# Magnification Factors for the SOM and GTM Algorithms

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

Neural Computing Research Group,
Dept. of Computer Science and Applied Mathematics,
Aston University, Birmingham, U.K.

To appear in *Proceedings 1997 Workshop on Self-Organizing Maps*, Helsinki, Finland.
Available as technical report NCRG/97/008 from `http://www.ncrg.aston.ac.uk/`

**Abstract**

Magnification factors specify the extent to which the area of a small patch of the latent (or 'feature') space of a topographic mapping is magnified on projection to the data space, and are of considerable interest in both neuro-biological and data analysis contexts. Previous attempts to consider magnification factors for the self-organizing map (SOM) algorithm have been hindered because the mapping is only defined at discrete points (given by the reference vectors). In this paper we consider the batch version of SOM, for which a continuous mapping can be defined, as well as the Generative Topographic Mapping (GTM) algorithm of Bishop *et al.* [2] which has been introduced as a probabilistic formulation of the SOM. We show how the techniques of differential geometry can be used to determine magnification factors as continuous functions of the latent space coordinates. The results are illustrated here using a problem involving the identification of crab species from morphological data.

## 1 The Batch SOM Algorithm

We begin by reviewing the batch form of the SOM [4] and showing how it leads to a continuous mapping from latent space to data space. The batch SOM algorithm involves a set of $K$ reference vectors $\{\mathbf{y}_i\}$ defined in the data space, in which each vector $\mathbf{y}_i$ is associated with a node $i$ on a regular lattice in a (typically) two-dimensional latent space (often called a 'feature' space). We denote the coordinate system in latent space by $\mathbf{x}$, so that the $i$th node is at position $\mathbf{x}_i$. The algorithm begins by initializing the reference vectors using, for example, principal component analysis. At each cycle the corresponding 'winning node' $j(n)$ is identified for every data vector $\mathbf{t}_n$, corresponding to the reference vector $\mathbf{y}_i$ having the smallest Euclidean distance $\|\mathbf{y}_i - \mathbf{t}_n\|^2$ to $\mathbf{t}_n$. The reference vectors are then updated by setting them equal to weighted averages of the data points given by

$$\mathbf{y}_i = \frac{\sum_n h(\mathbf{x}_i, \mathbf{x}_{j(n)})\mathbf{t}_n}{\sum_n h(\mathbf{x}_i, \mathbf{x}_{j(n)})}. \tag{1}$$

in which $h(\mathbf{x}, \mathbf{x}')$ is the neighbourhood function, which we assume to be a continuous function of the latent space coordinates (a Gaussian is a common choice). The steps of identifying the winning nodes and updating the reference vectors are repeated iteratively. A key ingredient in the algorithm is that the 'width' of the neighbourhood function $h(\mathbf{x}, \mathbf{x}')$ starts with a relatively large value and is gradually reduced after each iteration.

As pointed out by Mulier and Cherkassky [5], the value of the neighbourhood function $h(\mathbf{x}_i, \mathbf{x}_{j(n)})$ depends only on the identity of the winning node $j$ and not on the value of the corresponding data vector $\mathbf{t}_n$. We can therefore perform partial sums over the groups $\mathcal{G}_j$ of data vectors assigned to each node $j$, and hence re-write (1) in the form

$$\mathbf{y}_i = \sum_j K(\mathbf{x}_i, \mathbf{x}_j)\mathbf{m}_j \tag{2}$$

where $\mathbf{m}_j$ is the mean of the vectors in group $\mathcal{G}_j$ and is given by

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{n \in \mathcal{G}_j} \mathbf{t}_n \tag{3}$$

in which $N_j$ is the number of data vectors in group $\mathcal{G}_j$, and

$$K(\mathbf{x}, \mathbf{x}_j) = \frac{N_j h(\mathbf{x}, \mathbf{x}_j)}{\sum_{j'} N_{j'} h(\mathbf{x}, \mathbf{x}_{j'})}. \tag{4}$$

The result (2) is analogous to the Nadaraya-Watson kernel regression formula [1] with the kernel functions given by $K(\mathbf{x}, \mathbf{x}_j)$.

