A Kernel Approach to Metric Multidimensional Scaling

Andrew Webb

QinetiQ, St. Andrews Road, Malvern, WR14 3PS
webb@signal.qinetiq.com

Abstract. The solution for the parameters of a nonlinear mapping in a metric multidimensional scaling by transformation, in which a stress criterion is optimised, satisfies a nonlinear eigenvector equation, which may be solved iteratively. This can be cast in a kernel-based framework in which the configuration of training samples in the transformation space may be found iteratively by successive linear projections, without the need for gradient calculations. A new data sample can be projected using knowledge of the kernel and the final configuration of data points.
Keywords. multidimensional scaling; kernel representation; nonlinear feature extraction;

1 Introduction

Multidimensional scaling by transformation (MST) describes a class of procedures that implements a nonlinear, dimension-reducing mapping that minimises a criterion, stress, in the output or *representation* space with the aim of retaining the structure and important relationships within the original dataset defined in the data or *observation space*. Often, these mappings are characterised by feed-forward neural network models (for example, multilayer perceptrons [6,7,9], or radial basis functions [8,16]) whose parameters are adjusted to optimise the stress. The stress criterion is strongly related to the conventional metric multidimensional scaling objective function [3], sometimes termed Sammon mappings in the pattern recognition literature after (Sammon, 1969 [14]), with generalisations to include subjective information [8] and class information [5,16].

Many *linear* methods of feature extraction are based on matrices of first and second order statistics. These include transformations based on principal components analysis and linear discriminant analysis and variants [18], and lead to eigenvector / generalised eigenvector equations for the weights of the transformation.

Many methods of *nonlinear* feature extraction, and also methods for discrimination and regression, are linear models. The nonlinear transformation from the data space to the output space is expressed as a linear combination of basis functions. This linear transformation can be determined by first explicitly mapping the input data to the feature space defined by the outputs of the basis functions and then choosing the linear transformation that optimises a criterion defined in

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 452–460, 2002.
© Springer-Verlag Berlin Heidelberg 2002

the output space. This criterion may be, for example, a least squares error, stress (in multidimensional scaling) or variance and, in the case of nonlinear principal components analysis, leads to eigenvector equations for the weights.

An alternative to the approach of *explicitly* mapping data to an intermediate feature space and then determining the nonlinear transformation is to design nonlinear data processing algorithms using linear techniques in an *implicit* feature space induced by kernel functions defined on the data space. These algorithms include support vector machines, kernel principal components analysis [15] and kernel multidimensional scaling [20]. In this paper we show that MST in which a stress function is minimised (and so differs from [20] which is the application of classical scaling in the feature space) can also be expressed in such a framework.

2 Statement of the Problem

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n\}$ denote the dataset of N n-dimensional measurement vectors, $\boldsymbol{x}_i, i = 1, \ldots, N$. Each measurement vector $\boldsymbol{x}_i$ may also have an associated class label, $z_i \in \{1, \ldots, C\}$, the class to which $\boldsymbol{x}_i$ belongs, where C is the number of classes. MST seeks a transformation $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ from n-dimensional space to m-dimensional space ($m < n$) such that a loss function of the form,

$$\sigma^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{ij} (q_{ij} - d_{ij}(\boldsymbol{X}))^2 \tag{1}$$

is minimised, where $\alpha_{ij}, i, j = 1, \ldots, N$ are positive weights that may depend on the classes of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$; the term $d_{ij}(\boldsymbol{X})$ given by

$$d_{ij}(\boldsymbol{X}) = |\boldsymbol{x}_i - \boldsymbol{x}_j|$$

is a distance in the observation space and

$$q_{ij} = |\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{f}(\boldsymbol{x}_j)|$$

is a distance in the representation space. The distances are usually taken to be Euclidean, but other forms may be considered [17]. Minimisation of the loss, σ^2, is performed with respect to the functional form of the transformation, $\boldsymbol{f}$. A convenient choice for $\boldsymbol{f}$ is a basis function expansion of the form

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{l} \boldsymbol{w}_j \phi_j(\boldsymbol{x}) \tag{2}$$

for a set of basis functions, $\{\phi_j, j = 1, \ldots, l\}$, where $\{\boldsymbol{w}_j \in \mathbb{R}^m, j = 1, \ldots, l\}$ is a set of weights optimised by the procedure. Equation (2) may be written

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{3}$$

where $\boldsymbol{W}$ is the $l \times m$ matrix with (i, j) elements w_{ij} and $\boldsymbol{\phi} = [\phi_1(\boldsymbol{x})| \ldots |\phi_l(\boldsymbol{x})]^T$ is the l-dimensional vector of nonlinear responses.

