Chapter 5

Classical Multidimensional Scaling and the Principal Subspace Network

5.1 Introduction

Chapter 3 introduced the 'NEUROSCALE' technique and reviewed recent research effort into exploiting the properties of neural network models for producing similar classes of topographic mappings. These approaches either implemented a Sammon mapping, a nonlinear MDS scheme or in the case of NEUROSCALE, what may be considered to be a hybrid of the two.

In this chapter, a generalisation of the NEUROSCALE concept to the *classical multidimensional scaling* (CMDS) procedure is proposed. Considering the established limited utility of CMDS discussed in Section 2.5.3 previously, this may not appear a fruitful avenue for further investigation. However, analysis of a neural network CMDS scheme reveals a close and illuminating relationship with the field of neural network principal components analysis and the *principal subspace network* [Oja 1989] in particular.

The CMDS procedure is briefly summarised in the next section, where its topographic properties are analysed. This emphasises that for Euclidean input data (dissimilarities), CMDS is equivalent to principal components analysis (PCA). A relative supervision algorithm is then developed in Section 5.3 to train a neural network model to effect the procedure. By considering purely objective ($\alpha = 0$) mappings only, the learning rule that emerges from this approach is then compared with that of the principal subspace network, as described in Section 5.4. Considerable effort is evident in the literature concerning analysis of the global behaviour of the weight update rule for Oja's network, and an important result of this chapter is to demonstrate that such a rule descends the STRESS cost surface of the related CMDS network.

5.2 Classical Multidimensional Scaling Revisited

5.2.1 The CMDS Procedure

Under the metric MDS model, given a $(N \times N)$ matrix of dissimilarities, $\mathbf{\Delta}$, it is desired to produce a set of points, generally in a low dimension, whose inter-point distances, d_{ij}, optimally fit the corresponding dissimilarities, δ_{ij}. One method for generating a set of points $\mathbf{Y}$ is the 'classical' procedure developed by Torgerson [1958], covered previously in section 2.5.3, the optimality properties of which will be discussed shortly. This mapping is effectively linear and makes more constrictive assumptions of the data than the nonmetric methods, but has the advantage of being analytic in its derivation.

The procedure is summarised as follows:

❶ Generate the matrix of squared dissimilarities $\mathbf{\Delta}_2 = \{\delta_{ij}^2\}$.

❷ From this, generate the $(N \times N)$ *double-centred inner-product matrix*, $\mathbf{B}^* = -\frac{1}{2}\mathbf{H}\mathbf{\Delta}_2\mathbf{H}$, where $\mathbf{H}$ is the centring matrix ($\mathbf{H} = \mathbf{I} - \mathbf{1}/N$). If $\mathbf{B}^*$ is being generated from an explicit set of points, rather than from subjective dissimilarities, then it can be given directly by $\mathbf{B}^* = \mathbf{X}\mathbf{X}^{\mathsf{T}}$, under the assumption that the points $\mathbf{X}$ are centred at the origin. (This assumption implies no loss of generality and will be retained for the remainder of this chapter.)

❸ If $\mathbf{\Delta}$ represents a Euclidean distance matrix, then $\mathbf{B}^*$ will be positive semi-definite.

❹ Spectrally decompose $\mathbf{B}^*$ into $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{T}} = \mathbf{Y}\mathbf{Y}^{\mathsf{T}}$, where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{B}^*$ and $\mathbf{U}$ is the corresponding matrix of column eigenvectors. Hence $\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}^{1/2}$ is the configuration of points, still in p dimensions.

❺ Choose the q dimensions (columns of $\mathbf{Y}$) corresponding to the q largest eigenvalues of $\mathbf{B}^*$. This gives a final configuration of $\mathbf{Y}_q = \mathbf{U}_q\mathbf{\Lambda}_q^{1/2}$, which is also centred at the origin.

Whilst MDS methods are traditionally applied to measures of dissimilarity, they may of course be applied directly to a matrix of Euclidean distances, and the classical technique is no exception. In fact, if the dissimilarities in $\mathbf{\Delta}$ correspond to a set of Euclidean distances between data points in some space $\mathbb{R}^p$, then a q-dimensional CMDS configuration is identical to a projection onto the first q principal eigenvectors of the covariance matrix of those data points. Indeed, CMDS is also known as *principal co-ordinates analysis* [Gower 1966]. In this spatial case, the derived matrix $\mathbf{B}^*$ is positive semi-definite (it has rank p) and is equal to $\mathbf{X}\mathbf{X}^{\mathsf{T}}$. Note that $\mathbf{B}^* = \mathbf{X}\mathbf{X}^{\mathsf{T}}$ is a $(N \times N)$ matrix with eigenvectors $\mathbf{U}$ and at most p non-zero eigenvalues, while $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is a $(p \times p)$ matrix with the same set of p eigenvalues but *different* eigenvectors.

