}$. Consider a local transformation, at some point P' in $\mathcal{S}$, to a set of rectangular Cartesian coordinates $\zeta^i = \zeta^i(\boldsymbol{\xi})$. Then the squared length element in these coordinates is given by

$$\begin{aligned} ds^2 &= \delta_{\mu\nu} d\zeta^\mu d\zeta^\nu \\ &= \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j} d\xi^i d\xi^j \\ &= g_{ij} d\xi^i d\xi^j \end{aligned} \tag{6}$$

where g_{ij} is the metric tensor, which is therefore given by

$$g_{ij} = \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j} \tag{7}$$

and we are implicitly summing over repeated indices. The area element in the manifold $\mathcal{S}$ can be related to the corresponding area element in the latent space by the determinant of the Jacobian of the transformation $\xi \to \zeta$

$$\begin{aligned} dA' &= \prod_\mu d\zeta^\mu = J \prod_i d\xi^i \\ &= J \prod_i dx^i = J dA \end{aligned} \tag{8}$$

where the determinant J of the Jacobian is given by

$$J = \det\left(\frac{\partial \zeta^\mu}{\partial \xi^i}\right) = \det\left(\frac{\partial \zeta^\mu}{\partial x^i}\right). \tag{9}$$

We now introduce the determinant g of the metric tensor which we can write in the form

$$\begin{aligned} g &= \det(g_{ij}) \\ &= \det\left(\delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial x^i} \frac{\partial \zeta^\nu}{\partial x^j}\right) \\ &= \det\left(\frac{\partial \zeta^\mu}{\partial x^i}\right) \det\left(\frac{\partial \zeta^\nu}{\partial x^j}\right) \\ &= J^2 \end{aligned} \tag{10}$$

and so, using (8), we obtain an expression for the area element in curvilinear coordinates in the form

$$\frac{dA'}{dA} = J = \sqrt{g}. \tag{11}$$

[1] We shall talk about area since we are mainly interested in the case $L = 2$. In fact our derivation is equally applicable for $L > 2$.

We now seek an expression for g in terms of the non-linear mapping (1). Consider again the squared length element ds^2 lying within the manifold $\mathcal{S}$. Since $\mathcal{S}$ is embedded within the Euclidean data space, this also corresponds to the squared length element of the form

$$\begin{aligned} ds^2 &= \delta_{kl} dy^k dy^l \\ &= \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j} dx^i dx^j \\ &= g_{ij} dx^i dx^j \end{aligned} \qquad (12)$$

and so we have

$$g_{ij} = \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j}. \qquad (13)$$

Using (11) we then obtain

$$\frac{dA'}{dA} = \det{}^{1/2} \left(\delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j} \right). \qquad (14)$$

Making use of the expression (1) we can write this explicitly in terms of the derivatives of the basis functions $\phi_j(\mathbf{x})$ in the form

$$\frac{dA'}{dA} = \det{}^{1/2} \left(\boldsymbol{\psi}^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \boldsymbol{\psi} \right) \qquad (15)$$

where $\boldsymbol{\psi}$ has elements $\psi_{ji} = \partial \phi_j / \partial x^i$.

3 Results: Toy Data

As a simple illustration of the evaluation of the local magnification factor for the GTM algorithm we consider a data set consisting of 400 data points drawn from a mixture of two Gaussians in two dimensions, shown in Figure 3.

The corresponding visualization plot is shown in Figure 4 in which each data point is represented by the mean of the corresponding posterior probability distribution. This figure also shows the corresponding magnification factor plotted as a function of the latent space coordinates. It can be seen that, while the data points form well separated clusters in the original data space, they appear to be much more uniformly distributed when viewed in the latent space. This is a consequence of the model adapting to give a good representation of the distribution in data space, and tends to obscure the presence of distinct clusters. However, by superimposing the magnification factor dA'/dA, as a function of $\mathbf{x}$, over the latent space, we can see that the central region of the map suffers a relatively large magnification on projection to the data space (corresponding to the region between the clusters where the data are sparse) and so appears as a darker band on the right-hand plot in Figure 3. Such darker regions thus serve to delineate the boundaries of clusters.