Thus the batch SOM algorithm replaces the reference vectors at each cycle with a convex combination of the node means $\mathbf{m}_j$, with coefficients determined by the neighbourhood function. Note that the kernel coefficients satisfy $\sum_j K_{ij} = 1$ for every $i$. We see that the batch SOM update equations (2) define a natural, continuous mapping from latent space to data space, given by

$$\mathbf{y}(\mathbf{x}) = \sum_j K(\mathbf{x}, \mathbf{x}_j)\mathbf{m}_j \tag{5}$$

which coincides with the reference vectors $\mathbf{y}_i$ when $\mathbf{x} = \mathbf{x}_i$.

## 2 The GTM Algorithm

The goal of the GTM algorithm is to model a probability distribution in data space in terms of two 'latent' variables corresponding to the coordinates of the latent space. The non-linear mapping from latent space to data space is introduced explicitly in GTM in the form

$$\mathbf{y}(\mathbf{x}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{x}) \tag{6}$$

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 $D \times M$ matrix of parameters. The mapping (6) defines a two-dimensional non-Euclidean manifold $\mathcal{S}$ embedded in the $D$-dimensional Euclidean data space. 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 (6) induces a corresponding distribution in data space which will be confined to the two-dimensional manifold. Since our data will not live exactly on such a manifold, we convolve this distribution with an isotropic Gaussian distribution in data space of the form

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left\{-\frac{\beta}{2}\|\mathbf{y}(\mathbf{x}; \mathbf{W}) - \mathbf{t}\|^2\right\}. \tag{7}$$

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

$$p(\mathbf{t}|\mathbf{W}, \beta) = \int p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta)p(\mathbf{x})\, d\mathbf{x}. \tag{8}$$

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

$$p(\mathbf{x}) = \frac{1}{K} \sum_{l=1}^{K} \delta(\mathbf{x} - \mathbf{x}_l). \tag{9}$$

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 mapped to a corresponding point $\mathbf{y}(\mathbf{x}_l; \mathbf{W})$ in data space, which forms the centre of a Gaussian density function. From (8) and (9) we see that the distribution function in data space takes the form

$$p(\mathbf{t}|\mathbf{W}, \beta) = \frac{1}{K} \sum_{l=1}^{K} p(\mathbf{t}|\mathbf{x}_l, \mathbf{W}, \beta) \tag{10}$$

which represents a mixture of Gaussians in which the centres of the Gaussian functions are constrained to lie on the two-dimensional manifold $\mathcal{S}$. The parameters $\mathbf{W}$ and $\beta$ can be determined by maximum likelihood using the EM (expectation-maximization) algorithm [1]. 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. The SOM algorithm can be derived as an approximation to GTM in which the soft, probabilistic assignments of data points to nodes are replaced with hard 0/1 assignments, as discussed by Bishop *et al.* [2]

# 3 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 [7] who uses a gray-scale scheme to display the Euclidean distances between reference vectors on the visualization plot. This necessarily gives a discrete representation of the local magnification since the effective surface in data space for the standard SOM is defined only in terms of the positions of the reference vectors. We now show how the local magnification factor for the batch SOM and GTM algorithms can be evaluated as *continuous* functions of the latent space coordinates, in terms of the mapping $\mathbf{y}(\mathbf{x})$, 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 $\xi^i$ in the manifold in which each point $P'$ is labelled with the coordinate values $\xi^i = x^i$ of $P$, as illustrated in Figure 1. 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.

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

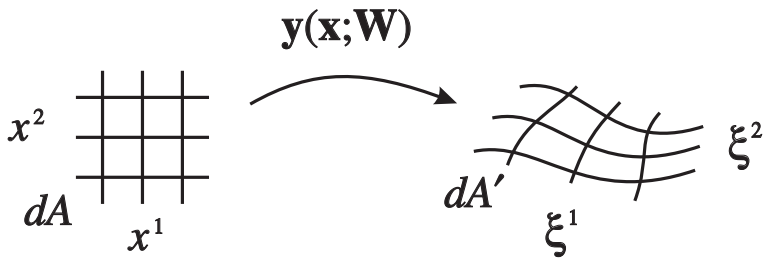


Figure 1: This diagram shows the mapping of the Cartesian coordinate system $x^i$ in latent space onto a curvilinear coordinate system $\xi^i$ in the $L$-dimensional manifold $\mathcal{S}$.