5.2.2 Optimality Properties

Because of the equivalence with PCA, the CMDS configuration may be generated as a linear projection of the original data, where $\mathbf{Y}_q = \mathbf{X}\mathbf{W}$ with $\mathbf{W}$ a $p \times q$ weight matrix whose columns are the principal eigenvectors of the covariance matrix of $\mathbf{X}$. In terms of a topographic property, the CMDS linear projection solution has been previously shown [Mardia, Kent, and Bibby 1979] to be optimal with respect to the distance-retaining STRESS measure

$$E = \sum_i \sum_j (d_{ij}^*)^2 - (d_{ij})^2, \tag{5.1}$$

under the constraint that the columns of $\mathbf{W}$ are orthonormal. The error terms here need not be squared as the orthonormality of the projection ensures that all inter-point distances are contracted. This optimality property was shown by Mardia et al. [1979], but a more efficient derivation is given below.

Proof. Let $\mathbf{U} = (\mathbf{W}, \mathbf{V})$ where $\mathbf{U}$ is orthogonal such that $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$ and $\mathbf{W}$ is the projection matrix whose q columns are orthonormal. $\mathbf{V}$ is a matrix whose $(p-q)$ columns complete the orthonormal set in $\mathbf{U}$. Then

$$\begin{aligned}(d^*_{ij})^2 &= (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j),\\ &= \left[\mathbf{U}^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right]^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j),\\ &= \sum_{k=1}^{p}\left[\mathbf{u}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right]^2,\end{aligned}$$

and in corresponding form,

$$(d_{ij})^2 = \sum_{k=1}^{q}\left[\mathbf{w}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right]^2.$$

Thus E can be given by:

$$\begin{aligned}E &= \sum_{ij}\sum_{k=q+1}^{p}\left[\mathbf{v}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\right]^2,\\ &= \sum_{ij}\sum_{k=q+1}^{p}\mathbf{v}_k^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\mathbf{v}_k,\\ &= \sum_{k=q+1}^{p}\mathbf{v}_k^{\mathrm{T}}\left[\sum_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\right]\mathbf{v}_k,\\ &= 2N^2\sum_{k=q+1}^{p}\mathbf{v}_k^{\mathrm{T}}\mathbf{C}\mathbf{v}_k,\end{aligned}$$

where $\mathbf{C} = E[\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ is the sample covariance matrix of the points $\mathbf{x}$.

For orthonormal $\mathbf{v}_k$, this is minimised by the $\mathbf{v}_k$ spanning the space of the $(p-q)$ eigenvectors of $\mathbf{C}$ corresponding to the smallest eigenvalues. Thus, the *optimal* $\mathbf{W}$ is a projection in the q largest principal components of the data $\mathbf{X}$, which gives the Classical MDS solution in q dimensions.
□

However, the apparent topographic property of the projection can be misleading. The distance term $(d^*_{ij})^2$ from equation (5.1) can be expanded as

$$\begin{aligned}(d^*_{ij})^2 &= (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}(\mathbf{x}_i - \mathbf{x}_j)\\ &= \mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i + \mathbf{x}_j^{\mathrm{T}}\mathbf{x}_j - 2\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j,\end{aligned}$$

and similarly for $(d_{ij})^2$. Since the points are centred, $\sum_{ij}\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j = 0$ (and correspondingly for $\mathbf{y}$), and the expression for E becomes

$$E = 2N\sum_i(\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i - \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i).$$

Minimisation of this error effectively results in a maximisation of the $\sum_i^N \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_i$ term (under the constraint of orthonormality of the columns of $\mathbf{W}$), and this is evidently a *variance maximisation* procedure. A standard result in this respect is that the optimal $\mathbf{W}$ be the matrix of principal eigenvectors of the covariance matrix of the data in $\mathbf{X}$.