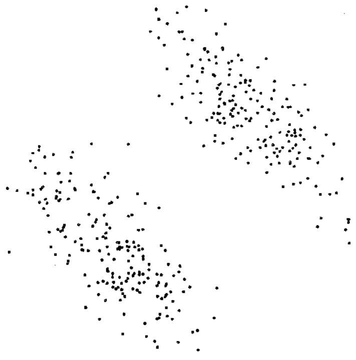

Figure 3: The toy data set in two dimensions, consisting of 400 data points generated from a mixture of two Gaussians.

4 Results: Crabs Data

As a second illustration of magnification factors we consider a data set[2] of measurements taken from the genus *Leptograpsus* of rock crabs. Measurements were taken from two species classified by their colour (orange or blue) with the aim of discovering morphological differences which would allow preserved specimens (which have lost their colour) to be distinguished. The data set con-

[2] Available from Brian Ripley at: `http://markov.stats.ox.ac.uk/pub/PRNN`.

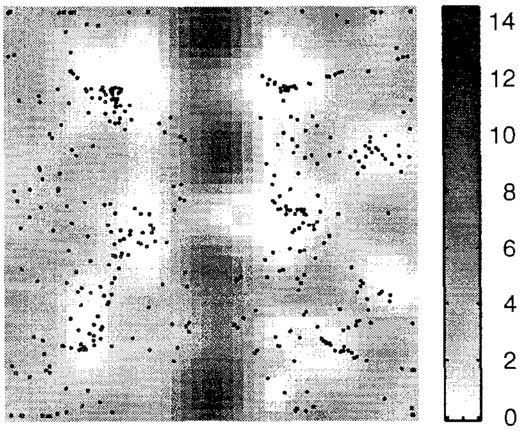

Figure 4: This shows the toy data set visualized in the latent space of a trained GTM model, with the local magnification factor superimposed using a grey-scale representation. Darker shades correspond to high values of dA'/dA while lighter shades correspond to low values.

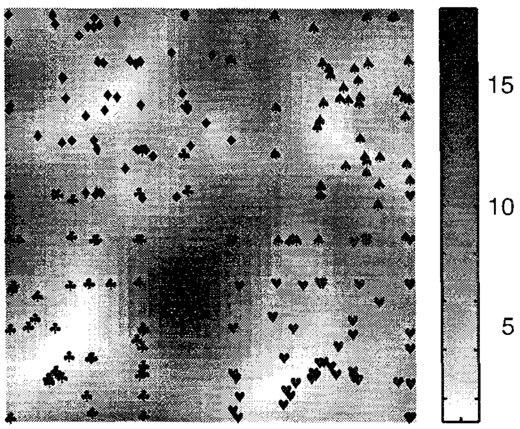

Figure 5: Plot of the latent-space distribution of the crabs data, in which ♣ denotes blue males, ♦ denotes blue females, ♥ denotes orange males, and ♠ denotes blue females. The grey-scale background shows the corresponding magnification factor as a function of the latent space coordinates.

tains 50 examples of each sex from each species, and the measurements correspond to length of frontal lip, rear width, length along mid-line, maximum width of carapace, and body length. Since all of the variables correspond to length measurements, the dominant feature of the crabs data is an overall scaling of the data vector in relation to the size of the crab. To remove this effect each data vector $\mathbf{t}_n = (t_{1n}, \ldots, t_{Dn})^{\mathrm{T}}$ is normalized to unit mean, so that

$$\widetilde{t}_{kn} = t_{kn} \Bigg/ \sum_{k'=1}^{D} t_{k'n}. \qquad (16)$$

Results from the crabs data are shown in Figure 5. It can be seen that the two species form distinct clusters, with the manifold undergoing a relatively large stretching in the region between them. Within each cluster there is a partial separation of males from females. Ripley (1996) shows a visualization of the SOM code-book vectors for the crab data using the representation of Ultsch (1993), which corresponds to a rough discrete approximation to the magnification factors of the GTM model.