3 Iterative Solution

The minimisation of σ^2 (Equation (1)) with respect to $\boldsymbol{W}$, the parameters of the nonlinear transformation $\boldsymbol{f}$, may be performed using standard nonlinear optimisation schemes [12] that require evaluation of the gradient of σ^2 with respect to $\boldsymbol{W}$. It may also be performed without gradient calculations [16,19] using an *iterative majorisation* approach. Given an initial starting point for the parameters, a *majorising function* is specified that touches the loss function to be minimised at this point, but everywhere else lies above it. The majorising function is simple to minimise and the position of the minimum is used as the starting point for the next iteration (see Figure 1).

It is shown in [16] that the weights $\boldsymbol{W}$ may be determined iteratively using the equation

$$\boldsymbol{S}\boldsymbol{W}^{(t+1)} \;=\; 2\boldsymbol{R}(\boldsymbol{W}^{(t)})\boldsymbol{W}^{(t)} \tag{4}$$

for symmetric $l \times l$ matrices $\boldsymbol{S}$ and $\boldsymbol{R}$ given by

$$\begin{aligned} \boldsymbol{S} \;&=\; 2\sum_i \sum_j \alpha_{ij}(\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j))(\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j))^T \\ &=\; 4N\boldsymbol{\Phi}^T\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{\Phi} \end{aligned} \tag{5}$$

where the α_{ij} are taken to be unity in the last equality and $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\boldsymbol{x}_1)|\ldots|\boldsymbol{\phi}(\boldsymbol{x}_N)]^T$ is the $N \times l$ matrix of nonlinear features; and

$$\boldsymbol{R}(\boldsymbol{W}^{(t)}) \;=\; 2\boldsymbol{\Phi}^T(\tilde{\boldsymbol{C}} - \boldsymbol{C})\boldsymbol{\Phi} \tag{6}$$

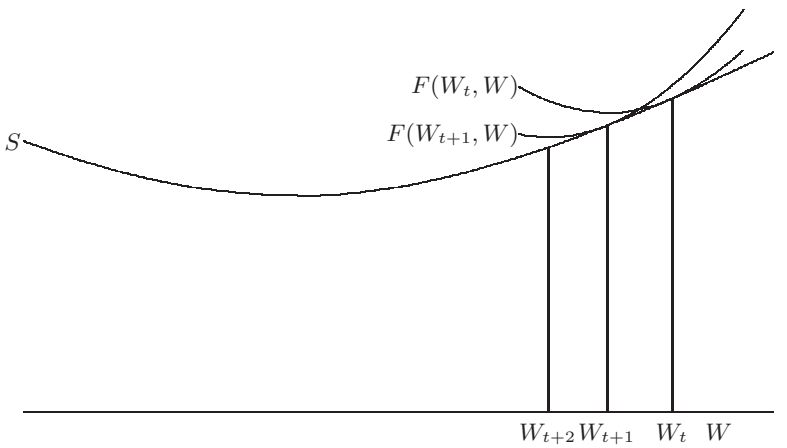

Fig. 1. Illustration of iterative majorisation principle: minimisation of $S(W)$ is achieved through successive minimisations of the majorisation functions, F

where $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{W}^t)$ is the $N \times N$ matrix that depends on the configuration at stage t through the q_{ij} with (i, j) element

$$c_{ij} \;=\; \begin{cases} d_{ij}(\boldsymbol{X})/q_{ij}(\boldsymbol{W}^{(t)}) & q_{ij}(\boldsymbol{W}^{(t)}) > 0 \\ 0 & q_{ij}(\boldsymbol{W}^{(t)}) = 0 \end{cases}$$

and $\tilde{\boldsymbol{C}} = \text{Diag}\{\boldsymbol{C}\boldsymbol{1}\}$, the diagonal matrix with (i, i) element $(\boldsymbol{C}\boldsymbol{1})_i$.