There is in fact a further cost measure that the CMDS configuration minimises [Mardia 1978]. Having generated the matrix $\mathbf{B}^*$, a q-dimensional CMDS solution, $\mathbf{Y}$, can be shown to produce an optimal (in the least squares sense) fit to the inner-product matrix. That is,

$$E = \mathrm{tr}\left[(\mathbf{B}^* - \mathbf{Y}\mathbf{Y}^{\mathrm{T}})^2\right] \tag{5.2}$$

is minimised.[1]

Proof. Equation (5.2) may be minimised with respect to $\mathbf{Y}$ by differentiating to find the stationary points:

$$\frac{\partial E}{\partial \mathbf{Y}} = -4\mathbf{B}^*\mathbf{Y} + 4\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\mathbf{Y}.$$

So at the stationary points,

$$(\mathbf{B}^* - \mathbf{Y}\mathbf{Y}^{\mathrm{T}})\mathbf{Y} = \mathbf{0}. \tag{5.3}$$

Equation (5.3) can obviously be satisfied if the inner product matrices are identical, but this solution is unattainable if $\mathbf{Y}$ is of lower rank than $\mathbf{B}^*$ (which will generally be the case since $p > q$). However, other solutions to (5.3) are given by:

$$\mathbf{Y} = \mathbf{U}_q\mathbf{\Lambda}_q^{1/2}\mathbf{R}, \tag{5.4}$$

where $\mathbf{U}_q$ is a matrix whose q column vectors are eigenvectors of $\mathbf{B}^*$ and $\mathbf{\Lambda}_q$ is a diagonal matrix containing their corresponding eigenvalues. $\mathbf{R}$ is an arbitrary, orthogonal, rotation matrix, such that $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$. That equation (5.4) is a solution is clear from:

$$\begin{aligned}(\mathbf{B}^* - \mathbf{Y}\mathbf{Y}^{\mathrm{T}})\mathbf{Y} &= (\mathbf{B}^* - \mathbf{U}_q\mathbf{\Lambda}_q^{1/2}\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{\Lambda}_q^{1/2}\mathbf{U}_q^{\mathrm{T}})\mathbf{U}_q\mathbf{\Lambda}_q^{1/2}\mathbf{R} \\ &= \mathbf{B}^*\mathbf{U}_q\mathbf{\Lambda}_q^{1/2}\mathbf{R} - \mathbf{U}_q\mathbf{\Lambda}_q\mathbf{U}_q^{\mathrm{T}}\mathbf{U}_q\mathbf{\Lambda}_q^{1/2}\mathbf{R}, \\ &= \mathbf{U}_q\mathbf{\Lambda}_q^{3/2}\mathbf{R} - \mathbf{U}_q\mathbf{\Lambda}_q^{3/2}\mathbf{R} = \mathbf{0}.\end{aligned}$$

Then the error is tr $\left[\mathbf{B}^* - \mathbf{U}_q\mathbf{\Lambda}_q\mathbf{U}_q^{\mathrm{T}}\right]^2$, and by expanding it is easy to show that this is minimised when $\mathbf{\Lambda}_q$ contains the q largest eigenvalues of $\mathbf{B}^*$, and $\mathbf{U}_q$ the corresponding eigenvectors. To within the equivalence defined by the rotation $\mathbf{R}$, this represents a unique global minimum. Other combinations of eigenvectors in $\mathbf{U}_q$ are saddle points on the cost surface.
□

This property is particularly appropriate to the context of NEUROSCALE and related models, as the trace notation of equation (5.2) may be expanded as a sum of individual terms thus:

$$E = \sum_i^N \sum_j^N (b_{ij}^* - b_{ij})^2. \tag{5.5}$$

This may be seen to be of very similar form to the Sammon STRESS measure of equation (2.3). In fact, the inter-point distance between two points i and j has now been replaced by their respective inner, or scalar, product. (Compare this with the distinction between the methods for calculation of hidden unit activations in MLPs and RBFs.) In the same manner that a neural network-based variant of the Sammon mapping was developed from that latter measure, a similar approach may be adopted in order to naively derive a relative supervision algorithm for a CMDS neural network implementation.

5.3 A Neural Network CMDS Transformation

By choosing a linear, single-layer, neural network to effect the CMDS transformation, such that $\mathbf{y}_i = \mathbf{W}^{\mathrm{T}}\mathbf{x}_i$, the cost function of equation (5.5) becomes

$$E = \frac{1}{4N^2}\sum_i^N \sum_j^N (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j - \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j)^2, \tag{5.6}$$

[1]This use of the trace notation in error measures is convenient since if $\mathbf{E}$ is a (not necessarily square) matrix of residual errors then tr $[\mathbf{E}^{\mathrm{T}}\mathbf{E}]$ = tr $[\mathbf{E}\mathbf{E}^{\mathrm{T}}] = \sum_i \sum_j e_{ij}^2$, and is thus the sum-of-squares error function.

with the constant term introduced for later convenience. This simple network is illustrated in figure 5.1 below.