5 Discussion

One of the key differences between GTM and SOM is that in the GTM algorithm the definition of the manifold is independent of the Gaussian centres, whereas in SOM a manifold is defined only by the discrete set of code-book vectors, and requires some arbitrary form of interpolation to specify the location of the manifold at other points. In this paper we have shown how the techniques of differential geometry allow the local magnification factor for the GTM algorithm to be computed as a continuous function of the latent variables. We have also shown how this magnification factor augments the posterior latent space plot by providing important information on the clustering properties of the data.

We note that, although the magnification factor represents the extent to which areas are magnified on projection to the data space, it gives no information about which directions in latent space correspond to the stretching. Also, stretching in one direction may be compensated by compression in the orthogonal direction, and such distortion would therefore not be apparent from the magnification factor alone. We can recover this information by considering the decomposition of the metric tensor in terms of its eigenvectors and eigenvalues. It is convenient to display the information by selecting a regular grid in latent space (which could correspond to the reference vector grid, but could also be much finer) and to plot at each grid point an ellipse with principal axes oriented according to the eigenvectors, with principal radii given by the square roots of the eigenvalues. This is illustrated for the crabs data in Figure 6. The standard area magnification factor is given from (11) by the square root of the product of the eigenvalues, and so corresponds to the area of the ellipse.

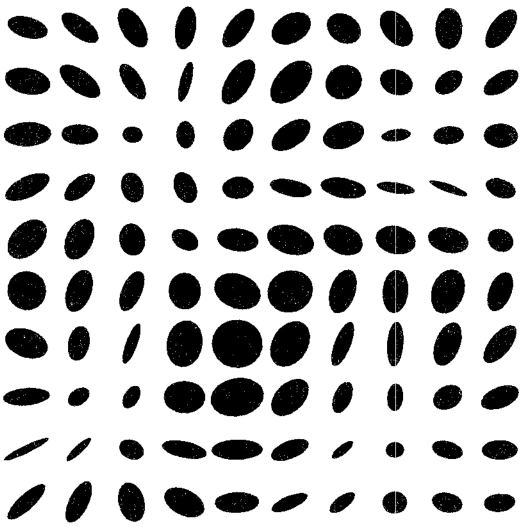

Figure 6: Plot of the local magnification factor for the crabs data, using the ellipse representation discussed in the text.

It should also be noted that, for the batch version of the self-organizing map, it is possible to define an interpolating surface by interpreting the reference vector update equations of the SOM as a kernel smoother (Mulier and Cherkassky, 1995). For a differentiable neighbourhood function, it is then straightforward to apply the techniques developed in this paper.

Acknowledgements

This work was supported by EPSRC grant GR/K51808: *Neural Networks for Visualisation of High-Dimensional Data.* Papers relating to the original GTM algorithm, as well as software implementations of GTM and data sets used in the development of GTM, can be found at `http://www.ncrg.aston.ac.uk/GTM/`.

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press.

Bishop, C. M., M. Svensén, and C. K. I. Williams (1996). GTM: the generative topographic mapping. Accepted for publication in *Neural Computation.* Available as NCRG/96/015 from `http://www.ncrg.aston.ac.uk/`.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B* **39** (1), 1–38.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**, 59–69.

Mulier, F. and V. Cherkassky (1995). Self-organization as an iterative kernel smoothing process. *Neural Computation* **7** (6), 1165–1177.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks.* Cambridge: Cambridge University Press.

Ultsch, A. (1993). Knowledge extraction from self-organizing neural networks. In O. Opitz, B. Lausen, and R. Klar (Eds.), *Information and Classification*, pp. 301–306. Berlin: Springer.