An alternative derivation for the iterative equation for $\boldsymbol{W}$ may be obtained by showing that $\boldsymbol{W}$ satisfies a *nonlinear eigenvector equation* with an algorithm for its solution based on the inverse iteration method for the ordinary eigenvector equation [19].

4 Kernel Representation

4.1 Iterative Solution Using Kernels

We now re-cast the iterative solution for the weights as an iterative solution for the final configuration in the transformed space that requires the specification of a kernel defined on the data space.

Defining $\boldsymbol{H}$ as the $N \times N$ idempotent centring matrix

$$\boldsymbol{H} \;=\; \left(I - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T\right)$$

so that $\boldsymbol{H}\boldsymbol{H} = \boldsymbol{H}$ and $\boldsymbol{H} = \boldsymbol{H}^T$, and using (5) and (6) Equation (4) may be written

$$N(\boldsymbol{H}\boldsymbol{\Phi})^T(\boldsymbol{H}\boldsymbol{\Phi})\boldsymbol{W}^{(t+1)} \;=\; (\boldsymbol{H}\boldsymbol{\Phi})^T(\tilde{\boldsymbol{C}} - \boldsymbol{C})(\boldsymbol{H}\boldsymbol{\Phi})\boldsymbol{W}^{(t)} \tag{7}$$

Writing $\boldsymbol{P}^{(t)} = (\boldsymbol{H}\boldsymbol{\Phi})\boldsymbol{W}^{(t)}$, the $N \times m$ matrix of centred data coordinates in the projected space at stage t, then Equation (7) may be written

$$N(\boldsymbol{H}\boldsymbol{\Phi})^T\boldsymbol{P}^{(t+1)} \;=\; (\boldsymbol{H}\boldsymbol{\Phi})^T(\tilde{\boldsymbol{C}} - \boldsymbol{C})\boldsymbol{P}^{(t)} \tag{8}$$

Taking the pseudo-inverse of $(\boldsymbol{H}\boldsymbol{\Phi})^T$, we can express $\boldsymbol{P}^{(t+1)}$ as

$$N\boldsymbol{P}^{(t+1)} \;=\; [(\boldsymbol{H}\boldsymbol{\Phi})(\boldsymbol{H}\boldsymbol{\Phi})^T]^\dagger(\boldsymbol{H}\boldsymbol{\Phi})(\boldsymbol{H}\boldsymbol{\Phi})^T(\tilde{\boldsymbol{C}} - \boldsymbol{C})\boldsymbol{P}^{(t)} \tag{9}$$

Equation (9) above provides an iterative equation for the coordinates of the transformed data samples. This is the procedure followed in standard approaches to multidimensional scaling [4]. The difference here is that constraints on the form of the nonlinear transformation describing the multidimensional scaling projection are incorporated into the procedure through the $N \times N$ matrix $(\boldsymbol{H}\boldsymbol{\Phi})(\boldsymbol{H}\boldsymbol{\Phi})^T$.

The matrix $(\boldsymbol{H}\boldsymbol{\Phi})(\boldsymbol{H}\boldsymbol{\Phi})^T$ depends on dissimilarities in feature space and may be written

$$(\boldsymbol{H}\boldsymbol{\Phi})(\boldsymbol{H}\boldsymbol{\Phi})^T \;=\; \boldsymbol{H}\boldsymbol{F}\boldsymbol{H} \tag{10}$$

where the $N \times N$ matrix, $\boldsymbol{F}$ has (i,j) element $f_{ij} = -\frac{1}{2}\hat{\delta}_{ij}^2$, where

$$\hat{\delta}_{ij}^2 \;=\; (\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j))^T(\boldsymbol{\phi}(\boldsymbol{x}_i) - \boldsymbol{\phi}(\boldsymbol{x}_j))$$

is the squared Euclidean distance in feature space. Denoting the inner product $\boldsymbol{\phi}^T(\boldsymbol{x}_j)\boldsymbol{\phi}(\boldsymbol{x}_i)$ by the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, then $\boldsymbol{F}$ may be written

$$\boldsymbol{F} \;=\; \boldsymbol{K} - \frac{1}{2}\boldsymbol{k}\boldsymbol{1}^T - \frac{1}{2}\boldsymbol{1}\boldsymbol{k}^T \tag{11}$$

where $\boldsymbol{K}$ is the matrix with (i,j) element $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the ith element of the vector $\boldsymbol{k}$ is $K(\boldsymbol{x}_i, \boldsymbol{x}_i)$. Substituting for $\boldsymbol{F}$ from Equation (11) into (10) gives

$$(\boldsymbol{H\Phi})(\boldsymbol{H\Phi})^T \;=\; \boldsymbol{HKH} \tag{12}$$

Thus, the iterative procedure for the projected data samples depends on the kernel function K, which must satisfy the usual conditions [2] to ensure that it is an inner product on some feature space. Example kernels are polynomials and gaussians.