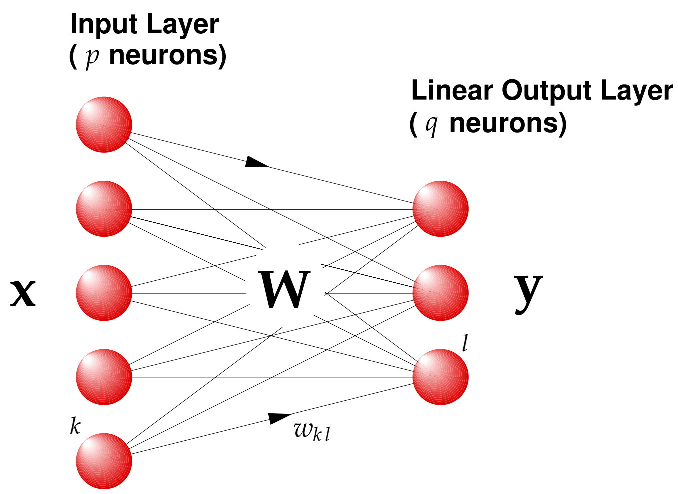

Figure 5.1: A simple linear neural network to perform CMDS.

The derivative of this error measure with respect to a weight w_{kl} from figure 5.1 is given by:

$$\frac{\partial E}{\partial w_{kl}} = -\frac{1}{N^2}\sum_i \sum_j (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j - \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j) y_{jl} x_{ik}. \tag{5.7}$$

These individual weight derivatives may be combined into a matrix $\partial E/\partial \mathbf{W}$ thus:

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{W}} &= -\frac{1}{N^2}\sum_i \sum_j (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j - \mathbf{y}_i^{\mathrm{T}}\mathbf{y}_j)\mathbf{x}_i\mathbf{y}_j^{\mathrm{T}} \quad \text{and expanding for } \mathbf{y}_i, \mathbf{y}_j, \\
&= -\frac{1}{N^2}\sum_i \sum_j (\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j - \mathbf{x}_i^{\mathrm{T}}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{x}_j)\mathbf{x}_i\mathbf{x}_j^{\mathrm{T}}\mathbf{W}, \\
&= -\frac{1}{N^2}\sum_i \sum_j [\mathbf{x}_i^{\mathrm{T}}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{x}_j]\mathbf{x}_i\mathbf{x}_j^{\mathrm{T}}\mathbf{W}, \\
&= -\frac{1}{N^2}\sum_i \sum_j \mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}}\mathbf{W}.
\end{aligned} \tag{5.8}$$

Finally, summing separately over j and then i in equation (5.8) gives:

$$\frac{\partial E}{\partial \mathbf{W}} = -\mathbf{C}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{C}\mathbf{W}, \tag{5.9}$$

where $\mathbf{C}$ is the sample covariance matrix of the data $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. This expression might also have been derived more simply by a direct differentiation of equation (5.2), but this would have obscured the connection with the NEUROSCALE algorithm and the relative supervision approach.[2] Deriving it from equation (5.5) emphasises the correspondence between the two approaches. Thus, applying a steepest-gradient minimisation method to the training of the network leads to a learning rule:

$$\Delta \mathbf{W}_{cmds} = \eta \mathbf{C}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{C}\mathbf{W}, \tag{5.10}$$

where η is a small constant.

[2]Note that Sammon's algorithm cannot be derived from a trace error measure as the matrix of distances (or even squared distances) cannot be expressed as a function of the data matrix $\mathbf{X}$.

Because of the previously outlined properties of CMDS, a network trained in such a manner will produce a (rotated) principal component projection of the input data. In this respect, there is already a considerable body of work concerned with the implementation of *principal components analysis* (PCA) with neural networks. These are linear networks, usually trained via (nearly) local Hebbian algorithms, to generate a reduced dimension principal component projection of the input data. Notable developments are the single principal component analyser [Oja 1982], its generalisation to the Principal Subspace Network [Oja 1989] which extracts the q-dimensional *principal subspace*, and the Generalised Hebbian Algorithm [Sanger 1989], which extracts the actual q principal components themselves. A good overview of these algorithms, the relationships between them and some of the theoretical issues that they raise, is given by Baldi and Hornik [1995].