length element in these coordinates is given by

$$ds^2 = \delta_{\mu\nu} d\zeta^\mu d\zeta^\nu = \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j} d\xi^i d\xi^j = g_{ij} d\xi^i d\xi^j \tag{11}$$

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

$$g_{ij} = \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial \xi^i} \frac{\partial \zeta^\nu}{\partial \xi^j}. \tag{12}$$

We now seek an expression for $g$ in terms of the non-linear mapping $\mathbf{y}(\mathbf{x})$. 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

$$ds^2 = \delta_{kl} dy^k dy^l = \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j} dx^i dx^j = g_{ij} dx^i dx^j \tag{13}$$

and so we have

$$g_{ij} = \delta_{kl} \frac{\partial y^k}{\partial x^i} \frac{\partial y^l}{\partial x^j}. \tag{14}$$

For the batch SOM, the the metric tensor can be expressed explicitly in terms of the derivatives of the neighbourhood function using (4) and (5). Similarly, using (6) the metric tensor for GTM can be expressed in terms of the derivatives of the basis functions $\phi_j(\mathbf{x})$ in the form

$$\mathbf{g} = \boldsymbol{\psi}^{\mathrm{T}} \mathbf{W}^{\mathrm{T}} \mathbf{W} \boldsymbol{\psi} \tag{15}$$

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

Our goal is to find an expression for the area $dA'$ of the region of $\mathcal{S}$ corresponding to an infinitesimal rectangle in latent space with area $dA = \prod_i dx^i$. The area element in the manifold $\mathcal{S}$ can be related to the corresponding area element in the latent space by the Jacobian of the transformation $\xi \to \zeta$

$$dA' = \prod_\mu d\zeta^\mu = J \prod_i d\xi^i = J \prod_i dx^i = J dA \tag{16}$$

where the Jacobian $J$ is given by

$$J = \det \left( \frac{\partial \zeta^\mu}{\partial \xi^i} \right) = \det \left( \frac{\partial \zeta^\mu}{\partial x^i} \right). \tag{17}$$

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

$$g = \det(g_{ij}) = \det \left( \delta_{\mu\nu} \frac{\partial \zeta^\mu}{\partial x^i} \frac{\partial \zeta^\nu}{\partial x^j} \right) = \det \left( \frac{\partial \zeta^\mu}{\partial x^i} \right) \det \left( \frac{\partial \zeta^\nu}{\partial x^j} \right) = J^2 \tag{18}$$

and so, using (16), we obtain an expression for the volume element in curvilinear coordinates in the form

$$\frac{dA'}{dA} = J = \det{}^{1/2}\mathbf{g}. \tag{19}$$

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. We can recover this information by considering the decomposition of the metric tensor in terms of its eigenvectors and eigenvalues. As we shall see in the next section, it is convenient to display this 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. The standard area magnification factor is given from (19) by the square root of the product of the eigenvalues, and so corresponds to the area of the ellipse.

# 4 Results: Crabs Data

As an illustration of magnification factors we consider a data set[1] of measurements taken from the genus *Leptograpsus* of rock crabs [3]. 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 contains 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

$$\widetilde{t}_{kn} = t_{kn} \Big/ \sum_{k'=1}^{D} t_{k'n}. \tag{20}$$

Results from the crabs data are shown in Figure 2. It can be seen that the two species form