However, note that if the pseudo-inverse is used to calculate $\boldsymbol{P}^{(t+1)}$ from $\boldsymbol{P}^{(t)}$ in (9), then the only influence of the kernel is through the space spanned by the (non-zero) eigenvectors of $\boldsymbol{HKH}$. That is, if we write $\boldsymbol{HKH}$ as its singular value decomposition, $\boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{U}_r^T$, for $N \times r$ matrix of eigenvectors $\boldsymbol{U}_r$ and $\boldsymbol{\Sigma}_r = \mathrm{Diag}\{\sigma_1, \ldots, \sigma_r\}$ for non-zero singular values $\sigma_i, 1 \leq i \leq r$, then

$$(\boldsymbol{HKH})^\dagger(\boldsymbol{HKH}) \;=\; \boldsymbol{U}_r\boldsymbol{U}_r^T$$

and

$$N\boldsymbol{P}^{(t+1)} \;=\; \boldsymbol{U}_r\boldsymbol{U}_r^T(\tilde{\boldsymbol{C}} - \boldsymbol{C})\boldsymbol{P}^{(t)} \tag{13}$$

Thus, the new coordinates comprise a transformation of the coordinates at step t followed by a projection onto the subspace defined by the columns of $\boldsymbol{U}_r$. The final solution for the coordinates, which we denote by $\tilde{\boldsymbol{P}}$, must lie in the subspace defined by $\boldsymbol{U}_r$.

The matrix $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{P}^{(t)})$ depends on the configuration of points in the transformed space and is given by

$$c_{ij} \;=\; \begin{cases} \alpha_{ij} d_{ij}(\boldsymbol{X})/q_{ij}(\boldsymbol{P}^t) & q_{ij}(\boldsymbol{P}^{(t)}) > 0 \\ 0 & q_{ij}(\boldsymbol{P}^{(t)}) = 0 \end{cases}$$

where $q_{ij}(\boldsymbol{P}^{(t)})$ is the distance between transformed points i and j at stage t.

4.2 Projection of New Data Samples

The iterative procedure described above finds a configuration of the data samples in the transformed space. We would also like to determine where in the transformed space a new sample maps to without having to calculate a weight vector explicitly.

Let $\tilde{\boldsymbol{P}}$ denote the final $N \times m$ matrix of coordinates of the N data samples in the projected m-dimensional space. For a data sample $\boldsymbol{x}$, the new projection, $\boldsymbol{z}$, is given by

$$\boldsymbol{z} \;=\; \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x})$$

and using the solution for the weights ($\boldsymbol{W} = (\boldsymbol{H\Phi})^\dagger \tilde{\boldsymbol{P}}$) we have

$$\begin{aligned} \boldsymbol{z} \;&=\; \tilde{\boldsymbol{P}}^T [(\boldsymbol{H\Phi})^\dagger]^T \boldsymbol{\phi}(\boldsymbol{x}) \\ &=\; \tilde{\boldsymbol{P}}^T [\boldsymbol{HKH}]^\dagger \boldsymbol{H\Phi\phi}(\boldsymbol{x}) \\ &=\; \tilde{\boldsymbol{P}}^T [\boldsymbol{HKH}]^\dagger \boldsymbol{Hl} \end{aligned} \tag{14}$$

where $\boldsymbol{l} = [l_1, \ldots, l_N]^T$ and $l_i = k(\boldsymbol{x}, \boldsymbol{x}_i)$. Thus, a new projection can be expressed using the kernels only, and not the feature space representation and is a weighted sum of the final training data projections, $\tilde{\boldsymbol{P}}$.

5 Choice of Kernel

We adopt a Gaussian kernel of the form

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) \;=\; \exp\left(-\theta(\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\right)$$

with inverse scale parameter θ.