Further investigation of the learning dynamics of the neural network CMDS algorithm implied by equation (5.10) indeed reveals a close relationship to the Principal Subspace Network, and both networks are considered in more detail in the next section. This will review and clarify some of the properties of the weight adaptation rule of the latter and will show that such a rule minimises the cost function for a neural network implementation of the CMDS procedure.

5.4 The Principal Subspace Network

Of the various neural network implementations for PCA, the Principal Subspace Network (PSN) of Oja [1989] has perhaps been the most studied. The PSN is a single-layer neural network with linear output activation functions, such that $\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x}$, and is of identical architecture to the network depicted in figure 5.1. Its simple weight adaptation rule and associated qualities of symmetry and homogeneity are attractive from the neuro-biological perspective.

The PSN is generally trained on-line, pattern by pattern, with adjustments made to the weights according to:

$$\Delta\mathbf{W} = \eta(\mathbf{x} - \mathbf{W}\mathbf{y})\mathbf{y}^{\mathrm{T}}, \tag{5.11}$$

which is often known as 'Oja's rule'. The first term represents a Hebbian weight reinforcement, and the second term is a weight normalisation term which constrains $\mathbf{W}^{\mathrm{T}}\mathbf{W} \approx \mathbf{I}$.

It is clearer, and facilitates the analysis, to consider the averaged form of the weight update rule (which assumes that the input data is drawn from a stationary distribution and that $\mathbf{W}$ changes slowly):

$$\Delta\mathbf{W}_{psn} = \eta\,(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{C}\mathbf{W}, \tag{5.12}$$

where $\mathbf{C} = E[\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ is the sample covariance matrix of the input data, which are again assumed to be centred at the origin, and η is a small constant. This rule, and its associated differential equation, has been analysed in [Williams 1985; Oja 1992; Karhunen 1994] with respect to the asymptotic stability about its fixed points. Although any $\mathbf{W}$ comprised of eigenvectors of $\mathbf{C}$ are fixed points of equation (5.12), only those $\mathbf{W}$ whose column vectors span the *principal* subspace of $\mathbf{C}$ are stable. (The other combinations of eigenvectors represent saddle points.)

This analysis describes the local behaviour of the PSN algorithm about its fixed points in weight space. Although experimental evidence implied that the algorithm was also globally convergent, only recently has any theoretical insight been provided. Yan, Helmke, and Moore [1994] presented results defining global convergence behaviour based on analysis of the related Riccati differential equation, while Plumbley [1995] derived three Lyapunov functions that described the domains of convergence.

One particular approach to global analysis is to consider that if the PSN rule could be interpreted as the gradient of a particular cost function, this would define the behaviour of the rule through all space with respect to that cost surface. Unfortunately, equation (5.12) cannot be formulated in such a manner, as its implied Hessian is asymmetric [Baldi and Hornik 1995]. However, Oja's rule has been shown to be a gradient system in the transformed co-ordinates $\mathbf{Z} = \mathbf{C}^{1/2}\mathbf{W}$ [Wyatt and Elfadel 1995], although the earliest such gradient relationship was exposed by Xu [1993] with further appreciation afforded by Karhunen and Joutsensalo [1994]. They showed that equation (5.12) is a *descent direction* on the surface of a cost function of the form

$$E_{rec} = \sum_i \| \mathbf{x}_i - \mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{x}_i \|^2 \,. \tag{5.13}$$

This measure can be interpreted as the mean-square error at the output of an auto-associative network with a single hidden layer of q units, where the weight matrix from the hidden layer to the output is constrained to be the transpose of that from the inputs to the hidden layer. As such, equation (5.13) represents an optimal linear reconstruction criterion whose implied error surface is well characterised [Baldi and Hornik 1989], and it is known that this measure is minimised when the hidden-layer space is the principal subspace of the input data. Thus, if $\mathbf{W}^{(t+1)} = \mathbf{W}^t + \Delta\mathbf{W}_{psn}$ implies that E_{rec} is monotonically decreasing with t, then Oja's rule will converge on the principal subspace.

That this is so may be shown as follows (in a more concise and intuitive form than that given by Xu [1993]).

Equation (5.13) may be expressed in matrix trace notation as:

$$E_{rec} = \frac{1}{2N}\mathrm{tr}\left[(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^{\mathrm{T}})\right], \tag{5.14}$$

with the constant term again introduced for convenience.