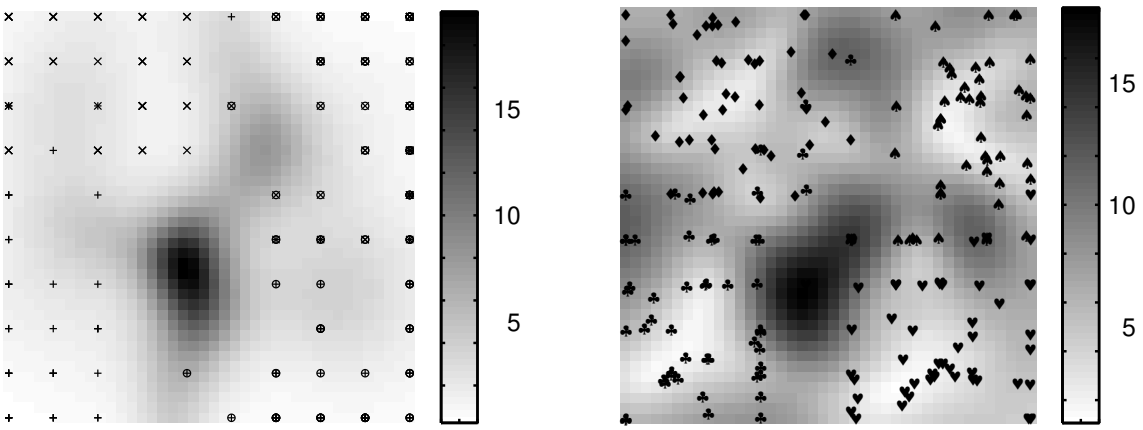


Figure 2: Plots of the latent-space distribution of the crabs data, in which + and × denote male and female blue crabs, while the circles and squares denote male and female orange crabs, respectively. Results for SOM is shown on the left and GTM on the right. The grey-scale background in each case shows the corresponding area magnification factor as a function of the latent space coordinates.

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

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. Corresponding plots of the local eigenvector decomposition of the metric are given in Figure 3, showing both the direction and magnitude of the stretching. Ripley [6] shows a visualization of the SOM reference vectors for

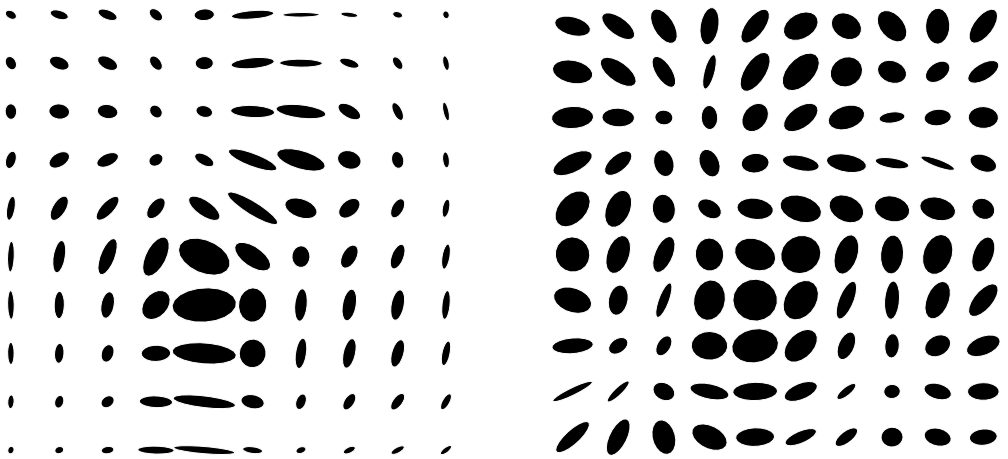

Figure 3: Plots of the local stretching of the latent space, using the ellipse representation discussed in Section 3, for SOM (left) and GTM (right) algorithms.

the crab data using the representation of [7], which corresponds to a discrete approximation to the magnification factors of the GTM model.

**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

[1] C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[2] C. M. Bishop, M. Svensén, and C. K. I. Williams. GTM: the generative topographic mapping, 1997. Accepted for publication in Neural Computation. Available as NCRG/96/015 from `http://www.ncrg.aston.ac.uk/`.

[3] N. A. Campbell and R. J. Mahon. A multi-variate study of variation in two species of rock crab of genus *leptograpsus*. *Australian Journal of Zoology*, 22:417–425, 1974.

[4] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

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

[6] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, 1996.

[7] A. Ultsch. Knowledge extraction from self-organizing neural networks. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification*, pages 301–306, Berlin, 1993. Springer.