As $\theta \to \infty$, the matrix $\boldsymbol{K} \to \boldsymbol{I}$ and $\boldsymbol{HKH} \to \boldsymbol{H} = \boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T/N$, which is independent of the training data.

As $\theta \to 0$, the matrix $\boldsymbol{K} \to \boldsymbol{1}\boldsymbol{1}^T - \theta\boldsymbol{D}$, where $\boldsymbol{D}$ is the matrix of squared distances in the data space, $\boldsymbol{D}_{ij} = |\boldsymbol{x}_i - \boldsymbol{x}_j|^2$. The matrix $\boldsymbol{HKH} \to -\theta\boldsymbol{HDH}$, showing that the kernel is equivalent to the quadratic kernel[1] $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = |\boldsymbol{x}_i - \boldsymbol{x}_j|^2$, which does not depend on θ.

6 Illustration

The technique is illustrated using a simulated dataset comprising 500 points uniformly distributed over a part-sphere with added noise.

$$\begin{aligned} x_1 \;&=\; A\cos(\psi)\sin(\phi) + n_1 \\ x_2 \;&=\; A\cos(\psi)\cos(\phi) + n_2 \\ x_3 \;&=\; A\sin(\psi) + n_3 \end{aligned}$$

where $\phi = 2\pi u$, $\psi = \sin^{-1}(v(1 + \sin(\psi_{\max})) - 1)$ and u and v are uniformly-distributed on $[0, 1]$; $A = 1$ and n_1, n_2 and n_3 are normally-distributed with variance 0.1. A value of $\pi/4$ was taken for $\psi_{\max}$, so that the surface covers the

[1] Any scaling of a kernel does not affect the final configuration of training samples or the point to which a test pattern is projected.

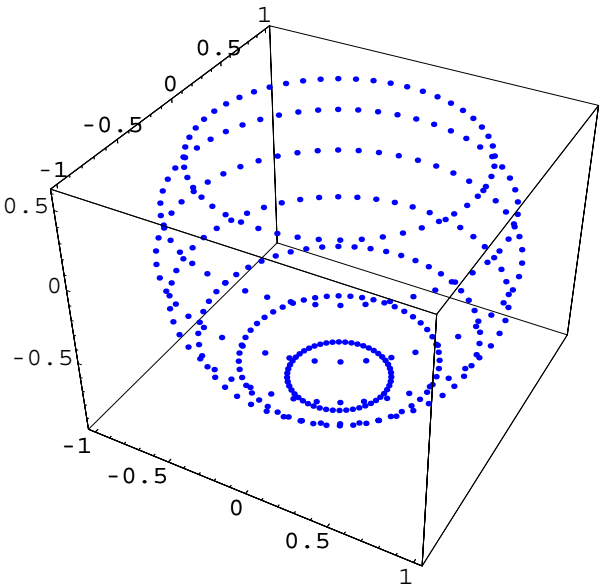

Fig. 2. Lines of latitude on underlying surface

lower hemisphere ($-\pi/2 \leq \psi \leq 0$), together with the upper hemisphere up to a latitude of 45 degrees. Figure 2 shows lines of latitude on the underlying sphere.

The algorithm is trained to convergence on the noisy sphere data and a projection to two dimensions is sought. Figure 3 plots the normalised stress (after the algorithm has converged),

$$\sigma^2 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(q_{ij} - d_{ij}(\boldsymbol{X}))^2}{\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}(\boldsymbol{X})^2} \tag{15}$$

as a function of θ for a test dataset generated using the same distribution as the training data.

For small values of θ, there is very little variation in the stress, showing that a quadratic kernel is close to optimal. Figure 4 give a two-dimensional plot of the training data and the points on the underlying surface (lines of latitude on the sphere) for a value of $\theta = 1.0$. We see that the transformation has 'opened out' the sphere to produce a two dimensional projection.

7 Summary

The main results of this paper can be summarised as follows.

1. The solution for the weights of a generalised linear model that minimise a stress criterion can be obtained using an iterative algorithm (Equations (4)).