Minimising this cost by taking a step $\Delta\mathbf{W}_{rec}$ in the direction of steepest descent gives

$$\begin{aligned}
\Delta\mathbf{W}_{rec} &= -\eta\frac{\partial E}{\partial \mathbf{W}}, \\
&= \frac{\eta}{N}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{W} + \frac{\eta}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{W}, \\
&= \eta\mathbf{V}\mathbf{C}\mathbf{W} + \eta\mathbf{C}\mathbf{V}\mathbf{W},
\end{aligned} \tag{5.15}$$

where $\mathbf{V} = (\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})$. It is evident that the first term is simply the learning rule for the PSN, and it is this term that dominates equation (5.15) after the initial few training steps. This may be understood by noticing that the second term $\mathbf{X}^{\mathrm{T}}\mathbf{X}(\mathbf{I} - \mathbf{W}\mathbf{W}^{\mathrm{T}})\mathbf{W} \to \mathbf{0}$ as $\mathbf{W}^{\mathrm{T}}\mathbf{W} \to \mathbf{I}$, whilst the first (Oja) term only tends to the null matrix as $\mathbf{W}^{\mathrm{T}}\mathbf{W} \to \mathbf{I}$ *and* the columns of $\mathbf{W}$ span the principal eigenspace of $\mathbf{C}$. The imbalance between these two terms is accentuated as $(p - q)$ increases. If the columns of $\mathbf{W}$ are initially orthogonal, the off-line (batch) form of Oja's update rule and the gradient-based minimisation of (5.13) become almost equivalent. (To the extent that Oja's rule is formulated to retain $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I} + O(\eta^2)$, and while $\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}$ the non-PSN term is zero.)

It is possible to determine whether the weight adaptation prescribed by the PSN rule (the first term) will increase or reduce the reconstruction error (5.13) by calculating its scalar product, r_1, with the (downhill) gradient $\Delta\mathbf{W}_{rec}$ on the cost surface of the associated error measure. A positive value of scalar product implies that the weight adjustment according to the PSN rule is a *descent direction* on that surface. The quantity r_1 is given by $\mathrm{vec}[\eta\mathbf{V}\mathbf{C}\mathbf{W}]^{\mathrm{T}}\mathrm{vec}[\Delta\mathbf{W}_{rec}]$, where 'vec[·]' is the operator that converts a matrix into a vector by stacking its columns one above each other, and thus has the property that $\mathrm{vec}[\mathbf{A}]^{\mathrm{T}}\mathrm{vec}[\mathbf{A}] = \mathrm{tr}\,[\mathbf{A}^{\mathrm{T}}\mathbf{A}]$. (This operation serves to convert the parameters in the matrix $\mathbf{W}$ into a single vector in the objective space of equation (5.13).) Therefore

$$\begin{aligned}
r_1 &= \eta\,\mathrm{tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{C}\mathbf{V}(\Delta\mathbf{W}_{rec})\right], \\
&= \eta^2\,\mathrm{tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{C}\mathbf{V}(\mathbf{V}\mathbf{C}\mathbf{W} + \mathbf{C}\mathbf{V}\mathbf{W})\right], \\
&= \eta^2\,\mathrm{tr}\left[\mathbf{W}^{\mathrm{T}}\mathbf{C}\mathbf{V}\mathbf{V}\mathbf{C}\mathbf{W} + \mathbf{W}^{\mathrm{T}}\mathbf{C}\mathbf{V}\mathbf{C}\mathbf{V}\mathbf{W}\right], \\
&= \eta^2\sum_i^q \mathbf{w}_i^{\mathrm{T}}[\mathbf{C}\mathbf{V}^2\mathbf{C} + (\mathbf{C}\mathbf{V})^2]\mathbf{w}_i.
\end{aligned} \tag{5.16}$$

The matrix $\mathbf{CV^2C}$ is clearly positive (semi-)definite (indeed, it must be so as it represents the inner-product of the PSN term with itself), but more importantly, so is the second matrix $(\mathbf{CV})^2$. Even though that matrix is asymmetric, it may still possess a definite-ness property, which is shown as follows.

Proof. Let $(\mathbf{CV})$ have eigenvalues λ_i with corresponding eigenvectors $\mathbf{u}_i = \mathbf{x}_i + i\mathbf{y}_i$. Then $(\mathbf{VC})$ has the same eigenvalues λ_i and eigenvector $\mathbf{Vu}_i$ [e.g. see Mardia et al. 1979, pp 468].

Consider a single eigenvalue $\lambda = a + ib$:

$$\begin{aligned}(\mathbf{CV})\mathbf{u} &= (\mathbf{CV})\mathbf{x} + i(\mathbf{CV})\mathbf{y},\\ &= a\mathbf{x} + ia\mathbf{y} + ib\mathbf{x} - b\mathbf{y}.\end{aligned}$$

Equating the real parts gives:

$$(\mathbf{CV})\mathbf{x} = a\mathbf{x} - b\mathbf{y}. \tag{5.17}$$

Similarly,

$$\begin{aligned}(\mathbf{VC})\mathbf{Vu} &= (\mathbf{VC})\mathbf{Vx} + i(\mathbf{VC})\mathbf{Vy},\\ &= a\mathbf{Vx} + ia\mathbf{Vy} + ib\mathbf{Vx} - b\mathbf{Vy},\end{aligned}$$

and equating imaginary parts gives:

$$(\mathbf{VC})\mathbf{Vy} = a\mathbf{Vy} + b\mathbf{Vx}. \tag{5.18}$$

From (5.17), premultiplying by $\mathbf{y}^{\mathrm{T}}\mathbf{V}$ gives:

$$\mathbf{y}^{\mathrm{T}}\mathbf{V}(\mathbf{CV})\mathbf{x} = a\mathbf{y}^{\mathrm{T}}\mathbf{Vx} - b\mathbf{y}^{\mathrm{T}}\mathbf{Vy}. \tag{5.19}$$

From (5.18), premultiplying by $\mathbf{x}^{\mathrm{T}}$ gives:

$$\mathbf{x}^{\mathrm{T}}(\mathbf{VC})\mathbf{Vy} = a\mathbf{x}^{\mathrm{T}}\mathbf{Vy} + b\mathbf{x}^{\mathrm{T}}\mathbf{Vx}. \tag{5.20}$$

Because $(\mathbf{VC})^{\mathrm{T}} = (\mathbf{CV})$ and $\mathbf{V}^{\mathrm{T}} = \mathbf{V}$, transposing (5.20) and subtracting from (5.19) gives $b = 0$.

Therefore all the eigenvalues of $(\mathbf{CV})$ are real, and the eigenvalues of $(\mathbf{CV})^2$ are all non-negative.
□

The definite-ness properties of both these terms implies that r_1 must be positive *for all possible values of* $\Delta\mathbf{W}$ (except for the highly pathological case where all the $\mathbf{w}_i$ initially lie in the null-space of $\mathbf{C}$) and so the the weight adaptation for the principal subspace network performs gradient descent on the surface defined by the cost function of equation (5.13), for sufficiently small η.

5.5 The Relationship between the PSN and CMDS Learning Rules

In addition to the result of the previous section, it may also be shown that the weight update for the PSN minimises the cost function, derived from equation (5.5), for a neural network implementation of CMDS. That the PSN and CMDS are equivalent (to within a rigid rotation) is self-evident, as both networks generate a principal subspace of the input data. However, comparison of the averaged forms of the learning rules offers further insight into the global behaviour of the PSN rule.

It was shown from equation (5.10) in section 5.3 that the weight update rule for the CMDS network was $\eta\mathbf{C}(\mathbf{I} - \mathbf{WW}^{\mathrm{T}})\mathbf{CW}$. This is similar, by a factor $\mathbf{C}$, to the rule for the PSN of equation (5.12). To investigate the behaviour of the PSN rule, the inner-product of its weight update and that of the direction of steepest descent for the CMDS cost function (5.5) is again calculated. The inner-product is simply $r_2 = \mathrm{vec}[\Delta\mathbf{W}_{psn}]^{\mathsf{T}}\mathrm{vec}[\Delta\mathbf{W}_{cmds}]$. Now,

$$\begin{aligned} r_2 &= \mathrm{vec}[\Delta\mathbf{W}_{psn}]^{\mathsf{T}}\mathrm{vec}[\mathbf{C}\Delta\mathbf{W}_{psn}], \\ &= \mathrm{tr}\left[\Delta\mathbf{W}_{psn}^{\mathsf{T}}\mathbf{C}\Delta\mathbf{W}_{psn}\right], \\ &= \sum_i^q \Delta\mathbf{w}_i^{\mathsf{T}}\mathbf{C}\Delta\mathbf{w}_i. \end{aligned} \tag{5.21}$$

Since the covariance matrix $\mathbf{C}$ is positive (semi-)definite, the value of r_2 will be positive for all $\Delta\mathbf{W}$, excepting again the case where all $\mathbf{w}_i$ lie in the null space of $\mathbf{C}$. Therefore, for sufficiently small η, the averaged weight adaptation for the PSN minimises the cost function of equation (5.5).