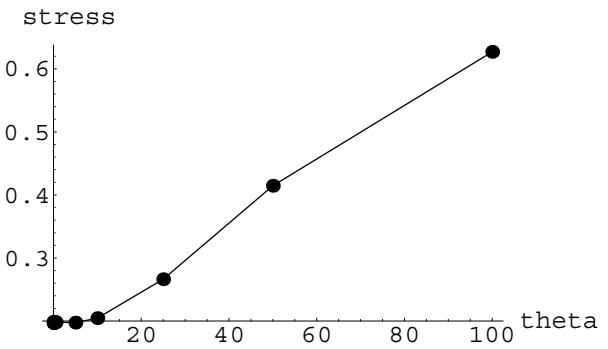

Fig. 3. Normalised stress on test set as a function of θ

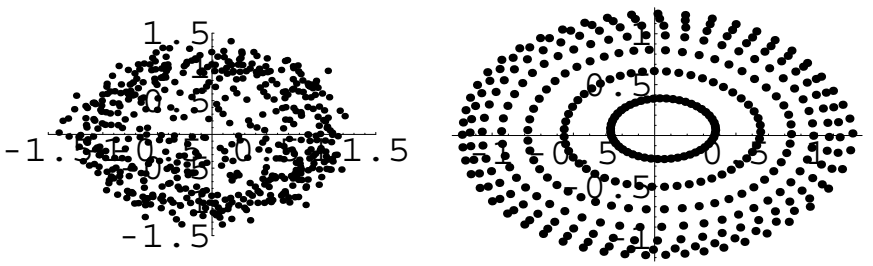

Fig. 4. Projection of training data (left) and points on the underlying surface (right) for the noisy sphere dataset

2. The iterative algorithm for the weights may be re-expressed as an iterative algorithm for the projected data samples (Equation (9)), which depends on a kernel function defined in the data space.
3. For a Gaussian kernel, there is one model selection parameter, θ, that can be determined using a validation set.
4. The projection of new data points may be achieved using the solution for the projected training samples (Equation (14)). The projection is a weighted sum of the projected training samples.

Acknowledgments

This research was sponsored by the UK MOD Corporate Research Programme.

References

1. J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimisation. Theory and Examples.* Springer-Verlag, New York, 2000.
2. N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines.* Cambridge University Press, Cambridge, 2000. 456
3. W. R. Dillon and M. Goldstein. *Multivariate Analysis Methods and Applications.* John Wiley and Sons, New York, 1984. 452
4. W. J. Heiser. Convergent computation by iterative majorization: theory and applications in multidimensional data analysis. In W. J. Krzanowski, editor, *Recent Advances in Descriptive Multivariate Analysis*, pages 157–189. Clarendon Press, Oxford, 1994. 455
5. W. L. G. Koontz and K. Fukunaga. A nonlinear feature extraction algorithm using distance information. *IEEE Transactions on Computers*, 21(1):56–63, 1972. 452
6. B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein. A comparative study of neural network based feature extraction paradigms. *Pattern Recognition Letters*, 120:7–14, 1999. 452
7. B. Lerner, H. Guterman, M. Aladjem, I. Dinstein, and Y. Romem. On pattern classification with Sammon's nonlinear mapping – an experimental study. *Pattern Recognition*, 31(4):371–381, 1998. 452
8. D. Lowe and M. Tipping. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing and Applications*, 4:83–95, 1996. 452
9. J. Mao and A. K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6(2):296–317, 1995. 452
10. R. Mathar and R. Meyer. Algorithms in convex analysis to fit l_p-distance matrices. *Journal of Multivariate Analysis*, 51:102–120, 1994.
11. R. Meyer. Nonlinear eigenvector algorithms for local optimisation in multivariate data analysis. *Linear Algebra and its Applications*, 264:225–246, 1997.
12. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes. The Art of Scientific Computing.* Cambridge University Press, Cambridge, second edition, 1992. 454
13. R. T. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, New Jersey, 1970.
14. J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 18(5):401–409, 1969. 452
15. B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. 453
16. A. R. Webb. Multidimensional scaling by iterative majorisation using radial basis functions. *Pattern Recognition*, 28(5):753–759, 1995. 452, 454
17. A. R. Webb. Radial basis functions for exploratory data analysis: an iterative majorisation approach for Minkowski distances based on multidimensional scaling. *Journal of Classification*, 14(2):249–267, 1997. 453
18. A. R. Webb. *Statistical Pattern Recognition.* Arnold, London, 1999. 452
19. A. R. Webb. A kernel approach to metric multidimensional scaling. In preparation, 2002. 454, 455
20. C. K. I. Williams. On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1/3):11–19, 2001. 453