To reveal the underlying differences between the two algorithms, consider the perturbation of a single weight vector $\Delta\mathbf{w}$ due to the PSN rule. It can be seen that in the CMDS network, pre-multiplication by the covariance matrix $\mathbf{C}$ will move $\Delta\mathbf{w}$ more into the direction of the principal eigenvector. The weight change determined by Oja's rule may be expressed as a weighted sum of the eigenvectors of $\mathbf{C}$ such that $\Delta\mathbf{w} = \sum_i^p \beta_i\mathbf{u}_i$, where the β_i are some constants. Then $\mathbf{C}\Delta\mathbf{w} = \sum_i^p \lambda_i\beta_i\mathbf{u}_i$. The eigenvector components of $\Delta\mathbf{w}$ are multiplied by their corresponding eigenvalues, and so $\Delta\mathbf{w}$ will further approach the direction of the principal eigenvector. However, the weight adaptation $\mathbf{C}\Delta\mathbf{w}$ will no longer retain orthogonality of the columns of $\mathbf{W}$ in the same manner as the PSN algorithm. For the PSN, if

$$\begin{aligned} \mathbf{W}(t)^{\mathsf{T}}\mathbf{W}(t) &= \mathbf{I}, \quad \text{and} \\ \mathbf{W}(t+1) &= \mathbf{W}(t) + \Delta\mathbf{W}_{psn}, \quad \text{then} \\ \mathbf{W}(t+1)^{\mathsf{T}}\mathbf{W}(t+1) &\approx \mathbf{I}, \end{aligned}$$

assuming that η is small and so ignoring the term in $O(\eta^2)$. In contrast, for the CMDS network

$$\begin{aligned} \mathbf{W}(t+1) &= \mathbf{W}(t) + \mathbf{C}\Delta\mathbf{W}_{psn} \quad \text{giving} \\ \mathbf{W}(k+1)^{\mathsf{T}}\mathbf{W}(k+1) &\approx \mathbf{I} + 2\eta\mathbf{W}^{\mathsf{T}}\mathbf{C}(\mathbf{W}\mathbf{W}^{\mathsf{T}} - \mathbf{I})\mathbf{C}\mathbf{W}. \end{aligned}$$

The second term will, however, tend to the null matrix as the columns of $\mathbf{W}$ approach the principal eigenvectors of $\mathbf{C}$, and so as equation (5.5) is minimised, $\mathbf{W}^{\mathsf{T}}\mathbf{W}$ will converge on the identity matrix.

5.6 Conclusions

In the MDS field and for application to dissimilarity data, the classical technique is now rarely employed, apart from its use as a 'first guess' to initialise the iterative procedures of other more effective schemes. For application to explicit spatial data, as a form of 'topographic' mapping, its utility is even more limited, as it is well known that in such applications, it is exactly equivalent to principal components analysis. Its asserted topographic property, that it minimises the measure $\sum_i \sum_j (d^*_{ij})^2 - (d_{ij})^2$, is, in truth, simply constrained variance maximisation in disguise — or PCA once more .

Development of a relatively supervised neural network CMDS model is therefore obviously of limited practical value. Nevertheless, it is of interest to compare such a scheme to established neural network strategies for implementing principal component projections.

One such approach, Oja's principal subspace network, has seen significant study with respect to the convergence properties of its learning rule. While experiment reveals that the network consistently extracts the principal subspace of the input data, significant recent research effort has been directed towards further understanding of the global behaviour of Oja's rule. One initial approach was to show that the learning rule descends a cost function implied by a linear reconstruction criterion. In this chapter, it has been revealed that in addition, Oja's learning rule descends the cost function associated with the CMDS neural network model, $\mathrm{tr}\left[(\mathbf{B}^* - \mathbf{Y}\mathbf{Y}^{\mathrm{T}})^2\right]$. This result affords further appreciation of the dynamics of the PSN, although the NEUROSCALE CMDS approach cannot be considered a realistic competitor to that network as the learning rule requires a double loop over the input patterns and cannot be implemented on-line.